

StorageTek Enterprise Library Software

Guide de récupération après sinistre et de gestion des données hors site

Version 7.3

E63460-02

Septembre 2016

StorageTek Enterprise Library Software

Guide de récupération après sinistre et de gestion des données hors site

E63460-02

Copyright © 2015, 2016, Oracle et/ou ses affiliés. Tous droits réservés.

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf stipulation expresse de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, accorder de licence, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l'accompagne, est concédé sous licence au Gouvernement des Etats-Unis, ou à toute entité qui délivre la licence de ce logiciel ou l'utilise pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique :

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer un risque de dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour des applications dangereuses.

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

Intel et Intel Xeon sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. UNIX est une marque déposée de The Open Group.

Ce logiciel ou matériel et la documentation qui l'accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers, sauf mention contraire stipulée dans un contrat entre vous et Oracle. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'accès à des contenus, produits ou services tiers, ou à leur utilisation, sauf mention contraire stipulée dans un contrat entre vous et Oracle.

Table des matières

| | |
|--|----|
| Préface | 13 |
| Public | 13 |
| Prérequis | 13 |
| A propos de ce manuel | 13 |
| Choix d'une solution adaptée à mes besoins | 15 |
| Accessibilité de la documentation | 15 |
| Nouveautés | 17 |
| 1. Introduction à la récupération après sinistre | 19 |
| Définition de l'objectif de délai de récupération (RTO, Recovery Time Objective) | 20 |
| Définition de l'objectif du point de récupération (RPO) | 21 |
| Traitement des pannes temporaires | 22 |
| Concept clé : récupération du point de synchronisation | 22 |
| Mise en relation d'un RPO avec la récupération du point de synchronisation | 23 |
| Planification de la haute disponibilité des données (D-HA) | 24 |
| Bande physique hautement disponible | 25 |
| Bande virtuelle hautement disponible | 26 |
| D-HA et récupération du point de synchronisation | 29 |
| Exécution d'une réelle récupération après sinistre | 29 |
| Planification de tests DR | 30 |
| Déplacement de données pour les tests DR | 31 |
| Tests DR avec exportation/importation physique | 31 |
| Tests DR avec CDRT | 33 |
| Tests DR avec fonction CTR (Cross-Tapeplex Replication) de VSM | 35 |
| 2. Exportations et importations physiques | 37 |
| Exportation et importation par classe de gestion | 38 |
| Exemple : exportation par classe de gestion à partir du système VSM source | 38 |
| Exemple : importation par classe de gestion dans le système VSM cible | 39 |
| Exportation et importation par classe de stockage | 41 |

| | |
|--|-----------|
| Exemple : exportation par classe de stockage à partir du système VSM source | 41 |
| Exemple : importation par classe de stockage dans le système VSM cible | 43 |
| Exportation/importation dans une VLE | 45 |
| 3. Utilisation de la fonction d'archivage sécurisé externe d'ELS | 47 |
| Préparation de l'archivage sécurisé externe d'ELS | 47 |
| Création de MVC pour la récupération après sinistre et la conservation à long terme | 49 |
| Considérations relatives à DELSCR lors de l'utilisation de la fonction d'archivage sécurisé | 50 |
| Opérations exécutées lors du renvoi des volumes archivés de façon sécurisée vers l'ACS | 50 |
| Archivage sécurisé de MVC pour la récupération après sinistre | 51 |
| Archivage sécurisé DR de base avec des MVC | 51 |
| Etape 1 – Création de VTV/MVC d'archivage sécurisé | 51 |
| Etape 2 – Exportation des MVC d'archivage sécurisé | 51 |
| Etape 3 – Ecriture de jeux de données supplémentaires sur la bande du fichier manifeste (facultatif) | 53 |
| Etape 4 – Ejection de MVC d'archivage sécurisé | 53 |
| Etape 5 - Ejection de volumes natifs (y compris de la bande du fichier manifeste) | 53 |
| Etape 6 – Création d'une liste déroulante de volumes pour renvoi à partir de l'espace d'archivage sécurisé | 54 |
| Etape 7 – Préparation des MVC archivées pour le renvoi | 55 |
| Etape 8 – Préparation des volumes natifs archivés pour le renvoi | 56 |
| Etape 9 - Entrée de volumes renvoyés | 57 |
| Archivage sécurisé DR pendant plusieurs semaines | 58 |
| Archivage sécurisé DR avec des MVC dans une bibliothèque distante | 59 |
| Archivage sécurisé de MVC pour la conservation à long terme | 61 |
| Ejection de volumes spécifiques vers un espace d'archivage sécurisé local (sol) | 61 |
| 4. Utilisation de la réplication entre TapePlex dans une solution de récupération après sinistre | 65 |
| Fonctionnement de CTR | 65 |
| Considérations relatives aux VTV CTR en lecture seule | 67 |
| Configuration pour la fonction CTR | 68 |
| Configuration et démarrage de CTR | 68 |
| Définition de stratégies pour CTR | 71 |

| | |
|--|------------|
| Stratégies pour le TapePlex émetteur | 71 |
| Stratégies pour le TapePlex récepteur | 73 |
| Utilisation de la fonction CTR en l'absence de partitions LPAR sur le site distant | 74 |
| Utilisation de la fonction CTR comme solution de récupération après sinistre | 76 |
| Utilisation de la fonction CTR pour la continuité de l'activité | 76 |
| Utilisation de la fonction CTR pour la reprise de l'activité | 78 |
| Tests de récupération après sinistre à l'aide de la fonction CTR | 79 |
| Tests DR en l'absence de partitions LPAR sur le site DR | 80 |
| Gestion des VTV répliqués à l'aide de CTR (Cross-TapePlex Replication) | 81 |
| 5. Configuration d'une bibliothèque distante | 83 |
| Modification du fichier SCMDS du SMC | 83 |
| Mise à jour de la console CONFIG du VTCS pour définir une bibliothèque distante | 84 |
| Considérations relatives au pool de MVC | 85 |
| 6. Utilisation de configurations VTSS en cluster | 87 |
| Définition d'un VTSS en cluster | 87 |
| Conditions requises pour les VTSS en cluster | 88 |
| Fonctionnement des configurations VTSS en cluster | 90 |
| Fonctionnement du rapprochement VTSS | 92 |
| Clusters unidirectionnels et bidirectionnels | 93 |
| Clusters unidirectionnels | 93 |
| Fonctionnement des clusters VTSS unidirectionnels | 93 |
| Clusters bidirectionnels | 94 |
| Fonctionnement des clusters VTSS bidirectionnels | 95 |
| Clustering étendu | 96 |
| Réplication synchrone ou asynchrone | 97 |
| Implémentation de la réplication synchrone | 97 |
| Implémentation de la réplication asynchrone avec surveillance des travaux | 98 |
| Clustering avec connexions TCP/IP | 99 |
| Environnement TCP/IP | 99 |
| Configuration de VTCS pour CLINK TCP/IP | 100 |
| Instruction CONFIG CLINK | 100 |
| 7. Utilisation du logiciel Concurrent Disaster Recovery Test (CDRT) | 101 |
| Considérations relatives aux métadonnées | 102 |

| | |
|---|-----|
| Origine des données VTV du CDRT | 102 |
| Opérations exécutées sur les données à la fin du test | 103 |
| Gestion des ressources CDRT | 104 |
| Ressources de volume | 104 |
| Sous-pools provisoires | 104 |
| Ressources MVC | 105 |
| Ressources VTSS | 105 |
| Ressources VTSS non partagées | 106 |
| VTSS en cluster dans CDRT | 106 |
| Préparation d'un cluster VTSS en vue d'un test de récupération après sinistre | 106 |
| Gestion du contenu VTSS avant et après le test DR | 107 |
| Ressources VTSS partagées | 108 |
| Ressources ACS | 108 |
| Restrictions relatives aux ressources ACS pour le test de récupération après sinistre | 108 |
| Conditions requises pour l'hôte de production DR | 109 |
| Ressources VLE | 109 |
| Stratégies VTCS | 110 |
| Définition de MGMTCLAS/STORCLAS pour des VTSS non partagés | 110 |
| Définition de MGMTCLAS pour des VTSS partagés | 111 |
| Optimisation de l'accès aux ressources de test et de production | 111 |
| Exécution d'un test DR | 111 |
| Nettoyage après un test DR | 113 |
| Nettoyage après un test DR | 114 |
| Reprise des opérations normales | 114 |
| Scénarios opérationnels | 115 |
| Scénario 1 : sites de production et de test, ACS sur chaque site, VTSS de rechange sur le site de test | 115 |
| Exemple de JCL pour le scénario 1 | 116 |
| Scénario 2 : sites de production et de test, ACS sur chaque site, reprise VTSS sur le site de test | 117 |
| Exemple de JCL pour le scénario 2 | 118 |
| Scénario 3 : sites de production et de test, ACS sur chaque site, aucun VTSS | 119 |
| Exemple de JCL pour le scénario 3 | 120 |
| Scénario 4 : VTSS en cluster avec sites de production et de test DR | 120 |
| Exemple de JCL pour le scénario 4 | 121 |
| Scénario 5 : sites de production et de test, ACS et VLE sur chaque site | 122 |

| | |
|--|-----|
| Exemple de JCL pour le scénario 5 | 123 |
| Scénario 6 : sites de production et de test, VLE uniquement sur chaque site | 124 |
| Exemple de JCL pour le scénario 6 | 125 |
| Scénario 7 : VTSS en cluster (sans bandes) avec sites de production et de test | |
| DR | 126 |
| Exemple de JCL pour le scénario 7 | 126 |
| 8. Création de points de récupération système dans des environnements | |
| VSM | 129 |
| Exemples d'utilisation de points de reprise | 130 |
| Exemple 1 : copies MVC locales et copies MVC distantes | 130 |
| Exemple 2 : utilisation de CONFIG RECLAIM PROTECT | 133 |
| 9. Utilisation de VLE pour la récupération après sinistre | 135 |
| Mode de production normale | 135 |
| Exécution d'un test DR avec VLE | 136 |
| Nettoyage après un test DR avec VLE | 140 |
| Utilisation de VLE pour la continuité de l'activité | 142 |
| A. Exemples de VTSS en cluster | 143 |
| VTSS en cluster unidirectionnel | 143 |
| Configuration et gestion d'un système VTSS en cluster unidirectionnel | 145 |
| VTSS en cluster bidirectionnel | 148 |
| Configuration et gestion d'un système en cluster bidirectionnel | 151 |
| Clustering étendu | 155 |
| Configuration et gestion d'un système en cluster à 3 VTSS | 155 |
| Cluster VSM5 à VSM5 avec CLINK TCP/IP | 158 |
| Cluster VSM5 à VSM6 avec CLINK TCP/IP, VLE interconnectées | 160 |
| Cluster "sans bandes" VSM 6 à VSM 6 avec CLINK TCP/IP | 162 |
| Choix entre un cluster unidirectionnel ou bidirectionnel | 163 |
| Glossaire | 165 |
| Index | 181 |

Liste des tableaux

| | |
|--|-----|
| 6.1. Conditions requises pour les VTSS en cluster | 89 |
| 6.2. Conditions requises pour la réplication synchrone | 90 |
| 6.3. Conditions requises pour la réplication améliorée | 90 |
| A.1. Valeurs IPIF CLINK pour VSMPR1 | 159 |
| A.2. Valeurs IPIF CLINK pour VSMPR2 | 159 |
| A.3. Valeurs IPIF CLINK pour VSMPR1 | 161 |
| A.4. Valeurs IPIF RTD pour VSMPR1 | 161 |

Liste des exemples

| | |
|---|----|
| 4.1. Console CONFIG pour TapePlex A | 69 |
| 4.2. Console CONFIG pour TapePlex B | 70 |

Préface

StorageTek Enterprise Library Software (ELS) d'Oracle est une solution composée des logiciels de base suivants :

- StorageTek Storage Management Component (SMC)
- StorageTek Host Software Component (HSC)StorageTek Virtual Tape Control Software (VTCS)
- StorageTek Concurrent Disaster Recovery Test (CDRT)

Les logiciels suivants sont également fournis dans le package ELS :

- StorageTek Library Content Manager (LCM, anciennement ExLM) LCM inclut une version améliorée du produit anciennement connu sous le nom d'Offsite Vault Feature.
- StorageTek Client System Component pour les environnements MVS (MVS/CSC)
- StorageTek LibraryStation

Public

Le présent guide s'adresse au personnel ou aux clients StorageTek chargés de la planification et de l'implémentation des solutions de récupération après sinistre (DR) et de gestion de données hors site.

Prérequis

Pour effectuer les tâches décrites dans ce guide, vous devez être familiarisé avec les notions suivantes :

- Système d'exploitation Z/OS
- JES2 ou JES3
- Enterprise Library Software (ELS)

A propos de ce manuel

Le *Guide de récupération après sinistre et de gestion des données hors site d'ELS* contient les éléments suivants :

- [Chapitre 1, Introduction à la récupération après sinistre](#) , qui est une introduction aux solutions de récupération après sinistre et une présentation de ces dernières.
- Pour la gestion des données hors site, voir :

- [Chapitre 2, Exportations et importations physiques](#) , qui traite de l'utilisation des fonctions EXPORT et IMPORT. La fonction EXPORT crée des MVC portables sur le site source, que vous éjectez ensuite de l'ACS source, transportez physiquement vers un autre site, avant d'utiliser la fonction IMPORT pour les importer sur le site cible.
- [Chapitre 3, Utilisation de la fonction d'archivage sécurisé externe d'ELS](#) , qui décrit l'utilisation de cette fonction de récupération après sinistre (DR), qui utilise à son tour les fonctions EXPORT et IMPORT ainsi que des utilitaires et des commandes supplémentaires pour automatiser le processus d'archivage sécurisé des MVC hors site.
- [Chapitre 5, Configuration d'une bibliothèque distante](#) , qui présente un exemple de configuration avec une bibliothèque distante et des informations permettant de le configurer.
- [Chapitre 6, Utilisation de configurations VTSS en cluster](#) , qui vous fournit toutes les informations indispensables sur l'implémentation de VTSS en cluster, notamment le clustering étendu (3 VTSS ou plus par cluster), qui constitue une nouveauté dans ELS 7.0 et les versions suivantes. Un VTSS en cluster n'est pas à proprement parler une solution de DR. Toutefois, grâce à ses fonctions distribuées/de redondance/sauvegarde de VTSS en cluster, il se prête bien aux solutions de récupération après sinistre. Je configure par exemple des classes de gestion et de stockage afin de réaliser deux copies MVC, l'une sur un site local et l'autre sur un site distant. Pour plus d'informations, voir aussi [Annexe A, Exemples de VTSS en cluster](#).
- Pour des solutions de DR spécifiques, voir :
 - [Chapitre 4, Utilisation de la réplication entre TapePlex dans une solution de récupération après sinistre](#) ,, qui décrit l'utilisation de la fonction CTR (Cross-Tapeplex Replication, réplication entre tapeplex) pour configurer une solution complète de récupération après sinistre pour les données de bande.
 - [Chapitre 7, Utilisation du logiciel Concurrent Disaster Recovery Test \(CDRT\)](#) ,, qui vous indique comment utiliser cette fonction pour valider les solutions de DR de votre site. Notez que pour ELS 7.2, CDRT prend en charge Virtual Library Extension (VLE) 1.0 et versions suivantes.
 - [Chapitre 8, Création de points de récupération système dans des environnements VSM](#) ,, qui vous indique comment implémenter des points de récupération après sinistre dans un environnement VSM.
 - [Chapitre 9, Utilisation de VLE pour la récupération après sinistre](#) ,, qui décrit l'utilisation de VLE (Virtual Library Extension) en tant que solution de récupération après sinistre proposant une méthode sans interruption et simplifiée de tests DR.

Tous ces chapitres abordent les méthodes de déplacement de données entre deux emplacements et de gestion des données. Ces processus peuvent être utilisés conjointement avec d'autres processus de reprise/continuité de l'activité/récupération après sinistre. La fonction ELS Vault (archivage sécurisé d'ELS) vous permet de gérer des volumes de DR et de conservation à long terme (LTR) hors site, ainsi que des volumes d'armoire manuelle ("plancher").

Vous disposez donc de toute une gamme de solutions de récupération après sinistre et de gestion des données hors site. Comment sélectionner la meilleure option pour votre site parmi ces offres ?

Choix d'une solution adaptée à mes besoins

Les solutions décrites dans le *Guide de récupération après sinistre et de gestion des données hors site d'ELS* varient en termes de coûts et d'avantages associés, où la règle générale veut que les solutions les plus chères garantissent un délai de récupération plus rapide ou des cycles de gestion de données plus courts. Par exemple, l'exportation/importation de MVC (éventuellement à l'aide de la fonction ELS External Vault (archivage sécurisé externe d'ELS)) est relativement peu chère, mais la récupération prend du temps uniquement en raison du déplacement des MVC d'un site à l'autre.

Toutefois, la fonction CTR est relativement chère. Vous avez besoin de deux sites, ainsi que de plusieurs VTSS, RTD et MVC sur chacun d'eux, etc. CTR utilise le clustering VTSS pour copier des données sur des liens de cluster FICON ou ESCON directement à partir du VTSS d'un TapePlex vers le VTSS d'un autre TapePlex. Le déplacement de données (et de métadonnées) entre les sites étant électronique, la sauvegarde et la récupération peuvent s'effectuer plus rapidement qu'avec l'archivage sécurisé externe.

Accessibilité de la documentation

Pour plus d'informations sur l'engagement d'Oracle pour l'accessibilité à la documentation, visitez le site Web Oracle Accessibility Program, à l'adresse <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Accès aux services de support Oracle

Les clients Oracle qui ont souscrit un contrat de support ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> ou le site <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> si vous êtes malentendant.

Nouveautés

Cette révision inclut les mises à jour suivantes :

- Ajout d'une remarque sur l'exportation et l'importation de MVC par classe de gestion.
Voir " [Exportation et importation par classe de gestion](#)".
- Ajout d'une remarque sur l'exportation et l'importation de MVC par classe de stockage.
Voir " [Exportation et importation par classe de stockage](#)".
- Ajout d'une procédure relative à l'importation/exportation pour les VLE.
Voir " [Exportation/importation dans une VLE](#)".
- Ajout d'une remarque à la description des VTSS présentant la relation de la réplication synchrone ou améliorée avec le gestionnaire MIH (Missing Interrupt Handler).
Voir " [Définition d'un VTSS en cluster](#) "
- Ajout du tableau Conditions requises pour la réplication améliorée.
Voir " [Conditions requises pour les VTSS en cluster](#)".

Chapitre 1. Introduction à la récupération après sinistre

Les pratiques recommandées en matière de récupération après sinistre d'ELS consistent généralement à concevoir et implémenter des systèmes matériels et logiciels résistants aux pannes qui peuvent perdurer après un sinistre ("continuité de l'activité") et reprendre un fonctionnement normal ("reprise de l'activité"), avec une intervention minimale de l'utilisateur et, idéalement, sans perte de données. La mise en place d'environnements résistants aux pannes pour répondre aux objectifs de reprise après sinistre d'ELS et les contraintes budgétaires du monde réel nécessitant du temps et de l'argent, l'entreprise doit être totalement impliquée dans ce projet.

La récupération après sinistre est généralement prévue pour répondre à un ou plusieurs des types de sinistres suivants :

- dommages importants ou étendus subis par les installations informatiques, qui ont été provoqués par une catastrophe naturelle (tremblement de terre, tempête, inondation, etc.) ou d'autres causes (incendie, vandalisme, vol, etc.) ;
- perte importante de services essentiels pour l'installation informatique (coupure de courant, refroidissement ou accès réseau, par exemple) ;
- perte de personnel clé.

Le processus de planification de récupération après sinistre (DR) commence par identifier et déterminer les types de sinistres auxquels une entreprise doit faire face, ainsi que les opérations de reprise qu'elle doit mettre en place. Le processus de planification identifie les conditions de haut niveau requises en matière de continuité et de reprise de l'activité, notamment le degré nécessaire de tolérance aux pannes. Le produit de planification de la reprise après sinistre est une architecture de récupération et de reprise qui permet à des applications, des données et des systèmes tolérants aux pannes de prendre en charge ces conditions requises selon les contraintes en vigueur. La récupération après sinistre doit généralement prendre en compte les contraintes suivantes : objectif de délai de récupération (RTO), objectif du point de récupération (RPO) et budget disponible. L'architecture DR associée aux contraintes de l'activité génère des procédures de récupération après sinistre qui intègrent tous les éléments du système dans un véritable schéma "de bout en bout", afin de garantir des résultats prévisibles pour le processus global de récupération après sinistre.

Les systèmes tolérants aux pannes font généralement preuve de robustesse et de résilience grâce à la *redondance*. Souvent très coûteux, un système totalement redondant ne dispose pas d'un point de panne unique au sein de son architecture et peut continuer et reprendre

son fonctionnement après le pire des sinistres dans la limite de ses moyens. Les systèmes de contrôle de vol et de navette spatiale constituent d'excellents exemples de systèmes totalement redondants. Les applications informatiques moins critiques utilisent généralement des systèmes moins robustes avec une redondance inférieure. Ces systèmes sont moins chers à mettre en place et impliquent nécessairement une interruption de service après des sinistres, au cours de laquelle l'entreprise s'efforce de rétablir ses applications, données et systèmes récupérables.

Finalement, la nature d'une entreprise, les exigences de ses clients et le budget disponible pour la récupération après sinistre constituent des facteurs clés à prendre en compte lors de la formulation des conditions requises pour la récupération après sinistre. Une solution de récupération après sinistre complète peut coûter cher...mais il est essentiel qu'elle repose sur une architecture. L'argent, le matériel et les logiciels ne suffisent pas pour venir à bout d'un éventuel sinistre, tout en espérant que votre activité pourra reprendre et perdurer. Cependant, même si vous planifiez et mettez en place une architecture de façon intelligente, vous risquez de devoir faire face à des coupures plus longues et/ou une détérioration du service tant que l'ensemble des services ne sera pas rétabli, mais vous pourrez toujours disposer d'une solution de reprise après sinistre limitée fiable.

Toutefois, vous devez savoir que probablement *aucune* planification ne peut anticiper **tous** les scénarios de récupération après sinistre possibles et y répondre. Par exemple, ce qui semble être de prime abord un problème banal sur un système peut dégénérer au fil du temps et affecter d'autres systèmes de diverses manières, pour finir en catastrophe pour laquelle il n'existe aucun scénario de récupération. De même, la capacité d'une entreprise à honorer ses contrats de service peut se trouver affectée si des hypothèses clés ne sont pas vérifiées. Par exemple, si des composants ou des services essentiels ne sont pas disponibles, ou si le fournisseur de la solution de récupération après sinistre ne peut pas livrer le service aussi efficacement qu'il le prétendait. Toutefois, si un sinistre se révèle **plus grave** que le pire scénario que vous aviez prévu, la récupération risque de ne pas être possible.

Définition de l'objectif de délai de récupération (RTO, Recovery Time Objective)

Le RTO est un objectif de niveau de service du temps mis pour atteindre une fonction opérationnelle donnée après un sinistre. Par exemple, les exigences de l'entreprise peuvent mettre en place le RTO stipulant que tous les systèmes de production fonctionneront à 80 % des capacités antérieures au sinistre dans les 30 minutes suivant une coupure non planifiée qui durerait plus d'une heure (en cas d'absence de fonction de récupération après sinistre). Le temps de traitement du RPO, la disponibilité de personnel informatique qualifié et la complexité des processus informatiques manuels requis après un sinistre constituent des exemples de contraintes pouvant influencer sur la détermination du RTO. Le RTO ne s'applique pas aux systèmes entièrement tolérants aux pannes, car ces derniers effectuent une reprise implicite pendant et après un sinistre, sans interruption de service.

Les planificateurs de DR peuvent déterminer des RTO différents pour tout ou partie des exigences de continuité de l'activité (BC) définies. Les divers types d'activités peuvent

nécessiter des RTO différents, par exemple pour les systèmes en ligne et les fenêtres de traitement par lot. En outre, des RTO différents peuvent s'appliquer aux phases d'un plan de DR, dans lequel un RTO est défini pour chaque phase. Enfin, une application récupérable peut disposer de RTO différents pour chacun de ses niveaux de service.

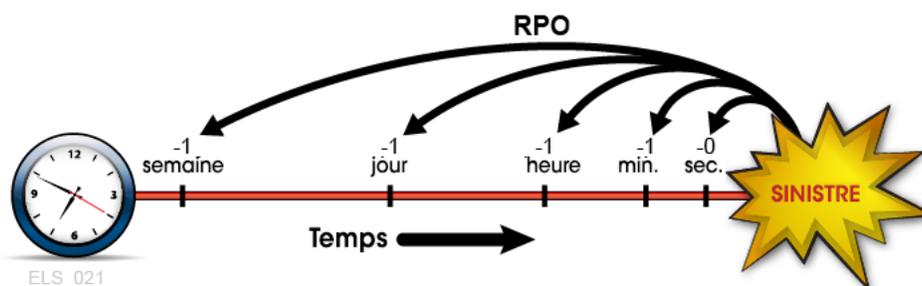
Les exigences en matière de disponibilité des données de continuité de l'activité sont extrêmement importantes pour la planification d'un RTO. Lorsque les données qui doivent être saisies dans le processus de DR ne se trouvent pas sur le site de récupération après sinistre, le temps mis pour extraire les données sur site entraînera un retard du RTO. Par exemple, la durée d'extraction des données situées dans des archivages de stockage hors site sera longue. La récupération peut s'effectuer rapidement si les données d'entrée à jour sont dupliquées sur le site de récupération avant le début des opérations de récupération après sinistre.

Définition de l'objectif du point de récupération (RPO)

Le RPO est un objectif de continuité de l'activité qui détermine l'état de l'activité, ou niveau de l'activité, obtenu après le rétablissement des systèmes récupérables par le processus de récupération après sinistre. Conceptuellement, un RPO est un état cible de synchronisation ou de "restauration" connu antérieur au sinistre. Le RPO est donc le point de récupération après sinistre à partir duquel le traitement des applications récupérables interrompues peut reprendre. Toutes les transactions qui ont lieu dans l'intervalle situé entre le RPO et le moment du sinistre sont irrécupérables. Le RPO ne s'applique pas aux systèmes entièrement tolérants aux pannes, car le sinistre n'affecte pas la continuité de l'activité de ces derniers.

La [Figure 1.1, « Objectifs de point de récupération »](#) illustre les concepts de RPO en suggérant divers points de récupération qui doivent être pris en compte par les planificateurs de DR. La planification doit s'assurer que le RPO souhaité est réalisable en fonction du RTO choisi et inversement. En général, les plans de récupération après sinistre qui exigent des RPO plus proches du moment du sinistre nécessitent davantage de tolérance aux pannes et sont plus chers à implémenter que les RPO plus éloignés. Comme pour les RTO, les planificateurs de DR peuvent définir différents RPO en fonction des exigences de continuité de l'activité, des phases du plan de récupération après sinistre ou des niveaux de service de l'application.

Figure 1.1. Objectifs de point de récupération



Plus généralement, la planification de RPO doit identifier tous les éléments associés qui doivent être présents pour rétablir chaque système récupérable, notamment les données, les métadonnées, les plates-formes, les installations et le personnel. La planification doit garantir que ces éléments sont disponibles au niveau de l'activité voulu pour la récupération. Les exigences en matière de mise à jour des données de continuité de l'activité sont particulièrement importantes pour la planification du RPO. Par exemple, si les exigences de continuité de l'activité exigent un RPO d'une heure, les données ou métadonnées qui alimentent le processus de récupération doivent être au niveau du RPO, sous peine de ne pouvoir l'atteindre. Les processus de récupération après sinistre de l'organisation spécifient les procédures permettant d'atteindre tous les RPO définis au sein des RTO établis.

Les métadonnées système requises pour la récupération de RPO incluent des structures de catalogue de système d'exploitation et des informations système de gestion des bandes. Ces éléments doivent être mis à jour lors du processus de récupération après sinistre pour activer tous les RPO choisis. Par exemple, pour assurer la cohérence entre les diverses entrées de métadonnées dans le processus de DR, les jeux de données existants qui seront recréés sur le RPO ne doivent pas être catalogués, les jeux de données mis à jour entre le RPO et le moment du sinistre doivent être restaurés à la version antérieure ou simultanée au RPO et les modifications du catalogue relatives aux bandes doivent être synchronisées avec le système de gestion des bandes.

Traitement des pannes temporaires

La récupération après sinistre fournit une solution pour les pannes très longues qui risqueraient de rendre un site de production inutilisable pendant une longue période. Alors que le reste de cette introduction concerne les pratiques relatives à la récupération après sinistre, il est également important de mettre en place des procédures permettant de réduire les pannes relativement brèves qui peuvent affecter négativement la production si elles ne sont pas résolues. Prenons l'exemple d'une interruption de service au cours de laquelle certaines installations matérielles ou réseau sont indisponibles pendant une heure ou deux, mais durant laquelle la production peut continuer en "mode dégradé" avec quelques rapides ajustements temporaires. Une procédure applicable à une panne temporaire doit indiquer comment isoler le problème, les modifications à apporter, les personnes à avertir et comment rétablir l'environnement d'exploitation normal après la restauration du service.

Concept clé : récupération du point de synchronisation

Le redémarrage des applications de production sur des RPO définis est une activité clé qui est effectuée lors d'une réelle récupération après sinistre et des tests DR. Les environnements DR les plus résistants veilleront à s'assurer que chaque application récupérable, qu'elle provienne d'un fournisseur extérieur ou soit développée en interne, applique une exigence essentielle en matière de récupération après sinistre, à savoir que l'application soit conçue pour redémarrer à partir d'un point planifié, appelé point de synchronisation, pour atténuer les effets d'une interruption non programmée lors de son exécution. Lorsqu'une application interrompue est redémarrée à un point de synchronisation, ses résultats sont identiques à ceux d'une application non interrompue.

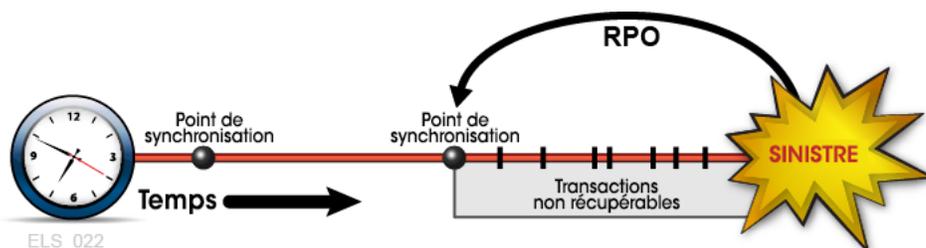
La procédure de redémarrage pour une application récupérable dépend de la nature de l'application et de ses entrées. La procédure de redémarrage d'une application pour une réelle récupération après sinistre ou des tests DR est assez souvent identique à celle utilisée pour le redémarrage de l'application en cas d'échec lors d'une exécution de production normale. Lorsque cela est possible, la réutilisation des procédures de redémarrage de la production pour la réelle récupération après sinistre ou les tests DR simplifie la création et la maintenance des procédures DR et exploite ces procédures éprouvées. Dans le cas le plus simple, une application récupérable est une étape de travail unique avec un seul point de synchronisation, qui constitue le début du programme appelé par cette étape. Dans ce cas, la procédure de récupération peut simplement consister à soumettre à nouveau le travail interrompu. Une procédure de redémarrage légèrement plus complexe peut impliquer le non-catalogage de l'ensemble des jeux de données de sortie générés par l'application lors de sa dernière exécution, puis le redémarrage de l'application.

Les procédures de redémarrage pour les applications ayant le choix entre plusieurs points de synchronisation internes risquent de ne pas être aussi simples. Les applications qui utilisent des techniques de point de reprise/redémarrage pour implémenter ces points de synchronisation enregistrent périodiquement leur progression et peuvent par exemple utiliser les informations de point de reprise enregistrées pour redémarrer au dernier point de synchronisation interne enregistré avant une interruption. Les procédures de redémarrage se conformeront aux exigences de chaque point de synchronisation. Lorsque des points de reprise sont utilisés, les jeux de données qui leur sont associés ne doivent pas avoir expiré, être retirés du catalogue ou supprimés alors que le point de reprise reste valide pour la récupération d'application. Une façon simple de définir un point de synchronisation pour une étape de travail qui modifie ses jeux de données d'entrée existants consiste à réaliser une copie de sauvegarde de chaque jeu de données modifiable avant d'exécuter l'étape. Ces jeux de données d'entrée modifiables peuvent être facilement identifiés en recherchant l'attribut JCL DISP=MOD dans des instructions DD ou dans des demandes d'allocation dynamique. En cas d'échec ou d'interruption d'une étape de travail, effacez simplement les jeux de données d'entrée modifiés, restaurez-les à partir des copies de sauvegarde et redémarrez l'étape à partir des copies restaurées. Ces copies de sauvegarde sont également utiles pour redémarrer une étape de travail défectueuse ou interrompue qui a expiré, a été retirée du catalogue ou dont les originaux ont été supprimés.

Mise en relation d'un RPO avec la récupération du point de synchronisation

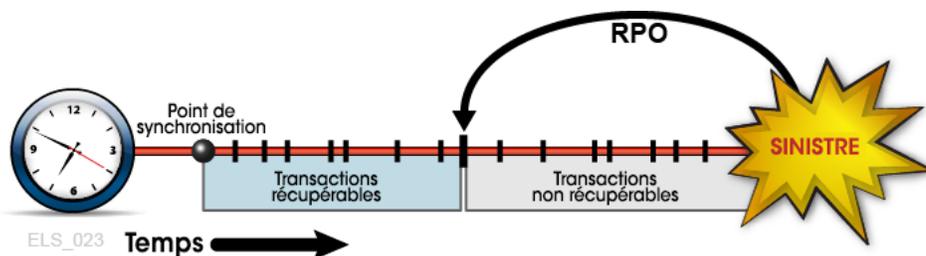
Lorsque le RPO s'aligne sur un point de synchronisation, l'exécution de la procédure de redémarrage d'une application qui a été développée pour ce point de synchronisation fera redémarrer l'application au début si aucune interruption n'a eu lieu ([Figure 1.2, « RPO au point de synchronisation »](#)). Toutes les transactions traitées entre ce RPO et le sinistre sont censées être irrécupérables.

Figure 1.2. RPO au point de synchronisation



Les exigences de continuité de l'activité peuvent parfois justifier de placer le RPO entre des points de synchronisation. Dans ces cas, la récupération entre points de synchronisation se fonde sur des données complémentaires qui décrivent les modifications critiques apportées à l'état de l'application ou les événements qui se sont produits après la définition du point de synchronisation le plus récent. Examinons par exemple le RPO situé une minute avant le sinistre. Supposons qu'une application récupérable soit conçue pour utiliser des points de reprise afin d'enregistrer sa progression, mais que le temps système mis pour définir ces points de reprise à des intervalles d'une minute soit intolérable. Une solution consiste à définir des points de reprise moins fréquents et à consigner toute les transactions validées entre ces points de reprise. Ce journal des transactions constitue ensuite des données d'entrée complémentaires, qui sont utilisées par le processus de récupération du point de reprise pour redémarrer à partir d'un RPO situé au delà du point de synchronisation le plus récent. Dans cet exemple, la procédure de redémarrage de l'application accède aux données du point de reprise le plus récent et utilise le journal des transactions complémentaire pour rétablir toutes les transactions validées qui ont été traitées après le point de reprise et avant le RPO (Figure 1.3, « RPO entre des points de synchronisation »). Ainsi, la récupération du point de reprise peut obtenir un RPO cible en utilisant les données d'entrée de plusieurs sources. Toutes les transactions traitées entre le RPO et le sinistre sont censées être irrécupérables.

Figure 1.3. RPO entre des points de synchronisation



Planification de la haute disponibilité des données (D-HA)

Les données constituent souvent l'une des ressources les plus précieuses que détient une entreprise. La plupart des sociétés veillent particulièrement à protéger leurs données essentielles contre une éventuelle perte et réalisent pour ce faire des investissements supplémentaires. De même, elles tiennent à s'assurer que les données sont utilisées à bon

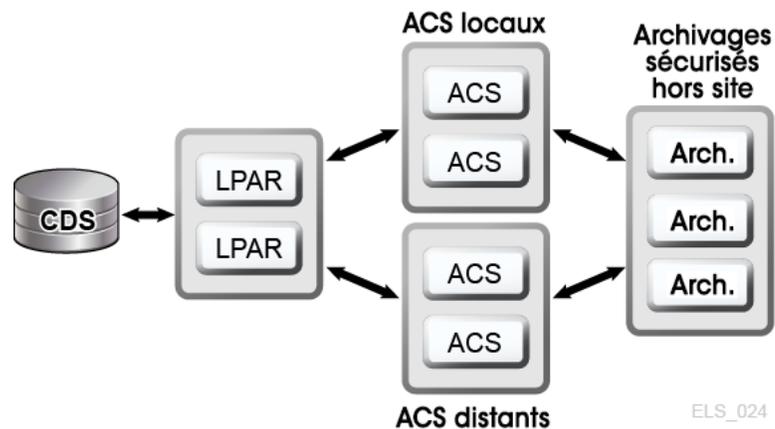
escent et au moment opportun. Les conséquences risquent d'être désastreuses pour une entreprise qui ne peut pas se prémunir contre une perte de données critiques. La méthode la plus répandue pour se protéger contre la perte de données consiste peut-être à stocker des copies des données critiques sur différents sous-systèmes ou médias de stockage et à conserver certaines de ces copies dans divers emplacements physiques. Les copies conservées sur des médias de stockage amovibles, notamment des bandes à cartouche magnétique, des CD-ROM et des DVD, sont généralement archivées dans des emplacements de stockage hors site. De même, les copies supplémentaires sont généralement stockées sur site dans des installations informatiques où des applications peuvent traiter ces données. La création et le stockage de copies de données critiques augmentent la redondance des données et améliorent leur tolérance aux pannes. Pour les médias amovibles, en particulier pour les bandes à cartouche magnétique, l'augmentation de la redondance des données seule ne suffit généralement pas à garantir la haute disponibilité des données pour les applications qui les utiliseront. Par exemple, le système VSM d'Oracle pour les bandes virtuelles mainframe stocke les données sur des volumes de bande physiques appelés MVC. VSM peut réaliser automatiquement des copies MVC pour améliorer la redondance des données et réduire les risques liés à un incident sur le média ou au mauvais positionnement d'une cartouche de bande. Un système VSM de production utilise de nombreux composants matériels spécialisés pour extraire les données stockées sur une cartouche MVC, notamment un lecteur tampon VTSS, une bibliothèque automatisée et des lecteurs de bande connectés à une bibliothèque, appelés RTD, qui sont également reliés au lecteur tampon VTSS. Les applications hôtes dépendent de tous ces composants VSM qui interagissent pour extraire les données des MVC. Bien que la plupart des gens ne considèrent pas la panne d'un seul composant comme un sinistre comparable à la perte de la totalité d'un centre de données dans un tremblement de terre, il deviendra quasiment impossible d'extraire des données MVC si un seul composant VSM critique tombe en panne sans sauvegarde, quel que soit le nombre de copies MVC redondantes existantes. Par conséquent, alors que la création de copies MVC constitue l'une des pratiques recommandées éprouvées pour réduire la vulnérabilité et les risques, cette opération ne suffit pas toujours à garantir la haute disponibilité des données (D-HA) en cas de pannes. Les exigences en matière de D-HA sont fondamentales pour la continuité de l'activité lors de la planification de la récupération après sinistre. La D-HA est généralement obtenue en augmentant les redondances pour éliminer les points de panne uniques qui empêcheraient les applications d'accéder aux données en cas de pannes du système de stockage. Un système VSM incluant des composants redondants améliore par exemple sa tolérance aux pannes. L'installation de plusieurs périphériques VTSS, HandBots SL8500 redondants et RTD vise à éliminer les points de panne VSM uniques sur le chemin de données entre l'application et les données critiques stockées sur une MVC. L'architecture VSM est conçue pour prendre en charge l'ajout de composants redondants afin d'augmenter la tolérance aux pannes et de promouvoir la D-HA.

Bande physique hautement disponible

Les solutions d'automatisation de bande mainframe d'Oracle activent la D-HA pour les applications de bande physique en stockant des copies de données redondantes dans différents ACS au sein d'un tapeplex, à savoir dans un complexe de bandes mappé par un seul CDS. Par

exemple, les applications exécutées sur une installation informatique avec un seul tapeplex peuvent facilement stocker des copies de jeux de données de bandes en double dans un ou plusieurs ACS au sein de ce tapeplex. Cette technique améliore la D-HA en ajoutant des médias redondants, des transports de bandes et des bandothèques automatisées. Dans un scénario simple, une application stocke des copies redondantes d'un jeu de données critique sur deux bandes de cartouche différentes dans une seule bibliothèque SL8500 avec Redundant Electronics, deux HandBots sur chaque rail et au moins deux transports de bandes connectés à la bibliothèque sur chaque rail, qui sont compatibles avec le média de jeu de données. Pour supprimer la bibliothèque SL8500 comme possible point de panne unique, une deuxième bibliothèque SL8500 est ajoutée à l'ACS pour stocker davantage de copies redondantes du jeu de données critique. Pour éliminer l'installation informatique elle-même comme point de panne unique, les copies redondantes du jeu de données peuvent être archivées hors site ou créées sur un ACS distant par transport de bande, via une extension de canal (Figure 1.4, « Configuration de bandes physiques FD-HA »).

Figure 1.4. Configuration de bandes physiques FD-HA



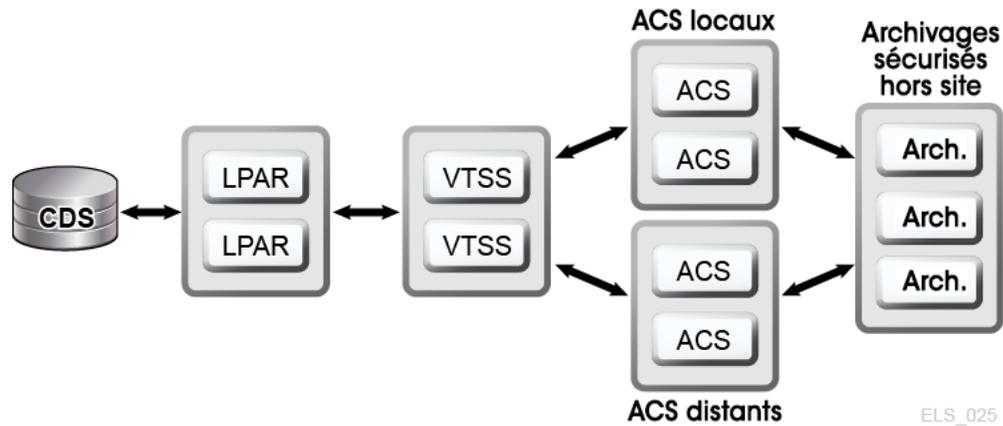
Vous pouvez également réaliser plusieurs copies de bandes physiques dans divers emplacements physiques lorsque chacun d'eux dispose de son propre CDS indépendant, à savoir lorsque le matériel présent dans chaque emplacement représente un tapeplex distinct. Grâce à l'utilisation de la fonctionnalité client/serveur de SMC et à la définition de stratégies qui dirigent les copies du jeu de données vers un tapeplex distant, les travaux peuvent créer des copies de bandes dans un ACS figurant sur un autre tapeplex sans apporter de modifications au JCL.

Bande virtuelle hautement disponible

VSM fournit des technologies de clustering et de multiplexage MVC pour activer la D-HA pour les bandes virtuelles mainframe. Le multiplexage VSM implique la création de plusieurs copies MVC (par exemple, duplex, quadruplex) dans un ou plusieurs ACS afin d'optimiser la redondance (Figure 1.5, « Configuration du multiplexage VSM de la D-HA »). Les ACS qui reçoivent des copies multiplexées peuvent être des bibliothèques locales ou des ACS distants par transport de bande, via une extension de canal. Les stratégies de migration VSM

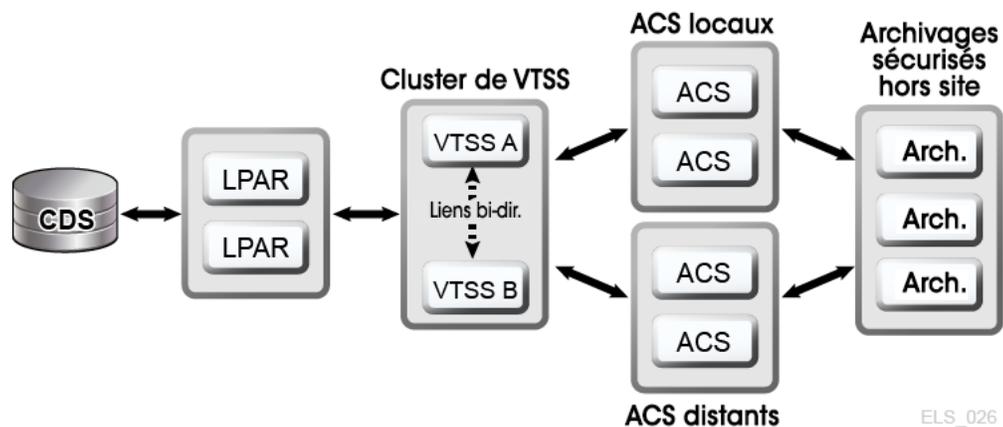
contrôlent le mouvement des VTV situés sur le tampon VTSS vers des MVC locales ou distantes, qui peuvent être archivées hors site.

Figure 1.5. Configuration du multiplexage VSM de la D-HA



Un cluster VSM contient plusieurs périphériques VTSS (nœuds) qui sont mis en réseau pour permettre l'échange de données via un lien de communications (CLINK.) Les CLINK sont des canaux unidirectionnels ou bidirectionnels. La configuration de cluster VSM la plus élémentaire se compose de deux nœuds VTSS reliés à un CLINK unidirectionnel dans le même tapeplex, mais des CLINK bidirectionnels sont fréquemment déployés (Figure 1.6, « Configuration de cluster VSM de la D-HA »). Chaque nœud de cluster peut se trouver sur un site différent. Les stratégies de stockage unidirectionnel VSM contrôlent la réplification automatique des volumes de bande virtuelle (VTV) entre VTSS A et VTSS B via un CLINK unidirectionnel. Les stratégies de stockage bidirectionnel et les CLINK bidirectionnels permettent au VTSS A d'effectuer une réplification sur le VTSS B et inversement.

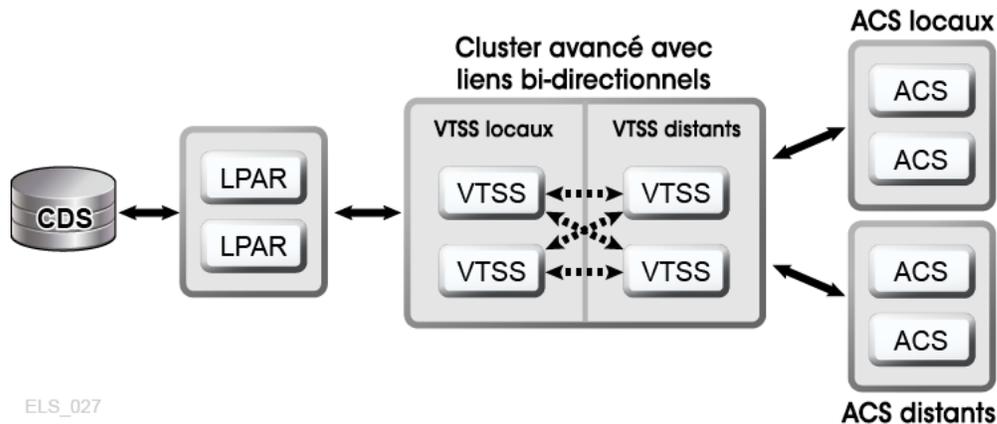
Figure 1.6. Configuration de cluster VSM de la D-HA



Le clustering étendu VSM permet une connectivité de plusieurs à plusieurs entre trois périphériques VTSS au minimum au sein d'un tapeplex, afin d'optimiser la disponibilité des

données (Figure 1.7, « Configuration de cluster étendue de la D-HA (archivages hors site non indiqués) »). L'installation de périphériques de cluster VTSS sur plusieurs sites au sein d'un tapeplex comme dans l'illustration permet d'augmenter la redondance en éliminant chaque site en tant que point de panne unique.

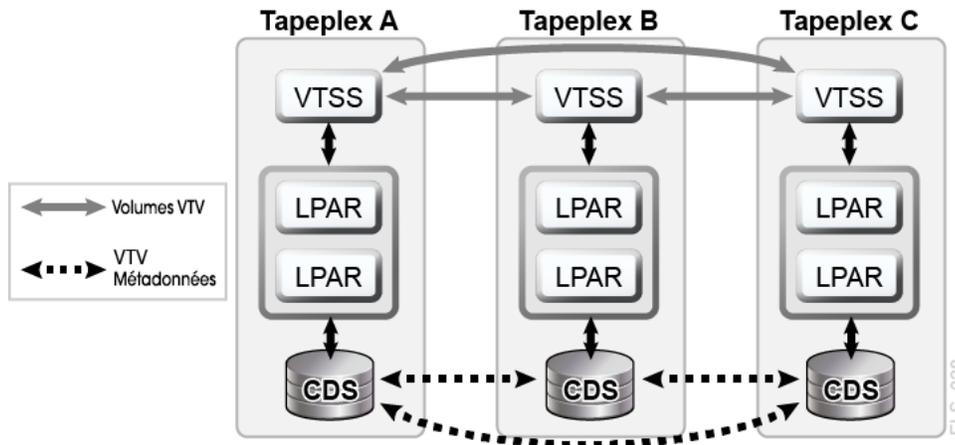
Figure 1.7. Configuration de cluster étendue de la D-HA (archivages hors site non indiqués)



Le produit LCM d'Oracle rationalise les processus d'archivage sécurisé hors site pour les volumes MVC en gérant le processus de recyclage entre les archivages et les bibliothèques de production. La fonction d'archivage sécurisé de LCM planifie le renvoi des volumes MVC archivés lorsque la quantité de données ayant expiré dépasse un seuil défini.

Un cluster CTR (Cross-Tapeplex Replication) VSM permet à des périphériques de cluster VTSS de figurer dans différents tapeplex et de répliquer des VTV entre un tapeplex et un ou plusieurs autres tapeplex, en activant des modèles de répllication de cluster de plusieurs à plusieurs via des CLINK unidirectionnels ou bidirectionnels (Figure 1.8, « Configuration de la répllication entre Tapeplex (CTR) VSM de la D-HA »). L'envoi et la réception de tapeplex peut avoir lieu sur des sites différents. Les VTV répliqués sont entrés dans le CDS du tapeplex récepteur sous forme de volumes en lecture seule. Cela garantit ainsi une excellente protection des données contre leur modification par les applications exécutées dans le tapeplex récepteur. Le CDS du tapeplex récepteur indique également que les copies VTV répliquées via CTR appartiennent au tapeplex émetteur. Ainsi, pour renforcer la protection, CTR s'assure qu'un tapeplex ne peut pas modifier un VTV dont il n'est pas propriétaire.

Figure 1.8. Configuration de la réplication entre Tapeplex (CTR) VSM de la D-HA



D-HA et récupération du point de synchronisation

La création de plusieurs copies de volumes physiques (MVC ou non) améliore la redondance de données, mais ces copies posent des conditions particulières pour la récupération du point de synchronisation. L'aspect le plus important de la récupération du point de synchronisation consiste à s'assurer que les données créées au point de synchronisation restent à l'état de lecture seule tout en restant valides à des fins de récupération après sinistre. Cela signifie que les copies de volume de bande physique qui peuvent être utilisées pour la récupération après sinistre doivent rester à l'état de lecture seule. Une méthode pour effectuer cette opération consiste à envoyer ces copies vers un emplacement d'archivage hors site sur lequel il n'existe aucune fonction de traitement de bande. Sachez que les copies non protégées qui subissent une modification deviennent inutilisables pour la récupération du point de synchronisation, car le contenu mis à jour ne reflète plus le point de synchronisation associé. Les environnements de bande virtuelle ajoutent une dimension supplémentaire à la gestion de plusieurs de copies de volume pour la récupération du point de synchronisation. Des copies VTV peuvent cohabiter simultanément dans plusieurs tampons VSM et sur plusieurs MVC. Même lorsque toutes les MVC d'un VTV donné sont archivées hors site, il est possible de modifier les copies VTV qui demeurent sur site dans les tampons VSM. Une copie VTV mise à jour dans un tampon ne doit pas être utilisée pour la récupération du point de synchronisation, sauf si cet VTV appartient à un nouveau point de synchronisation qui invalide les copies hors site archivées à des fins de récupération après sinistre.

Exécution d'une réelle récupération après sinistre

Le succès d'une opération de réelle récupération après sinistre repose sur le fait de disposer d'un site DR adéquat, de personnel formé, d'une procédure DR qui a fait ses preuves, d'une charge globale de production récupérable avec des points de synchronisation pour répondre aux RPO définis, ainsi que de l'ensemble des données d'entrée et des métadonnées système nécessaires à la réalisation de ces objectifs. Les données d'entrée et les métadonnées système doivent être accessibles sur le site DR si nécessaire et disponibles aux niveaux requis de

mise à jour. Avec une planification minutieuse, une préparation approfondie et une exécution bien rôdée, les opérations de réelle récupération après sinistre peuvent se dérouler sans problème selon le plan établi pour atteindre les RPO et les RTO définis. Les données de production générées sur le site DR doivent être protégées de façon adéquate alors que le site DR fonctionne comme un site de production. Supposons par exemple que l'architecture D-HA nécessite que la charge globale de production réplique les copies de données redondantes sur trois sites distants et que le site DR soit l'un de ces sites de réplication distants avant le sinistre. Lorsque le site de production subit un sinistre et que sa charge globale est transférée vers le site DR, ce dernier ne peut plus servir de site de réplication distant pour la charge globale de production actuellement exécutée en local sur le site. Pour répondre à l'exigence de D-HA des trois sites de réplication distants, un troisième nouveau site de réplication distant doit être mis en ligne tant que la production reste sur le site DR. Cet exemple illustre la façon dont une analyse minutieuse des exigences de D-HA permet aux planificateurs de DR de répondre à toutes les exigences de D-HA critiques qui doivent être satisfaites lorsque la production est transférée vers un site DR. Un plan de récupération après sinistre complet n'englobe pas seulement les activités permettant de rétablir la production sur le site DR, mais également le processus de suppression du site DR lorsque le site de production est réparé et prêt à l'emploi, en supposant que le site DR ne soit qu'un substitut temporaire au site de production. Par exemple, lorsque le site de production est prêt à reprendre son fonctionnement, les données de production doivent être rétablies sur ce site. Les méthodes proposées incluent le clustering bidirectionnel entre le site DR et le site de production, accordant ainsi suffisamment de temps au travail de production exécuté sur le site DR pour réalimenter l'ancien site de production via la réplication de données. Il peut toutefois être nécessaire, voire plus opportun ou efficace, de simplement retransférer les MVC physiques vers le site de production rétabli. Les méthodes choisies dépendront des exigences de récupération après sinistre.

Planification de tests DR

La préparation de la réelle récupération après sinistre est évaluée en testant l'efficacité et le rendement des systèmes et procédures de DR lors du rétablissement d'une charge globale de production sur un site de test DR donné. L'environnement de test DR peut être une plate-forme de test DR dédiée, mais il est généralement plus économique de partager des ressources entre les systèmes de production et de test DR. Les tests DR effectués en parallèle à la production et l'utilisation de ressources partagées avec la production constituent ce que l'on appelle les tests DR simultanés. Si une application doit s'exécuter en parallèle aux systèmes de production et de test DR, les planificateurs de la récupération après sinistre doivent s'assurer que ces deux instances de l'application ne se gêneront pas lorsqu'elles s'exécuteront simultanément. Le fait d'isoler les systèmes de production et de test DR sur des partitions LPAR distinctes et de limiter l'accès aux données de production à partir du système de test DR suffit généralement. Les tests DR sont souvent effectués petit à petit pour permettre des tests ciblés des diverses applications à différents moments, au lieu de tester la récupération de l'ensemble de l'environnement de production en une seule fois. Un test ciblé est essentiel pour réduire la quantité de matériel dédié requis pour le système de test DR. Par exemple, si les tests DR d'une application récupérable ne requièrent qu'un petit

sous-ensemble de ressources VSM, ces ressources peuvent être partagées entre les systèmes de production et de test DR, et réaffectées au système de test DR pour le cycle de test DR. Cette approche réduit les dépenses matérielles liées au système de test DR mais peut avoir une influence négative sur les performances du système de production lors de l'exécution du test DR. Toutefois, un cycle de test DR ne dédie généralement qu'un faible pourcentage des ressources partagées au système de test DR et les tests DR parallèles n'ont pas de réel impact sur l'environnement de production diminué. Néanmoins, certaines organisations ont mis en place des stratégies pour lutter contre l'altération ou la détérioration de la production afin de faciliter les tests DR. Vos auditeurs peuvent exiger une correspondance exacte entre les résultats des tests DR et les résultats de production pour certifier le processus de récupération après sinistre. Pour répondre à cette exigence, vous pouvez définir un point de synchronisation juste avant un cycle de production planifié, enregistrer une copie des résultats de production, reprendre le cycle de production à ce point de synchronisation sur le site de test DR et comparer la sortie avec les résultats de production enregistrés. Toute différence entre les résultats met en évidence un écart qui doit être analysé. L'impossibilité de résoudre à temps les écarts détectés peut mettre en péril la capacité de réelle récupération après sinistre d'une organisation. Qu'un test DR soit conçu pour récupérer une charge globale complexe ou une seule application, le processus doit suivre les mêmes procédures que celles en vigueur pour la réelle récupération après sinistre. Il s'agit de la seule méthode fiable pour s'assurer que le test DR a réussi.

Déplacement de données pour les tests DR

Il existe deux méthodes pour transférer les données d'application pour les tests DR sur un site de test DR : le déplacement de données physiques et le déplacement de données électroniques. Le déplacement de données physiques implique le transport de cartouches de bande physiques vers le site de test DR au cours d'un processus décrit ci-dessous et appelé exportation/importation physique. Le déplacement de données électroniques utilise des lecteurs de bande distants, des RTD distants ou des techniques de cluster VSM pour créer des copies des données d'application sur un site de test DR. Ces deux méthodes de déplacement de données permettent d'effectuer des tests DR, mais le déplacement de données électroniques évite le transfert de données physiques et les éventuels problèmes liés à la perte de bandes, etc. Le transfert électronique améliore également le temps d'accès aux données en les plaçant à l'endroit approprié pour la réelle récupération après sinistre, ou en transférant les données vers un tampon VSM en amont d'un cycle de test DR. Le déplacement de données électroniques pour les volumes virtuels s'effectue au sein d'un seul tapeplex à l'aide du clustering étendu VSM, ou entre deux tapeplex à l'aide de la fonction CTR (Cross-Tapeplex Replication). Pour les données situées sur un seul tapeplex, le logiciel CDRT (Concurrent Disaster Recovery Test) d'Oracle rationalise les tests DR.

Tests DR avec exportation/importation physique

Supposons que vous vouliez effectuer des tests DR pour une application de production utilisant des bandes virtuelles et physiques. Vous souhaitez tester cette application sur le site de test DR en répétant un cycle de production récent et en vérifiant que la sortie du test

correspond à la sortie de production récente. Pour préparer l'opération, vous devez enregistrer des copies des jeux de données d'entrée utilisés par le cycle de production et une copie de la sortie de production afin de les comparer. Supposons que le site de test DR soit isolé et ne partage aucun équipement avec la production. Vous pouvez effectuer le test DR à l'aide de ce processus d'exportation/importation physique.

Site de production :

1. Effectuez une copie des VTV et volumes physiques requis.
2. Exportez ces copies VTV.
3. Ejectez les copies MVC et les copies de volume physique associées à partir de l'ACS de production.
4. Transférez les MVC et les volumes physiques éjectés vers le site de test DR.

Site de test DR :

1. Entrez les volumes transférés dans l'ACS de récupération après sinistre.
2. Synchronisez les catalogues du système d'exploitation et le système de gestion des bandes avec les volumes entrés.
3. Importez les données VTV/MVC.
4. Exécutez l'application.
5. Comparez les résultats.
6. Ejectez tous les volumes entrés pour ce test.
7. Transférez à nouveau les volumes éjectés vers le site de production.

Site de production :

1. Entrez à nouveau les volumes transférés dans l'ACS de production.

Ce processus permet d'exécuter des tests DR en toute sécurité parallèlement à la production, car le système de test DR est isolé du système de production. Le système de test DR possède son propre CDS et le processus de test DR entre des informations de volume dans le CDS de test DR pour préparer le test DR, comme illustré ci-dessus. L'application récupérée peut alors effectuer le test avec les mêmes volumes et noms de jeux de données que ceux qu'elle utilise en production. Pour les jeux de données de bande virtuels, la fonction d'archivage sécurisé du logiciel LCM d'Oracle simplifie le placement des VTV sur les MVC et rationalise les étapes d'aller-retour ci-dessus pour exporter et éjecter des volumes sur le site de production, importer ces volumes sur le site de test DR et éjecter ces volumes pour qu'ils soient à nouveau transférés vers le site de production. L'exportation/importation physique induit des dépenses sur le site pour le traitement des bandes physiques, ainsi que des dépenses de coursier pour le transport des cartouches de bande entre les sites de production et de test DR. Les données sensibles transportées par coursier doivent figurer sur des cartouches de bande chiffrées. La rapidité d'exécution des tests DR est affectée par le temps de transport et de traitement des cartouches de bande lors de leur déplacement entre les sites.

Tests DR avec CDRT

Grâce à une planification et à un matériel suffisant sur les sites de production et de récupération après sinistre, CDRT, lorsqu'il est combiné au déplacement de données électroniques, permet d'éviter le transport de cartouches de bande physiques vers le site DR et d'effectuer des tests DR simultanés pour un coût inférieur à celui induit par le maintien d'un site de test DR dédié isolé. CDRT permet d'effectuer des tests DR sur presque toutes les charges globales de production, configurations, ou tous les RPO ou RTO imaginables. La procédure de test DR inclura quelques étapes supplémentaires pour démarrer CDRT et effectuer un nettoyage après un test DR. Avant d'exécuter un test DR avec CDRT, vous devez déplacer électroniquement toutes les données d'application et métadonnées système (informations du catalogue du système d'exploitation et du système de gestion de bandes) nécessaires au test vers le site de test DR. Vous pouvez déplacer électroniquement des données d'application par le biais du clustering VSM ou de la migration de copies VTV vers des MVC sur le site DR. Utilisez ensuite CDRT pour créer un CDS spécial pour le système de test DR, qui met en miroir le CDS de production. Les systèmes de production et de test DR sont des environnements distincts. L'environnement de test DR utilisera le CDS de test DR spécial au lieu du CDS de production. Etant donné que CDRT crée le CDS de test DR à partir des informations figurant dans le CDS de production, il contient des métadonnées pour l'ensemble des volumes qui ont été déplacés électroniquement vers le site de test DR avant le test DR. Cela permet aux applications de test DR d'utiliser les mêmes numéros de série de volume et noms de jeux de données de bande qu'en production. CDRT impose des restrictions opérationnelles au système de test DR pour empêcher l'environnement de récupération après sinistre d'interférer avec l'environnement de production. Vous pouvez renforcer ces protections en utilisant les fonctions VOLPARM/POOLPARM d'ELS pour définir des plages volser distinctes pour les MVC et les VTV provisoires qui seront exclusivement utilisées par CDRT. CDRT permet au système de test DR de lire les MVC de production et d'écrire dans son propre pool de MVC dédié, qui est effacé logiquement après chaque cycle de test DR. Pour les applications de bandes virtuelles, CDRT requiert au moins un périphérique VTSS dédié pendant la durée du cycle de test DR. Ces VTSS dédiés peuvent être temporairement réaffectés à partir de la production pour faciliter l'exécution d'un test DR et le système VSM de test DR peut accéder aux ACS de production parallèlement à la charge globale de production. La [Figure 1.9, « Cluster de production avec le nœud de cluster distant VTSS2 sur le site de test DR »](#) et la [Figure 1.10, « Configuration de production avec VTSS2 prêté pour les tests de récupération après sinistre CDRT »](#) illustrent la scission d'un cluster VSM de production pour prêter un périphérique de cluster au système de test DR du CDRT (ici VTSS2 sur le site de test DR). Lorsque ce cluster est scindé, vous devez modifier les stratégies de production pour remplacer la migration par la réplication, afin que VTSS1 crée des copies VTV redondantes sur le site DR dans ACS01. Ainsi, la capacité de VTSS1 ne sera pas saturée si le cluster est scindé. VTSS2 est mis hors ligne en production et en ligne sur la partition LPAR de test DR. Dans la [Figure 1.9, « Cluster de production avec le nœud de cluster distant VTSS2 sur le site de test DR »](#), CDRT a créé le CDS de test DR à partir d'une copie distante du CDS de production. Seul le système de production peut accéder aux volumes de VTSS1 et ACS00 au cours du cycle de test DR. De même, seul le système de test DR peut accéder à VTSS2. Les systèmes de production et de test DR partagent un accès

simultané aux volumes dans ACS01. Dans la [Figure 1.9](#), « Cluster de production avec le nœud de cluster distant VTSS2 sur le site de test DR » et la [Figure 1.10](#), « Configuration de production avec VTSS2 prêté pour les tests de récupération après sinistre CDRT », une copie distante du CDS de production sur le site de test DR, par exemple par le biais de la mise en miroir distante, est conservée pour s'assurer qu'un CDS de production à jour est disponible sur le site DR en vue de son utilisation pour la réelle récupération après sinistre. Notez toutefois que le CDS de test DR créé par CDRT à partir de la copie CDS distante est une version de test DR spéciale du CDS de production, utilisable uniquement par CDRT. Avant de reconstituer le cluster de production à la fin du cycle de test DR, le VTSS de la récupération après sinistre doit être purgé pour éviter la perte des données de production, comme cela serait le cas si VTSS2 contenait une version plus récente d'un VTV qui était également présent dans VTSS1. Vous devez également modifier les stratégies de production pour passer de la migration à la réplication lorsque le cluster est reconstitué. S'il n'est pas possible de scinder un cluster de production comme indiqué ici, vous pouvez conserver un VTSS distinct sur le site DR exclusivement pour les tests DR. Dans ce cas, les VTV nécessaires au test seront rappelés à partir des copies MVC.

Figure 1.9. Cluster de production avec le nœud de cluster distant VTSS2 sur le site de test DR

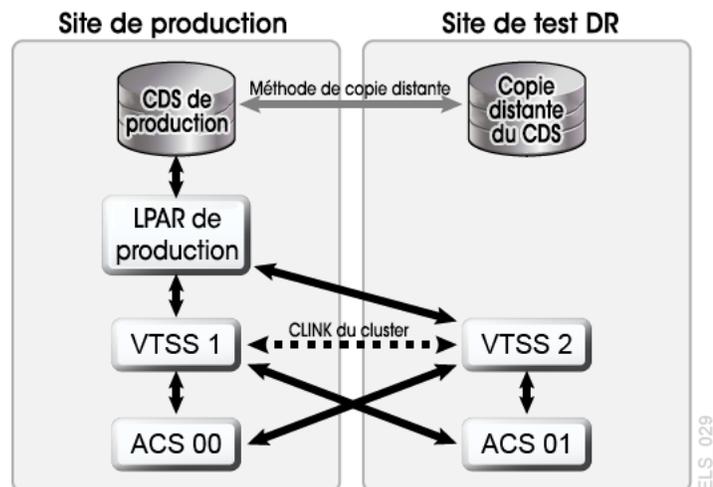
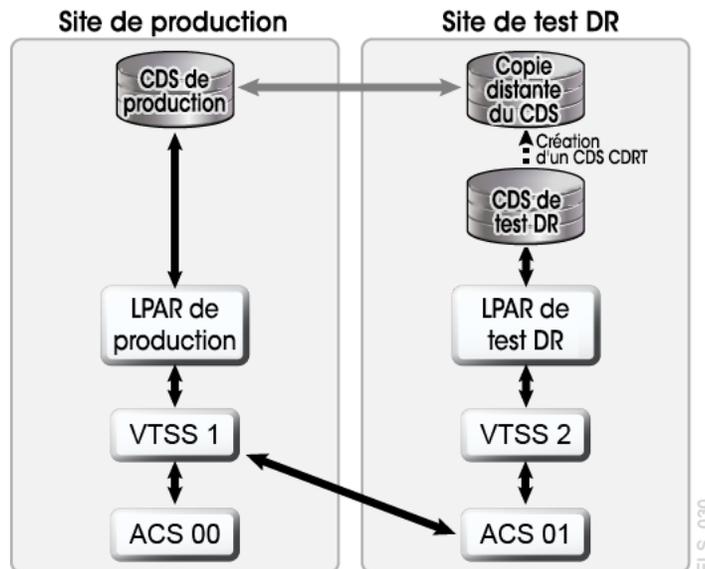
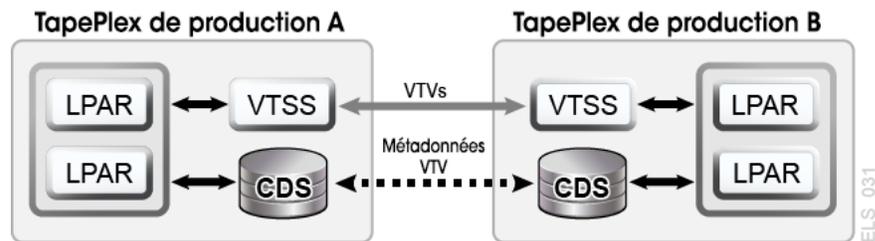


Figure 1.10. Configuration de production avec VTSS2 prêt pour les tests de récupération après sinistre CDRT



Tests DR avec fonction CTR (Cross-Tapeplex Replication) de VSM

La fonction CTR (Cross-Tapeplex Replication) de VSM active des architectures tapeplex de production, symétriques, en cluster, qui facilitent l'exécution de tests DR sans utiliser CDRT, sans avoir besoin de matériel VTSS dédié uniquement aux tests DR et sans modifier l'environnement de production pour les tests DR. Par exemple, la fonction CTR permet à chaque tapeplex de production de répliquer des données sur les autres tapeplex de production dans le même cluster CTR. Les clusters entre homologues de la fonction CTR de production peuvent éviter l'emploi d'un site de test DR dédié. La fonction CTR active de nombreux types d'architectures tapeplex en cluster différentes et facilite les tests DR d'une configuration ou d'une charge globale de production, avec des RPO ou RTO réalisables. Dans un exemple simple, un cluster CTR bidirectionnel relie symétriquement deux tapeplex de production et chaque tapeplex réplique des données sur l'autre TapePlex (Figure 1.11, « Cluster CTR de production symétrique pour les tests DR »). Un tapeplex récepteur entre un VTV répliqué dans son CDS à l'état de lecture seule et le marque comme appartenant au tapeplex émetteur. Dans cet exemple, les tests DR d'une application sur le tapeplex A impliquent la répllication des données d'application sur le tapeplex B et la récupération de l'application sur le tapeplex B.

Figure 1.11. Cluster CTR de production symétrique pour les tests DR

La symétrie de cette architecture de cluster CTR homologue signifie que l'application récupérée qui est testée sur le site homologue s'exécute de la même façon pendant un test DR qu'en production. Le CDS homologue contient toutes les informations de volume répliquées nécessaires aux tests DR, qui s'exécutent en parallèle à la production, tandis que le même matériel VTSS permet une utilisation simultanée par les charges globales de production et de test DR. Les clusters VTSS de production peuvent se trouver au sein de chaque TapePlex et n'ont pas besoin d'être scindés pour partager du matériel entre tapeplex pour les tests DR. Le tapeplex de production sur lequel sont exécutés les tests DR de l'application ne pouvant pas modifier les VTV répliqués par la fonction CTR, toutes les données de production répliquées sont totalement protégées lors du cycle de test DR. En outre, les tests DR basés sur la fonction CTR garantissent qu'une procédure de test DR validée produira des résultats identiques lors d'une réelle récupération après sinistre. Le logiciel hôte SMC enverra un message en cas de tentative de mise à jour d'un VTV répliqué à l'aide de la fonction CTR, qui sert à identifier l'application comme l'une de celles modifiant un jeu de données d'entrée existant. En suivant les pratiques recommandées ci-dessus pour la gestion des points de synchronisation, vous devez vérifier que l'environnement de production enregistre une copie de ce jeu de données avant qu'il ne soit modifié par l'application, au cas où une copie de sauvegarde serait nécessaire pour la récupération du point de synchronisation.

Chapitre 2. Exportations et importations physiques

Les fonctions EXPORT et IMPORT vous fournissent les outils pour créer des MVC portables physiquement. Sur le site source, utilisez la fonction EXPORT pour consolider des VTV sur des MVC (le cas échéant) et générez un fichier manifeste décrivant le contenu des MVC (VTV sur la MVC). Ejectez ensuite les MVC à partir du site source, transportez-les physiquement vers le site cible, puis importez-les, en utilisant le fichier manifeste pour mettre à jour le CDS à l'aide des informations figurant sur les MVC et VTV importés. Notez que vous pouvez importer des VTV dans un CDS même si VTCS n'est pas actif. Entrez ensuite les MVC dans le site cible.

Remarque :

- Si vous décidez de renvoyer les MVC exportées vers le système source, aucun traitement VTCS spécial n'est requis. Il vous suffit d'entrer les MVC dans un LSM sur le système source.
- Pour chaque VTV importé, les seules copies MVC qui seront créées sont destinées aux MVC qui ont été exportées et importées via les mêmes instructions. Cela s'avère particulièrement important lors de l'importation de VTV mis en duplex. Les VTV de ce type disposeront uniquement de copies sur les deux MVC après l'importation, si ces dernières se trouvent dans le même fichier manifeste et sont importées suite à l'exécution de la même instruction IMPORT.

Effectuez l'exportation à l'aide de l'une des méthodes générales suivantes :

- Exportation par VTV ou classe de gestion, qui consolide les VTV sélectionnées sur un nouveau jeu de MVC. Etant donné que la consolidation prend du temps et nécessite des ressources VTSS, l'option privilégiée consiste à effectuer l'exportation par MVC ou classe de stockage. Pour plus d'informations, voir "[Exportation et importation par classe de gestion.](#)"
- Exportation par MVC ou classe de stockage. Une exportation par classe de stockage ou MVC ne requiert aucun post-traitement de consolidation des VTV, ni aucun mouvement de données. L'exportation crée simplement un fichier manifeste qui décrit le contenu des MVC sélectionnées. Pour plus d'informations, voir "[Exportation et importation par classe de stockage.](#)"

Remarque :

Si vous exportez par :

- **volsers VTV** - Utilisez un rapport TMS, LCM ou VTVRPT pour identifier les VTV requis.
 - **volsers MVC** - Utilisez un rapport LCM ou MVCRPT pour identifier les MVC requises.
 - **Classe de gestion** - Consultez vos définitions de classe de gestion pour identifier les classes de gestion requises.
 - **Classes de stockage** - Consultez vos définitions de classe de stockage pour identifier les classes de stockage requises.
-

Exportation et importation par classe de gestion

Les exemples suivants présentent l'exportation et l'importation de MVC par classe de gestion.

Remarque :

Lorsqu'une exportation est effectuée par MGMTclas ou un VTV, le CDS ne gère pas les données de la MVC dans l'enregistrement de VTV. Dans ce cas, la MVC est signalée comme (E)exportée.

Exemple : exportation par classe de gestion à partir du système VSM source

Il s'agit de la phase "d'envoi" de l'exportation/importation, au cours de laquelle vous obtenez les données souhaitées sous forme de package déplacé à partir du système VSM source.

Pour effectuer une exportation à partir d'un système VSM source, procédez comme suit :

1. Identifiez les classes de gestion utilisées pour l'exportation.
2. Lancez l'exportation par classe de gestion :

```
//EXPORT EXEC PGM=SLUADMIN, PARM=' MIXED' REGION=6M
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//MOVE1 DD DSN=h1q.REMOTE2, DISP=(, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, (1,1), RLSE),
// DCB=(RECFM=FB, LRECL=80, BLKSIZE=27920)
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
EXPORT MGMT (PAY,ACCOUNT) MANIFEST(MOVE1)
```

Dans cet exemple, le fichier manifeste de sortie, qui est requis pour l'importation, est `<Emphasis Role="CodeInline">MOVE1</Emphasis>`. Puisque vous avez choisi l'exportation par classe de gestion, la fonction EXPORT consolide les VTV sélectionnés (en effectue des copies) sur les MVC d'exportation. Les MVC d'exportation sont marquées en tant que MVC en lecture seule et exportées dans le CDS. Elles sont désormais disponibles pour l'éjection à partir d'un LSM de système source. Ces copies

VTV consolidées sont des copies supplémentaires et ne sont pas enregistrées dans le CDS. Par exemple, si le VTV était mis en duplex avant l'exportation, le CDS enregistre les deux copies mises en duplex, mais pas la troisième copie supplémentaire utilisée pour la consolidation. Par conséquent, les VTV d'origine sont encore accessibles par le système source. Vous pouvez utiliser les données sur les VTV d'origine ou les supprimer et les réutiliser.

Attention :

Planifiez l'exportation à un moment où les données exportées ne sont pas mises à jour.

3. Supprimez les MVC à exporter du pool MVC. Pour plus d'informations, voir *Gestion du HSC et du VTCS*.
4. Ejectez les MVC à exporter à partir d'un LSM du système VSM source. Pour plus d'informations, voir *Gestion du HSC et du VTCS*.
5. Si vous le souhaitez, supprimez ou rendez indisponibles les VTV exportés sur le système source, ou réutilisez les données qu'ils contiennent.

Après l'exportation, le système source conserve les enregistrements CDS des VTV et MVC exportés. Les MVC d'exportation sont marqués comme exportées et en lecture seule dans le CDS du système source. A ce stade, vous disposez de deux options, en fonction de la raison pour laquelle vous avez exporté les VTV :

- **Si vous avez exporté les VTV pour fournir une copie de sauvegarde à un deuxième site**, laissez les VTV en lecture seule dans le CDS du système source, afin que leur mise à jour soit impossible.
- **Si vous déplacez définitivement les VTV exportés vers un deuxième site**, supprimez-les ou rendez-les indisponibles dans le CDS du système source. Servez-vous des utilitaires de suppression de HSC ou de la fonction LCM SYNCVTV pour supprimer les VTV exportés.

Exemple : importation par classe de gestion dans le système VSM cible

Un mois plus tard, vous êtes enfin prêt pour la partie "réception" (importation) de l'opération d'exportation/importation.

Pour effectuer une importation dans un système VSM cible, procédez comme suit :

1. Si les VTV et MVC que vous importez ne se trouvent pas dans le CDS du système cible, recréez vos définitions *POOLPARAM/VOLPARAM* pour ajouter ces volsers, comme décrit dans la section *Configuration du HSC et du VTCS*.

Si nécessaire, augmentez la taille du CDS sur le système VSM cible. Pour plus d'informations, voir *Configuration du HSC et du VTCS* ou *Gestion du HSC et du VTCS*.

Que faire s'il existe des volsers VTV en double sur les systèmes source et cible ? En général, procédez comme suit :

- Si les VTV qui se trouvent sur le système source sont plus récents que ceux qui figurent sur le système cible et ont les mêmes volsers, spécifiez *REPLACE(ALL)*.
 - Si vous déplacez les VTV du système source vers le système cible (exportation/importation initiale), spécifiez *REPLACE(NONE)*. Vous devez alors décider ce que vous voulez faire des VTV en double au cas par cas.
2. Entrez les MVC à importer dans un LSM de système VSM cible.

Pour plus d'informations, voir *Gestion du HSC et du VTCS*. Vous voulez que les MVC soient physiquement en place avant d'utiliser la fonction *IMPORT* pour informer le CDS qu'il dispose de nouveaux MVC et VTV.

3. Effectuez éventuellement une "validation" de la fonction *IMPORT* :

```
//IMPORT EXEC PGM=SLUADMIN,PARM='MIXED' REGION=6M
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//REMOTE1 DD DSN=h1q.REMOTE1,DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
      IMPORT MANIFEST(MOVE1) NOUPDATE
```

Il s'agit d'un exemple de JCL pour exécuter l'utilitaire *IMPORT*, où :

- le fichier manifeste est le manifeste d'exportation indiqué à l'étape 2 ;
- *REPLACE(NONE)* (option par défaut) indique que VTCS n'écrase pas les VTV en double ;
- *IMMDRAIN(NO)* (option par défaut) indique que VTCS ne purge pas tous les VTV importés dans l'espace VTSS ;
- *NOUPDATE* indique que le CDS n'a pas été mis à jour (validation uniquement) ;
- *INACTCDS* n'est pas spécifié, ce qui signifie que HSC est actif.

La validation est facultative mais fortement recommandée, car vous voulez réellement voir ce qui va se passer avant d'appuyer sur le bouton. Etudiez attentivement le rapport d'importation. Est-ce qu'il vous convient ? Passez à l'étape 4.

Remarque :

- *IMPORT* est valide uniquement si *FEATures VSM(ADVGMGT)* est indiqué.
 - Vérifiez que le CDS cible dispose des mêmes fonctionnalités (activées par le niveau de CDS) que le CDS source. Par exemple, si la fonction de prise en charge des tailles de page VTV volumineuses est activée et que des VTV de 2/4 Go ont été créés sur le CDS source, le CDS cible doit être doté des mêmes fonctionnalités, sous peine de voir l'importation échouer.
-

4. Lancez une exécution réelle de la fonction *IMPORT* :

```
//IMPORT EXEC PGM=SLUADMIN,PARM='MIXED' REGION=6M
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//REMOTE1 DD DSN=h1q.REMOTE1,DISP=SHR
```

```
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
IMPORT MANIFEST(MOVE1) REPLACE(ALL)
```

Il s'agit d'un exemple de JCL pour exécuter l'utilitaire *IMPORT* où, comme dans la validation, *REPLACE(NONE)* (option par défaut) indique que VTCS n'écrase pas les VTV en double.

Remarque :

Que devez-vous faire pour renvoyer les MVC vers le système source ? Vous pouvez spécifier *IMMDRAIN(YES)* pour purger les MVC d'importation.

5. Ajustez vos définitions de VTV, le cas échéant. Par exemple, vous devez définir les nouveaux VTV sur le TMS du système cible.
6. Effectuez l'une des opérations suivantes :
 - Exécutez éventuellement *MVCMaint* pour que les MVC importées soient inscriptibles. VTCS importe les MVC en tant que MVC en lecture seule. Pour les rendre inscriptibles, exécutez *MVCMaint* en indiquant l'option *READONLY OFF*. Vous voudrez certainement utiliser les nouvelles MVC en tant que système cible, ce qui ne constitue que la première étape.

Ajoutez ensuite les MVC importées au pool MVC, comme décrit dans la section *Gestion du HSC et du VTCS*. A ce stade, les MVC peuvent être récupérées, purgées, migrées, rappelées, etc.

- Si vous avez spécifié *IMMDRAIN(YES)* à l'étape 4, vous pouvez renvoyer les MVC vers le système source.

Exportation et importation par classe de stockage

Les exemples suivants présentent l'exportation et l'importation par classe de stockage à partir d'un VSM source.

Remarque :

Lorsqu'une exportation est effectuée par *STORclas* ou une MVC, le CDS gère les données de la MVC dans l'enregistrement de VTV. Dans ce cas, la MVC est signalée comme étant en lecture seule (R).

Exemple : exportation par classe de stockage à partir du système VSM source

Il s'agit de la phase "d'envoi" de l'exportation/importation, au cours de laquelle vous obtenez les données souhaitées sous forme de package déplacé à partir du système VSM source.

Pour effectuer une exportation à partir d'un système VSM source, procédez comme suit :

1. Identifiez les classes de stockage utilisées pour l'exportation.
2. Lancez l'exportation par classe de stockage :

```
//EXPORT EXEC PGM=SLUADMIN, PARM=' MIXED ' REGION=6M
//STEPLIB DD DSN=h1q .SEALINK, DISP=SHR
//MOVE2 DD DSN=h1q .REMOTE2, DISP=(, CATLG, DELETE),
// UNIT=SYSDA, SPACE=( CYL, (1, 1), RLSE),
// DCB=(RECFM=FB, LRECL=80, BLKSIZE=27920)
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
EXPORT STOR(OFF1, OFF2) MANIFEST(MOVE2)
```

Dans cet exemple, le fichier manifeste de sortie, qui est requis pour l'importation, est *MOVE2*. Etant donné que vous avez effectué une exportation par classe de stockage, le système crée un fichier manifeste, mais aucune consolidation de VTV n'a lieu. Les MVC d'exportation sont marquées en tant que MVC en lecture seule dans le CDS. Elles sont désormais disponibles pour l'éjection à partir d'un LSM de système source. Les VTV qui se trouvaient sur les MVC qui ont été supprimées du LSM peuvent toujours être utilisés, à condition qu'ils soient situés sur d'autres MVC.

Attention :

Planifiez l'exportation à un moment où les données exportées ne sont pas mises à jour.

3. Supprimez les MVC à exporter du pool MVC. Pour plus d'informations, voir *Gestion du HSC et du VTCS*.
4. Ejectez les MVC à exporter à partir d'un LSM du système VSM source. Pour plus d'informations, voir *Gestion du HSC et du VTCS*.
5. Si vous le souhaitez, supprimez ou rendez indisponibles les VTV exportés sur le système source, ou réutilisez les données qu'ils contiennent.

Après l'exportation, le système source conserve les enregistrements CDS des VTV et MVC exportés. Les MVC d'exportation sont marqués comme exportés et en lecture seule dans le CDS du système source. A ce stade, vous disposez de deux options, en fonction de la raison pour laquelle vous avez exporté les VTV :

- **Si vous avez exporté les VTV pour fournir une copie de sauvegarde à un deuxième site**, laissez les VTV en lecture seule dans le CDS du système source, afin que leur mise à jour soit impossible.
- **Si vous déplacez définitivement les VTV exportés vers un deuxième site**, supprimez-les ou rendez-les indisponibles dans le CDS du système source. Servez-vous des utilitaires de suppression de HSC ou de la fonction LCM SYNCVTV pour supprimer les VTV exportés.

Exemple : importation par classe de stockage dans le système VSM cible

Un mois plus tard, vous êtes enfin prêt pour la partie "réception" (importation) de l'opération d'exportation/importation.

Pour effectuer une importation dans un système VSM cible, procédez comme suit :

1. Si les VTV et MVC que vous importez ne se trouvent pas dans le CDS du système cible, recréez vos définitions *POOLPARAM/VOLPARAM* pour ajouter ces volsers, comme décrit dans la section *Configuration du HSC et du VTCS*.

Si nécessaire, augmentez également la taille du CDS sur le système VSM cible. Pour plus d'informations, voir *Configuration du HSC et du VTCS* ou *Gestion du HSC et du VTCS*.

Que faire s'il existe des volsers VTV en double sur les systèmes source et cible ? En général, procédez comme suit :

- Si les VTV qui se trouvent sur le système source sont plus récents que ceux qui figurent sur le système cible et ont les mêmes volsers, spécifiez *REPLACE(ALL)*.
 - Si vous déplacez les VTV du système source vers le système cible (exportation/importation initiale), spécifiez *REPLACE(NONE)*. Vous devez alors décider ce que vous voulez faire des VTV en double au cas par cas.
2. Entrez les MVC à importer dans un LSM de système VSM cible.

Pour plus d'informations, voir *Gestion du HSC et du VTCS*. Examinez ce qui se passe ici. Vous voulez vraiment que les MVC soient physiquement en place avant d'utiliser la fonction *IMPORT* pour informer le CDS qu'il dispose de nouveaux MVC et VTV.

3. Effectuez éventuellement une "validation" de la fonction *IMPORT*.

```
//IMPORT EXEC PGM=SLUADMIN, PARM='MIXED' REGION=6M
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//REMOTE1 DD DSN=h1q.REMOTE1, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
IMPORT MANIFEST(REMOTE1) NOUPDATE
```

Il s'agit d'un exemple de JCL pour exécuter l'utilitaire *IMPORT*, où :

- le fichier manifeste est le manifeste d'exportation indiqué à l'étape 2 ;
- *REPLACE(NONE)* (option par défaut) indique que VTCS n'écrase pas les VTV en double ;
- *IMMDRAIN(NO)* (option par défaut) indique que VTCS ne purge pas tous les VTV importés dans l'espace VTSS ;
- *NOUPDATE* indique que le CDS n'a pas été mis à jour (validation uniquement) ;

- *INACTCDS* n'est pas spécifié, ce qui signifie que HSC est actif.

La validation est facultative mais fortement recommandée, car vous voulez réellement voir ce qui va se passer avant d'appuyer vraiment sur le bouton. Etudiez attentivement le rapport d'importation. Est-ce qu'il vous convient ? Passez à l'étape 4.

Remarque :

- *IMPORT* est valide uniquement si *FEATures VSM(ADVGMT)* est indiqué.
 - Vérifiez que le CDS cible dispose des mêmes fonctionnalités (activées par le niveau de CDS) que le CDS source. Par exemple, si la fonction de prise en charge des tailles de page VTV volumineuses est activée et que des VTV de 2/4 Go ont été créés sur le CDS source, le CDS cible doit être doté des mêmes fonctionnalités, sous peine de voir l'importation échouer.
-

4. Lancez une exécution réelle de la fonction *IMPORT* :

```
//IMPORT EXEC PGM=SLUADMIN, PARM=' MIXED' REGION=6M
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//REMOTE1 DD DSN=h1q.REMOTE1, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
IMPORT MANIFEST(REMOTE1)
```

Il s'agit d'un exemple de JCL pour exécuter l'utilitaire *IMPORT* où, comme dans la validation, *REPLACE(NONE)* (option par défaut) indique que VTCS n'écrase pas les VTV en double.

Remarque :

Que devez-vous faire pour renvoyer les MVC vers le système source ? Vous pouvez spécifier *IMMDRAIN(YES)* pour purger les MVC d'importation.

5. Ajustez vos définitions de VTV, le cas échéant.
6. Effectuez l'une des opérations suivantes :
- Exécutez éventuellement *MVCMAINT* pour que les MVC importées soient inscriptibles. VTCS importe les MVC en tant que MVC en lecture seule. Pour les rendre inscriptibles, exécutez *MVCMAINT* en indiquant l'option *READONLY OFF*. Vous voudrez certainement utiliser les nouvelles MVC en tant que système cible, ce qui ne constitue que la première étape.

Ajoutez ensuite les MVC importées au pool MVC, comme décrit dans la section *Gestion du HSC et du VTCS*. A ce stade, les MVC peuvent être récupérées, purgées, migrées, rappelées, etc.

- Si vous avez spécifié *IMMDRAIN(YES)* à l'étape 4, vous pouvez renvoyer les MVC vers le système source.

Exportation/importation dans une VLE

Vous pouvez exporter des VTV et des MVC vers une VLE partagée entre deux TapePlex à condition qu'ils possèdent des pools de MVC séparés.

Pour effectuer une exportation vers une VLE connectée à deux TapePlex, inspirez-vous de l'exemple suivant :

1. Supposons que le TapePlex A possède le MVCPOOL A00-A10 et le TapePlex B, le MVCPOOL B00-B10.
2. Exportez A00-A10 vers le pool de MVC du VLE TapePlex A de la VLE.
3. Importez le pool de MVC A00-A10 dans le TapePlex B à l'aide de la commande `IMPORT REPLACE(UPDATED)`.

Chapitre 3. Utilisation de la fonction d'archivage sécurisé externe d'ELS

La fonction d'archivage sécurisé externe d'ELS (ELS Vault) remplace et améliore considérablement la fonction précédente, VSM Offsite Vault. La fonction ELS Vault offre les améliorations suivantes en matière d'archivage sécurisé des volumes de bande réels :

- Utilisation du CDS HSC pour le stockage de l'archivage sécurisé et des données de volume archivées. L'utilisation des informations d'archivage sécurisé du CDS à la place du TMS évite :
 - le risque d'erreur humaine lors du renvoi des volumes vers l'environnement automatisé ;
 - l'arrivée de volumes dans l'emplacement d'archivage sécurisé lorsqu'ils ne figurent pas sur la liste déroulante de renvoi ;
 - de laisser accidentellement les volumes d'archivage sécurisé renvoyés dans l'environnement automatisé.
- Utilisation de LCM pour gérer le processus d'archivage sécurisé, qui dispose de trois méthodes :
 - archivage sécurisé des MVC pour la récupération après sinistre (DR) ; pour plus d'informations, voir "[Archivage sécurisé de MVC pour la récupération après sinistre](#)".
 - archivage sécurisé des MVC pour la conservation à long terme (LTR) ; pour plus d'informations, voir "[Archivage sécurisé de MVC pour la conservation à long terme](#)".
 - éjection de volumes vers un archivage sécurisé au sol ; pour plus d'informations, voir "[Ejection de volumes spécifiques vers un espace d'archivage sécurisé local \(sol\)](#)".

Préparation de l'archivage sécurisé externe d'ELS

La première étape consiste à définir la zone de volumes d'archivage sécurisé du CDS HSC. Pour ce faire, exécutez l'utilitaire SLUADMIN SET VAULTVOL. Par exemple :

```
SET VAULTVOL NBRVOLS(40000)
```

Remarque :

- Si vous avez besoin d'ajouter ultérieurement des volumes d'archivage sécurisé supplémentaires, vous devez utiliser *MERGEcds*. Il convient donc de définir suffisamment de volumes lors de la définition initiale pour répondre en amont à vos besoins futurs.
 - Vous devez disposer d'un nombre suffisant de blocs LIBRES dans votre CDS pour pouvoir accueillir le nombre de volumes que vous prévoyez d'archiver (volumes de bande réels), sans oublier un espace supplémentaire pour répondre aux augmentations de volume. Pour plus d'informations sur le calcul de l'espace pour les volumes d'archivage sécurisé, voir *Configuration du HSC et du VTCS*. Pour savoir comment développer la taille du CDS si elle n'est pas suffisamment vaste pour contenir les volumes archivés, voir *Gestion du HSC et du VTCS*.
-

La deuxième étape consiste à définir les espaces d'archivage sécurisé qui contiendront vos volumes archivés. Pour ce faire, exécutez l'utilitaire *SLUADMIN SET VAULT* pour chaque espace d'archivage sécurisé. Par exemple :

```
SET VAULT ADD NAME(DRVLT1) SLOTS(10000) DESC('DR Vault')
SET VAULT ADD NAME(LTRVLT1) SLOTS(20000) DESC('LTR Vault')
SET VAULT ADD NAME(FLOOR) SLOTS(500) DESC('Floor Vault')
```

Remarque :

Le nombre total d'emplacements dans l'ensemble des espaces d'archivage sécurisé que vous définissez ne doit pas être supérieur au nombre de volumes indiqué dans l'instruction *VAULTVOL*.

Alors que HSC définit les espaces d'archivage sécurisé et les volumes archivés, LCM est responsable de leur gestion. Vous devez particulièrement prendre en compte les paramètres d'archivage sécurisé LCM suivants :

GRACEPERIOD

Nombre de jours entre la sélection d'un volume en vue de son renvoi par l'espace d'archivage sécurisé et son renvoi effectif vers l'environnement automatisé. Le délai de grâce offre une marge de sécurité afin que les nouveaux volumes arrivent dans l'espace d'archivage sécurisé avant le renvoi des anciens volumes. Si aucune valeur n'est indiquée, la valeur par défaut est trois jours.

DEFAULT

Les paramètres *DEFAULT* et *GRACEPERIOD* s'excluent mutuellement. Le premier est généralement utilisé pour l'espace d'archivage sécurisé "au sol" (armoire manuelle), qui contient automatiquement tous les volumes éjectés de l'environnement automatisé à l'aide de *LCM EJECT(ASNEEDED)*. D'autres volumes éjectés de l'ACS peuvent également être affectés à cet espace d'archivage sécurisé ; par exemple, un volume qui est actif mais qui ne constitue plus un bon candidat à l'automatisation. Il s'agit en général d'un espace d'archivage sécurisé défini, qui est constitué d'armoires placées sur le sol du centre de données. *DEFAULT* représente un délai de grâce de zéro jour, afin de permettre à ces volumes d'être à nouveau entrés dans l'ACS à tout moment.

Notez également que des options d'éjection standard sont disponibles pour tous les volumes d'archivage sécurisé. Elles indiquent par exemple quels CAP utiliser, la

définition d'un message d'éjection, le mode utilisé pour les éjections et si ces dernières s'effectueront selon l'ordre des emplacements ou des séries de volume.

Création de MVC pour la récupération après sinistre et la conservation à long terme

Lorsque des MVC sont archivées de façon sécurisée pour la récupération après sinistre, vous devez réaliser au moins deux copies de chaque VTV sur des MVC distinctes, l'une restant sur site, tandis que l'autre est éjectée et placée dans un espace d'archivage sécurisé externe. Cette opération est effectuée en affectant deux classes de stockage sur la classe de gestion qui est affectée au VTV.

Vous souhaitez protéger vos données mais vous voulez utiliser le moins d'espace MVC possible, comme suit :

- **Définissez un nombre de classes de stockage aussi bas que possible.** Un nombre trop élevé de classes de stockage signifie généralement un nombre trop important de MVC et/ou MVC avec peu de VTV.
- **Utilisez un nombre de VTSS aussi bas que possible pour créer des MVC.** Si vous le pouvez, utilisez un seul VTSS pour créer la totalité des MVC d'archivage sécurisé.

Quels sont les autres facteurs à prendre en compte lors de la création et de l'archivage sécurisé des MVC ? Prenez les éléments suivants en considération :

- **Tout d'abord, les VTV doivent être migrés dès que possible vers des MVC d'archivage sécurisé** afin de pouvoir être éjectés et déplacés vers l'espace d'archivage sécurisé, car ces VTV ne sont généralement pas utilisés en entrée pour d'autres étapes du travail.
- **Ensuite, les VTV de récupération après sinistre expirant à des dates différentes,** pensez à les regrouper par dates d'expiration sur les MVC. Toutefois, pour réduire le nombre total de MVC créées à envoyer vers l'espace d'archivage sécurisé, limitez le nombre de ces groupes. Comme une opération visant à consolider les VTV sur un nombre inférieur de MVC aura déjà eu lieu, l'avantage représenté par l'utilisation de plus de deux groupes est minime, à savoir un pour les VTV avec des dates d'expiration très proches, 7 jours par exemple, et l'autre pour tous les autres volumes. Les VTV sous le contrôle du catalogue doivent être considérés comme faisant partie du deuxième groupe, car il n'existe aucun moyen de connaître leur date d'expiration réelle.
- **Enfin, alors qu'il n'est pas nécessaire de disposer d'espaces d'archivage sécurisé distincts pour la récupération après sinistre et la conservation à long terme,** il peut être intéressant de placer les MVC de récupération après sinistre dans un espace d'archivage sécurisé distinct. Cela permet de regrouper ces volumes pour les envoyer au besoin vers le site DR.

Pour les données LTR, les éléments à prendre en compte sont légèrement différents. Tout d'abord, par définition, les données de conservation à long terme n'expireront pas avant une longue période. Ainsi, contrairement aux données de récupération après sinistre qui expireront à terme, les MVC de conservation à long terme ne seront pas fragmentées. Un

traitement périodique initial de ces MVC tentera de remplir autant que possible les MVC archivées, comme c'est le cas pour les MVC de récupération après sinistre, mais une fois qu'elles seront suffisamment remplies, ces MVC resteront statiques. Il n'est pas nécessaire de disposer de plusieurs classes de stockage pour ces volumes. Notez toutefois que vous pouvez vouloir migrer immédiatement certaines données LTR, tandis que d'autres données peuvent être migrées une fois qu'elles auront été sélectionnées par la migration automatique VTCS. Par conséquent, vous voudrez donc peut-être disposer d'une classe de stockage mais de deux classes de gestion pour les données LTR.

Considérations relatives à DELSCR lors de l'utilisation de la fonction d'archivage sécurisé

Utilisez le paramètre *DELSCR* de l'instruction *MGMTclas* pour indiquer si VSM doit supprimer les VTV mis à l'état provisoire (*DELSCR(YES)*) afin de libérer de l'espace sur le tampon VTSS et de l'espace MVC. Spécifiez *DELSCR(YES)* pour les classes de gestion DR et LTR. Si vous spécifiez *DELSCR(YES)*, utilisez uniquement la fonction LCM *SYNCVTV* pour la synchronisation de volumes provisoires. Pour plus d'informations sur l'utilisation de la fonction LCM pour gérer la synchronisation de volumes provisoires, consultez le *Guide de l'utilisateur LCM*.

Opérations exécutées lors du renvoi des volumes archivés de façon sécurisée vers l'ACS

Le processus d'entrée de HSC a été modifié pour la fonction d'archivage sécurisé d'ELS, de sorte que chaque volume entré soit vérifié afin de déterminer s'il s'agit d'un volume archivé de façon sécurisée en externe. Pour ce type de volumes, l'une des deux actions suivantes sera exécutée en fonction de la date de renvoi figurant dans l'enregistrement de l'espace d'archivage sécurisé du CDS :

- Si la date de renvoi est passée, le volume est entré, les métadonnées de volume stockées qui proviennent du processus d'éjection sont restaurées et le volume est supprimé des enregistrements de l'espace d'archivage sécurisé.
- Si la date de renvoi **n'est pas** passée ou qu'aucune date de renvoi n'a été définie pour le volume, le volume est entré, les métadonnées de volume stockées qui proviennent du processus d'éjection sont restaurées, mais le volume est conservé dans les enregistrements de l'espace d'archivage sécurisé et sera automatiquement éjecté lors du prochain processus d'éjection. Pourquoi cela se passe-t-il ainsi ? Il existe plusieurs raisons à cela, les deux les plus courantes étant les suivantes : le volume a été renvoyé pour un processus de récupération des données, ou le volume incorrect a été retiré de l'espace d'archivage sécurisé (cause très fréquente). Quelle qu'en soit la raison, le volume appartient à l'espace d'archivage sécurisé et sera renvoyé vers cet emplacement, où il reprendra son statut protégé.

Notez qu'il s'agit du processus pour les volumes archivés dans un espace d'archivage sécurisé physique. Pour les volumes archivés dans une bibliothèque distante, le processus est

légèrement différent ; voir " [Archivage sécurisé DR avec des MVC dans une bibliothèque distante](#)".

Archivage sécurisé de MVC pour la récupération après sinistre

Dans un scénario de récupération après sinistre, vous cherchez globalement à optimiser l'utilisation du tampon VTSS, en garantissant une migration rapide des données critiques, tout en conservant la disponibilité des données.

Archivage sécurisé DR de base avec des MVC

Dans cette approche, des volumes DR sont créés chaque jour. Par conséquent, le traitement de leur déplacement vers l'espace d'archivage sécurisé DR s'effectue chaque jour pour s'assurer que les MVC sont déplacées hors site et protégées en toute sécurité.

Remarque :

- Tous les VTV créés à des fins de récupération après sinistre et toutes les bandes natives, notamment la bande du fichier manifeste créée à l' [Etape 2 – Exportation des MVC d'archivage sécurisé](#), sont contrôlés par le TMS du site (pour l'expiration de volume). Ce processus se limite à l'archivage sécurisé des MVC connexes et des volumes natifs sélectionnés impliqués.
 - Les MVC ne peuvent être affectées qu'à un seul espace d'archivage sécurisé. Pour affecter un volume à un nouvel espace d'archivage sécurisé, son affectation à un espace d'archivage sécurisé antérieur doit d'abord être supprimée.
 - L' [Etape 7 – Préparation des MVC archivées pour le renvoi](#) " démarre le traitement périodique, qui permet le recyclage des MVC de récupération après sinistre qui ont été fragmentées ou uniquement remplies partiellement lors de leur création. Pour ce faire, effectuez des purges MVC "logiques" des MVC archivées, en utilisant les copies des VTV toujours actifs qui se trouvent sur les MVC locales. Le traitement périodique réduit le nombre total de volumes dans l'espace d'archivage sécurisé, tout en garantissant que l'ensemble des activités connexes est réduit au minimum grâce à l'utilisation de critères de sélection propres à l'environnement en question. Une fois que la purge "logique" d'une MVC archivée a abouti, la date de renvoi est définie dans le CDS pour ce volume. Dans le cas de volumes natifs, tels que les bandes du fichier manifeste, les volumes qui sont passés à l'état provisoire dans le TMS sont sélectionnés et définis pour le renvoi. Choisissez la fréquence d'exécution du traitement périodique, qui peut être quotidienne ou mensuelle.
-

Etape 1 – Création de VTV/MVC d'archivage sécurisé

Les VTV d'archivage sécurisé de récupération après sinistre sont créés à l'aide d'une classe de gestion qui pointe vers deux classes de stockage, l'une créant des MVC qui resteront dans l'environnement local et l'autre créant des MVC qui sont archivées de façon sécurisée. Par exemple :

```
STOR NAME(DRLOC)ACS(00) MEDIA(STK1RD)
STOR NAME(DRVLT1) ACS(00) MEDIA(STK1RD)
```

Etape 2 – Exportation des MVC d'archivage sécurisé

Exportez les MVC d'archivage sécurisé via un fichier de paramètres LCM, comme indiqué dans l'exemple suivant.

```

Options
  NoSync
  NoTMS
;
Vault
  Name('DRVLT')
  NoSync
  GracePeriod(3)
;
Action
  Export
  Control(Serial )
  MVC
  DSN(DRVAULT.MANIFEST)
  Storageclass(DRVLT1)
  Vault('DRVLT')
;

```

Dans cet exemple :

- L'instruction *OPTIONS* indique *NOSYNC* et *NOTMS*, car l'archivage sécurisé n'utilise pas les informations de TMS et aucune métadonnée TMS n'est requise.
- L'instruction *VAULT* indique *DRVLT* comme espace d'archivage sécurisé de récupération après sinistre.
- L'instruction *ACTION EXPORT* spécifie :
 - qu'elle exporte des MVC par volser ;
 - qu'elle crée un fichier manifeste d'exportation (*DRVAULT.MANIFEST*), ici, un volume de l'ACS qui est éjecté et stocké avec les MVC de récupération après sinistre ;
 - qu'elle renvoie vers la classe de stockage d'archivage sécurisé créée à l' [Etape 1 – Création de VTV/MVC d'archivage sécurisé](#) ;
 - qu'elle définit l'espace d'archivage sécurisé DR (*DRVLT*) et lui affecte des MVC qui ne lui étaient pas affectées auparavant.

Remarque :

Les MVC exportées sont marquées en lecture seule par le processus d'exportation.

Vous pouvez créer plusieurs classes de stockage d'archivage sécurisé (par exemple, pour séparer les MVC dont les VTV portent des dates d'expiration différentes). Vous pouvez affecter différentes classes de stockage d'archivage sécurisé au même espace d'archivage sécurisé dans une seule instruction *ACTION EXPORT*. Par exemple, les instructions suivantes affectent les classes de stockage *DRVLT1* et *DRVLT2* au même espace d'archivage sécurisé (*DRVLT*) :

```

Action
  Export
  Control(Serial )
  MVC
  DSN(DRVAULT.MANIFEST)
  Storageclass(DRVLT1
                DRVLT2)
;

```

```
Vault('DRVLT')
;
```

Etape 3 – Ecriture de jeux de données supplémentaires sur la bande du fichier manifeste (facultatif)

Une fois que vous avez créé la bande du fichier manifeste à l' [Etape 2 – Exportation des MVC d'archivage sécurisé](#), vous pouvez exécuter un travail qui copie le jeu de données de contrôle (CDS) HSC, le catalogue TMS, le catalogue système et d'autres jeux de données "ponctuels" importants sur la bande du fichier manifeste afin de fournir des points de reprise DR supplémentaires.

Etape 4 – Ejection de MVC d'archivage sécurisé

Ejectez des MVC d'archivage sécurisé via un fichier de paramètres LCM, comme indiqué dans l'exemple suivant.

```
Options
  NoSync
  NoTMS
;
Vault
  Name('DRVLT')
  GracePeriod(3)
;
Action
  Eject
  When(
    (inLsm)
    and
    (VaultName EQ 'DRVLT')
  )
  Control(Serial)
  Ejmsg('Move to DR Vault')
;
```

Dans cet exemple :

- L'instruction *OPTIONS* indique *NOSYNC* et *NOTMS*, car l'archivage sécurisé n'utilise pas les informations de TMS et aucune métadonnée TMS n'est requise.
- L'instruction *VAULT* indique *DRVLT* comme espace d'archivage sécurisé de récupération après sinistre.
- L'instruction *ACTION EJECT* spécifie :
 - qu'elle éjecte par volser les MVC affectées à *DRVLT* ;
 - le message d'éjection.

Etape 5 - Ejection de volumes natifs (y compris de la bande du fichier manifeste)

Ejectez des volumes natifs (y compris la bande du fichier manifeste) à l'aide d'un fichier de paramètres *LCM*, comme indiqué dans l'exemple suivant.

```
Options
  NoSync
  ;
TMS
  RMM
  Dateform(J)
  DDname(LCMTMSDB)
  ;
Vault
  Name('DRVLT')
  GracePeriod(3)
  ;
Action
  Eject
  When(
    (InLsm)
    and
    (DataSetName EQ 'DRVLT.MANIFEST')
    and
    (TMSScratch EQ False)
  )
  Control(Serial)
  Ejmsg('Move to DR Vault')
  ;
```

Dans cet exemple :

- L'instruction *OPTIONS* indique *NOSYNC*, car l'archivage sécurisé n'utilise pas les informations de TMS.
- L'instruction *VAULT* indique *DRVLT* comme espace d'archivage sécurisé de récupération après sinistre.
- L'instruction *ACTION EJECT* spécifie :
 - Ejectez la bande du fichier manifeste.
 - Ejectez les volumes natifs qui n'ont pas été supprimés par le TMS.
 - le message d'éjection.

Etape 6 – Création d'une liste déroulante de volumes pour renvoi à partir de l'espace d'archivage sécurisé

Créez une liste déroulante à l'aide d'un fichier de paramètres *LCM*, comme indiqué dans l'exemple suivant.

```
Options
  NoSync
  NoTMS
  ;
Vault
  Name('DRVLT')
  GracePeriod(3)
  ;
Report
  Volume
  Sysout(*)
  Title('Return Report')
```

```

When(
  (VaultName EQ 'DRVLT')
  and
  (VaultReturnDate LE TODAY)
  and
  (VaultReturnDate NE MISSING)
)
Column (Serial,
        VaultSlot)
;

```

Dans cet exemple :

- L'instruction *OPTIONS* indique *NOSYNC* et *NOTMS*, car l'archivage sécurisé n'utilise pas les informations de TMS et aucune métadonnée TMS n'est requise.
- L'instruction *VAULT* indique *DRVLT* comme espace d'archivage sécurisé de récupération après sinistre.
- L'instruction *REPORT VOLUME* crée un rapport qui répertorie les volumes dans l'espace d'archivage sécurisé qui ont atteint la date de retour qui leur a été précédemment affectée. Il s'agit d'un exemple simple, mais vous pouvez ajouter des critères de sélection supplémentaires pour les volumes à renvoyer.

Remarque :

- *TODAY* et *MISSING* sont des valeurs de date uniques. *TODAY* correspond à la date d'exécution de LCM. *MISSING* indique qu'il n'y a pas de valeur de date. Dans cet exemple, aucune date n'a été définie. Les deux conditions sont requises car une absence de date reviendrait à indiquer *Less Than* (antérieure à la date actuelle).
 - Les étapes 4, 5 et 6 peuvent être combinées au sein d'une étape de travail unique. Dans certains cas, l'étape 6 est exécutée périodiquement, en général la veille de la date prévue pour le renvoi des volumes par l'espace d'archivage sécurisé.
-

Etape 7 – Préparation des MVC archivées pour le renvoi

Préparez les MVC archivées pour le renvoi via un fichier de paramètres *LCM*, comme indiqué dans l'exemple suivant.

```

Options
  NoSync
  NoTMS
;
Vault
  Name('DRVLT')
  GracePeriod(3)
;
Action
  Drain
  When(
    (MVC EQ True)
    and
    (VaultName EQ 'DRVLT')
    and
    (MVCVTVCount LE 30)
  )
;

```

```

and
(MVCInUse LE 30)
and
(Days_Since(VaultAssignmentDate) GT 7)
)
Control(MCVTVCount
Ascending)
Limit(30)
;

```

Dans cet exemple :

- L'instruction *OPTIONS* indique *NOSYNC* et *NOTMS*, car l'archivage sécurisé n'utilise pas les informations de TMS et aucune métadonnée TMS n'est requise.
- L'instruction *VAULT* indique *DRVLT* comme espace d'archivage sécurisé de récupération après sinistre.
- Pour les MVC qui se trouvent actuellement dans l'espace d'archivage sécurisé DR, l'instruction *ACTION DRAIN* spécifie une purge des MVC qui :
 - contiennent moins de 30 VTV ;
 - sont utilisés moins de 30 % du temps ;
 - sont restés dans l'espace d'archivage sécurisé pendant au moins 7 jours ;
 - limitent le nombre de MVC renvoyées à 30 au maximum ;
 - comportent une date de renvoi définie sur 3 jours par le paramètre *GracePeriod*.

Lorsque vous créez un fichier de paramètres pour purger des MVC, vous voulez définir un juste milieu entre les cycles de traitement de la purge et le recyclage nécessaire des MVC fragmentées ou partiellement remplies. Les paramètres *MCVTVCount*, *MVCInUse*, *Days_Since* et *LIMIT* permettent de trouver un équilibre entre ces conditions requises.

Etape 8 – Préparation des volumes natifs archivés pour le renvoi

Préparez les volumes natifs pour le renvoi via un fichier de paramètres *LCM*, comme indiqué dans l'exemple suivant.

```

Options
  NoSync
  ;
TMS
  RMM
  Dateform(J)
  DDname(LCMTMSDB)
  ;
Vault
  Name('DRVLT')
  GracePeriod(3)
  ;
Action
  Vault
  Return
  When(

```

```

Not (MVC)
and
(VaultName EQ 'DRVLT')
and
(TMSScratch EQ True)
)
;

```

Dans cet exemple :

- L'instruction *OPTIONS* indique *NOSYNC*, car l'archivage sécurisé n'utilise pas les informations de TMS.
- L'instruction *TMS RMM* est nécessaire pour ajouter le traitement des métadonnées TMS.
- L'instruction *VAULT* indique *DRVLT* comme espace d'archivage sécurisé de récupération après sinistre.
- L'instruction *ACTION VAULT RETURN* définit une date de renvoi (à l'aide du paramètre *GracePeriod*) pour les volumes qui ne sont pas des MVC et qui sont à l'état provisoire dans le TMS.

Etape 9 - Entrée de volumes renvoyés

Les volumes qui apparaissent sur le rapport créé à l'" [Etape 6 – Création d'une liste déroulante de volumes pour renvoi à partir de l'espace d'archivage sécurisé](#) " sont supprimés de l'espace d'archivage sécurisé DR et renvoyés vers l'environnement local. Lorsque vous entrez ces volumes dans l'ACS, HSC vérifie pour chaque volume si la date de renvoi affectée pour l'espace d'archivage sécurisé est passée. Si tel est le cas, le volume a atteint sa date de renvoi planifiée et est supprimé de l'espace d'archivage sécurisé. Les MVC renvoyées sont ensuite éligibles pour la migration et les volumes natifs sont mis à l'état provisoire dans le CDS lors du traitement LCM SYNC suivant. Si la date de renvoi pour l'espace d'archivage sécurisé n'est pas passée, le volume est éjecté par l' [Etape 4 – Ejection de MVC d'archivage sécurisé](#).

Si vous le souhaitez, vous pouvez combiner les étapes 7 et 8 dans un fichier de paramètres *LCM* unique, comme indiqué dans l'exemple suivant :

```

Options
  NoSync
;
TMS
  RMM
  Dateform(J)
  DDname(LCMTMSDB)
Vault
  Name('DRVLT')
  GracePeriod(3)
;
Action
  Drain
  When(
    (MVC EQ True)
    and
    (VaultName EQ 'DRVLT')
    and

```

```

(MVCVTVCount LE 30)
and
(MVCInUse LE 30)
and
(Days_Since(VaultAssignmentDate) GT 7)
)
Control(MCVTVCount
Ascending)
Limit(30)
;
Action
Vault
Return
When(
Not (MVC)
and
(VaultName EQ 'DRVLT')
and
(TMSScratch EQ True)
)
;

```

Archivage sécurisé DR pendant plusieurs semaines

Certains sites peuvent choisir d'utiliser un processus qui dure plusieurs semaines, dans lequel le traitement de la récupération après sinistre inclut une sauvegarde hebdomadaire des données critiques sur la totalité des volumes, suivie de sauvegardes incrémentielles quotidiennes les six jours suivants. Le processus d'archivage sécurisé externe déplace les volumes hors site le jour de sa création. Ce processus suppose un cycle de quatre semaines qui conserve les données de récupération après sinistre hors site et se termine au début de la quatrième semaine. Notez que la seule modification entre la procédure "[Archivage sécurisé DR de base avec des MVC](#)" et le processus qui dure plusieurs semaines réside dans la différence de critères de sélection pour les étapes 7 et 8, comme indiqué ci-dessous. Pour ce faire, définissez les dates d'expiration de sorte que les VTV d'archivage sécurisé sur les MVC associées et les bandes du fichier manifeste (ainsi que toute autre bande native impliquée) expirent tous 22 jours après leur création.

L'échéancier sur plusieurs semaines se présente comme suit :

- Jour 1 – Sauvegardes sur la totalité des volumes (expiration le jour 22).
- Jour 1 – Sauvegarde incrémentielle n°1 (expiration le jour 22).
- Jour 3 – Sauvegarde incrémentielle n°2 (expiration le jour 22).
- Jour 4 – Sauvegarde incrémentielle n°3 (expiration le jour 22).
- Jour 5 – Sauvegarde incrémentielle n°4 (expiration le jour 22).
- Jour 6 – Sauvegarde incrémentielle n°5 (expiration le jour 22).
- Jour 7 – Sauvegarde incrémentielle n°6 (expiration le jour 22).
- Jours 8 à 21 – Les volumes restent hors site.
- Jour 22 – Les sauvegardes et les bandes du fichier manifeste réalisées entre les jours 1 à 7 expirent et les VTV passent en mode provisoire dans le CDS via le processus LCM VTVSYNC. Les MVC sont purgées à l'aide du paramètre suivant dans les critères de

sélection de l'" [Etape 7 – Préparation des MVC archivées pour le renvoi](#) " et de l' [Etape 8 – Préparation des volumes natifs archivés pour le renvoi](#).

```
DAYS SINCE (VaultAssignmentDate) GT 15
```

La date de renvoi pour les MVC purgées et les bandes du fichier manifeste est définie sur le jour 25.

Remarque :

Toutes les MVC affectées à l'espace d'archivage sécurisé au cours des 7 premiers jours du cycle sont purgées. Si la date d'expiration des VTV de récupération après sinistre a été définie correctement, il ne doit y avoir aucun VTV en cours à ce stade et le processus de purge logique s'exécutera rapidement. Si les VTV en cours sont écrits sur une nouvelle cartouche MVC, une date d'expiration incorrecte a peut-être été définie.

- Jours 23 et 24 – Les volumes restent hors site.
- Jour 25 – Les volumes archivés entre les jours 1 à 7 sont renvoyés, sortent de l'état archivé et sont disponibles pour la réutilisation.
- Jour 29 – Le cycle se répète.

Remarque :

Certains sites peuvent choisir de ne prendre hors site que les sauvegardes sur la totalité des volumes, en conservant les sauvegardes incrémentielles sur site. Dans ce cas, placez les MVC incrémentielles, ainsi que les volumes natifs et de fichier manifeste connexes dans un conteneur verrouillé et non ouvert jusqu'à son renvoi le jour 25. A ce stade, tous les volumes créés le jour 1 doivent avoir expiré.

Archivage sécurisé DR avec des MVC dans une bibliothèque distante

Dans ce processus, les volumes sont archivés dans une bibliothèque distante (ACS) au lieu d'être archivés dans un espace d'archivage sécurisé physique. Ce processus est similaire à celui présenté à la section " [Archivage sécurisé DR de base avec des MVC](#) ", avec les exceptions suivantes :

- Etapes 1 à 3 - Aucune modification.
- Etape 4 - Supprimée car aucune MVC archivée n'est éjectée.
- Etape 5 - Comme dans l'étape 4, les volumes natifs ne sont pas éjectés. A la place, une instruction *Action Vault Assign* affecte des volumes natifs à l'espace d'archivage sécurisé à l'aide des mêmes critères de sélection que ceux utilisés pour l'exécution et l'éjection de ces volumes natifs, comme indiqué dans l'exemple suivant.

```
Options
  NoSync
  ;
TMS
```

```
RMM
Dateform(J)
DDname(LCMTMSDB)
;
Vault
Name('DRVLT')
GracePeriod(3)
;
Action
Vault
Assign
Vault('DRVLT')
When(
(InLsm)
and
(DataSetName EQ 'DRVLT.MANIFEST')
and
(TMSScratch EQ False)
)
;
```

- Etape 6 - Supprimée car aucun volume n'est réentré.
- Etape 7 - Aucune modification.

Remarque :

Le traitement de la purge s'effectuant au sein de la bibliothèque distante, les purges s'exécutent plus efficacement que le traitement de la purge logique des volumes dans un espace d'archivage sécurisé manuel.

- Etape 8 - Aucune modification.
- Etape 9 - Dans le processus de base, l'affectation de l'espace d'archivage sécurisé est supprimée par le traitement de l'entrée qui n'a pas lieu dans le scénario de bibliothèque distante. Pour supprimer le volume archivé, utilisez l'instruction *Action Vault Release*. Les volumes archivés à libérer sont sélectionnés par le biais des paramètres *Vault Name* et *Return Date*, comme c'était le cas précédemment pour la génération d'une liste déroulante. L'instruction *Action Vault Release* ne traite que les volumes archivés dont la date de renvoi (*Return Date*) est passée.

Vous trouverez ci-dessous un exemple d'instruction *Action Vault Release*.

```
Options
NoSync
NoTMS
;
Vault
Name('DRVLT')
```

```

    GracePeriod(3)
    ;
Action
    Vault
    Release
    When(
        (VaultName EQ 'DRVAULT')
        and
        (VaultReturnDate LE TODAY)
        and
        (VaultReturnDate NE MISSING)
    )
    ;

```

Archivage sécurisé de MVC pour la conservation à long terme

Le processus d'utilisation des MVC dans la conservation à long terme (LTR) est pratiquement identique à celui décrit dans le processus de base pour la récupération après sinistre, qui est présenté à la section " [Archivage sécurisé DR de base avec des MVC](#)". Les deux principaux éléments à prendre en compte pour la conservation à long terme sont les suivants :

- Il est possible que le déplacement vers l'espace d'archivage sécurisé, " [Etape 2 – Exportation des MVC d'archivage sécurisé](#) " à " [Etape 5 - Ejection de volumes natifs \(y compris de la bande du fichier manifeste\)](#)", ne soit pas effectué quotidiennement, afin que davantage de volumes LTR soient complètement remplis avant leur archivage sécurisé.
- Les VTV de conservation à long terme n'expirant pas avant une longue période, le processus périodique décrit risque de se produire à des intervalles moins fréquents. Etant donné qu'il y aura toujours des MVC LTR qui ne seront que partiellement remplies, ce type de MVC, qui contient moins de VTV et utilise moins de MVC, devra être traité à un moment, afin que les VTV situés sur les MVC partiellement remplies soient consolidés sur un nombre inférieur de MVC LTR.

Il sera peut-être nécessaire d'effectuer les purges logiques des MVC LTR ultérieurement pour déplacer les données archivées vers un nouveau média. Le processus de base exécute facilement cette activité en sélectionnant simplement les critères appropriés et en limitant le nombre de MVC LTR archivées qui sont traitées simultanément. Lors de chaque exécution, les VTV encore actifs sont déplacés vers le nouveau média, les nouvelles MVC sont déplacées vers l'espace d'archivage sécurisé et les MVC purgées logiquement sont renvoyées en vue de leur réutilisation ou destruction.

Ejection de volumes spécifiques vers un espace d'archivage sécurisé local (sol)

De nombreux sites demandant la suppression de volumes spécifiques dans l'environnement automatisé, ils doivent stocker efficacement ces volumes dans des armoires au sein de

l'environnement local. Cette exigence peut venir du fait qu'ils ne souhaitent plus utiliser certains volumes, ou qu'ils désirent garder les volumes disponibles pendant un certain temps. Plusieurs espaces d'archivage sécurisé locaux/au sol peuvent être définis. Des volumes spécifiques peuvent être envoyés vers différents espaces d'archivage sécurisé, mais un seul espace d'archivage sécurisé local/au sol peut être défini en tant qu'espace d'archivage sécurisé par défaut. La date en cours est automatiquement affectée comme date de renvoi à tous les volumes placés dans un espace d'archivage sécurisé local/au sol. Ainsi, ces volumes peuvent être renvoyés vers l'environnement automatisé et supprimés de l'espace d'archivage sécurisé qui leur a été affecté sans qu'aucune action supplémentaire ne soit nécessaire.

L'exemple suivant présente un exemple de fichier de paramètres LCM pour l'éjection de volumes spécifiques vers un espace d'archivage sécurisé au sol.

```
Options
  NoSync
  ;
TMS
  RMM
  Dateform(J)
  DDname(LCMTMSDB)
  ;
Vault
  Name('FLOOR')
  Default
  ;
Action
  Eject
  When(
    (InLsm EQ True)
    and
    (DaysSinceReference GT 100)
    and
    (MVC EQ False)
    and
    Not
    (DataSetName Matches 'HMIG.**')
  )
  Control(
    VaultSlot
    Ascending
  )
  Ejmsg('Move to Floor Vault')
  ;
Manage
  ACSID(00)
  Numfree(500)
  ;
```

Dans cet exemple :

- L'instruction *OPTIONS* indique *NOSYNC*, car l'archivage sécurisé n'utilise pas les informations de TMS.
- L'instruction *TMS RMM* est nécessaire pour ajouter le traitement des métadonnées TMS.
- L'instruction *VAULT* indique *FLOOR* comme espace d'archivage sécurisé au sol par défaut.

- L'instruction *ACTION EJECT* indique d'éjecter vers l'espace d'archivage sécurisé au sol les volumes qui :
 - se trouvent dans le LSM ;
 - n'ont pas été référencés depuis plus de 100 jours ;
 - *ne sont pas* des MVC ;
 - ont pour masque de nom de jeu de données *HMIG. ***.
- L'instruction *ACTION EJECT* spécifie également :
 - que les volumes sont traités par ordre volser croissant ;
 - que l'éjection s'effectue par numéros d'emplacement TMS.
 - le message d'éjection.
- L'instruction *MANAGE* spécifie :
 - que les volumes situés dans l'ACS 00 sont gérés ;
 - que la présence de 500 cellules libres est garantie dans l'ACS.

Chapitre 4. Utilisation de la réplication entre TapePlex dans une solution de récupération après sinistre

Le " *Exportations et importations physiques* " explique comment créer des MVC portables "d'exportation" à partir d'un site source, afin de les déplacer physiquement vers un site cible, et comment importer les MVC (et les VTV qu'elles contiennent) sur le site cible. Grâce à la fonction CTR (Cross-Tapeplex Replication, réplication entre Tapeplex), vous n'avez plus besoin d'utiliser la méthode PTAM (Pickup Truck Access Method) pour déplacer des MVC d'un site à l'autre. A la place, déplacez électroniquement les VTV du site source vers le site cible, à savoir d'un TapePlex à un autre, où ils sont ensuite migrés vers des MVC, éliminant ainsi l'étape PTAM. Lors du déplacement d'une copie du VTV du Tapeplex source vers le TapePlex cible, une copie des métadonnées du VTV est transférée du CDS du TapePlex source vers le CDS du TapePlex cible. Notez que le TapePlex source continue à "posséder" et gérer la suppression des VTV CTR.

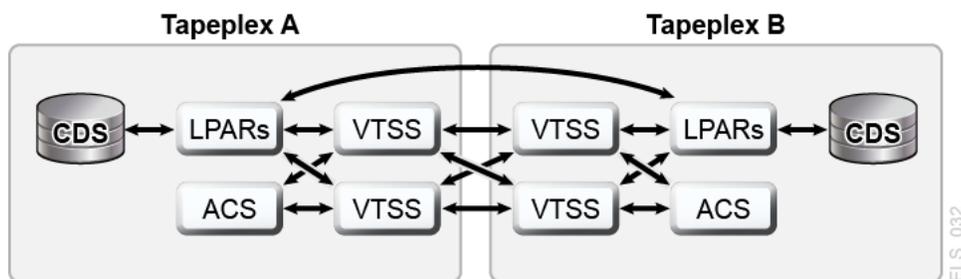
Attention :

Si vous utilisez la fonction CTR, l'arrêt de SMC fait cesser l'envoi des métadonnées vers un TapePlex CTR par VTCS, ce qui arrête effectivement le transfert de données. Par conséquent, si vous utilisez une fonctionnalité HSC qui se sert des services de communication SMC, comme CTR, vous devez vous assurer que l'activité HSC est mise au ralenti ou que HSC a pris fin avant d'arrêter SMC.

Fonctionnement de CTR

Examinez une réplication entre Tapeplex comme celle présentée à la [Figure 4.1](#), « Configuration de la fonction CTR d'ELS ».

Figure 4.1. Configuration de la fonction CTR d'ELS



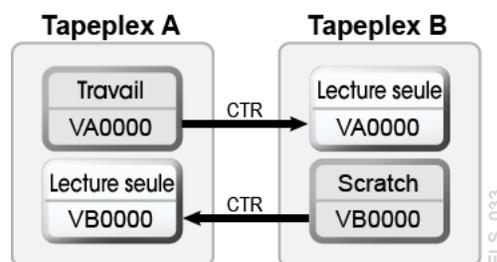
Comme indiqué à la [Figure 4.1, « Configuration de la fonction CTR d'ELS »](#) :

- CTR utilise la connexion entre deux VTSS (CLINK) situés dans deux TapePlex distincts pour envoyer des données d'un VTSS à l'autre. La connexion peut être unidirectionnelle ou bidirectionnelle.
- CTR utilise les services de la fonctionnalité client/serveur de SMC pour envoyer des métadonnées du TapePlex émetteur vers le TapePlex récepteur. Notez que vous N'AVEZ PAS besoin d'utiliser la fonctionnalité client/serveur pour communiquer entre SMC et HSC pour vous servir de CTR, mais vous devez définir des commandes HTTP et SERVER dans SMC pour permettre le transfert de métadonnées.
- Etant donné qu'il existe des CDS distincts (et gérés séparément) sur chaque site, une perte de connectivité ou la non-disponibilité du matériel sur un site n'affecte pas directement les autres sites.
- Les conditions requises pour la configuration et la connexion physique sont simples.
- Vous pouvez exécuter des tests DR simultanés plus facilement et sans interruption du travail existant (sans avoir recours à l'utilitaire CDRT).
- Vous pouvez désormais basculer automatiquement la charge globale d'un site à l'autre.
- Les plages de volume VTV pour les deux TapePlex sont indiquées à la [Figure 4.2, « Relations des volumes VTV à l'intérieur d'un site »](#). **Notez que** chaque TapePlex dispose de son propre ensemble de volumes inscriptibles et que ces derniers sont mis en miroir sur l'autre TapePlex par des versions en lecture seule.
- Dans la configuration présentée, les deux VTSS sur le TapePlex émetteur sont connectés aux deux VTSS sur le TapePlex récepteur afin d'optimiser la résilience.

Remarque :

Dans les configurations CTR et VTSS en cluster, vous devez vérifier que les 16 premiers VTD de chaque VTSS (0-F) sont réservés à la réplication. Ces périphériques doivent être HORS LIGNE sur MVS et leurs chemins doivent être en ligne sur chaque hôte serveur HSC. VTCS n'inscrit pas les 16 premiers VTD auprès de SMC/HSC, ce qui empêche le montage de VTV sur ces VTD.

Figure 4.2. Relations des volumes VTV à l'intérieur d'un site



Lisez la section "[Considérations relatives aux VTV CTR en lecture seule](#)", puis allez à "[Configuration pour la fonction CTR](#)".

Considérations relatives aux VTV CTR en lecture seule

Lorsque vous utilisez la fonction CTR, tous les VTV répliqués d'un site à l'autre sont en lecture seule sur le site distant. Bien que ces VTV puissent être supprimés (et leurs numéros de série de volume respectifs réutilisés) par le TapePlex distant en cas de sinistre réel, leur statut en lecture seule ne peut pas être modifié tant que les volumes n'ont pas le statut SCRATCH. Notez que les volumes situés dans un pool *POOLPARM EXTERNAL* ne peuvent jamais prendre le statut SCRATCH.

Par conséquent, si vous choisissez d'utiliser la fonction CTR comme stratégie de continuité de l'activité de de récupération après sinistre, vous devez vérifier que vos applications ne tentent pas de mettre à jour ces volumes, soit pendant un test DR, soit lors d'un sinistre réel. Les scénarios suivants doivent être pris en compte :

1. Les applications qui utilisent l'attribut *DISP=MOD* dans le JCL ou l'allocation dynamique pour ajouter des données à un jeu de données existant doivent implémenter un mécanisme de point de reprise/redémarrage et enregistrer un point de reprise avant des créer des volumes *DISP=MOD*. Ces applications sont récupérées en effectuant un redémarrage au point de reprise et doivent recréer ces volumes *DISP=MOD*, le cas échéant, une fois qu'elles ont redémarré. Notez que l'utilisation de l'attribut *DISP=MOD* lui-même ne pose pas de problème avec la fonction CRT. Tant que l'application dispose d'un point de reprise qui lui permet de mettre de côté des mises à jour partielles, ou d'une conception qui autorise la sortie de nouvelles données sur un nouveau volume, elle doit s'exécuter sur des VTV en lecture seule ne présentant pas de problème.
2. Si des VTV qui sont répliqués sur un autre TapePlex appartiennent à HSM, le processus suivant permet de démarrer la collecte de données sur un nouveau volume et évite de mettre à jour les VTV HSM existants :
 - a. Marquez les volumes existants saturés.
 - b. Si nécessaire, modifiez *ARCCMD* pour *USERUNITTABLE*, *MIGRATION*, *BACKUP* et *RECYCLE*.
 - c. Vérifiez que *RECYCLEDALLOCFREQ* est défini sur 1. Cela permet à l'allocation HSM d'affecter un nouveau volume et un nouveau périphérique, le cas échéant.
 - d. En fonction de *MGMTCLAS VTVSIZE*, définissez *PERCENTFULL* :

Pour les VTV de 800 Mo, définissez HSM PERCENTFULL sur 97.

Pour les VTV de 4 Go, définissez HSM PERCENTFULL sur 450.

Pour les VTV de 32 Go, définissez HSC PERCENTFULL sur 3 600.

Les restrictions *DISP=MOD* ci-dessus sont également valables pour les applications qui empilent des jeux de données sur un volume existant.

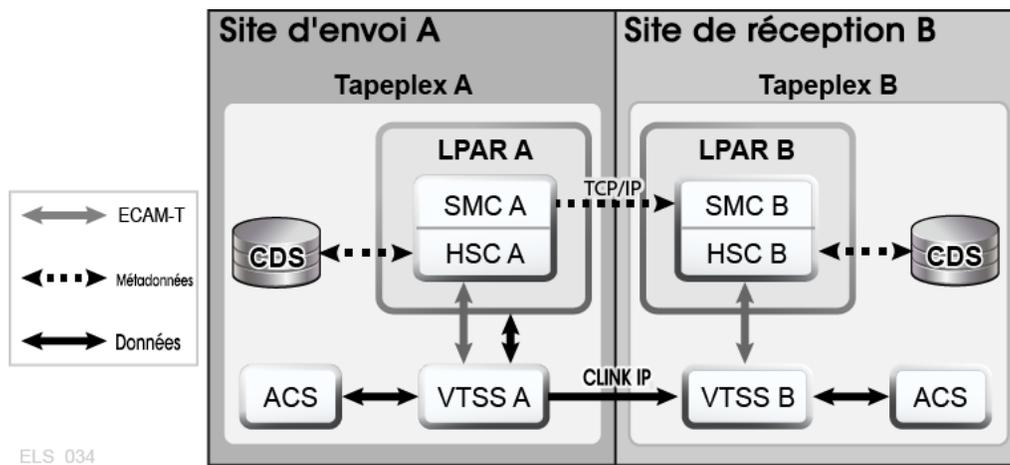
Configuration pour la fonction CTR

La [Figure 4.3, « Configuration CTR »](#) présente un exemple de configuration CTR. Dans ce système, le VTSS VTSSA se trouve dans le TapePlex TAPEPLXA et dispose de CLINK "partenaire" vers le VTSS VTSSB situé dans le TapePlex TAPEPLXB. Les VTV répliqués sur VTSSB se trouvent désormais dans le CDS de TAPEPLXB, tout comme les MVC vers lesquelles les VTV sont ensuite migrés. Les VTV sont donc répliqués sur les TapePlex, puis migrés localement. Les VTSS figurant dans le TapePlex émetteur ne peuvent pas disposer de connexions aux RTD dans le TapePlex récepteur.

Remarque :

L'exemple suivant présente une CTR unidirectionnelle. Pour effectuer une CTR bidirectionnelle, définissez simplement les instructions de configuration et de contrôle client/serveur de SMC de manière identique sur les deux TapePlex. Notez qu'un seul TapePlex peut également recevoir des VTV de plusieurs autres TapePlex. Pour définir une configuration dans laquelle un TapePlex reçoit des données de plusieurs autres TapePlex, il vous suffit d'ajouter des noms de TapePlex supplémentaires à *CONFIG of TAPEPLXB*.

Figure 4.3. Configuration CTR



Configuration et démarrage de CTR

Pour configurer et démarrer l'exemple de système CTR présenté dans la rubrique " Configuration pour la fonction CTR ", procédez comme suit :

1. Vérifiez que votre système répond aux conditions requises pour les VTSS en cluster, qui sont décrites à la section *Installation d'ELS*.
2. Démarrez le serveur HTTP sous le SMC exécuté sur l'hôte LPARB.

Vous pouvez effectuer cette opération dans le fichier CMDS de votre SMC. Par exemple :

```
HTTP START PORT(999)
```

3. Définissez vos commandes TAPEPLEX et SERVER sur l'hôte LPARB.

Là encore, vous pouvez effectuer cette opération dans le fichier CMDS de votre SMC.
Par exemple :

```
TAPEPLEX NAME(TAPEPLXA) LOCSUB(HSCA)
TAPEPLEX NAME(TAPEPLXB)
SERVER NAME(REMB)TAPEPLEX(TAPEPLXB) HOSTNAME(LPARB) PORT(999)
```

Remarque :

Dans l'exemple de configuration, le TapePlex TAPEPLXB existe (du point de vue du TapePlex TAPEPLXA) dans le seul but de conserver un CDS contenant des métadonnées relatives aux VTV qui ont été répliqués à partir de TAPEPLXA. Toutefois, si les définitions HSC ou VTCS dans les TapePlex TAPEPLXA et TAPEPLXB utilisent les mêmes adresses de périphérique référant des périphériques physiques différents, vous devez définir des commandes *SMC UNITATTR* pour indiquer à SMC quel TapePlex définit les périphériques sur son hôte. Bien que l'instruction *UNITATTR* doive indiquer un modèle (*MODEL*), si le modèle spécifié ne correspond pas à celui signalé par le TapePlex, le modèle réel remplace le modèle *UNITATTR MODEL*. Vous trouverez ci-dessous un exemple d'instruction *SMC UNITATTR* à utiliser si les TapePlex TAPEPLXA et TAPEPLXB définissent tous deux la plage d'adresses 9000-90FF :

```
UNITATTR ADDR(9000-90FF) TAPEPLEX(TAPEPLXA) MODEL(VIRTUAL)
```

4. Codez une console *CONFIG* pour TapePlex A, comme indiqué dans l'[Exemple 4.1](#), « Console *CONFIG* pour TapePlex A ».

Dans cette figure, notez les points suivants :

- L'instruction *TAPEPLEX* définit ce TapePlex.
- Les instructions *CLINK* définissent les CLINK qui sont utilisés pour la fonction CTR de VTSSA vers VTSSB.
- Le paramètre de réplication conditionnelle dans les instructions *CONFIG GLOBAL* est *CHANGED* pour TAPEPLXA.

5. Codez une console *CONFIG* pour TapePlex B, comme indiqué dans l'[Exemple 4.2](#), « Console *CONFIG* pour TapePlex B ».

Dans cette figure, notez les points suivants :

- L'instruction *TAPEPLEX* inclut un paramètre *RECVPLEX=TAPEPLXA* pour indiquer que TAPEPLXB peut recevoir des VTV de TAPEPLXA.
- Il n'existe aucune instruction *CLINK* , car les CLINK sont définis dans la console *CONFIG* pour TAPEPLXA.

Exemple 4.1. Console *CONFIG* pour TapePlex A

```
//CREATCFG EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSCNTL DD DSN=h1q.TAPEPLXA.DBASEPRM, DISP=SHR
```

```
//SLSCNTL2 DD DSN=h1q.TAPEPLXA.DBASESEC,DISP=SHR
//SLSSTBY DD DSN=h1q.TAPEPLXA.DBASESBY,DISP=SHR
//SLSPRINT DD   SYSOUT=*
//SLSIN DD   *
CONFIG RESET CDSLEVEL(V62ABOVE)
GLOBAL MAXVTV=65000 MVCFREE=60 VTVATTR=SCRATCH RECALWER=YES
LOCKSTR=STK_VTCS_LOCKS VTVPAGE=LARGE REPLICAT=CHANGED
RECLAIM THRESHLD=70 MAXMVC=30 START=98 CONMVC=1
TAPEPLEX THISPLEX=TAPEPLXA
VTSS NAME=VTSSA LOW=71 HIGH=80 MAXMIG=8 MINMIG=1 RETAIN=10
RTD NAME=VSMA1A00 DEVNO=1A00 CHANIF=0C
RTD NAME=VSMA1A01 DEVNO=1A01 CHANIF=0D
RTD NAME=VSMA1A02 DEVNO=1A02 CHANIF=0K
RTD NAME=VSMA1A03 DEVNO=1A03 CHANIF=0L
RTD NAME=VSMA2A08 DEVNO=2A08 CHANIF=1C
RTD NAME=VSMA2A09 DEVNO=2A09 CHANIF=1D
RTD NAME=VSMA2A0A DEVNO=2A0A CHANIF=1K
RTD NAME=VSMA2A0B DEVNO=2A0B CHANIF=1L
VTD LOW=7900 HIGH=79FF
VTD LOW=8900 HIGH=89FF
CLINK VTSS=VTSSA CHANIF=0G REMPLEX=TAPEPLXB PARTNER=VTSSB
CLINK VTSS=VTSSA CHANIF=00 REMPLEX=TAPEPLXB PARTNER=VTSSB
```

Exemple 4.2. Console CONFIG pour TapePlex B

```
//CREATCFG EXEC PGM=SLUADMIN,PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//SLSCNTL DD DSN=h1q.TAPEPLXB.DBASEPRM,DISP=SHR
//SLSCNTL2 DD DSN=h1q.TAPEPLXB.DBASESEC,DISP=SHR
//SLSSTBY DD DSN=h1q.TAPEPLXB.DBASESBY,DISP=SHR
//SLSPRINT DD   SYSOUT=*
//SLSIN DD   *
CONFIG RESET CDSLEVEL(V62ABOVE)
GLOBAL MAXVTV=65000 MVCFREE=60 VTVATTR=SCRATCH RECALWER=YES
LOCKSTR=STK_VTCS_LOCKS VTVPAGE=LARGE
RECLAIM THRESHLD=70 MAXMVC=30 START=98 CONMVC=1
TAPEPLEX THISPLEX=TAPEPLXB RECVPLEX=TAPEPLXA
VTSS NAME=VTSSB LOW=75 HIGH=80 MAXMIG=8 MINMIG=1 RETAIN=10
RTD NAME=VSMB3A00 DEVNO=3A00 CHANIF=0C
RTD NAME=VSMB3A01 DEVNO=3A01 CHANIF=0D
RTD NAME=VSMB3A02 DEVNO=3A02 CHANIF=0K
RTD NAME=VSMB3A03 DEVNO=3A03 CHANIF=0L
RTD NAME=VSMB4A08 DEVNO=4A08 CHANIF=1C
```

```

RTD  NAME=VSMB4A09 DEVNO=4A09 CHANIF=1D
RTD  NAME=VSMB4A0A DEVNO=4A0A CHANIF=1K
RTD  NAME=VSMB4A0B DEVNO=4A0B CHANIF=1L

```

Définition de stratégies pour CTR

Consultez les procédures suivantes pour définir des stratégies pour la fonction CTR.

Stratégies pour le TapePlex émetteur

Pour définir des stratégies pour le TapePlex émetteur (TAPEPLXA) de l'exemple de système CTR présenté à la [Figure 4.3, « Configuration CTR »](#), procédez comme suit :

1. Créez des définitions *MVC POOLPARAM/VOLPARAM* pour TAPEPLXA :

```

POOLPARAM TYPE(MVC) NAME(MVCPLA) INITMVC(YES) MVCFREE(25) -
MAXMVC(98) THRESH(85) START(98)
VOLPARAM VOLSER(AM1000-AM1299) MEDIA(STK1R)

```

2. Créez des définitions de pool provisoire *VTV POOLPARAM/VOLPARAM* pour TAPEPLXA :

```

POOLPARAM TYPE(SCRATCH) NAME(ASCRPL)
VOLPARAM VOLSER(AV1000-AV1999) MEDIA(VIRTUAL) REC(VIRTUAL)

```

3. Pour TAPEPLXA, créez les classes de stockage pour les MVC qui contiennent les VTV migrés localement et les classes de stockage CTR.

```

STOR NAME(LOCAL1) ACS(00) MEDIA(STK1R)
STOR NAME(EIPA1) TAPEPLEX(TAPEPLXB)

```

Dans l'exemple ci-dessus, les instructions *STORcLas* définissent :

- la classe de stockage LOCAL1, qui est destinée aux VTV migrés localement à partir de chaque VTSS ;
 - la classe de stockage EIPA1, qui est destinée à CTR et spécifie le TapePlex récepteur (TAPEPLXB).
4. Créez la classe de gestion qui pointe vers les classes de stockage à l'étape 3.

```

MGMT NAME(LOCEEX1) MIGPOL(LOCAL1) EEXPOL(EIPA1)

```

5. Créez une stratégie SMC qui spécifie un média virtuel et affecte la classe de gestion créée à l'étape 4.

```

POLICY NAME(PPAY) MEDIA(VIRTUAL) MGMT(LOCEEX1)

```

6. Créez une instruction *TAPEREQ* pour acheminer les données critiques vers VSM et affecter la stratégie correspondante aux données.

```
TAPEREQ DSN(*.PAYROLL.***) POLICY(PPAY)
```

Dans l'exemple ci-dessus, l'instruction *TAPEREQ* indique d'acheminer les jeux de données avec le masque HLQ *.PAYROLL.*** vers VSM et d'affecter la stratégie PPAY.

Remarque :

- Bien que vous puissiez utiliser des stratégies SMC pour acheminer vos CTR vers un groupe ésoérique spécifique, StorageTek recommande d'utiliser uniquement MGMTCLAS afin que l'allocation SMC/VTCS puisse utiliser n'importe quel VTSS prenant en charge les conditions requises pour MGMTCLAS.
 - Vous pouvez utiliser la commande EEXPORT pour effectuer une CTR manuelle. Pour plus d'informations, reportez-vous au manuel *Référence des commandes, des instructions de contrôle et des utilitaires ELS*.
-

7. Consultez votre membre SYS1.PARMLIB SMFPRMxx pour vérifier que les enregistrements de sous-type 28 sont activés.

Si tel est le cas, VTSS écrit un enregistrement de sous-type 28 incluant le nom VTSS cible pour chaque événement CTR. Vous savez que la fonction CTR a abouti lorsque les VTV arrivent sans encombre sur le VTSS du site récepteur. Existe-t-il un moyen de confirmer cela ? Oui, servez-vous de l'utilitaire *DRMONitr*, comme décrit à l'étape 8.

8. Créez un JCL pour surveiller la CTR.

Pour ce faire, servez-vous de l'utilitaire *DRMONitr* afin de surveiller la CTR. *DRMONitr* provoque la suspension du travail MVS associé jusqu'à ce que la CTR aboutisse. Par exemple :

```
//MONITOR EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
/* If HSC IS NOT OR MAY NOT BE ACTIVE, INCLUDE THE
/* FOLLOWING:
//SLSCNTL DD DSN=primary.cds.name, DISP=SHR
//SLSCNTL2 DD DSN=secondary.cds.name, DISP=SHR
//SLSSTBY DD DSN=standby.cds.name, DISP=SHR
//SLSPARMP DD DSN=h1q.PARMLIB(BKPCNTL), DISP=SHR
//SLSPARMS DD DSN=h1q.PARMLIB(BKPCNTL2), DISP=SHR
//SLSPARMB DD DSN=h1q.PARMLIB(BKPSTBY), DISP=SHR
//SYSIN DD UNIT=SYSDA, SPACE=(TRK,1)
/* THE FOLLOWING IS USED BY THE SNAPSHOT UTILITY:
//SYSPRINT DD SYSOUT=*
//SLSPRINT DD SYSOUT=*
//SLSPRINT DD SYSOUT=*
DRMON MGMT(LOCEEX1) STOR(EIPA1) MAXAGE(24) TIMEOUT(120)
```

Dans cet exemple, l'utilitaire *DRMON* surveille les migrations pour la classe de stockage EEPA1, qui est la classe de stockage CTR spécifiée par la classe de gestion LOCEEX1 à l'étape 4. En outre, surveillez uniquement les VTV qui ont été mis à jour au cours des dernières 24 heures et spécifiez le délai d'expiration de *DRMON* sur 120 minutes.

Stratégies pour le TapePlex récepteur

Pour définir des stratégies pour le TapePlex récepteur (TAPEPLXB) de l'exemple de système CTR présenté dans la section " [Configuration pour la fonction CTR](#)", procédez comme suit :

1. Créez des définitions MVC POOLPARAM/VOLPARAM pour le pool MVC défini pour le TapePlex TAPEPLXB, afin qu'elles contiennent les VTV CTR provenant de TAPEPLX :

```
POOLPARAM TYPE(MVC) NAME(AMVCCTR) INITMVC(YES) MCVFREE(25) -
MAXMVC(98) THRESH(85) START(98)
VOLPARAM VOLSER(BM1000-BM1099) MEDIA(STK1R)
```

Remarque :

StorageTek recommande vivement d'utiliser la fonctionnalité POOLPARAM/VOLPARAM pour s'assurer que des plages de volumes sont réservées aux volumes répliqués par la CTR sur le site distant.

2. Créez un pool VTV externe pour les VTV exportés à partir de TAPEPLXA :

```
POOLPARAM TYPE(EXTERNAL) NAME(AEXTBPL) OWNRPLEX(TAPEPLXA)
VOLPARAM VOLSER(AV1000-AV1999) MEDIA(VIRTUAL) REC(VIRTUAL)
```

Remarque :

Cet exemple ne définit pas de pools pour le travail de production sur TAPEPLXB, mais uniquement les pools utilisés par TAPEPLXA. Si le travail de production doit être exécuté sur TAPEPLXB, des définitions POOLPARAM et VOLPARAM supplémentaires sont nécessaires pour les pools provisoires et MVC du travail TAPEPLXB.

3. Créez un pool provisoire VTV que le TapePlex TAPEPLXB utilisera pour le travail TAPEPLXA :

```
POOLPARAM TYPE(SCRATCH) NAME(ASCRPL)
VOLPARAM VOLSER(BV1000-BV1999) MEDIA(VIRTUAL) REC(VIRTUAL)
```

4. Créez un pool MVC pour que le TapePlex TAPEPLXB y stocke les VTV provenant du test DR de TAPEPLXA ou de la production (dans un sinistre) :

```
POOLPARAM TYPE(MVC) NAME(AMVCDR) INITMVC(YES) MCVFREE(25) -
MAXMVC(98) THRESH(85) START(98)
```

```
VOLPARM VOLSER(BM2000-BM2099) MEDIA(STK1R)
```

5. Pour TAPEPLXB, créez les classes de stockage pour la migration locale.

```
STOR NAME(TPEPLXA1) MVCPPOOL(AMVCTR)  
STOR NAME(TPEPLXA2) MVCPPOOL(AMVCDR)
```

Dans cet exemple, les instructions STORclas définissent les classes de stockage TPEPLXA1 et TPEPLXA2 pour la migration locale. Les noms de classe de stockage permettent de séparer ce travail du travail local TAPEPLXB.

6. Créez les classes de gestion qui pointent vers les classes de stockage à l'étape 5.

```
MGMT NAME(LOCEEX1) MIGPOL(TPEPLXA1)  
MGMT NAME(LOCPLXA) MIGPOL(TPEPLXA2)
```

Notez que le nom LOCEEX1 correspond au nom de classe de gestion qui est utilisé sur TAPEPLXA (cette classe de gestion est indiquée dans les métadonnées VTV qui sont envoyées à partir du VTSS sur TAPEPLXA), mais l'exemple fait référence à la classe de stockage pour la migration locale. Les définitions des classes de gestion et de stockage sur TAPEPLXB peuvent utiliser n'importe quel paramètre, y compris EEXPOL, pour effectuer la réplication sur un troisième TapePlex. En outre, créez une autre classe de gestion MGMTCLAS, LOCPLXA, qui sera utilisée pour la migration lors d'un test DR de la charge globale de TAPEPLXA.

Utilisation de la fonction CTR en l'absence de partitions LPAR sur le site distant

Dans certains environnements, seul un site dispose de partitions LPAR utilisant des bandes, alors qu'un deuxième site contient uniquement une bibliothèque et un matériel VTSS mais aucune partition LPAR MVS. Il est possible de configurer cet environnement afin d'utiliser la fonction CTR en tant que mécanisme de récupération après sinistre et de test DR.

Pour ce faire, vous devez :

1. Lancer la fonctionnalité client/serveur de SMC sur votre environnement de production, afin de disposer d'au moins une partition LPAR de production n'exécutant pas HSC/VTCS.

Vous pouvez également exécuter le TapePlex DR sur la même partition LPAR que le TapePlex de production à l'aide de la fonctionnalité de mode MULT. Pour plus d'informations sur l'utilisation de cette fonctionnalité, voir *Configuration du HSC et du VTCS*.

Dans cet exemple, le TapePlex de production est TAPEPLXA.

2. Créez un nouveau CDS définissant le matériel (bibliothèques et VTSS) sur le site distant.
3. Démarrez un HSC/VTCS à l'aide du nouveau CDS sur une partition LPAR (MVSX) qui n'exécute pas actuellement de HSC/VTCS de production, ou sur la partition LPAR où vous avez décidé d'exécuter plusieurs copies de HSC/VTCS à l'aide de la fonctionnalité de mode MULT.

Remarque :

Pour des raisons de fiabilité, il est conseillé d'exécuter deux instances de HSC/VTCS pointant vers TAPEPLXB sur deux partitions LPAR différentes, de sorte que si une instance n'est pas disponible, les métadonnées pour les VTV répliqués par la CRT peuvent être envoyées à la deuxième instance.

Ce système est le TapePlex TAPEPLXB.

4. Définissez les paramètres du système SMC sur MVSX en configurant les TapePlex TAPEPLXA et TAPEPLXB.

Chaque système SMC du complexe doit définir les deux TapePlex TAPEPLXA (le TapePlex de production) et TAPEPLXB (le TapePlex DR). Pour que la réplication de VTV lors d'un test DR continue à être prise en charge, vous devez définir un serveur pour le TapePlex TAPEPLXB qui pointe vers l'hôte sur le site distant. Par exemple :

```
TAPEPLEX NAME(TAPEPLXA) LOCSUB(HSCA)
TAPEPLEX NAME(TAPEPLXB)
SERVER NAME(TPLXBPR) TAPEPLEX(TAPEPLXB) HOST(MVSX) PORT(999)
SERVER NAME(TPLXBDR) TAPEPLEX(TAPEPLXB) HOST(MVSXDR) PORT(1234)
```

Remarque :

Cet exemple suppose que les deux sites portent des noms d'hôte TCP/IP uniques, bien que les noms des partitions LPAR (MVSX) puissent être identiques sur le site de production et le site DR.

5. Définissez vos stratégies VTCS sur TAPEPLXA pour autoriser la CTR sur TAPEPLXB.

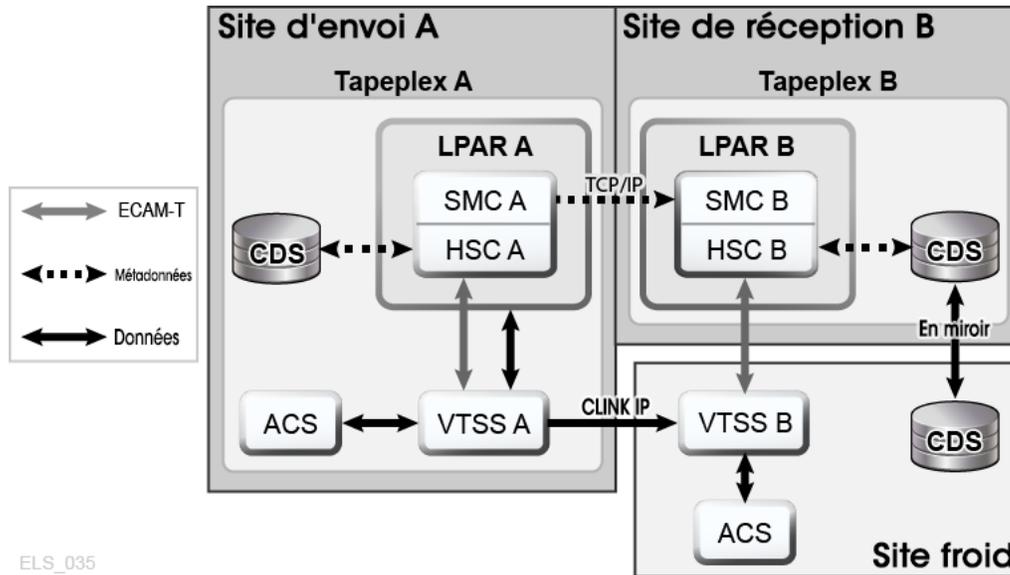
Voir "[Définition de stratégies pour CTR](#)".

6. A l'aide de votre solution de réplication de disque, conservez une copie du contenu du CDS pour TAPEPLXB sur votre emplacement distant.

S'il existe une connectivité fiable, vous voudrez peut-être également conserver les copies principales (et d'autres) du CDS HSC sur le site DR, en utilisant des connexions FICON pour accéder au CDS à partir du site de production.

La [Figure 4.4, « Copie du CDS - sans partitions LPAR sur le site distant »](#) présente une copie du CDS sans partitions LPAR sur le site distant.

Figure 4.4. Copie du CDS - sans partitions LPAR sur le site distant



ELS_035

Utilisation de la fonction CTR comme solution de récupération après sinistre

Une solution de récupération après sinistre vous permet d'effectuer trois opérations :

- configurer et démarrer la solution comme décrit à la section " [Configuration et démarrage de CTR](#)" ;
- utiliser la solution en cas de sinistre pour poursuivre l'activité sur le site distant, comme décrit dans la section " [Utilisation de la fonction CTR pour la continuité de l'activité](#)" ;
- utiliser la solution pour reprendre l'activité sur le site local une fois qu'il est de nouveau opérationnel, comme décrit dans la section " [Utilisation de la fonction CTR pour la reprise de l'activité](#)".

Utilisation de la fonction CTR pour la continuité de l'activité

Si le site TAPEPLXA est victime d'une panne, vous pouvez poursuivre votre activité sur le site TAPEPLXB en exécutant simplement la charge globale à l'aide du TapePlex de TAPEPLXB. Pour protéger les données, les VTV qui ont été répliqués à partir de TAPEPLXA restent à l'état de lecture seule (voir " [Considérations relatives aux VTV CTR en lecture seule](#) "). Toutefois, une fois que vous aurez rétabli la charge globale de TAPEPLXA, vous voudrez peut-être supprimer des VTV qui ont été répliqués à partir de TAPEPLXA. Vous devez être prudent avant d'effectuer cette étape et vérifier que votre travail de production pour les deux TapePlex h TAPEPLXA et TAPEPLXB est stable. En outre, vous voudrez probablement recréer ultérieurement un environnement TapePlex distinct pour TAPEPLXA en vue de restaurer votre configuration d'origine.

Pour utiliser la fonction CTR pour la continuité de l'activité :

1. Modifiez les définitions *POOLPARAM/VOLPARAM* dans le CDS du TapePlex TAPEPLXB pour le pool nommé (*AEXTBPL*) de *TYPE(EXTERNAL)* en *TYPE(SCRATCH)* :

```
POOLPARAM TYPE(SCRATCH) NAME(AEXTBPL)
VOLPARAM VOLSER(AV1000-AV1999) MEDIA(VIRTUAL)
```

Notez que la plage *VOLPARAM VOLSER* demeure inchangée.

2. Vous pouvez désormais exécuter un travail de synchronisation des volumes provisoires sous TAPEPLXB pour supprimer les VTV situés dans la plage AV1000-AV1999 en vous basant sur le statut provisoire indiqué dans TMS, soit pour revenir à un point de reprise, soit pour traiter normalement une mise à jour provisoire.

Vous voudrez peut-être qu'un certain délai s'écoule entre le début du traitement de la production et la réutilisation des numéros de série de volume VTV figurant dans la plage AV1000-AV1999 en tant que volumes provisoires. L'utilisation de la fonctionnalité *POOLPARAM/VOLPARAM* empêche ces volumes d'être sélectionnés en tant que volumes provisoires sauf si une stratégie demande spécifiquement *SUBPOOL (AEXTBPL)*.

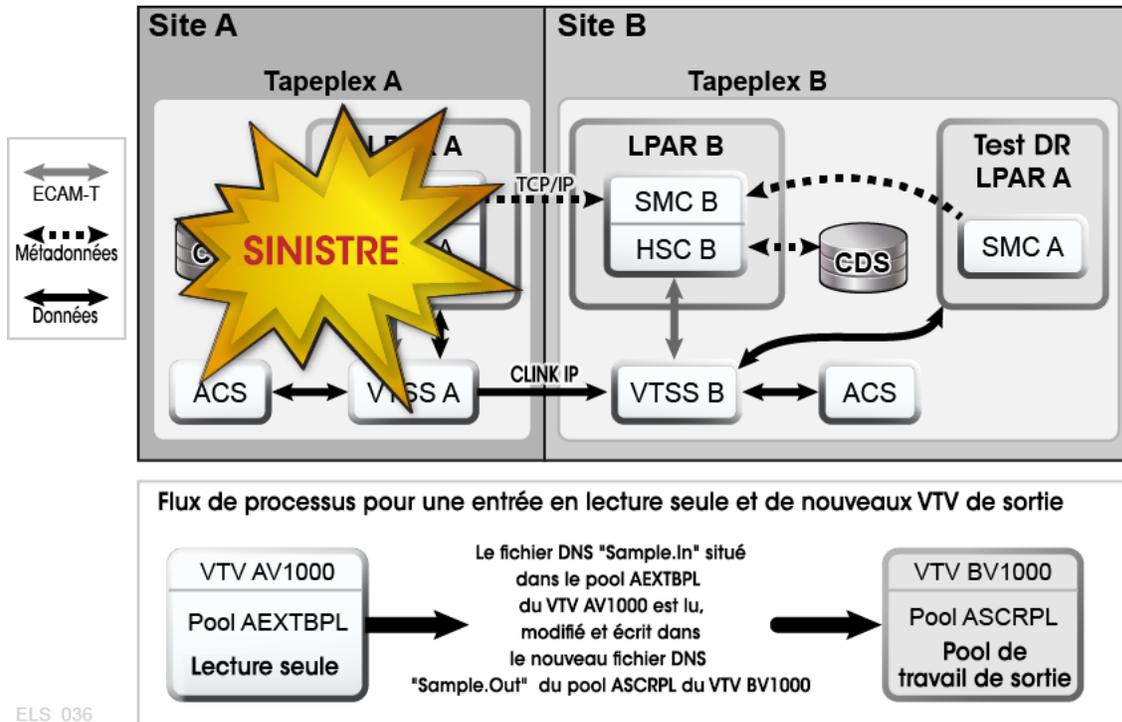
Au cours de cette période, vous utiliserez les volumes du sous-pool provisoire de TAPEPLXB ASCRPL (plage volser BV1000-BV1999) pour le travail de production de TAPEPLXA.

Une fois que votre environnement de récupération après sinistre est stabilisé, vous pouvez modifier à nouveau vos définitions *POOLPARAM/VOLPARAM* pour sélectionner des volumes provisoires dans la plage AV1000-AV1999 :

```
POOLPARAM TYPE(SCRATCH) NAME(ASCRPL)
VOLPARAM VOLSER(AV1000-AV1999) MEDIA(VIRTUAL)
VOLPARAM VOLSER(BV1000-BV1999) MEDIA(VIRTUAL)
```

La [Figure 4.5, « Système lors de la continuité de l'activité »](#) présente un exemple de CTR utilisant la continuité de l'activité.

Figure 4.5. Système lors de la continuité de l'activité



Utilisation de la fonction CTR pour la reprise de l'activité

Le site local a subi une panne, mais vous avez continué votre activité sur le site distant. Maintenant que le site local est à nouveau opérationnel, comment faire pour reprendre votre activité dessus ? La reprise de votre activité dépend essentiellement des événements qui se sont produits pendant et après la panne. Supposons que toutes les données locales d'origine aient été perdues et que vous disposiez d'un tout nouveau VTSS vide sur le site local.

Pour reprendre l'activité après la perte de toutes les données sur le site local :

1. Créez un nouveau CDS et exécutez un audit HSC pour déterminer le contenu des bibliothèques physiques.

Vous devez ensuite effectuer une "réplication inverse" des données et des métadonnées sur le site local à partir du site distant.

2. Configurez votre console CONFIG pour le site distant afin qu'elle puisse envoyer des données vers le site local.
3. Effectuez une réplication inverse à l'aide de EEXPORT.

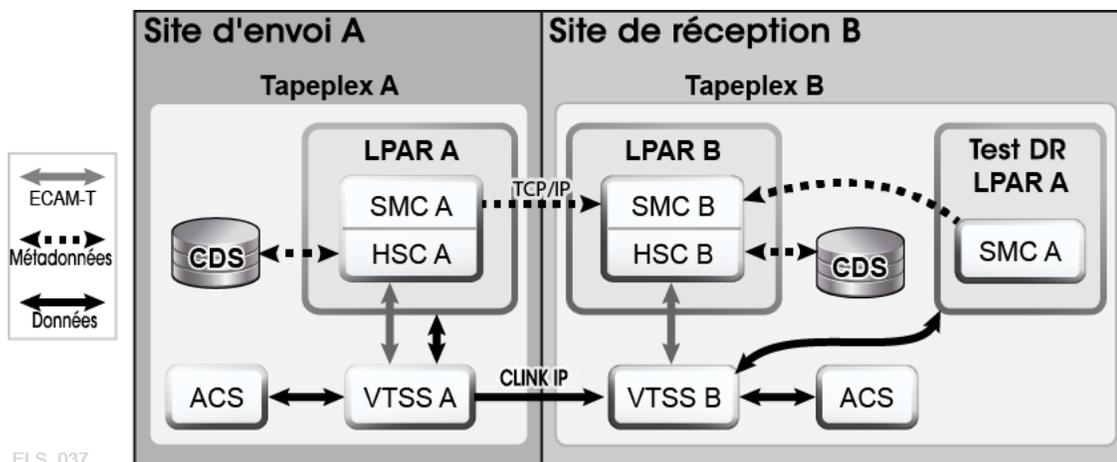
Par exemple :

```
EEXPORT MGMTCLAS(LOCEEX1, LOCEEX2) TOPLEX(TAPEPLXA)
```

Tests de récupération après sinistre à l'aide de la fonction CTR

Pour donner un exemple, prenons deux sites, TAPEPLXA et TAPEPLXB, chacun défini comme son propre TapePlex (CDS HSC). Vous avez utilisé la fonction CTR pour répliquer vos VTV critiques de TAPEPLXA vers TAPEPLXB. La [Figure 4.6, « Tests de récupération après sinistre à l'aide de la fonction CTR »](#) présente des tests DR utilisant la fonction CTR.

Figure 4.6. Tests de récupération après sinistre à l'aide de la fonction CTR



Pour effectuer un test DR sur TAPEPLXB pour le travail de TAPEPLXA, il est conseillé d'appliquer la procédure suivante :

1. Vérifiez que le CDS de TAPEPLXB contient un ou plusieurs sous-pools provisoires pour les données de sortie de TAPEPLXA, qui sont distincts des sous-pools provisoires utilisés pour le travail de TAPEPLXB.

Pour obtenir des exemples, voir "[Stratégies pour le TapePlex récepteur.](#)"

2. Vérifiez que les données de catalogue et de gestion de bande de TAPEPLXA sont disponibles.
3. Affichez SMC sur la partition LPAR de test de TAPEPLXA, en définissant son TapePlex en tant que TAPEPLXB et en indiquant des commandes *SERVER* pour un ou plusieurs hôtes HSC dans TAPEPLXB.
4. Commencez l'exécution de votre charge globale de test.

SMC accèdera automatiquement aux VTV antérieurs ou postérieurs au début du test, car ils continuent à être répliqués à partir de TAPEPLXA. Vérifiez que les VTV qui seront utilisés par votre test DR ne sont pas supprimés ou modifiés par le TapePlex TAPEPLXA.

5. Une fois le test terminé, supprimez tous les VTV dans le ou les sous-pools de test DR utilisés par le test.

Dans cette approche, aucun CDS spécial n'est requis et aucune règle spécifique n'est nécessaire pour vérifier que deux systèmes HSC distincts peuvent partager des ressources matérielles. Toutefois, cette méthode requiert que le test DR soit exécuté à l'aide des données actuelles, ou au moins à l'aide des données actuellement disponibles. Notez que la sortie de votre test DR et les VTV répliqués à partir de TAPEPLXA utiliseront l'espace tampon VTSS de TAPEPLXB.

Les données répliquées à partir du TapePlex TAPEPLXA étant en lecture seule, toute tentative de modification des données par le test DR générera le message SMC0247 "Mount failed for write-protected VTV vvvvvv on drive dddd from SMC indicating that the VTV cannot be mounted" (échec du montage du VTV protégé en écriture vvvvvv sur l'unité dddd à partir de SMC indiquant que le VTV ne peut pas être monté). L'affichage de ce message peut indiquer que votre processus DR n'a pas clairement défini les points de reprise de l'application (voir "[Considérations relatives aux VTV CTR en lecture seule](#)"). Si tel est le cas, l'utilisation de CTR pour votre stratégie DR risque de ne pas être un choix judicieux.

Remarque :

Si vous utilisez la fonction CRT pour créer des copies de vos VTV sur un site distant, il est conseillé de ne pas utiliser CDRT pour vos tests DR, car l'utilisation de CDRT ne permet la mise à jour des VTV en lecture seule, même dans un environnement CDRT distinct.

Tests DR en l'absence de partitions LPAR sur le site DR

Lorsque vous gérez le matériel CTR sur le site DR à l'aide d'un TapePlex en cours d'exécution sur le site de production, il convient de prendre en compte quelques points supplémentaires pour un test DR. Cet exemple utilise TAPEPLXA pour le TapePlex de production et TAPEPLXB pour le TapePlex qui s'exécute normalement sur le site de production, mais s'exécute sur le site DR lors d'un test DR.

1. Vous devez arrêter le TapePlex de récupération après sinistre TAPEPLXB sur le site de production avant le test.

Lors du test DR, TAPEPLXB s'exécutera sur le site DR sur une copie du CDS TAPEPLXB.

2. Les VTV de production continueront d'être envoyés vers TAPEPLXB et représentés dans le CDS sur le site DR.

Au cours de cette période, le CDS TAPEPLXB sur le site de production devient obsolète, car il ne représente plus les VTV répliqués lors du test DR. Les instructions TAPEPLEX et SERVER sur les partitions LPAR de production permettent de s'assurer que la réplication de données continue lors du test DR :

```
TAPEPLEX NAME(TAPEPLXB)
SERVER NAME(TPLXBPR) TAPEPLEX(TAPEPLXB) HOSTNAME(MVSX) PORT(999)
```

```
SERVER NAME(TPLXBDR) TAPEPLEX(TAPEPLXB) HOST(MVSXDR) PORT(1234)
```

3. Lorsque vous démarrez le HSC/VTCS pour le TapePlex TAPEPLXB sur le site DR, vous devez veiller à démarrer le serveur HTTP sur SMC :

```
HTTP START PORT(1234)
```

Le numéro de port (1234) correspond à celui qui a été défini dans l'instruction *TAPEPLXBDR SERVER*.

4. A la fin du test, supprimez tous les VTV qui ont été créés par le test.

Vous n'avez pas besoin de déterminer quels VTV ont été réellement créés par le test : il vous suffit de supprimer tous les volumes du ou des sous-pools. Par exemple :

```
SCRATCH VOL(BV1000-BV2999)
```

5. Arrêtez le HSC/VTCS pour le TapePlex TAPEPLXB sur le site DR.
6. Vous devez maintenant vérifier que le CDS TAPEPLXB sur le site DR est renvoyé vers le site de production.

Idéalement, cette opération peut être réalisée en remettant en miroir le CDS TAPEPLXB avec le site de production lors du test DR. Si cela n'est pas possible, vous pouvez utiliser FTP ou un autre mécanisme de votre choix pour recopier vers le site de production le CDS provenant de la version actuelle qui se trouve sur le site DR.

7. Redémarrez TAPEPLXB sur les partitions LPAR du site de production.

Puisqu'il n'existe aucune copie active de TAPEPLXB, les VTV planifiés pour la fonction CTR sur TAPEPLXB resteront dans le tampon VTSS. Lorsque TAPEPLXB sera à nouveau actif sur le site de production, ces VTV seront répliqués sur le VTSS du site DR.

Gestion des VTV répliqués à l'aide de CTR (Cross-TapePlex Replication)

Vous pouvez utiliser *VTVMaint* pour modifier le statut des VTV répliqués par CTR en procédant comme suit :

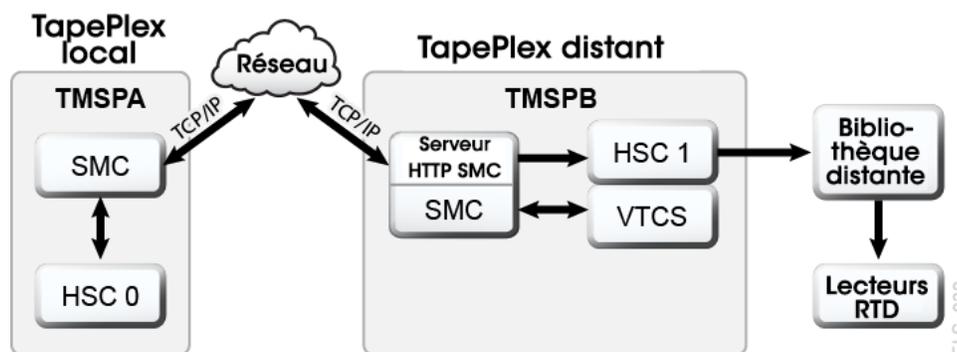
- Utilisez *VTVMaint DELEXPot* pour supprimer le nom d'un TapePlex faisant référence à un VTV. Par exemple, si vous répliquez un VTV de TAPEPLXA vers TAPEPLXB, puis que vous supprimez la copie située sur TAPEPLXA, vous pouvez utiliser *VTVMaint DELEXPot* pour supprimer la référence de TAPEPLXA au VTV.
- Utilisez *VTVMaint ADDEXPot* pour ajouter le nom d'un TapePlex faisant référence à un VTV, comme décrit à la section "[Utilisation de la fonction CTR pour la continuité de l'activité](#)".
- Servez-vous de l'utilitaire *VTVMaint* pour modifier la propriété d'un VTV reçu via CTR. Cependant, notez que le VTV doit avoir actuellement le statut provisoire. Par

exemple, *VTVMaint Ownrplex(TapePlxB)* modifiera la propriété d'un VTV envoyé par TAPEPLXA pour qu'il appartienne au TapePlex où il se trouve actuellement.

Chapitre 5. Configuration d'une bibliothèque distante

La Figure 5.1, « RTD fonctionnant sous un TapePlex distant » présente un exemple de configuration avec une bibliothèque distante. Les sections suivantes montrent comment configurer cet exemple.

Figure 5.1. RTD fonctionnant sous un TapePlex distant



Modification du fichier SCMDS du SMC

Le composant SMC gère toutes les communications entre VTCS et un TapePlex distant ; SMC doit donc savoir comment se connecter au TapePlex distant. Pour ce faire, définissez un TapePlex local et un TapePlex distant, plus une ou plusieurs instructions *SMC SERVER* qui déterminent les chemins de contrôle TCP/IP vers le TapePlex distant. Vous voudrez peut-être effectuer cette opération dans le fichier *CMDS* du *SMC*, comme indiqué dans l'exemple suivant :

```
TAPEPLEX NAME(TMVSA)LOCSUB(HSC0)
TAPEPLEX NAME(TMVSB)LOCSUB(HSC1)
SERVER NAME(TMVSB)IP(192.168.1.10)PORT(60000)
```

Cet exemple contient :

- une instruction *TAPEPLEX*, qui définit un TapePlex local, TMVSA, avec un HSC exécuté sur l'hôte MVS local (HSC0) ;
- une deuxième instruction *TAPEPLEX*, qui définit un TapePlex distant, TMVSB, avec un HSC exécuté sur l'hôte MVS distant (HSC1) ;

- une commande *SERVER*, qui définit un chemin de communication UUI vers TMVSB, où :
 - le nom du serveur distant est TMVSB ;
 - la valeur du paramètre *IP*. est l'adresse IP du port ELS; à savoir 192.168.1.10 pour les communications UUI ;
 - la valeur du paramètre *PORT* est 60000 ; cette valeur est toujours utilisée pour le paramètre *SERVER PORT* pour les communications SMC avec un TMVSB.

Mise à jour de la console CONFIG du VTCS pour définir une bibliothèque distante

Vous devez mettre à jour la console *CONFIG* du VTCS pour définir la bibliothèque distante et la connectivité vers celle-ci à partir du VTSS. Comme indiqué dans l'exemple suivant, la bibliothèque distante est définie par une instruction *CONFIG STORMNGR*.

```
TAPEPLEX THISPLEX=TMVSA
STORMNGR NAME=TMVSB
VTSS NAME=VTSS1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTDPATH NAME=RM1RTD1 STORMNGR=TMVSB DEVNO=1A00 CHANIF=0A:0
RTDPATH NAME=RM1RTD2 STORMNGR=TMVSB DEVNO=1A01 CHANIF=0A:1
RTDPATH NAME=RM1RTD3 STORMNGR=TMVSB DEVNO=1I00 CHANIF=0I:0
RTDPATH NAME=RM1RTD4 STORMNGR=TMVSB DEVNO=1I01 CHANIF=0I:1
VTD LOW=6900 HIGH=69FF
```

Dans cet exemple, notez les points suivants :

- L'instruction *CONFIG TAPEPLEX* spécifie TMVSA comme TapePlex local.
- Une instruction *STORMNGR* spécifie TMVSB comme bibliothèque distante.
- Les instructions *CONFIG RTDPATH* pour VTSS1, qui spécifient :
 - le nom du chemin *RTDPATH*,
 - la connexion à la bibliothèque distante (*STORMNGR=TMVSB*),
 - le numéro du périphérique (*DEVNO*),
 - la valeur *CHANIF* de chaque connexion VTSS au RTD au format *ci:p*, où :
 - › *c* représente 0 ou 1 ;
 - › *i* représente A ou I ;
 - › *p* représente un chiffre entre 0 et 3.

Remarque :

Pour les VSM5, la valeur *CHANIF* doit correspondre aux valeurs spécifiées dans l'écran IFF Configuration Status Screen du VSM5. Pour les VSM 6, la valeur doit être unique pour chaque VTSS mais ne correspond pas à une valeur réelle des ports TCP/IP du VSM 6.

Vous pouvez maintenant utiliser le paramètre *STORclas STORMNGR* pour acheminer les données vers la bibliothèque distante. Par exemple :

```
STOR NAME(REMLIB) STORMNGR(TMVSB)
```

Considérations relatives au pool de MVC

Une MVC située dans une bibliothèque distante doit être incluse dans les définitions de pool MVC local, de préférence via des définitions *VOLPARM/POOLPARM*. Le HSC local ne traite pas les MVC distantes comme des bibliothèques.

Un serveur de bibliothèque distante a généralement besoin de définir les MVC comme des éléments "gérés en externe" par une méthode. Le partage d'un pool de MVC locales par un TapePlex distant n'est pas pris en charge car il n'existe aucune sérialisation.

Si un pool de MVC est partagé à des fins de transfert de données, un seul TapePlex peut écrire activement sur les MVC. Les autres TapePlex ne disposent que d'un accès en lecture seule. La sérialisation des MVC sur les unités incombe à l'utilisateur.

Chapitre 6. Utilisation de configurations VTSS en cluster

Les VTSS en cluster vous permettent de copier des VTV entre deux VTSS. Un VTSS en cluster est un outil puissant pour des applications telles que les solutions de récupération après sinistre (DR), mais pas seulement. Les sections suivantes décrivent les principes fondamentaux des clusters VTSS, ainsi que leur fonctionnement

- " Définition d'un VTSS en cluster "
- " Conditions requises pour les VTSS en cluster "
- " Fonctionnement des configurations VTSS en cluster "

Voici également des variantes et fonctionnalités supplémentaires de VTSS en cluster que vous devez connaître :

- " Clusters unidirectionnels et bidirectionnels "
- " Clustering étendu "
- " Réplication synchrone ou asynchrone "
- " Clustering avec connexions TCP/IP "

Définition d'un VTSS en cluster

Un cluster VTSS est une solution de haute disponibilité (HA, High Availability) qui permet d'optimiser la disponibilité des données. Elle se compose de plusieurs systèmes VTSS connectés à l'aide de liens de communication (CLINK) FICON ou TCP/IP. En outre, chaque système VTSS au sein du cluster peut accéder à l'ensemble des données créées dans le cluster (VTSS résident ou migré). Les données (VTV) créées sur un cluster sont répliquées entre deux systèmes VTSS situés au sein du même cluster sous le contrôle de stratégies VTCS.

Remarque :

Pour s'assurer que chaque système VTSS peut accéder à toutes les données créées au sein du cluster, les configurations en cluster peuvent se présenter comme suit :

- chaque VTSS figurant dans le cluster est connecté à des RTD ou VLE ;
 - "sans bandes" - aucun VTSS n'est connecté à des VLE ou RTD.
-

Les configurations en cluster offrent donc la meilleure disponibilité de données avec une récupération à chaud si un VTSS situé dans un cluster est victime d'une panne (les données

répliquées restent disponibles sans qu'il soit nécessaire de recourir à des rappels à partir de MVC).

Avant VTCS 7.0, un cluster ne pouvait être composé que de deux VTSS. Depuis VTCS 7.0, un cluster peut comprendre un grand nombre de VTSS. Toutefois, un VTV peut uniquement se trouver dans deux VTSS à un moment donné.

Un cluster peut s'étendre sur plusieurs emplacements géographiques. Cependant, un cluster **doit se trouver au sein d'un TapePlex unique** (contrôlé par un seul CDS).

Un VTV peut être répliqué (copié) entre deux VTSS :

- De façon asynchrone par rapport à la création de VTV. Exécution programmée dès que possible après le démontage de VTV.
- De façon synchrone par rapport à la création de VTV. Le démontage de VTV ne s'achèvera pas avant la fin de la réplication.

Remarque :

Pour la réplication synchrone, via des CLINK ou la réplication améliorée, le VTSS met fin à la réplication si celle-ci entraîne le dépassement du gestionnaire MIH (Missing Interrupt Handler).

Pour VSM 6 et VSM 7, la valeur de dépassement MIH peut être configurée de 20 à 45 minutes par incréments de 5 minutes. Pour VSM4 et VSM 5, la valeur de dépassement MIH peut être configurée à 20 ou 45 minutes. La valeur définie dans le VTSS doit correspondre à celle définie dans le mainframe.

Les connexions entre les VTSS d'un cluster peuvent être unidirectionnelles, lorsque les données (VTV) circulent dans un seul sens, ou bidirectionnelles, lorsque les données (VTV) peuvent circuler dans les deux sens. L'utilitaire CONFIG indique si un cluster est unidirectionnel ou bidirectionnel, tandis qu'une classe de gestion de VTV détermine sa stratégie de réplication, le cas échéant, et si la réplication s'effectue de façon synchrone ou asynchrone.

Ainsi, l'archivage sécurisé de MVC (tel que décrit à la section "[Utilisation de la fonction d'archivage sécurisé externe d'ELS](#)") et la réplication VTV peuvent tous deux faciliter la mise en place d'une solution de continuité de l'activité/récupération après sinistre. Toutefois, la réplication VTV présente plus d'avantages qu'une solution de haute disponibilité car :

- les données peuvent être sauvegardées de façon synchrone ;
- les données récentes, qui ont été répliquées sur un VTSS "de récupération", peuvent être restaurées plus rapidement, car vous n'avez pas besoin de monter des MVC.

Conditions requises pour les VTSS en cluster

Le [Tableau 6.1, « Conditions requises pour les VTSS en cluster »](#) présente les conditions requises pour les VTSS en cluster.

Tableau 6.1. Conditions requises pour les VTSS en cluster

| Composant | Conditions requises |
|--|---|
| Clusters étendus | Microcode VTSS D02.07.00.00 ou niveau ultérieur pour les VSM4 ou VSM5 avec connexions FICON uniquement. Pour VSM 6 et VSM 7, tous les niveaux de microcode. |
| 2 VTSS dans un cluster (interfaces ESCON) | Les VTSS principaux et secondaires peuvent représenter n'importe quelle combinaison de VSM4, dans laquelle la capacité du VTSS secondaire n'a pas d'importance. Les VSM5 ne possèdent pas d'interfaces ESCON et ne peuvent pas figurer dans un cluster avec d'autres VTSS utilisant ESCON. |
| 2 VTSS dans un cluster (interfaces FICON) | Les VTSS principaux et secondaires peuvent représenter n'importe quelle combinaison de VSM4 et VSM5, dans laquelle la capacité du VTSS secondaire n'a pas d'importance. Par exemple, toutes les combinaisons suivantes sont admises : <ul style="list-style-type: none"> • VSM5 principal, VSM4 secondaire • VSM5 principal, VSM5 secondaire • VSM4 principal, VSM4 secondaire • VSM4 principal, VSM5 secondaire (non recommandé) |
| Microcode des VTSS principaux et secondaires | Le microcode du VTSS principal doit être à un niveau qui prend en charge l'envoi de VTV répliqués. Le microcode du VTSS secondaire doit être à un niveau qui prend en charge l'envoi de VTV répliqués et l'utilisation du VTSS secondaire comme VTSS de production. Une fois le microcode installé, la fonction de clustering doit être activée à la fois sur les VTSS principaux et secondaires à l'aide d'une disquette d'options. Pour plus de détails, consultez le représentant du service technique StorageTek. |
| VTD réservés au clustering | Dans les configurations VTSS en cluster, vous devez vérifier que les 16 premiers VTD de chaque VTSS (0-F) sont réservés au clustering. Ces périphériques doivent être HORS LIGNE sur MVS et leurs chemins doivent être en ligne sur chaque hôte serveur HSC. Il en va de même pour les VTSS entrant en jeu dans la fonction CTR (Cross TapePlex Replication). VTCS n'inscrit pas les 16 premiers VTD auprès de SMC/HSC, ce qui empêche le montage de VTV sur ces VTD. |
| RTD | <p>Dans les environnements à deux ACS, les mêmes types de périphérique doivent être représentés dans les RTD connectés à chaque ACS de sorte que les données migrées par un VTSS d'un cluster puissent être rappelées par n'importe quel autre VTSS du cluster. Le nombre de MVC, le type de média et l'emplacement utilisés pour la migration sont déterminés par le paramètre <i>MIGPOL</i> de l'instruction MGMTclas.</p> <p>Tous les ACS doivent être pris en compte. En outre, si un type d'unité dans un ACS est connecté à l'un des VTSS d'un environnement VTSS en cluster, une unité de même type et située dans le même ACS doit être connectée à tous les autres VTSS de cet environnement en cluster.</p> |
| Protocole IP natif (clustering avec TCP/IP) | Le protocole IP natif requiert CDSLEVEL F ou niveau ultérieur, avec les PTF suivants : <ul style="list-style-type: none"> • Pour 6.2 : <ul style="list-style-type: none"> L1A00P7 - SMC6200 L1H14IM - SMS6200 L1H14O2 - SOS6200 |

| Composant | Conditions requises |
|-----------|---|
| | L1H14IL - SWS6200 |
| | <ul style="list-style-type: none"> • Pour 7.0, L1H150G (SES7000) • Pour 7.1 et versions ultérieures, la prise en charge est incluse dans la base. |
| | Les connexions suivantes sont prises en charge pour le protocole IP natif : |
| | <ul style="list-style-type: none"> • VSM5 à VSM5 • VSM5 à VSM 6 • VSM 6 à VSM 6 • VSM 6 ou VSM5 à VLE |

La réplication synchrone, qui s'applique uniquement à VSM4 et supérieur, répond aux exigences décrites dans le [Tableau 6.2, « Conditions requises pour la réplication synchrone »](#).

Tableau 6.2. Conditions requises pour la réplication synchrone

| Condition requise pour la réplication synchrone | Niveau de microcode VTSS | Niveau de CDS |
|---|---|-------------------------|
| Ports FICON ou IP natifs pour les CLINK | Microcode niveau D02.03.00.00 ou suivant pour VSM4 et VSM5. Pour VSM 6 et VSM 7, tous les niveaux de microcode. | "F" ou niveau ultérieur |

La réplication améliorée, qui s'applique à VSM6 et supérieur, présente les conditions requises décrites dans le [Tableau 6.3, « Conditions requises pour la réplication améliorée »](#).

Tableau 6.3. Conditions requises pour la réplication améliorée

| Condition requise pour la réplication améliorée | Niveau de microcode VTSS | Niveau de CDS |
|---|--|-------------------------|
| Ports FICON ou IP natifs pour les CLINK | VSM 6.3.0.03.000 (6.3.0.05.000 recommandé) pour VSM 6 ; VSM 7.0.05.000 pour VSM 7. Appliquer PTF L1H18MB. | "H" ou niveau ultérieur |

Fonctionnement des configurations VTSS en cluster

Vous pouvez utiliser VSM pour connecter deux VTSS par des liens de cluster (CLINK) afin de générer une *configuration VTSS en cluster*. Utilisez les instructions ci-dessous pour implémenter une configuration en cluster.

- Les clusters peuvent être unidirectionnels ou bidirectionnels en fonction des instructions *CLINK*.
- Le VTSS secondaire (ou deuxième homologue) peut se trouver sur le même emplacement physique que le VTSS principal (ou premier homologue), ou sur un emplacement distant.

- L'instruction *CONFIG CLUSTER* spécifie les VTSS qui forment le cluster.
- L'instruction *CONFIG CLINK* définit les CLINK qui connectent les VTSS. Le mode d'écriture des instructions *CLINK* détermine si la réplication est unidirectionnelle ou bidirectionnelle. Pour obtenir des exemples, consultez les sections " [VTSS en cluster unidirectionnel](#) " et " [VTSS en cluster bidirectionnel](#) ."
- Le paramètre *MGMTclas REPLICAT* identifie la classe de gestion contenant les VTV que VSM réplique (copie) entre deux VTSS du cluster.

Le paramètre *CONFIG GLOBAL REPLICAT* indique maintenant quand répliquer un VTV, comme suit :

REPLICAT

spécifie quand VSM réplique le VTV.

ALWAYS

La demande de réplication est ajoutée à la file d'attente de réplication VTCS chaque fois que le VTV est démonté, qu'il ait été modifié ou non lors du montage (valeur par défaut).

CHANGED

La demande de réplication est ajoutée à la file d'attente de réplication VTCS si le VTV :

- a été modifié lors de son montage **ou**
- a été lu uniquement lors du montage mais un nombre de fois inférieur au nombre attendu de copies MVC existantes du VTV.

Quel que soit le paramètre *CONFIG GLOBAL REPLICAT*, la réplication requiert **également** que les conditions ci-dessous soient remplies.

- Le VTV doit être démonté dans un VTSS prenant en charge la réplication **et** il ne doit pas exister de copie identique du VTV dans l'autre VTSS au sein du cluster ;
- Outre la valeur *CONFIG GLOBAL REPLICAT* , vous **devez** indiquer *REPLICAT(YES)* sur la classe de gestion d'un VTV pour que la réplication ait lieu.

Pour plus d'informations, reportez-vous au manuel *Référence des commandes, des instructions de contrôle et des utilitaires ELS*.

- VTCS migre immédiatement (à l'aide de *KEEP*) les VTV répliqués. Vous pouvez indiquer le VTSS source pour la migration des VTV répliqués dans le paramètre *MIGRATE* de l'instruction *STORclas*. **Notez également** que vous **devez** spécifier la réplication sur une classe de gestion qui pointe vers une classe de stockage avec une valeur de paramètre *MIGRATE* pour effectuer la migration à partir du VTSS souhaité. Sinon, la migration à partir du VTSS souhaité peut ne pas avoir lieu.

VTCS migrant immédiatement (à l'aide de *KEEP*) les VTV répliqués quel que soit le paramètre *MGMTclas IMMDELAY*, StorageTek **recommande vivement de ne pas** définir explicitement une stratégie *MGMTclas IMMDELAY* pour les VTV répliqués. Dans

le cas contraire, VTCS honore la demande de migration explicite immédiate et migre immédiatement le VTV en question à partir du premier VTSS capable d'effectuer la migration (à savoir le premier VTSS disposant d'une copie VTV résidente et d'un RTD disponible pour réaliser la migration). La définition d'une stratégie *MGMTclas IMMDELAY* explicite est donc redondante et peut perturber les performances de réplication et de migration des VTV.

Notez également que la migration immédiate (*KEEP*) qui suit la réplication **n'est pas identique** à la migration automatique. Ainsi, lors de la migration immédiate implicite, aucun VTV n'est supprimé d'un VTSS pour gérer la DBU. Les VTV sont simplement "pré-transférés" à l'aide de la migration vers un MVC à partir du VTSS récepteur, laissant alors le contenu des **deux** tampons VTSS en l'état. Pour la gestion d'espace dans un cluster VTSS, VTCS migre automatiquement les VTV en fonction du cycle de gestion d'espace/ de migration de **l'un des** VTSS. Si la capacité du VTSS récepteur est supérieure ou égale à celle du VTSS émetteur, la migration automatique sur le VTSS émetteur supprime un VTV répliqué des deux VTSS. Si la capacité du VTSS récepteur est inférieure à celle du VTSS émetteur, la migration automatique peut démarrer sur le VTSS récepteur. Dans ce cas, la migration automatique supprime un VTV répliqué uniquement sur le VTSS récepteur, laissant la copie sur le VTSS émetteur.

- **Notez que** les conditions requises pour la réplication des données sont déterminées à la suite d'un démontage et **non** après un rappel. Le rappel d'un VTV n'entraînera pas de réplication – par conséquent, le rappel selon les besoins, *MVCdrain* et la récupération ne provoqueront pas de réplication. Cependant, si le VTV est rappelé et monté sur un VTD, il sera répliqué sur le VTSS secondaire ou homologue lors du démontage.
- Un cluster peut prendre en charge des charges globales différentes dans chacun des quatre modes de fonctionnement. Par exemple, seuls les clusters pleines fonctions peuvent prendre en charge la réplication active. Cependant, en mode principal dégradé, vous pouvez basculer en ligne les VTD du VTSS secondaire vers MVS pour reprendre la charge globale. Vous pouvez utiliser *Query* pour afficher le statut d'un cluster, d'un lien de cluster, d'une réplication et d'un VTSS. Vous pouvez utiliser *VARY VTSS* pour modifier les états des VTSS et *VARY CLink* pour modifier les états des CLINK.

Fonctionnement du rapprochement VTSS

- Lorsqu'une paire de VTSS en cluster repasse à l'état pleines fonctions, VTCS rapproche le contenu des deux VTSS. Cela se produit lors de l'initialisation de VTCS ou lors du passage en ligne d'un VTSS, alors que son VTSS partenaire est également en ligne.
- Le rapprochement consiste à supprimer ou migrer et supprimer des VTV (ou à répliquer un VTV si l'opération n'avait pas abouti précédemment). Le rappel n'entre donc pas en jeu dans le rapprochement du contenu des VTSS.

Par exemple, dans un cluster unidirectionnel avec un VTV qui se trouve sur le VTSS récepteur mais pas sur le VTSS émetteur, VTCS supprime le VTV du récepteur (après s'être assuré que toutes les copies MVC requises ont été réalisées). Cela évite ainsi l'envoi d'un rappel à l'émetteur.

De même, dans un cluster unidirectionnel avec un VTV qui se trouve sur le VTSS émetteur mais pas sur le VTSS récepteur, VTCS réplique le VTV sur le récepteur au lieu de le rappeler à partir d'un MVC.

- Le processus de rapprochement suppose qu'un VTV répliqué qui se trouve sur le VTSS émetteur est une copie valide. Si la copie figurant sur le récepteur est différente, VTCS la supprime.
- Pour que les opérations de rapprochement restent cohérentes dans un cluster bidirectionnel, le VTSS où se trouve le VTV ou il se trouvait la dernière fois (comme indiqué par l'enregistrement VTV du CDS), est considéré comme le VTSS émetteur. Le traitement du rapprochement décrit ci-dessus concerne les clusters unidirectionnels.

Clusters unidirectionnels et bidirectionnels

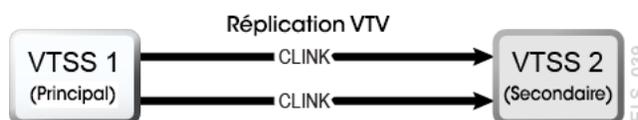
Les clusters de deux VTSS peuvent être :

- unidirectionnels, si un VTSS est le VTSS principal et l'autre le VTSS secondaire. Pour plus d'informations, voir "[VTSS en cluster unidirectionnel](#)."
- bidirectionnels, si les deux VTSS sont homologues et que la réplication s'effectue d'homologue à homologue dans n'importe quel sens. Pour plus d'informations, voir "[VTSS en cluster bidirectionnel](#)."

Clusters unidirectionnels

Comme indiqué à la [Figure 6.1, « VTSS en cluster unidirectionnel »](#), dans un cluster unidirectionnel, la réplication s'effectue uniquement du VTSS principal vers le VTSS secondaire.

Figure 6.1. VTSS en cluster unidirectionnel



Fonctionnement des clusters VTSS unidirectionnels

- Le VTSS secondaire peut recevoir les VTV répliqués du VTSS principal et la charge globale de production non répliquée à l'aide d'une des méthodes de routage standard (par exemple, *TAPEREQ*). Vous devez basculer en ligne les VTD du VTSS secondaire afin que ce dernier puisse accepter le travail de production. Vous ne pouvez pas basculer en ligne les adresses VTD utilisées par les interruptions CLINK vers MVS, comme décrit dans la section "[Fonctionnement des configurations VTSS en cluster](#)."
- Un VTV pour lequel la réplication est activée est alloué à un VTSS principal en ligne, sauf si aucun d'eux n'est disponible, auquel cas le VTV est alloué à un VTSS secondaire en

ligne. Si aucun VTSS secondaire en ligne n'est disponible, le VTV est alloué à un VTSS qui n'est pas un cluster. Un VTV sans réplication peut être alloué à n'importe quel VTSS en ligne, y compris le VTSS secondaire d'un cluster pleines fonctions.

- Lors du démontage, un VTV pour lequel la réplication est activée et qui se trouve sur un cluster pleines fonctions est placé en file d'attente sur le VTSS secondaire. Si un VTV pour lequel la réplication est activée est démonté d'un VTD dans un VTSS qui ne fait pas partie d'un cluster pleines fonctions, le VTV est placé en file d'attente en vue d'une migration immédiate.

Lorsque le VTSS secondaire reçoit un VTV répliqué du VTSS principal, le VTV est alors immédiatement migré (à l'aide de l'option *KEEP*) quels que soient les paramètres de l'option Immediate Migrate Management Class (migration immédiate de la classe de gestion) pour ce VTV.

- **Les VTSS principaux et secondaires** peuvent tous gérer l'ensemble des récupérations d'espace.
- Si vous utilisez des interfaces ESCON ou FICON, sur le VTSS principal, les CIP/FIP CLINK sont configurés en *mode Nearlink*, tandis que sur le VTSS secondaire, les CIP/FIP sont configurés en *mode Hôte*.

Par conséquent, configurez les *CLINK* uniquement pour le VTSS principal, comme illustré dans l'exemple suivant, où VTSS1 correspond au VTSS principal.

```

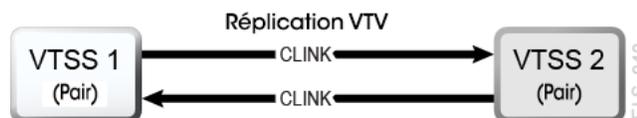
.
.
CLUSTER NAME=CLUSTER1 VTSSs(VTSS1,VTSS2)
  CLINK VTSS=VTSS1 CHANIF=0G
  CLINK VTSS=VTSS1 CHANIF=00
  CLINK VTSS=VTSS1 CHANIF=1G
  CLINK VTSS=VTSS1 CHANIF=10
.
.

```

Clusters bidirectionnels

Comme indiqué à la [Figure 6.2, « VTSS en cluster bidirectionnel »](#), le clustering bidirectionnel requiert des paires de *CLINK* unidirectionnels, afin que les données circulent en sens opposé sur les *CLINK*.

Figure 6.2. VTSS en cluster bidirectionnel



Fonctionnement des clusters VTSS bidirectionnels

Dans un cluster bidirectionnel, en mode de fonctionnement normal, les deux VTSS sont en ligne sur VTCS, comme suit :

- Dans un cluster bidirectionnel, chaque VTSS homologue peut recevoir un travail de production via les méthodes de routage standard (par exemple, TAPEREQ). Vous devez basculer en ligne les VTD des deux VTSS vers MVS, afin que chacun d'eux puisse accepter le travail de production. Toutefois, **notez que** vous **ne pouvez pas** basculer en ligne les adresses VTD utilisées par les connexions CLINK vers MVS, comme décrit dans la section "[Fonctionnement des configurations VTSS en cluster.](#)"
- Dans un cluster bidirectionnel, un VTV pour lequel la réplication est activée est alloué à l'un ou l'autre des VTSS homologues. Si l'un des deux VTSS homologues est hors ligne ou au ralenti, la charge globale de production peut s'exécuter sur le VTSS en ligne restant. Toutefois, les VTV nécessitant une réplication sont alloués au VTSS restant uniquement si aucun autre cluster pleines fonctions n'est disponible et adapté. Dans ce cas, les VTV répliqués sont migrés immédiatement à l'aide de KEEP et placés en file d'attente de réplication lorsque l'autre VTSS passe en ligne.
- Dans un cluster bidirectionnel, lors du démontage, un VTV pour lequel la réplication est activée et qui se trouve sur un cluster pleines fonctions est placé en file d'attente de réplication sur l'autre VTSS homologue. Si un VTV pour lequel la réplication est activée est démonté d'un VTD dans un VTSS qui ne fait pas partie d'un cluster pleines fonctions, le VTV est placé en file d'attente en vue d'une migration immédiate. **Notez que** les conditions requises pour la réplication des données sont déterminées à la suite d'un démontage et **non** après un rappel. Le rappel d'un VTV n'entraînera pas de réplication – par conséquent, le rappel selon les besoins, MVCdrain et la récupération ne provoqueront pas de réplication. Toutefois, si le VTV est rappelé et monté sur un VTD, lors du démontage il sera répliqué sur le VTSS secondaire, à moins que vous n'indiquiez REPLICAT(CHANGED) (option recommandée), qui provoque à nouveau la réplication du VTV uniquement en cas de modification des données.
- Les deux VTSS homologues peuvent gérer toutes les récupérations d'espace.
- Si vous utilisez des interfaces ESCON ou FICON :
 - Sur chaque VTSS homologue, les *CIP/FIP CLINK* émetteur sont configurés en *mode Nearlink*, tandis que les *CIP/FIP CLINK* récepteur sont configurés en *mode Hôte*.

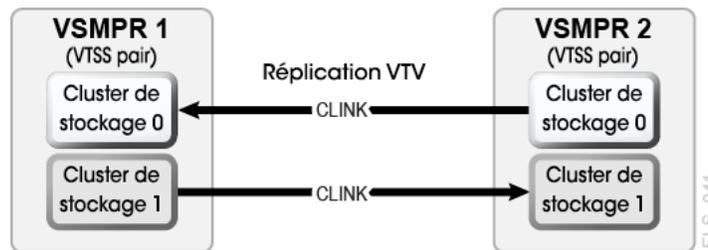
Par conséquent, configurez les *CLINK* émetteur sur chaque VTSS homologue, comme illustré dans l'exemple suivant, où VSMR1 et VSMR2 sont des VTSS homologues.

```
.
.
CLUSTER NAME=CLUSTER1 VTSSs(VSMR1,VSMR2)
CLINK VTSS=VSMR1 CHANIF=00:0
CLINK VTSS=VSMR1 CHANIF=00:1
CLINK VTSS=VSMR2 CHANIF=10:0
CLINK VTSS=VSMR2 CHANIF=10:1
```

- Chaque CLINK doit être connecté au *même cluster de stockage* sur chaque VTSS (cluster de stockage 0 connecté au cluster de stockage 0 ou cluster de stockage 1 connecté au cluster de stockage 1). Si la configuration ne répond pas à cette exigence, vous risquez d'être confronté à des erreurs de réplication, de canal et de communication.

Comme illustré dans l'exemple de la [Figure 6.3, « CLINK ESCON/FICON pour VTSS en cluster bidirectionnels »](#), le port CLINK émetteur (mode Nearlink) sur VSMMPR1 se trouve sur le cluster de stockage 1 et se connecte à un port CLINK récepteur (mode Hôte), *qui se trouve également sur le cluster de stockage 1* de VSMMPR2. De même, un port CLINK émetteur sur le cluster de stockage 0 de VSMMPR2 se connecte à un port CLINK récepteur sur le cluster de stockage 0 de VSMMPR1.

Figure 6.3. CLINK ESCON/FICON pour VTSS en cluster bidirectionnels

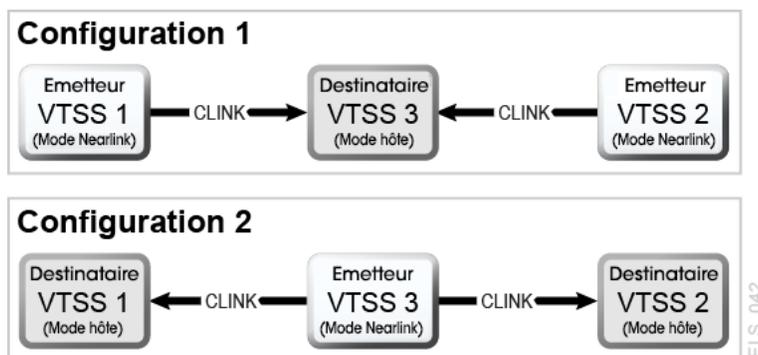


Clustering étendu

Le clustering étendu permet à trois VTSS au minimum d'être connectés par des CLINK dans une configuration à un seul Tapeplex (1 CDS). Le clustering est une solution de haute disponibilité conçue de sorte qu'une charge globale puisse continuer à s'exécuter sans interruption lors d'une coupure VTSS. Le clustering requiert que tous les sous-systèmes VTSS appartenant à un cluster aient accès à tous les MVC générés par un seul sous-système VTSS dans ce cluster. Si un VTSS au sein d'un cluster se connecte à un Tapeplex distant (CTR), tous les systèmes VTSS du cluster doivent se connecter au même Tapeplex pour conserver la fonctionnalité HA.

Le clustering étendu permet de configurer un VSM avec des CLINK connectés à plusieurs VTSS. Le nombre de connexions CLINK se limite uniquement au nombre de connexions physiques disponibles. Le microcode D02.07.00.00 ou niveau supérieur est requis. Toutes les règles de clustering et de réplication disponibles s'appliquent au clustering étendu. Toutes les configurations de cluster étendu se fondent sur les deux configurations unidirectionnelles de base présentées à la [Figure 6.4, « Configurations de cluster étendu de base »](#).

Figure 6.4. Configurations de cluster étendu de base



Réplication synchrone ou asynchrone

Vous avez le choix entre la réplication synchrone ou asynchrone, en fonction des stratégies en vigueur sur votre site.

Implémentation de la réplication synchrone

Attention :

Avec la réplication synchrone, le temps requis pour la réplication d'un volume virtuel retarde l'achèvement d'un travail de création de données soumis à une stratégie de réplication synchrone.

1. Vérifiez que votre système répond aux conditions requises pour la réplication synchrone, qui sont décrites dans le [Tableau 6.2, « Conditions requises pour la réplication synchrone »](#).
2. Lorsque tous les systèmes HSC/VTCS sont arrêtés, utilisez `CONFIG GLOBAL` pour activer la réplication synchrone :

```
CONFIG GLOBAL SYNCHREP=YES
```

3. Vérifiez que le paramètre `CONFIG GLOBAL REPLICAT` est défini comme vous le voulez :

ALWAYS

La demande de réplication est ajoutée à la file d'attente de réplication VTCS chaque fois que le VTV est démonté, qu'il ait été modifié ou non lors du montage (valeur par défaut).

CHANGED

La demande de réplication est ajoutée à la file d'attente de réplication VTCS si le VTV :

- a été modifié lors de son montage **ou**
- a été lu uniquement lors du montage mais un nombre de fois inférieur au nombre attendu de copies MVC existantes du VTV.

4. Spécifiez la réplication synchrone dans les instructions `MGMTCLas` souhaitées :

```
MGMT (name) . . . . . REP(YES_SYNC)
```

Implémentation de la réplication asynchrone avec surveillance des travaux

Vous pouvez choisir d'utiliser la réplication asynchrone tout en voulant savoir si la réplication a abouti. Dans cette procédure, vous vous servez de l'utilitaire *DRMONitr* pour surveiller la suspension du travail MVS associé jusqu'à ce que la réplication aboutisse.

1. Vérifiez que votre système répond aux conditions requises pour la réplication synchrone, qui sont décrites dans le [Tableau 6.2, « Conditions requises pour la réplication synchrone »](#).
2. Lorsque tous les systèmes HSC/VTCS sont arrêtés, utilisez *CONFIG GLOBAL* pour activer la réplication asynchrone :

```
CONFIG GLOBAL SYNCHREP=NO
```

3. Vérifiez que le paramètre *CONFIG GLOBAL REPLICAT* est défini comme vous le voulez :

ALWAYS

La demande de réplication est ajoutée à la file d'attente de réplication VTCS chaque fois que le VTV est démonté, qu'il ait été modifié ou non lors du montage (valeur par défaut).

CHANGED

La demande de réplication est ajoutée à la file d'attente de réplication VTCS si le VTV :

- a été modifié lors de son montage *ou*
- a été lu uniquement lors du montage mais un nombre de fois inférieur au nombre attendu de copies MVC existantes du VTV.

4. Spécifiez la réplication asynchrone dans les instructions *MGMTCLas* souhaitées :

```
MGMT (mgmtname) . . . . . REP(YES)
```

5. Créez un JCL pour surveiller la réplication asynchrone.

Pour ce faire, servez-vous de l'utilitaire *DRMONitr* afin de surveiller la réplication. *DRMONitr* provoque la suspension du travail MVS associé jusqu'à ce que la réplication aboutisse. Par exemple :

```
//MONITOR EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//* If HSC IS NOT OR MAY NOT BE ACTIVE, INCLUDE THE
//* FOLLOWING:
//SLSCNTL DD DSN=primary.cds.name, DISP=SHR
//SLSCNTL2 DD DSN=secondary.cds.name, DISP=SHR
```

```

//SLSSTBY DD DSN=standby.cds.name,DISP=SHR
//SLSPARMP DD DSN=h1q.PARMLIB(BKPCNTL),DISP=SHR
//SLSPARMS DD DSN=h1q.PARMLIB(BKPCNTL2),DISP=SHR
//SLSPARMB DD DSN=h1q.PARMLIB(BKPSTBY),DISP=SHR
//SYSIN DD UNIT=SYSDA,SPACE=(TRK,1)
// * THE FOLLOWING IS USED BY THE SNAPSHOT UTILITY:
//SYSPRINT DD SYSOUT=*
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
DRMON MGMT(mgmtname) REPL MAXAGE(24) TIMEOUT(120)

```

Dans cet exemple, l'utilitaire *DRMON* surveille les répliquions pour la classe de gestion spécifiée. En outre, surveillez uniquement les VTV qui ont été mis à jour au cours des dernières 24 heures et spécifiez le délai d'expiration de *DRMON* sur 120 minutes.

Clustering avec connexions TCP/IP

La fonctionnalité de connexion IP native de VTSS vous permet d'utiliser le protocole TCP/IP pour "mettre en cluster" (connecter) deux VTSS au minimum pour la répliquion VTV. Avec le clustering IP natif, chaque VTSS dispose de ports Ethernet pour la connexion au réseau TCP/IP. Auparavant, vous étiez limité aux connexions ESCON ou FICON pour la répliquion. L'utilisation de TCP/IP pour les CLINK permet d'améliorer les performances de répliquion sur les protocoles ESCON ou FICON et, le cas échéant, d'utiliser les ports ESCON ou FICON existants exclusivement pour les connexions RTD et hôtes, où les combinaisons suivantes sont prises en charge :

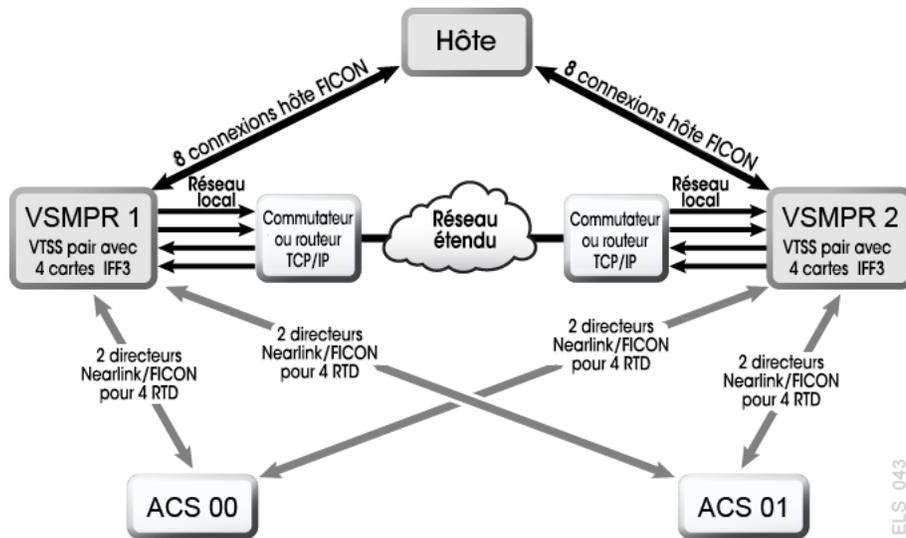
- VSM5 à VSM5
- VSM5 à VSM 6
- VSM 6 à VSM 6

Cette section décrit uniquement l'implémentation VTCS pour le protocole IP natif. Le personnel du support technique StorageTek ou les autres QSP sont responsables de la configuration VTSS annexe.

Environnement TCP/IP

Les CLINK connectés à TCP/IP ont la même fonction que les CLINK connectés à un canal FICON ou ESCON, mais un CLINK TCP/IP se connecte à l'aide d'un port Ethernet sur le VTSS et non à partir d'un port ESCON ou FICON. L'exemple de la [Figure 6.5, « Environnement TCP/IP avec deux VSM5 »](#) présente des VSM5 homologues, chacun doté de 4 cartes IFF3 avec ports Ethernet. Les câbles Ethernet provenant des ports Ethernet sur les cartes IFF3 se connectent à des réseaux locaux (LAN, un pour chaque VTSS), qui sont eux-mêmes connectés via un réseau WAN.

Figure 6.5. Environnement TCP/IP avec deux VSM5



Configuration de VTCS pour CLINK TCP/IP

Vous trouverez ci-dessous les paramètres pour l'instruction `CONFIG CLINK`.

Instruction CONFIG CLINK

L'instruction `CONFIG CLINK` fournit deux types de connexions VTSS à VTSS par le biais des paramètres suivants :

CLINK CHANIF=*nn* ou *nm:n*

définit un port FICON ou ESCON à utiliser comme CLINK.

CLINK IPIF=*ci:p*

définit un port Ethernet à utiliser comme CLINK. Les valeurs admises pour `CONFIG RTD IPIF` sont $c=0$ ou 1 , $i=A$ ou I , $p=0$ à 3 pour les VSM5 et VSM 6. Pour les VSM5, cette valeur doit correspondre aux valeurs spécifiées dans l'écran IFF Configuration Status Screen du VSM5. Pour les VSM 6, la valeur doit être unique pour chaque VTSS et ne correspond pas à une valeur réelle sur les ports TCP/IP du VSM 6.

Remarque :

L'instruction `CLINK` doit contenir le paramètre `CHANIF` ou `IPIF`, mais pas les deux.

Chapitre 7. Utilisation du logiciel Concurrent Disaster Recovery Test (CDRT)

Les clients qui utilisent ou gèrent un site de récupération après sinistre (DR) dans le cadre d'un plan de continuité de l'activité voudront peut-être valider périodiquement leur capacité à poursuivre un traitement de production normal avant qu'un réel sinistre ne se produise ; d'autres n'ont pas le choix et doivent prouver régulièrement la préparation de leur modèle de continuité de l'activité pour répondre aux exigences de leur assurance et/ou de leurs auditeurs.

Grâce à la fonction Concurrent Disaster Recovery Test (CDRT), qui est désormais intégrée au logiciel StorageTek ELS, les entreprises qui utilisent actuellement des bandothèques StorageTek Streamline et/ou Nearline (matériel réel), VSM (matériel virtuel) et les logiciels associés (HSC, VTCS) peuvent valider leur fonctionnalité de continuité de l'activité sur bande réelle et virtuelle sans avoir besoin d'acquérir du matériel et des logiciels supplémentaires.

CDRT prend en charge un test parallèle des applications et hôtes de production avec un accès simultané aux données de production à la fois par les systèmes de production et de test DR.

Les principaux concepts de CDRT sont les suivants :

- Grâce à CDRT, un test DR peut s'exécuter avec du matériel réel et/ou virtuel.
- CDRT, HSC et VTCS appliquent par programmation certaines restrictions fonctionnelles lors de la préparation du CDS et du test DR réel pour tenter de garantir l'intégrité du système.
- CDRT sépare logiquement une partie du matériel virtuel et réel de production existant et les pools de volume de bande pendant la durée du test DR. Cela vous permet de tester votre configuration DR tout en exécutant simultanément un travail de production, de garantir l'intégrité des données de production et de réduire les conflits pour les ressources matérielles et les volumes de bande.
- CDRT crée une copie de test du CDS de production. Le sous-système ELS de production et les sous-systèmes ELS de test DR ne communiquent donc pas entre eux. Les changements qui ont lieu dans le CDS de test DR ne sont pas représentés dans la copie du CDS de production et inversement. Les hôtes de test DR utilisent uniquement le matériel séparé logiquement. Les hôtes de production continuent à utiliser tous les matériels à une exception près : les hôtes de récupération après sinistre disposent d'une utilisation exclusive des VTSS séparés logiquement lors du test DR. D'autres ressources, comme les RTD, les cartouches MVC (Multiple Virtual Cartridge) et les bandes provisoires réelles, doivent être contrôlées en définissant des pools distincts sur chaque jeu d'hôtes.

- Un test DR peut être exécuté à l'aide de ressources locales uniquement ou d'une combinaison de ressources locales et distantes ; les configurations composées d'un site distant contenant uniquement du matériel réel et virtuel, ou de matériel réel et virtuel avec un processeur mainframe sont également prises en charge.
- Le matériel de test DR peut inclure toute combinaison d'ACS, VTSS et VLE.
- Après un test DR, la copie de test du CDS et toutes les données créées depuis le test DR sont généralement effacées et le matériel séparé logiquement est redéployé vers l'environnement de production normal.

Remarque :

- Pour répondre aux exigences de récupération après un sinistre réel, il est essentiel que les travaux figurant dans un flux de travaux de test DR **ne mettent pas à jour** les volumes créés en production, que ce soit via *DISP=MOD* ou en les écrasant. Si un véritable sinistre se produisait, de telles pratiques rendraient l'état de ces volumes imprévisible.
 - Pour l'exécution du test DR, il est **fortement recommandé** que les volumes de production qui auraient pu être modifiés pendant le test DR soient copiés sur de nouveaux volumes au début du test et que les volumes **copiés** soient mis à jour par le test DR, non les volumes d'origine. De même, le JCL doit être modifié si possible, afin que le statut de tous les volumes de bande au moment du sinistre soit connu.
-

Considérations relatives aux métadonnées

Pour que l'exécution d'un test DR à l'aide de la fonction CDRT aboutisse, il est fondamental de disposer d'une copie cohérente de l'état de tous les volumes de bande gérés par le logiciel ELS, ainsi que du matériel réel et virtuel. C'est la cohérence de l'état des volumes de bande entre les hôtes de production et les hôtes de récupération après sinistre au démarrage du test DR qui permet le traitement parallèle des applications client. Etant donné que le CDS reflète l'état de tous les volumes de bande et ressources dans le matériel réel et virtuel, CDRT répond partiellement à cette exigence de cohérence lorsqu'il réalise sa copie de test du CDS.

Toutefois, dans un environnement de volume de bande, il arrive souvent que des données d'état (métadonnées) de ce volume de bande soient conservées et gérées en dehors du sous-système ELS ainsi que du matériel réel et virtuel. Généralement, les métadonnées de volume de bande (à savoir VOLSER, DSN, date d'expiration, état de travail, désignation réelle ou virtuelle, etc.) sont stockées dans un ou plusieurs catalogues de gestion de bandes (TMC), un ou plusieurs catalogues z/OS et le jeu CDS.

Vous devez coordonner la création de copies des métadonnées conservées et gérées en dehors de ELS (et du matériel réel et virtuel) avec la création de la copie de test du CDS par CDRT.

Origine des données VTV du CDRT

CDRT obtient ses données VTV d'une ou plusieurs des ressources de site DR suivantes :

- Cartouche MVC
- VLE

- VTSS

Le CDS de production étant la source d'informations relatives aux VTV accessibles par le test DR, il est important de s'assurer que le cycle de synchronisation de volumes provisoires permet aux volumes utilisés dans le test DR de ne pas être mis à l'état provisoire avant le début du test. StorageTek recommande également de basculer votre VTSS de test hors ligne avant de lancer l'opération *DRTEST CREATE*.

Notez que l'exécution d'une suppression lors du test **n'affecte pas** le contenu du VTSS du test DR, ni des cartouches MVC, s'ils sont définis sur *READONLY* à l'aide de l'utilitaire *ACTMVCGN*.

La copie du CDS du site de récupération après sinistre indique l'emplacement des VTV au moment de la copie, généralement sur des cartouches MVC ou des VLE. Toutefois, vous pouvez parfois choisir d'utiliser des copies VTV dans un VTSS qui se trouve sur le site de test. En général, vous procédez comme cela dans les situations suivantes :

- Vous définissez votre VTSS de récupération après sinistre à l'aide du mot-clé *SHARE*, qui empêche la mise à jour du contenu d'un VTV de production.
- Vous disposez de deux VTSS en cluster (l'un sur le site de production et l'autre sur le site DR) et votre test DR **ne modifie pas** le contenu d'un VTV de production.
- Vous disposez de deux VTSS en cluster (l'un sur le site de production et l'autre sur le site DR) **et** vous voulez passer du temps après le test DR à identifier et migrer manuellement les VTV mis à jour par le test DR.

Remarque :

Vous pouvez vous servir de l'utilitaire *DRMONitr* pour vous assurer que les données critiques de récupération après sinistre atteignent l'emplacement de récupération désigné avant d'exécuter un test DR.

Opérations exécutées sur les données à la fin du test

Les données qui sont créées par le test DR sont uniquement représentées dans le CDS du test DR (qui est effacé après le test) et sur les cartouches MVC de test DR, qui se trouvent dans une plage de volumes distincte des MVC de production. En outre, les VTV créés par le test DR restent sur le VTSS de test DR après le test, sauf si vous effectuez l'une des opérations suivantes :

1. Servez-vous de l'utilitaire *SLUADMIN SCRATCH* dans l'environnement de test DR pour supprimer (à l'aide de *MGMTCLAS DELSCR(YES)*) tous les VTV dans la plage du sous-pool de test DR. Cette option requiert l'utilisation de la fonction *POOLPARM/VOLPARM*.
2. Vérifiez que tous les VTV de production qui ont été modifiés par le test DR sont migrés vers les MVC de test DR à l'aide de la commande *MIGRATE* avec *DELETE(YES)* à partir du système de test DR. Si vous n'effectuez pas cette opération, le système de production utilise les données qui ont été modifiées par le test DR.
3. Demandez à votre CSE StorageTek ou à un autre QSP de "nettoyer" le VTSS de test DR après le test pour supprimer toutes les données du VTSS.

Remarque :

Si vous utilisez l'option 2 ou 3, le contenu du VTSS **ne correspondra pas** au système de production, car les VTV manquent sur le CDS de test DR lorsqu'il est remis en production. Bien que cette condition soit traitée par le logiciel, les performances de production risquent de s'en trouver diminuées.

Gestion des ressources CDRT

Les sections suivantes expliquent comment gérer les ressources CDRT.

Ressources de volume

La première étape de gestion des ressources CDRT consiste à définir les volumes dans votre système à l'aide de l'utilitaire *POOLPARAM/VOLPARAM*. Cette fonctionnalité simplifie la gestion globale des volumes et pools de bande. Elle fournit également une méthode de séparation des volumes inscriptibles dans une configuration CDRT. L'utilisation de *POOLPARAM/VOLPARAM* est requise lorsque les VTSS de test DR sont partagés et est **fortement recommandée** dans les autres scénarios.

Remarque :

L'utilitaire *SLUADMIN SET VOLPARAM* doit s'exécuter à l'aide du CDS de production et définir à la fois les pools de test DR et de production. L'utilitaire *SET VOLPARAM* **n'est pas** valide pour un CDS de test DR.

Sous-pools provisoires

Les sous-pools provisoires peuvent s'appliquer à tous les scénarios de test DR. La syntaxe suivante montre l'utilisation des définitions *POOLPARAM/VOLPARAM* pour configurer les sous-pools provisoires de test DR et de production. Si vous utilisez les mêmes noms pour les sous-pools de production et de test DR (avec des plages de volume différentes), vous n'avez pas besoin de modifier vos stratégies de production lors de l'exécution du test DR, comme indiqué dans l'exemple suivant.

```
*   SCRATCH POOLS
POOLPARAM NAME(SCR1) TYPE(SCRATCH)
VOLPARAM VOLSER(T11000-T11999) MEDIA(T10000T1) RECTECH(T1AE)
POOLPARAM NAME(SCR1) TYPE(SCRATCH) DRTEST
VOLPARAM VOLSER(T12000-T12999) MEDIA(T10000T1) RECTECH(T1AE)
POOLPARAM NAME(SCR1TV1) TYPE(SCRATCH)
VOLPARAM VOLSER(V1000-V1999) MEDIA(VIRTUAL)
POOLPARAM NAME(SCR1TV1) TYPE(SCRATCH) DRTEST
VOLPARAM VOLSER(V2000-V2999) MEDIA(VIRTUAL)
```

Lorsque vous définissez des sous-pools provisoires à l'aide de l'utilitaire *POOLPARAM/VOLPARAM*, vous pouvez vous servir de l'utilitaire *HSC SLUADMIN SCRATCH* pour supprimer des volumes de sortie de test DR dans l'environnement de test DR. Combinée à une classe de gestion spécifiant **DELSCR(YES)**, l'exécution de l'utilitaire de suppression supprime du VTSS les VTV créés par le test DR.

Ressources MVC

Les ressources MVC sont utilisées dans tous les scénarios de test DR, à l'exception des VSM sans bandes et à bandes réelles uniquement. L'exemple suivant montre l'utilisation des définitions *POOLPARAM/VOLPARAM* pour configurer les pools MVC de test DR et de production.

```
*   MVC POOLS
POOLPARAM NAME(MVCP1) TYPE(MVC) MCVFREE(40) MAXMVC(4) THRESH(60) +
START(70)
VOLPARAM VOLSER(T14000-T14999) MEDIA(T10000T1) RECTECH(T1AE)
POOLPARAM NAME(MVCP1) TYPE(MVC) MCVFREE(40) MAXMVC(4) THRESH(60) +
START(70) DRTEST
VOLPARAM VOLSER(T13000-T13999) MEDIA(T10000T1) RECTECH(T1AE)
```

Conservez le contenu des MVC de production en vue de l'utiliser dans un scénario de test DR en suivant la procédure ci-dessous.

- Servez-vous de l'utilitaire *ACTMVCGN* pour définir les MVC qui seront utilisées comme entrée du test DR sur *READONLY*, comme indiqué dans l'exemple suivant.

```
//ACTMVCG1 EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
/* NOTE: MVCMAINT READONLY(ON) STATEMENTS
//SLUSMVON DD DSN=h1q.SLUSMVON, DISP=(NEW, CATLG, DELETE),
// SPACE=(CYL, 1)
/* NOTE: MVCMAINT READONLY(OFF) STATEMENTS
//SLUSMVOF DD DSN=h1q.SLUSMVOF, DISP=(NEW, CATLG, DELETE),
SPACE=(CYL, 1)
/* NOTE: THE FOLLOWING STEP SELECTS ALL "ACTIVE" MVCS
/* IN ACS 01.
//SLSIN DD *
ACTMVCGN ACS(01)
/*
//ACTMVCG2 EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
/* NOTE: EXEC MVCMAINT TO SET READONLY(ON)
```

- Vérifiez que la récupération ne s'exécute pas sur tous les hôtes de production à l'aide de *VTCS CONFIG* :

```
CONFIG HOST NAME(host) NORECLAM
```

Ressources VTSS

Il existe deux types de ressources VTSS : non partagées et partagées.

Ressources VTSS non partagées

Dans un environnement où les VTV sont migrés vers des MVC ou des VLE, les ressources VTSS sont séparées lors d'un test DR. Le système de test DR n'a accès qu'aux VTSS qui sont définis via les utilitaires *DRTEST PRIMEPRD/CREATE* et qui doivent rester hors ligne en production. La commande *DRTEST START* est rejetée si un VTSS de récupération après sinistre est en ligne dans l'environnement de production.

Dans certains cas, deux VTSS peuvent porter le même nom dans les environnements de production et de test. Le paramètre *SPARE* de l'utilitaire *DRTEST PRIMEPRD/CREATE* indique alors que le VTSS de test DR porte le même nom qu'un VTSS de production, mais n'est pas physiquement le même périphérique, permettant ainsi au périphérique de production de rester en ligne lors du test. N'utilisez pas le paramètre *SPARE* dans les autres cas.

VTSS en cluster dans CDRT

Le " [Scénario 4 : VTSS en cluster avec sites de production et de test DR](#) " utilise des VTSS en cluster pour fournir des données au VTSS sur le site DR. Dans ce scénario, le cluster **ne fonctionne pas** lors du test DR, car le VTSS de test DR est hors ligne en production.

Si les VTV de production sont modifiés de quelque façon que ce soit lors du test DR, vous **devez** vérifier que les VTV modifiés sont supprimés du VTSS avant de rebasculer en ligne le VTSS de test en production. Sinon, les VTV modifiés peuvent être sélectionnés par le site de production. Pour empêcher cela, **StorageTek recommande vivement** de concevoir le test DR de sorte que les VTV de production ne soient pas modifiés par le test.

Dans un scénario de VTSS en cluster, il se peut qu'aucun VTD du VTSS de test ne soit accessible par l'hôte de production. Pour autoriser une communication ECAM entre la production et le VTSS de test DR, spécifiez l'instruction suivante dans *VTCS CONFIG* :

```
VTD LOW=xxxx HIGH=xxxx NOVERIFY
```

Préparation d'un cluster VTSS en vue d'un test de récupération après sinistre

1. Basculez le VTSS de test DR à l'état Mis au ralenti. Par exemple :

```
VARY VTSS1 QUIESCED
```

Ici, l'objectif vise à arrêter (progressivement) la réplication sur VTSS1 afin que vous puissiez l'utiliser exclusivement pour le test DR.

2. Surveillez la réplication jusqu'à ce qu'elle se termine à l'aide de *Display REPLICat*. Ici, la réplication est encore active :

```
VTSS  HOST  QDEPTHVTSS0  PRODUCTION  1
```

Vous savez que la réplication est terminée lorsque le message suivant s'affiche :

```
VTSS  HOST  QDEPTH
VTSS0  PRODUCTION  0
```

3. Vérifiez que la réplication est terminée en consultant le statut du CLINK à l'aide de *Display CLINK*. Ici, le CLINK est encore actif :

```
VTSS  CLINK  STATUS  USAGE      HOST
VTSS0  7  ONLINE  REPLICATING  PRODUCTION
VTSS0  8  ONLINE  REPLICATING  PRODUCTION
You know the CLINK is no longer active when you see this:
VTSS  CLINK  STATUS  USAGE      HOST
VTSS0  7  ONLINE  FREE
VTSS0  7  ONLINE  FREE
```

4. Basculez hors ligne le VTSS de test DR :

```
VARY VTSS1 OFFLINE
```

Gestion du contenu VTSS avant et après le test DR

Avant de démarrer un test DR, vous **devez** :

1. planifier votre test afin que la sortie du test ne passe pas en production par erreur ;
2. préparer votre test afin que votre CDS de test DR corresponde au contenu de récupération après sinistre.

En outre, **StorageTek recommande vivement** que tous les VTSS du site de test soient nettoyés dès que possible après le test. Cela permet de s'assurer que les VTSS du site de test sont vides au début du test suivant et qu'aucune donnée de test DR ne reste sur les VTSS qui sont renvoyés en production.

Vous pouvez nettoyer les VTSS de test en effectuant l'une des opérations suivantes :

1. N'autorisez pas le test DR à effectuer des mises à jour (remplacement ou ajout) sur les VTV de production et utilisez *POOLPARAM/VOLPARAM* pour définir des sous-pools provisoires de test DR. Cette méthode permet d'exécuter l'utilitaire de suppression pour supprimer tous les VTV figurant dans la plage de test DR. Ils sont automatiquement supprimés du VTSS (si la classe de gestion indique *DELSCR(YES)*).
2. Si votre test DR peut mettre à jour les VTV de production, vous devez vérifier que les VTSS utilisés pour la sortie de test DR sont vidés avant de remettre le VTSS en production. Cette opération peut être effectuée via une migration vers zéro dans

l'environnement de test DR, ou en demandant au CSE StorageTek ou à un autre QSP de "nettoyer" le VTSS.

Lorsque vous exécutez l'utilitaire *DRTEST CREATE* pour créer le CDS de test DR, les métadonnées relatives au contenu du VTSS sont propagées vers l'environnement de test. L'environnement de test DR peut accéder aux VTV situés sur le VTSS de test DR.

Si le VTSS de test DR est défini en tant que VTSS de rechange, le contenu du VTSS physique de récupération après sinistre ne correspondra pas aux métadonnées du CDS, qui font référence à l'instance de production du VTSS de rechange. Pour supprimer la divergence, avant de créer le CDS de test DR, migrez l'instance de production du VTSS de rechange vers 0 dans l'environnement de production. Vous pouvez exécuter *VTCS VTVRPT OPTION(UNAVAIL)* pour vérifier que tous les VTV sont migrés et accessibles par les autres VTSS. Si vous n'effectuez pas cette étape, les tentatives d'accès aux VTV sur le VTSS de rechange par le test DR provoqueront l'affichage de messages SLS6680E pour les montages VTV.

Ressources VTSS partagées

Si l'environnement de bande n'inclut pas de MVC de bande réelle ou VLE, vous pouvez exécuter CDRT dans un environnement VTSS partagé. Le paramètre *DRTEST PRIMEPRD/CREATE SHARE* indique qu'un VTSS est partagé entre la production et le test DR lors du test. Cet environnement applique les restrictions suivantes :

1. *DRTEST CREATE* ne peut pas non plus définir de ressources DRACS ou STORMNGR. Par conséquent, aucune donnée ne peut être migrée à partir du VTSS partagé vers le média externe.
2. Le système de production doit définir les ressources de volume à l'aide de la fonction *POOLPARM/VOLPARM*.
3. Les montages d'un VTV de sous-pool non *DRTEST* dans l'environnement de test DR impliquent obligatoirement que le VTV soit en lecture seule. Le test DR n'est donc pas autorisé à effectuer un ajout à un VTV de production, ni à le remplacer.

Ressources ACS

Les ressources ACS sont utilisées dans les scénarios de test DR avec des bandes réelles uniquement ou avec des bandes virtuelles sans VLE dans un environnement VSM avec bandes. Les ressources ACS pour CDRT sont spécifiées dans le paramètre *DRTEST PRIMEPRD/CREATE DRACS*.

Restrictions relatives aux ressources ACS pour le test de récupération après sinistre

Vous devez émettre la commande HSC *CAPPREF* pour définir les ports d'accès aux cartouches (CAP) sur Manuel, avant de lancer l'utilitaire *DRTEST CREATE*. Le logiciel garantit

qu'ils resteront dans cet état tant que le test sera en cours. Une fois que vous avez entré la commande *DRTEST START*, des restrictions s'appliquent à l'ACS de test de récupération après sinistre à la fois en production et dans l'environnement de test DR, pour assurer la cohérence entre le matériel et les CDS respectifs afin de permettre aux deux sites d'accéder aux données. La commande *DRTEST START* est rejetée si un CAP dans un ACS de test DR est en mode automatique dans l'environnement de production.

Les restrictions ci-dessous sont automatiquement appliquées par le logiciel.

Conditions requises pour l'hôte de production DR

Les conditions requises pour l'hôte de production DR lors d'un test DR actif pour l'ACS de récupération après sinistre sont les suivantes :

- Les CAP doivent rester en mode manuel.
- *FLOAT(OFF)* et *EJCTAUTO(OFF)* sont automatiquement appliqués, quels que soient les paramètres de la commande MNTD.
- Vous ne pouvez pas exécuter d'utilitaires de rejet, de déplacement, d'audit et de suppression pour l'ACS de test DR.

Conditions requises pour l'hôte de test DR

Les conditions requises pour l'hôte de test DR lors d'un test de récupération après sinistre sont les suivantes :

- Les ACS qui ne sont pas des ACS de test DR ne peuvent pas être basculés en ligne.
- Les CAP doivent rester en mode manuel.
- *FLOAT(OFF)* et *EJCTAUTO(OFF)* sont automatiquement appliqués et les autres valeurs ne sont pas valides dans la commande MNTD.
- Vous ne pouvez pas exécuter d'utilitaires de rejet, de déplacement, d'audit et de suppression pour l'ACS de test DR.
- Les mises à jour de volumes provisoires sont autorisées uniquement pour les volumes définis à l'aide de *POOLPARAM/VOLPARAM* en tant que volumes de sous-pool provisoire de test DR.

Ressources VLE

Définissez des ressources VLE à l'aide du paramètre *STORMNGR* des commandes *DRTEST PRIMEPRD* et *CREATE*. En général, les ressources de test DR sont des ACS ou des VLE, bien qu'il n'existe aucune restriction concernant l'utilisation de ces deux types de ressources.

Comme les MVC physiques, les VMVC VLE sont définies à l'aide de la fonctionnalité *POOLPARAM/VOLPARAM*, avec une plage de VMVC réservée au test DR. Le test DR dispose

également d'un accès en lecture aux VMVC de production, comme indiqué dans l'exemple suivant.

```
//ACTMVCG1 EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
/* NOTE: MVCMAINT READONLY(ON) STATEMENTS
//SLUSMVON DD DSN=h1q.SLUSMVON, DISP=(NEW, CATLG, DELETE),
// SPACE=(CYL, 1)
/* NOTE: MVCMAINT READONLY(OFF) STATEMENTS
//SLUSMVOF DD DSN=h1q.SLUSMVOF, DISP=(NEW, CATLG, DELETE),
SPACE=(CYL, 1)
/* NOTE: THE FOLLOWING STEP SELECTS ALL "ACTIVE" MVCS
/* IN VLE1 AND MVCPPOOL MVCP1.
//SLSIN DD *
ACTMVCGN STORMNGR=VLE1, MVCP=MVCP1
/*
//ACTMVCG2 EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
/* NOTE: EXEC MVCMAINT TO SET READONLY(ON)
//SLSIN DD DSN=h1q.SLUSMVON, DISP=SHR
```

Stratégies VTCS

Pour réduire les différences entre vos environnements de production et de test DR, CDRT vous permet de définir des stratégies de gestion pour les VTV qui portent le même nom dans les environnements de production et de test DR, mais qui possèdent des définitions de stratégie différentes.

Notez que les sites de production et de test peuvent partager un VTSS à l'aide du paramètre *DRTEST DRVTSS SHARE*. Notez également que l'indication d'un ACS de récupération après sinistre factice n'est plus requise pour un VTSS partagé ou un test DR utilisant une VLE. La spécification d'un VTSS partagé est soumise aux restrictions suivantes :

- Le VTSS de test DR partagé ne doit pas contenir des connexions RTD actives à partir du site de production ou de test DR.
- Le CDS doit contenir des définitions *VOLPARM* pour configurer les sous-pools provisoires de test DR.
- Les VTV de production (ceux qui ne se trouvent pas dans un sous-pool de test DR) sont définis en lecture seule lors de leur montage et ne peuvent pas être modifiés.

Définition de MGMTCLAS/STORCLAS pour des VTSS non partagés

Lorsque vous définissez des classes de gestion pour un système de test DR, vous ne définissez en général qu'une seule copie MVC des VTV de sortie. Pour permettre un nettoyage VTSS après le test, il est conseillé de spécifier *DELSCR(YES)* comme indiqué dans l'exemple suivant.

```
STOR NAME(LOCAL) ACS(01) MVCPool(MVCP1)
MGMT NAME(CRITICAL) MIGPOL(LOCAL) IMMWAIT(0) DELSCR(YES)
```

Définition de MGMTCLAS pour des VTSS partagés

Lorsque vous utilisez un VTSS partagé pour le test DR, aucune copie de sortie migrée n'est autorisée. Vous devez spécifier *DELSCR(YES)* pour autoriser le nettoyage après le test, comme indiqué dans l'exemple suivant.

```
MGMT NAME(CRITICAL) DELSCR(YES)
```

Optimisation de l'accès aux ressources de test et de production

Lors d'un test de récupération après sinistre, il est recommandé d'appliquer des procédures permettant d'optimiser l'accès aux ressources dans les environnements de test et de production. Plus précisément :

- Avant de démarrer le test DR, définissez des classes de gestion de production qui spécifient une migration immédiate vers les ACS de production et de test DR, afin que les VTV soient disponibles sur des MVC accessibles par le système de test DR et le système de production.
- Définissez des classes de gestion de test DR qui spécifient une copie de migration unique, étant donné que le site de test DR ne peut généralement accéder qu'à un seul ACS.
- Utilisez la fonctionnalité *POOLPARAM/VOLPARAM* pour séparer les sous-pools provisoires et les pools MVC entre la production et le test DR.
- Si possible, vérifiez que le traitement de votre test DR ne met pas à jour des VTV préexistants (*DISP=MOD* ou ne les remplace pas à l'aide de *DISP=OLD*).
- Pour réduire le conflit entre les travaux de production qui effectuent une migration vers l'ACS de test DR et les travaux de test DR qui accèdent aux VTV sur des MVC dans l'ACS de test DR, lancez l'utilitaire *ACTMVCGN* pour marquer les MVC actives en tant que MVC en lecture seule dans l'environnement de production.
- Désactivez la récupération d'espace MVC (à l'aide de *CONFIG HOST NORECLAM*) sur les MVC de production lors d'un test DR, afin de préserver le contenu MVC des volumes utilisés par le système de test DR.

Exécution d'un test DR

Remarque :

Pour plus d'informations sur les commandes et utilitaires employés dans cette procédure, reportez-vous au manuel *Référence des commandes, des instructions de contrôle et des utilitaires ELS*.

Pour exécuter un test DR :

1. Définissez des pools de volume dans le CDS de production à l'aide de la commande *SET VOLPARM* et des instructions *POOLPARM/VOLPARM* figurant dans *SLSPARM DD*.

Pour plus d'informations, voir " [Ressources de volume](#)."

2. Vérifiez que vos ressources de test sont définies correctement.

Pour plus d'informations, reportez-vous aux sections suivantes :

- " [Ressources ACS](#) "
- " [Ressources VTSS](#) "
- " [Ressources VLE](#) "

3. Créez des instructions *MGMTCLAS/STORCLAS* pour l'environnement DRTEST.

Pour plus d'informations, reportez-vous aux sections suivantes :

- " [Définition de MGMTCLAS/STORCLAS pour des VTSS non partagés](#) "
- " [Définition de MGMTCLAS pour des VTSS partagés](#) "

4. Copiez le catalogue MVS pour le site de test DR, le cas échéant.
5. Vous pouvez éventuellement copier la base de données TMS (si un système TMS est utilisé) pour le site de test DR.
6. Sur le site de production, exécutez l'utilitaire DRTEST (avec le mot-clé *PRIMEprd*) pour préparer le CDS de production pour le test DR.

Par exemple, consultez la section " [Exemple de JCL pour le scénario 1](#)."

Il vous suffit d'exécuter *PRIMEprd* **une fois** dans votre environnement, quel que soit le nombre d'itérations de DRTEST, **sauf** en cas de modification de votre configuration. Si votre configuration de test DR change, vous devez réexécuter *PRIMEprd*. De même, vous n'avez pas besoin d'exécuter l'utilitaire *DRTEST RESET* une fois que le test DR est terminé. Bien que les indicateurs restent définis dans le CDS de production, ils n'ont aucune incidence sur le traitement tant que le test DR n'est pas actif.

7. Sur le système de production, utilisez la commande HSC *CAPPREF* pour définir tous les CAP situés dans l'ACS de test DR sur le mode manuel.
8. Sur le site de test DR, exécutez l'utilitaire DRTEST (avec le mot-clé *CREATE*) sur la copie mise en miroir ou la copie de sauvegarde du CDS de production, afin de créer le nouveau CDS pour le test DR.

Chaque scénario fournit un exemple de *DRTEST CREATE*.

Les CDS doivent être alloués à l'aide des instructions DD sur l'utilitaire. Lorsque *NOUPD* est utilisé, seule l'instruction *SLSCNTL DD* est requise. Il peut s'agir du CDS principal réel, d'une sauvegarde ou d'une copie mise en miroir.

9. Démarrez le test DR sur le site de production, en pointant vers les définitions DRTEST *MGMTCLAS/STORCLAS* que vous avez créées à l'étape 3.

Par exemple :

```
/PRIME EXEC PGM=SLUADMIN, PARM='MIXED'  
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR  
//SLSIN DD *  
DRTEST START
```

Remarque :

Vous pouvez également démarrer le test en entrant une commande *DRTEST START* à partir de la console.

10. Démarrez le système SMC sur le ou les hôtes client DRTEST.
11. Démarrez le ou les systèmes SMC/HSC/VTCS (en pointant vers le CDS que vous avez créé à l'étape 8) sur les systèmes de test DR.
12. Vérifiez que les VTD des chemins et VTSS de test sont en ligne.
13. Basculez les VTSS de récupération après sinistre en ligne sur le système DR.
14. Basculez les RTD de récupération après sinistre en ligne sur le système DR, le cas échéant.
15. Exécutez les tests sur le site de test DR.

Lors du test DR, les conditions suivantes sont appliquées par programmation :

- Le ou les ACS du site de production sont déconnectés du ou des hôtes de test DR.
- Les VTSS du site de production sont hors ligne sur le ou les hôtes de test DR.
- Aucune opération flottante de démontage, d'éjection, de déplacement, de mise à jour des volumes provisoires, d'audit ou de redistribution des volumes provisoires ne peut être effectuée sur le site de test DR.
- Aucune opération flottante de démontage, d'entrée/éjection, de déplacement, d'audit ou de redistribution des volumes provisoires sur l'ACS de test DR ne peut être effectuée sur le site de production.
- Tous les CAP de l'ACS de test DR sont en mode manuel.

Remarque :

Vous pouvez entrer des volumes dans l'ACS de test DR, mais une fois que le test est terminé, vous devez les éjecter ou auditer les cellules pour synchroniser le CDS de production avec les volumes de bibliothèque réels.

Nettoyage après un test DR

Comme décrit au début de ce chapitre, il est essentiel que les travaux figurant dans un flux de travaux de test DR ne mettent pas à jour les volumes créés en production.

Remarque :

Pour obtenir des informations sur la commande *DRTEST* et l'utilitaire *DRTEST*, reportez-vous au manuel *Référence des commandes, des instructions de contrôle et des utilitaires ELS*. Pour obtenir des informations sur les messages CDRT, voir *Messages et codes ELS*.

Nettoyage après un test DR

Remarque :

Pour ELS 7.1 et versions ultérieures, si le module CDRT Cleanup Enhancement SPE est installé, vous n'avez pas besoin d'exécuter cette procédure immédiatement après le test DR et avant la reprise de l'environnement de production normal. A la place, vous pouvez l'exécuter sans interruption (par exemple, avant le test DR suivant).

1. Exécutez l'utilitaire *SLUADMIN SCRAtch* pour supprimer tous les VTV possibles dans vos sous-pools DRTEST.

Etant donné que vous définissez *DELSCR(YES)* dans votre classe de gestion, les VTV sont automatiquement supprimés du tampon lorsque vous les supprimez à la fin du test.

AVERTISSEMENT :

Si vous n'utilisez pas SET VOLPARM et que vous ne configurez pas de pools provisoires distincts, vous risquez de perdre des données.

2. Si votre test DR a modifié des VTV de production, vous devez également effectuer l'opération suivante pour vous assurer qu'aucune donnée de test DR ne demeure sur le VTSS de production :
 - Exécutez un rapport VTV sur le CDS DRTEST et examinez la sortie pour déterminer si des VTV figurant dans les plages de VTV de production ont été modifiés lors du test.

Notez que *VTVRPT COPIES* signale désormais les copies VTV qui sont des copies de test DR, à l'aide de l'indicateur "D" dans la colonne DRT.

- Si des VTV ont été modifiés, vous devez effectuer l'une des opérations suivantes :
 - Demandez à migrer les VTV modifiés en vous basant sur le rapport VTV.
 - Migrez le VTSS de test DR vers 0.
 - Demandez au CSE de "nettoyer" le VTSS de test.
3. Arrêtez HSC/VTCS et SMC sur le système MVS de test DR.
 4. Si l'utilitaire *ACTMVCGN* a été exécuté avant le test pour faire passer les MVC à l'état *READONLY* (lecture seule), exécutez *SLUADMIN* en utilisant la sortie de l'instruction *SLUSMVOF DD* comme entrée afin de restaurer l'état *READONLY*.

Reprise des opérations normales

Suivez cette procédure pour redémarrer les opérations et arrêter le test DR.

1. Arrêtez le test DR sur le système MVS *PRODUCTION*.

Par exemple :

```
/STOP EXEC PGM=SLUADMIN, PARM='MIXED'  
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
```

```
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
DRTEST STOP
```

De même, vous n'avez pas besoin d'exécuter l'utilitaire *DRTEST RESET* une fois que le test DR est terminé. Bien que les indicateurs restent définis dans le CDS de production, ils n'ont aucune incidence sur le traitement tant que le test DR n'est pas actif.

2. Si vous le souhaitez, placez les CAP en mode automatique.

Scénarios opérationnels

Cette section indique comment utiliser le logiciel de test DR pour configurer l'environnement pour le démarrage et l'arrêt des tests DR. Cette section contient les informations suivantes :

- " Scénario 1 : sites de production et de test, ACS sur chaque site, VTSS de rechange sur le site de test "
- " Scénario 2 : sites de production et de test, ACS sur chaque site, reprise VTSS sur le site de test "
- " Scénario 3 : sites de production et de test, ACS sur chaque site, aucun VTSS "
- " Scénario 4 : VTSS en cluster avec sites de production et de test DR "
- " Scénario 5 : sites de production et de test, ACS et VLE sur chaque site "
- " Scénario 6 : sites de production et de test, VLE uniquement sur chaque site "
- " Scénario 7 : VTSS en cluster (sans bandes) avec sites de production et de test DR "

Pour obtenir des informations sur la commande DRTEST et l'utilitaire DRTEST, reportez-vous au manuel *Référence des commandes, des instructions de contrôle et des utilitaires ELS*. Pour obtenir des informations sur les messages CDRT, voir *Messages et codes ELS*.

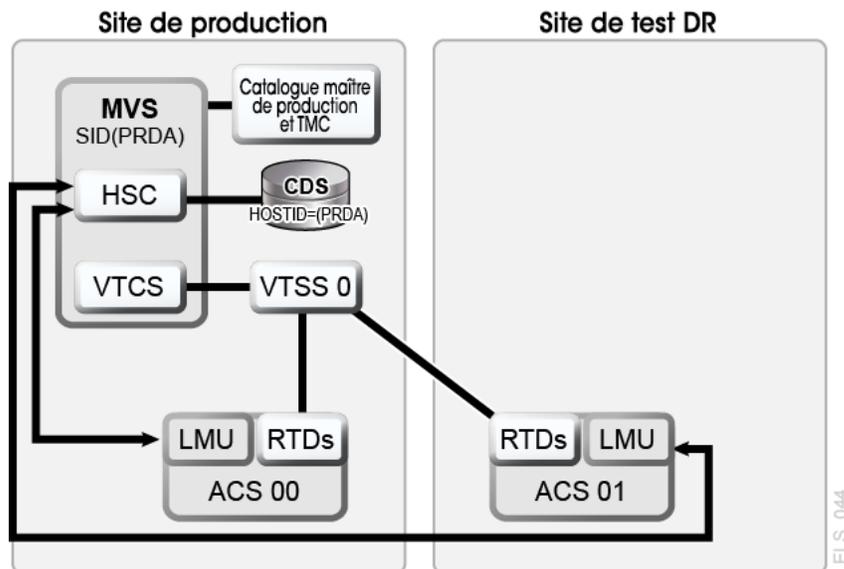
Remarque :

Pour tous les scénarios, exécutez la procédure indiquée dans la section " [Nettoyage après un test DR](#)" après le test.

Scénario 1 : sites de production et de test, ACS sur chaque site, VTSS de rechange sur le site de test

Dans le scénario 1, il y a un seul ACS, à la fois sur les sites de production et de test, et un ou plusieurs VTSS "de rechange" sur le site de test utilisé exclusivement pour les tests (il n'est pas nécessaire de migrer ou restaurer le contenu des VTSS "de rechange"). Dans des opérations normales, le site de production écrit les VTV sur les VTSS et y accède sur le site de production. Les VTV de sortie sont toujours migrés immédiatement et mis en duplex sur des MVC distinctes, une dans chaque ACS. La [Figure 7.1, « Configuration de VTSS de rechange - avant l'exécution de l'utilitaire DRTEST »](#) présente le système dans le scénario 1 avant l'exécution de l'utilitaire DRTEST.

Figure 7.1. Configuration de VTSS de rechange - avant l'exécution de l'utilitaire DRTEST



Exemple de JCL pour le scénario 1

Etape PRIMEPRD :

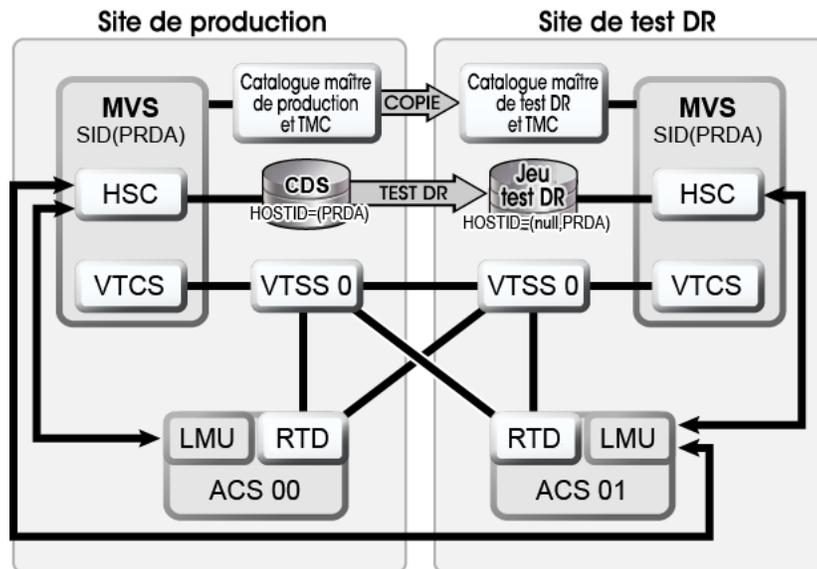
```
//DRTPRIME EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
DRTEST PRIMEPRD +
DRACS(01) DRVTSS(VTSS0) SPARE HOST(MVS1, MVS2)
```

Etape CREATE :

```
//DRTCREAT EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSNEW1 DD DSN=h1q.DRTEST.SLSCNTL, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW2 DD DSN=h1q.DRTEST.SLSCNTL2, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW3 DD DSN=h1q.DRTEST.SLSSTBY, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSIN DD *
DRTEST CREATE NOUPDRD +
DRACS(01) DRVTSS(VTSS0) SPARE HOST(MVS1, MVS2)
```

La [Figure 7.2, « Configuration de VTSS de rechange - après l'exécution de l'utilitaire DRTEST »](#) présente le système dans le scénario 1 (VTSS de rechange sur le site de test) après l'exécution de l'utilitaire *DRTEST*.

Figure 7.2. Configuration de VTSS de rechange - après l'exécution de l'utilitaire DRTEST

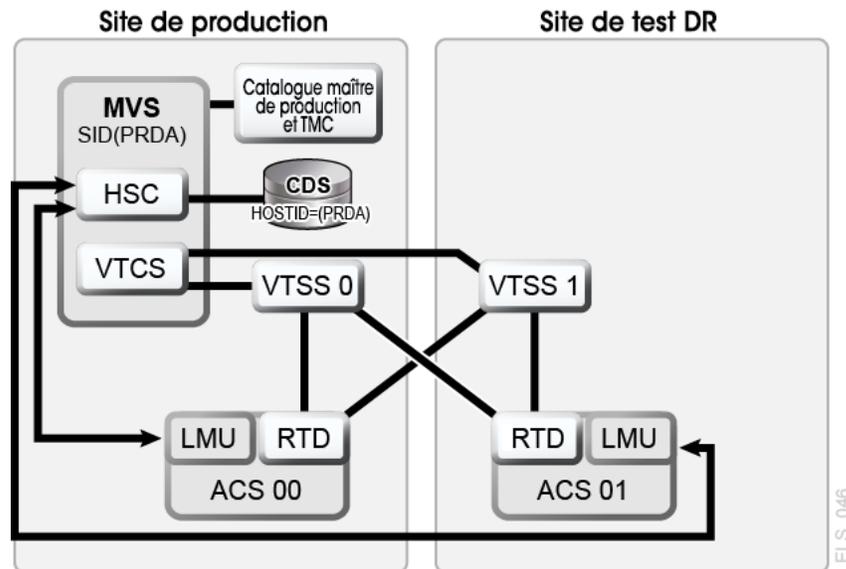


Scénario 2 : sites de production et de test, ACS sur chaque site, reprise VTSS sur le site de test

Dans le scénario 2, il y a un seul ACS, à la fois sur les sites de production et de test, mais aucun VTSS "de rechange" sur le site de test utilisé pour les tests. Dans des opérations normales, le site de production écrit les VTV sur les VTSS et y accède sur les deux sites. Les VTV de sortie sont toujours migrés immédiatement et mis en duplex sur des MVC distinctes, une dans chaque ACS. Dans cette configuration, vous devez demander la migration vers zéro d'un ou plusieurs VTSS sur le site de test et basculer ces VTSS hors ligne sur le système de production, afin que les tests puissent reprendre les ressources VTSS requises. En outre, une ou plusieurs partitions LPAR sur le site de test fonctionnent comme des systèmes de production déplacés, en s'exécutant parallèlement aux systèmes de production réels. Les deux ACS sont en ligne sur le système de production.

La [Figure 7.3, « Configuration de la reprise VTSS - avant l'exécution de l'utilitaire DRTEST »](#) présente le système dans le scénario 2 (reprise VTSS sur le site de test) avant l'exécution de l'utilitaire *DRTEST*.

Figure 7.3. Configuration de la reprise VTSS - avant l'exécution de l'utilitaire DRTEST



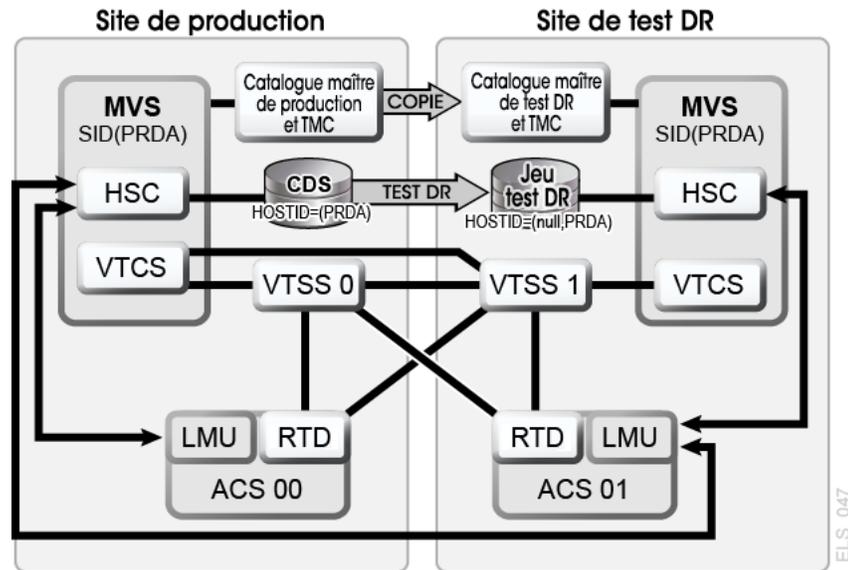
Exemple de JCL pour le scénario 2

Étape CREATE uniquement, étape PRIMEPRD exécutée précédemment :

```
//DRTCREAT EXEC PGM=SLUADMIN, PARM=' MIXED '
//STEPLIB DD DSN=h1q. SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSNEW1 DD DSN=h1q. DRTEST. SLSCNTL, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW2 DD DSN=h1q. DRTEST. SLSCNTL2, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW3 DD DSN=h1q. DRTEST. SLSSTBY, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSIN DD *
DRTEST CREATE NOUPDPRD +
DRACS(01) DRVTSS(VTSS1) HOST(MVS1, MVS2)
```

La [Figure 7.4, « Configuration de la reprise VTSS - après l'exécution de l'utilitaire DRTEST »](#) présente le système dans le scénario 2 (reprise VTSS sur le site de test) après l'exécution de l'utilitaire *DRTEST*.

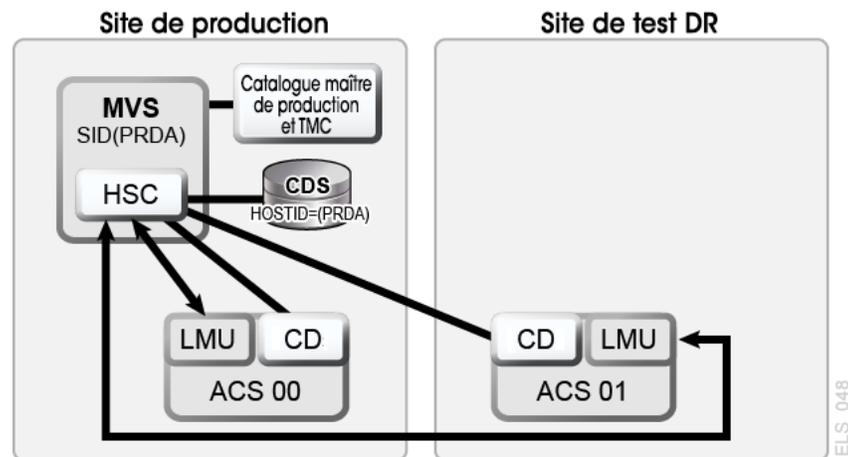
Figure 7.4. Configuration de la reprise VTSS - après l'exécution de l'utilitaire DRTEST



Scénario 3 : sites de production et de test, ACS sur chaque site, aucun VTSS

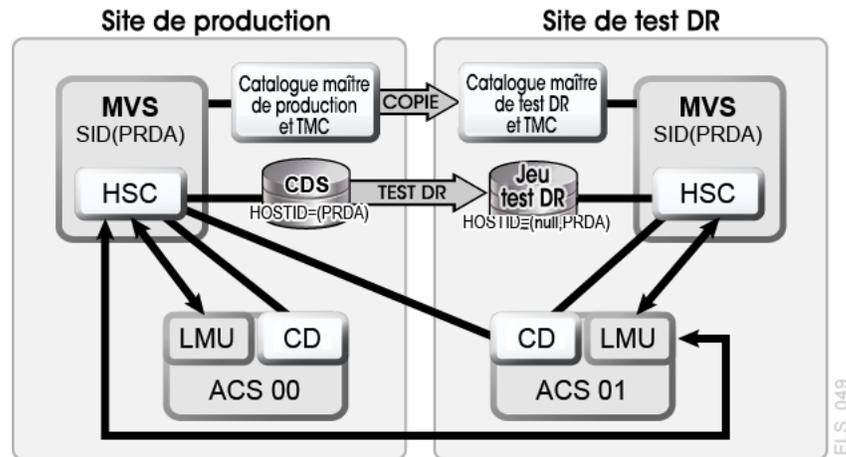
Dans le scénario 3, il y a un seul ACS, à la fois sur les sites de production et de test, mais aucun VTSS sur le site de test utilisé pour les tests. Dans des opérations normales, le site de production écrit les jeux de données de bande et y accède sur les deux sites. La [Figure 7.5, « Configuration en lecture seule - avant l'exécution de l'utilitaire DRTEST »](#) présente le système dans le scénario 3 (configuration en lecture seule) avant l'exécution de l'utilitaire DRTEST.

Figure 7.5. Configuration en lecture seule - avant l'exécution de l'utilitaire DRTEST



La Figure 7.6, « Configuration en lecture seule - après l'exécution de l'utilitaire DRTEST » présente le système dans le scénario 3 (configuration en lecture seule) après l'exécution de l'utilitaire *DRTEST*.

Figure 7.6. Configuration en lecture seule - après l'exécution de l'utilitaire DRTEST



Exemple de JCL pour le scénario 3

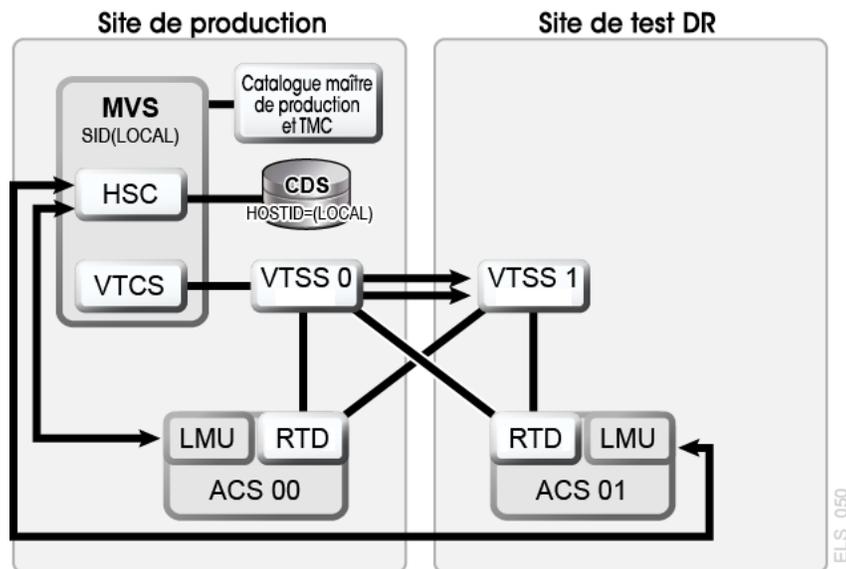
Etape CREATE uniquement, étape PRIMEPRD exécutée précédemment :

```
//DRTCREAT EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSNEW1 DD DSN=h1q.DRTEST.SLSCNTL, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW2 DD DSN=h1q.DRTEST.SLSCNTL2, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW3 DD DSN=h1q.DRTEST.SLSSTBY, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSIN DD *
DRTEST CREATE NOUPDRD +
DRACS(01) HOST(MVS1, MVS2)
```

Scénario 4 : VTSS en cluster avec sites de production et de test DR

Comme indiqué à la Figure 7.7, « Configuration VTSS en cluster principal/secondaire - opérations normales », dans des opérations normales, le scénario 4 est une configuration VTSS en cluster utilisée pour la récupération après sinistre, avec des sites de production et de test DR interconnectés avec les ACS de production et de test DR. VTSS0 est le VTSS principal sur le site de production et VTSS1 est le VTSS secondaire sur le site de test DR.

Figure 7.7. Configuration VTSS en cluster principal/secondaire - opérations normales



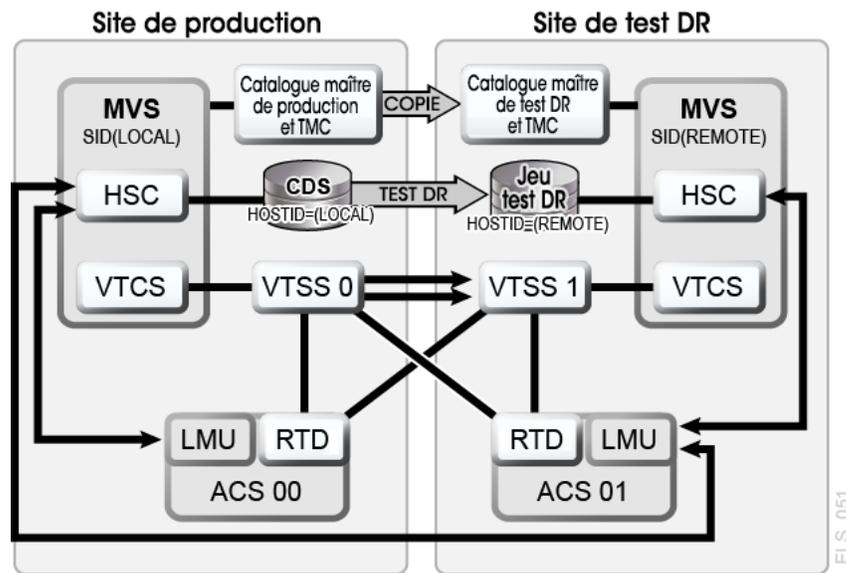
Exemple de JCL pour le scénario 4

Etape CREATE uniquement, étape PRIMEPRD exécutée précédemment :

```
//DRTCREAT EXEC PGM=SLUADMIN, PARM=' MIXED '
//STEPLIB DD DSN=h1q. SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSNEW1 DD DSN=h1q. DRTEST. SLSCNTL, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW2 DD DSN=h1q. DRTEST. SLSCNTL2, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW3 DD DSN=h1q. DRTEST. SLSSTBY, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSIN DD *
DRTEST CREATE NOUPDRD +
DRACS(01) DRVTSS(VTSS1) SHARE HOST(MVS1, MVS2)
```

Et si vous utilisiez le site de test DR pour les tests ? La [Figure 7.8, « Configuration VTSS en cluster principal/secondaire - lors du test »](#) présente le système dans le scénario 4 lors du test DR.

Figure 7.8. Configuration VTSS en cluster principal/secondaire - lors du test

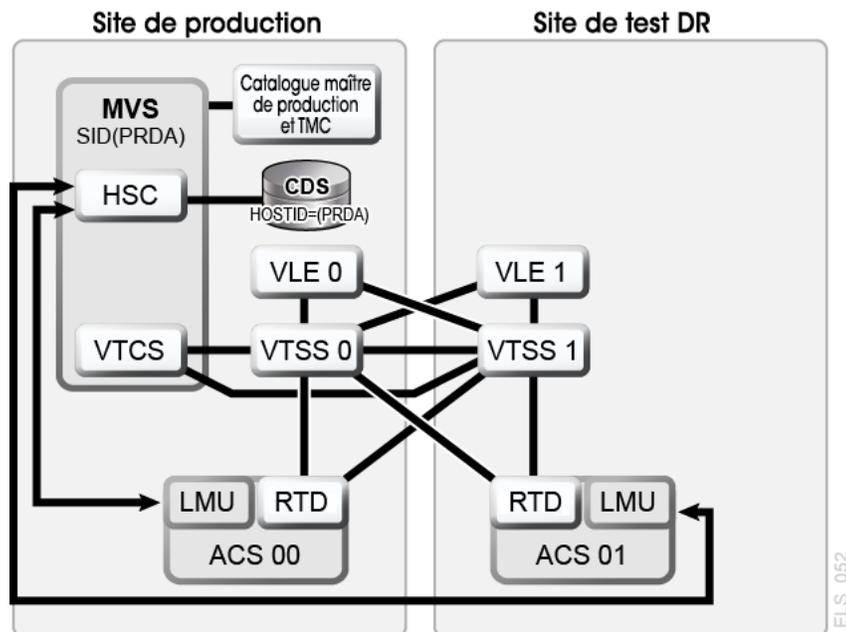


Scénario 5 : sites de production et de test, ACS et VLE sur chaque site

Dans le scénario 5, il y a un seul ACS, à la fois sur les sites de production et de test, mais aucun VTSS "de rechange" sur le site de test utilisé pour les tests. Dans des opérations normales, le site de production écrit les VTV sur les VTSS et y accède sur les deux sites. Les VTV de sortie sont toujours migrés immédiatement et mis en duplex, une copie sur une MVC dans l'ACS, une autre sur une VMVC dans la VLE. Dans cette configuration, vous devez demander la migration vers zéro d'un ou plusieurs VTSS sur le site de test et basculer ces VTSS hors ligne sur le système de production, afin que les tests puissent reprendre les ressources VTSS requises. En outre, une ou plusieurs partitions LPAR sur le site de test fonctionnent comme des systèmes de production déplacés, en s'exécutant parallèlement aux systèmes de production réels. **Les ACS et les VLE sont en ligne sur le système de production.**

La [Figure 7.9, « Configuration VLE et ACS - avant l'exécution de l'utilitaire DRTEST »](#) présente le système dans le scénario 5 avant l'exécution de l'utilitaire *DRTEST*.

Figure 7.9. Configuration VLE et ACS - avant l'exécution de l'utilitaire DRTEST



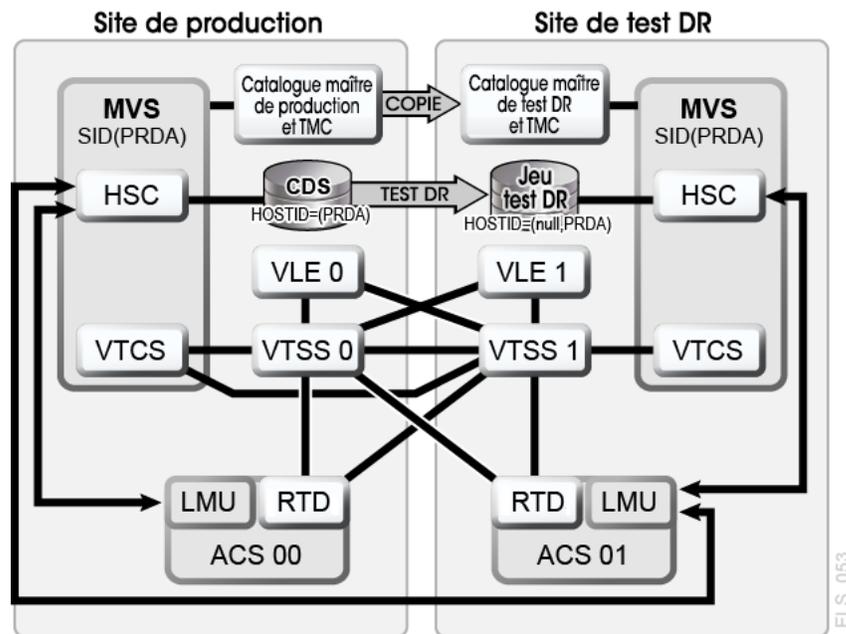
Exemple de JCL pour le scénario 5

Étape CREATE uniquement, étape PRIMEPRD exécutée précédemment :

```
//DRTCREAT EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSNEW1 DD DSN=h1q.DRTEST.SLSCNTL, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW2 DD DSN=h1q.DRTEST.SLSCNTL2, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW3 DD DSN=h1q.DRTEST.SLSSTBY, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSIN DD *
DRTEST CREATE NOUPDPRD +
DRACS(01) DRVTSS(VTSS1) HOST(MVS1, MVS2) STORMNGR(VLE1)
```

La [Figure 7.10](#), « Configuration VLE et ACS - après l'exécution de l'utilitaire DRTEST » présente le système dans le scénario 5 après l'exécution de l'utilitaire *DRTEST*.

Figure 7.10. Configuration VLE et ACS - après l'exécution de l'utilitaire DRTEST



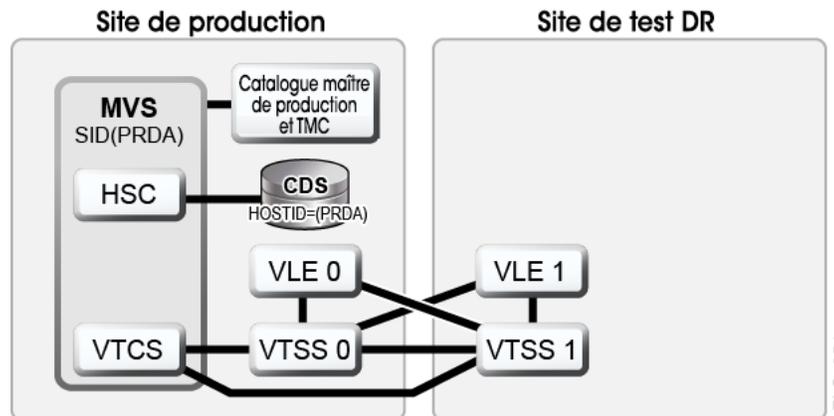
Scénario 6 : sites de production et de test, VLE uniquement sur chaque site

Dans le scénario 6, il y a un seul VTSS connecté à une VLE sur chaque site. Le VTSS sur le site de test **n'est pas** un VTSS de rechange et est utilisé par le site de production dans des opérations normales. Les VTV de sortie sont toujours migrés immédiatement et mis en duplex sur des VMVC distinctes, une dans chaque VLE.

Dans cette configuration, vous devez demander la migration vers zéro d'un ou plusieurs VTSS sur le site de test et basculer ces VTSS hors ligne sur le système de production, afin que les tests puissent reprendre les ressources VTSS requises. En outre, une ou plusieurs partitions LPAR sur le site de test fonctionnent comme des systèmes de production déplacés, en s'exécutant parallèlement aux systèmes de production réels. Les deux VLE sont en ligne sur le système de production.

La [Figure 7.11, « Configuration VLE uniquement - avant l'exécution de l'utilitaire DRTEST »](#) présente le système dans le scénario 6 avant l'exécution de l'utilitaire *DRTEST*.

Figure 7.11. Configuration VLE uniquement - avant l'exécution de l'utilitaire DRTEST



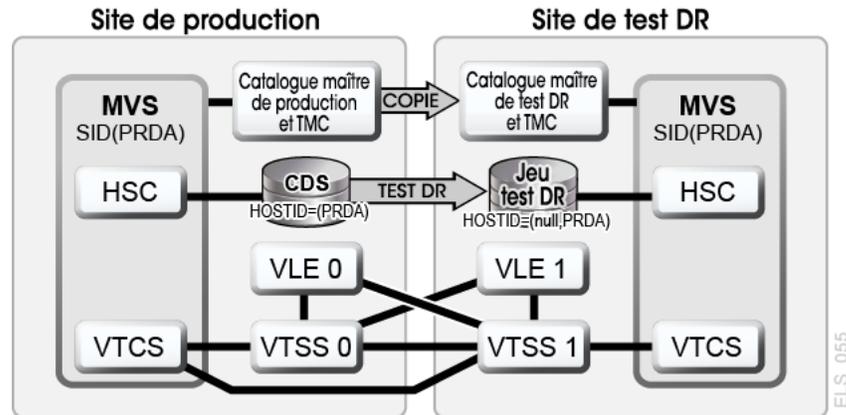
Exemple de JCL pour le scénario 6

Étape CREATE uniquement, étape PRIMEPRD exécutée précédemment :

```
//DRTCREAT EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSNEW1 DD DSN=h1q.DRTEST.SLSCNTL, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW2 DD DSN=h1q.DRTEST.SLSCNTL2, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSNEW3 DD DSN=h1q.DRTEST.SLSSTBY, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
//SLSIN DD *
DRTEST CREATE NOUPDPRD +
STORMNGR(VLE1) DRVTSS(VTSS1) HOST(MVS1, MVS2)
```

La [Figure 7.12](#), « Configuration VLE uniquement - après l'exécution de l'utilitaire DRTEST » présente le système dans le scénario 6 après l'exécution de l'utilitaire *DRTEST*.

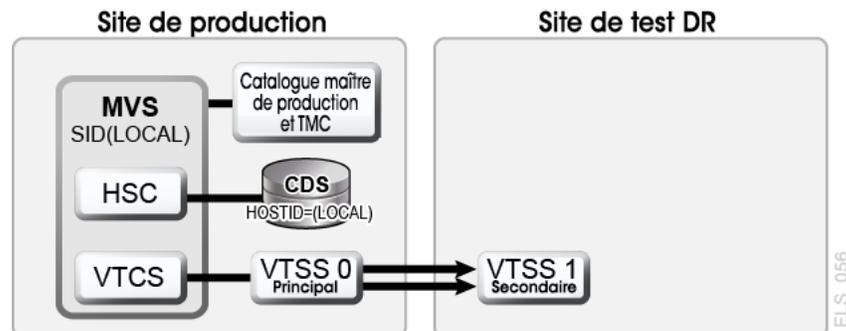
Figure 7.12. Configuration VLE uniquement - après l'exécution de l'utilitaire DRTEST



Scénario 7 : VTSS en cluster (sans bandes) avec sites de production et de test DR

Comme indiqué à la [Figure 7.13](#), « Configuration VTSS en cluster principal/secondaire sans bandes - avant l'exécution de l'utilitaire DRTEST », dans des opérations normales, le scénario 7 est une configuration VTSS en cluster (sans bandes) qui est utilisée pour la récupération après sinistre, avec des sites de production et de test DR. VTSS0 est le VTSS principal sur le site de production et VTSS1 est le VTSS secondaire sur le site de test DR.

Figure 7.13. Configuration VTSS en cluster principal/secondaire sans bandes - avant l'exécution de l'utilitaire DRTEST



Exemple de JCL pour le scénario 7

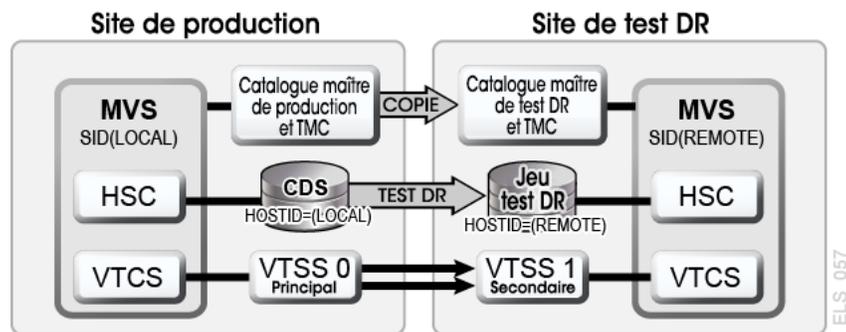
Etape CREATE uniquement, étape PRIMEPRD exécutée précédemment :

```
//DRTCREAT EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSNEW1 DD DSN=h1q.DRTEST.SLSCNTL, DISP=(NEW, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL, x)
```

```
//SLSNEW2 DD DSN=h1q.DRTEST.SLSCNTL2,DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,SPACE=(CYL,x)
//SLSNEW3 DD DSN=h1q.DRTEST.SLSSTBY,DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,SPACE=(CYL,x)
//SLSIN DD *
DRTEST CREATE NOUPDPRD +
DRVTTSS(VTSS1) SHARE HOST(MVS1,MVS2))
```

Et si vous utilisiez le site de test DR pour les tests ? La [Figure 7.14](#), « Configuration VTSS en cluster principal/secondaire sans bandes - lors du test » présente le système dans le scénario 7 lors du test DR.

Figure 7.14. Configuration VTSS en cluster principal/secondaire sans bandes - lors du test



Chapitre 8. Création de points de récupération système dans des environnements VSM

Comme décrit dans la section " [Définition de l'objectif du point de récupération \(RPO\)](#)", l'une des clés du succès d'une solution de récupération après sinistre réside dans sa capacité à définir des points de reprise système qui garantissent l'utilisation d'un jeu de données cohérent comme valeur de référence de la récupération après sinistre (DR).

Pour les environnements VSM, dans une valeur de référence valide pour la récupération après sinistre :

- toutes les données critiques de l'entreprise sont sécurisées dans l'emplacement DR indiqué ;
- une copie sécurisée des métadonnées (CDS, catalogue MVS, TMC) a été capturée ;
- la validité de la copie des métadonnées est garantie en cas de déclaration de sinistre (réel ou de test).

VTCS permet de créer une valeur de référence DR grâce aux fonctions suivantes :

- L'utilitaire *DRMONitr* surveille les données critiques de récupération après sinistre et vérifie qu'elles atteignent l'emplacement de récupération désigné. Il permet de bloquer le traitement d'un flux de travaux en attendant que les données arrivent à destination. Une fois que toutes les données ont atteint leur but, l'utilitaire s'arrête. L'utilitaire *DRMONitr* peut s'exécuter en tant qu'étape d'un travail. Une fois que l'étape du travail est terminée, vous êtes sûr que toutes les données surveillées ont atteint leur destination et qu'elles sont sécurisées dans l'emplacement DR indiqué.
- L'utilitaire *DRCHKPT* permet de vérifier que les données auxquelles les métadonnées CDS ont accédé restent valides pendant une période donnée. Cela garantit la validité d'une sauvegarde CDS pendant une période déterminée et vous permet donc de restaurer une valeur de référence DR pour un système VSM. L'utilitaire *DRCHKPT* définit un horodatage dans le CDS actif, qui définit le point de récupération à partir duquel le contenu MVC peut être récupéré. A partir de ce point de récupération, la validité du contenu des données sera garantie pendant un certain temps. Sans l'utilitaire *DRCHKPT*, une sauvegarde CDS ne peut pas être utilisée pour restaurer une valeur de référence DR, car les éléments figurant dans le CDS (position du VTV sur une MVC) risquent de ne plus être valides.

Pour plus d'informations, reportez-vous au manuel *Référence des commandes, des instructions de contrôle et des utilitaires ELS*.

En outre, notez les points suivants :

- Pour les VMVC, *MVCDRAIN* avec le paramètre *EJECT* supprime physiquement les VTV.

Attention :

Si vous vous servez de l'utilitaire *DRCHKPT* et/ou du paramètre *CONFIG GLOBAL PROTECT* pour protéger le contenu de la sauvegarde CDS pour les VMVC, la saisie de *MVCDR EJECT* invalide le contenu VMVC de la sauvegarde CDS.

- Pour les VMVC et les MVC, l'utilitaire *MVCDRAIN* sans le paramètre *EJECT* ne supprime pas les VTV, mais met à jour l'enregistrement CDS de manière à n'afficher aucun VTV sur les VMVC/MVC.

Pour plus d'informations, reportez-vous au manuel *Référence des commandes, des instructions de contrôle et des utilitaires ELS*.

Exemples d'utilisation de points de reprise

Les exemples suivants sont étudiés :

- " [Exemple 1 : copies MVC locales et copies MVC distantes](#) "
- " [Exemple 2 : utilisation de CONFIG RECLAIM PROTECT](#) "

Exemple 1 : copies MVC locales et copies MVC distantes

Dans cet exemple :

- Les utilitaires *DRMONitr* et *DRCHKPT* vérifient que les données de récupération après sinistre ont atteint leur emplacement de récupération et que les métadonnées associées (sauvegarde CDS) extraient les données du VTV, si nécessaire.
- Un site local est présenté avec un VTSS plus un ACS (ACS 00), tandis qu'un site distant contient uniquement un ACS (ACS 01), comme indiqué à la [Figure 8.1, « Exemple de points de récupération système VSM \(locaux et distants\) »](#).

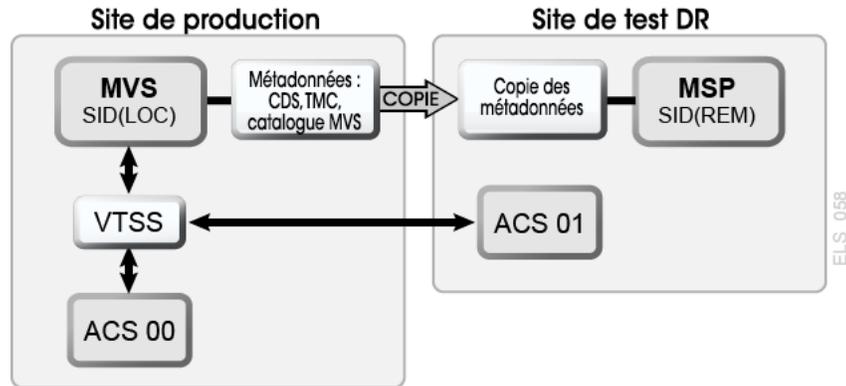
L'exemple illustre une stratégie de récupération après sinistre simple, où des copies des données critiques sont sécurisées quotidiennement sur le site distant avec les métadonnées. Les copies VTV distantes sont les copies DR indiquées.

Une fois les travaux de production terminés, un travail est planifié.

- Il surveille la fin de l'exécution des copies distantes (*DRMONitr*).
- Il définit des points de reprise pour le CDS (*DRCHKPT*).
- Il sauvegarde les métadonnées (CDS, TMC, catalogue MVS) et les sécurise sur le site distant. **Notez que** les sauvegardes des métadonnées étant essentielles pour la récupération après sinistre, elles sont placées dans un emplacement "bien connu" ou sécurisé.

Un point de reprise DR synchronisé est alors créé quotidiennement. Si une récupération après sinistre est déclarée, l'environnement de bande est restauré au point de reprise et les travaux sont réexécutés à partir de cet état connu.

Figure 8.1. Exemple de points de récupération système VSM (locaux et distants)



Pour exécuter cet exemple à l'aide de la configuration illustrée à la [Figure 8.1, « Exemple de points de récupération système VSM \(locaux et distants\) »](#) :

1. Créez les instructions de stratégie ci-dessous.

```
MGMT NAME(DR)  MIGPOL(LOCAL,REMOTE) IMMDELAY(0)
STOR NAME(LOCAL) ACS(00)
STOR NAME(REMOTE) ACS(01)
```

Remarque :

Pour disposer d'un environnement de récupération après sinistre efficace, vous pouvez également utiliser les instructions *MIGRSEL* et *MIGRVTV*, qui servent à vérifier que les copies DR sont sécurisées dès que possible.

2. Pour vous assurer que les données critiques sont sécurisées dans l'emplacement distant, exécutez l'étape de travail *DRMONitr* de l'exemple suivant.

```
//MONITOR EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SYSIN DD UNIT=SYSDA, SPACE=(TRK, 1)
//*
//SYSPRINT DD SYSOUT=*
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
DRMON MGMT(DR) STOR(REMOTE) MAXAGE(24) TIMEOUT(30)
```

Dans cet exemple, l'utilitaire *DRMONitr* attend que toutes les copies VTV dont la récupération après sinistre de la classe de gestion est inférieure à 24 heures soient

envoyées à l'ACS distant. L'utilitaire est défini de sorte à abandonner l'opération si la durée d'exécution (ou le délai d'attente) dépasse 30 minutes.

- Une fois que toutes les VTV ont été envoyées à l'ACS distant (affichage du code retour zéro), *DRCHKPT* s'exécute pour définir le point de reprise, comme indiqué dans l'exemple suivant.

```
//CHKPT EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SYSPRINT DD SYSOUT=*
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
DRCHKPT SET
```

Dans cet exemple, l'utilitaire *DRCHKPT* définit un horodatage, ou point de récupération, dans le CDS actif. A partir de ce point de récupération, la validité du contenu de la copie MVC sera garantie pendant un certain temps (par exemple jusqu'à l'exécution d'un autre utilitaire *CHKPT*).

- Lorsque le point de récupération est défini dans le CDS actif, une sauvegarde CDS doit être effectuée immédiatement, comme indiqué dans l'exemple suivant.

```
//CHKPT EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SYSIN DD UNIT=SYSDA, SPACE=(TRK, 1)
// *
//SLSCNTL DD DSN=h1q.DBSEPRM, DISP=SHR
//SLSBKUP DD DSN=h1q.DBSEPRM.BKUP, DISP=SHR
//SYSPRINT DD SYSOUT=*
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
BACKUP OPTION(COPY)
```

Une fois la sauvegarde effectuée, la validité du contenu MVC, ou des métadonnées, sera garantie pendant un certain temps (jusqu'à la définition du point de récupération ou de reprise suivant).

Cette procédure est maintenant terminée. Une récupération après sinistre peut être déclarée (le site de production local n'est plus disponible) pour l'une des raisons suivantes :

- Les MVC et toutes les autres données critiques (copies de métadonnées, par exemple) sont transportées vers une autre installation, où une copie miroir du site local de production est disponible.

ou

- Une réplique du site local de production est créée sur l'emplacement distant.

Les métadonnées sont restaurées (CDS, TMC, catalogue MVS). Lors du redémarrage de l'environnement de bande, toutes les opérations peuvent continuer (réapplication des modifications) à partir du point de synchronisation DR.

Exemple 2 : utilisation de CONFIG RECLAIM PROTECT

Dans cet exemple, le CDS est sauvegardé toutes les 24 heures. Le contenu MVC, ou métadonnées CDS, qui se trouve dans la sauvegarde CDS doit rester valide jusqu'à la sauvegarde CDS suivante.

Cet exemple présente une protection MVC définie sur 28 heures. Pour plus d'informations sur le paramètre *CONFIG RECLAIM PROTECT*, reportez-vous au manuel *Référence des commandes, des instructions de contrôle et des utilitaires ELS*.

1. Définissez le paramètre *CONFIG GLOBAL PROTECT = 28*.
2. Le jour 1, sauvegardez le CDS.
 - Les MVC purgées/récupérées après cette sauvegarde ne peuvent pas être remplacées pendant 28 heures.
 - La sauvegarde CDS du jour 1 est désormais le point de récupération jusqu'à la prochaine sauvegarde CDS.
3. Le jour 2, sauvegardez le CDS.
 - Les MVC purgées/récupérées après cette sauvegarde ne peuvent pas être remplacées pendant 28 heures.
 - La sauvegarde CDS du jour 2 est désormais le point de récupération jusqu'à la prochaine sauvegarde CDS.

Chapitre 9. Utilisation de VLE pour la récupération après sinistre

L'utilisation de VLE (Virtual Library Extension) en tant que solution de récupération après sinistre propose une méthode sans interruption et simplifiée de tests DR, ainsi qu'une méthode de récupération après un événement d'interruption d'activité.

Le système gère VLE comme une bibliothèque (ACS). Toutefois, étant donné que la VLE utilise le stockage sur disque plutôt que sur bande et conserve un inventaire interne des VTV qu'elle contient, elle offre davantage de fonctionnalités qu'une vraie bibliothèque :

- La VLE est une solution "sans bandes", qui évite les problèmes de gestion de média.
- Les données sont envoyées à la VLE via le protocole IP et ne requièrent pas d'extension de canal.
- Par rapport au montage et à la lecture d'une cartouche MVC, la VLE peut effectuer un audit MVC en quelques secondes, à l'aide de sa base de données interne.

Ce chapitre décrit l'utilisation de la VLE dans un environnement simple à deux sites. Il convient toutefois de noter que la solution peut prendre en charge autant de sites que vous le souhaitez, qui contiennent chacun autant de VLE que nécessaire. De même, l'un des sites peut être un site DR uniquement, qui n'exécute pas de partitions LPAR MVS, sauf pendant un DR ou un sinistre déclaré.

La procédure ci-après utilise l'environnement suivant : deux sites, SITE1 et SITE2. Chaque site dispose d'un VSM et d'une VLE. Dans cet exemple, SITE2 est décrit en tant que site DR uniquement, mais peut également être un site de production défini en tant qu'image miroir de SITE1.

Remarque :

La taille du tampon VLE sur SITE2 doit être suffisante pour contenir à la fois les données de production migrées et les données créées lors d'un test DR.

Mode de production normale

Lors d'une production normale, des stratégies sont définies sur SITE1 pour migrer une copie des données vers la VLE locale sur le site SITE1 et une deuxième copie vers la VLE distante sur le site SITE2. Vous pouvez créer des copies supplémentaires si vous le souhaitez, en incluant les deux copies dans une autre VLE, ainsi que des copies de bande.

Vous trouverez ci-dessous un exemple des stratégies définies sur SITE1.

Les définitions SMC sont utilisées pour affecter un nom *MGMTCLAS* de VLEPROD à des jeux de données avec un qualificatif de haut niveau "PAYROLL".

```
POLICY NAME(VLEPOL) MEDIA(VIRTUAL) MGMT(VLEMGMT) + SUBP(VIRTSCR)
TAPEREQ DSN(PAYROLL.*) POLICY(VLEPOL)
```

Les définitions HSC *POOLPARAM/VOLPARAM* sont utilisées pour définir les volumes de production :

```
POOLPARAM TYPE(MVC) NAME(LOCAL)
VOLPARAM VOLSER(VLL000-VLL099)
POOLPARAM TYPE(MVC) NAME(VAULT1)
VOLPARAM VOLSER(VLV000-VLV099)
POOLPARAM TYPE(SCRATCH) NAME(VIRTSCR)
VOLPARAM VOLSER(V00000-V99999) MEDIA(VIRTUAL)
```

Remarque :

Notez que les MVC dans les pools LOCAL et VAULT1 sont des VMVC (cartouches MVC virtuelles) dans les VLE SITE1 et SITE2, respectivement, et ne sont associées à aucun type de média.

Les instructions VTCS *STORCLAS* et *MGMTCLAS* sont utilisées pour définir les stratégies VTCS :

```
STOR NAME(VLE1) STORMNGR(SITE1VLE) MVCPPOOL(LOCAL)
STOR NAME(VLE2) STORMNGR(SITE2VLE) MVCPPOOL(VAULT1)
MGMT NAME(VLEMGMT) DELSCR(YES) MIGPOL(VLE1,VLE2)
```

Lorsqu'un travail s'exécute avec un jeu de données débutant par le qualificatif de haut niveau "PAYROLL," SMC utilise *TAPEREQ* et *POLICY* pour affecter une instruction *MGMTCLAS* de VLEPROD à la demande de montage. VTCS sélectionne un volume virtuel provisoire dans le pool LOCSCR (plage V00000-V99999) et lui affecte un *MGMTCLAS* de VLEPROD. Une fois que le volume est démonté, une copie est migrée vers la VLE locale (STORMNGR SITE1VLE) tandis que la deuxième copie est migrée vers la VLE distante (STORMNGR SITE2VLE).

Exécution d'un test DR avec VLE

Le processus de configuration d'un test DR sur SITE2 est simple et rapide. En outre, il requiert des restrictions minimales sur SITE1.

Les principales étapes sont les suivantes :

1. Créez un nouveau CDS sur SITE2 contenant uniquement les données de configuration de base.

2. Marquez les VMVC de SITE1 à l'aide de l'attribut *READONLY* (lecture seule) pour éviter les conflits.
3. Effectuez un audit des cartouches MVC de production virtuelles dans la VLE SITE2VLE. Cette étape remplit le CDS à l'aide des métadonnées virtuelles existantes. En fonction du nombre de VTV dans la VLE, cette étape peut durer de quelques minutes à moins d'une heure.
4. Exécutez la charge globale de test DR, en utilisant une plage de VTV et de MVC qui ne chevauche pas les volumes de production.

Le reste de cette section fournit des détails relatifs à la définition des paramètres sur le site DR et décrit les étapes que vous devez exécuter pour vérifier que les VMVC de production de contenu ne sont pas modifiées lors du test.

1. Créez le CDS du test DR.
 - a. Utilisez le processus *LIBGEN/SLICREAT* pour créer le CDS sur le site SITE2. Notez que vous devez créer ce CDS même si vous exécutez déjà un travail de production sur SITE2. Le nouveau CDS ne contient que des données DR provenant de SITE1. Vous devez également définir au moins un ACS dans les macros *LIBGEN*, même si votre configuration ne contient pas de bande physique.
 - b. Lancez l'utilitaire *SET VOLPARM* pour définir les volumes pour le test DR :

```
POOLPARM TYPE(MVC) NAME(VAULT1)
VOLPARM VOLSER(VLV000-VLV099)
POOLPARM TYPE(EXTERNAL) NAME(PRODVTVS)
VOLPARM VOLSER(V00000-V99999) MEDIA(VIRTUAL)
POOLPARM TYPE(MVC) NAME(DRMVC)
VOLPARM VOLSER(VLT000-VLT099)
POOLPARM TYPE(SCRATCH) NAME(VIRTSCR)
VOLPARM VOLSER(VT0000-VT9999) MEDIA(VIRTUAL)
```

Les deux premiers pools définissent les volumes créés par SITE1 qui seront utilisés comme entrée du test sur SITE2. Le type de pool *EXTERNAL* indique que les volumes ne font pas partie d'un sous-pool provisoire. Les deux derniers pools sont des pools locaux qui seront utilisés comme sortie du test sur SITE2.

- c. Définissez les instructions *VTCS MGMTCLAS* et *STORCLAS* qui seront utilisées pour le test DR :

```
STOR NAME(DRVLE) STORMNGR(SITE2VLE) MVCPPOOL(DRMVC)
MGMT NAME(VLEMGMT) DELSCR(YES) MIGPOL(DRVLE)
```

- d. Etant donné que les sous-pools provisoires et *MGMTCLAS* sur le système DR de SITE2 portent les mêmes noms que les stratégies de production (mais ont des définitions différentes), vous pouvez désormais utiliser les mêmes instructions *SMC POLICY* et *TAPEREQ* pour le test DR de votre SITE2 que sur le site de production SITE1.

- e. Affichez HSC/VTCS sur la partition LPAR de test DR.
2. Marquez les cartouches MVC de production à l'aide de l'attribut *READONLY* (lecture seule).
 - a. Cela constitue une étape essentielle du processus, qui doit être effectuée à la fois sur le CDS de production de SITE1 et sur le CDS de test DR de SITE2. Une fois que les cartouches MVC ont été définies en tant que *READONLY* sur le CDS de production, vous pouvez continuer à exécuter un traitement normal, incluant :

RECLAIM. La récupération automatique ne sélectionnera pas de cartouche MVC avec le statut *READONLY*.

SCRATCH. Bien que les VTV soient mis à jour dans le CDS de production pour passer à l'état provisoire et qu'ils puissent être réutilisés, la copie sur la cartouche MVC virtuelle en lecture seule de la VLE ne s'en trouve pas affectée.

Le traitement normal permet d'effectuer un ajout aux VTV ou de les écraser sur les VMVC. Les nouvelles versions VTV seront migrées vers les nouvelles VMVC, tandis que la copie située sur la cartouche MVC virtuelle en lecture seule de la VLE demeurera inchangée.

Remarque :

Toutefois, vous ne pouvez pas exécuter l'utilitaire *DRAIN* sur ces MVC, car il supprime la copie VLE des métadonnées de la cartouche MVC virtuelle.

- b. Utilisez la fonction d'utilitaire *ACTMVCGN* pour sélectionner les cartouches MVC de production sur le site de production, en utilisant le CDS de production. Cet utilitaire génère des instructions de contrôle pour définir l'indicateur *READONLY* sur les cartouches MVC qu'il sélectionne, ainsi que des instructions de contrôle pour désactiver l'indicateur *READONLY* une fois le test terminé. L'utilisation du mot-clé *ALL* dans l'instruction de contrôle *ACTMVCGN* garantit que la totalité des cartouches MVC seront sélectionnées pour un traitement en lecture seule (*READONLY*), permettant ainsi l'exécution d'une récupération automatique sur le système de production sans impact sur le test DR. L'instruction *SLUSMAUD DD* doit également être incluse pour générer des instructions *AUDIT* pour les VMVC qui seront utilisées dans le test. Si vous voulez, vous pouvez exécuter l'utilitaire *ACTMVCGN* sur le site de production pour créer les mises à jour de production et sur une copie en miroir du CDS sur le site DR pour créer les mises à jour du CDS de test DR. Vous trouverez ci-dessous un exemple du JCL permettant d'exécuter cet utilitaire :

```
//ACTMVCGN JOB (ACCT), 'ACTMVCGN', NOTIFY=&SYSUID
//ACTMVCG1 EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//* NOTE: CDS DD statements are optional if running at the production
//* site with an active HSC LPAR.
```

```

//SLSCNTL DD DSN=h1q.DBASEPRM,DISP=SHR
//SLSCNTL2 DD DSN=h1q.DBASESEC,DISP=SHR
//SLSSTBY DD DSN=h1q.DBASESBY,DISP=SHR
/* NOTE: MVCMAINT READONLY(ON) STATEMENTS
//SLUSMVON DD DSN=h1q.SLUSMVON,DISP=(NEW,CATLG,DELETE),
//
//          SPACE=(CYL,1)
/* NOTE: MVCMAINT READONLY(OFF) STATEMENTS
//SLUSMVOF DD DSN=h1q.SLUSMVOF,DISP=(NEW,CATLG,DELETE),
//
//          SPACE=(CYL,1)
/*
//          NOTE: AUDIT MVC(VVVVVV) STATEMENTS
//SLUSMAUD DD DSN=h1q.SLUSMAUD,DISP=(NEW,CATLG,DELETE),
//
//          SPACE=(CYL,1)
/* NOTE: THE FOLLOWING SELECTS ALL "NON-EMPTY" VMVCS
//SLSIN DD *
ACTMVCGN ALL MVCPOOL(VAULT1)
/*

```

3. Sur le site de production, exécutez la fonction d'utilitaire *MVCMAINT* pour marquer les VMVC à l'aide de l'indicateur *READONLY*.

```

//RONLYON EXEC PGM=SLUADMIN,PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//SLSPRINT DD SYSOUT=*
/* NOTE: EXEC MVCMAINT TO SET READONLY(ON). Output of
/*
//          ACTMVCGN utility.
//SLSIN DD DSN=h1q.SLUSMVON,DISP=SHR

```

4. Affichez le HSC/VTCS sur le site DR.
5. Exécutez un audit MVC des VMVC de production dans la VLE SITE2 à l'aide du CDS SITE2 récemment créé et de la sortie de l'utilitaire *ACTMVCGN*. Cette étape remplit les métadonnées du CDS contenant les relations entre les VTV et les VMVC.

```

//AUDIT EXEC PGM=SLUADMIN
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//SLSPRINT DD SYSOUT=*
/*
//          NOTE: AUDIT CONTROL STATEMENTS FROM ACTMVCGN UTILITY
//SLSIN DD DSN=h1q.SLUSMAUD,DISP=SHR

```

Vous pouvez éventuellement rappeler dans le tampon VTSS des VTV qui seront utilisés dans le test DR, soit à l'aide de LCM, soit à l'aide d'une autre méthode de sélection des VTV à rappeler. Cette étape n'est cependant pas nécessaire, étant donné que les rappels à partir du tampon VLE sont relativement rapides.

6. Exécutez l'utilitaire *MVCMaint* à l'aide de la sortie de la commande *ACTMVCgn READONLY(ON)* pour définir les VMVC de production sur *READONLY* sur le site *SITE2* du CDS DR.

```
//RONLYON EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
/* NOTE: EXEC MVCMaint TO SET READONLY(ON). Output of
/* ACTMVCgn utility.
//SLSIN DD DSN=h1q.SLUSMVON, DISP=SHR
```

7. Facultatif: avant de démarrer le test DR, vous voudrez peut-être exécuter un *VTVRPT* et un *MVCRPT* pour valider le contenu de votre CDS de test DR.
8. Exécutez votre charge globale de test DR.
 - a. Affichez SMC. Si vous avez utilisé les mêmes noms sur vos sous-pools provisoires et *MGMTCLAS* que sur le système de production, vous pouvez vous servir de vos instructions de production *TAPEREQ* et *POLICY*. Il n'est pas obligatoire, mais conseillé, d'utiliser un nom de TapePlex différent pour le TapePlex de test DR.
 - b. Exécutez votre charge de travail de test DR à l'aide du SMC et du nouveau CDS de HSC/VTCS.
 - c. Il n'existe pas de restrictions quant à la mise à jour des volumes VTV de production lors du test DR. Les données figurant sur les VTV de production peuvent être ajoutées à (*DISP=MOD*) ou remplacées (*DISP=OLD*). Ces mises à jour n'affectent pas le contenu de la copie VTV sur la cartouche MVC virtuelle de production *READONLY* et n'affectent donc pas non plus la copie de production des données.

Nettoyage après un test DR avec VLE

Une fois le test DR terminé, l'objectif du nettoyage vise à supprimer des métadonnées du VTSS et de la VLE afin que le test DR suivant ne voie pas ces données. Notez que le HSC/VTCS du test DR doit rester actif jusqu'à la fin du nettoyage. Pour ce faire, procédez comme suit :

1. Exécutez la fonction d'utilitaire *SCRATCH* pour supprimer tous les VTV créés lors du test, à la fois du VTSS et des VMVC de test DR de la VLE. Lorsque le paramètre *DELSCR(YES)* est spécifié sur la *MGMTCLAS* du test DR, l'exécution de l'utilitaire de suppression provoque la suppression des VTV dans le tampon et les métadonnées VLE.

```
//SCRATCH EXEC PGM=SLUADMIN
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
SCRATCH VOL(VT0000-VT9999)
```

Si vous avez modifié des VTV de production à l'aide de *DISP=MOD* ou *DISP=OLD*, ces VTV restent dans le tampon et sur la VLE.

En supprimant les VTV dans le sous-pool de test DR après le test, vous réduisez le temps requis pour le nettoyage du VTSS ainsi que la quantité de données restantes dans la VLE à la fin du test.

2. Migrez le VTSS vers 0.

```
//MIGRTO0 EXEC PGM=SLUADMIN
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
MIGRATE VTSS(DRVTSS) THRESHLD(0)
```

Cette étape est requise uniquement si la sortie de votre test DR incluait de nouvelles versions des VTV de production.

3. Vérifiez que le VTSS DR est maintenant vide.

```
//AUDVTSS EXEC PGM=SLUADMIN
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
AUDIT VTSS(DRVTSS)
```

Si vous avez modifié des VTV de production lors de votre test DR, les copies de ces données, ainsi que les métadonnées, restent dans la VLE pour le pool MVC de test DR (VLT000-VLT099, VTV V00000-V99999). Lors du test DR suivant, ces VMVC seront écrites en commençant au début logique de la bande et toutes les données qu'ils contiennent seront supprimées de la VLE. Etant donné que le nouveau CDS du test DR ne connaît pas ces données, cela n'aura aucun impact sur le test DR suivant.

4. Sur le site de production, utilisez les cartes de contrôle *READONLY(OFF)* créées par l'utilitaire *ACTMVCGM* au début du test pour que les VMVC de production se trouvent à nouveau dans un statut inscriptible.

```
//RONLYOF EXEC PGM=SLUADMIN,PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//SLSPRINT DD SYSOUT=*
//* NOTE: EXEC MVCMAINT TO SET READONLY(OFF)
//SLSIN DD DSN=h1q.SLUSMVOF,DISP=SHR
```

Utilisation de VLE pour la continuité de l'activité

En cas de panne sur le site SITE1 nécessitant que SITE2 reprenne la charge globale de SITE1, le processus est presque identique à la procédure de test DR.

S'il arrive qu'un test DR soit en cours d'exécution lors de la panne de SITE1, suivez le processus ci-dessus pour effectuer un nettoyage après le test DR et arrêter ce dernier.

Pour commencer à exécuter la charge globale de SITE1 sur SITE2, suivez la procédure décrite ci-dessus pour le démarrage d'un test DR. Certes, vous omettez l'étape de marquage des VMVC de production à l'aide du paramètre *READONLY* sur le CDS de production, car il n'existe pas de CDS de "production" à mettre à jour. Toutefois, vous utiliserez la copie en miroir du CDS de production pour générer les cartes de contrôle *MVCMaint READONLY* pour les cartouches MVC de production dans la VLE.

Vous voudrez également utiliser les stratégies de test DR qui séparent les VTV créés et les VMVC de sortie en deux plages distinctes, afin d'éliminer toute possibilité d'endommager les données de production tant que la continuité de l'activité n'a pas été vérifiée.

Remarque :

Si aucun point de synchronisation n'est défini pour les travaux de production qui exécutent le traitement *DISP=MOD* pour les données de bande, il est possible que le contenu d'un VTV au moment d'une panne soit imprévisible. StorageTek recommande de passer en revue toutes les procédures de récupération après sinistre pour garantir des points de synchronisation prévisibles pour les données de bande.

Annexe A. Exemples de VTSS en cluster

La section "[Utilisation de configurations VTSS en cluster](#)" explique les bases du clustering VTSS et cette annexe fournit les exemples de clustering suivants :

- "[VTSS en cluster unidirectionnel](#)"
- "[VTSS en cluster bidirectionnel](#)"
- "[Clustering étendu](#)"
- "[Cluster VSM5 à VSM5 avec CLINK TCP/IP](#)"
- "[Cluster VSM5 à VSM6 avec CLINK TCP/IP, VLE interconnectées](#)"
- "[Cluster "sans bandes" VSM 6 à VSM 6 avec CLINK TCP/IP](#)"
- "[Choix entre un cluster unidirectionnel ou bidirectionnel](#)"

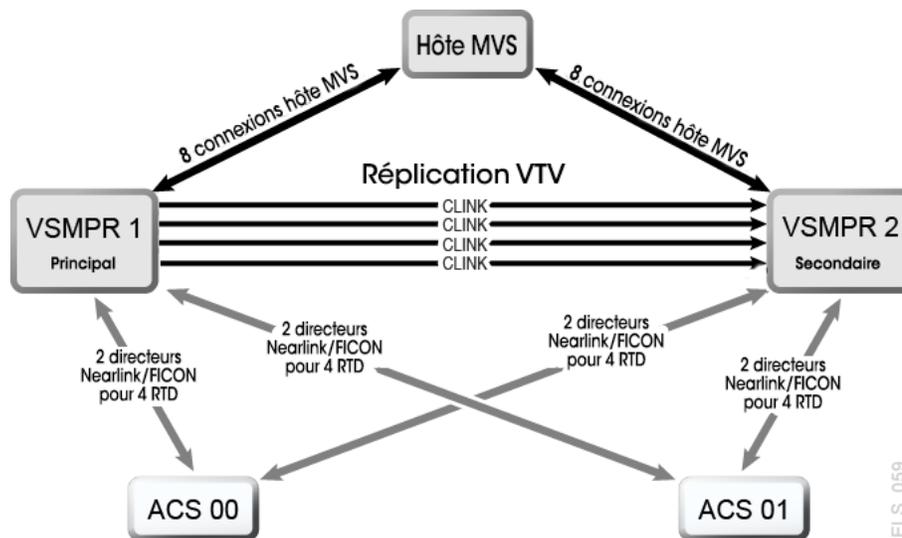
VTSS en cluster unidirectionnel

La [Figure A.1, « Configuration ACS double de VTSS en cluster unidirectionnels »](#) présente un exemple de système ACS double de VTSS en cluster unidirectionnels. Notez que dans cet exemple les ports FICON fournissent les connexions CLINK. Ici, il n'y a qu'un seul hôte MVS, mais il dispose d'un grand nombre de données critiques que vous devez protéger à l'aide de deux tout nouveaux VSM4 que vous venez d'acquérir.

VTSS1 est le VTSS principal, qui est connecté au VTSS secondaire (VTSS2) à l'aide de liens de cluster (CLINK). Si la classe de gestion d'un VTV spécifie une réplication, lorsque le VTV arrive sur VTSS1, il est répliqué (copié) sur VTSS2 et également immédiatement migré (avec KEEP).

Par conséquent, vous avez augmenté la disponibilité des données (il existe une copie du VTV dans chaque VTSS en cas de panne de l'un d'eux) et la protection des données (le VTV se trouve également sur bande au cas où les deux VTSS seraient hors ligne). Un VTSS en cluster constitue donc une excellente solution pour la continuité et la reprise de l'activité.

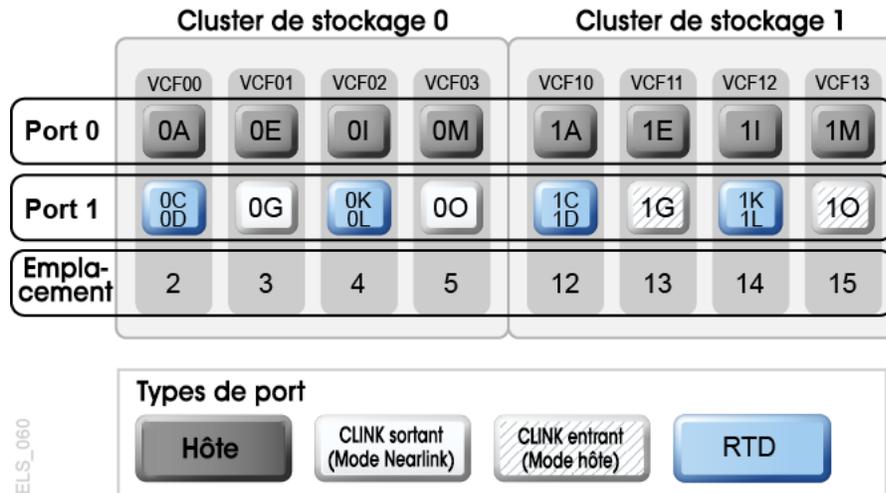
Figure A.1. Configuration ACS double de VTSS en cluster unidirectionnels



Examinons maintenant le matériel pour cette configuration en cluster. La [Figure A.2, « VSM4 avec 8 cartes VCF, 8 ports d'hôte, directeurs FICON pour 8 RTD, 4 ports CLINK »](#) présente les identificateurs d'interface de canal CONFIG pour un VSM4 avec 8 cartes VCF. Dans cette configuration, vous avez alloué :

- 8 ports d'hôte.
- 4 ports pour les RTD. Les ports RTD sont tous connectés aux directeurs FICON, chacun étant relié aux RTD, de sorte que les identificateurs *CHANIF* pour les deux RTD s'affichent sur chaque port. Cela permet une connexion d'arrière-plan vers 8 RTD, bien qu'un seul RTD par port/directeur ne puisse être actif à la fois.
- 4 ports pour les connexions CLINK afin de former un cluster VTSS unidirectionnel et 8 ports vers les connexions hôtes. Pour former le VTSS en cluster, deux VSM4 (VTSS1 et VTSS2) sont configurés de façon identique comme indiqué à la [Figure A.2, « VSM4 avec 8 cartes VCF, 8 ports d'hôte, directeurs FICON pour 8 RTD, 4 ports CLINK »](#).

Figure A.2. VSM4 avec 8 cartes VCF, 8 ports d'hôte, directeurs FICON pour 8 RTD, 4 ports CLINK



Vous savez désormais à quoi ressemble un exemple de cluster unidirectionnel et vous avez vu les configurations de port de carte VCF requises. Consultez maintenant la section "[Configuration et gestion d'un système VTSS en cluster unidirectionnel.](#)"

Configuration et gestion d'un système VTSS en cluster unidirectionnel

Pour configurer et gérer le système en cluster unidirectionnel présenté à la [Figure 6.1, « VTSS en cluster unidirectionnel »](#), procédez comme suit :

1. Vérifiez que votre système dispose de la configuration pour un VTSS en cluster.
2. Utilisez *CONFIG* pour créer des instructions *CLUSTER* et *CLINK* afin de définir le cluster VTSS et ses connexions.

L'exemple suivant montre un JCL CONFIG qui définit un cluster unidirectionnel de deux VSM4 (VTSS1 et VTSS2) comme présenté à la [Figure 6.1, « VTSS en cluster unidirectionnel »](#).

Prenez en compte les éléments suivants :

- L'instruction *CLUSTER* définit le cluster en tant qu'élément composé de VTSS1 et VTSS2.
- Des instructions *CLINK* utilisent les ports émetteur (mode Nearlink) de **VTSS1 uniquement** pour activer le cluster en tant qu'unidirectionnel, où VTSS1 est le VTSS principal et VTSS2 le VTSS secondaire.

```
//CREATECFG EXEC PGM=SWSADMIN,PARM='MIXED'
```

```

//STEPLIB DD DSN=h1q.SLSLINK,DISP=SHR
//SLS_CNTL DD DSN=FEDB.VSMLMULT.DBASEPRM,DISP=SHR
//SLS_CNTL2 DD DSN=FEDB.VSMLMULT.DBASESEC,DISP=SHR
//SLSSTBY DD DSN=FEDB.VSMLMULT.DBASESETBY,DISP=SHR
//CFG22202 DD DSN=FEDB.VSMLMULT.CFG22202,DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
CONFIG RESET CDSLEVEL(V62ABOVE)
GLOBAL MAXVTV=65000 MVCFREE=60 VTVATTR=SCRATCH RECALWER=YES
LOCKSTR=STK_VTCS_LOCKS VTVPAGE=LARGE
RECLAIM THRESHLD=70 MAXMVC=30 START=40 CONMVC=5
VTVVOL LOW=905000 HIGH=999999 SCRATCH
VTVVOL LOW=C00000 HIGH=C25000 SCRATCH
VTVVOL LOW=RMM000 HIGH=RMM020 SCRATCH
MVCVOL LOW=N25980 HIGH=N25989
MVCVOL LOW=N35000 HIGH=N35999
VTSS NAME=VTSS1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTD NAME=PR11A00 DEVNO=1A00 CHANIF=0C
RTD NAME=PR11A01 DEVNO=1A01 CHANIF=0D
RTD NAME=PR11A02 DEVNO=1A02 CHANIF=0K
RTD NAME=PR11A03 DEVNO=1A03 CHANIF=0L
RTD NAME=PR12A08 DEVNO=2A08 CHANIF=1C
RTD NAME=PR12A09 DEVNO=2A09 CHANIF=1D
RTD NAME=PR12A0A DEVNO=2A0A CHANIF=1K
RTD NAME=PR12A0B DEVNO=2A0B CHANIF=1L
VTD LOW=9900 HIGH=99FF
VTSS NAME=VTSS2 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTD NAME=PR23A00 DEVNO=3A00 CHANIF=0C
RTD NAME=PR23A01 DEVNO=3A01 CHANIF=0D
RTD NAME=PR23A02 DEVNO=3A02 CHANIF=0K
RTD NAME=PR23A03 DEVNO=3A03 CHANIF=0L
RTD NAME=PR24A08 DEVNO=4A08 CHANIF=1C
RTD NAME=PR24A09 DEVNO=4A09 CHANIF=1D
RTD NAME=PR24A0A DEVNO=4A0A CHANIF=1K
RTD NAME=PR24A0B DEVNO=4A0B CHANIF=1L
VTD LOW=9900 HIGH=99FF
CLUSTER NAME=CLUSTER1 VTSSs(VTSS1,VTSS2)
CLINK VTSS=VTSS1 CHANIF=0G
CLINK VTSS=VTSS1 CHANIF=0O
CLINK VTSS=VTSS1 CHANIF=1G
CLINK VTSS=VTSS1 CHANIF=1O

```

3. Indiquez le paramètre de réplication conditionnelle dans l'instruction *CONFIG GLOBAL*.

```
CONFIG GLOBAL REPLICAT=CHANGED
```

Dans cet exemple, *CONFIG GLOBAL REPLICAT=CHANGED* spécifie les points suivants :

- Répliquez des VTV uniquement si le VTV est mis à jour et s'il n'en existe pas de copie identique sur le VTSS secondaire.
- A l'aide du paramètre *MIGPOL*, migrez les VTV mis en duplex sur les ACS 00 et 01 par les classes de stockage que vous avez créées lors de l'étape 5.

Pour répliquer des VTV de façon inconditionnelle, spécifiez *CONFIG GLOBAL REPLICAT=ALWAYS*.

4. Créez une classe de gestion qui spécifie la réplication VTV et deux classes de stockage pour migrer (mettre en duplex) les VTV répliqués.

```
MGMT NAME(VSMREPL) REPLICAT(YES) MIGPOL(REPLSTR1,REPLSTR2)
```

Remarque :

- Notez l'interaction entre *GLOBAL REPLICAT*, qui spécifie le moment où la réplication peut se produire, et *MGMTclass REPLICAT(YES)*, grâce à laquelle la condition *GLOBAL REPLICAT* déclare que la réplication peut débiter.
- La classe de gestion *VSMREPL* n'indique pas de stratégie de migration immédiate. La réplication VTV applique automatiquement la migration immédiate. Les VTV qui se trouvent dans cette classe de gestion sont ajoutés à la file d'attente de migration immédiate sur VTSS, une fois la réplication terminée. Notez que la mise en duplex n'est pas une condition requise pour la réplication de VTV. Pour plus d'informations, voir " [Fonctionnement des configurations VTSS en cluster.](#)"

5. Créez les classes de stockage des MVC qui contiennent les VTV répliqués et migrés.

```
STOR NAME(REPLSTR1) ACS(00) MEDIA(STK1R) MIRATE(RECEIVER)
STOR NAME(REPLSTR2) ACS(01) MEDIA(STK1R) MIGRATE(RECEIVER)
```

Dans cet exemple, l'instruction *STORclass* définit les classes de stockage *REPLSTR1* et *REPLSTR2* référencées dans le paramètre *MIGPOL* à l'étape 4. Notez également que les paramètres *MIGRATE* sur les classes de stockage indiquent que le VTSS qui reçoit le VTV répliqué (ici le VTSS secondaire, VTSS2), effectue la migration vers les deux ACS. Cela permet de s'assurer que le VTSS secondaire fonctionne en tant que moteur de migration.

6. Chargez les instructions de contrôle *MGMTclass* et *STORclass* à l'aide de la commande *MGMTDEF*.

```
MGMTDEF DSN(hsc.parms)
```

7. Créez une instruction *TAPEREQ* pour acheminer les données critiques vers VSM et affecter la classe de gestion *VSMREPL* aux données.

```
TAPEREQ DSN(*.PAYROLL.***) MEDIA(VIRTUAL) MGMT(VSMREPL)
```

Dans cet exemple, l'instruction *TAPEREQ* spécifie :

- d'acheminer les jeux de données avec le masque HLQ *.PAYROLL.** vers VSM ;
- d'affecter la classe de gestion VSMREPL que vous avez créée à l'étape 4.

Attention :

Pour répliquer des VTV, VTSS1 et VTSS2 doivent être basculés en ligne sur VTCS, afin que ce dernier puisse envoyer des commandes de contrôle aux deux VTSS. Pour plus d'informations, voir "[Fonctionnement des configurations VTSS en cluster](#)".

Vous pouvez également utiliser la substitution de groupe ésotérique via l'instruction SMC *TAPEREQ* ou les routines ACS SMC DFSMS pour acheminer des travaux de réplication vers VSM. Pour plus d'informations, voir le manuel *SMC Configuration and Administration Guide*.

8. Consultez vos options HSC PARMLIB pour vérifier que les enregistrements de sous-type 28 sont activés.

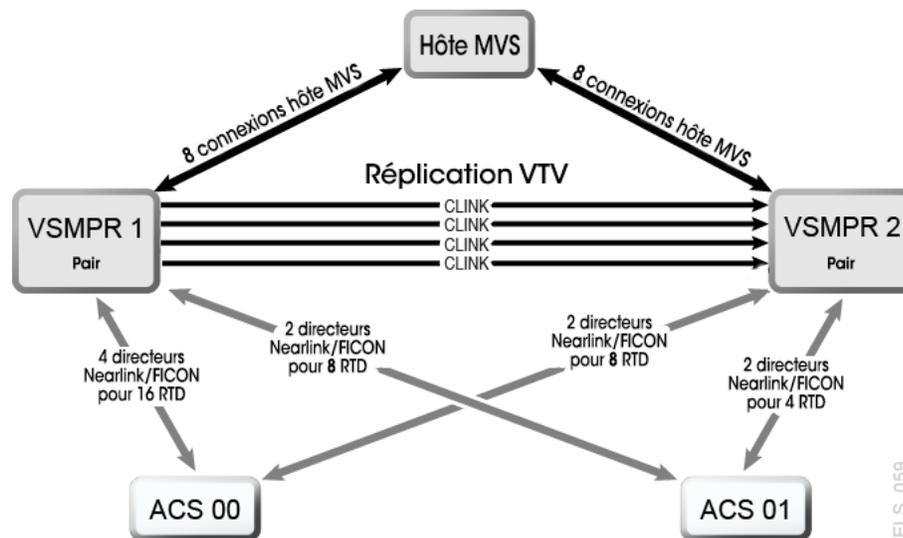
Si tel est le cas, le clustering VTSS écrit un enregistrement de sous-type 28 pour chaque réplication effectuée.

VTSS en cluster bidirectionnel

La [Figure A.3, « Configuration ACS double de VTSS en cluster bidirectionnels »](#) présente un exemple de système ACS double de VTSS en cluster bidirectionnels. Notez que dans cet exemple les ports FICON fournissent les connexions CLINK.

Ce système ressemble beaucoup à l'exemple unidirectionnel, mais va plus loin : deux hôtes MVS partagent un CDS et tous les éléments de l'image sont connectés entre eux. Ces sites sont mis en miroir l'un avec l'autre pour optimiser la disponibilité et la protection des données. Pour les rendre bidirectionnels, vous devez configurer les deux VTSS en tant qu'homologues à l'aide des instructions *CLINK*.

Figure A.3. Configuration ACS double de VTSS en cluster bidirectionnels

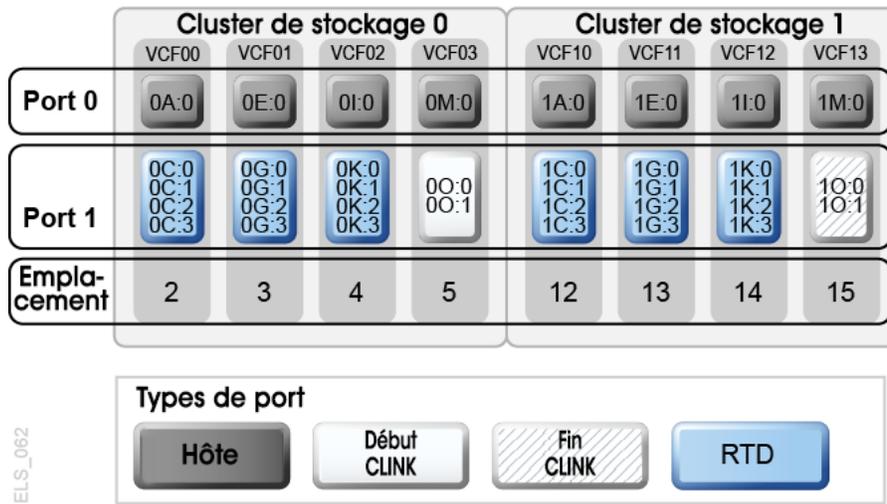
**Remarque :**

- Le clustering bidirectionnel requiert VTCS 6.1 et versions ultérieures. Vous ne pouvez pas configurer un cluster bidirectionnel dans des versions antérieures à VTCS 6.1.
- Cette configuration est présentée avec un total de 16 transferts d'E/S NearLink simultanés activés. Ils peuvent être déployés sur plusieurs cibles sur 14 ports NearLink au maximum, avec un total de 2 transferts d'E/S NearLink simultanés par port. Cette fonctionnalité requiert le microcode VTSS D02.06 .00.00 ou version ultérieure.

La Figure A.4, « VSMPR1 - VSM5 avec 8 cartes VCF, 8 ports d'hôte, directeurs FICON pour 24 RTD, 4 CLINK » présente les identificateurs d'interface de canal *CONFIG* pour VSMPR1 indiqués à la Figure A.3, « Configuration ACS double de VTSS en cluster bidirectionnels ». Dans cette configuration, vous avez alloué :

- 8 ports d'hôte.
- 6 ports pour les RTD. Les ports RTD sont tous connectés aux directeurs FICON, chacun étant relié à 4 RTD, de sorte que les identificateurs CHANIF pour les 4 RTD s'affichent sur chaque port. Cela permet une connexion d'arrière-plan vers 24 RTD, bien qu'un seul RTD par port/directeur ne puisse être actif à la fois.
- 4 ports utilisant des directeurs FICON. Deux sont de type Nearlink pour le port de début, tandis que les deux autres sont en mode hôte pour le port de fin, afin que les connexions CLINK forment un cluster VTSS bidirectionnel.

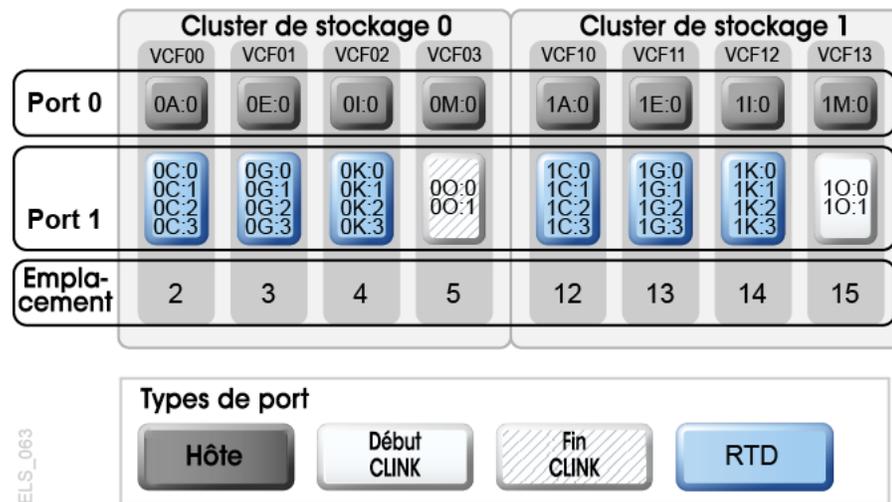
Figure A.4. VSMR1 - VSM5 avec 8 cartes VCF, 8 ports d'hôte, directeurs FICON pour 24 RTD, 4 CLINK



La Figure A.5, « VSMR2 - VSM5 avec 8 cartes VCF, 8 ports d'hôte, directeurs FICON pour 24 RTD, 4 CLINK » présente les identificateurs d'interface de canal *CONFIG* pour un VSMR1, un VSM5 dans un cluster bidirectionnel avec 8 cartes VCF et la fonctionnalité de 32 RTD au maximum activée. Dans cette configuration, vous avez alloué :

- 8 ports d'hôte.
- 6 ports pour les RTD. Les ports RTD sont tous connectés aux directeurs FICON, chacun étant relié à 4 RTD, de sorte que les identificateurs *CHANIF* pour les 4 RTD s'affichent sur chaque port. Cela permet une connexion d'arrière-plan vers 24 RTD, bien qu'un seul RTD par port/directeur ne puisse être actif à la fois.

Figure A.5. VSMRP2 - VSM5 avec 8 cartes VCF, 8 ports d'hôte, directeurs FICON pour 24 RTD, 4 CLINK



- 4 ports utilisant des directeurs FICON. Deux sont de type Nearlink pour le port de début, tandis que les deux autres sont en mode hôte pour le port de fin, afin que les connexions CLINK forment un cluster VTSS bidirectionnel.

Attention :

Comme indiqué à la [Figure 6.3, « CLINK ESCON/FICON pour VTSS en cluster bidirectionnels »](#), chaque CLINK doit obligatoirement être connecté au même cluster de stockage sur chaque VTSS. Si la configuration ne répond pas à cette exigence, vous risquez d'être confronté à des erreurs de réplication, de canal et de communication. Comme indiqué, les ports Nearlink (CLINK de début) de VSMRP1 se trouvent donc sur le cluster de stockage 0, tout comme les ports d'hôte (CLINK de fin) de VSMRP2. Il en va de même pour les connexions CLINK pour les données circulant dans l'autre sens : elles se trouvent toutes les deux sur le cluster de stockage 1.

Configuration et gestion d'un système en cluster bidirectionnel

Pour configurer et gérer le système en cluster bidirectionnel présenté à la [Figure A.3, « Configuration ACS double de VTSS en cluster bidirectionnels »](#), procédez comme suit :

1. Vérifiez que votre système répond aux conditions requises pour les VTSS en cluster, qui sont décrites à la section *Installation d'ELS*.
2. Utilisez *CONFIG* pour créer des instructions *CLUSTER* et *CLINK* afin de définir le cluster VTSS et ses connexions.

L'exemple suivant montre un JCL CONFIG qui définit un cluster bidirectionnel de deux VSM4 (VSMRP1 et VSMRP2), comme présenté à la [Figure A.3, « Configuration ACS double de VTSS en cluster bidirectionnels »](#).

- Notez que l'instruction *CLUSTER* définit le cluster en tant qu'élément composé de VSMR1 et VSMR2.
- Des instructions *CLINK* utilisent les ports émetteur (mode Nearlink) des deux VTSS pour activer le cluster en tant que bidirectionnel et se connectent à l'aide du même cluster de stockage sur chaque VTSS pour les ports émetteur et récepteur de chaque CLINK.

```
//CREATECF EXEC PGM=SWSADMIN, PARM='MIXED'  
//STEPLIB DD DSN=h1q.SLSLINK, DISP=SHR  
//SLSCNTL DD DSN=FEDB.VSMLMULT.DBASEPRM, DISP=SHR  
//SLSCNTL2 DD DSN=FEDB.VSMLMULT.DBASESEC, DISP=SHR  
//SLSSTBY DD DSN=FEDB.VSMLMULT.DBASESTBY, DISP=SHR  
//SLSPRINT DD SYSOUT=*  
//SLSIN DD *  
CONFIG RESET CDSLEVEL(V61ABOVE)  
GLOBAL MAXVTV=32000 MCVFREE=40 VTVATTR=SCRATCH RECALWER=YES LOCKSTR=VTCS_LOCKS  
REPLICAT=ALWAYS VTVPAGE=LARGE SYNCHREP=YES MAXRTDS=32  
RECLAIMTHRESHLD=70 MAXMVC=40 START=35  
RECLAIMTHRESHLD=70 MAXMVC=40 START=35  
VTVOL LOW=905000 HIGH=999999 SCRATCH  
VTVOL LOW=C00000 HIGH=C25000 SCRATCH  
VTVOL LOW=RMM000 HIGH=RMM020 SCRATCH  
MVCVOL LOW=N25980 HIGH=N25989  
MVCVOL LOW=N35000 HIGH=N35999  
VTSS NAME=VSMR1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5  
RTD NAME=VPR12A00 DEVNO=2A00 CHANIF=0C:0  
RTD NAME=VPR12A01 DEVNO=2A01 CHANIF=0C:1  
RTD NAME=VPR12A02 DEVNO=2A02 CHANIF=0C:2  
RTD NAME=VPR12A03 DEVNO=2A03 CHANIF=0C:3  
RTD NAME=VPR12A04 DEVNO=2A04 CHANIF=0G:0  
RTD NAME=VPR12A05 DEVNO=2A05 CHANIF=0G:1  
RTD NAME=VPR12A06 DEVNO=2A06 CHANIF=0G:2  
RTD NAME=VPR12A07 DEVNO=2A07 CHANIF=0G:3  
RTD NAME=VPR12A08 DEVNO=2A08 CHANIF=0K:0  
RTD NAME=VPR12A09 DEVNO=2A09 CHANIF=0K:1  
RTD NAME=VPR12A0A DEVNO=2A0A CHANIF=0K:2  
RTD NAME=VPR12A0B DEVNO=2A0B CHANIF=0K:3  
RTD NAME=VPR13A00 DEVNO=3A00 CHANIF=1C:0  
RTD NAME=VPR13A01 DEVNO=3A01 CHANIF=1C:1  
RTD NAME=VPR13A02 DEVNO=3A02 CHANIF=1C:2  
RTD NAME=VPR13A03 DEVNO=3A03 CHANIF=1C:3  
RTD NAME=VPR13A04 DEVNO=3A04 CHANIF=1G:0  
RTD NAME=VPR13A05 DEVNO=3A05 CHANIF=1G:1  
RTD NAME=VPR13A06 DEVNO=3A06 CHANIF=1G:2  
RTD NAME=VPR13A07 DEVNO=3A07 CHANIF=1G:3
```

```

RTD  NAME=VPR13A08  DEVNO=3A08  CHANIF=1K:0
RTD  NAME=VPR13A09  DEVNO=3A09  CHANIF=1K:1
RTD  NAME=VPR13A0A  DEVNO=3A0A  CHANIF=1K:2
RTD  NAME=VPR13A0B  DEVNO=3A0B  CHANIF=1K:3
VTD  LOW=9900  HIGH=99FF
VTSS NAME=VSMR2  LOW=70  HIGH=80  MAXMIG=8  MINMIG=4  RETAIN=5
RTD  NAME=VPR22B00  DEVNO=2B00  CHANIF=0C:0
RTD  NAME=VPR22B01  DEVNO=2B01  CHANIF=0C:1
RTD  NAME=VPR22B02  DEVNO=2B02  CHANIF=0C:2
RTD  NAME=VPR22B03  DEVNO=2B03  CHANIF=0C:3
RTD  NAME=VPR22B04  DEVNO=2B04  CHANIF=0G:0
RTD  NAME=VPR22B05  DEVNO=2B05  CHANIF=0G:1
RTD  NAME=VPR22B06  DEVNO=2B06  CHANIF=0G:2
RTD  NAME=VPR22B07  DEVNO=2B07  CHANIF=0G:3
RTD  NAME=VPR22B08  DEVNO=2B08  CHANIF=0K:0
RTD  NAME=VPR22B09  DEVNO=2B09  CHANIF=0K:1
RTD  NAME=VPR22B0A  DEVNO=2B0A  CHANIF=0K:2
RTD  NAME=VPR22B0B  DEVNO=2B0B  CHANIF=0K:3
RTD  NAME=VPR23B00  DEVNO=3B00  CHANIF=1C:0
RTD  NAME=VPR23B01  DEVNO=3B01  CHANIF=1C:1
RTD  NAME=VPR23B02  DEVNO=3B02  CHANIF=1C:2
RTD  NAME=VPR23B03  DEVNO=3B03  CHANIF=1C:3
RTD  NAME=VPR23B04  DEVNO=3B04  CHANIF=1G:0
RTD  NAME=VPR23B05  DEVNO=3B05  CHANIF=1G:1
RTD  NAME=VPR23B06  DEVNO=3B06  CHANIF=1G:2
RTD  NAME=VPR23B07  DEVNO=3B07  CHANIF=1G:3
RTD  NAME=VPR23B08  DEVNO=3B08  CHANIF=1K:0
RTD  NAME=VPR23B09  DEVNO=3B09  CHANIF=1K:1
RTD  NAME=VPR23B0A  DEVNO=3B0A  CHANIF=1K:2
RTD  NAME=VPR23B0B  DEVNO=3B0B  CHANIF=1K:3
VTD  LOW=9900  HIGH=99FF
CLUSTER NAME=CLUSTER1  VTSSs(VSMR1,VSMR2)
CLINK VTSS=VSMR1  CHANIF=00:0
CLINK VTSS=VSMR1  CHANIF=00:1
CLINK VTSS=VSMR2  CHANIF=10:0
CLINK VTSS=VSMR2  CHANIF=10:1

```

3. Indiquez le paramètre de réplication conditionnelle dans l'instruction *CONFIG GLOBAL*.

```
CONFIG GLOBAL REPLICAT=CHANGED
```

Comme dans l'exemple unidirectionnel, utilisez ici aussi *CONFIG GLOBAL REPLICAT=CHANGED*.

4. Créez une classe de gestion qui spécifie la réplication VTV et deux classes de stockage pour migrer (mettre en duplex) les VTV répliqués.

```
MGMT NAME(VSMREPL) REPLICAT(YES) MIGPOL(REPLSTR1,REPLSTR2)
```

Dans cet exemple, répliquez des VTV uniquement s'ils ont été modifiés et s'ils ne se trouvent pas dans l'autre VTSS du cluster. Migrez les éléments mis en duplex sur les ACS 01 et 00 par les classes de stockage que vous allez créer à l'étape 5.

5. Créez les classes de stockage des MVC qui contiennent les VTV répliqués et migrés.

```
STOR NAME(REPLSTR1) ACS(01) MEDIA(STK1R) MIRATE(EITHER)
STOR NAME(REPLSTR2) ACS(00) MEDIA(STK1R) MIGRATE(EITHER)
```

Dans cet exemple, l'instruction *STORcLas* définit les classes de stockage REPLSTR1 et REPLSTR2 référencées dans le paramètre *MIGPOL* à l'étape 4. Notez également que pour optimiser les ressources VTSS et RTD, les paramètres *MIGRATE* sur les classes de stockage autorisent les migrations à partir de n'importe quel VTSS. Il s'agit d'une stratégie courante pour les clusters VTSS bidirectionnels ou entre homologues.

6. Chargez les instructions de contrôle *MGMTcLas* et *STORcLas* à l'aide de la commande *MGMTDEF*.

```
MGMTDEF DSN(hsc.parms)
```

7. Créez une instruction *TAPEREQ* pour acheminer les données critiques vers VSM et affecter la classe de gestion VSMREPL aux données.

```
TAPEREQ DSN(*.PAYROLL.***) MEDIA(VIRTUAL) MGMT(VSMREPL)
```

Dans cet exemple, l'instruction *TAPEREQ* spécifie :

- d'acheminer les jeux de données avec le masque HLQ *.PAYROLL.** vers VSM ;
- d'affecter la classe de gestion VSMREPL que vous avez créée à l'étape 4.

Attention :

Pour répliquer des VTV, VSMR1 et VSMR2 doivent être basculés en ligne sur VTCS afin que ce dernier puisse envoyer des commandes de contrôle aux deux VTSS. Pour plus d'informations, voir "[Fonctionnement des configurations VTSS en cluster](#)".

- Vous pouvez également utiliser la substitution de groupe ésotérique via l'instruction SMC *TAPEREQ* ou les exits utilisateur d'ELS pour acheminer des travaux de réplication vers VSM. En cas de substitution d'un groupe ésotérique qui est déployé sur tous les VTD de tous les VTSS homologues, VTCS peut continuer à influencer l'allocation si l'un des VTSS homologues d'un cluster est mis hors ligne.

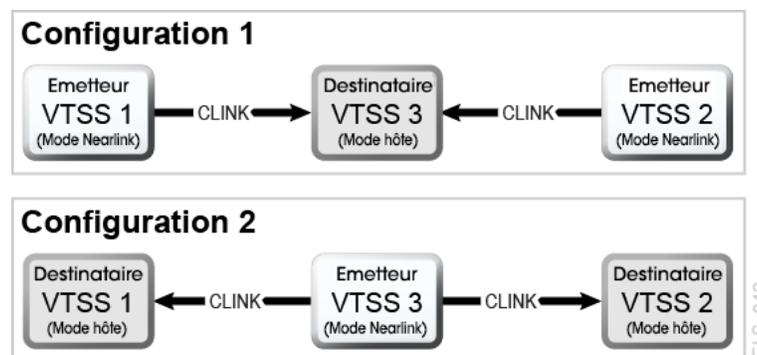
- Pour SMC, un nom de classe de gestion, s'il est affecté dans l'interface DFSMS StorageTek, est disponible lors de l'allocation. Le groupe ésoérique affecté dans l'interface n'a donc plus besoin de contenir uniquement des VTSS appartenant à un cluster. Tant que le groupe ésoérique contient des unités situées sur le VTSS principal d'un cluster pleines fonctions, SMC dispose de suffisamment d'informations pour diriger l'allocation vers une unité qui se trouve sur un VTSS principal, si la classe de gestion indique que la réplication est activée.
8. Consultez vos options HSC PARMLIB pour vérifier que les enregistrements de sous-type 28 sont activés.

Si tel est le cas, le clustering VTSS écrit un enregistrement de sous-type 28 pour chaque réplication effectuée.

Clustering étendu

Le clustering étendu (EC, Extended Clustering) permet de connecter trois VTSS au minimum par le biais de CLINK dans une configuration à un seul Tapeplex (1 CDS), comme illustré dans l'exemple présenté à la [Figure A.6, « Configurations de cluster étendu de base »](#).

Figure A.6. Configurations de cluster étendu de base



Configuration et gestion d'un système en cluster à 3 VTSS

Comme indiqué à la [Figure A.6, « Configurations de cluster étendu de base »](#), la configuration 1 montre 2 VTSS répliqués sur un seul VTSS "collecteur". Cette configuration est la plus pratique, car un emplacement principal avec plusieurs VSM peut placer des VTV dans un emplacement secondaire avec un VSM collecteur unique. Les répliquations synchrone et asynchrone peuvent être utilisées sur chaque VTSS émetteur. Chaque VTSS doit être connecté à des RTD équivalents (comme un modèle). Comme indiqué dans les instructions CONFIG de la configuration 1 suivante :

- l'instruction *CLUSTER* définit tous les noms VTSS configurés pour le clustering ;
- l'instruction *CLINK* définit l'emplacement de port Nearlink sur le VTSS émetteur et son VTSS PARTENAIRE ou de destination.

```
/CREATCFG EXEC PGM=SLUADMIN, PARM='MIXED'  
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR  
//SLSCNTL DD DSN=h1q.DBASEPRM, DISP=SHR  
//SLSCNTL2 DD DSN=h1q.DBASESEC, DISP=SHR  
//SLSSTBY DD DSN=h1q.DBASETBY, DISP=SHR  
//SLSPRINT DD SYSOUT=*  
//SLSIN DD *  
CONFIG RESET CDSLEVEL(V62ABOVE)  
GLOBAL MAXVTV=65000 MVCFREE=60 VTVATTR=SCRATCH RECALWER=YES  
LOCKSTR=STK_VTCS_LOCKS VTVPAGE=LARGE  
RECLAIM THRESHLD=70 MAXMVC=30 START=40 CONMVC=5  
VTSS NAME=VTSS1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5  
RTD NAME=PA11A00 DEVNO=1A00 CHANIF=0C  
RTD NAME=PA11A01 DEVNO=1A01 CHANIF=0D  
RTD NAME=PA11A02 DEVNO=1A02 CHANIF=0K  
RTD NAME=PA11A03 DEVNO=1A03 CHANIF=0L  
RTD NAME=PA12A08 DEVNO=2A08 CHANIF=1C  
RTD NAME=PA12A09 DEVNO=2A09 CHANIF=1D  
RTD NAME=PA12A0A DEVNO=2A0A CHANIF=1K  
RTD NAME=PA12A0B DEVNO=2A0B CHANIF=1L  
VTD LOW=7900 HIGH=79FF  
VTSS NAME=VTSS2 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5  
RTD NAME=PA23A00 DEVNO=3A00 CHANIF=0C  
RTD NAME=PA23A01 DEVNO=3A01 CHANIF=0D  
RTD NAME=PA23A02 DEVNO=3A02 CHANIF=0K  
RTD NAME=PA23A03 DEVNO=3A03 CHANIF=0L  
RTD NAME=PA24A08 DEVNO=4A08 CHANIF=1C  
RTD NAME=PA24A09 DEVNO=4A09 CHANIF=1D  
RTD NAME=PA24A0A DEVNO=4A0A CHANIF=1K  
RTD NAME=PA24A0B DEVNO=4A0B CHANIF=1L  
VTD LOW=8900 HIGH=89FF  
VTSS NAME=VTSS3 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5  
RTD NAME=PA33A00 DEVNO=3A00 CHANIF=0C  
RTD NAME=PA33A01 DEVNO=3A01 CHANIF=0D  
RTD NAME=PA33A02 DEVNO=3A02 CHANIF=0K  
RTD NAME=PA33A03 DEVNO=3A03 CHANIF=0L  
RTD NAME=PA34A08 DEVNO=4A08 CHANIF=1C  
RTD NAME=PA34A09 DEVNO=4A09 CHANIF=1D  
RTD NAME=PA34A0A DEVNO=4A0A CHANIF=1K  
RTD NAME=PA34A0B DEVNO=4A0B CHANIF=1L  
VTD LOW=9900 HIGH=99FF  
CLUSTER NAME=CLUSTER1 VTSS(VTSS1, VTSS2, VTSS3)  
CLINK VTSS=VTSS1 CHANIF=0G PART=VTSS3  
CLINK VTSS=VTSS2 CHANIF=0G PART=VTSS3
```

Comme illustré à la [Figure A.6, « Configurations de cluster étendu de base »](#), la configuration 2 présente un seul VTSS de réplication connecté à 2 VTSS récepteur. Notez que le terme "collecteur" n'est pas utilisé ici, car un VTV est uniquement répliqué sur un VTSS (VTSS1 ou VTSS2) et le VTSS récepteur n'est pas configurable. Il est très important de comprendre ce concept car aucun paramètre de classe de gestion ne sélectionne actuellement un VTSS spécifique pour l'acheminement d'un VTV. Cette configuration n'est pas utile pour une implémentation dans un environnement composé d'emplacements principaux et secondaires, où le VTV doit arriver sur un emplacement secondaire spécifique et risque de rendre les configurations bidirectionnelles étendues peu souhaitables. Les réplications synchrone et asynchrone peuvent être utilisées sur le VTSS émetteur. Chaque VTSS doit être connecté à des RTD équivalents (comme un modèle).

La configuration 2 est préférable si vous décidez d'implémenter une réplication bidirectionnelle dans un environnement de cluster étendu. La réplication bidirectionnelle est requise et utilise alors la configuration de type "plusieurs VTSS vers un VTSS" dans un sens, et "paire de VTSS" dans l'autre, où la "paire de VTSS" est configurée entre les deux VTSS sur lesquels doit se trouver le VTV répliqué.

Comme indiqué dans les instructions *CONFIG* de la configuration 2 suivante :

- l'instruction *CLUSTER* définit tous les noms VTSS configurés pour le clustering ;
- l'instruction *CLINK* définit l'emplacement de port Nearlink sur le VTSS émetteur et son VTSS *PARTENAIRE* ou de destination.

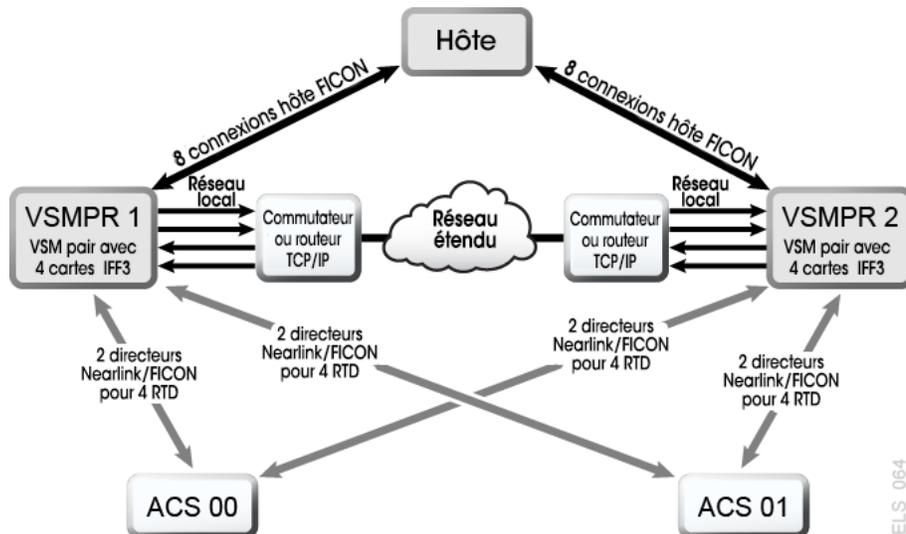
```
//CREATCFG EXEC PGM=SLUADMIN,PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK,DISP=SHR
//SLSCNTL DD DSN=h1q.DBASEPRM,DISP=SHR
//SLSCNTL2 DD DSN=h1q.DBASESEC,DISP=SHR
//SLSSTBY DD DSN=h1q.DBASETBY,DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
CONFIG RESET CDSLEVEL(V62ABOVE)
GLOBAL MAXVTV=65000 MCVFREE=60 VTVATTR=SCRATCH RECALWER=YES
LOCKSTR=STK_VTCS_LOCKS VTVPAGE=LARGE
RECLAIM THRESHLD=70 MAXMVC=30 START=40 CONMVC=5
VTSS NAME=VTSS1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTD NAME=PA11A00 DEVNO=1A00 CHANIF=0C
RTD NAME=PA11A01 DEVNO=1A01 CHANIF=0D
RTD NAME=PA11A02 DEVNO=1A02 CHANIF=0K
RTD NAME=PA11A03 DEVNO=1A03 CHANIF=0L
RTD NAME=PA12A08 DEVNO=2A08 CHANIF=1C
RTD NAME=PA12A09 DEVNO=2A09 CHANIF=1D
RTD NAME=PA12A0A DEVNO=2A0A CHANIF=1K
RTD NAME=PA12A0B DEVNO=2A0B CHANIF=1L
VTD LOW=7900 HIGH=79FF
```

```

VTSS NAME=VTSS2 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTD NAME=PA23A00 DEVNO=3A00 CHANIF=0C
RTD NAME=PA23A01 DEVNO=3A01 CHANIF=0D
RTD NAME=PA23A02 DEVNO=3A02 CHANIF=0K
RTD NAME=PA23A03 DEVNO=3A03 CHANIF=0L
RTD NAME=PA24A08 DEVNO=4A08 CHANIF=1C
RTD NAME=PA24A09 DEVNO=4A09 CHANIF=1D
RTD NAME=PA24A0A DEVNO=4A0A CHANIF=1K
RTD NAME=PA24A0B DEVNO=4A0B CHANIF=1L
VTD LOW=8900 HIGH=89FF
VTSS NAME=VTSS3 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTD NAME=PA33A00 DEVNO=3A00 CHANIF=0C
RTD NAME=PA33A01 DEVNO=3A01 CHANIF=0D
RTD NAME=PA33A02 DEVNO=3A02 CHANIF=0K
RTD NAME=PA33A03 DEVNO=3A03 CHANIF=0L
RTD NAME=PA34A08 DEVNO=4A08 CHANIF=1C
RTD NAME=PA34A09 DEVNO=4A09 CHANIF=1D
RTD NAME=PA34A0A DEVNO=4A0A CHANIF=1K
RTD NAME=PA34A0B DEVNO=4A0B CHANIF=1L
VTD LOW=9900 HIGH=99FF
CLUSTER NAME=CLUSTER1 VTSS(VTSS1,VTSS2,VTSS3)
CLINK VTSS=VTSS3 CHANIF=0G PART=VTSS1
CLINK VTSS=VTSS3 CHANIF=0G PART=VTSS2
    
```

Cluster VSM5 à VSM5 avec CLINK TCP/IP

Figure A.7. VSM5 en cluster avec CLINK TCP/IP



La [Figure A.7, « VSM5 en cluster avec CLINK TCP/IP »](#) présente un exemple de cluster VSM5 à VSM5 avec des CLINK TCP/IP.

Dans la [Figure A.7, « VSM5 en cluster avec CLINK TCP/IP »](#), supposons que vous utilisiez pour la redondance des cibles situées sur des cartes IFF distinctes sur chaque VSM5 pour le protocole IP natif, comme indiqué dans le [Tableau A.1, « Valeurs IPIF CLINK pour VSMPR1 »](#) et l'[Tableau A.2, « Valeurs IPIF CLINK pour VSMPR2 »](#).

Tableau A.1. Valeurs IPIF CLINK pour VSMPR1

| Carte IFF | N° cible | Exemple de protocole IP | Valeur IPIF CLINK correspondante |
|-----------|----------|-------------------------|----------------------------------|
| IFF0 | Cible 0 | 128.0.1.1 | 0A:0 |
| IFF1 | Cible 0 | 128.0.2.1 | 0I:0 |
| IFF2 | Cible 0 | 128.0.3.1 | 1A:0 |
| IFF3 | Cible 0 | 128.0.4.1 | 1I:0 |

Tableau A.2. Valeurs IPIF CLINK pour VSMPR2

| Carte IFF | N° cible | Exemple de protocole IP | Valeur IPIF CLINK correspondante |
|-----------|----------|-------------------------|----------------------------------|
| IFF0 | Cible 0 | 128.0.1.2 | 0A:0 |
| IFF1 | Cible 0 | 128.0.2.2 | 0I:0 |
| IFF2 | Cible 0 | 128.0.3.2 | 1A:0 |
| IFF3 | Cible 0 | 128.0.4.2 | 1I:0 |

L'exemple suivant montre un JCL CONFIG qui définit la configuration présentée à la [Figure A.7, « VSM5 en cluster avec CLINK TCP/IP »](#) avec les valeurs indiquées dans le [Tableau A.1, « Valeurs IPIF CLINK pour VSMPR1 »](#) et le [Tableau A.2, « Valeurs IPIF CLINK pour VSMPR2 »](#).

```
//CREATECF EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSCTL DD DSN=FEDB.VSMLMULT.DBASEPRM, DISP=SHR
//SLSCTL2 DD DSN=FEDB.VSMLMULT.DBASESEC, DISP=SHR
//SLSSTBY DD DSN=FEDB.VSMLMULT.DBASEBY, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
CONFIG CDSLEVEL(V61ABOVE)
GLOBAL MAXVTV=32000 MVCFREE=40 VTVATTR=SCRATCH RECALWER=YES
LOCKSTR=VTCS_LOCKS REPLICAT=ALWAYS VTVPAGE=LARGE INITMVC=YES
SYNCHREP=YES MAXRTDS=16 FASTMIGR=YES
RECLAIM THRESHLD=70 MAXMVC=40 START=35
VTSS NAME=VSMPR1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
VTD LOW=8900 HIGH=89FF
RTD NAME=VPR12A00 DEVNO=2A00 CHANIF=0C:0
RTD NAME=VPR12A01 DEVNO=2A01 CHANIF=0C:1
RTD NAME=VPR12A02 DEVNO=2A02 CHANIF=0C:2
RTD NAME=VPR12A03 DEVNO=2A03 CHANIF=0C:3
RTD NAME=VPR12A04 DEVNO=2A04 CHANIF=0G:0
RTD NAME=VPR12A05 DEVNO=2A05 CHANIF=0G:1
```

```

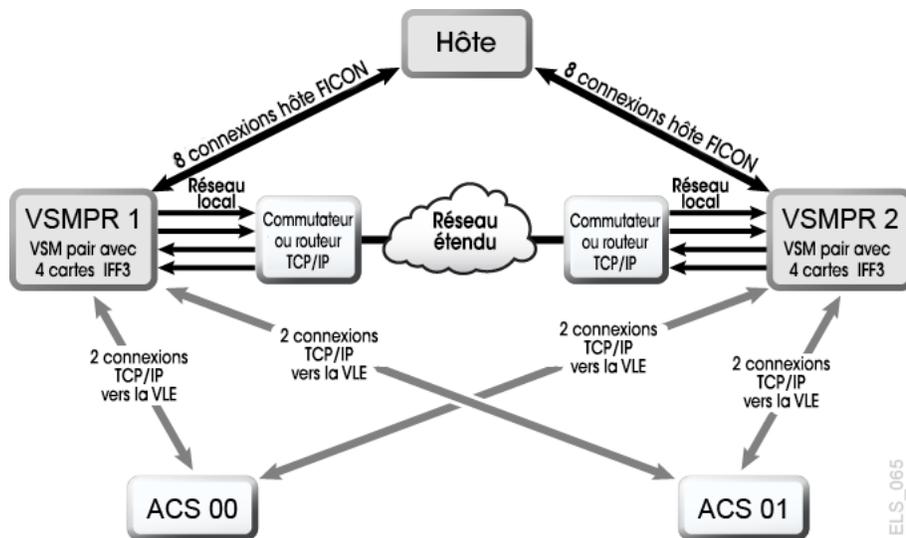
RTD  NAME=VPR12A06  DEVNO=2A06  CHANIF=0G:2
RTD  NAME=VPR12A07  DEVNO=2A07  CHANIF=0G:3
VTSS NAME=VSMPR2  LOW=70  HIGH=80  MAXMIG=8  MINMIG=4  RETAIN=5
VTD  LOW=9900  HIGH=99FF
RTD  NAME=VPR22B00  DEVNO=2B00  CHANIF=0C:0
RTD  NAME=VPR22B01  DEVNO=2B01  CHANIF=0C:1
RTD  NAME=VPR22B02  DEVNO=2B02  CHANIF=0C:2
RTD  NAME=VPR22B03  DEVNO=2B03  CHANIF=0C:3
RTD  NAME=VPR22B04  DEVNO=2B04  CHANIF=0G:0
RTD  NAME=VPR22B05  DEVNO=2B05  CHANIF=0G:1
RTD  NAME=VPR22B06  DEVNO=2B06  CHANIF=0G:2
RTD  NAME=VPR22B07  DEVNO=2B07  CHANIF=0G:3
CLUSTER NAME=CLUSTER1  VTSSs (VSMPR1, VSMPR2)
CLINK VTSS=VSMPR1  IPIF=0A:0
CLINK VTSS=VSMPR1  IPIF=0I:0
CLINK VTSS=VSMPR1  IPIF=1A:0
CLINK VTSS=VSMPR1  IPIF=1I:0
CLINK VTSS=VSMPR2  IPIF=0A:0
CLINK VTSS=VSMPR2  IPIF=0I:0
CLINK VTSS=VSMPR2  IPIF=1A:0
CLINK VTSS=VSMPR2  IPIF=1I:0

```

Cluster VSM5 à VSM6 avec CLINK TCP/IP, VLE interconnectées

La Figure A.8, « Cluster VSM5 à VSM6 avec CLINK TCP/IP, VLE interconnectées » présente un exemple de cluster VSM5 à VSM6 avec des CLINK TCP/IP, où chaque VTSS est interconnecté avec deux VLE.

Figure A.8. Cluster VSM5 à VSM6 avec CLINK TCP/IP, VLE interconnectées



Dans la Figure A.8, « Cluster VSM5 à VSM6 avec CLINK TCP/IP, VLE interconnectées », supposons que vous utilisiez pour la redondance des cibles situées sur des cartes IFF distinctes pour le VSM5 (VSMPR1) pour le protocole IP natif et pour les connexions VLE, comme indiqué dans le Tableau A.3, « Valeurs IPIF CLINK pour VSMPR1 » et le Tableau A.4, « Valeurs IPIF RTD pour VSMPR1 ».

Tableau A.3. Valeurs IPIF CLINK pour VSMPR1

| Carte IFF | N° cible | Exemple de protocole IP | Valeur IPIF CLINK correspondante |
|-----------|----------|-------------------------|----------------------------------|
| IFF0 | Cible 0 | 128.0.1.1 | 0A:0 |
| IFF1 | Cible 0 | 128.0.2.1 | 0I:0 |
| IFF2 | Cible 0 | 128.0.3.1 | 1A:0 |
| IFF3 | Cible 0 | 128.0.4.1 | 1I:0 |

Tableau A.4. Valeurs IPIF RTD pour VSMPR1

| Carte IFF | N° cible | Exemple de protocole IP | Valeur IPIF CLINK correspondante |
|-----------|----------|-------------------------|----------------------------------|
| IFF0 | Cible 1 | 128.0.1.2 | 0A:1 |
| IFF1 | Cible 1 | 128.0.2.2 | 0I:1 |
| IFF2 | Cible 1 | 128.0.3.2 | 1A:1 |
| IFF3 | Cible 1 | 128.0.4.2 | 1I:1 |

L'exemple suivant montre un JCL CONFIG qui définit la configuration présentée à la [Figure A.8, « Cluster VSM5 à VSM6 avec CLINK TCP/IP, VLE interconnectées »](#) avec les valeurs indiquées dans le [Tableau A.3, « Valeurs IPIF CLINK pour VSMPR1 »](#) et le [Tableau A.4, « Valeurs IPIF RTD pour VSMPR1 »](#)..

```
//CREATECF EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLS_CNTL DD DSN=FEDB.VSMLMULT.DBASEPRM, DISP=SHR
//SLS_CNTL2 DD DSN=FEDB.VSMLMULT.DBASESEC, DISP=SHR
//SLSSTBY DD DSN=FEDB.VSMLMULT.DBASETBY, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
CONFIG CDSLEVEL(V61ABOVE)
GLOBAL MAXVTV=32000 MCVFREE=40 VTVATTR=SCRATCH RECALWER=YES
LOCKSTR=VTCS_LOCKS REPLICAT=ALWAYS VTPAGE=LARGE INITMVC=YES
SYNCHREP=YES MAXRTDS=16 FASTMIGR=YES
RECLAIM THRESHLD=70 MAXMVC=40 START=35
VTSS NAME=VSMPR1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
VTD LOW=9900 HIGH=99FF
RTD NAME=VL1RTD1 STORMNGR=VLE1 IPIF=0A:1
RTD NAME=VL1RTD2 STORMNGR=VLE1 IPIF=0I:1
RTD NAME=VL2RTD1 STORMNGR=VLE2 IPIF=1A:1
RTD NAME=VL2RTD2 STORMNGR=VLE2 IPIF=1I:1
CLUSTER NAME=CLUSTER1 VTSSs(VSMPR1, VSMPR2)
CLINK VTSS=VSMPR1 IPIF=0A:0
CLINK VTSS=VSMPR1 IPIF=0I:0
CLINK VTSS=VSMPR1 IPIF=1A:0
CLINK VTSS=VSMPR1 IPIF=1I:0
CLINK VTSS=VSMPR2 IPIF=0A:0
CLINK VTSS=VSMPR2 IPIF=0I:0
CLINK VTSS=VSMPR2 IPIF=1A:0
CLINK VTSS=VSMPR2 IPIF=1I:0
```

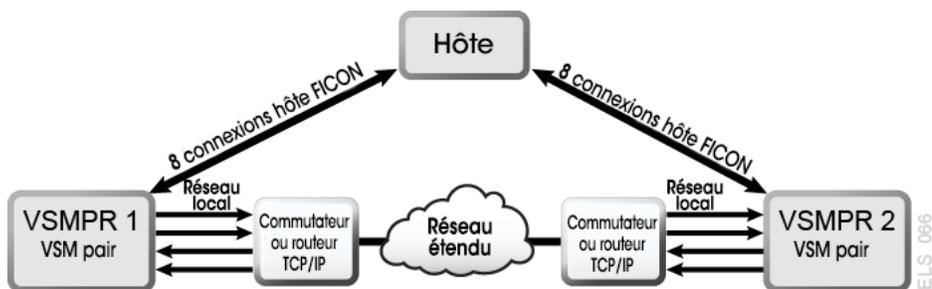
Dans cet exemple, notez qu'alors que les valeurs des paramètres *IPIF CLINK* et *IPIF RTD* pour le VSM5 (VSMPR1) doivent correspondre à celles affichées dans le [Tableau A.3](#),

« Valeurs IPIF CLINK pour VSMPR1 » et le Tableau A.4, « Valeurs IPIF RTD pour VSMPR1 », les valeurs *IPIF CLINK* et *IPIF RTD* pour le VSM 6 (VSMPR2) doivent uniquement se conformer aux restrictions VTCS en vigueur et doivent être uniques pour chaque VTSS ; elles ne correspondent à aucune valeur réelle sur les ports TCP/IP VSM 6.

Cluster "sans bandes" VSM 6 à VSM 6 avec CLINK TCP/IP

La Figure A.9, « Cluster sans bandes VSM6 à VSM6 avec CLINK TCP/IP » présente un exemple de cluster "sans bandes" VSM 6 à VSM 6 avec des CLINK TCP/IP.

Figure A.9. Cluster sans bandes VSM6 à VSM6 avec CLINK TCP/IP



L'exemple suivant montre un JCL CONFIG qui définit la configuration présentée à la Figure A.9, « Cluster sans bandes VSM6 à VSM6 avec CLINK TCP/IP ».

```
//CREATECF EXEC PGM=SLUADMIN, PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//SLSCNTL DD DSN=FEDB.VSMLMULT.DBASEPRM, DISP=SHR
//SLSCNTL2 DD DSN=FEDB.VSMLMULT.DBASESEC, DISP=SHR
//SLSSTBY DD DSN=FEDB.VSMLMULT.DBASETBY, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
CONFIG CDSLEVEL(V61ABOVE)
GLOBAL MAXVTV=32000 MVCFREE=40 VTVATTR=SCRATCH RECALWER=YES
LOCKSTR=VTCS_LOCKS REPLICAT=ALWAYS VTPAGE=LARGE INITMVC=YES
SYNCHREP=YES MAXRTDS=16 FASTMIGR=YES
RECLAIM THRESHLD=70 MAXMVC=40 START=35
VTSS NAME=VSMPR1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
VTD LOW=8900 HIGH=89FF
CLUSTER NAME=CLUSTER1 VTSSs(VSMPR1, VSMPR2)
CLINK VTSS=VSMPR1 IPIF=0A:0
CLINK VTSS=VSMPR1 IPIF=0A:1
CLINK VTSS=VSMPR1 IPIF=0A:2
CLINK VTSS=VSMPR1 IPIF=0A:3
CLINK VTSS=VSMPR2 IPIF=0A:0
CLINK VTSS=VSMPR2 IPIF=0A:1
CLINK VTSS=VSMPR2 IPIF=0A:2
CLINK VTSS=VSMPR2 IPIF=0A:3
```

Dans cet exemple, notez que les valeurs *IPIF CLINK* pour les deux VSM 6 doivent uniquement se conformer aux restrictions VTCS en vigueur et doivent être uniques pour chaque VTSS ; elles ne correspondent pas à une valeur réelle sur les ports TCP/IP VSM 6.

Notez également que le cluster étant sans bandes, il n'existe aucune instruction *CONFIG RTD* pour les VSM 6.

Choix entre un cluster unidirectionnel ou bidirectionnel

Vous pouvez utiliser les instructions *VTSSLST* et *VTSSSEL* pour transformer un cluster bidirectionnel en cluster unidirectionnel. Pourquoi voudriez-vous effectuer cette opération ? Et si vous inversiez les rôles des VTSS principaux et secondaires ? Commencez par utiliser la configuration décrite dans la procédure débutant à la section "[Configuration et gestion d'un système en cluster bidirectionnel](#)". Une fois que vous avez terminé l'étape 5, effectuez une modification à l'aide des instructions *VTSSLST* et *VTSSSEL* ci-dessous.

```
VTSSLST NAME(SITEA) VTSS(VSMPR1)
VTSSSEL FUNCTION(SCRATCH) HOST(MVSA) VTSSLST(SITEA)
VTSSSEL FUNCTION(SPECIFIC) HOST(MVSA) VTSSLST(SITEA)
```

Dans cet exemple :

- L'instruction *VTSSLST* définit la liste de VTSS SITEA qui contient uniquement VSMPR1.
- Les instructions *VTSSSEL* dirigent les montages VTV provisoires et spécifiques de MVSA vers SITEA, qui contient **uniquement** VSMPR1 et devient par conséquent le VTSS principal.

Ce cluster est ainsi vraiment bidirectionnel, mais les instructions *VTSSLST* et *VTSSSEL* nous offrent la flexibilité nécessaire pour que l'un des deux VTSS soit indifféremment le VTSS principal et l'autre le secondaire, simplement en chargeant les instructions de contrôle *MGMTclas*, *STORclas*, *VTSSLST* et *VTSSSEL* correspondantes à l'aide de la commande *MGMTDEF*.

Et si vous inversiez les VTSS principaux et secondaires ? Dans ce cas, réécrivez les instructions *VTSSLST* et *VTSSSEL* pour que VSMPR2 soit le VTSS principal et VSMPR1 le VTSS secondaire.

```
VTSSLST NAME(SITEB) VTSS(VSMPR2)
VTSSSEL FUNCTION(SCRATCH) HOST(MVSB) VTSSLST(SITEB)
VTSSSEL FUNCTION(SPECIFIC) HOST(MVSB) VTSSLST(SITEB)
```

Dans cet exemple :

- L'instruction *VTSSLST* définit la liste SITEB qui contient uniquement VSMPR2.
- Les instructions *VTSSSEL* dirigent les montages VTV provisoires et spécifiques de MVSB vers SITEB, qui contient **uniquement** VSMPR2 et devient par conséquent le VTSS principal.

Enfin, que faire si cela se passe mieux avec ce cluster plutôt qu'avec un véritable cluster bidirectionnel ? Dans ce cas, supprimez les instructions *VTSSLST* et *VTSSSEL* et rechargez vos définitions.

Glossaire

Remarque :

Les entrées du glossaire qui incluent la désignation (I) sont référencées dans le *IBM Dictionary of Computing*.

| | |
|--|--|
| 4410 | Module de stockage de bibliothèque (LSM, Library Storage Module) standard StorageTek d'Oracle |
| 4480 | Transport de cartouches 0,5" et 18 pistes StorageTek d'Oracle. |
| 4490 | Transport de cartouches 36 pistes StorageTek d'Oracle compatible avec ESCON. Egalement appelé Silverton. |
| 9310 | Module de stockage de bibliothèque (LSM, Library Storage Module) StorageTek d'Oracle, version haute performance du LSM 4410 standard. Egalement appelé PowderHorn. |
| 9360 | Module de stockage de bibliothèque (LSM, Library Storage Module) StorageTek d'Oracle. Egalement appelé WolfCreek. |
| 9740 | Module de stockage de bibliothèque (LSM, Library Storage Module) StorageTek d'Oracle. Egalement appelé TimberWolf. |
| ACS | Automated Cartridge System (système de cartouches automatisé) : sous-système de bibliothèque de récupération et de stockage de cartouches constitué d'un ou plusieurs modules de stockage de bibliothèque (LSM) connectés via des Pass-thru Ports. |
| ACSid | Méthode utilisée dans le processus LIBGEN pour identifier les ACS en utilisant des chiffres décimaux allant de 00 à 99. |
| ACSLs | Automated Cartridge System Library Software (logiciel automatisé de la bibliothèque de bandes du système de cartouches) : logiciel de contrôle de la bibliothèque StorageTek d'Oracle qui s'exécute dans le système de contrôle de bibliothèque UNIX®. |
| Adaptateur de chemin de données | Périphérique matériel qui traduit le protocole de données d'un système informatique client dans le protocole de données de l'unité de contrôle ou IMU StorageTek. Le produit TC44-AA/BA STI-to-4400 ACS Interconnect de DEC en est un exemple. |
| adresse | Représentation codée d'un ID matériel, destination ou origine des données. |
| adresse Internet | Système de numérotation utilisé pour spécifier un réseau ou un hôte sur ce réseau pour les communications TCP/IP. Les notations d'adresse Internet standard sont des notations décimales à point. |

| | |
|---|--|
| allocation | Affectation des ressources à une tâche spécifique. |
| allocation dirigée | Voir la hiérarchisation de lecteur. |
| Automated Cartridge System (ACS, système de cartouches automatisé) | Voir ACS . |
| bande de travail | Bande disponible à n'importe quel utilisateur car n'appartenant à personne. |
| base de données de bibliothèque | Fichier ou jeu de données contenant des informations sur l'emplacement et le statut des volumes de médias amovibles, telles que l'emplacement de cellule et le statut provisoire. Egalement nommé jeux de données de contrôle (CDS, Control Data Set). |
| bibliothèque | Voir TapePlex. |
| Bibliothèque ACS | Une bibliothèque consiste en un ou plusieurs systèmes de cartouches automatisés (ACS, Automated Cartridge System), en lecteurs de cartouches rattachés et en cartouches résidant sur les ACS. |
| BISYNC | Communications synchrones binaires. L'un des premiers protocoles de niveau bas développé par IBM et utilisé pour transmettre des données sur un lien de communications synchrones. Il s'agit d'une forme de transmission de données où la synchronisation des caractères est contrôlée par des signaux de calage générés aux stations d'envoi et de réception. |
| câble coaxial | Support de transmission utilisé dans les transmissions de données sur les réseaux utilisant des communications synchrones, par opposition au câble à paire torsadée, support principal utilisé dans les communications RS-232 asynchrones. |
| canal | Périphérique qui connecte l'hôte et le stockage principal aux unités de contrôle des périphériques d'entrée et de sortie. Un canal en duplex intégral comporte deux chemins (2 câbles ou un câble avec des signaux à deux fréquences). Un canal en mode semi-duplex nécessite qu'un port reçoive tandis que l'autre transmet. |
| canal d'utilitaire de couplage | Canal de fibre optique à largeur de bande élevée fournissant la connectivité haute vitesse requise pour le partage des données entre un utilitaire de couplage et les complexes de processeur central qui lui sont directement rattachés.(I) |
| CAPid | Un ID de port CAP définit de manière unique l'emplacement d'un port CAP par le module LSM sur lequel il réside. Un ID de port CAP observe le format "AA:LL:CC", AA:LL étant l'ID du module LSM et CC le numéro de port CAP à deux chiffres. |

| | |
|--|--|
| cartouche | Etui en plastique englobant la bande. Il mesure environ 100 mm par 125 mm par 25 mm. La bande est automatiquement embobinée lorsqu'elle est chargée dans un transport. Une languette en plastique est rattachée à la bande pour l'embobinage automatique. Le dos de la cartouche porte une étiquette OCR/ de code-barre indiquant l'identifiant de volume de bande (VOLSER). |
| cartouche cylindrique | Cartouche de numérisation cylindrique haute capacité pouvant contenir jusqu'à 50 Go de données non compressées. Cette cartouche ne peut être utilisée que sur les transports RedWood (SD-3). |
| cartouche de capacité standard | Cartouche utilisable dans n'importe quel transport longitudinal (4480, 4490, 9490 ou 9490EE). |
| Cartridge Access Port (CAP, port d'accès de cartouche) | Assemblage permettant l'insertion de plusieurs cartouches ou leur éjection d'un module LSM sans intervention humaine dans le LSM. |
| Cartridge Drive (CD, lecteur de cartouche) | Périphérique matériel contenant deux ou quatre transports de cartouches et les blocs d'alimentation électrique et pneumatique associés. |
| cellule | Réceptacle dans le volume LSM où est stockée une cartouche unique. |
| CGI | Common Gateway Interface (interface de passerelle commune). |
| channel-to-channel (CTC, canal à canal) | Désigne la communication (transfert de données) entre des programmes situés à des côtés opposés d'un adaptateur canal à canal.(I) |
| chargement initial | Processus activant une réinitialisation de machine. |
| cliché | Représentation imprimée du contenu du stockage principal à l'instant t. Cette représentation est utilisée à des fins de débogage. |
| client | Utilisateur final des services ACS fournis par le système de contrôle de bibliothèque. |
| Client System Component (CSC, composant système client) | Logiciel fournissant une interface entre le système d'exploitation du système informatique client et le système de contrôle de bibliothèque (LCS, Library Control System) StorageTek. |
| client/serveur | Modèle d'interaction d'un système distribué dans lequel le programme d'un site envoie une requête à un programme sur un autre site et attend une réponse. Le programme à l'origine de la requête est nommé client et le programme répondant à la requête serveur. |
| code-barre | Code consistant en une série de barres dont la largeur varie. Ce code figure sur l'étiquette externe rattachée au dos de la cartouche et équivaut au numéro de série de volume (volser, volume serial number). Ce code est lu par le système de vision de la machine du robot. |

| | |
|---|--|
| commutation de serveur dynamique | Capacité à commuter entre des processeurs de serveurs si une panne de système se produit sur le serveur actif. |
| complexe | Système composé d'autres systèmes, en particulier le système de serveur ACS et le système client. |
| Complexe de bibliothèques | Un complexe de bibliothèques consiste en un CDS (Control Data Set, jeu de données de contrôle) HSC et peut contenir jusqu'à 256 ACS (Automatic Cartridge System, système de cartouches automatisé), chacun pouvant contenir jusqu'à 24 LSM (Library Storage Module, module de stockage de bibliothèque). |
| composant de contrôle de bibliothèque | Logiciel contrôlant le montage et le démontage des cartouches dans un ACS. |
| configuration mixte | Configuration contenant différents types de lecteurs de cartouches dans les modes manuel et de bibliothèque. |
| console | Périphérique d'E/S principal permettant de contrôler une session sur un système. |
| console de processeur LCS | La console de processeur LCS (Library Control System) permet d'utiliser le système d'exploitation de VM (pour le LCS basé sur la VM). |
| console de système MVS | Le MVS/CSC fournit une interface d'opérateur via la console de système MVS. |
| console opérateur | Dans ce document, le terme console opérateur désigne la console client système MVS. |
| Control Path Adaptor (CPA, adaptateur de chemin de contrôle) | Périphérique matériel Bus-Tech, Inc. autorisant les communications entre le canal multiplexeur de bloc d'un processeur hôte et un réseau local. |
| Control Unit (CU, unité de contrôle) | Unité basée sur un microprocesseur et située localement entre un canal et un périphérique d'E/S. Elle traduit les commandes du canal en commandes du périphérique et communique au canal le statut du périphérique. |
| CTC | Channel-to-channel (canal à canal). |
| demande | Terme désignant les commandes exécutées sur l'ACS 4400 pour effectuer une opération de bande. |
| DFSMS | Data Facility Storage Management Subsystem (sous-système de gestion de stockage de données). |
| Direct Access Storage Device (DASD, | Terme IBM désignant un périphérique de stockage de lecteur de disque. |

**périphérique de stockage
d'accès direct)**

ECART

Enhanced Capacity Cartridge (cartouche à capacité améliorée).

**Enhanced Capacity
Cartridge (cartouche à
capacité améliorée)**

Une cartouche a une longueur de 335 mètres et ne peut être utilisée qu'avec des transports 36 pistes (4490, 9490 et 9490EE).

**Enterprise Systems
Connection (ESCON,
connexion des systèmes
d'entreprise)**

Jeu de produits et de services qui fournit un environnement connecté dynamiquement utilisant des câbles optiques et un support de transmission.(I)

**Error Code (EC, code
d'erreur)**

Code numérique affiché par un message et indiquant le type de problème à l'origine d'une erreur.

**Error Recovery Procedure
(ERP, procédure de
récupération après erreur)**

Procédures visant à isoler les erreurs survenant dans l'équipement et, si possible, à se remettre de ces erreurs.

ESCON

Enterprise Systems Connection (connexion des systèmes d'entreprise).

établissement de liaison

Signal de flux de contrôle envoyé par un processus à un autre.

Ethernet

Architecture LAN utilisant une topologie de bus qui permet à une variété d'ordinateurs d'être connectés à une colonne coaxiale blindée commune. L'architecture Ethernet est semblable à la norme IEEE 802.3.

étiquette OCR

Étiquette de reconnaissance optique de caractères. Étiquette externe rattachée au dos d'une cartouche détectable par l'homme et par la machine.

**Event Control Block
(ECB, bloc de contrôle
d'événement)**

Fournit un espace permettant de stocker un code de fin lorsqu'une opération est terminée.

exclusion de lecteur

(auparavant, séparation de périphérique) Désigne la fonction SMC consistant à exclure des lecteurs pour une demande d'allocation en fonction de critères d'exclusion SMC.

fichier

Jeu d'enregistrements associés traités en tant qu'unité.

fichier trace

Fichier contenant des informations utiles au débogage du système.

**File Transfer Protocol
(FTP, protocole de
transfert de fichier)**

Commande TCP/IP fournissant un moyen de transférer des fichiers entre des machines connectées via TCP/IP.

| | |
|---|--|
| Fin de tâche anormale (abend, abnormal end of task) | Problème logiciel ou matériel qui met fin à une tâche de traitement de l'ordinateur. |
| Go | 1 073 741 834 octets de stockage |
| hiérarchisation de lecteur | (auparavant, allocation dirigée) Désigne la fonction SMC consistant à influencer la sélection d'un lecteur spécifique en fonction de critères d'allocation, dont l'emplacement de volume. |
| HTTP | Hypertext Transfer Protocol (protocole de transfert hypertexte). |
| ICRC | Capacité d'enregistrement de cartouche améliorée. Fonction de compression et de compactage qui augmente la quantité de données pouvant être stockées sur une cartouche 0,5". |
| ID utilisateur | Parfois nommé ID utilisateur de VM, l'ID utilisateur est le nom qui identifie un utilisateur ou un client de machine virtuelle spécifique. |
| IEEE 802.3 | Norme standard produit par l'IEEE et acceptée mondialement pour les réseaux LAN utilisant CSMA/CD (Carrier Sense Multiple Access with Collision Detection). |
| Interactive Storage Management Facility (utilitaire de gestion de stockage interactif) | Série d'applications permettant de définir des groupes et des classes de stockage DFSMS/MVS. |
| Internet | Ensemble de réseaux utilisant le protocole TCP/IP et opérant comme un réseau virtuel. |
| Internet Protocol (IP, protocole Internet) | Description formelle des messages et des stratégies utilisée par deux réseaux pour échanger des messages. |
| ISMF | Interactive Storage Management Facility (utilitaire de gestion de stockage interactif) |
| JES | Job Entry Subsystem (sous-système d'entrée de tâche).(I) |
| JES2 | Sous-système MVS qui reçoit des tâches du système, les convertit au format interne, les sélectionne en vue de les exécuter, traite leur sortie et les purge du système. Dans une installation comptant plusieurs processeurs, chaque processeur JES2 contrôle indépendamment l'entrée et la planification de ses tâches ainsi que le traitement des sorties. Voir aussi JES3.(I) |
| JES3 | Sous-système MVS qui reçoit des tâches du système, les convertit au format interne, les sélectionne en vue de les exécuter, traite leur sortie et les purge du système. Dans les complexes comptant plusieurs unités de traitement |

| | |
|--|---|
| | étroitement couplées, le programme JES3 gère des processeurs afin que le processeur global exerce un contrôle centralisé sur les processeurs locaux et qu'il leur distribue des tâches à l'aide d'une file d'attente de tâches commune. Voir aussi JES2.(I) |
| jeu de données | Jeu d'enregistrements traités en tant qu'unité. |
| jeux de données de contrôle (CDS, Control Data Set) | Jeu de données utilisé par le logiciel de l'hôte pour contrôler les fonctions de la bibliothèque de bandes automatisée. Egalement appelé base de données de bibliothèque. |
| Job Control Language (JCL, langue de contrôle des tâches) | Langage axé sur les problèmes conçu pour décrire les exigences de traitement d'une tâche à un système d'exploitation. |
| LAN | Local Area Network (réseau local) : réseau englobant une zone géographique (locale) restreinte |
| LAN synchrone | Réseau local construit sur des communications synchrones. |
| LCS | Library Control System (LCS, système de contrôle de bibliothèque) |
| LCU | Voir l'unité de contrôle de bibliothèque. |
| lecteur d'E/S de bande de cartouche | Logiciel de système d'exploitation émettant des commandes (par exemple, de lecture, d'écriture et de rembobinage) aux sous-systèmes de cartouches. Il s'agit du point focal logiciel pour rattacher un type particulier d'unité de contrôle. (Le produit StorageTek CARLIB d'Oracle en est un exemple.) |
| lecteur de bande | Périphérique de traitement de bande consistant en jusqu'à quatre transports dans une armoire. Un lecteur peut désigner un transport individuel. |
| lecteur de bibliothèque | Lecteur de cartouche dans ACS, qui se distingue d'un lecteur de cartouche autonome. |
| LIBGEN | Processus consistant à définir la configuration d'une bibliothèque sur le VM/HSC. |
| Library Control System (LCS, système de contrôle de bibliothèque) | Plate-forme de contrôle de bibliothèque et logiciel de contrôle de bibliothèque. |
| Library Control Unit (LCU, unité de contrôle de bibliothèque) | Portion d'un LSM qui contrôle les mouvements du robot. |
| Library Management Unit (LMU, unité de gestion de bibliothèque) | Voir LMU . |

| | |
|---|---|
| Library Storage Module (LSM, module de stockage de bibliothèque) | Voir LSM . |
| lien client | Lien de communications entre le LCS et un client. |
| LMU | Unité de gestion de bibliothèque : produit matériel et logiciel coordonnant les activités d'un ou plusieurs LSM/LCU. |
| LMU de secours | LMU redondante dans une configuration LMU double prête à prendre la suite en cas de panne de la LMU maître ou lorsque le processeur émet une commande SWITCH. |
| LMU maître | LMU contrôlant actuellement le fonctionnement de l'ACS dans une configuration LMU double. |
| local area network (LAN, réseau local) | Voir LAN . |
| logical port (LP, port logique) | Voir LP . |
| Logiciel automatisé de la bibliothèque de bandes du système de cartouches (ACSL) | Voir ACSL . |
| logiciel de contrôle de bibliothèque | Composant de contrôle de bibliothèque, interface client système et utilitaires de bibliothèque. |
| LP | Port logique, logiciel CLS faisant l'interface avec le système client. Le CLSLP est l'un des composants logiciels utilisés pour transmettre les données entre le système client et le VM/HSC. |
| LSM | Module de stockage de bibliothèque, LSM (4410) standard, une structure à douze côtés comportant un espace de stockage pour jusqu'à environ 6 000 cartouches. Doté également d'une fonction de vision assistée qui déplace les cartouches entre leurs cellules de stockage et les transports liés. Voir aussi PowderHorn, SL3000, SL8500 et WolfCreek. |
| LSMid | Un LSMid se compose du ACSid accompagné (concaténé avec) du numéro LSM. |
| LUM double | Fonctionnalité matérielle/de microcode fournissant une capacité LMU redondante. |
| mode automatique | Relation entre un LSM et tous les hôtes rattachés. Les LSM opérant en mode automatique gèrent les cartouches sans l'intervention de l'opérateur. Il s'agit |

| | |
|--|--|
| | du mode de fonctionnement normal d'un LSM ayant été modifié en ligne. La situation opposée est le "mode normal". Voir le mode manuel. |
| mode connecté | Relation entre un hôte et un ACS. Dans ce mode, l'hôte et un ACS sont capables de communiquer (en ce sens qu'au moins une station de cet ACS est en ligne). |
| mode de bibliothèque | Fonctionnement d'un sous-système de cartouches 4480 dans le cadre d'un système de cartouches automatisé 4400, par opposition au mode manuel, dans lequel l'opérateur insère les cartouches dans les transports. Voir le mode manuel. |
| mode déconnecté | Relation entre un hôte et un ACS. Dans ce mode, l'hôte et l'ACS sont incapables de communiquer (aucune station de cet ACS n'est en ligne). |
| mode manuel | Fonctionnement d'un lecteur de cartouche en dehors d'ACS. Voir mode de bibliothèque. |
| modem | Périphérique permettant de transmettre des données numériques via un utilitaire de transmission analogique. |
| multi client | Environnement dans lequel plusieurs systèmes clients (homogènes ou hétérogènes) sont connectés à un LCS. |
| nom de groupe ésotérique | Nom affecté aux transports ayant le même type de périphérique. |
| notation décimale à point | Représentation syntactique d'un nombre entier à 32 bits composé de quatre chiffres à 8 bits écrits en base 10 et séparés par des points. Dans les descriptions TCP/IP, la notation décimale à point est utilisée pour les adresses Internet. |
| numéro de connexion | Identifiant unique sur le serveur pour un chemin de communications. Le numéro est assigné par TCP/IP pour identifier la connexion unique entre le nœud de serveur et un port spécifique sur le serveur et entre le nœud de client et un port spécifique sur le client. Le numéro de connexion n'existe que si la connexion existe. |
| numéro de périphérique | Numéro hexadécimal à quatre chiffres identifiant de manière unique un périphérique rattaché à un processeur. |
| numéro LSM | Méthode utilisée pour identifier un LSM. Un numéro LSM est le résultat de la définition du paramètre LSM de la macro SLIACS pendant une opération LIBGEN. Le premier LSM répertorié dans ce paramètre acquiert le numéro LSM 00 (décimal), le second LSM répertorié acquiert le numéro 01, etc., jusqu'à ce que tous les LSM soient identifiés (jusqu'à un maximum de 99). |
| operating system (OS, système d'exploitation) | Logiciel qui contrôle l'exécution des programmes et qui facilite le fonctionnement général du système. |

| | |
|---|--|
| ordinateur hôte | Ordinateur contrôlant un réseau d'ordinateurs. |
| package préconfiguré | Package de serveur logiciel incluant tous les paramètres matériels, logiciels et de configuration livrés par le fournisseurs. |
| panneau de lecteurs | Mur LSM contenant des transports de bandes. Le panneau de lecteurs d'un transport T9840 peut contenir 10 ou 20 transports. Le panneau de lecteurs d'un transport non T9840 peut contenir jusqu'à 4 transports. |
| partage de données | Capacité de sous-systèmes ou d'applications s'exécutant simultanément à consulter et modifier directement les même données tout en maintenant l'intégrité de ces données.(I) |
| Pass-thru Port (PTP, port pass-thru) | Mécanisme permettant à une cartouche de passer d'un LSM à un autre dans un ACS à plusieurs LSM. |
| permutation | Prise en charge de la fonction LMU maître par la LMU standard. |
| plate-forme de contrôle de bibliothèque | Matériel et logiciel fournissant au système de contrôle de bibliothèque l'environnement approprié. |
| port local | Désignation d'une application ou d'un processus donné parmi de nombreux d'autres disponibles pour un processeur hôte TCP/IP. |
| port physique | Logiciel de communications nécessaire pour prendre en charge une liaison serveur/client. |
| PowderHorn (9310) | Version hautes performances du LSM standard. |
| préférence de périphérique | Processus consistant à préférer un type de transport 36 pistes à un autre type de transport 36 pistes. |
| processeur de contrôle de bibliothèque | Matériel informatique correctement configuré qui prend en charge le fonctionnement du système de contrôle de bibliothèque. |
| product change request (PCR, demande de changement de produit) | Demande d'amélioration d'un produit. Cette demande provient normalement d'un client mais peut également provenir d'Oracle. |
| Program Temporary Fix (PTF, correctif temporaire de programme) | Version logicielle conçue pour corriger un ou plusieurs défauts. |
| program update tape (PUT, bande de mise à jour de programme) | Une ou plusieurs bandes contenant des mises à jour ou de nouvelles versions du logiciel MVS/CSC. |
| protocole | Description formelle de formats de messages et stratégies que deux machines ou plus doivent observer pour échanger ces messages. |

| | |
|-----------------------------------|--|
| récupération | Procédures automatiques ou manuelles permettant de résoudre les problèmes survenant sur un système de serveur. |
| reel-id | Identificateur d'un volume de bande spécifique. Equivaut au numéro de série de volume (VOLSER, Volume Serial Number). |
| SD-3 | Transport de cartouches cylindrique de StorageTek d'Oracle. Cette fonctionnalité est également connue sous le nom de RedWood. |
| séparation de périphérique | Voir exclusion de lecteur . |
| serveur | Système de contrôle de bibliothèque ELS tel qu'un HSC. Dans le SMC, un serveur est représenté par un chemin SERVER nommé vers un TAPEPLEX nommé. Bien que le composant de serveur SMC HTTP soit requis en tant que middleware sur l'hôte distant, le serveur, en ce qui concerne le SMC, est le système de contrôle de bibliothèque ELS opérant sur l'hôte distant. |
| serveur de stockage | Jeu de produits matériels et logiciels conçus pour permettre à des systèmes informatiques hétérogènes d'utiliser des services de bibliothèques de cartouches de bandes automatisées. |
| SL3000 | La bibliothèque modulaire StorageTek SL3000 d'Oracle offre des capacités de prise en charge des médias mixtes, de partitionnement physique et logique, de gestion avancée et une haute disponibilité. Elle prend en charge les environnements mixtes, y compris les systèmes mainframe et ouverts. Egalement modulable, cette bibliothèque peut prendre en charge de 200 à 6 000 emplacements de cartouches. |
| SL8500 | La bibliothèque modulaire StorageTek SL8500 d'Oracle offre des capacités de prise en charge des médias mixtes, de partitionnement physique et logique, de gestion avancée ainsi qu'une haute capacité et disponibilité. Il prend en charge les environnements mixtes, y compris les systèmes mainframe et ouverts. Ce système est également modulable et peut prendre en charge de 1 450 à 100 880 emplacements de cartouches dans une configuration complexe. |
| socket | Adresse unique sur un réseau accompagnée d'une adresse de nœud et de l'ID d'une application spécifique sur un réseau spécifique. Abstraction utilisée par le protocole TCP/IP. |
| socket étranger | Un point parmi deux points d'extrémité dans un protocole orienté sur une connexion TCP/IP. Spécifie l'adresse d'un hôte étranger pouvant se connecter au serveur. |
| socket local | Combinaison d'adresse d'une adresse de réseau d'un hôte TCP/IP et port spécifique d'un processus d'application. |

| | |
|---|---|
| sous-pool de bandes de travail | Sous-jeu défini de toutes les bandes de travail. Les sous-pools sont constitués d'une ou plusieurs plages de numéros de série de volumes présentant des caractéristiques physiques similaires (type de volume – bobine ou cartouche, taille de bobine, longueur, emplacement physique, etc.). Certaines installations peuvent aussi sous-diviser leurs pools de travail par d'autres caractéristiques telles que le type d'étiquette. |
| station | Chemin matériel entre l'ordinateur hôte et une LMU par lequel la VM/le HSC et la LMU envoient des informations de contrôle. |
| stockage virtuel | Fonctionnalité du système d'exploitation où les exigences de stockage principales sont allouées par segments (ou pages) selon les besoins des programmes, permettant ainsi la création d'un stockage virtuel ou illimité. |
| Storage Management Component (SMC, composant de gestion de stockage) | Interface logicielle entre le système d'exploitation z/OS d'IBM et les systèmes de contrôle de bibliothèque automatisée StorageTek, HSC et MVS/CSC. Le SMC procède au traitement des allocations, à la gestion des messages et au traitement des SMS pour la solution ELS. |
| synchrone | Voir BISYNC . |
| système client | Système auquel le LCS fournit une interface vers un système de cartouches automatisé StorageTek. |
| Systems Network Architecture (SNA, architecture de réseau de systèmes) | Description de la structure logique, des formats, protocoles et séquences du point de vue fonctionnel, à des fins de transmission des unités d'informations et de contrôle de la configuration et du fonctionnement des réseaux. |
| T10000A | Transport de cartouches haute capacité StorageTek d'Oracle orienté capacité permettant la lecture et l'écriture de cartouches T10000A de 120 Go ou 500 Go. |
| T10000B | Transport de cartouches haute capacité StorageTek d'Oracle orienté capacité permettant la lecture et l'écriture de cartouches T10000B de 240 Go ou 1 To. |
| T10000C | Le lecteur de bande natif haute capacité/vitesse StorageTek T10000C d'Oracle, capable de fournir jusqu'à 252 Mo/sec et 5 To, est idéal pour les opérations des centres de données dont les volumes sont en constante augmentation. |
| T10000D | Le lecteur de bande haute capacité/vitesse StorageTek T10000D d'Oracle, capable de fournir jusqu'à 252 Mo/sec et 8,5 To et de capacité native, est idéal pour les opérations des centres de données dont les volumes sont en constante augmentation. |
| T9840A | Transport de cartouches StorageTek d'Oracle orienté accès permettant la lecture et l'écriture de cartouches 9840A. |

| | |
|--|--|
| T9840B | Transport de cartouches StorageTek d'Oracle orienté accès permettant la lecture et l'écriture de cartouches T9840B. |
| T9840C | Transport de cartouches StorageTek d'Oracle orienté accès permettant la lecture et l'écriture de cartouches T9840C. |
| T9840D | Transport de cartouches StorageTek d'Oracle orienté accès permettant la lecture et l'écriture de cartouches T9840D. |
| T9940A | Transport de cartouches StorageTek d'Oracle orienté capacité permettant la lecture et l'écriture de cartouches T9940A de 60 Go. |
| T9940B | Transport de cartouches StorageTek d'Oracle orienté capacité permettant la lecture et l'écriture de cartouches T9940B de 200 Go. |
| tape library management system (TLMS, système de gestion de bibliothèque de bandes) | Dans ce document, TLMS désigne un système de gestion de bibliothèque de bandes et non pas à CA-1. |
| TapePlex | (auparavant, bibliothèque) Configuration matérielle StorageTek simple, généralement représentée par un CDS (Control Data Set, jeu de données de contrôle) HSC unique. Un TapePlex peut contenir plusieurs ACS (Automated Cartridge System, système de cartouches automatisé) et VTSS (Virtual Tape Storage Subsystem, sous-système de stockage de bandes virtuel). |
| TCP/IP | Transmission Control Protocol/Internet Protocol (protocole de contrôle de transmission/protocole Internet). |
| transaction | Jeu d'entrées spécifique qui déclenche l'exécution d'un processus spécifique. |
| transmission asynchrone | Transmission de données orientée sur les caractères (distincte de la transmission en mode bloc IBM). |
| Transmission Control Protocol (TCP, protocole de contrôle de transmission) | Protocole standard inter réseau offrant un service de flux en duplex intégral. |
| transport | Périphérique électromécanique qui permet de traiter, de positionner et de lire ou d'écrire sur une bande. |
| transport de cartouche de bibliothèque | Voir le transport. |
| type d'événement de trace | Types d'événements tracés à travers le système lorsque la fonction de trace est activée. |

| | |
|---|---|
| UCB | Unit Control Block (bloc de contrôle d'unité). |
| utilitaire | Programme exécutant une fonction auxiliaire à la ou aux fonctions principales d'un système informatique. |
| utilitaire de couplage | Partition logique spéciale fournissant des fonctions de mise en cache haute vitesse, de traitement de liste et de verrouillage dans un sysplex.(I) |
| veille | Statut d'une station ayant pu basculer sur "en ligne" mais qui est connectée à la LMU de secours d'un ACS LMU double. |
| virtual machine (VM, machine virtuelle) | Voir VM . |
| Virtual Storage Manager (VSM, gestionnaire de stockage virtuel) | Solution de stockage permettant de virtualiser des volumes et des transports dans le tampon d'un VTSS afin d'améliorer l'utilisation des médias et des transports. |
| Virtual Tape Control System (VTCS, système de contrôle de bandes virtuel) | Code d'hôte principal de la solution VSM (Virtual Storage Manager, gestionnaire de stockage virtuel). Ce code opère dans un espace d'adressage séparé, mais communique étroitement avec le HSC. |
| Virtual Tape Storage Subsystem (VTSS, sous-système de stockage de bandes virtuel) | Tampon du DASD contenant des volumes virtuels (VTV) et des lecteurs virtuels (VTD). Le VTSS est un périphérique matériel StorageTek RAID 6 avec microcode permettant l'émulation de transport. Le périphérique RAID peut lire et écrire des données de bande depuis/vers un disque et lire et écrire des données depuis/vers un RTD (Real Tape Drive, lecteur de bande physique). |
| Virtual Telecommunications Access Method (VTAM, méthode d'accès aux télécommunications virtuelles) | Logiciel de communications résidant sur un hôte IBM et faisant office d'interface commune pour les communications. |
| VM | Machine virtuelle, simulation fonctionnelle d'un ordinateur et de ses périphériques associés. Chaque machine virtuelle est contrôlée par un système d'exploitation adapté. |
| VM/SP ou VM/XA | Système d'exploitation propriétaire d'IBM constitué principalement de deux composants majeurs, CP et CMS. |
| volume | Cartouche de bande (porteuse de données) montée ou démontée en tant qu'unité. |
| volume physique | Unité physiquement liée de média de fichier de données. Voir cartouche. |

**volume serial number
(VOLSER, numéro de
série de volume)**

Identificateur d'un volume physique.

WolfCreek (9360)

LSM hautes performances ayant une capacité plus faible que celle du LSM standard.

ZCART

Cartouche améliorée étendue utilisant un média plus fin pour fournir deux fois la capacité de la cartouche à capacité améliorée (ECART). Cette cartouche a une longueur de 770 mètres et ne peut être utilisée qu'avec des transports 36 pistes TimberLine 9490EE.

Index

A

- Archivage sécurisé de MVC pour la conservation à long terme, 61
- Archivage sécurisé de MVC pour la récupération après sinistre, 51
- Archivage sécurisé externe d'ELS
 - Préparation, 47

B

- Bibliothèque distante
 - Configuration, 83
 - Considérations relatives au pool de MVC, 85
 - Mise à jour de la console CONFIG du VTCS pour définir une bibliothèque distante, 84
 - Modification du fichier SCMDS du SMC, 83

C

- CDRT
 - Considérations relatives aux métadonnées, 102
 - Exécution d'un test DR, 111
 - Gestion des ressources, 104
 - Nettoyage après un test DR, 113
 - Optimisation de l'accès aux ressources de test et de production, 111
 - Origine des données VTV, 102
 - Scénarios opérationnels, 115
 - Stratégies VTCS, 110
 - Utilisation du logiciel, 101
- Cluster bidirectionnel
 - Exemple de configuration et de gestion, 151
 - Exemple de passage à un cluster unidirectionnel, 163
- Cluster unidirectionnel
 - Exemple de configuration et de gestion, 145
 - Exemple de VTSS, 143
- Clustering étendu, 96
- Clusters bidirectionnels, 94
 - Exemple de VTSS, 148
- Clusters unidirectionnels, 93
- Configuration d'une bibliothèque distante, 83
- Configurations VTSS en cluster
 - Bidirectionnel, 94

Clustering avec connexions TCP/IP, 99

Clustering étendu, 96

Conditions requises, 88

Définition, 87

Fonctionnement, 90

Réplication synchrone ou asynchrone, 97

Unidirectionnel, 93

Utilisation, 87

Considérations relatives au pool de MVC pour les bibliothèques distantes, 85

D

Définition de l'objectif de délai de récupération (RTO), 20

Définition de l'objectif du point de récupération (RPO), 21

E

Ejection de volumes spécifiques vers un espace d'archivage sécurisé local (sol), 61

Exécution d'une réelle récupération après sinistre, 29

Exemple de cluster sans bandes VSM6 à VSM6 avec CLINK TCP/IP, 162

Exemple de cluster VSM5 à VSM5 avec CLINK TCP/IP, 158

Exemple de cluster VSM5 à VSM6 avec CLINK TCP/IP, VLE interconnectées, 160

Exemple de clustering étendu, 155

Exportation et importation de MVC par classe de gestion, 38

Exportation et importation de MVC par classe de stockage, 41

Exportations et importations physiques

- Introduction, 37

G

Gestion des ressources CDRT, 104

H

Haute disponibilité des données (D-HA)

Bande physique hautement disponible, 25

Bande virtuelle hautement disponible, 26

Planification, 24

I

Implémentation de la réplication synchrone, 97
Implémentation de la réplication synchrone avec surveillance des travaux, 98
Importation et exportation de MVC par classe de gestion, 38
Importation et exportation de MVC par classe de stockage, 41

M

Mise à jour de la console CONFIG du VTCS pour définir une bibliothèque distante, 84
Mise en relation d'un RPO avec la récupération du point de synchronisation, 23
Modification du fichier SCMDS du SMC, 83

P

Planification de la haute disponibilité des données (D-HA), 24
Planification de tests DR, 30
Points de récupération système
 Création dans des environnements VSM, 129
Points de reprise
 Création dans des environnements VSM, 129
 Exemples, 130
Préparation de l'archivage sécurisé externe d'ELS, 47

R

Récupération après sinistre
 Archivage sécurisé de MVC, 51
 Archivage sécurisé de MVC pour la conservation à long terme, 61
 Définition de l'objectif de délai de récupération (RTO), 20
 Définition de l'objectif du point de récupération (RPO), 21
 Exécution avec VLE, 136
 Exécution d'une réelle récupération après sinistre, 29
 Introduction, 19
 Mise en relation d'un RPO avec la récupération du point de synchronisation, 23
 Nettoyage avec VLE, 140
 Planification de tests DR, 30
 Récupération du point de synchronisation, 22

 Tests à l'aide de la fonction CTR, 79
 Tests DR avec CDRT, 33
 Tests DR avec exportation et importation physiques, 31
 Tests DR avec fonction CTR (Cross-Tapeplex Replication) de VSM, 35
 tests en l'absence de partitions LPAR sur le site DR, 80
 Traitement des pannes temporaires, 22
 Utilisation de la réplication entre TapePlex (CTR, Cross-Tapeplex Replication), 65
Récupération du point de synchronisation, 22
Réplication asynchrone
 Implémentation avec surveillance des travaux, 98
Réplication entre TapePlex (CTR, Cross-Tapeplex Replication)
 Configuration, 68
 Configuration et démarrage, 68
 Considérations relatives aux VTV en lecture seule, 67
 Définition de stratégies, 71
 Description, 65
 Gestion des VTV répliqués par CTR, 81
 Tests de récupération après sinistre, 79
 Utilisation en l'absence de partitions LPAR sur le site distant, 74
 Utilisation pour la continuité de l'activité, 76
 Utilisation pour la reprise de l'activité, 78
Réplication synchrone
 Implémentation, 97

S

Stratégies VTCS pour CDRT, 110
Système VTSS en cluster
 Exemple de configuration et de gestion, 155

T

Tests DR avec CDRT, 33
Tests DR avec exportation et importation physiques, 31
Tests DR avec fonction CTR (Cross-Tapeplex Replication) de VSM, 35
Traitement des pannes temporaires, 22

U

- Utilisation de configurations VTSS en cluster, 87
- Utilisation de la réplication entre TapePlex dans une solution de récupération après sinistre, 65
- Utilisation de VLE pour la récupération après sinistre, 135

V

- VLE
 - Exécution d'un test DR, 136
 - Mode de production normale, 135
 - Nettoyage après un test DR, 140
 - Utilisation pour la continuité de l'activité, 142
 - Utilisation pour la récupération après sinistre, 135
- VTSS en cluster
 - Exemples, 143

