Sun Java™ System

Portal Server 6
配備計画ガイド

2005Q4

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819-4618

目 次

図目次

表目次

はじめに

この『配備計画ガイド』では、Sun Java™ System Portal Server 6 2005Q4 ソフトウェアの計画と配備方法について説明します。Portal Server は、組織の統合データ、知識管理、およびアプリケーションのポータルを作成するプラットフォームです。Portal Server プラットフォームは、企業間、企業対社員、および企業対顧客を始めとするすべての種類のポータルを構築および配備するための総合的なインフラストラクチャーソリューションを提供します。

お読みになる前に

Portal Server は、ネットワークまたはインターネット環境全体に分散したエンタープライズアプリケーションをサポートするソフトウェアインフラストラクチャーである Sun Java Enterprise System のコンポーネントです。Sun Java Enterprise System のマニュアルに精通してください。マニュアルは、http://docs.sun.com/db/prod/entsys?l=ja にアクセスしてオンラインで入手できます。

対象読者

この『配備計画ガイド』は、Portal Server を配備する責任者を対象としています。

Portal Server の配備を実施する前に、次のテクノロジを理解しておく必要があります。

- Sun Java Enterprise System
- Solaris™ オペレーティングシステムの管理手順
- Sun Java System Access Manager
- Sun Java System Directory Server
- Java™ Web Server

- JavaServer Pages™ テクノロジ
- LDAP (Lightweight Directory Access Protocol)
- HTML (Hypertext Markup Language)
- XML (Extensible Markup Language)

本書の構成

第 1 章から第 5 章では、Portal Server の配備について説明します。次の表に本書の内容をまとめます。

章	説明
21 ページの第 1 章「Portal Server のアーキテクチャー」	この章では、ポータルサーバーのタイプ、オープンモードとセキュアモードの Sun Java System Portal Server、Portal Server コンポーネントについて説明します。
37 ページの第 2 章「Portal Server Secure Remote Access アーキテクチャー」	この章では、企業イントラネットの外部から内部のリソースへのセキュリティー保護されたリモートアクセスを実現するために Secure Remote Access の主要なコンポーネントが果たす役割など、Portal Server Secure Remote Access アーキテクチャーについて説明します。
51 ページの第 3 章「ビジネス要件と技術要件の特定と評価」	この章では、ポータルの配備を設計する前の組織の必要と要件の分析方法について説明します。
63 ページの第 4 章「配備前の注意点」	この章では、ポータルの基準サイズの確立方法について説明します。基準サイズが確立されれば、その数値を微調整して、スケーラビリティー、高可用性、信頼性、および良好なパフォーマンスを実現できます。
79 ページの第 5 章「ポータルの設計」	この章では、ポータルの高レベルおよび低レベルの設計を行う方法について説明し、設計計画の各部分の設計に必要な情報を提供します。
131 ページの第 6 章「本稼働環境」	この章では、ポータルの調整および監視方法について説明します。
137 ページの付録 A 「インストールされた製品のレイアウト」	この付録では、Portal Server および Sun Java System Portal Server Secure Remote Access (SRA) のディレクトリと設定ファイルについて説明します。
141 ページの付録 B 「分析ツール」	この付録では、オペレーティングシステムを調整する場合の分析ツールについて説明します。

章	説明
151 ページの付録 C「Portal Server とアプリケーションサーバー」	この付録では、アプリケーションサーバーのサポート体制について説明します。
157 ページの付録 D「ポータルの配備の障害追跡」	この付録では、Portal Server ソフトウェアと Portal Server Secure Remote Access (SRA) 製品の問題を解決する方法について説明します。
165 ページの付録 E「ポータル配備ワークシート」	この付録では、開発プロセスで役立つさまざまなワークシートを紹介します。
177 ページの付録 F「Linux プラットフォームの Portal Server」	この付録では、Portal Server を Linux プラットフォームで実行するための注意事項について説明します。
用語集	用語集

表記上の規則

このセクションの表に本書で使用される表記上の規則をまとめます。

表記上の規則

次の表に、本書で使用される表記上の規則を示します。

表 1　表記上の規則

書体	意味	例
AaBbCc123 (モノスペース)	API および言語要素、HTML タグ、Web サイトの URL、コマンド名、ファイル名、ディレクトリのパス名、画面上のコンピュータ出力、サンプルコードです。	`.login` ファイルを編集します。 `ls -a` を使用してすべてのファイルをリスト表示します。 `% You have mail.`
AaBbCc123 (モノスペース太字)	画面のコンピュータ出力と区別する場合のユーザーの入力です。	**`% su`** `Password:`
AaBbCc123 (斜体)	実際の名前または値で置き換えられる、コマンド名またはパス名の可変部分です。	ファイルは、*install-dir*`/bin` ディレクトリに格納されています。

関連ドキュメント

Sun の技術文書には、http://docs.sun.com/app/docs?l=ja からオンラインでアクセスできます。アーカイブを参照するか、個々の書名または件名を検索できます。

このドキュメントセットのマニュアル

次の表に、Portal Server コアドキュメントセットを構成するマニュアルを示します。

マニュアルタイトル	説明
『Portal Server 管理ガイド』 http://docs.sun.com/app/docs/doc/819-1198	Access Manager 管理コンソールとコマンド行による Portal Server 6 の管理方法を説明しています。
『Portal Server Secure Remote Access 管理ガイド』 http://docs.sun.com/app/docs/doc/819-1202	Portal Server 6 Secure Remote Access の管理方法を説明しています。
『Portal Server リリースノート』 http://docs.sun.com/app/docs/doc/819-1494	製品のリリース後に発行されます。内容は、この現行リリースの新機能の説明、既知の問題点と制限、インストール上の注意事項、およびソフトウェアまたはドキュメントの問題点の報告方法など、リリース時に判明している情報です。
『Portal Server Technical Reference Guide』 http://docs.sun.com/db/doc/817-7696	ディスプレイプロファイルやリライタなどの Portal Server の技術的な概念、コマンド行ユーティリティー、ソフトウェアのタグライブラリ、およびテンプレートや JSP などのファイルに関する詳細な情報を提供します。このガイドが一冊あれば、このような基本的な背景情報を得ることができます。

その他の Portal Server のドキュメント

その他、次のような Portal Server のマニュアルがあります。

- 『Portal Server Desktop Customization Guide』
  http://docs.sun.com/doc/817-5318
- 『Portal Server Developer's Guide』
  http://docs.sun.com/doc/817-5319

- 『Portal Server Mobile Access Developer's Guide』
  http://docs.sun.com/doc/817-6258
- 『Portal Server Mobile Access Developer's Reference』
  http://docs.sun.com/doc/817-6259
- 『Portal Server Mobile Access 配備計画ガイド』
  http://docs.sun.com/app/docs/doc/817-7153
- 『Portal Server Mobile Access Tag Library Reference』
  http://docs.sun.com/doc/817-6260

その他のサーバーのドキュメント

その他のサーバーのドキュメントは次のとおりです。

- Directory Server のドキュメント
  http://docs.sun.com/db/prod/entsys?l=ja
- Web Server のドキュメント
  http://docs.sun.com/db/prod/entsys?l=ja
- Application Server のドキュメント
  http://docs.sun.com/db/prod/entsys?l=ja
- Web Proxy Server のドキュメント
  http://docs.sun.com/prod/s1.webproxys#hic

Sun のオンラインリソースへのアクセス

製品のダウンロード、専門的なサービス、パッチとサポート、および補足的な開発者情報が必要な場合は、次のサイトにアクセスしてください。

- Download Center
  http://wwws.sun.com/software/download/
- 専門的サービス
  http://www.sun.com/service/sunps/sunone/index.html
- Sun エンタープライズサービス、Solaris パッチ、およびサポート
  http://sunsolve.sun.com/
- 開発者向け情報
  http://developers.sun.com/prodtech/index.html

Sun テクニカルサポートへの問い合わせ

この製品に関して、製品のマニュアルには回答が記載されていない技術的な質問がある場合は、http://www.sun.com/service/contacting にアクセスしてください。

関連するサードパーティーの Web サイト

Sun は、このマニュアルに記載されているサードパーティー Web サイトの利用について責任を負いません。Sun は、このようなサイトまたはリソースで得られるあらゆる内容、広告、製品、およびその他素材を保証するものではなく、責任または義務を負いません。Sun は、このようなサイトまたはリソースで得られるあらゆるコンテンツ、製品、またはサービスによって生じる、または生じたと主張される、または使用に関連して生じる、または信頼することによって生じる、いかなる損害または損失についても責任または義務を負いません。

ご意見をお寄せください

Sun は、ドキュメントの質を向上させることに努力しており、読者の皆様からのご意見、ご提案を歓迎いたします。

ご意見をお寄せくださる場合は、http://docs.sun.com/app/docs?l=ja にアクセスして「コメントの送信」をクリックしてください。オンラインフォームには、ドキュメントのタイトルと Part No. を入力してください。Part No. は 7 桁または 9 桁の数字で、マニュアルのタイトルページかドキュメントの上部に記載されています。たとえば、本書のタイトルは『Sun Java System Portal Server 2005Q4 配備計画ガイド』で、Part No. は 819-4618 です。

ご意見をお寄せください

第 1 章

Portal Server のアーキテクチャー

この章で説明する内容は次のとおりです。

ポータルとは

ポータルにより、1 つのページから企業全体のさまざまなコンテンツ、データ、およびサービスにアクセスすることができます。ポータルプロバイダ、チャネル、およびポートレットによってポータルページに表示されるコンテンツは、ユーザー設定、組織内でのユーザーの役割や部門、サイトのデザイン、およびエンドユーザーとしての顧客に対するマーケティングキャンペーンに基づいて、パーソナライズすることができます。

ポータルは、いくつもの Web アプリケーションにアクセスする際の統一されたアクセスポイントとして機能します。また、セキュリティー、検索、コラボレーション、およびワークフローなどの便利な機能も備えています。ポータルは、統合されたコンテンツとアプリケーションを配信することに加え、統一された共同のワークスペースも

提供します。実際に、ポータルは次世代のデスクトップであり、Web 上の電子商取引アプリケーションをすべての種類のクライアントデバイスに配信することができます。優れたポータルソリューションがあれば、作業を完了するために必要なすべてのものに、いつでもどこでも、セキュリティー保護された方法でアクセスすることが可能です。

ポータルのタイプ

多くの新規ポータル製品が発表されているため、市場はますます混乱してきています。実際、ビジネスコンテンツへの Web インタフェースを備える製品やアプリケーションは、どれもポータルに分類されているのが現状です。このような理由から、ポータルにはさまざまな用途があり、次のいずれかに分類することができます。

- 共同ポータル
- ビジネスインテリジェンスポータル

共同ポータル

共同ポータルを利用することにより、ビジネスユーザーは、電子メール、ディスカッショングループで使用する資料、オフィスドキュメント、各種フォーム、メモ、議事録、Web ドキュメント、およびライブデータ送信のサポートなど、構造化されていないオフィスコンテンツを整理、検索、および共有することができます。共同ポータルは、広範な情報に対応できる点だけでなく、一連のコンテンツ管理サービスと共同サービスを提供する点でも、インターネットポータルやイントラネットポータルと異なっています。

次のようなコンテンツ管理サービスが提供されます。

- 以前には知られていなかった新しい情報の発見を意味するテキストマイニング
- 構造化されていない関連する情報の分類
- 情報のカテゴリ化
- ドキュメントの要約を生成するサマリー機能
- パブリッシングと登録
- 人名検索
- 高度な追跡機能

共同ポータルは、企業の機能として主に内部で使用されます。

共同サービスにより、ユーザーは次の操作を実行できます。

- チャット

- 会議の計画
- カレンダ情報の共有
- ユーザーコミュニティーの定義
- ネット会議への出席
- ディスカッショングループとホワイトボード内での情報の共有

ビジネスインテリジェンスポータル

ビジネスインテリジェンスポータルにより、経営幹部、部門マネージャー、およびビジネスアナリストは、ビジネスインテリジェンス機能にアクセスしてビジネス上の決定を下すことができます。一般にこのタイプのポータルでは、ビジネスインテリジェンスレポート、分析、および事前定義クエリーのインデックスを作成し、それらを財務管理、カスタマーリレーションシップ管理、およびサプライチェーンパフォーマンス管理と関連付けています。ビジネスインテリジェンスポータルからは、レポート作成、OLAP、データマイニングなどのビジネスインテリジェンスツール、パッケージ化された分析用アプリケーション、警告、パブリッシング、および登録機能にもアクセスできます。ビジネスインテリジェンスタイプのポータルを提供するベンダーとして代表的なのは、Peoplesoft です。

ビジネスインテリジェンスポータルには、次のようなタイプがあります。

- 調達ポータル
- セルフサービスポータル
- ビジネスポータル
- 電子商取引ポータル
- 販売サポート
- カスタマーリレーションシップ管理、業務、および社員用ポータル
- 消費者用ポータル

Portal Server の機能

Sun Java™ System Portal Server 6 2005Q4 ソフトウェアにより、組織では次の機能を使用することができます。

- セキュリティー保護されたアクセスと認証接続。オプションで、ユーザーのブラウザと企業の間で暗号化技術を使用できます。
- ユーザー認証。各ユーザーに固有のリソースセットへのアクセスを許可する前に適用します。
- 抽象化のサポート。さまざまなソースからコンテンツを引き出し、ユーザーのデバイスに適した出力形式でそれらの情報を集約およびパーソナライズできます。
- 検索エンジンインフラストラクチャー。イントラネットのコンテンツを整理し、ポータルからアクセスできます。
- ユーザーおよびサービスに特定の持続データのストア機能。
- 一般に必要なアプリケーションへのアクセス。メール、カレンダ、ファイルストレージなどのサービスにアクセスできます。
- 管理インタフェース。代理またはリモートの管理が可能です。
- シングルサインオン機能とセキュリティー機能。企業のアプリケーションとコンテンツに標準的な方法でアクセスできます。
- パーソナライズ。ポータルプロバイダ、ポートレット、および Web サービスリモートポートレットを使用します。
- コンテンツのパブリッシングと管理 (FatWire などのサードパーティーアプリケーションによる機能)。

Sun Java System Portal Server

Portal Server は、Sun Java™ Enterprise System テクノロジのコンポーネントです。Sun Java Enterprise System テクノロジでは、企業のコンピュータ環境におけるさまざまな必要性をサポートします。たとえば、セキュリティー保護されたイントラネットポータルを作成し、電子メールや社内ビジネスアプリケーションに企業の従業員が安全にアクセスできるようにします。

Portal Server 製品は、アイデンティティーを有効活用するポータルサーバーソリューションです。ユーザー、ポリシー、およびアイデンティティー管理のすべてを提供して、セキュリティー、Web アプリケーションのシングルサインオン (SSO)、およびエンドユーザーコミュニティーへのアクセス機能を実現します。また Portal Server は、パーソナライズ、集約、セキュリティー、統合、検索などのポータルサービスを結合させます。内部のリソースやアプリケーションへのセキュリティー保護されたリモー

トアクセスを可能にする独自の機能により、企業対社員、企業間、および企業対顧客の各ポータルを配備する包括的なポータルプラットフォームを提供します。Sun Java System Portal Server Secure Remote Access (SRA) により、リモートアクセス機能の安全性をさらに高めて、Web に対応しているリソースと対応していないリソースにアクセスできます。

それぞれの企業は個別の必要性を見積もり、Java Enterprise System テクノロジの独自の配備計画を作成します。各企業に合った最適な配備方法は、Java Enterprie System テクノロジによってサポートするアプリケーションのタイプ、ユーザー数、使用可能なハードウェアの種類、およびこの種の他の考慮点によって異なります。

Portal Server は、すでにインストール済みのソフトウェアコンポーネントと連動することができます。この場合 Portal Server は、ソフトウェアのバージョンが適切であれば、インストール済みのソフトウェアを使用します。

Secure Remote Access

Sun Java System Portal Server Secure Remote Access (SRA) により、Java テクノロジが有効なすべてのリモートブラウザから、ポータルのコンテンツおよびサービスにセキュリティー保護された状態でアクセスできます。

SRA には、Java テクノロジが有効なすべてのブラウザからアクセスできるので、クライアントソフトウェアが不要です。Portal Server ソフトウェアと統合すると、アクセス権のあるコンテンツおよびサービスに対して暗号化された安全なアクセスが保証されます。

SRA は、安全性の高いリモートアクセスポータルを提供する企業を対象に設計されています。このようなポータルは、イントラネットリソースのセキュリティー、保護、およびプライバシーに重点が置かれています。Access List、Gateway、NetFile、Netlet、およびプロキシレットなどの SRA サービスにより、インターネット上にリソースを公開することなく、インターネットを介してイントラネットのリソースにセキュリティー保護してアクセスできます。

Portal Server は、SRA を使用する場合はセキュアモード、しない場合はオープンモードで動作します。

オープンモードの Portal Server

オープンモードの場合、Portal Server は SRA なしでインストールされます。標準的な公開ポータルは、HTTP プロトコルのみを使用し、セキュリティー保護されたアクセス機能なしで運営されています。オープンモードでも、インストール時またはインストール後に HTTPS プロトコルを使用するように Portal Server を設定できますが、セキュリティー保護されたリモートアクセスはできません。つまり、リモートファイルシステムとアプリケーションにはアクセスできません。

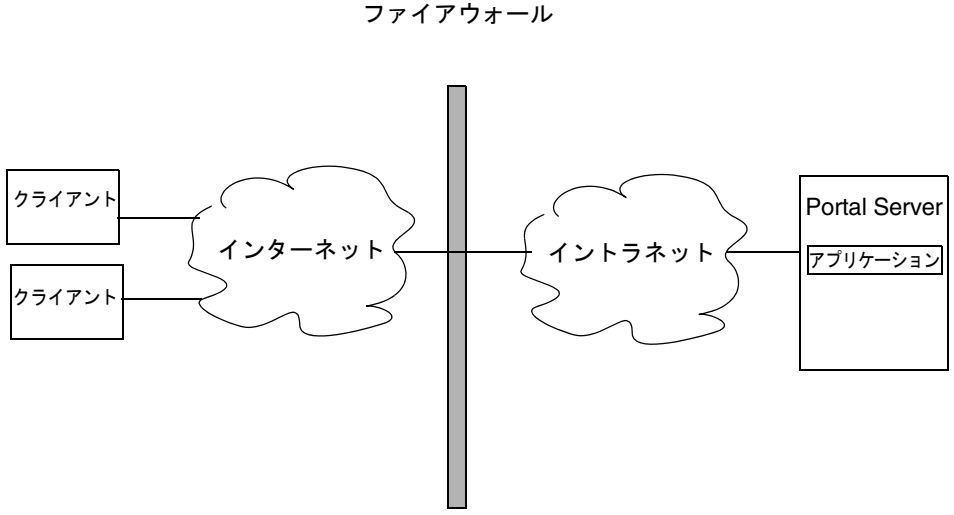
オープンポータルとセキュアポータルの主な違いは、オープンポータルを通じて提供されるサービスは、通常は保護されたイントラネット内ではなく非武装ゾーン (DMZ) 内に存在する点にあります。

ポータルに機密情報が含まれていない場合 (公開情報を配置し、無償アプリケーションへのアクセスを許可)、大量のアクセスに対する応答は、セキュアモードに比べて速くなります。

図 1-1 は、オープンモードに設定された Portal Server を示しています。この図では、ファイアウォールの背後にある単独サーバーに Portal Server がインストールされています。複数のクライアントが、インターネット全体から 1 箇所のファイアウォールを通して、またはファイアウォールの背後に設置された Web プロキシサーバーから Portal Server にアクセスします。

注 HTTPS プロトコルを有効にして Portal Server をオープンモードで動作させることにより、Web 対応のリソースのユーザーにセキュリティー保護されたアクセス機能を提供できます。ただし、SRA がなければ、ファイルシステムや TCP/IP アプリケーションへのセキュリティー保護されたリモートアクセス機能は提供できません。

図 1-1　オープンモードの Portal Server



セキュアモードの Portal Server

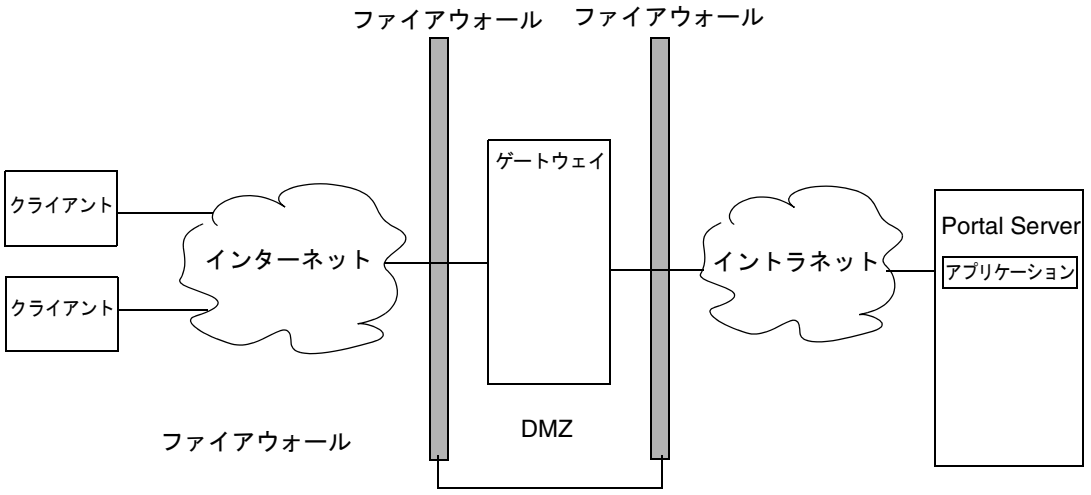
セキュアモードの場合、Portal Server は SRA とともにインストールされます。セキュアモードは、必要とされるイントラネットファイルシステムとアプリケーションへのセキュリティー保護されたリモートアクセスを可能にします。

SRA による主な利点は、ゲートウェイの IP アドレスのみがインターネットに公開されることです。その他すべてのサービスおよびその IP アドレスは隠され、インターネットなどの公衆ネットワークで稼働するドメインネームサービス (DNS) には一切公開されません。

ゲートウェイは非武装ゾーン (DMZ) に常駐します。ゲートウェイは、すべてのイントラネット URL とアプリケーションへの単一のセキュアアクセスポイントとして働くため、ファイアウォールで開かれるポートの数は減ります。セッション、認証、およびポータルデスクトップなど、他のすべての Sun Java System サービスは、保護されたイントラネットの DMZ の背後に常駐します。クライアントブラウザからゲートウェイへの通信は、SSL (Secure Socket Layer) を使った HTTP を使って暗号化されます。ゲートウェイからサーバーおよびイントラネットリソースへの通信には HTTP か HTTPS が使用されます。

図 1-2 は、SRA とともにインストールされた Portal Server を示しています。SSL はクライアントとゲートウェイの接続をインターネット上で暗号化するために使用されます。また SSL は、ゲートウェイと Portal Server システム間の接続の暗号化にも使用されます。イントラネットとインターネットの間にゲートウェイが存在するので、クライアントと Portal Server システム間のセキュアパスが延長されます。

図 1-2 セキュアモードの Portal Server



サーバーとゲートウェイを追加してサイトを拡張することもできます。また、ビジネス要件に基づいて、SRA のコンポーネントをさまざまな方法で設定することも可能です。

セキュリティー、暗号化、および認証

Portal Server システムのセキュリティー機能では、UNIX システムのセキュリティー機能に加え、HTTPS 暗号化プロトコルに依存して、Portal Server システムソフトウェアを保護しています。

セキュリティーは Web コンテナによって実現され、必要に応じて SSL を使用するように設定できます。Portal Server は、認証とエンドユーザー登録の場合の SSL もサポートしています。Web サーバーで SSL 証明書を有効にすることにより、ポータルデスクトップや他のアプリケーションにもセキュリティー保護してアクセスできます。Access Manager ポリシーを使用して、URL ベースのアクセスポリシーも設定できます。

Portal Server は、Sun Java System Access Manager によって提供される認証サービスを利用して、Access Manager SSO メカニズムを使用するすべての製品間でのシングルサインオン (SSO) をサポートします。SSO メカニズムでは、エンコードされたクッキーを使用してセッション状態を保持します。

SRA には、さらに別のセキュリティー機能があります。SRA では、デフォルトで HTTPS を使用して、クライアントのブラウザをイントラネットに接続します。ゲートウェイでは、リライタを使用して、インターネットに直接コンテンツを公開せずにイントラネットの Web サイトにアクセスする仕組みを実現しています。またゲートウェイにより、アクセスされる Web サーバーに変更を加えずに、URL ベースのアクセスポリシーも設定できます。

ゲートウェイからサーバーおよびイントラネットリソースへの通信には、HTTP または HTTPS を使用できます。Web アプリケーションとディレクトリサーバーとの間の通信のように、Portal Server 内での通信では、デフォルトで暗号化を使用しませんが、SSL を使用するように設定できます。

Portal Server の配備コンポーネント

Portal Server の配備は、次のコンポーネントで構成されます。

- Access Manager

  Access Manager により、ユーザーとサービスの管理、認証サービスとシングルサインオンサービス、ポリシー管理、ロギングサービス、デバッグユーティリティー、管理コンソール、および Portal Server のクライアントサポートインタフェースを使用できます。次のコンポーネントで構成されています。

  - Java Development Kit™ (JDK™)。Java Development Kit ソフトウェアは、Portal Server のすべての Java ソフトウェアおよび Portal Server を構成するコンポーネントに、Java ランタイム環境を提供します。Portal Server は、Web コンテナの JDK ソフトウェアに依存して動作します。
  - Java ソフトウェアのネットワークセキュリティーサービス
  - Sun Java System Web Server
  - Java API for XML Processing (JAXP)

- Sun Java System Directory Server

  Directory Server は、Portal Server の主要な設定およびユーザープロファイルデータリポジトリです。Directory Server は LDAP に準拠し、拡張可能なオープンスキーマを実装しています。

- Web コンテナ
  - Sun Java System Web Server
  - Sun Java System Application Server Enterprise Edition

  We b Server および Application Server ソフトウェアの代わりに、以下の Web コンテナを使用できます。
  - BEA WebLogic Server™
  - IBM WebSphere® Application Server

  さまざまな Web コンテナへの Portal Server の配備については、『Sun Java System インストールガイド』を参照してください。

注 Portal Server でサポートされる製品の特定のバージョンについては、『Portal Server 6 リリースノート』を参照してください。

Portal Server のアーキテクチャー

必ずではありませんが、通常は、以下のさまざまなポータルノード (サーバー) Portal Server ソフトウェアを配備して、連携動作することによってポータルを実装します。

- **Portal Server ノード** : Portal Server が常駐する Web サーバーです。必要であれば、このノードに検索コンポーネントをインストールすることもできます。Access Manager もここに常駐可能です。
- **Access Manager ノード** : Access Manager が常駐可能なサーバーです。Access Manager は、Portal Server と同じノードに常駐する必要はありません。
- **検索ノード** : オプションです。Portal Server の検索サービスで使用するサーバーです。Portal Server 検索サービスは、パフォーマンス、スケーラビリティー、および可用性を高めるために、独自のサーバーにインストールできます。
- **ゲートウェイノード** : オプションです。SRA ゲートウェイが常駐するサーバーです。ゲートウェイはポータルノードにインストールできます。ゲートウェイは DMZ に配置するので、分離されたポータル以外のノードにインストールされます。
- **Netlet プロキシノード** : オプションです。ユーザーのイントラネットでアプリケーションを実行しているリモートデスクトップとサーバーの間で、アプリケーションをセキュリティー保護して実行するために使用されるサーバーです。
- **リライタプロキシノード** : オプションです。ユーザーのイントラネットでアプリケーションを実行しているリモートデスクトップとサーバーの間で、アプリケーションをセキュリティー保護して実行するために使用されるサーバーです。
- **Directory Server ノード** : Directory Server ソフトウェアを実行しているサーバーです。Directory Server はポータル以外のノードにインストールできます。
- **その他のサーバー** : メールサーバーやファイルサーバーのようなサーバーおよび旧バージョンのサーバーは、バックエンドサポート、データ、およびアプリケーションをポータルユーザーに提供します。

アイデンティティー管理

Portal Server は、コンテンツ、アプリケーション、およびサービスにアクセスする際には、Access Manager を利用して、組織内時には組織外に及ぶさまざまな役割を持つ数多くのユーザーを制御します。課題としては、だれがアプリケーションを使用するのか、ユーザーはどんな能力範囲で組織または企業に労働力を提供するのか、ユーザーの使命は何か、ユーザーは何にアクセスする権限をもつべきか、他の人は管理作業をどのように支援できるか、などの点が挙げられます。

Access Manager ソフトウェアは、次のコンポーネントで構成されます。

- SSO トークン、ユーザープロファイル、ロギング、およびデバッグにアクセスするために使用する Java ソフトウェア API
- amadmin、amserver、ampassword などのコマンド行ツール
- セッション、認証、ログイン、ネーミングなどの Web アプリケーションサービス
- 管理コンソール Web アプリケーション
- Access Manager SDK
- Access Manager コンソール SDK
- Web アプリケーションをサポートする認証デーモン

詳細については、『Access Manager 配備計画ガイド』を参照してください。

Portal Server ソフトウェアの配備

このセクションでは、Portal Server に配備されるソフトウェアについて説明します。内容は、ソフトウェアのパッケージ化メカニズム、システム内部のソフトウェアカテゴリ、および Java ソフトウェアとの互換性です。

ソフトウェアのパッケージ化

Portal Server では、「動的 WAR ファイル」式のアプローチを使用して、ソフトウェアをシステムに配備します。Portal Server は Solaris™ パッケージを使用してインストールされます。Solaris™ パッケージは、JAR、JSP、テンプレート、および HTML ファイルなど、Web アプリケーションを構成する個々のファイルで構成されています。パッケージには、WAR ファイルや EAR ファイルは含まれていませんが、インストール時に Portal Server WAR ファイルを構成するために使用する、`web.xml` フラグメン

トが含まれています。この動的に構成されるファイルが、Web アプリケーションコンテナに配備されます。ローカリゼーションなどの場合に追加パッケージがシステムにインストールされると、Web アプリケーションファイルは再構成および再配置されます。

注 WAR ファイルのパッケージ化と配備の仕組みは、Portal Server 製品だけが使用します。現行では、WAR ファイルや WAR ファイルを構成するために使用されるファイルにユーザーが変更を加えることはできません。

ソフトウェアのカテゴリ

Portal Server は、Portal Server ノードにインストールするソフトウェアの種類を、次のように区別します。

- **動的 Web アプリケーション**: これには、Java プラットフォームで動作するサーブレット、JSP ファイル、コンテンツプロバイダ、およびユーザーのブラウザからアクセスされたときに Web コンテナが処理するその他の項目が含まれます。Portal Server の場合は、これらのファイルが Web Server にインストールされます。
- **静的 Web コンテンツ**: これには、静的 HTML ファイル、画像、アプレット JAR ファイル、および Web Server コンテナを使用せずに Web Server によって直接サービスを提供可能なその他の項目が含まれます。Portal Server の場合は、これらのファイルも Web Server にインストールされます。

注 静的 Web コンテンツと動的 Web アプリケーションは、すべて 1 つの WAR ファイルにグループ化されます。

- **設定データ**: これには、ディレクトリにインストールされるデータが含まれます。つまり、Access Manager サービスの定義、およびインストール時にディレクトリに変更を加えるそれ以外のデータです。これには、Portal Server 拡張機能で接続する、コンソール設定データへの変更などがあります。設定データは、Portal Server の数に関係なく、1 回だけインストールされます。
- SDK: JAR ファイルまたはコンポーネントによって使用可能になる Java API を含むファイルです。開発者は、このパッケージを開発システムにインストールして、API を使用するクラスをコンパイルできるようにする必要があります。コンポーネントが公開 Java API をエクスポートしない場合は、このパッケージは含まれていません。

標準的な Portal Server のインストール

35 ページの図 1-3 は、ポータル配備のコンポーネントをいくつか図示していますが、実際の物理ネットワーク設計、シングルポイント障害、または高可用性については説明していません。ポータル設計の詳細については、第 5 章「ポータルの設計」を参照してください。

この図は、企業サイトにインストールされた標準的な企業対社員用ポータルの高レベルアーキテクチャーを示しています。この図では、プロキシ / キャッシュサーバー、Web サーバー、メールゲートウェイなどのインターネットからアクセス可能な他のシステムと一緒に、ゲートウェイが企業の DMZ に配置されています。ポータルノード、ポータル検索ノード、およびディレクトリサーバーは、個々の社員のデスクトップシステムから旧バージョンのシステムにいたるまで、ユーザーがアクセス可能なシステムやサービスが存在する内部ネットワークに配置されています。

注 ビジネスの顧客ごとに別々の Portal Server インスタンスをホストする ISP ホスティングの配備を設計している場合は、Sun Java System の担当者にご連絡ください。Portal Server は、ISP ホスティング機能を提供するためにカスタマイズする必要があります。

35 ページの図 1-3 では、インターネットのユーザーがブラウザからゲートウェイにアクセスします。ゲートウェイは、ユーザーがアクセスしようとしているポータルの IP アドレスとポートにユーザーを接続します。たとえば、B2B ポータルは通常、HTTPS ポートである 443 番ポートにのみアクセスを許可します。ゲートウェイは、認証された使用法に応じて、要求をポータルノードに転送するか、企業の内部ネットワークのサービスに直接転送します。

図 1-3　企業対社員用ポータルの高レベルアーキテクチャー

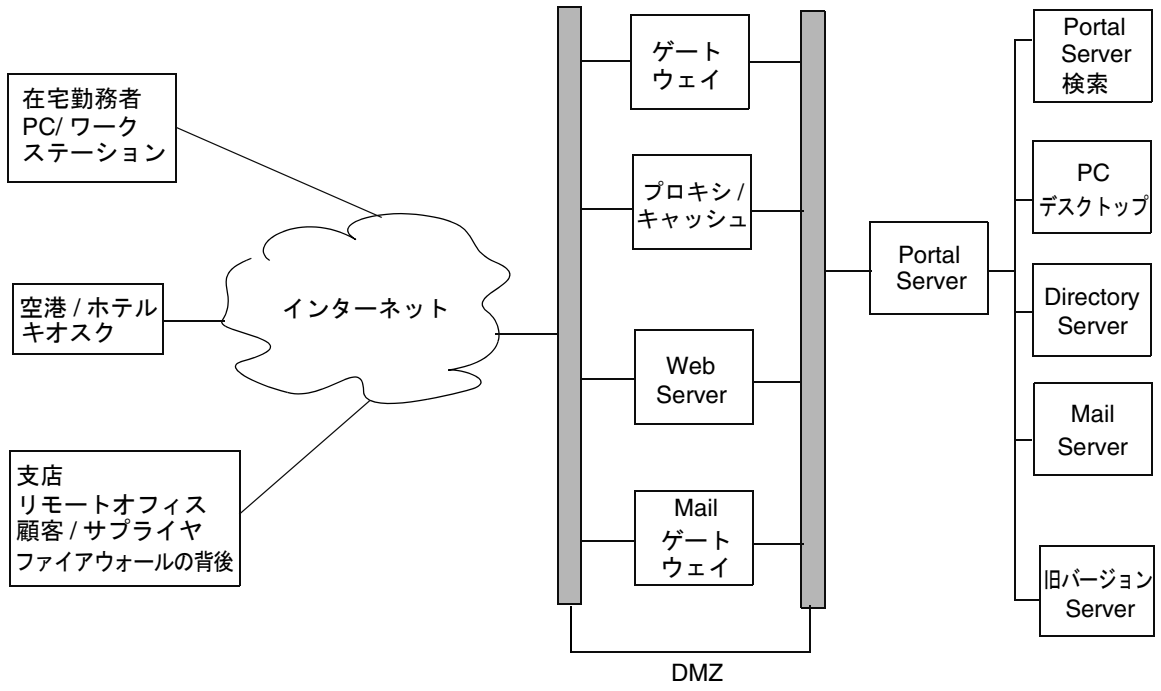
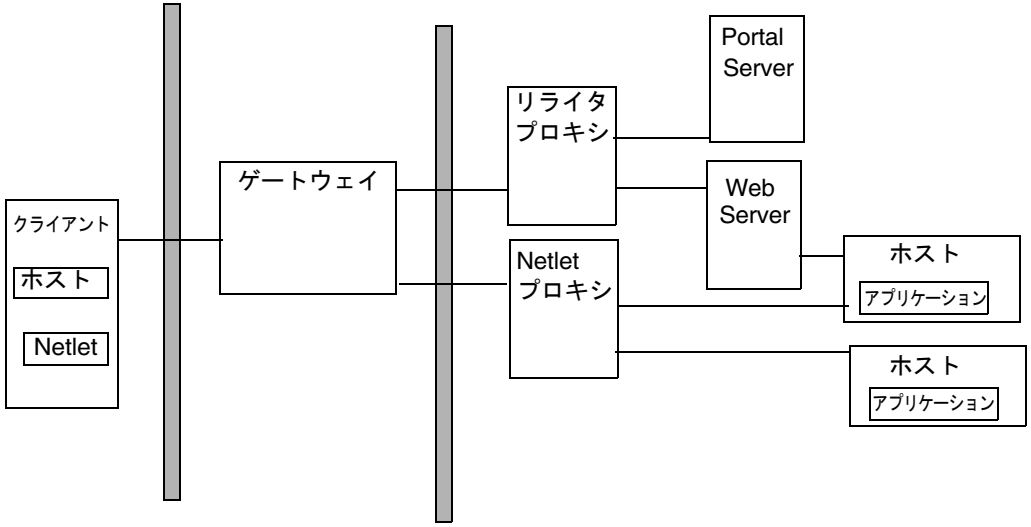


図 1-4 は、SRA サービスを使用した Portal Server の配備を示しています。詳細については、第 2 章「Portal Server Secure Remote Access アーキテクチャー」を参照してください。

図 1-4　SRA の配備


Portal
Server
リライタ
プロキシ
ゲートウェイ
Web
Server
クライアント
ホスト
Netlet
Netlet
プロキシ
ホスト
アプリケーション
ホスト
アプリケーション

第 2 章

Portal Server Secure Remote Access アーキテクチャー

この章では、Sun Java™ System Portal Server Secure Remote Access (SRA) アーキテクチャーについて説明します。

設定情報は、Access Manager 管理コンソールを使用して管理します。

この章では、次の SRA コンポーネントについて説明します。

- SRA ゲートウェイ
- Netlet
- Netlet プロキシ
- NetFile
- リライタ
- リライタプロキシ
- プロキシレット

SRA ゲートウェイ

SRA ゲートウェイはスタンドアロン Java プロセスです。状態情報をエンドユーザーにとって透過的に再構築することができるので、ステートレスと考えることができます。SRA ゲートウェイは、設定されたポートで待機し、HTTP 要求と HTTPS 要求を受け入れます。要求を受け取ると、セッションの有効性とヘッダー情報を確認して、要求のタイプを判別します。SRA ゲートウェイは要求のタイプに応じて次の処理を実行します。

- **Netlet 要求 :** ユーザーがポータルデスクトップでクリックした Netlet ルールで指定されるサーバーに、要求 (トラフィック) を経路指定します。

- **HTTP トラフィック** : HTTP ヘッダーによって指定されたサーバーに要求を経路指定します。サーバーから応答を受け取ると、ゲートウェイは応答を変換し、応答内のすべてのイントラネットリンクがエクストラネットで有効になるようにします。

すべてのゲートウェイ設定情報は、Access Manager の LDAP データベースにプロファイルとして保管されます。ゲートウェイプロファイルは、ゲートウェイに関連するすべての設定情報を含んでいます。

ホスト名や IP アドレスなどのマシン固有の情報はすべて、ゲートウェイがインストールされているローカルファイルシステムの設定ファイルに保管されます。これにより、複数のマシンで動作するゲートウェイどうしの間で 1 つのゲートウェイプロファイルを共有できます。

上述のとおり、SRA ゲートウェイは、HTTP モードと HTTPS モードの両方で同時に稼働するように設定できます。これにより、エクストラネットのユーザーは HTTPS を使用し、イントラネットのユーザーは SSL オーバーヘッドのない HTTP を使用して、イントラネットとエクストラネットの両方のユーザーが同じゲートウェイにアクセスできます。

SRA ゲートウェイは、`chroot` 環境でも実行できます。詳細は、『Portal Server Secure Remote Access 6 管理ガイド』を参照してください。

複数のゲートウェイインスタンス

必要に応じて、1 台のマシンで複数のゲートウェイインスタンスを実行できます。これは、マルチホームゲートウェイと呼ばれます。それぞれのゲートウェイインスタンスは、別々のポートで待機します。ゲートウェイインスタンスを設定して、同じ Portal Server インスタンスまたは異なる Portal Server インスタンスと通信できます。同じマシンのゲートウェイで複数インスタンスを実行する場合は、個別の証明書データベースをゲートウェイの各インスタンスに関連付け、そのゲートウェイをドメインにバインドすることができます。基本的には、これにより各ドメインで異なるゲートウェイサーバーの証明書を使用でき、柔軟性が高まります。

複数の Portal Server インスタンス

Portal Server の複数インスタンスによってゲートウェイを設定すると、ゲートウェイは、ユーザーを異なるサーバーに交互にログインさせて、自動的にラウンドロビン方式のロードバランスを実行します。またゲートウェイは、稼働中のサーバーのリストを維持して、停止中のサーバーにユーザーをログインさせることのないようにします。この仕組みにより、Portal Server のシングルポイント障害を防ぎます。

注 ゲートウェイの前では、Netlet を使用しているのでない限りセッション固定は不要ですが、セッション固定によりパフォーマンスが向上します。一方、Portal Server インスタンスへのセッション固定は SRA によって維持されます。

プロキシの設定

ゲートウェイは、ゲートウェイのプロファイルで指定されたプロキシを使用して、イントラネットとエクストラネット内部のさまざまな Web サーバーからコンテンツを取得します。プロキシは、ホストおよび DNS のサブドメインとドメインの専用にすることができます。ゲートウェイは、プロキシ設定に応じて適切なプロキシを使用し、要求されたコンテンツを取得します。プロキシで認証が要求される場合は、プロキシ名がゲートウェイプロファイルの一部として保管され、ゲートウェイはプロキシに接続する際にその名前を自動的に使用します。

ゲートウェイと HTTP 基本認証

ゲートウェイは基本認証をサポートするので、ユーザー ID とパスワードの入力を求めるプロンプトを表示しますが、ユーザーのコンピュータからサイトの Web サーバーまでの送信時にそれらのクレデンシャルを保護しません。送信するクレデンシャルを保護するには、セキュリティー保護された HTTP 接続を確立する必要があり、通常は SSL を使用します。

Web サーバーが基本認証を要求する場合、クライアントはユーザー名とパスワードの入力を求めてプロンプトを表示し、要求しているサーバーに情報を送信します。HTTP 基本認証が有効なゲートウェイは、ユーザー名とパスワードの情報を捕捉し、その後に行われる認証とログインに備えて、それらの情報のコピーを Access Manager のユーザープロファイルに保存します。元のデータはゲートウェイから宛先 Web サーバーに送信され、基本認証が実行されます。Web サーバーはユーザー名とパスワードを検証します。

ゲートウェイにより、個々のホストでこの機能を拒否および許可する設定を細かく制御することもできます。

ゲートウェイと SSL サポート

SRA ゲートウェイは、HTTPS モードで動作するとき、SSL v2 と SSL v3 の両方の暗号化をサポートします。Access Manager 管理コンソールを使用して、特定の暗号化機能を有効または無効にできます。ゲートウェイは Transport Layer Security (TLS) もサポートします。

SSL v3 には、2 つの認証モードがあります。

- **必須サーバー認証** : クライアントは、サーバーを認証する必要があります。
- **オプション認証** : サーバーはクライアント認証を行うように設定されています。

Personal Digital Certificate (PDC) 認証は、SSL クライアント認証によってユーザーを認証するメカニズムです。ゲートウェイは、Access Manager 認証モジュールをサポートすることによって PDC 認証をサポートします。SSL クライアント認証を使用して、SSL ハンドシェークがゲートウェイで終了します。この PDC ベースの認証は、Access Manager の証明書ベースの認証とともに統合されます。したがってクライアント認証は、ゲートウェイによってではなく Access Manager によって処理されます。

セッション情報が HTTP または HTTPS 要求の一部として検出されない場合、ゲートウェイは、Access Manager からログイン URL を取得して、ユーザーを直接認証ページに誘導します。同様に、セッションが要求の一部として無効であることを検出すると、ゲートウェイはユーザーをログイン URL に誘導し、ログインが正常に行われるとユーザーを要求された宛先に誘導します。

SSL セッションが確立されると、ゲートウェイは送信されてくる要求を受信し続け、セッションの有効性を確認し、宛先の Web サーバーに要求を転送します。

ゲートウェイサーバーはすべての Netlet トラフィックを処理します。送信されてくるクライアント要求が Netlet トラフィックの場合、ゲートウェイはセッションの有効性を確認し、トラフィックを暗号化して、アプリケーションサーバーに転送します。Netlet プロキシが有効な場合、ゲートウェイはセッションの有効性を確認して、Netlet プロキシに転送します。Netlet プロキシはトラフィックを復号化してアプリケーションサーバーに転送します。

注 40 ビットの暗号化は安全性が低いので、ゲートウェイには、40 ビット暗号化ブラウザからの接続を拒否するためのオプションが用意されています。

ゲートウェイのアクセス制御

ゲートウェイは、許可される URL リストと拒否される URL リストを使用してアクセス制御を実施します。URL のアクセスが許可されている場合でも、ゲートウェイは、Access Manager セッションサーバーと照会してセッションの有効性を確認します。許可される URL リストと拒否される URL リストと同様、非認証 URL リストにある URL はセッション検証を省略します。拒否される URL リストのエントリは、許可される URL リストのエントリよりも優先されます。特定の URL がいずれかのリストに記載されていない場合、その URL に対するアクセスは拒否されます。許可される URL リストと拒否される URL リストのどちらの場合でも、URL の一部としてワイルドカード文字 * も使用できます。

ゲートウェイのロギング

ゲートウェイでのロギングを有効にすることにより、ユーザーの動作をすべて監視できます。SRA ゲートウェイは Access Manager ロギング API を使用してログを作成します。

ゲートウェイでのアクセラレータの使用

専用ハードウェアコプロセッサであるアクセラレータを設定して、サーバーの CPU から SSL 機能の負荷を取り除くことができます。アクセラレータを使用すると、CPU を他のタスクに振り分けられるようになり、SSL トランザクションの処理速度が向上します。

Netlet

Netlet により、イントラネットで使用可能な固定ポートアプリケーションと一部の動的ポートアプリケーションに対するイントラネットの外部からのアクセスをセキュリティー保護できます。クライアントは、リモートファイアウォールと SSL プロキシの背後に配置することも、直接インターネットに接続することもできます。Netlet を介して、イントラネットの外部からイントラネットアプリケーションにセキュリティー保護して確立される接続は、Netlet ルールに従って制御されます。

ブラウザで動作する Netlet アプレットにより、リモートクライアントマシンとリモートホストのイントラネットアプリケーションの間で、暗号化された TCP/IP トンネルが設定されます。Netlet は事前設定されたポートで待機して接続を受け入れ、クライアントと宛先サーバーの間で送受信されるトラフィックを経路指定します。送受信さ

れる両方のトラフィックは、ユーザーが選択した、または管理者によって設定された暗号化アルゴリズムによって暗号化されます。Netlet ルールには、接続で使用されるすべてのサーバー、ポート、および暗号化アルゴリズムに関する詳細が記述されています。管理者は、Access Manager 管理コンソールを使用して Netlet ルールを作成します。

静的および動的なポートアプリケーション

静的ポートアプリケーションは、既知つまり静的ポートで動作します。たとえば、IMAP や POP サーバー、Telnet デーモン、jCIFS などがあります。静的ポートアプリケーションの場合は、Netlet ルールに宛先サーバーポートが記述されているので、要求を直接宛先に経路指定できます。

動的アプリケーションは、ハンドシェークの一部で通信用ポートについて同意します。宛先サーバーポートを Netlet ルールの一部にすることもできます。Netlet は、プロトコルを理解し、データを調べて、クライアントとサーバーの間で使用されるポートを検出する必要があります。FTP は動的ポートアプリケーションです。FTP の場合、クライアントとサーバー間の実際のデータ転送用ポートは、`PORT` コマンドによって指定されます。この場合、Netlet はトラフィックをパースして、動的にデータチャネルポートを取得します。

現行では、FTP と Microsoft Exchange だけが、Portal Server によってサポートされる動的ポートアプリケーションです。

注 Microsoft Exchange 2000 は Netlet によってサポートされていますが、以下の制約があります。

- Exchange は静的ポートを使用するように設定する必要があります。
- Netlet は Microsoft Windows 2000 および XP では動作しません。これは、Microsoft Windows 2000 と XP のクライアントが、RPC Portmapper の Exchange 用ポート (ポート 135) を Active Directory 用に予約しているからです。それ以前のバージョンの Microsoft Windows は、このポートを予約していません。ポートが予約されているので、そのポートに Netlet を割り当てることができず、そのためポートで必要なトンネリング機能を提供できません。
- Outlook 2000 クライアントの場合は、Exchange サーバーに接続するときに使用するポートを変更できません。

Netlet とアプリケーション統合

Netlet は、Graphon、Citrix、pcAnywhere などの多くのサードパーティー製品で動作します。それらの各製品は、リモートマシンからユーザーのポータルデスクトップへのアクセスを Netlet を使用してセキュリティー保護します。

スプリットトンネリング

スプリットトンネリングにより、VPN クライアントは、VPN に接続または VPN から切断することなく、セキュリティー保護されたサイトおよびセキュリティー保護されていないサイトの両方に接続できます。この場合の VPN は Netlet です。クライアントは、暗号化パスを通して情報を送信するか、または非暗号化パスを使用して送信するかどうかを判別します。スプリットトンネリングの問題点は、セキュリティー保護されていないインターネットから、クライアントを介して VPN によるセキュリティー保護ネットワークに直接接続が可能であることです。スプリットトンネリングをオフにすると両方の接続が同時に許可されることがなくなり、インターネット侵入に対する VPN 接続 (この場合は Netlet 接続) の脆弱性が低減します。

Portal Server は、ポータルサイトに接続されている間、複数のネットワーク接続を禁止したりシャットダウンしたりしませんが、権限のないユーザーが他のユーザーのセッションに便乗 (piggybacking) する行為を次の方法で阻止します。

- Netlet は、アプリケーション特有の VPN であり、汎用 IP ルーターではありません。Netlet は、Netlet ルールによって定義されたパケットを転送するだけです。この仕組みは、一度ネットワークに接続すれば LAN 全体へのアクセス権が与えられる標準的な VPN とは異なります。
- Netlet を実行できるのは、認証済みのポータルユーザーだけです。ポータルアプリケーションはユーザーが正常に認証されるまで実行されることはなく、認証されたセッションがなければ新規接続は確立されません。
- アプリケーション側の所定のアクセス制御すべては有効に機能し続けるので、攻撃者はバックエンドアプリケーションにも侵入しなければならなくなります。
- Netlet 接続が確立されると、認証されたユーザーの JVM™ で動作する Netlet によって毎回ダイアログボックスによる通知が出され、認証されたユーザーの画面に表示されます。ダイアログボックスでは、検証と確認を行なって新規接続を許可するかどうか尋ねられます。攻撃者が Netlet 接続を利用するためには、Netlet が実行していたこと、Netlet が待機していたポート番号、およびバックエンドアプリケーションへの侵入方法を知っており、ユーザーに安心して接続を認めさせることが必要になります。

Netlet プロキシ

Netlet プロキシは、ゲートウェイと宛先ホストを接続するために、ファイアウォールで開く必要のあるポート数を少なくするのに役立ちます。

たとえば、イントラネット内部で多数の Telnet、FTP、および Microsoft Exchange サーバーを接続するために、Netlet を必要とする設定について考えてみます。ゲートウェイは DMZ 内にあると仮定します。ゲートウェイがトラフィックをすべての宛先サーバー向けて経路指定する場合は、第 2 ファイアウォールでかなりの数のポートを開いておく必要があります。この問題を軽減するため、第 2 ファイアウォールの背後に Netlet プロキシを配置し、Netlet プロキシにトラフィックを転送するようにゲートウェイを設定できます。その後、Netlet プロキシがすべてのトラフィックをイントラネット内のすべての宛先サーバーに経路指定するので、第 2 ファイアウォールで開く必要のあるポート数を減らすことができます。また、第 2 ファイアウォールの背後に複数の Netlet プロキシを配置して、シングルポイント障害を回避することもできます。

サードパーティーのプロキシを使用して第 2 ファイアウォールのポートを 1 つだけ使用することも可能です。

注 Netlet プロキシを別のノードにインストールすると、Netlet トラフィックの負荷が別のノードに分散されるので、Portal Server の応答時間を短くすることができます。

NetFile

NetFile により、企業のイントラネット内部にセキュリティー保護されて常駐するファイルシステムに、リモートアクセスして操作することができます。

NetFile は、NFS、jCIFS、および FTP などの標準プロトコルを使用して、ユーザーがアクセス権を持つすべての UNIX® または Microsoft Windows ファイルシステムに接続します。NetFile を使用して、ファイル管理アプリケーションで一般的なほとんどのファイル操作を実行できます。詳細は、『Portal Server Secure Remote Access 6 管理ガイド』を参照してください。

コンポーネント

さまざまなファイルシステムにアクセスするため、NetFile には 3 つのコンポーネントが組み込まれています。

- NetFile Java 1 アプレット : AWT ベースのユーザーインタフェースを搭載しています。Java 2 をサポートできない旧バージョンのブラウザの場合に使用します。
- NetFile Java 2 アプレット : Swing ベースのユーザーインタフェースを搭載しています。Java プラグインをサポートするブラウザの場合に使用します。
- NetFile サーブレット : Web コンテナには、NetFile アプレットの種類ごとに 1 つずつ、合計 2 つの NetFile サーブレットが入っています。サーブレットは、異なるタイプのファイルシステムに接続し、NetFile に設定された操作を実行し、表示用の情報をアプレットに返信する機能を実行します。

NetFile は国際化されているので、ロケール (文字エンコード) に関係なくファイルシステムにアクセスできます。

NetFile は、Access Manager を使用して、NetFile 自体のプロファイル、ユーザー設定、および優先設定を保存します。NetFile は、Access Manager 管理コンソールを使用して管理します。

初期化

Portal Server デスクトップでユーザーが NetFile リンクを選択すると、NetFile サーブレットは、ユーザーが有効な SSO トークンを持ち、NetFile を実行する権限を持っているかどうかを確認します。有効なトークンと権限を確認できると、アプレットがブラウザにレンダリングされます。NetFile アプレットは、もう一度サーブレットに接続して、サイズ、ロケール、リソースバンドル、およびユーザー設定と優先設定など、それ自体の設定情報を取得します。NetFile は、ユーザーの SSO トークンを使用して、ロケール情報と、ユーザー名、メール ID、およびメールサーバーなどの他のユーザー情報を取得します。ユーザー設定には、ユーザーが組織またはロールから継承した設定、ユーザーによってカスタマイズされた設定、および前の NetFile セッションを終了するときにユーザーが保存した設定などがあります。

クレデンシャルの検証

NetFile は、ユーザーによって入力されるクレデンシャルを使用して、ファイルシステムへのアクセスを許可する前にユーザーを認証します。

クレデンシャルには、ユーザー名、パスワード、および Microsoft Windows または Novell のどちらか該当するドメインなどがあります。それぞれの共有では独立パスワードを使用するので、ユーザーは共通ホストを除き、すべての追加する共有ごとに自分のクレデンシャルを入力する必要があります。

NetFile は、Access Manager から UNIX 認証を使用して、NFS ファイルシステムへのアクセス権を与えます。FTP プロトコルと jCIFS プロトコルに従ってアクセスするファイルシステムの場合は、プロトコル自体が備える手法を使用してクレデンシャルを検証します。

アクセス制御

NetFile は、さまざまな方法でファイルシステムのアクセス制御を行います。プロトコルに基づいて、特定のファイルシステムへのユーザーのアクセスを拒否できます。たとえば、NFS によってのみアクセス可能なファイルシステムへの特定のユーザー、ロール、または組織のアクセスを拒否できます。

NetFile を設定し、組織からサブ組織、およびユーザーにいたるまでのすべてのレベルで、ファイルシステムへのアクセスを許可または拒否できます。また、特定のサーバーへのアクセスを許可または拒否することも可能です。ユーザーのファイルシステムへのアクセスは、Microsoft Windows、FTP、NFS、および NetWare を介した FTP などのホストのタイプに応じて許可または拒否できます。たとえば、組織のすべてのユーザーに対し Microsoft Windows ホストへのアクセスを拒否できます。また、組織またはロールのレベルで共通ホストのセットを指定し、その組織またはロールのすべてのユーザーが共通ホストにアクセスできるようにして、ユーザーが各組織またはロールのすべてに自分を追加する必要をなくすこともできます。

NetFile サービスの一部として、許可される URL リストまたは拒否される URL リストを設定し、組織、ロール、またはユーザーのレベルでサーバーへのアクセスを許可または拒否できます。拒否される URL リストは、許可される URL リストよりも優先されます。許可される URL リストと拒否される URL リストには、ワイルドカード文字 * を使用して、同一ドメインまたはサブドメイン内のサーバーセットへのアクセスを許可または拒否できます。

セキュリティー

SSL 対応の SRA を有効にして NetFile を使用すると、NetFile アプレットから背後のファイルシステムへのすべての接続は、ゲートウェイとブラウザの間に確立された SSL 接続を使用して実行されます。通常は、ゲートウェイを DMZ に設置し、第 2 ファイルウォールで開くポートの数を制限する (通常は 1 つのみ) ので、ファイルシステムへのアクセスを許可してもセキュリティーは低下しません。

特殊操作

NetFile は、一般的なファイル管理アプリケーションと同様、リモートファイル管理アプリケーションに適した機能セットを備えています。NetFile によりユーザーは、ローカルとリモートのファイルシステムの間でファイルをアップロードまたはダウンロードできます (共有)。ローカルからリモートファイルシステムにアップロードするファイルのサイズは、Access Manager 管理コンソールを使用して制限できます。

またユーザーは、複数のファイルを選択し、GZIP および ZIP で圧縮することも可能です。複数のファイルを選択し、複数の添付ファイルとして 1 通の電子メールで送信できます。NetFile は、Access Manager の SSO トークンを使用し、IMAP サーバー、ユーザー名、パスワード、および返信アドレスなどのユーザーの電子メール設定にアクセスして、電子メールを送信します。

NetFile ウィンドウのファイルをダブルクリックすると、MIME タイプに対応したアプリケーションが起動し、ファイルが開きます。NetFile には、一般的なほとんどのファイルタイプ (拡張子) をマッピングしたデフォルトの MIME タイプ設定ファイルと、編集して新規マッピングを追加できる MIME タイプがあります。

NetFile を使用してファイルを検索し、別のウィンドウにリストを表示できます。各検索結果は、以前の検索結果のウィンドウを表示した状態で、新規ウィンドウに表示されます。特定の共有で使用される文字エンコードのタイプはユーザーが設定でき、共有の設定項目の一部になっています。文字エンコード方法を指定しないと、共有機能で動作する間、NetFile は ISO-8859-1 を使用します。ISO-8859-1 エンコードは、ほとんどの共通言語を処理することができます。ISO-8859-1 エンコードにより、NetFile は任意の言語でファイルのリストを作成し、内容を破壊せずに任意の言語でファイルを変換できます。

NetFile は、ファイルを電子メールで送信する場合にのみ、NetFile Java 1 と Java 2 の両方で一時ファイルを作成します。一時ファイルは、Microsoft Windows ファイルシステムとローカルシステムファイルの間で、jCIFS プロトコルによってファイルをアップロードおよびダウンロードする時には作成されません。

注 NetFile は、ディレクトリとリモートファイルの削除をサポートします。リモートディレクトリ内のすべての内容は、再帰的に削除されます。

NetFile とマルチスレッド化

NetFile では、マルチスレッドを使用して、複数の操作を同時に実行する際の柔軟性を高めています。たとえば、ユーザーは検索操作を行い、ファイルのアップロードを開始して、電子メールでファイルを送信できます。NetFile はこれら 3 つの操作を同時に実行し、さらにユーザーがファイルのリストを参照することを許可します。

リライタ

リライタは、HTML と JavaScript コードの両方ですべての URI を変換し、イントラネットのコンテンツを常にゲートウェイを通して取得できるようにする独立コンポーネントです。ルールの集合であるルールセットを定義し、ページにリライトする必要のあるすべての URL を特定します。ルールセットは、Document Type Definition (DTD) に従って記述される XML コードです。リライタに付属する一般ルールセットを使用すれば、ルールを追加せずにほとんどの URL をリライトできます。ルールセットをドメインに関連付け、ドメインベースの変換も実行できます。詳細については、『Portal Server Secure Remote Access 6 管理ガイド』を参照してください。

外部ルールセットはコンテンツの URI を識別します。SRA によるサービスを必要とするは要求すべて、次の経路で処理されます。

1. SRA は、サービスを必要とするイントラネットページまたはインターネットページの URI を要求から特定します。
2. SRA は、プロキシ設定を使用して特定された URI に接続します。
3. URI のドメインは、このコンテンツをリライトするために使用するルールセットを特定するために使用されます。
4. コンテンツとルールセットを取得すると、SRA はそれらの情報をリライタに入力し、特定された URI はそこで変換されます。
5. 元の URI はリライトされた URI によって置き換えられます。
6. このプロセスがドキュメントの最後まで繰り返されます。
7. 生成されたリライタ出力は、ブラウザに経路指定されます。

リライタプロキシ

ファイアウォールで開くポート数を最小限に抑えるには、リライタプロキシを使用します。リライタプロキシをインストールすると、HTTP 要求は、宛先ホストに直接送信される代わりに、リライタプロキシにリダイレクトされます。次にリライタプロキシは、受け取った要求を宛先サーバーに送信します。

リライタプロキシを使用すると、ゲートウェイとイントラネットコンピュータの間の HTTP トラフィックをセキュリティー保護し、次の 2 つの利点を生かすことができます。

- ファイアウォールがゲートウェイとサーバーの間にある場合、ファイアウォールで開く必要のあるポートは 2 つだけです。1 つのファイアウォールをゲートウェイとリライタプロキシの間に、もう 1 つをゲートウェイと Portal Server の間に配置します。
- サードパーティーのプロキシを使用して第 2 ファイアウォールのポートを 1 つだけ使用し、リライタプロキシを読み取ることができます。
- 宛先サーバーが HTTPS ではなく HTTP プロトコルのみをサポートしている場合でも、ゲートウェイとイントラネットの間の HTTP トラフィックはセキュリティー保護されます。

注 複数のリライタプロキシを稼働させれば、シングルポイント障害を防止し、ロードバランスを実現できます。

プロキシレット

プロキシレットは、クライアントマシン上で稼働する動的なプロキシサーバーです。プロキシレットは URL をゲートウェイにリダイレクトします。プロキシレットは、この機能を実現するために、クライアントマシン上のブラウザのプロキシ設定を読み込んでから、その設定がローカルプロキシサーバー (プロキシレット) をポイントするように変更します。

プロキシレットでは、HTTP および SSL がサポートされ、ゲートウェイのトランスポートモードが継承されます。ゲートウェイが SSL に基づいて動作するように設定されている場合には、クライアントマシンとゲートウェイ間のチャネルのセキュリティーが確保されます。クライアントの JVM が 1.4 以降の場合または必要な jar ファイルがクライアントマシン上にある場合には、JSSE API が使用されます。それ以外の場合には、KSSL API が使用されます。

プロキシレットは、クライアントの IP アドレスとポートが指定されている Access Manager 管理コンソールから有効にします。

プロキシレットは、リライタと異なり、インストール後の変更をほとんどまたはまったく必要としません。また、プロキシレットは Web コンテンツを処理しないので、ゲートウェイのパフォーマンスが向上します。

第 3 章

ビジネス要件と技術要件の特定と評価

配備計画の最初のステップは、Sun Java™ System Portal Server のビジネス要件と技術要件を見極めることです。アーキテクチャーと設計の問題に取り組む前に、ビジネス要件と技術要件に関する情報を収集する必要があります。

この章で説明する内容は次のとおりです。

ビジネスの目的

ビジネス要件では、組織の問題と機会について述べ、次のような要素について考慮します。

- サービス
- サービスの可用性
- 将来の成長
- 新しいテクノロジ
- 投資規模

設計要件を作成する際に役立つものとするには、ビジネス要件で具体的な目標と目的について記述する必要があります。

ポータルのビジネス目標は、配備に関する決定に影響を与えます。目的を理解してください。ビジネス要件を把握していないと、誤った前提条件を設定しやすくなり、配備見積もりの正確さに影響を与えかねません。

次の点について自問し、ビジネスの目的を見極めてください。

- このポータルのビジネス目標は何か。たとえば、顧客へのサービスを強化することか。または、社員の生産性を向上したり、営業コストを削減したりすることか。
- どんな種類のポータルが必要か。たとえば、企業間、企業対顧客、企業対企業、またはこれらの混合か。
- 対象利用者はだれか。
- ポータルによりどのようなサービスまたは機能をユーザーに提供するのか。
- 対象利用者はポータルからどのように利益を得るのか。
- ポータルの優先度はどれほどか。ポータルを段階的に配備する計画の場合は、各段階の優先順位を定める。

(オプション) セキュリティー保護されたポータルを配備する場合は、次の点を自問してビジネスの目的を見極めます。

- イントラネットのアプリケーションやサーバーにインターネットからアクセスできるようにして、社員の生産性を高める必要があるか。
- ポータルへのアクセスをセキュリティー保護する必要があるか。
- 既存の Virtual Private Network (VPN) ソリューションの所有コストを削減する必要があるか。
- Citrix や pcAnywhere などのイントラネットアプリケーションに、インターネットからアクセスする利便性を社員に提供するか。
- イントラネットのサーバーやマシンを、インターネットから参照する利便性を社員に提供するか。
- 対象利用者は、すべてのポータルユーザー、社員、顧客などのうちだれか。

技術目標

技術要件 (機能要件とも呼ばれる) では、組織のシステムの必要と期待される結果について詳しく述べ、次の要素について考慮します。

- パフォーマンス
- セキュリティー
- 信頼性
- ポータルに期待されるパフォーマンス基準

技術要件では、アーキテクチャーに必要なすべての機能を定義し、各コンポーネントの働きとそれらを統合してシステム全体を構成する仕組みについて説明します。組織は、最善の設計手法を作成し、適切な技術を適用してポータルに適したアーキテクチャーソリューションを導き出すための技術要件が必要です。

ポータルを提供する理由は、ポータルの実装方法に直接影響します。対象となる利用者、パフォーマンス標準、および目標に関連する他の要素について定義する必要があります。

次の点について自問し、ポータルの目標を見極めてください。

- ポータルでもっとも優先されるものは何か。
- ポータルが配信するアプリケーションは何か。
- どのような利用者を対象とするか。
- どの程度のパフォーマンス標準が必要か。
- どれほどのトランザクションの量が予期されるか。ピーク使用時にはどれほどのトランザクションの量が予期されるか。
- ピーク使用時に許容される応答時間はどれほどか。
- どの程度の並行性が必要か。並行性は、任意の時点で接続可能なユーザー数。
- ポータルへのアクセスには、イントラネットとインターネットのどちらを使用するのか。
- ポータルは、1 段階で、または数段階で配備するのか。各段階について、および段階ごとの変化について説明する。

Portal Server の機能とビジネスの必要性の対応付け

前のセクションでは、ビジネスと技術の必要を概観して、Portal Server システムのさまざまな領域について自問できる点を紹介しました。このセクションでは、組織にとって重要性の高いテクノロジを判断することを目標に、特定のテクノロジの特長について考慮します。組織の当面および中長期的な計画を念頭に、これらの機能について検討してください。

続くセクションとそれぞれの表を参考にしてリストにある機能の利点を評価し、組織にとっての相対的な優先順位を定めてください。この情報は、タイミングよく費用対効果に優れた方法で配備計画を作成するのに役立ちます。

注　おそらく、Sun Java System の販売担当者との間でこれらのトピックについての話し合いは以前に行われています。したがってこのセクションでは、そのプロセスについて再検討します。

アイデンティティー管理

Portal Server は、アイデンティティー管理を使用して、コンテンツ、アプリケーション、およびサービスにアクセスする際、組織全体で、時には組織外でさまざまなロールを持つ数多くのユーザーを管理します。課題としては、だれがアプリケーションを使用するのか、ユーザーはどんな能力範囲で組織または企業に労働力を提供するのか、ユーザーの使命は何か、ユーザーは何にアクセスする権限をもつべきか、他の人は管理作業をどのように支援できるか、などの点が挙げられます。

表 3-1 に、アイデンティティー管理機能とその利点を示します。

表 3-1 アイデンティティー管理機能と利点

機能	説明	利点
ディレクトリサービス	Portal Server は Access Manager と Directory Server を使用します。	Portal Server は、認証、シングルサインオン (SSO)、管理の委任、およびパーソナライズの目的で、LDAP ディレクトリを使用し、ユーザープロファイル、ロール、および識別情報を保存します。 Portal Server は中央集中型のユーザーディレクトリに常駐可能なオープンスキーマを使用することにより、Access Manager および Directory Server 製品に対する企業またはサービスプロバイダの投資を有効活用します。

表 3-1 アイデンティティー管理機能と利点 (続き)

機能	説明	利点
ユーザー、ポリシー、およびプロビジョニング管理	Access Manager により、コンテンツ、アプリケーション、およびサービスにアクセスする間、組織全体で、時には組織外でさまざまなロールを持つ数多くのユーザーを管理できます。	識別情報の保存と管理を行う中央集中型のアイデンティティー管理ソリューションです。ポリシーソリューションと統合してアクセス権を制御し、これらの課題を大幅に簡素化します。共通アイデンティティーを拡張して新規アプリケーションを処理し、アプリケーションで管理作業を共有できるようにします。また、普通ならこれらのサービスを最初から構築することに関連付けられるタスクを簡素化します。 ユーザーとアプリケーションの管理を統合します。コンテンツとサービスの配信をパーソナライズします。情報とサービスへのアクセスを簡素化し、無駄を省きます。アクセスおよび配信の管理に関係するコストを削減します。 アプリケーションへのアクセスをポリシーベースで行い、セキュリティーで保護します。ポータル配備が社員の LAN アクセス範囲を超えて拡張される場合でも、アクセスを確実にセキュリティー保護します。
シングルサインオン (SSO)	Access Manager は、SSO API によってユーザー認証とシングルサインオンを統合します。一度認証されたユーザーは、SSO API が継承します。認証されたユーザーが保護されたページへアクセスしようとすると、毎回 SSO API は、認証クレデンシャルに基づいてユーザーが適切なアクセス権を持っているかどうかを判断します。ユーザーが有効であれば、追加認証なしでページへのアクセスが許可されます。無効であれば、ユーザーは再認証するように求められます。	社員、パートナー、および顧客による、コンテンツ、アプリケーション、およびサービスへのアクセスを有効にしながら、認証とシングルサインオンを管理する一貫した中央集中型メカニズムを提供することによって、ユーザーの生産性を向上させます。

表 3-1　アイデンティティー管理機能と利点 (続き)

機能	説明	利点
管理の委任	Access Manager 管理コンソールは、ロールベースの管理の委任の機能を異なる種類の管理者に割り当て、指定されたアクセス権に基づいて組織、ユーザー、ポリシー、ロール、チャネル、およびポータルデスクトッププロバイダを管理します。	IT がポータル管理の任務を委任して、貴重な IT リソースと管理を解放できるようにします。
セキュリティー	ポータルでの集約アプリケーションのシングルサインオンを可能にします。	セキュリティーはポータルにとって重要な機能です。セキュリティーによってポータル内のさまざまな必要を満たすことができます。たとえば、ポータルへの認証、ポータルとエンドユーザー間の通信の暗号化、アクセス権を持つユーザーに限定したコンテンツとアプリケーションの承認などがあります。

SRA

表 3-2 に、Sun Java System Portal Server Secure Remote Access (SRA) 機能とその利点を示します。

表 3-2　SRA の機能と利点

機能	説明	利点
統合セキュリティー	ユーザー、ポリシー、および認証サービスを提供すると同時に、エクストラネット機能と Virtual Private Network (VPN) 機能を「オンデマンド」で提供します。ゲートウェイコンポーネントは、インターネットから送信されるリモートユーザーセッションと企業イントラネットの間のインタフェースおよびセキュリティーバリアとして機能します。	ファイアウォールの背後に配置された企業のコンテンツ、アプリケーション、ファイル、およびサービスを、許可されているサプライヤ、ビジネスパートナー、および社員に提供します。 サービス妨害攻撃を防ぐため、内部と外部の両方の DMZ ベースゲートウェイを使用できます。

機能	説明	利点
SRA コア	ユーザーは次の 4 つのコンポーネントを通してリモートアクセスできます。 • ゲートウェイ • NetFile • Netlet • ホスト	このコンポーネントには 4 つの構成要素があります。 • ゲートウェイ - Portal Server とさまざまなゲートウェイインスタンスの間の通信を制御します。 • NetFile - ファイルシステムとディレクトリへのリモートアクセスと操作を可能にします。 • Netlet - クライアントブラウザの Netlet アプレット、ゲートウェイ、およびアプリケーションサーバー間の通信を確実にセキュリティー保護します。 • プロキシレット - URL をゲートウェイにリダイレクトします。
汎用アクセス	クライアントソフトウェアをインストールせずに、またはメンテナンスの必要をなくして、Web ブラウザベースの汎用アクセスを可能にします。	配備にかかる時間とコストを大幅に削減すると同時に、IT 管理およびメンテナンスのオーバーヘッドを簡素化します。
Netlet プロキシ	クライアントからゲートウェイ、およびイントラネットに常駐する Netlet Proxy までのセキュリティー保護されたトンネルを拡張するオプションコンポーネントを提供します。	非武装地帯 (DMZ) とイントラネットの間に配置されたファイアウォールの開いているポート数を制限します。

機能	説明	利点
リライタプロキシ	HTTP 要求を、宛先ホストに直接送る代わりに、リライタプロキシにリダイレクトします。次にリライタプロキシは、受け取った要求を宛先サーバーに送信します。	リライタプロキシを使用すると、ゲートウェイとイントラネットコンピュータの間の HTTP トラフィックをセキュリティー保護し、次の 2 つの利点を生かすことができます。 • ファイアウォールがゲートウェイとサーバーの間に配置されている場合、ファイアウォールで開放する必要のあるポートは、ゲートウェイとリライタプロキシの間のポートと、ゲートウェイと Portal Server の間のポートの 2 つのみです。 • 送信先のサーバーが、HTTPS ではなく HTTP プロトコルのみをサポートしている場合でも、ゲートウェイとイントラネットの間の HTTP トラフィックは安全です。

検索エンジン

検索エンジンサービスは、次のチャネルで使用されます。

- ユーザーが定義したカテゴリ別ドキュメントやディスカッションの各プロファイルエントリと一致したヒット件数 (関連情報) を要約する登録チャネル。
- 個別にコンテンツを検索し、コメントの重要度を設定するディスカッションチャネル。

表 3-3 に、検索機能とその利点を示します。

表 3-3 検索機能と利点

機能	説明	利点
検索エンジン	エンドユーザーが指定する条件に基づいてドキュメントを取得できます。	コンテンツにアクセスしてユーザーの時間を節約します。
カテゴリ化	ドキュメントを階層構造に整理します。このカテゴリ化は分類と呼ばれることもあります。	参照と取得が可能なドキュメントをさまざまに表示します。

表 3-3　検索機能と利点 (続き)

機能	説明	利点
ロボット	検索エンジンロボットは、イントラネットまたはインターネット全体の情報を見回りインデックスを作成するエージェントプログラムです。	リソースへのリンクを自動的に検索して抽出し、それらのリソースについて説明し、説明を検索データベースに格納します。これは、生成またはインデックス作成とも呼ばれます。
ディスカッション	複数のスレッド化されたディスカッションの場です。	コンテンツは個別に検索でき、すべてのコメントに重要度が指定されます。
登録	異なる関心領域の新規または変更された素材をユーザーが追跡できるようにします。	ディスカッション、検索カテゴリ、および自由形式検索 (保存された検索) を追跡できます。

パーソナライズ

パーソナライズは、選択した基準に基づいてコンテンツを配信し、ユーザーにサービスを提供する機能です。

表 3-4 に、パーソナライズ機能とその利点を示します。

表 3-4　パーソナライズ機能と利点

機能	説明	利点
ユーザーのロールに基づくコンテンツ配信	Portal Server には、組織内でのユーザーのロールに基づき、ユーザーがアクセスまたは使用可能なアプリケーションを自動的に選択する機能が組み込まれています。	コンテンツとサービスへのカスタマイズされた迅速なアクセスを可能にすることにより、社員の生産性を上げ、顧客関係を改善し、円滑なビジネス関係を促進します。
ユーザーによるコンテンツのカスタマイズ	Portal Server により、エンドユーザーは関心のあるコンテンツを選択して表示できます。たとえば、個人金融ポータルのユーザーは、財務ポートフォリオを表示する際に、閲覧する株価情報を選択できます。	ポータル内で表示可能な情報は、個別にパーソナライズされます。さらにユーザーは、その情報をカスタマイズして、自分の好みに合わせることができます。ポータルは、Web 体験の制御を、Web サイトの作成者ではなく Web の利用者の手に委ねます。
複数ユーザー対象コンテンツの集約とカスタマイズ	Portal Server により、企業またはサービスプロバイダは、パーソナライズされたコンテンツを集約し、複数のユーザーコミュニティーに同時に配信できます。	これにより、企業は 1 つの製品から複数の製品に複数のポータルを配備し、集中管理コンソールから管理できます。また、新規コンテンツとサービスを追加し、Portal Server を再起動せずにオンデマンドで配信できます。このすべてにより時間と資金が節約され、IT 組織の一貫性が確保されます。

集約と統合

ポータルの重要な役割に数えられるのは、アプリケーション、サービス、およびコンテンツなどの情報を集約し、統合する機能です。この機能には、株価情報などの変動する情報をポータルを介して埋め込み、ポータル内でアプリケーションを実行し、ポータルを通してそれらのアプリケーションを配信する能力が組み込まれています。

表 3-5 に、集約および統合機能とその利点を示します。

表 3-5 集約機能と利点

機能	説明	利点
集約情報	ポータルデスクトップは、Portal Server のプライマリエンドユーザーインタフェースであり、プロバイダアプリケーションプログラミングインタフェース (PAPI) による広範なコンテンツ集約のメカニズムを備えています。ポータルデスクトップには、コンテナ階層と、特定のチャネルを構築するための基本構築ブロックとを有効にするさまざまなプロバイダが表示されます。	ユーザーが情報を検索する必要はありません。代わりに、情報がユーザーを見つけます。
一貫したツールセット	ユーザーは、会社にいる間はずっと付いて回る、Web ベースの電子メールやカレンダ作成ソフトウェアのようなツールセットを使用できます。	ユーザーは、あるプロジェクトで 1 つのツールを、別の場所で別のツールを使用する必要がありません。また、これらのツールはすべてポータルのフレームワークで動作するので、ツールの見た目と使い心地、操作性に一貫性を持たせることでトレーニング時間を削減できます。
コラボレーション	Portal Server により、企業全体のリソースとしてデータを制御し、アクセスできます。	多くの企業は、個々の部門によって所有されるものとしてデータを考え、企業全体のリソースとは見ていません。ポータルは、こうしたデータの管理方法を変革し、データを必要とするユーザーが制御された方法で入手できるようにする触媒のような役割を果たします。この広範でより迅速なアクセスを可能にする仕組みにより、コラボレーションを促進します。
統合	Portal Server により、ユーザーがアプリケーションにアクセスしたり、アプリケーションを起動したり、データにアクセスしたりする際の唯一の場所としてポータルデスクトップを使用できます。	既存の電子メール、カレンダ、旧バージョン、または Web の各アプリケーションを統合することにより、統一されたアクセスポイントとしてポータルを活用し、社員、パートナー、または顧客などのユーザーが、必要とする情報にすばやく簡単にアクセスできるようにします。

ユーザーの動作と行動パターンについての理解

ポータルを利用するユーザーについて調査してください。ユーザーがポータルを利用する状況や前システムの使用方法などの要素は、ポータルの要件を識別するための鍵となります。組織にそうした行動パターンを提供できるだけの経験がない場合は、他の組織の経験を調査して評価します。

次の点を自問してユーザーを理解してください。

- 何人のエンドユーザーが利用するか。対象利用者はどれぐらいの規模か。
- ユーザーは毎日、同じ時刻にポータルにログインするか。ユーザーは仕事でポータルを使用するのか、それとも別の場所で使用するのか。
- ユーザーたちは、同じタイムゾーンまたは異なるタイムゾーンに属しているか。
- 標準的なユーザーの接続時間、または有効なポータルセッションが開いている時間としてどれぐらいの長さが予想されるか。既存のアプリケーションに関する使用方法の統計情報を何か持っているか。既存ポータルの Web トラフィック分析結果があるか。
- 事前定義された時間内に予想される、訪問者セッションの数、または 1 人の訪問者の訪問回数はどの程度か。
- ポータルの利用頻度は、時間の経過とともに増加することが期待できるか。または、安定した状態を維持しそうか。
- どれぐらいの割合でユーザーベースが成長するか。
- ユーザーは、ポータルで配信しようとするアプリケーションをどのように使用してきたか。
- ユーザーが定期的に使用すると期待されるのはどのポータルチャネルか。
- ユーザーはポータルのコンテンツに何を期待しているか。ポータルが提供するものの中で、前身となる Web ベース情報または他のリソースをユーザーはどのように使用してきたか。

第 4 章

配備前の注意点

この章で説明する内容は次のとおりです。

チューニング目標の決定

ポータルのチューニングに入る前に、ポータルシステム管理者およびポータル開発者と話し合い、計画される要件に基づいてポータルのパフォーマンス目標を設定します。目標には、ユーザー数、負荷ピーク時の並行処理ユーザー数、および Sun Java™ System Portal Server にアクセスする際のユーザーの使用パターンなどがあります。

次の 2 つの要素を決定する必要があります。

- ポータルアプリケーションの迅速な応答を目指してチューニングするのか。
- 多数のユーザーの並行処理に対応できるようにチューニングするのか。

  並行してポータルに接続するユーザー数が増加すると、ハードウェアとパラメータセットが同じであれば、応答時間が減少します。そのため、Sun Java System Portal Server の予想使用レベル、任意の時刻に予想される並行処理ユーザー数、ポータルデスクトップのアクティビティー要求数、ポータルチャネル使用量、組織によって決定されるエンドユーザーへの許容応答時間、および基準を満たす最適ハードウェア構成に関する情報を収集してください。

ポータルのサイジングのヒント

このセクションでは、サイジングプロセスで役立つヒントを紹介します。

- 企業対顧客用ポータルでは、ゲートウェイと SSL を使用するために SRA を配備する必要があります。サイジング要件には、必ずこの点を盛り込んでください。SSL を有効にすると、SSL なしの場合に比べ、ポータルのパフォーマンスが最大で 1/10 にまで低下します。
- 企業対社員用ポータルの場合は、基準として使用できるユーザープロファイルがあることを確認してください。
- どんなポータルであっても、余裕をもって構築して成長に備えます。つまり、現状の必要に合わせてサイジングするだけでなく、将来の必要と能力も考慮に入れます。これには、週末や祝日などの休日明けに生じる通常のピークの場合、またはポータルがより「スティッキー」であるために時間の経過とともに使用量が増加する場合などがあります。
- 地理的に複数のサイトに渡ってポータルソリューションを配備する場合は、ネットワークとデータセンターの配置を十分理解している必要があります。
- 必要な冗長機能のタイプを判断してください。生産ダウンタイム、アップグレード、およびメンテナンス作業などの要素を考慮します。ポータルサーバーの運用を停止する場合は、通常、処理能力に与える影響が全体の処理能力の 1/4 を超えないようにします。
- 一般に、企業対社員用ポータルの使用並行性は、企業対顧客用ポータルの場合より高くなります。

パフォーマンス方法論の確立

パフォーマンス目標を設定したら、次の手順に従ってポータル環境をチューニングします。

1. プロセッサ、メモリー、ネットワーク、およびディスクの明らかな障害を識別して取り除きます。
2. 制御された環境をセットアップして、同一条件での動作変動が 10 パーセント未満に定義されるエラーの許容範囲を最小限に抑えます。

   開始データの測定基準が分かれば、サンプル収集動作間のデータパフォーマンスの変動を測定できます。測定値は適切な長さの時間で収集し、これらのテストの結果を捕捉して評価できるようにします。

   Portal Server マシンとは別に、負荷シミュレーションデータを収集する専用マシンの使用を計画します。専用マシンは、パフォーマンス問題の原因を突き止めるのに役立ちます。

65 ページの「ポータルのサイジング」を参照してください。

3. ポータルの提供者、管理者、および開発者が同意する予想生産環境をできるだけ正確にシミュレーションする、プロトタイプワークロードを作成して微調整します。

   141 ページの「分析ツール」を参照してください。

4. ポートレットのようなカスタマイズされたポータルアプリケーションを監視します。

ポータルのサイジング

Portal Server の基準サイズを設定する必要があります。基準サイズを設定し、その数値を検証および微調整することで、スケーラビリティーな、高可用性、信頼性のある、優れたパフォーマンスを実現できます。

ポータルのサイジングプロセスは、以下のステップで構成されます。

1. 基準サイズの確立
2. 基準サイズのカスタマイズ
3. 基準サイズの検証
4. 基準サイズの微調整
5. 最終基準サイズの検証

以下のセクションではこれらのステップについて説明します。

基準サイズの確立

ビジネス要件と技術要件を特定し、Portal Server の機能と必要事項をマッピングすると、全体的な Portal Server の配備計画が進むにつれ、サイジング要件も判明してきます。設計上の決定事項は、Portal Server のユーザーセッションと並行性に関する正確な見積もりを作成する際に役立ちます。

注 Sun Java System Portal Server Secure Remote Access (SRA) ソフトウェアを使用した安全性の高いポータル配備のサイジング要件は、74 ページの「SRA サイジング」で説明されています。

Sun Java System の技術担当者は、Portal Server の配備で必要になる CPU の数の見積もりを算出する自動サイジングツールを提供します。サイジングツールへの入力値として、以下のメトリクスを収集する必要があります。

- ピーク数
- ページ要求間の平均時間
- 並行処理ユーザー
- 平均セッション時間
- 検索エンジンの要素

Portal Server の配備で必要になる CPU の数に影響を与えても、サイジングツールでは使用されないその他のパフォーマンスメトリクスは次のとおりです。

- ポータルデスクトップ設定
- ハードウェアとアプリケーション
- バックエンドサーバー
- トランザクション時間
- ワークロード条件

これらのパフォーマンス要因については、続くセクションで説明します。

ピーク数

最大並行セッション数は、配備される Portal Server で処理可能な接続ユーザー数を定義します。

最大並行セッション数を計算するには、次の計算式を使用します。

最大並行セッション数 = オンラインユーザーの予想割合 * ユーザーベース

企業ポータルの見込みユーザーのユーザーベースサイズまたはプールサイズを特定するには、次の提案を参考にしてください。

- アクティブなユーザーのみを識別します。たとえば、休暇や休日をとっているユーザーは対象にしません。
- ユーザーベースには有限数を使用します。匿名ポータルの場合は、この数字を控えめに見積もります。
- アクセスログを調べます。
- ユーザーベースの地理的な場所を識別します。
- ビジネス計画の中でユーザーについて記述した内容について忘れないでいてください。

ページ要求間の平均時間

ページ要求間の平均時間は、Portal Server からユーザーがページを要求する頻度の平均です。ページには、ポータルにログインしたときの最初のログインページや、ポータルデスクトップからアクセスする Web サイトや Web ページなどがあります。1 ページの表示とは、ページに配置されているアイテム数に関係なく、1 回の呼び出しで表示される 1 つの情報ページを指します。

Web サーバーのログにはページ要求が記録されますが、ログを使用して要求から要求までの平均時間をユーザー単位で計算することはできません。ページ要求間の平均時間を計算するには、WebLoad パフォーマンステストツールのような市販の統計ツールが必要です。ツールを使用して得られた数値を基にして、並行処理ユーザー数を判断できます。

注 ページ要求を基準にすると、「ヒット数」を基準にするよりも Web サーバーのトラフィックを正確に測定できます。Web サーバーからのファイル要求は、1 ヒットとして毎回計上されます。ページにあるすべてのアイテムが登録されているので、1 回のページ呼び出しで何度もヒットが記録されます。たとえば、10 個のグラフィックファイルが組み込まれているページの場合は、HTML ページ自体で 1 回、および 10 個のグラフィックファイルそれぞれが 1 回ずつカウントされ、合計 11「ヒット」が記録されます。したがって、ページ要求を基準にしたほうが Web サーバーのトラフィックをより正確に判断できます。

並行処理ユーザー

並行処理ユーザーは、実行中の Web ブラウザプロセスに接続し、Portal Server に要求を送信するか、要求の結果を受信しているユーザーです。最大並行処理ユーザー数は、あらかじめ定義された時間内に接続可能な並行処理ユーザーの最大数です。

最大並行処理ユーザー数は、最大並行セッション数を算出してから計算します。最大並行処理ユーザー数を計算するには、次の計算式を使用します。

```
並行処理ユーザー =
並行セッション数 / ヒット間の平均時間
```

たとえば、ユーザーが 50,000 人のイントラネット Portal Server の例について考えます。負荷ピーク時に接続されているセッション数を、登録されたユーザーベースの 80% と見積もります。ユーザーは、平均で 10 分に 1 回の割合でポータルデスクトップにアクセスします。

この例の場合の計算方法は次のようになります。

```
40000 / 10 = 4000
```

したがって、この Portal Server の負荷ピーク時に接続できる最大並行処理ユーザー数は 4,000 人になります。

平均セッション時間

平均セッション時間は、多くのユーザーを対象に算出した、ログインからログアウトまでの平均時間です。セッション時間の長さは、発生するログイン数に反比例します。つまり、セッション期間が長くなると、同じ並行処理ユーザーベースでの Portal Server に対する毎秒のログイン数が少なくなります。セッション時間は、ユーザーのログインからログアウトまでの時間です。

平均セッション時間は、多くの場合、ユーザーの Portal Server の使い方によって変化します。たとえば、対話型のアプリケーションが関係するユーザーセッションの場合は、情報のみのユーザーセッションの場合よりも、セッション時間が長くなるのが一般的です。

検索エンジンの要素

ポータルサイトで検索チャネルを提供する場合は、検索エンジンのサイジング要素をサイジングの計算に含める必要があります。検索エンジンのサイジング要件は、以下の要素によって決まります。

- インデックスディレクトリのアクティブリストにあるインデックスパーティションのサイズ

  パーティションサイズは、インデックス付きの検索可能な用語のサイズと数に正比例します。

- リソース記述 (RD) に必要な平均ディスクスペース

  次の式を使用して計算します。

  必要な平均ディスクスペース =
  データベースサイズ / データベース内の RD 数

  平均サイズは、RD サイズの変動に合わせて調整されます。多くのインデックス付き用語を使用した長くて複雑な RD の集合と、少数のインデックス付き用語による短い RD のリストでは、複雑な RD に同じ数の RD があっても、必要な検索時間が異なります。

  RD は階層型データベース形式で保存されます。この形式では、RD が保存されていなくても、データベースの組み込みサイズを考慮する必要があります。

- 検索関連のアクティビティーを実行する並行処理ユーザーの数

  次の式を使用して計算します。

  並行処理ユーザー数 / 検索ヒット間の平均時間

  67 ページの「ページ要求間の平均時間」で計算される並行処理ユーザー数の値を使用します。

- 使用する検索演算子のタイプ

  検索関数のタイプには、基本、結合、近接、パッセージとフィールド演算子、およびワイルドカードスキャンなどがあります。それぞれの検索関数では、異なる検索アルゴリズムとデータ構造を使用します。検索アルゴリズムとデータ構造の違いは、検索用語数とインデックス付き用語数の増加するのにつれて大きくなるので、検索関数のタイプは検索結果が返されるまでの時間に影響します。

ヒント ここまでで、上述の数値を技術担当者に伝え、サイジングツールを実行して CPU の見積もり数を算出するように依頼できます。

ポータルデスクトップ設定

ポータルデスクトップの設定によって、セッション単位でメモリーに保持するデータ量が明確に決定されます。

ポータルデスクトップのチャネルが多くなるほど、データのセッションサイズは大きくなり、Portal Server のスループットが低下します。

もう 1 つの要素は、ポータルデスクトップで提供される対話機能の性能です。たとえば、チャネルをクリックすると、Portal Server または他の外部サーバーに負荷が発生します。チャネルの選択によって Portal Server に負荷が発生すると、ポータルデスクトップをホストするノードのユーザークティビティープロファイルと CPU オーバーヘッドは、他の外部サーバーをホストするノードの場合より高くなります。

ハードウェアとアプリケーション

CPU 速度と、Java™ プラットフォーム (Java™ 仮想マシンまたは JVM™ ソフトウェア) のメモリーヒープにおける仮想マシンのサイズは、Portal Server のパフォーマンスに影響を与えます。

CPU 速度が速くなれば、スループットも高くなります。ヒープ生成チューニングパラメータとともに、JVM メモリーヒープのサイズも Portal Server のパフォーマンスを左右します。

バックエンドサーバー

Portal Server は外部ソースからコンテンツを集約します。外部のコンテンツプロバイダが、最大速度で動作する Portal Server に必要な帯域を確保できない場合、ポータルデスクトップのレンダリングとスループットの要求時間は最適化されません。ポータルデスクトップは、ブラウザに要求応答を返す前に、すべてのチャネル動作が完了するかタイムアウトになるまで待機します。

次のチャネルを使用する場合は、バックエンドのインフラストラクチャーを慎重に計画してください。

- 外部ソースからコンテンツを収集する
- 一般に応答時間の遅い企業データベースにアクセスする
- 電子メールコンテンツを提供する
- カレンダコンテンツを提供する

トランザクション時間

トランザクション時間は、HTTP または HTTPS 処理が完了するまでの遅延時間で、送信時間、処理時間、および応答時間を合計した時間です。

トランザクション時間に影響する要素について計画する必要があります。これには、次の要素があります。

- ネットワーク速度と待ち時間。

  ワイドエリアネットワーク (WAN) の場合は、特に待ち時間について検討する必要があります。大量のデータの場合には、待ち時間により取得時間が著しく長くなる可能性があります。

- ポータルデスクトップの複雑さ。
- ブラウザの接続速度。

  たとえば、接続速度が 33.6 kbps の場合は、LAN 接続の場合よりも応答時間の遅延が長くなります。ただし、処理時間は一定です。ダイヤルアップ接続によるトランザクション時間は、データ圧縮を実行するロード生成ツールによって表示されるトランザクション時間よりも短くなります。

トランザクション時間を計算する場合は、Portal Server のサイジングを行なって、通常またはピーク時の負荷状態がパフォーマンス要件のしきい値を超えないように、また時間が経過しても処理時間を維持できるようにします。

ワークロード条件

ワークロード条件は、システムにおいてもっとも集中的に使用される、システムリソースと JVM ソフトウェアリソースです。これらの条件は、大部分がユーザーの動作と配備するポータルのタイプによって決まります。

Portal Server ソフトウェアで一般的に発生するワークロード条件は、次の要素に影響を与えます。

- システムパフォーマンス

  Portal Server のパフォーマンスは、アクティビティープロファイルが高い場合など、大量の並行要求が処理される場合に影響を受けます。たとえば、企業対企業ポータルの負荷ピーク時には、同時に大勢の社員がポータルに接続します。そうした状況では、CPU にワークロードが集中します。また、接続されたユーザーに対する並行処理ユーザーの割合も高くなります。

- システム能力

  Portal Server の処理能力は、大勢のユーザーがログインすると影響を受け始めます。ログインするユーザーが多くなると、ユーザーがより多くのメモリー領域を使用するので、サーバーで要求を処理するために使用可能なメモリーが少なくなります。たとえば、企業対顧客 Web ポータルでは、ポータルデスクトップの初期表示がロードされるとすぐ、大勢のログインユーザーが外部の Web サイトにリダイレクトされます。しかし、さらに多くのユーザーがログインすると、Portal Server に要求を送信しているユーザーと単にログインしているだけのユーザーの割合が低くても、より多くのメモリー領域が必要になります。

  日、週、または月の特定の時間帯におけるユーザーの動作に応じて、Portal Server は、CPU 集中のワークロードとメモリー集中のワークロードを切り替えることができます。ポータルのサイト管理者は、最重要なワークロード条件を定めて、企業のビジネス目標に合ったサイトのサイジングとチューニングを実行する必要があります。

基準サイズのカスタマイズ

Portal Server を配備する場合の適切な見積もりサイズを設定する作業は反復プロセスです。入力値を変更すれば、幅のあるサイジング結果を生成できます。Portal Server の配備方法をカスタマイズするとパフォーマンスが大きく変化します。

サイジングの見積もりが完了したら、次の点を考慮します。

- LDAP トランザクション数
- アプリケーションサーバーの要件

LDAP トランザクション数

以下の、工場出荷状態でポータルを配備する場合の LDAP トランザクション数を使用して、LDAP マスターとレプリカのサービス要求に与える影響を理解してください。これらの数字は、システムのカスタマイズを開始すると変更されます。

- 認証なしの匿名ポータルへのアクセス - 0 ops
- ログインチャネルを使用したログイン - 2 BINDS、2 SRCH
- ポータルデスクトップからのチャネルの削除 - 8 SRCH、2 MOD
- ポータルデスクトップの再ロード - 0 ops

アプリケーションサーバーの要件

アプリケーションサーバーにインストールされた Portal Server の主な使用法の 1 つは、ポータルプロバイダを、アプリケーションサーバーで動作している Enterprise JavaBeans™ アーキテクチャー、および JDBC や JCA などの他の J2EE™ テクノロジスタック構造体と統合することです。これら他のアプリケーションとモジュールはリソースを消費するので、ポータルのサイジングに影響があります。

基準サイズの検証

ここまでで、ポータルを配備する場合の CPU の見積もり数が算出されるので、試験的な配備を行なってポータルのパフォーマンスを測定します。ロードバランス機能を使用して、ストレステストを実行して、次の要素を決定します。

- スループット : 指定された時間内に処理されるデータ量です。
- 待ち時間 : 1 つのコンポーネントが別のコンポーネントを待つときの時間です。
- 最大並行セッション数。

Portal Server にはポータルのサンプルが用意されています。使用するチャネルに類似のチャネルでサンプル使用し、システムに負荷をかけることができます。サンプルはポータルデスクトップにあります。

試験的な配備を使用して、最終的なサイジング見積もりを決定します。試験的な配備により、バックエンド統合をサイジングし、Portal Server の動作に関係する潜在的なボトルネックを回避します。

基準サイズの微調整

次のステップでは、導き出したサイズを微調整します。このセクションでは、スケーラビリティー、高可用性、信頼性、および優れたパフォーマンスを特長とするポータルサイトを配備できるように、適切な量の余裕値を組み込みます。

注	SRA を使用してセキュリティー保護するポータル配備の基準サイジング要件については、74 ページの「SRA サイジング」で説明されています。

基準サイズは多くの見積もり値に基づいて導き出されるので、微調整してから使用してください。

基準サイズを微調整する場合は、次の指示に従います。

- 見積もられた基準サイズを基準点として使用します。

- 基準サイズからの変動量を予想します。
- 他の人の経験から学びます。
- 自分の判断力と知識を活用します。
- 配備における他の要素について検討します。

  Portal Server の配備に、いくつかの大陸におよぶ複数のデータセンターと一様なトラフィックが関係している場合は、1 つの大陸でトラフィックの大きな 2 つの単独データセンターを使用する場合よりも、最終サイズを大きくする必要があります。

- 変更計画

  ポータルサイトは、運用を開始してからさまざまな変更が加えられることがよくあります。たとえば、次のような変更が加えられることがあります。

  - チャネル数の増加
  - ユーザーベースの増大
  - ポータルサイトの目的の変更
  - セキュリティーの必要事項の変更
  - 電源障害
  - メンテナンス要求

これらの要素について考慮すれば、柔軟性に富んだ見積もりサイズを導き出すことができ、ポータルの想定内容が配備後に変更される場合のリスクを回避できます。

導き出される見積もりサイズにより、ポータルサイトの次の特長を確保することができます。

- スケーラビリティー、高可用性、信頼性、および優れたパフォーマンス
- 望むサービスをすべて提供する余裕
- 変更に対応する柔軟性

最終基準サイズの検証

試験的な配備を使用して、ポータルの配備方法がビジネス要件と技術要件を満たすことを検証します。

SRA サイジング

このセクションは、組織において、SRA をインストールして安全性の高いポータルを実装する場合にのみお読みください。ポータルの場合に行なったのと同様、SRA の場合にも、最初にゲートウェイインスタンスの基準見積もりサイズを設定する必要があります。1 台のマシンにインストールできるゲートウェイは 1 つだけですが、複数のインスタンスを持つことはできます。SRA により、それぞれが複数のインスタンスで動作する複数のゲートウェイをインストールできます。設計上の決定事項は、SRA のユーザーセッションと並行性に関して正確な試算を行うのに役立ちます。

まず、ゲートウェイインスタンスの基準サイジング見積もりを設定する必要があります。この基準値は、ゲートウェイのユーザーセッションと並行処理の必要に応じて満たす必要のある条件を示しています。

SRA を配備する場合の適切なサイジング試算値を決定する作業は、反復プロセスです。入力値を変更すれば、幅のあるサイジング結果を生成できます。これらの結果をユーザーの本来の要件に照らしてテストします。要件を正しく定義し、SRA のパフォーマンスとして実際的な値に設定すれば、パフォーマンスが関係するほとんどの問題を回避できます。

このセクションでは、ゲートウェイインスタンスの基準サイジングプロセスが関係する次のタイプのパフォーマンス要素について説明します。

- ゲートウェイの主要なパフォーマンス要件の特定
- ゲートウェイの詳細設定

ゲートウェイの主要なパフォーマンス要件の特定

主要なパフォーマンス要素とは、技術担当者が自動サイジングツールへの入力値として使用するメトリクスのことです。サイジングツールは、SRA を配備する際に必要となるゲートウェイインスタンスの見積もり数を算出します。

これらの主要なパフォーマンス要素を特定し、それを技術担当者に伝えることは、基準サイズを決定するための第一歩です。

注　ゲートウェイを適正にサイジングする作業は複雑なので、ゲートウェイのサイジングツールを使用する段階は単なる始まりにすぎません。ゲートウェイのパフォーマンスはスループットに大きく依存し、それ以外にもユーザー数、アクティブユーザー数、またはユーザーセッション数が影響します。ゲートウェイのサイジング情報は、前提セットに基づいている必要があります。詳細については、152 ページの「Secure Remote Access Example」を参照してください。

主要なパフォーマンス要素は次のとおりです。

- セッション特性
- Netlet の使用特性

注 これらの主要なパフォーマンス要素を計算し終えたら、技術担当者に計算値を伝えます。ゲートウェイのサイジングツールを実行して、ゲートウェイインスタンスの見積もり数を導き出すように依頼します。

セッション特性

ゲートウェイのセッション特性は次のとおりです。

- SRA (ゲートウェイ) ユーザーの合計数

  セキュリティー保護ポータルの見込みユーザーのユーザーベースまたはプールのサイズを表します。この数字の試算方法の詳細については、139 ページの「Concurrent Sessions」を参照してください。

- 最大負荷時にゲートウェイを使用する合計ユーザーの予想割合

  合計ユーザー数にパーセンテージを当てはめてこの数値を導き出します。

- ページヒット間の平均時間

  ユーザーがポータルサーバーからページを要求する頻度の平均値です。

- セッション平均時間

  この特性により、一定数の並行処理ユーザーの場合に、ゲートウェイが保持する必要のある秒あたりのログイン数が決まります。

Netlet の使用特性

以下に示すゲートウェイの Netlet 特性について考慮してください。これらの特性は、ゲートウェイインスタンス数の計算に影響を与えます。

- Netlet は、Access Manager 管理コンソールで有効にします。

  Netlet を有効にした場合は、ゲートウェイは着信トラフィックが Netlet トラフィックであるか、または Portal Server トラフィックであるかを判断する必要があります。Netlet を無効にした場合は、ゲートウェイはすべての着信トラフィックが HTTP トラフィックと HTTPS トラフィックのいずれかであると仮定するため、オーバーヘッドが低減します。Netlet は、Portal Server でリモートアプリケーションをまったく使用しないことが確実な場合にだけ無効にしてください。

- Netlet を使用する合計ユーザーの予想パーセンテージ

  合計ユーザー数にパーセンテージを当てはめてこの数値を導き出します。

- 予想スループット

  ゲートウェイで予想されるスループットを決定し、kbps 単位で表します。

- 使用される Netlet 暗号化方式

  Native VM や Java ソフトウェアプラグイン暗号化方式を選択できます。

ゲートウェイの詳細設定

Portal Server の配備で使用するゲートウェイインスタンスの数を見積もる際に、このセクションの設定を使用して、より正確な数値を得ることができます。これらの詳細なゲートウェイの設定値は、自動サイジングツールの入力値として使用します。

ゲートウェイの詳細設定項目は次のとおりです。

- ページ設定
- スケーラビリティー
- セキュリティー保護されたポータルパイロット測定数

注	CPU の見積もり数に対する計算結果を技術担当者から得たら、これらの関連するパフォーマンス要素がその計算結果に与える影響について考察します。

ページ設定

認証されたポータルを使用する場合は、自動サイジングツールのページ設定セクションにある「Login Type」と「Desktop Type」の両方を指定する必要があります。

- Login Type: エンドユーザーがユーザー名とパスワードを入力したときに、エンドユーザーに最初に表示されるポータルページ (コンテンツ設定と配信方法) のタイプを記述します。このプロセスでは、クレデンシャルの確認、セッションの初期化、および初期コンテンツの配信などが行われるので、システムにかかる負担が大きくなるのが一般的です。

  ログインタイプに関連する測定された CPU パフォーマンスの特性は、*Initial Desktop Display* 変数です。

- Desktop Type: 最初のポータルページのあとにエンドユーザーに表示されるポータルページ (コンテンツ設定と配信方法) のタイプを記述します。これらのページは、そのあとに続くポータルとの対話型操作のたびに、またはデスクトップを更新したときに表示されます。セッションがすでに確立され、キャッシュされたコンテンツを利用できるので、通常必要とするシステムリソースが少なくて済み、ページがより短時間で配信されます。

デスクトップタイプに関連する測定された CPU パフォーマンスの特性は、*Desktop Reload* 変数です。

ログインタイプとデスクトップタイプの両方の場合に、次の適切なコンテンツ設定を選択します。

- Light-JSP: それぞれ 5 個のチャネルを持つ、2 つのタブの設定を記述します。
- Regular-JSP: それぞれ 7 個のチャネルを持つ、2 つのタブの設定を記述します。
- Heavy-JSP: それぞれ 17 個のチャネルを持つ、3 つのタブの設定を記述します。

スケーラビリティー

ゲートウェイインスタンスあたり、1 個、2 個、および 4 個の CPU を選択できます。ゲートウェイインスタンスにバインドされた CPU の数により、配備に必要なゲートウェイインスタンスの数が決まります。

セキュリティー保護されたポータルパイロット測定数

SRA ポータルのパイロットから得られた数値がある場合は、ゲートウェイサイジングツールでそれらの数値を使用して、より正確な結果を導き出すことができます。次の値を入力します。

- 測定された CPU パフォーマンス : ゲートウェイインスタンスの数を計算するために使用される値には、次のようなものがあります。
  - 初期ポータルデスクトップ表示、CPU あたりの毎秒ヒット数
  - ポータルデスクトップ再ロード、CPU あたりの毎秒ヒット数
- Netlet アプリケーションのブロックサイズ : この値により、Netlet アプリケーションのバイトサイズを指定します。Netlet は、使用するアプリケーションに基づいてブロックサイズを動的に決定します。Telnet 用に Netlet によって決定されるブロックサイズは、転送されるデータの量に基づいて決定されます。

注 試験的な配備数を使用する場合は、「Page Configuration」オプションと「Scalability」オプションを指定する必要はありません。

SRA ゲートウェイと SSL ハードウェアアクセラレータ

SRA ゲートウェイのような SSL 集中サーバーは、毎回のセキュリティー保護トランザクションで要求される暗号化を実行するために、大きな処理能力を必要とします。ゲートウェイのハードウェアアクセラレータを使用すれば、暗号化アルゴリズムの実行速度が上がり、動作速度が向上します。

Sun Crypto Accelerator 1000 ボードは、公開鍵と対称暗号化を加速する暗号化コプロセッサとして機能するショート PCI ボードです。この製品には外部インタフェースがありません。ボードは内部 PCI バスインタフェースを通じてホストと対話します。このボードの目的は、電子商取引アプリケーションのセキュリティープロトコルのために、計算を中心とするさまざまな暗号化アルゴリズムを高速化することです。

Sun Crypto Accelerator 1000 ボードと他のアクセラレータの詳細については、『Portal Server Secure Remote Access 6 管理ガイド』を参照してください。

注　Sun Crypto Accelerator 1000 ボードは、SSL ハンドシェークのみをサポートしており、対称鍵アルゴリズムはサポートしていません。これは、他のすべての暗号化アクセラレータと同様ではありません。他の暗号化アクセラレータも市場に出ており、中には対称鍵暗号をサポートするものもあります。詳細については、次の URL にアクセスしてください。

http://www.zeus.com/products/zws/security/hardware.html

Netlet プロキシとリライタプロキシのマシンでハードウェアアクセラレータを使用すれば、パフォーマンスをある程度改善できます。

SRA と Sun Enterprise ミッドフレームライン

通常の本稼働環境では、Portal Server と SRA を別々のマシンに配備します。ただし、複数のハードウェアドメインをサポートする Sun Enterprise™ ミッドフレームマシンの場合は、同じ Sun Enterprise ミッドフレームマシンの異なるドメインに Portal Server と SRA の両方をインストールできます。Portal Server と SRA に関係する通常の CPU とメモリーの要件は引き続き適用されるので、それぞれの要件を別々のドメインに実装します。

このタイプの設定では、セキュリティーの問題に注意します。たとえば、ほとんどの場合、Portal Server ドメインはイントラネットに配置され、SRA ドメインは DMZ に配置されます。

第 5 章

ポータルの設計

この章では、ポータルの高レベルおよび低レベルの設計を行う方法について説明し、設計計画の各部分の設計に必要な情報を提供します。

この章で説明する内容は次のとおりです。

ポータルの設計への取り組み方

Sun Java™ System Portal Server の配備のこの時点で、業務および技術上の要件を確認し、それらの要件を関係者に伝えて承認を得ているはずです。これで、設計段階を開始できます。設計段階では、高レベルと低レベルの設計を行います。

高レベルのポータルの設計は、システムのアーキテクチャーを明確にし、ソリューションの低レベルの設計の基になります。さらに、高レベルの設計では、事前に確定した業務および技術上の要件を満たす論理アーキテクチャーを記述する必要があります。論理アーキテクチャーは、システム全体を構成するさまざまなアプリケーションに従って、またユーザーがシステムと対話する方法に従って分解されます。一般に、論理アーキテクチャーには、Portal Server Secure Remote Access (SRA)、高可用性、セキュリティー (Access Manager を含む)、および Directory Server アーキテクチャーコンポーネントが含まれます。詳細は、81 ページの「論理ポータルアーキテクチャー」を参照してください。

高レベルおよび低レベルの設計では、ポータルの制御を超える要素、つまりネットワーク、ハードウェアの障害、不適切なチャネルの設計なども考慮する必要があります。

完成した高レベルの設計を基に低レベルの設計を行うことができます。低レベルの設計では、物理アーキテクチャー、ネットワークインフラストラクチャー、ポータルデスクトップのチャネルおよびコンテナの設計、実際のハードウェアおよびソフトウェアのコンポーネントなどのアイテムを指定します。高レベルおよび低レベルの設計を完了すると、組織内で試験的な配備のテストを開始できます。

ポータルの高レベルの設計の概要

高レベルの設計は、業務の要件と技術要件の両方をサポートするアーキテクチャーに取り組む最初の段階です。高レベルの設計では、次のような問いに答えます。

- 提案したアーキテクチャーが業務の要件と技術要件の両方をサポートするか。
- 何らかの変更でこの設計を強化できまるか。
- これを実現する代わりのアーキテクチャーがあるか。
- システムの物理的なレイアウトはどのようなものであるか。
- さまざまなコンポーネントと接続のマッピングはどのようなものであるか。
- ユーザー、システム、およびユーザーがアクセスできるアプリケーションのさまざまな分類を記述する論理定義はどのようなものであるか。
- 時間の経過に伴う Web トラフィックの増加によって必要になるシステムへのハードウェアの追加を考慮した設計になっているか。

ポータルの低レベルの設計の概要

低レベルの設計では、ポータルソリューションを構築するために使用するプロセスおよび標準の指定、また次のものを含む実際のハードウェアおよびソフトウェアコンポーネントの指定に焦点を合わせます。

- Portal Server サーバーの複雑性。
- ネットワーク接続 : ポータルコンプレックスを「外の世界」にどのように接続するかを示します。ここでは、セキュリティーの問題、プロトコル、速度、および他のアプリケーションまたはリモートサイトへの接続を考慮する必要があります。
- 情報アーキテクチャー : ユーザーインタフェース、コンテンツのプレゼンテーションおよび構成、データソース、フィードなど。
- Access Manager のアーキテクチャー : 長期にわたる成功に重要な、組織、サブ組織、ロール、グループ、ユーザーなどの方針および設計など。
- 統合方針 : さまざまな情報を統合し、新しい方法でユーザーをまとめるための統合点としてポータルがどのようにふるまうかなど。

論理ポータルアーキテクチャー

論理ポータルアーキテクチャーは、次のものを含む (しかし次のものに限定されない)、ポータルを構成するすべてのコンポーネントを定義します。

- Portal Server そのもの
- RDBM からのコンテンツ
- サードパーティーのコンテンツプロバイダ
- カスタム開発のプロバイダおよびコンテンツ
- メッセージングシステムやカレンダシステムなどのバックエンドシステムとの統合
- 配備のための Web コンテナ
- コンテンツ管理システムのロール
- 顧客リソースの管理
- ポータルがオープンモードまたはセキュアモード (Secure Remote Access が必要) のどちらで動作するか
- 使用の見積もり : これには、登録ユーザーの総数、1 日にログインする登録ユーザーの割合の平均、1 日に同時にログインするユーザーの平均の数、平均ログイン時間、ログインしたユーザーが選択したコンテンツチャネルの平均数、ログインしたユーザーが選択したアプリケーションチャネルの平均の数に対する予測が含まれます。

また、次の 3 つのネットワークゾーンがどのように設計に適合するかを考慮する必要があります。

- **インターネット** : 一般のインターネットとは、イントラネットと DMZ の外側にあるネットワークのことです。ユーザーのポータルサーバーとゲートウェイにここから安全にアクセスします。
- **非武装ゾーン** (DeMilitarized Zone、DMZ): 2 つのファイアウォールの間にある安全な領域であり、無許可の侵入を防止し、内部リソースへのアクセスを可能にします。ゲートウェイは、この場所に存在し、ここからアプリケーションサーバーやコンテンツサーバーからのトラフィックをインターネットへ安全に転送できます。
- **イントラネット** : すべてのリソースサーバーが含まれます。これには、イントラネットアプリケーション、Web コンテンツサーバー、およびアプリケーションサーバーが含まれます。Portal Server および Directory Server はここに存在します。

論理アーキテクチャーは、次のようなものを含む、ポータルデスクトップの見た目と使い心地を記述します。

- デフォルトのページ : デフォルトのバナー、ロゴ、チャネル、ページの重みの合計、つまりページのすべてのコンポーネント (HTML、スタイルシート、JavaScript™、イメージファイルなど) の総バイト数、ページに対する HTTP 要求の総数、つまりそのページをダウンロードするために必要な HTTP 要求の数が表示される。
- パーソナライズされたページ : ユーザーが表示できるチャネルと利用できる設定などが表示される。

サイトが必要とする場合、キャッシングの方針も論理アーキテクチャーに含めます。ユーザーに返されるページに多数のイメージへの参照が含まれる場合、Portal Server がそれらのイメージをすべてのユーザーに配信できます。ただし、それらのタイプの要求を逆プロキシタイプのキャッシング装置にオフロードできる場合、Portal Server が他のユーザーにサービスを提供できるようにシステムリソースを解放できます。また、キャッシング装置をエンドユーザーの近くに配置することによって、それらのイメージをエンドユーザーにいくらか速く配信することができるので、エンドユーザーの使い勝手がよくなります。

Portal Server とスケーラビリティー

スケーラビリティーとは、パフォーマンスを低下させることなく、処理リソースを追加することによって、増加するユーザー人口に対応するシステムの能力のことです。システムをスケーリングするための 2 つの一般的な方法には、垂直方向のスケーリングと水平方向のスケーリングがあります。このセクションの主題は、Portal Server 製品へのスケーリング技術の応用です。

スケーラブルなシステムの利点を次に示します。

- 応答時間の向上
- 障害許容性
- 管理容易性
- 消耗性
- 簡易化されたアプリケーションの開発
- 構築モジュール

垂直方向のスケーリング

垂直方向のスケーリングでは、CPU、メモリー、Portal Server の複数のインスタンスなどのリソースが 1 つのマシンに追加されます。これより、より多くのプロセスインスタンスが同時に実行できます。Portal Server では、必要な CPU の数に計画およびサイズ指定することによってこれを利用できます。詳細は、第 4 章「配備前の注意点」を参照してください。

水平方向のスケーリング

水平方向のスケーリングでは、マシンが追加されます。これは、複数の同時処理とワークロードの分散も可能にします。Portal Server では、Portal Server、Directory Server、および Access Mangaer を異なるノードで実行できるので、水平方向のスケーリングを利用します。水平方向のスケーリングは、さらに CPU を追加するなどして、垂直方向のスケーリングも利用できます。

また、サーバーコンポーネントインスタンスを複数のマシンにインストールすることによって、Portal Server インストールを水平方向にスケールできます。インストールされた各サーバーコンポーネントインスタンスは、HTTP プロセスを実行し、この HTTP プロセスはインストール時に決定された番号の TCP/IP ポートで待機します。

ゲートウェイのコンポーネントは、ラウンドロビンアルゴリズムを使用して新しいセッション要求をサーバーインスタンスに割り当てます。セッションが確立されている間は、クライアントに格納された HTTP cookie がセッションサーバーを示します。それ以降の要求はすべてそのサーバーに送られます。

89 ページの「Portal Server 構築モジュールの使用」のセクションでは、最適のパフォーマンスと水平方向のスケーラビリティーを提供する特定のタイプの構成への取り組み方について説明します。

Portal Server と高可用性

高可用性は、ポータルプラットフォームに週 7 日 24 時間アクセス可能であることを保証します。今日の組織では、データやアプリケーションが常に利用可能であることが要求されます。高可用性は、ミッションクリティカルなアプリケーションのみでなく、IT インフラストラクチャー全体にも要求されます。

システムの可用性は、コンピュータのハードウェアとソフトウェアのみでなく、システムの停止時間の 80 パーセントの原因である人およびプロセスによっても影響を受けます。可用性は、システムの管理を計画的に行うことで、また人為ミスの影響を最小限にする業界の最良の方法によって向上できます。

考慮する必要がある重要な問題の 1 つに、すべてのシステムの可用性要件が同じレベルであるとはかぎらないという点があります。ほとんどのアプリケーションは、次の 3 つのグループに分類できます。

- **タスククリティカル** : かぎられた数のユーザーに影響します。顧客には見えません。コストと利益に少し影響があります。
- **ビジネスクリティカル** : かなりの数のユーザーに影響します。一部の顧客に見えることがあります。コストと利益にかなりの影響があります。
- **ミッションクリティカル** : 多数のユーザーに影響します。顧客には見えます。コストと利益に大きな影響を及ぼします。

これらのレベルの目標は、次の点を向上することです。

- 人為ミスの削減、手順の自動化、および計画的な停止時間の削減による処理の向上
- シングルポイント障害の構成の除去と処理負荷の分散によるハードウェアおよびソフトウェアの可用性の向上

アプリケーションがミッションクリティカルになるほど、シングルポイント障害 (Single Point of Failure、SPOF) をなくし、人およびプロセスの問題を解決するために可用性にさらに焦点を合わせる必要がでてきます。

システムが常に利用可能であっても、障害回復のインスタンスがエンドユーザーに透過でない場合があります。障害の種類によっては、ユーザーはポータルアプリケーションのコンテキストを失うことがあり、ポータルデスクトップにアクセスするためにもう一度ログインする必要がある場合があります。

システムの可用性

システムの可用性は、システムの稼働時間の割合で表現されます。システムの可用性を計算するための基本的な式を次に示します。

```
可用性 = 稼働時間 / ( 稼働時間 + 停止時間 ) * 100
```

たとえば、サービスレベル契約の稼働時間が 4 桁 (99.99 %) である場合は、1 ヶ月にシステムが約 7 時間利用できなくなることを意味します。さらに、システムの停止時間は、システムが利用できない時間の合計です。この合計には、ハードウェアの障害やネットワークの停止などの計画外の停止時間のみでなく、計画された停止時間、予防保守、ソフトウェアのアップグレード、パッチなどの時間も含まれます。

システムを週 7 日間、1 日 24 時間利用できるようにする場合は、高可用性を保証するため、計画された停止時間と計画されていない停止時間を避けるためにアーキテクチャーには冗長性を含める必要があります。

高可用性のレベル

高可用性は、オンまたはオフにできるスイッチであるだけではありません。高可用性のさまざまなレベルは、システムが障害から回復する能力とシステムの可用性を測定する方法も指します。高可用性のレベルは、特定組織の障害の許容性の要件とシステムの可用性の測定方法によって決定されます。

たとえば、別のログイン画面にリダイレクトされた要求が成功したとみなされるように、組織がシステムの障害発生後の再認証の必要を許容する場合があります。他の組織では、システムが引き続きサービスを提供している場合でも、これは失敗とみなされる可能性があります。

特定のポータルアプリケーションのコンテキストはフェイルオーバー後に失われることがあるので、セッションのフェイルオーバーのみが透過なフェイルオーバーの究極の解決策ではありません。たとえば、ユーザーが NetMail Lite でメッセージを作成し、その電子メールにいくつかの文書を添付してから、サーバーに障害が発生した場合を考えてみてください。ユーザーは別のサーバーにリダイレクトされ、NetMail Lite はユーザーのセッションとメッセージの下書きを失います。コンテキストデータを現在の JVM™ に格納するその他のプロバイダでも同じ問題が発生します。

Portal Server の高可用性の実現

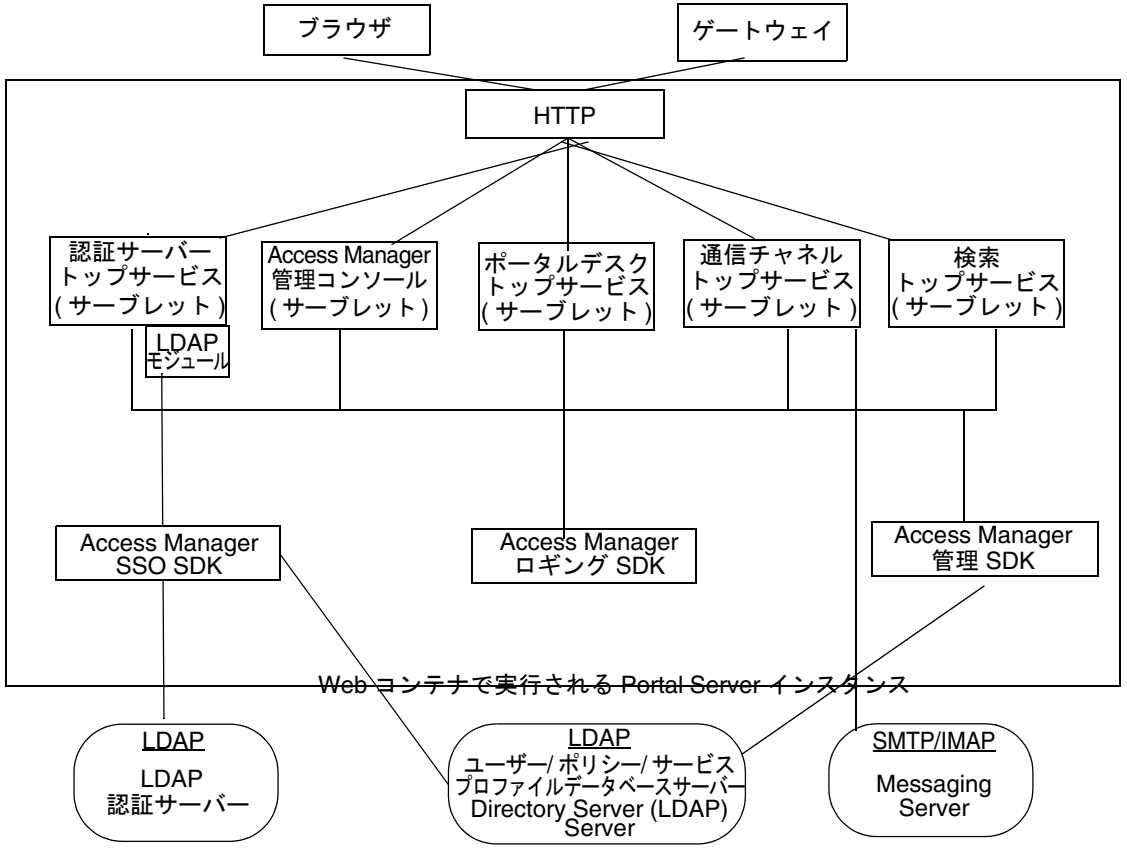
Portal Server の可用性を高めるには、次の各コンポーネントの可用性を高める必要があります。

- **ゲートウェイ** : ゲートウェイで使用するロードバランサは、障害が発生したゲートウェイコンポーネントを検出し、新しい要求を他のゲートウェイにルーティングします。ロードバランサは、ワークロードをサーバープールにインテリジェントに分散する能力もあります。障害が発生したゲートウェイが回復すると、ルーティングが元に戻ります。ゲートウェイコンポーネントはステートレスなので (セッション情報はクライアントで HTTP cookie に格納される)、障害が発生したゲートウェイを迂回した再ルーティングはユーザーには透過です。
- Portal Server: オープンモードでは、ロードバランサを使用して障害が発生したコンポーネントを検出し、要求を他のサーバーにリダイレクトします。セキュアモードでは、ゲートウェイコンポーネントが障害の発生したサーバーコンポーネントの存在を検出し、要求を他のサーバーにリダイレクトします。Web コンテナが Web Server であるかぎりこのようになります。
- Directory Server: 多数のオプションが、LDAP ディレクトリの可用性を高めます。詳細は、90 ページの「構築モジュールと高可用性のシナリオ」を参照してください。
- **Netlet プロキシとリライタプロキシ** : ソフトウェアのクラッシュが発生した場合、watchdog プロセスがプロキシを自動的に再起動します。さらに、ゲートウェイがプロキシのロードバランスと障害検出フェイルオーバーを実行します。

Portal Server システムの通信リンク

87 ページの図 5-1 は、ソリューションの可用性に重要な Portal Server システムのプロセスおよび通信リンクを示しています。

図 5-1　Portal Server の通信リンク



この図では、Web Server 技術で稼働する Portal Server インスタンスがボックスで囲まれています。このインスタンスには、5 つのサーブレット (認証、Access Manager 管理コンソール、ポータルデスクトップ、通信チャネル、および検索) と 3 つの SDK (Access Manager SSO、Access Manager ロギング、および Access Manager 管理) が含まれています。認証サービスサーブレットは、LDAP サービスプロバイダモジュールも利用します。

ユーザーは、ブラウザまたはゲートウェイのどちらかを使用して Portal Server と通信します。このトラフィックは、適切なサーブレットにダイレクトされます。通信は、認証サービスの LDAP モジュールと LDAP 認証サーバー間、通信チャネルサーブレットと SMTP/IMAP メッセージングサーバー間、Access Manager SSO SDK と LDAP サーバー間、Access Manager 管理 SDK と LDAP サーバー間で行われます。

87 ページの図 5-1 は、次のプロセスまたは通信リンクに障害が発生すると、エンドユーザーがポータルソリューションを利用できなくなることを示しています。

- **Portal Server インスタンス** : Web コンテナのコンテキストで実行されます。インスタンスに含まれるコンポーネントは、Java™ API を使用して JVM™ を介して通信します。インスタンスは、完全修飾ドメイン名と TCP ポート番号です。Portal Server サービスは、サーブレットまたは JSP™ ファイルとして実装される Web アプリケーションです。

  Portal Server は、認証シングルサイオン (セッション) 管理、ポリシー、およびプロファイルデータベースアクセスのために Access Manager 上に構築されます。したがって、Portal Server は可用性と障害許容性に関して Access Manager のすべての利点 (および制約) を継承します。

  設計により、Access Manager のサービスは、ステートレス、またはサービスがコンテキストデータを共有できるのどちらかになります。サービスは、サービスに障害が発生した場合に前の状態に戻ることができます。

  Portal Server では、ポータルデスクトップサービスと NetMail サービスはインスタンス間で状態データを共有しません。これは、インスタンスのリダイレクトによって、有効になったサービスに対してユーザーコンテキストが再作成されることを意味します。通常、リダイレクトされたユーザーはこれに気がつきません。これは、Portal Server サービスがユーザーコンテキストをユーザーのプロファイルから、また要求に格納されたコンテキストデータを使用することによって再作成できるためです。これは一般にインストール後即使用可能なサービスに当てはまりますが、チャネルやカスタムコードには当てはまらないことがあります。開発者は、インスタンスのフェイルオーバー時にコンテキストを失うのを避けるためにステートフルチャネルを設計しないように注意する必要があります。

- **プロファイルデータベースサーバー** : プロファイルデータベースサーバーは、Directory Server ソフトウェアによって実装されます。このサーバーは厳密には Portal Server の一部ではありませんが、このサーバーの可用性とデータベースの完全性はシステムの可用性に欠かせません。

- **認証サーバー** : これは LDAP 認証のためのディレクトリサーバーです (通常、プロファイルデータベースサーバーと同じサーバー)。このサーバーには、プロファイルデータベースサーバーと同じ高可用性技術を適用できます。

- **SRA ゲートウェイとプロキシ** : SRA ゲートウェイは、状態情報をエンドユーザーに透過に再作成できるため、ステートレスとみなすことのできるスタンドアロンの Java テクノロジプロセスです。ゲートウェイプロファイルは、Portal Server インスタンスのリストを維持し、ゲートウェイインスタンス間でラウンドロビン方式のロードバランスを行います。ゲートウェイの前ではセッション固定の必要はありませんが、セッションが固定されると、パフォーマンスが向上します。一方、Portal Server インスタンスに対するセッション固定は SRA によって実施されます。

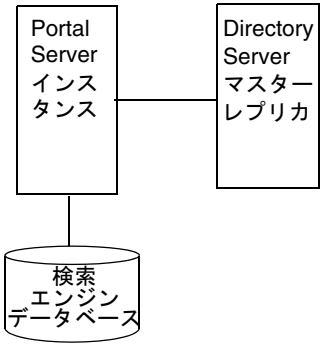
SRA には、Netlet プロキシとリライタプロキシと呼ばれる他の Java テクノロジプロセスも含まれます。これらのプロキシを使用して、ファイアウォールの背後からセキュリティーの適用範囲を拡張し、DMZ の穴の数を制限できます。これらのプロキシは、別々のノードにインストールできます。

Portal Server 構築モジュールの使用

Portal Server の配備は多くの他のシステムも関係する複雑な処理であるため、ここでは最高のパフォーマンスと水平方向のスケーラビリティーを提供する特定の構成について説明します。この構成は、Portal Server 構築モジュールと呼ばれます。

Portal Server 構築モジュールは、共有サービスに限定的に依存、またはまったく依存しないハードウェアおよびソフトウェアの構成です。一般的な配備では、複数の構築モジュールを使用して、最高のパフォーマンスと水平方向のスケーラビリティーを実現します。図 5-2 は、構築モジュールのアーキテクチャーを示しています。

図 5-2 Portal Server 構築モジュールのアーキテクチャー



注 Portal Server 構築モジュールは、単なる推奨構成です。場合によっては、別の構成のほうがスループットが若干よくなることがあります。この場合、一般的に構成はより複雑になります。たとえば、4 CPU システムに Portal Server の別のインスタンスを追加すると、スループットが最高 10 % 向上する可能性がありますが、単一システムのみを使用する場合でもロードバランサを追加する必要があるという代償を払う必要があります。

構築モジュールと高可用性のシナリオ

Portal Server は、高可用性に関して次の 3 つのシナリオを提供します。

- ベストエフォート

  ハードウェアに障害が発生しないかぎり、また watchdog プロセスによって Portal Server プロセスを再起動できるかぎり、システムを利用できます。

- ノーシングルポイント障害

  ハードウェアとソフトウェアのレプリケーションにより、ノーシングルポイント障害 (No Single Point of Failure、NSPOF) の配備を構築します。コンポーネントの連鎖のどこかで連続的に複数の障害が発生しないかぎり、システムは常に利用可能です。ただし、障害が発生した場合は、ユーザーセッションは失われます。

- 透過フェイルオーバー

  システムは常に利用可能ですが、NSPOF に加えて、バックアップインスタンスへのフェイルオーバーがエンドユーザーに透過に行われます。ほとんどの場合、ユーザーは別のノードまたはインスタンスにリダイレクトされたことに気がつきません。セッションは、ノード間にわたって維持されるので、ユーザーは再認証する必要がありません。Portal Server サービスは、ステートレス、またはチェックポイントメカニズムを使用して現在の実行コンテキストを特定の時点まで再構築します。

サポート可能なアーキテクチャーを次に示します。

- Sun Cluster エージェントをサポートするコンポーネントで Sun™ Cluster ソフトウェアの使用
- マルチマスター Directory Server 技術

ここでは、高可用性の観点から、これらのアーキテクチャーの実装方法について、また構築モジュール概念を活用する方法について説明します。

表 5-1 は、これらの高可用性シナリオと、シナリオをサポートする技術の要約です。

表 5-1 Portal Server 高可用性シナリオ

コンポーネントの要件	ベストエフォート配備に必要ですか？	NSPOF 配備に必要ですか？	透過フェイルオーバー配備に必要ですか？
冗長ハードウェア	はい	はい	はい
Portal Server の構築モジュール	いいえ	はい	はい
マルチマスター構成	いいえ	はい	はい
ロードバランス	はい	はい	はい
ステートレスなアプリケーションとチェックポイントメカニズム	いいえ	いいえ	はい

表 5-1　Portal Server 高可用性シナリオ (続き)

コンポーネントの要件	ベストエフォート配備に必要ですか？	NSPOF 配備に必要ですか？	透過フェイルオーバー配備に必要ですか？
セッションのフェイルオーバー	いいえ	いいえ	はい
Directory Server クラスタ	いいえ	いいえ	はい

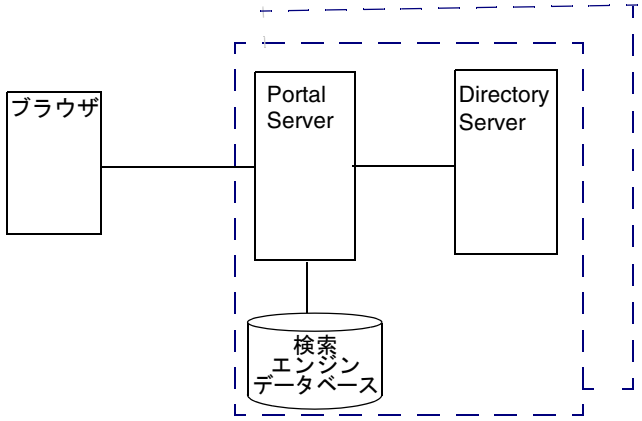
注　ロードバランスは、Web Server 製品では出荷時の状態では提供されていません。

ベストエフォート

このシナリオでは、可用性が続くように、Sun Fire UltraSPARC® III マシンなどの、ハードウェアが保護された単一ノードに Portal Server と Directory Server をインストールします。Solaris™ オペレーティング環境システムを保護するには、デフォルトの設定を変更する必要があります。

このタイプのサーバーではハードウェアが完全に冗長であり、次のものを備えています。冗長電源、冗長ファン、冗長システムコントローラ、動的再構成、CPU ホットプラグ、オンラインアップグレード、および RAID 0+1 (ストラインピングにミラーリングもプラス) またはボリューム管理システムを使用する RAID 5 で構成できるディスクラック (ディスククラッシュ発生時にデータが失われるのを防止する)。図 5-3 は、構築モジュールのアーキテクチャーを使用する、小規模のベストエフォート配備を示しています。

図 5-3　ベストエフォートのシナリオ



このシナリオでは、1 つの構築モジュールには、メモリーの割り当ては 4 CPU × 8G バイト RAM (4 × 8) で十分です。Access Manager コンソールは、他のリソースと共有できるように構築モジュールの外にあります。実際のサイズの計算結果は、これとは異なる割り当て量になる場合があります。

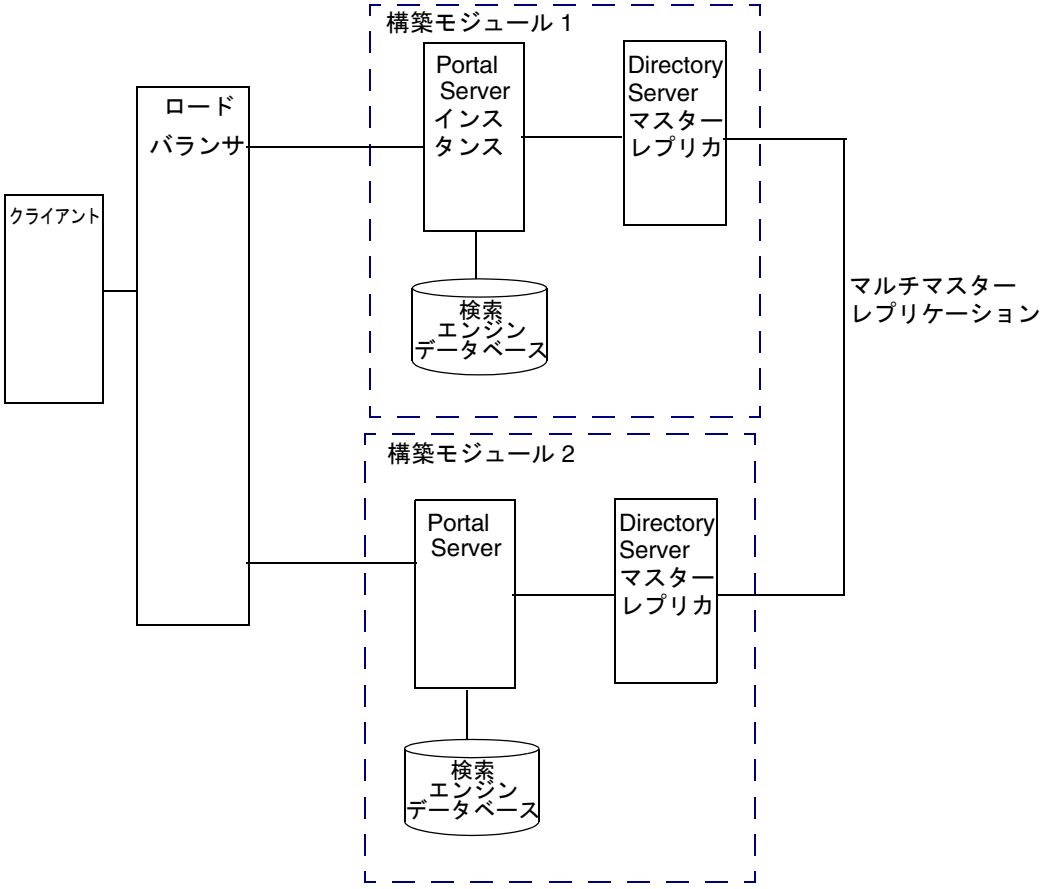
このシナリオは、タスククリティカルな要件には十分です。このシナリオの主な弱点は、システムのシャットダウンが必要な保守作業によってサービスが中断されるということです。

SRA を使用している場合に、ソフトウェアのクラッシュが発生すると、watchdog プロセスがゲートウェイ、Netlet プロキシ、およびリライタプロキシを自動的に再起動します。

ノーシングルポイント障害

Portal Server は、ノーシングルポイント障害 (NSPOF) シナリオを基本機能としてサポートします。NSPOF は、ベストエフォートシナリオをベースにし、それに加えてレプリケーションとロードバランスを採り入れています。

図 5-4 ノーシングルポイント障害の例



前述したとおり、構築モジュールは、Portal Server インスタンス、プロファイルの読み込みのための Directory Server マスターレプリカ、および検索エンジンのデータベースから構成されています。そのため、NSPOF を実現するには少なくとも 2 つの構築モジュールが必要であり、それによって構築モジュールのどちらかに障害が発生した場合のバックアップを提供します。これらの構築モジュールは、4 CPU × 8G バイト RAM で構成されます。

ロードバランサが Portal Server の障害を検出すると、ユーザーの要求をバックアップ構築モジュールにリダイレクトします。障害検出の正確さは、ロードバランス製品によって異なります。一部の製品は、サーバーのいくつかの機能領域に関係するサービス、たとえばサーブレットエンジンや JVM を検索することによってシステムの可用性を確認できます。特に、Resonate、Cisco、Alteon、およびその他のほとんどのベンダーソリューションを使用する場合は、ユーザーがサーバーの可用性のためのスクリプトを任意に作成できます。ロードバランサは Portal Server ソフトウェアの一部ではないので、サードパーティーベンダーから個別に入手する必要があります。

注 Access Manager 製品は、セッション固定を実施するためにロードバランスを設定することを要求します。これは、特定のインスタンスに対するセッションを確立すると、ロードバランサはそのセッションのために常に同じインスタンスに戻る必要があります。ロードバランサは、セッション cookie にインスタンスの識別名をバインドすることによってこれを実現します。原則としては、障害が発生したインスタンスを終了したときに、そのバインドは再設定されます。セッション固定は、パフォーマンス上の理由からも推奨します。

マルチマスターレプリケーション (Multi-master replication、MMR) は、構築モジュール間で行われます。各ディレクトリで発生する変更は他のディレクトリにレプリケートされます。つまり、各ディレクトリがサプライヤとコンシューマの両方の役割を果たします。MMR については、『Directory Server 6 Deployment Guide』を参照してください。

注 一般に、各構築モジュール内の Directory Server インスタンスは、他の場所で実行されるマスターディレクトリのレプリカとして構成されます。ただし、マスターディレクトリを構築モジュールの一部として使用するのを妨げるものはありません。専用ノードでマスターを使用しても、ソリューションの可用性は向上しません。専用マスターは、パフォーマンスのために使用します。

構築モジュール間でのコンシューマレプリケーションを伴う、管理コンソールまたはポータルデスクトップを使用したプロファイルの変更を常に維持できるように、冗長性もディレクトリマスターにとって同じように重要です。Portal Server と Access Manager は、MMR をサポートします。NSPOF シナリオは、マルチマスター構成を使用します。この構成では、2 つのサプライヤが更新を受け入れること、互いに同期をとること、またすべてのコンシューマを更新することが可能です。コンシューマは、更新要求を両方のマスターに任せることができます。

SRA は、NSPOF を実現するために Portal Server と同様にレプリケーションとロードバランスを採用します。そのため、このシナリオでは 2 つの SRA ゲートウェイとプロキシのペアが必要になります。SRA ゲートウェイは、特定のタイムアウト値後、要求に対して Portal Server インスタンスが応答しない場合に、Portal Server インスタンスの障害を検出します。これが発生すると、HTTPS 要求はバックアップサーバーにルーティングされます。SRA ゲートウェイは、最初の Portal Server インスタンスが再び稼働するまで定期的に可用性を確認します。

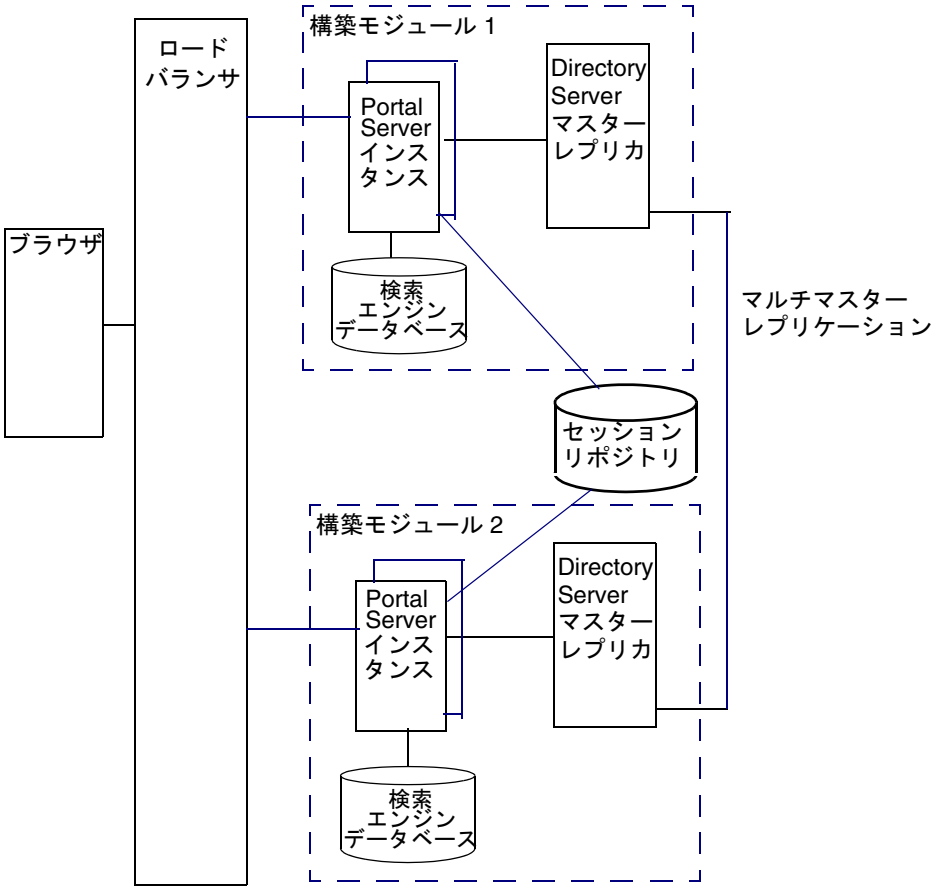
NSPOF の高可用性シナリオは、ビジネスクリティカルな配備に適しています。しかし、このシナリオの高可用性の制限の一部は、ミッションクリティカルな配備の要件を満たさない場合があります。

透過フェイルオーバー

透過フェイルオーバーは、NSPOF シナリオと同じレプリケーションモデルを使用しますが、追加の高可用性機能があり、これによってエンドユーザーに透過なバックアップサーバーにフェイルオーバーが行われます。

96 ページの図 5-5 は、透過フェイルオーバーのシナリオを示しています。4 CPU × 8G バイト RAM から構成される 2 つの構築モジュールを示しています。ロードバランスは、Portal Server の障害を検出し、構築モジュール内のバックアップ Portal Server に要求をリダイレクトする責任があります。構築モジュール 1 は、セッションをセッションリポジトリに格納します。クラッシュが発生した場合、アプリケーションサーバーは構築モジュール 1 が作成したセッションをセッションリポジトリから取得します。

図 5-5　透過フェイルオーバーの例のシナリオ



セッションリポジトリは、アプリケーションサーバーソフトウェアで提供されます。Portal Server は、アプリケーションサーバーで稼働します。Portal Server は、HttpSession フェイルオーバーをサポートするアプリケーションサーバーで透過フェイルオーバーをサポートします。詳細は、付録 C「Portal Server とアプリケーションサーバー」を参照してください。

セッションフェイルオーバーを使用すると、クラッシュ発生後にユーザーを再認証する必要はありません。また、ポータルアプリケーションは、セッションが固定されるという前提で、チェックポイントで使用するコンテキストデータを保存できます。`AMConfig.properties` ファイルで `com.iplanet.am.session.failover.enabled property` を **`true`** に設定することによって、セッションのフェイルオーバーを設定できます。

Netlet プロキシは、TCP プロトコルの制限により透過フェイルオーバーシナリオをサポートできません。Netlet プロキシは、TCP 接続をトンネルし、オープンな TCP 接続を別のサーバーに移すことはできません。Netlet プロキシのクラッシュによって、再確立する必要があるすべての未処理の接続がなくなります。

構築モジュールの制約

構築モジュールのスケーラビリティーの制約は、プロファイルの更新によって生じる LDAP の書き込みの数と LDAP データベースの最大サイズによります。詳細は、98 ページの「Directory Server の要件」を参照してください。

注	`/tmp` ディレクトリに `_db` ファイルがある場合に LDAP サーバーがクラッシュすると、サーバーが再起動するときにこのファイルは失われます。これはパフォーマンスを向上させますが、可用性にも影響します。

特定のサイトの分析結果が、LDAP の書き込み操作が実際に制約となっていることを示す場合は、この問題の解決策として、受信要求をポータルの適切なインスタンスに転送するディレクトリの特定の分岐とその前にある層のみをレプリケートする構築モジュールを作成することができます。

構築モジュールソリューションの配備

ここでは、構築モジュールソリューションを配備するためのガイドラインを説明します。

配備のガイドライン

構築モジュールの構築方法によってはパフォーマンスに影響します。構築モジュールを適切に配備するためには、次に示す推奨事項を考慮してください。

- 1 台のマシンに 1 つの構築モジュールを配備します。
- 複数のマシンを使用する場合、または Portal Server マシンが多数のインスタンスを稼働させている場合は、高速ネットワークインターコネクトを使用します。

- 8 個を超える CPU が搭載されているサーバーでは、2 個または 4 個の CPU からなるプロセッサセットまたはドメインを作成します。たとえば、Portal Server の 2 つのインスタンスを 8 個の CPU が搭載されたサーバーにインストールする場合は、4 個の CPU からなる 2 つのプロセッサセットを作成します。

Directory Server の要件

構築モジュールを配備するための Directory Server の要件を確認します。Directory Server の配備の特定の情報は、『Directory Server Deployment Guide』を参照してください。

Portal Server の配備を計画する際には、次に示す Directory Server のガイドラインを考慮してください。

- Directory Server コンシューマレプリカプロセッサセットに必要な CPU の量は、構築モジュールに含まれる Portal Server インスタンスの数、またパフォーマンスおよび容量の考慮事項によって決定されます。
- 可能であれば、1 つの Directory Server インスタンスを 1 つの構築モジュール内の Portal Server インスタンス専用にします。89 ページの図 5-2 を参照してください。
- ディレクトリデータベースインデックス全体とメモリー内のキャッシュをマップして、ディスクの遅延に関する問題を防止します。
- 複数の構築モジュールを配備するときは、Directory Server サプライヤに対するプロファイルの更新とレプリケーションのオーバーヘッドによる障害を回避するためにマルチマスター構成を使用します。

検索エンジンの構造

検索エンジンを構築モジュールソリューションの一環として配備するときには、次の点を考慮してください。

- 各構築モジュールでは、1 つの Portal Server インスタンスだけの検索エンジンデータベースに RD が含まれているようにします。残りの Portal Server インスタンスの検索エンジンデータベースは、デフォルトの空の状態であるようにします。
- ポータルの検索データベースに構築モジュールを使用するかどうかに影響する要素には、同時並行検索の数に加えて、Portal Server 配備の検索活動の程度、検索のヒットの範囲、すべてのユーザーの検索ヒットの平均数があります。たとえば、検索エンジンによるサーバーへの負荷は、大きなインデックスや高負荷のクエリーの場合はメモリーと CPU の両方をかなり使用することがあります。
- 検索機能を Portal Server とは別のマシンにインストールして、主なサーバーをポータルの活動専用にすることができます。そのようにする場合、検索プロバイダの `searchURL` プロパティーを使用して、検索機能がインストールされた 2 番目のマシンを指すようにします。検索インスタンスは、通常のポータルインスタンスです。ポータルインスタンスをインストールするのと同様に検索インスタンスをインストールしますが、検索インスタンスは検索機能のみに使用します。

- 検索データベースのサイズによって、複数のマシンにまたは構築モジュールに検索データベースをレプリケートすることで複数のマシンが検索データベースをホストする必要があるかどうかが決まります。ハイエンドのディスクアレイを使用することを考慮します。
- プロキシサーバーを使用して、検索ヒットの結果をキャッシュします。そのようにする場合、ドキュメントレベルのセキュリティーを無効にする必要があります。ドキュメントレベルのセキュリティーについては、『Portal Server 6 管理ガイド』を参照してください。

Portal の使用事例のシナリオの設計

使用事例のシナリオは、システムの能力をテストして示し、高レベルの設計の重要な部分を形成するために記述されたシナリオです。使用事例シナリオはプロジェクトの終わりの方で実現しますが、要件が定まったら、プロジェクトの早い段階でまとめておきます。

利用できる場合、使用事例はシステムをどのようにテストすべきかを判断する際に役立ちます。使用事例は、ユーザーインタフェースをどのように設計するかを、ナビゲーションの観点から決定する際に役立ちます。使用事例を設計する際には、使用事例を要件と比較して、使用事例の完成度、またテスト結果の解釈方法を判断します。

使用事例は、要件をまとめる手段にもなります。要件の一覧の代わりに、ユーザーがシステムをどのように使用できるかを説明するストーリーのように要件をまとめます。これにより完成度と一貫性が向上し、またユーザーの観点から見た要件の重要性について、さらに理解を深めることができます。

使用事例は、ポータルの機能要件を特定および明確にするために役立ちます。使用事例は、ユーザーとポータル間のやり取りのセット、またポータルが実行する必要があるサービス、タスク、および機能など、ポータルのさまざまな使い方をすべて網羅します。

使用事例は、外部のアクターとポータルシステム間の目的のあるやり取りのセットを定義します。アクターは、システムとやり取りするシステム外に存在するパーティーであり、ユーザーのクラス、ユーザーが担うことのできるロール、またはその他のシステムになります。

使用事例は、対象領域の用語を使用した、理解しやすい構造化された物語として記述されます。

使用事例のシナリオは、使用事例の 1 つの例であり、使用事例の 1 つの筋道を表します。したがって、使用事例の主な筋に対するシナリオ、また使用事例の起こりうるさまざまな筋 (たとえば、各オプションを表す) に対するシナリオを作成できます。

ポータルの使用事例の要素

ポータルの使用事例を開発する場合は、次に示す要素に注意してください。

- **優先順位** : 使用例の優先順位、または順位を記述します。たとえば、これは「高」、「中」、「低」の範囲にすることができます。
- **使用の背景** : 使用事例を実現する設定または環境を記述します。
- **範囲** : 使用事例の条件および制限を記述します。
- **プライマリユーザー** : これがあてはまるユーザーの種類、たとえば、エンドユーザーまたは管理者を記述します。
- **特別な要件** : 適用されるその他の条件を記述します。
- **関係者** : 製品の決定がどのように行われるか、または実行されるかに「利害関係」がある人々を記述します。
- **前提条件** : 使用事例を実現するために満たす必要のある必要条件を記述します。
- **最小限の保証** : 使用事例が成功しなかった場合に最低限行う必要があることを記述します。
- **成功の保証** : 使用例が成功した場合に何が起きるかを記述します。
- **トリガー** : イベントの発生の原因になる、システム内の特定のアイテムを記述します。
- **説明** : 使用事例の、始めから終わりまでの、段階的な記述です。

使用事例の例 : ポータルユーザーの認証

表 5-2 では、ポータルのユーザーがポータルに認証される使用事例について説明します。

表 5-2 使用事例 : ポータルユーザーの認証

アイテム	説明
優先順位	必須です。
使用の背景	認証済みのユーザーのみがポータルのリソースにアクセスを許可されます。このアクセス制限は、コンテンツおよびサービスを含む、すべてのポータルのリソースに適用されます。このポータルは、企業の LDAP ディレクトリで管理されているユーザー ID を利用します。

表 5-2　使用事例 : ポータルユーザーの認証 (続き)

アイテム	説明
範囲	ポータルユーザーは、完全なオンラインセッションのために 1 回だけ自分の身元を証明します。アイドルタイムアウトが発生する場合は、ユーザーは自分の身元を再度証明する必要があります。ポータルユーザーの身元証明の失敗回数が指定された許容再試行回数よりも多い場合、システム管理者がアカウントを再び有効にするまで、イントラネットへのアクセスは拒否または制限 (無効) される必要があります。この場合、ポータルのユーザーに、担当者に連絡するように勧める必要があります。身元が確認されたポータルユーザーは、許可されたデータおよび情報にだけアクセスできます。
プライマリユーザー	ポータルエンドユーザー。
特別な要件	なし。
関係者	ポータルエンドユーザー。
前提条件	ポータルユーザーは、承認されたユーザーです。 標準的な企業 LDAP ユーザー ID。 各従業員に提供する必要があります。 承認された LDAP エントリ。 すべての従業員が企業イントラネットにアクセスできます。 ゲストアカウントなし。
最小限の保証	顧客主体の親切なメッセージ。 ステータス - 誰に連絡するかを示すエラーメッセージ付き。
成功の保証	ポータルデスクトップのホームページを表示します。 認証。 権利の付与。 個人情報。
トリガー	ポータルページがアクセスされときに、ユーザーがまだログインしていない場合。
説明	1. ユーザーがポータル URL を入力します。 2. カスタマイズパラメータ [remember login] を設定した場合、ユーザーを自動的にログインさせ、セッション ID を提供します。 3. 初めてのユーザーの場合、LDAP ユーザー ID とパスワードの入力を要求します。 4. ユーザーは、事前に割り当てられたユーザー ID とパスワードを入力します。 5. 情報は検証のために Access Manager に渡されます。 6. 認証に成功した場合、セッション ID を割り当て、続行します。 7. 認証に失敗した場合、エラーメッセージを表示してユーザーをログインページに戻し、残りの試行回数を減分します。事前に設定された試行回数の制限を超えた場合、ユーザーに通知してアカウントをロックアウトします。

ポータルセキュリティーの設計方針

セキュリティーとは、サーバーおよびそのユーザーを悪意のある外部の者から保護するハードウェア、ソフトウェア、運用方法、および技術の集合のことです。それに関連して、セキュリティーは予期しない行為から保護します。

セキュリティーには、グローバルに対処し、ユーザーやプロセスだけでなく製品や技術も含める必要があります。あいにく、多くの組織が、唯一のセキュリティー方針としてファイアウォール技術のみに依存しています。それらの組織は、多くの攻撃は外部の者ではなく、従業員によるものであることに気づいていません。したがって、安全なポータル環境を構築するときには他のツールやプロセスを考慮する必要があります。

安全な環境で Portal Server を稼働させるには、Solaris™ オペレーティング環境、ゲートウェイとサーバーの設定、ファイアウォールのインストール、および Directory Server による認証と Access Manager による SSO に対して特定の変更を行う必要があります。また、証明書、SSL 暗号化、グループおよびドメインアクセスを使用できます。

オペレーティング環境の保護

「システムの強化」とよく呼ばれる次のことを実行して、オペレーティング環境におけるセキュリティー侵害の可能性を削減します。

- **オペレーティング環境のインストールサイズを最小限にします**。インターネット、または信頼できないネットワークにさらされた環境に Sun サーバーをインストールする場合は、Solaris のインストールをホストするアプリケーションをサポートするのに必要なパッケージの数を最小限にします。サービス、ライブラリ、およびアプリケーションを最小限の数にすると、管理する必要のあるサブシステムの数が少なくなるのでセキュリティーの向上に役立ちます。

  Solaris™ Security Toolkit ソフトウェアは、Solaris オペレーティング環境システムの最小化、強化、および保護のための柔軟で幅広い手段を提供します。このツールキットは主に Solaris システムの保護を容易に自動的に行えることを目的に開発されました。次の URL を参照してください。

  `http://www.sun.com/software/security/jass/`

- **ファイルシステムの変更を追跡および監視します**。セキュリティー対策を実施する必要があるシステムでは、ファイルの変更を追跡し、侵入を検出するためにファイルの変更の制御および監査を行うツールが必要不可欠です。Tripwire for Servers、または Solaris Fingerprint Database (SunSolve Online から入手可能) などの製品を使用できます。

プラットフォームセキュリティーの使用

通常は Portal Server を信頼できるネットワークにインストールします。ただし、このような安全な環境でも、それらのサーバーのセキュリティーには特別な注意が必要です。

UNIX ユーザーのインストール

次に示す 3 種類の UNIX ユーザー下に Portal Server をインストールおよび構成できます。

- **root**: これはデフォルトのオプションです。すべての Portal Server コンポーネントは、システムスーパーユーザーとして実行されるようにインストールおよび設定されます。この設定では、次のようなセキュリティーの問題が生じます。
  - アプリケーションのバグを利用して、システムに root アクセスが可能です。
  - 一部のテンプレートを変更するのに、root アクセスが必要になります。これは、システムを脅かす可能性があるシステム管理者でないユーザーにこの権限が通常付与されることになるので、セキュリティーの問題になる可能性があります。
- **ユーザー nobody**: Portal Server をユーザー nobody (uid 60001) としてインストールできます。ユーザー nobody には何の権限もないため、システムファイルを作成、読み取り、あるいは変更することはできないので、システムのセキュリティーを向上できます。この機能は、ユーザー nobody が Portal Server を使用してシステムファイルにアクセスし、システムに侵入するのを防止します。

  ユーザー nobody にはパスワードがないので、正規のユーザーが nobody になるのを防止します。スーパーユーザーのみが、パスワードの入力を求められずにユーザーを変更できます。したがって、Portal Server サービスを起動および停止するには引き続き root アクセスが必要です。

  詳細は、『Java Enterprise System インストールガイド』を参照してください。
- **非 root ユーザー** : 正規の UNIX ユーザーとして Portal Server を実行できます。正規ユーザーのセキュリティーの利点は、ユーザー nobody のセキュリティーの利点と似ています。正規の UNIX ユーザーには、サービスを起動、停止、および設定できるといった他の利点もあります。インストール後、一部のファイルの所有権を変更する必要があります。

  詳細は、『Java Enterprise System インストールガイド』を参照してください。

アクセス制御の制限

従来の UNIX のセキュリティーモデルは通常、絶対的ですが、代替ツールを使用していくらか柔軟にできます。それらのツールは、異なる UNIX コマンドなどの個々のリソースに対するきめ細かなアクセス制御を可能にするために必要な手段になります。たとえば、このツールセットは、Portal Server を `root` として稼働させるのを可能にし、また特定のユーザーおよびロールに Portal Server フレームワークの起動、停止、および維持のためのスーパーユーザー権限を与えます。

それらのツールを次に示します。

- Role-Based Access Control (RBAC): Solaris™ 8 および Solaris™ 9 には、スーパーユーザー権限をパッケージ化し、それらをユーザーカウントに割り当てるための Role-Based Access Control (RBAC) が含まれています。RBAC は、権限の分離、ユーザーに対する権限付き操作の付与の制御、およびアクセス制御のさまざまなレベルを実現可能にします。
- Sudo: Sudo は公開されているソフトウェアであり、システム管理者が特定のユーザーに別のユーザーとしてコマンドを実行する権限を付与することを可能にします。次の URL を参照してください。

  `http://www.courtesan.com/sudo/sudo.html`

非武装ゾーン (DMZ) の使用

最高のセキュリティーを実現するには、2 つのファイアウォール間の DMZ にゲートウェイをインストールします。もっとも外側のファイアウォールはインターネットからゲートウェイへの SSL トラフィックのみを通し、次にゲートウェイがトラフィックを内部ネットワークのサーバーへ転送します。

異なるノードにある Portal Server と Access Manager

Portal Server と Access Manager は異なるノードに置くことができます。このタイプの配備には、次の利点があります。

- アイデンティティーサービスをポータルサービスとは別に配備できます。Portal Server は、アイデンティティーサービスを使用する多くのアプリケーションのうちの 1 つにすることができます。
- 認証サービスとポリシーサービスを、Portal Server 関連のアプリケーションを含むプロバイダのアプリケーションから分けることができます。
- 他の Web コンテナが Access Manager を使用して、そのポータルのカスタマイズの開発を支援できます。

注 Portal Server と Access Manager を異なるノードに置く場合、Access Manager SDK は Portal Server と同じノードに存在する必要があります。Web アプリケーションとサポートする認証デーモンは、Portal Server インスタンスとは別のノードに置くことができます。

Access Manager SDK は、次のコンポーネントから構成されています。

アイデンティティー管理 SDK: ユーザー、ロール、グループ、コンテナ、組織、組織単位、およびサブ組織を作成および管理するための枠組みを提供します。

認証 API および SPI: 認証サービスのすべての機能へのリモートアクセスを可能にします。

ユーティリティー API: システムのリソースを管理します。

ロギング API および SPI: 数ある中でも、アクセスの承認、アクセスの拒否、およびユーザーの活動を記録します。

クライアント検出 API: リソースへアクセスしようとしているクライアントのブラウザのタイプを検出し、適切にフォーマットされたページで応答します。

SSO API: セッショントークンの検証と管理のインタフェース、またユーザー認証のクレデンシャルを管理するインタフェースを提供します。

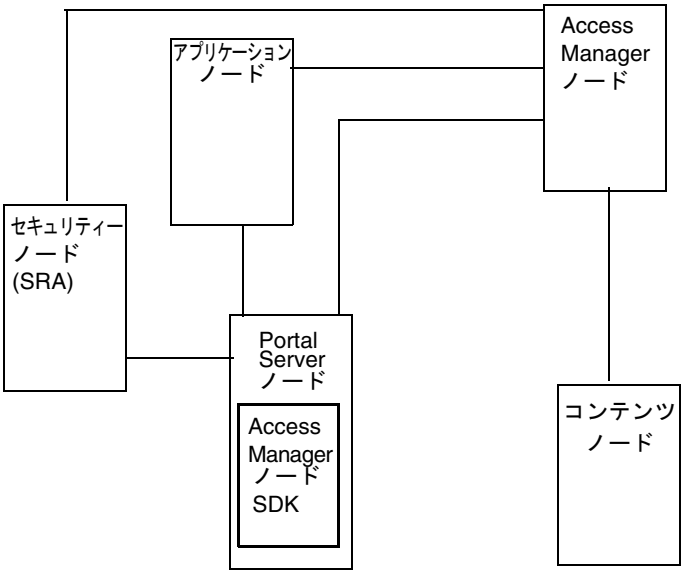
ポリシー API: Access Manager のポリシーを評価および管理し、ポリシーサービスの追加機能を提供します。

SAML API: 認証、承認決定、および属性の情報を交換します。

連携管理 API: Liberty Alliance Project 仕様に基づいた機能を追加します。

図 5-6 は、別々のノードに存在する Access Manager と Portal Server を示しています。

図 5-6 異なるノードにある Portal Server と Access Manager



Portal Server と Access Manager を分けて実装すると、次の 3 つの図が示すようなポータルサービスアーキテクチャーの配備に対する他のトポロジの並びが可能になります。

図 5-7 は、1 つの Access Manager および 2 つの Directory Server と機能するように構成された 2 つの Portal Server インスタンスを示しています。ここでは、Access Manager と Directory Server の両方が Java Enterprise System Sun Cluster 環境で動作します。この構成は、Access Manager インスタンスと Directory Server インスタンスが障害でない場合に理想的です。

図 5-7 2 つの Portal Server と 1 つの Access Manager

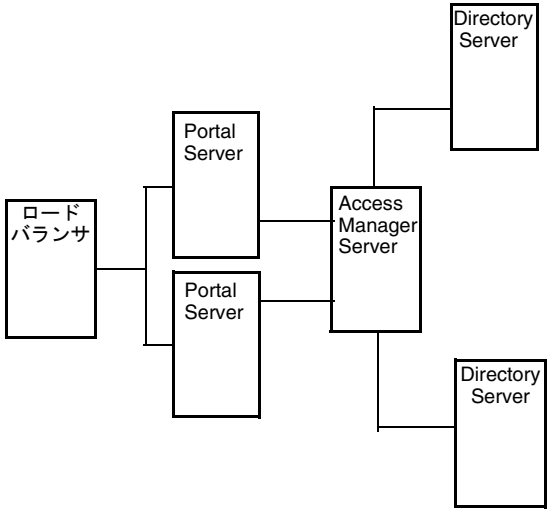


図 5-8 は、Portal Server からの認証スループットを 2 つの Access Manager にロードバランスすることを可能にする構成を示しています。

この構成は、Portal Server が広い帯域幅のネットワーク接続を備えたハイエンドの中規模から大規模のサーバー (つまり 1 ～ 4 個のプロセッサ) に存在するときに実現できます。ポリシーサービスと認証サービスを提供する Access Manager は、2 つの中規模のサーバーに置くことができます。

図 5-8　1 つの Portal Server と 2 つの Access Manager

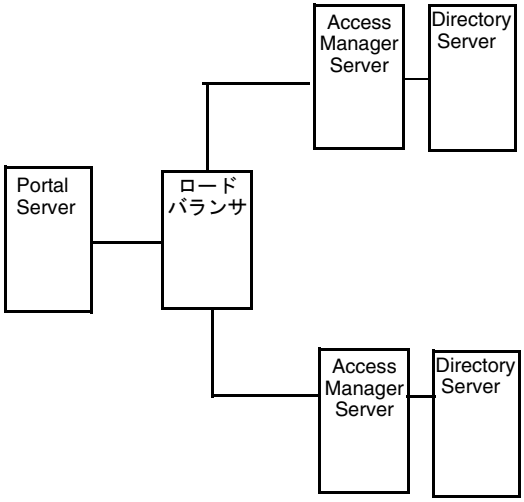


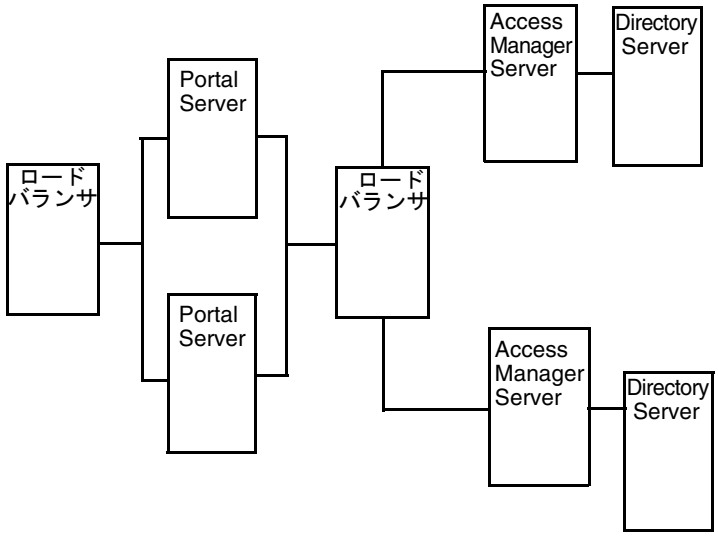
図 5-9 は、水平方向のサーバーファームによって実現された最大の水平方向のスケーラビリティーと、より高い可用性の構成を示しています。最大のスループットと高可用性の実現のために、2 つの Portal Server の前にロードバランサを置くことができます。

別のロードバランサを、Portal Server と Access Manager の間に置いて、認証プロセスとポリシープロセスが高可用性のための負荷の分散とフェイルオーバーの手段となるようにできます。

このシナリオでは、Portal サービスに Blade 1500s を利用して負荷を分散し、これと似た Blade を使用して Access Manager サービスと Directory サービスのそれぞれをホストできます。図 5-9 に示されたアーキテクチャーでは、製品スタックのそれぞれに冗長のサービスが存在するので、計画外の停止時間を最小限に、またはなくすことができます。

ただし、予定された停止時間は依然問題になります。アップグレードまたはパッチに Access Manager ソフトウェアが使用する Directory Server ソフトウェアスキーマの変更が含まれる場合、Directory Server に格納されたスキーマ情報を更新するためにこのソフトウェアのすべてのコンポーネントを停止する必要があります。ただし、スキーマ情報の更新は、ほとんどのパッチアップグレードではめったに発生しないとみなすことができます。

図 5-9　2 つの Portal Server と 2 つの Access Manager



Portal Server と Access Manager のサーバーの 2 つのインスタンスが同じ LDAP を共有している場合、後続のすべての Portal Server、Access Manager、およびゲートウェイについて次のように対処します。

1. Portal Server と Access Manager のサーバーに最初にインストールされたインスタンスと同期するように、AMConfig.properties 内の次の領域を変更します。

   # パスワードの暗号化と復号化に使用するキー。
   am.encryption.pwd=t/vnY9Uqjf12NbFywKuAaaHibwlDFNLO< == この文字列を、当初のポータルのインストール内容に置き換えてください。

   /* 次のキーは、アプリケーション認証モジュール用の共有秘密キーです。
   com.iplanet.am.service.secret=AQICxIPLNc0WWQRVlYZN0PnKgyvq3gTU8JA9 <== この文字列を、当初のポータルのインストール内容に置き換えてください。

2. /etc/opt/SUNWam/config/ums では、Portal Server と Access Manager のサーバーに最初にインストールされたインスタンスと同期するように、serverconfig.xml 内の次の領域を変更します。

   ```
   <DirDN>

       cn=puser,ou=DSAME Users,dc=sun,dc=net

   </DirDN>

       <DirPassword>
   ```

```
        AQICxIPLNc0WWQT22gQnGgnCp9rUf+FuaqpY <== この文字列を、当初のポータルのインストール内容に置き換えてください。
  </DirPassword>


<DirDN>
   cn=dsameuser,ou=DSAME Users,dc=sun,dc=net
</DirDN>
     <DirPassword>
          AQICxIPLNc0WWQT22gQnGgnCp9rUf+FuaqpY <== この文字列を、当初のポータルのインストール内容に置き換えてください。
     </DirPassword>
```

3. amserver サービスを再起動します。

SRA の配備シナリオの設計

SRA ゲートウェイは、インターネットから送信されるリモートユーザーセッションと企業イントラネットの間のインタフェースおよびセキュリティーバリアとして機能します。ゲートウェイには、次の 2 つの主な機能があります。

- 受信ユーザーセッションに対する、身元の確認やプラットフォームへのアクセスの許可または拒否などの基本認証サービスを提供します。
- ユーザー向けにイントラネットのコンテンツへの Web ベースのリンクを有効にするためのマッピングサービスと書き換えサービスを提供します。

インターネットアクセスの場合、128 ビット SSL を使用して、ユーザーのブラウザと Portal Server 間の最高のセキュリティー対策と暗号化、または通信を実現します。ゲートウェイ、Netlet、NetFile、Netlet プロキシ、リライタプロキシ、および Proxylet が SRA の主なコンポーネントです。

ここでは、それらのコンポーネントの可能な構成をいくつか示します。業務のニーズに基づいて正しい構成を選択してください。このセクションでは、指針を示すだけで、完全な配備の参考情報を提供するわけではありません。

ヒント authlessanonymous ページをゲートウェイ経由で表示するように設定するには、ゲートウェイプロファイルの非認証 URL に `/portal/dt` を追加します。ただし、これは、普通のユーザーの場合でも、ポータルページは認証を必要とせず、セッションの検証が実行されないことを意味します。

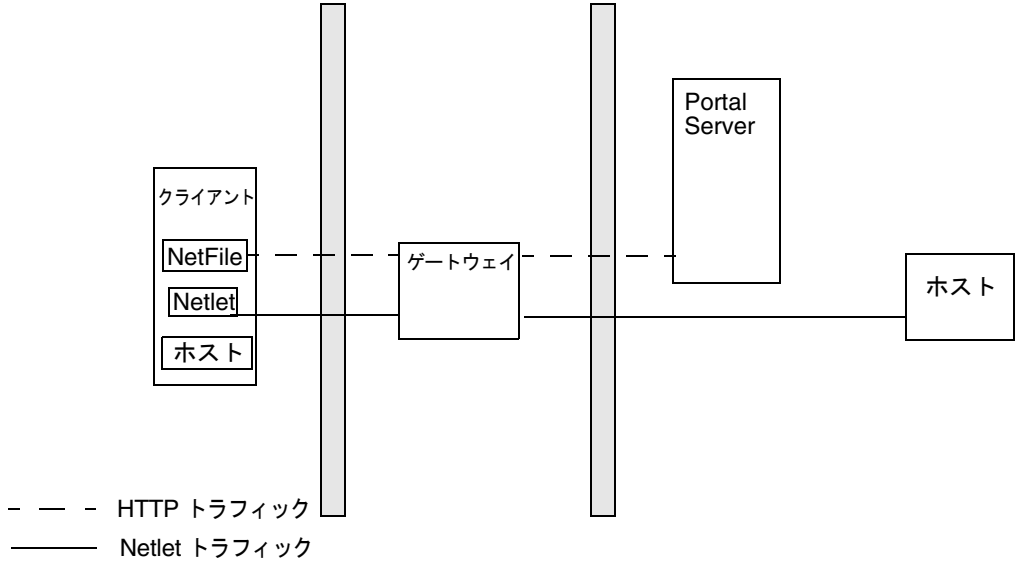
基本 SRA 構成

図 5-10 は、SRA のもっとも単純な構成を示しています。この図では、クライアントのブラウザが NetFile および Netlet を実行しています。ゲートウェイは、2 つのファイアウォールの間の DMZ 内の個別のマシンにインストールされています。Portal Server は、イントラネット内の 2 番目のファイアウォールの外側に置かれています。クライアントがアクセスする他のアプリケーションホストも、イントラネット内の 2 番目のファイアウォールの外側に置かれています。

ゲートウェイは、DMZ 内にあり、ファイアウォール内の開いた外部ポートを通してクライアントのブラウザがゲートウェイと通信します。2 番目のファイアウォールでは、HTTP または HTTPS トラフィックのために、ゲートウェイは内部ホストと直接通信できます。セキュリティーポリシーがこれを許可しない場合は、ゲートウェイと内部ホストとの間に SRA プロキシを使用します。Netlet トラフィックの場合、ゲートウェイから目的のホストへの直接接続になります。

SRA プロキシを使用しない場合、SSL トラフィックはゲートウェイに制限され、ゲートウェイと内部ホスト間ではトラフィックは暗号化されません (内部ホストが HTTPS モードで実行されていないかぎり)。内部ホストに対してゲートウェイが Netlet 接続を開始する必要がある内部ホストは、DMZ から直接アクセス可能である必要があります。これはセキュリティーの問題になる可能性があるので、この構成はもっとも単純なインストールにのみ推奨します。

図 5-10 基本 SRA 構成

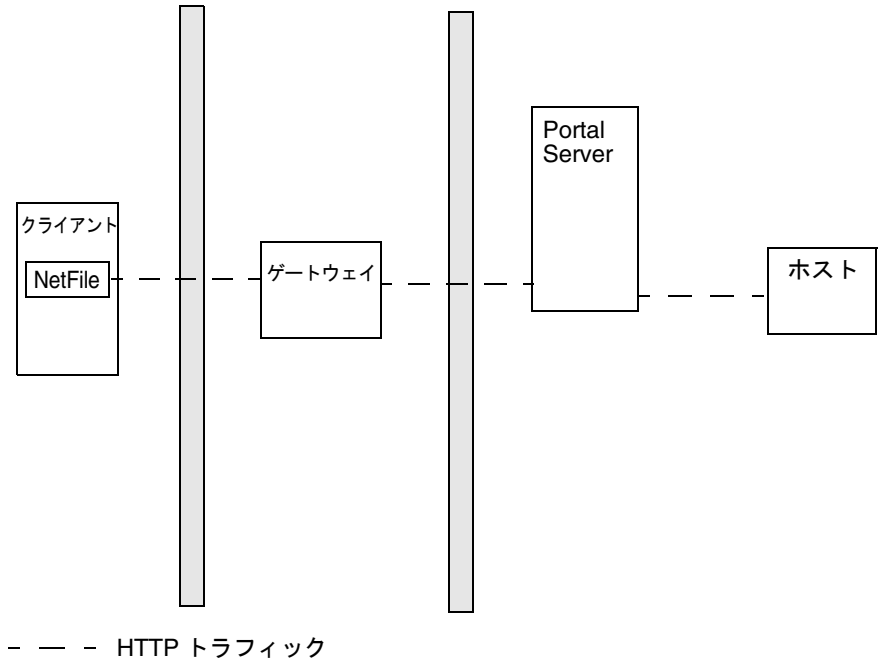


Netlet の無効化

図 5-11 は、Netlet が無効ということ以外は基本 SRA 構成と同様のシナリオを示しています。クライアントの配備がイントラネットと通信する必要があるアプリケーションを安全に実行するために Netlet を使用しない場合は、パフォーマンス向上のためにこの構成を使用します。

この構成を拡張して、他の配備シナリオと組み合わせて、パフォーマンスを向上し、拡張可能なソリューションを提供できます。

図 5-11　Netlet の無効化

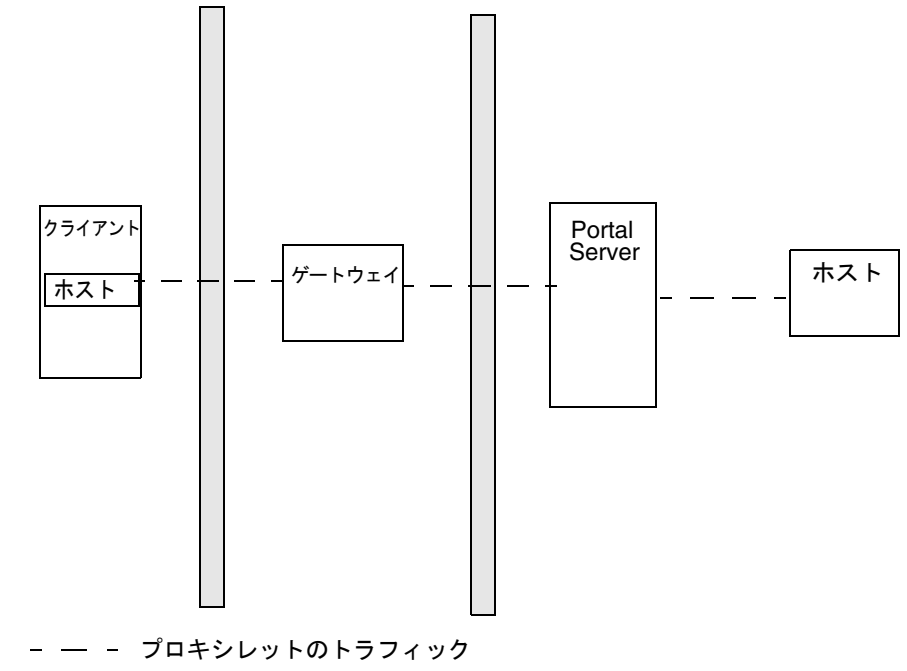


ホスト

図 5-12 で示すプロキシレットは、イントラネットのリソースをクライアントに公開しなくても、ユーザーがインターネットを使用してイントラネットのリソースに安全にアクセスできるようにします。

プロキシレットでは、ゲートウェイのトランスポートモード (HTTP または HTTPS のどちらか) が継承されます。

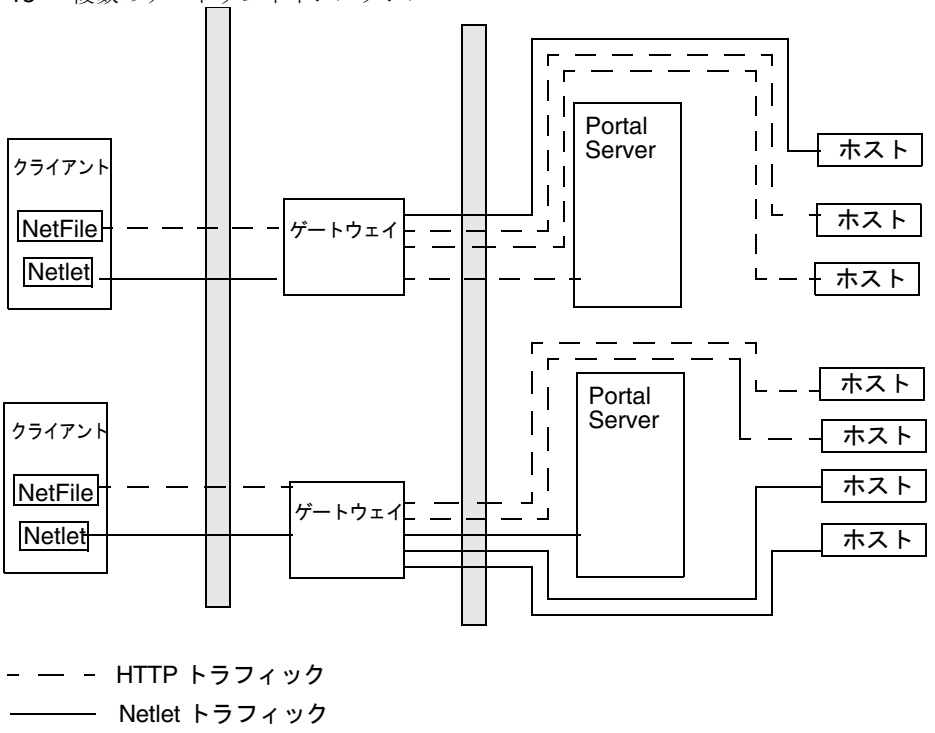
図 5-12　プロキシレット



複数のゲートウェイインスタンス

図 5-13 は、SRA の基本構成の拡張を示しています。複数のゲートウェイインスタンスが、同じマシンまたは複数のマシンで稼働します。別々のプロファイルで複数のゲートウェイインスタンスを起動できます。詳細は、『Portal Server Secure Remote Access 6 管理ガイド』の第 2 章「ゲートウェイの設定」を参照してください。

図 5-13　複数のゲートウェイインスタンス



注　114 ページの図 5-13 はゲートウェイと Portal Server の 1 対 1 の対応を示していますが、実際の配備では必ずしもこのようになる必要はありません。複数のゲートウェイインスタンスや複数の Portal Server インスタンスを配備可能であり、また構成によってはどのゲートウェイも任意の Portal Server にアクセスできます。

この構成の欠点は、各接続要求のために 2 番目のファイアウォールで複数のポートを開く必要があるということです。これは、セキュリティーの問題になる可能性があります。

Netlet プロキシとリライタプロキシ

図 5-14 は、イントラネットに Netlet プロキシとリライタプロキシがある構成を示しています。これらのプロキシの場合、2 番目のファイアウォールではポートが 2 つだけ開いている必要があります。

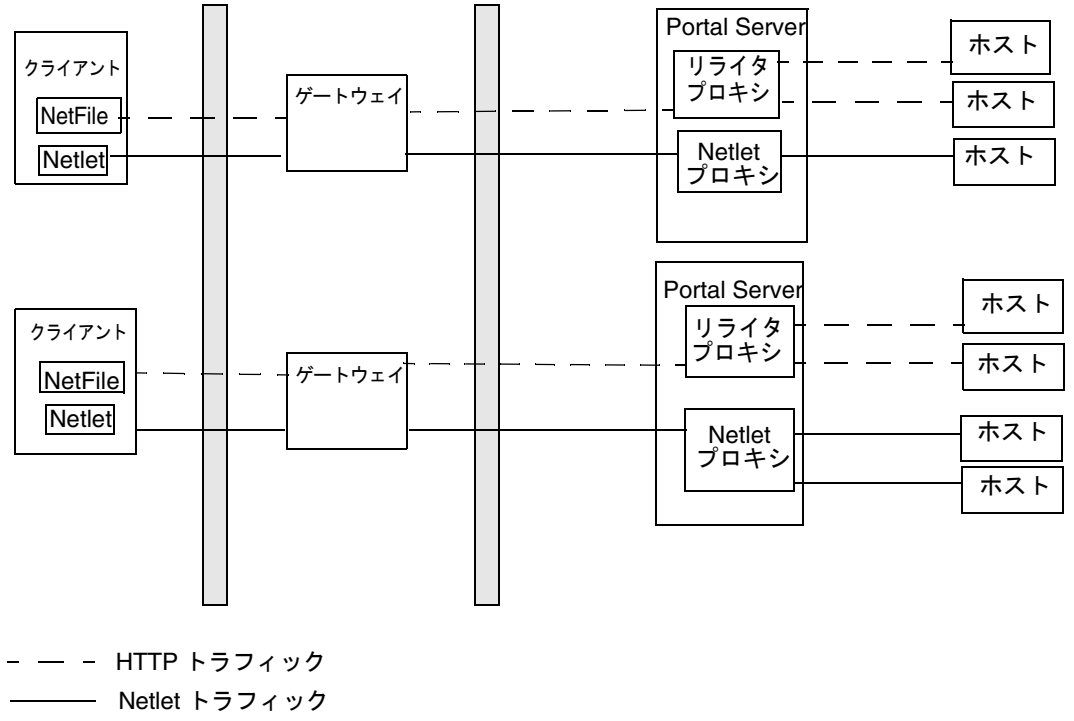
この構成では、ゲートウェイはアプリケーションホストと直接やり取りする必要はありませんが、すべての Netlet トラフィックを Netlet プロキシへ、リライタトラフィックをリライタプロキシへ転送します。Netlet プロキシはイントラネット内にあるので、2 番目のファイアウォールで複数のポートを開かなくても、すべての必要なアプリケーションホストと直接やり取りできます。

DMZ 内のゲートウェイと Netlet プロキシとの間のトラフィックは暗号化され、Netlet プロキシでのみ復号化されるので、セキュリティーが向上します。

リライタプロキシが有効な場合、要求が Portal Server ノード宛てのものかどうかにかかわらず、すべてのトラフィックがリライタプロキシに転送されます。これによって、DMZ 内のゲートウェイからイントラネットへのトラフィックは常に暗号化されることが保証されます。

Netlet プロキシ、リライタプロキシ、および Portal Server がすべて同じノードで稼働するので、そのような配備シナリオではパフォーマンスの問題が発生する場合があります。この問題は、Portal Server ノードの負荷を軽減するためにプロキシを別々のノードにインストールすると解決できます。

図 5-14　Netlet プロキシとリライタプロキシ



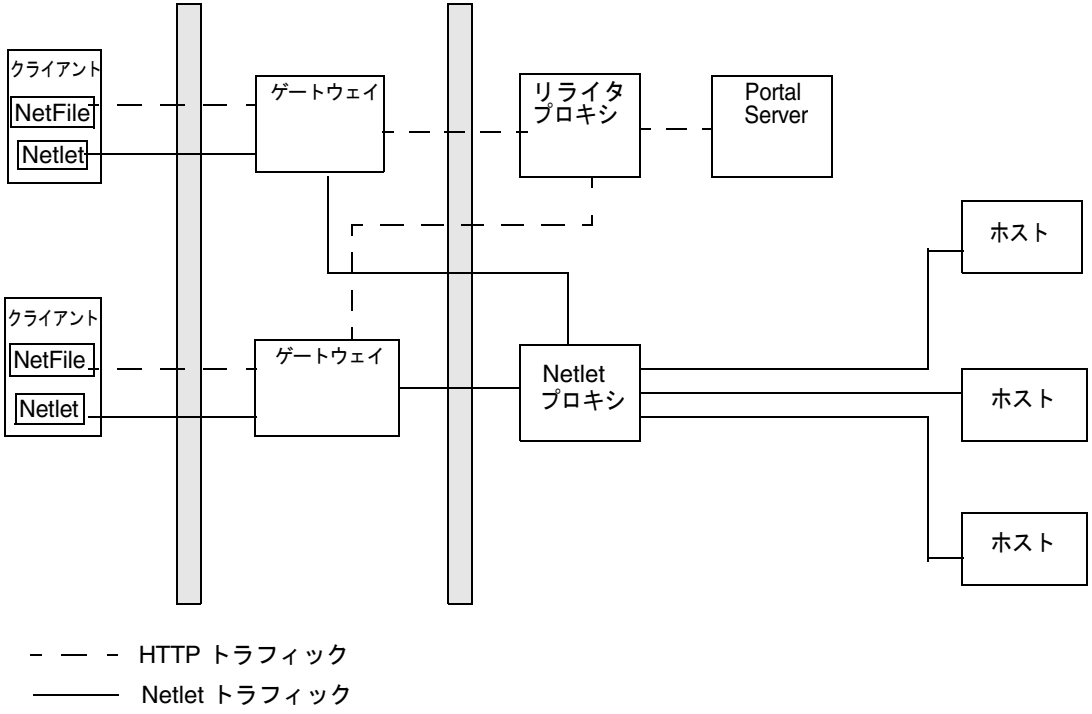
別々のノードにある Netlet プロキシとリライタプロキシ

Portal Server ノードの負荷を軽減し、なおかつパフォーマンスを向上させて同じレベルのセキュリティーを提供するには、Netlet プロキシとリライタプロキシを別々のノードにインストールできます。この配備は、プロキシを使用して Portal Server を DMZ から保護できるというさらなる利点があります。これらのプロキシを実行するノードは、DMZ から直接アクセス可能である必要があります。

図 5-15 は、別々のノードにある Netlet プロキシとリライタプロキシを示しています。ゲートウェイからのトラフィックは別のノードに転送され、そのノードはトラフィックをプロキシ経由で、必要なイントラネットのホストに転送します。

Netlet プロキシとリライタプロキシの複数のインスタンスを持つことや複数の Netlet プロキシとリライタプロキシをインストールすることが可能です。各ゲートウェイを、可用性に応じてラウンドロビン方式でプロキシのさまざまなインスタンスとのやり取りを試みるように設定できます。

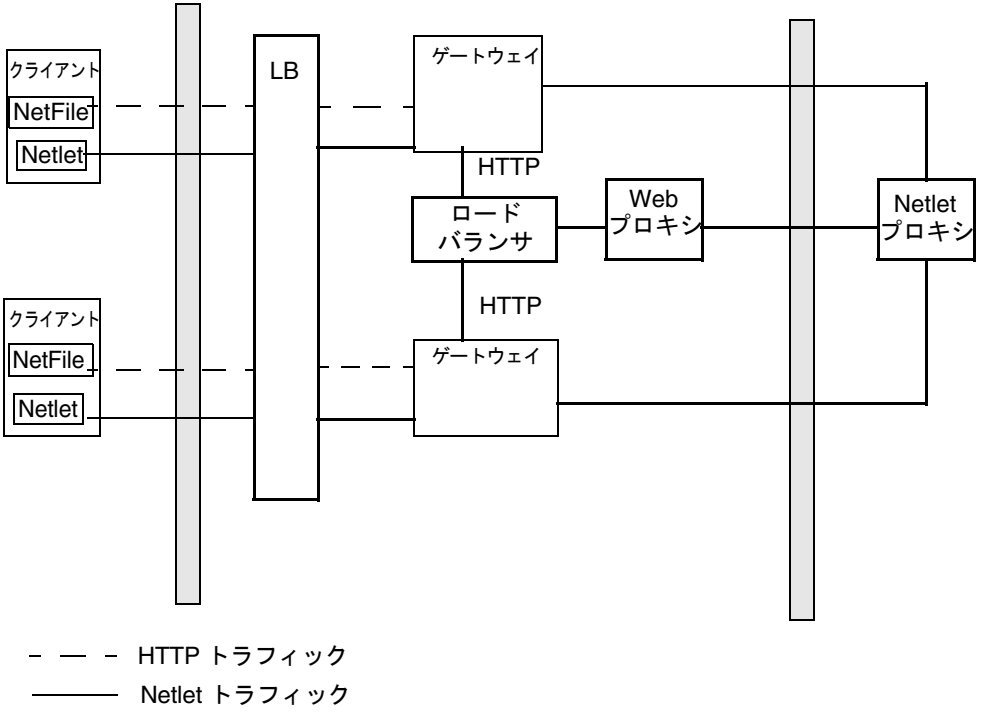
図 5-15　別々のノードにあるプロキシ



2 つのゲートウェイと Netlet プロキシの使用

ロードバランサは、Portal Server および Access Manager の冗長サービスの高可用性のためのフェイルオーバーのメカニズムを提供します。

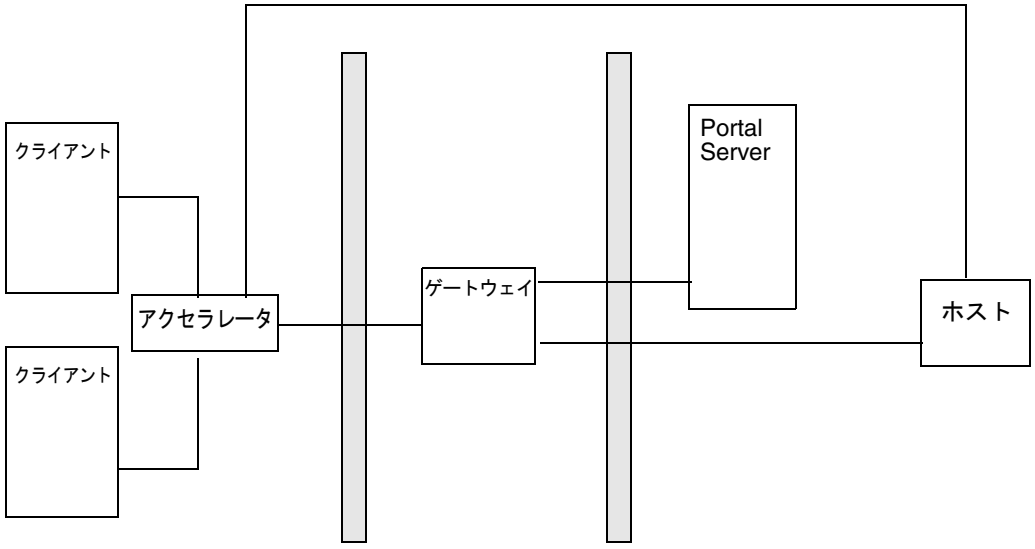
図 5-16　2 つのゲートウェイと Netlet プロキシ



アクセラレータの使用

外部の SSL デバイスをオープンモードでゲートウェイの前で実行するように設定できます。これは、クライアントと SRA の間に SSL リンクを提供します。アクセラレータの詳細は、『Portal Server Secure Remote Access 6 管理ガイド』を参照してください。

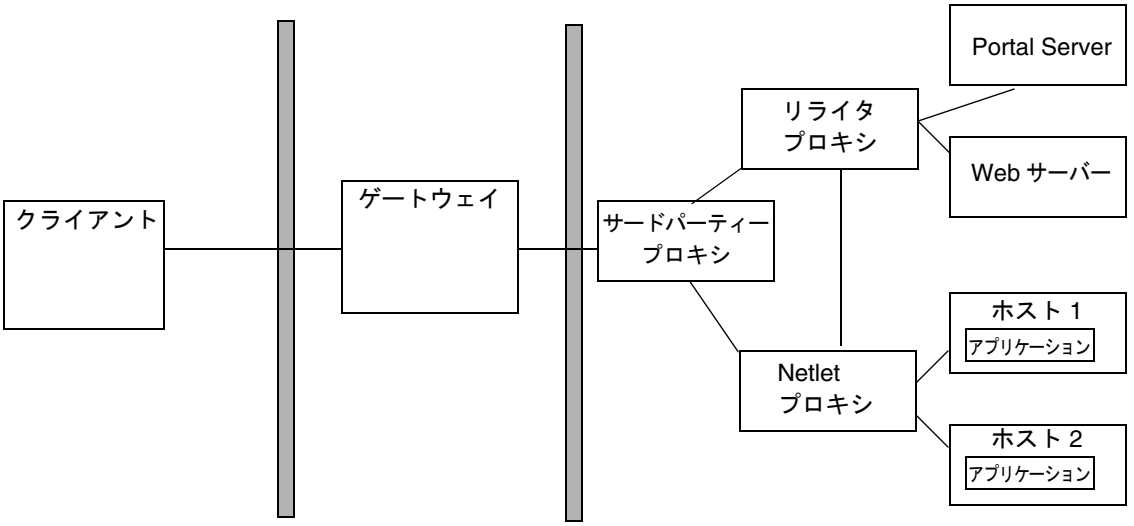
図 5-17　外部のアクセラレータを使用する SRA ゲートウェイ



サードパーティーのプロキシを使用する Netlet

図 5-18 は、サードパーティーのプロキシを使用して、2 番目のファイアウォールのポートの数を 1 つに制限する例を示しています。サードパーティーのプロキシを使用して、リライタプロキシまたは Netlet プロキシに到達するようにゲートウェイを設定できます。

図 5-18　Netlet とサードパーティーのプロキシ

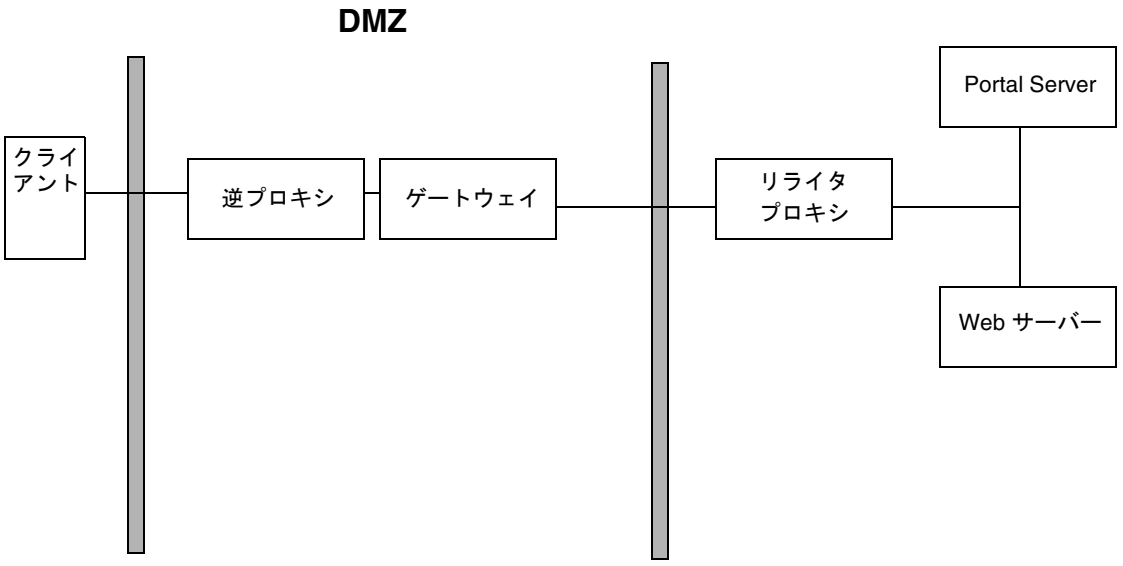


逆プロキシ

プロキシサーバーがインターネットのコンテンツをイントラネットに配信するのに対して、逆プロキシサーバーはイントラネットのコンテンツをインターネットに配信します。逆プロキシの特定の配備の際に、インターネットコンテンツのロードバランスおよびキャッシングが行われるように設定できます。

図 5-19 は、インターネットとイントラネットの両方のコンテンツを承認されたユーザーに配信するために、ゲートウェイの前に逆プロキシを配置する方法を示しています。ゲートウェイが Web コンテンツを配信するときには、このコンテンツに基づいた後続するブラウザの要求すべてがゲートウェイを通じてルーティングされるようにする必要があります。これは、このコンテンツ内のすべての URL を確認し、必要に応じて書き換えることによって実現します。

図 5-19　ゲートウェイの前に逆プロキシを使用



地域化の設計

地域化とは、テキストおよび文化的な内容を特定の対象者向けに適合させる処理のことです。地域化には、次の 2 つの方法で取り組むことができます。

1. 製品全体を提供していない他の言語に地域化します。これは、通常、専門のサービス組織が行います。
2. 地域化をサポートするために変換できる Portal Server のカスタマイズ可能な部分を次に示します。
    - テンプレートおよび JSP ファイル
    - リソースバンドル
    - ディスプレイプロファイルのプロパーティー

高度な言語の地域化の場合、テンプレートのディレクトリのために正しく定義されたディレクトリ構造を作成します。

アップグレードパスを保つには、カスタムコンテンツとカスタムコードをデフォルトのディレクトリ外に保持します。地域化については、『Portal Server 6 Developer's Guide』を参照してください。

コンテンツと設計の実装

ポータルデスクトップは、Portal Server のプライマリエンドユーザーインタフェースであり、プロバイダアプリケーションプログラミングインタフェース (PAPI) による広範なコンテンツ集約のメカニズムを備えています。ポータルデスクトップには、コンテナ階層と、特定のチャネルを構築するための基本構築ブロックとを有効にするさまざまなプロバイダが表示されます。コンテンツプロバイダとチャネルデータを保存する場合、ポータルデスクトップは Access Manager サービスのトップでディスプレイプロファイルデータ保管メカニズムを実行します。

コンテンツの集約に使用できるさまざまな技術を次に示します。

- 構築ブロックプロバイダを使用したチャネルの作成
- `JSPProvider` を使用したチャネルの作成
- Portal Server タグライブラリを使用したチャネルの作成
- カスタム構築ブロックプロバイダを使用したチャネルの作成
- コンテナチャネルを使用したコンテンツの編成

詳細は、『Portal Server 6 Developer's Guide』と『Portal Server 6 Desktop Customization Guide』を参照してください。

静的なポータルコンテンツの配置

静的なポータルコンテンツは、*web-container-install-root*/SUNWam/public_html ディレクトリまたは *web-container-install-root*/SUNWam/public_html ディレクトリ (Web コンテナのドキュメントルート) 下のサブディレクトリに配置します。

web-container-install-root/SUNWps/web-apps/https-*server*/portal/ ディレクトリは非公開のディレクトリであるため、コンテンツをこのディレクトリに配置しないでください。このディレクトリに配置されたコンテンツは、パッチまたはその他の更新時に Portal Server の Web アプリケーションが再配備されたときに削除の対象になります。

統合の設計

ここでは、低レベルの設計で考慮する必要がある統合関連の情報を提供します。

カスタム Access Manager サービスの作成

Access Manager のサービス管理は、Access Manager サービスとして属性のグループを定義、統合、および管理する手段になります。管理のためにサービスを準備するには、次の手順が伴います。

1. XML サービスファイルを作成します。
2. 新しいオブジェクトクラスで LDIF ファイルを設定し、XML サービスファイルと新しい LDIF スキーマの両方を Directory Service にインポートします。
3. Access Manager の管理コンソールを使用して、複数のサービスを組織またはサブ組織に登録します。
4. 組織ごとに属性 (登録後) の管理およびカスタマイズを行います。

詳細については、Access Manager のマニュアルを参照してください。

アプリケーションの統合

アプリケーションの Portal Server との統合および配備は、配備作業の中でも、もっとも重要な作業の 1 つです。アプリケーションのタイプを次に示します。

- **チャネル** : 限定されたコンテンツオプションを提供します。「ミニブラウザ」ではありません。
- **ポートレット** : ポータルのコンテキストの範囲内で要求を処理し、コンテンツを生成する、プラグイン可能な Web コンポーネント。Portal Server ソフトウェアでは、ポートレットコンテナがポートレットを管理します。概念的には、ポートレットはプロバイダと同等です。

- **ポータルアプリケーション** : ポータルアプリケーション専用のブラウザウィンドウ内のチャネルから起動されます。Portal Server は、Access Manager サービスとして作成された NetMail などのアプリケーションをホストします。また、Portal API と Access Manager API にアクセスします。
- **サードパーティーのアプリケーション** : Portal Server とは別にホストされますが、Portal Server からアクセスされます。リライタを呼び出す URL スクレイパーは、チャネルに表示できるように Web ページを書き換えます。Access Manager を使用してシングルサインオンを有効にします。

独立ソフトウェアベンダー

次に独立ソフトウェアベンダー (ISV) の統合機能のいくつかのタイプを示します。

- **アプリケーションのユーザーインタフェース** : この統合機能は、安全なアクセスのためにプロバイダの API と SRA を使用します。SRA は単独では統合タイプではありません。たとえば、FatWire、Interwoven、SAP、Tarantella、Documentum、Vignette、PeopleSoft、Siebel、Citrix、YellowBrix などがあります。
- **セキュリティー製品** : この統合機能は、Access Manager の Login API を使用して、カスタム認証スキームを使用したポータルアクセスを有効にます。たとえば、RSA などがあります。
- **コンテンツの管理** : この統合機能は、Portal Server へのデータアクセスを提供し、データの検索を可能にします。たとえば、FatWire、Interwoven、Vignette などがあります。
- **コンテンツのシンジケート** : この統合機能は、Web サイトに表示される情報の管理およびカスタマイズを行います。たとえば、YellowBrix、Pinnacor などがあります。
- **コラボレーションソフトウェア** : この統合機能は、Sun Java System Instant Messaging 製品がコラボレーションセッションを 1 つのフォーラムから別のフォーラムに移すことを可能にします。たとえば、WebEx、BeNotified、Lotus などがあります。
- **監視** : この統合機能は、課金、パフォーマンスの測定、および診断に的を絞り、このためにログファイル (または Access Manager の Logging API) およびトラフィックの snooping を利用します。たとえば、Mercury Interactive、Hyperion、Informatica などがあります。
- **ポータルの機能の拡張** : この統合機能は、製品が Portal Server に機能を追加することを可能にします。たとえば、Altio、Bowstreet、グループ機能を追加するルールエンジン、ダイナミックな標準のポータルデスクトップおよびプロバイダコンテンツ (HNC) などがあります。

- **統合可能なポータルスタック**：この統合機能には、Portal Server の要素を置き換える製品が含まれています。たとえば、Access Manager、LDAP などがあります。

注　Portal Server は現在別の LDAP ソリューションを統合できません。Access Manager と Portal Server は、他の LDAP 実装にない機能を利用します。

Portal Server とユーザーインタフェースの統合が行われる「深さ」は、統合がどの程度完了したかを示します。深さは、統合の補完的な性質を説明するために使用する用語であり、次のようなアイテムを指します。

- Portal Server そのものからのアプリケーションの可用性
- セキュアモードのアプリケーションの可用性 (SRA、Netlet ルールを使用)
- シングルサイオンを使用する能力

一般に、アプリケーションが Portal Server と統合する程度は、次のように見なすことができます。

- **浅い統合**：この統合は、基本的に Portal Server を開始点として使用します。ユーザーはポータルにログインし、Web アプリケーションを起動するリンクをクリックします。
- **深い統合**：ユーザーは、Portal Server 内のチャネルが提供するユーザーインタフェースに直接アクセスします。つまり、統合されたソフトウェアは、ポータル内で動作します。その他のウィンドウやアプレットは表示されません。

Microsoft Exchange の統合

JavaMail™ API の使用は、Microsoft Exchange メッセージングサーバーを Portal Server と統合する主なオプションの 1 つです。JavaMail API は、Java テクノロジに基づいたメールおよびメッセージングアプリケーションを構築するためのプラットフォーム独立およびプロトコル独立フレームワークです。JavaMail API は、Java プラットフォームのオプションのパッケージとして実装され、Java™ 2 Platform, Enterprise Edition の一部としても利用できます。

JavaMail は、メールを管理するための共通の統一 API を提供します。JavaMail は、サービスプロバイダが Java プログラミング言語を使用して標準ベースのまたは独自のメッセージングシステムへの標準のインタフェースを提供するのを可能にします。この API を使用して、アプリケーションはメッセージストアにアクセスし、メッセージを作成および送信できます。

アイデンティティーとディレクトリ構造の設計

ポータルの実装の主な部分は、ディレクトリ情報ツリー (DIT: Directory Information Tree) の設計です。DIT は、ユーザー、組織、サブ組織などを論理構造または階層構造に編成することにより、管理を効率的に行い、適切なアクセス権限をユーザーに割り当てることを可能にします。

Access Manager の組織ツリーの最上位は、デフォルトで `dc=`*fully-qualified-domain-name* と呼ばれ、インストール時に変更または指定できます。インストール後に、追加の組織を作成して、別の企業を管理することができます。作成された組織はすべて最上位組織の下に配置されます。これらのサブ組織内で、他のサブ組織を入れ子にできます。入れ子の構造の深さには制限がありません。

注 ツリーの最上位を `dc` と呼ぶ必要はありません。必要に応じてこの名前を変更できます。ただし、たとえば、`dc` など一般的な最上位で編成されたツリーでは、ツリー内の組織はロールを共有することができます。

ロールは、より効果的に、またより簡単にアプリケーションを使用するように設計されたグループ化メカニズムです。それぞれのロールはメンバー、あるいはロールを保有するエントリを持ちます。グループの場合と同じく、ロールのメンバーは明示的またはダイナミックに指定できます。

ロールメカニズムにより、そのエントリがメンバーになっているすべてのロール定義の識別名 (Distinguished Name、DN) を含む `nsRole` 属性が自動的に生成されます。各ロールは、1 人または複数のユーザーに付与できる、単一の権限や権限のセットを含んでいます。複数のロールを 1 人のユーザーに割り当てることができます。

ロールの権限はアクセス制御命令 (ACI) で定義されます。Portal Server には、いくつかのロールが事前に定義されています。Access Manager 管理コンソールを使用してロールの ACI を編集し、ディレクトリ情報ツリー内でアクセス権を割り当てることができます。用意されている例には、`SuperAdmin Role` および `TopLevelHelpDeskAdmin` が含まれます。組織間で共有できるその他のロールを作成することもできます。

Access Manager および Directory Server 構造の計画については、『Portal Server 6 管理ガイド』、『Directory Server Deployment Guide』、および『Access Manager Deployment Guide』を参照してください。

シングルサインオンの実装

Portal Server へのシングルサインオン (Single sign-on、SSO) は、Access Manager によって管理されます。SSO は、ポリシーによって許可される場合、Access Manager によってアクセスポリシーが管理されるアプリケーションをユーザーが使用できるようにします。ユーザーは、そのアプリケーションに再認証される必要はありません。

さまざまな SSO シナリオを次に示します。

- **ポータル Web アプリケーション** : 認証は Access Manager から行われ、アプリケーションは Access Manager によってユーザーのクレデンシャルを検証します。
- **スタンドアロン Web アプリケーション** : アプリケーションは別の Web コンテナでホストされ、Access Manager Web Agent は Access Manager による認証で使用されます。これには、アプリケーションのコーディングは必要ありません。さらに、Access Manager に対して直接検証を行うようにアプリケーションを変更できます。
- **スタンドアロン Java アプリケーション** : このシナリオでは、ユーザーのクレデンシャルを Access Manager に対して直接検証を行うようにアプリケーションを変更します。
- **非 Access Manager 対応アプリケーション** : このシナリオでは、アプリケーションはユーザーのクレデンシャルを保存し、必要に応じて提供します。ただし、クレデンシャルが変更される場合はユーザーが再認証を行う必要があるので、これは理想的な SSO ソリューションではありません。

ポータルデスクトップの設計

Portal Server そのもののパフォーマンスは、個々のチャネルの実行速度によって決定されます。また、ポータルに対するユーザーの体感は、ポータルデスクトップが表示される速度に基づきます。ポータルデスクトップは、最低の速度で表示されるチャネルと同じ速度でのみ読み込みを行うことができます。たとえば、10 個のチャネルから構成されるポータルデスクトップを考えてみます。9 つのチャネルが 1 ミリ秒で描画されるが、10 番目のチャネルが 3 秒かかる場合は、ポータルデスクトップはポータルが 10 番目のチャネルを処理するまで表示されません。各チャネルが要求を最短時間で処理できるようにすることによって、ポータルデスクトップのパフォーマンスの向上を実現できます。

正しい集約方針の選択と実装

速度とスケーラビリティーの向上を目的としてポータルチャネルを実装するために選択できるいくつかの方法を次に示します。

- ポータルサーバーではなく、バックエンドシステムおよびアプリケーションサーバーに処理機能を置きます。ポータルサーバーは、ユーザーからの要求の取得を最適化する必要があります。できるかぎり多くのビジネスロジック処理をバックエンドシステムに任せます。可能なかぎり、カスタマイズしたコンテンツを処理するためではなく、ユーザーに配信するためにポータルを使用します。
- バックエンドシステムが高度にスケーラブルでパフォーマンスがよい状態になるようにします。ポータルデスクトップは、(チャネルで表示する) 情報の入手先のサーバーと同程度の速度で応答します
- プロバイダを設計する際には、データの格納場所、ポータルがデータを入手する方法、プロバイダがデータを入手する方法、およびデータのタイプを理解します。たとえば、データが個々のユーザーに関係する動的なデータであるか、あるいはカスタマイズされたまたはパーソナライズされたデータを取得するのにコードが必要かどうかなどです。また、データが静的であり、小さなグループのユーザーによって共有されるかなどです。次に、データの存在場所 (たとえば、XML ファイル、データベース、フラットファイルなど) とデータの更新の頻度を理解する必要があります。最後に、プロバイダがパーソナライズされたチャネルをユーザーに配信できるように、データの処理にどのようにビジネスロジックが適用されるかを理解する必要があります。

プロバイダの操作

プロバイダの配備を計画するときには、次のことを考慮します。

- **`URLScraperProvider`**: 通常、このプロバイダは、別の Web コンテナの Web ベースのシステムが供給するダイナミックコンテンツにアクセスするために使用します。このプロバイダは、コンテンツを取得するために HTTP および HTTPS 呼び出しを使用します。バックエンドシステムに高いスケーラビリティーと可用性が要求されるので、このプロバイダは、バックエンドシステムに高い要求を課します。高いパフォーマンスを実現するために、パフォーマンスは 2 桁のミリ秒数、または 1/100 ミリ秒である必要があります。このプロバイダは、構成が単純であるため、ポータルの配備の試験段階で考え方が正しいか確認するのに役立ちます。

  `URLScraperProvider` は、ページを取得するたびにある程度の書き換えも実行します。たとえば、チャネルが別の Web サイトにホストされている写真を含むニュースのページを取得する場合、ポータルがその写真を表示できるようになるには、その写真の URL を書き換える必要があります。ポータルはその写真をホストしていないので、`URLScraperProvider` はポータルのユーザーに表示するためにその写真を書き換える必要があります。

  Portal Server の一部である URL スクレイパープロバイダは、ファイルスクレイパープロバイダとしても機能します。

  `URLScraperProvider` をファイルスクレイパープロバイダとして使用するには、次のように URL を指定します。

  ```
  String name="url" value="file://path/filename"
  ```

コンテンツの取得速度の観点から見ると、このプロバイダはもっとも優れているプロバイダです。コンテンツの最初の取得では、このプロバイダのパフォーマンスは通常 13 ～ 15 ミリ秒程度になります。それ以降の要求に、組込みのキャッシュメカニズムを使用すると、このプロバイダは通常 1 ミリ秒以下でコンテンツを配信できます。適用可能な場合、URL スクレイパープロバイダの代わりにファイルスクレイパープロバイダを使用することを考えてみます。

- **`JSPProvider`**: JavaServer Pages™ (JSP) 技術を使用します。`JSPProvider` は、1 つまたは複数の JSP ファイルからコンテンツを取得します。JSP ファイルは、静的なドキュメント (HTML のみ)、または HTML および Java プログラミング言語からなる標準の JSP ファイルにすることができます。JSP ファイルには別の JSP ファイルを含めることができます。ただし、最上位の JSP ファイルのみをディスプレイプロファイルによって設定できます。最上位の JSP ファイルは、`contentPage`、`editPage`、および `processPage` プロパーティーによって定義されます。
- **`LoginProvider`**: ポータルデスクトップチャネルによって、Access Manager の認証サービスへのアクセスを提供します。このプロバイダは、ユーザーがポータルデスクトップから直接ログインできるように、匿名ポータルデスクトップログインを可能にします。
- **`XMLProvider`**: XSLT (XML Style Sheet Language) ファイルを使用して、XML ドキュメントを HTML に変換します。XML ドキュメントタイプに一致する適切な XSLT ファイルを作成する必要があります。`XMLProvider` は、`URLScraperProvider` を拡張したものです。このプロバイダは、Web Server が提供する JAXP 1.2 JAR ファイルを使用します。
- LDAP **ベースのプロバイダ** : このタイプのプロバイダは、ユーザーおよびパーソナライズの適用についての情報をユーザープロファイルから取得します。このプロバイダは、格納された LDAP 属性の数が少ない間は効率的です。一般に、このタイプのプロバイダのパフォーマンスは良く、`URLScraperProvider` 内で提供されるファイルスクレイパープロバイダに次ぐパフォーマンスを提供します。
- **データベースプロバイダ** : このタイプのプロバイダは、そのコンテンツにバックエンドデータベースを利用します。このプロバイダは、データベース接続ポーリングを作成し、少数のクエリー (1 つまたは 2 つのクエリー) を使用することを要求します。また、HTML のフォーマットのために余分な作業を実行する必要がある場合があります。一般に、データベース接続ポーリング、大きなデータベースクエリー、不良コーディング、または取得されたデータにインデックスが作成されていないなどの原因により、このタイプのプロバイダのパフォーマンスはもっとも低くなります。さらに、データを取得すると、ポータルはデータをポータルデスクトップに表示するために大量の処理を実行する必要があります。このタイプのプロバイダを使用する場合、データ処理ロジックをできる限りデータベースに任せます。また、データベースチャネルがある状態とない状態でポータルのパフォーマンスを測定してユーザープロファイルに記録します。

クライアントのサポート

Portal Server は次のブラウザをクライアントとしてサポートします。

- Internet Explorer 5.5 および 6.0
- Netscape™ Communicator 4.7x 以上

最新のリストは、『Portal Server 6 リリースノート』を参照してください。

HTML、WML、またはその他のプロトコルのどれに基づいていようと、複数のクライアントタイプが、Access Manager に、またその結果 Portal Server にアクセスできます。この機能を有効にするには、Access Manager はクライアント検出サービス (クライアント検出 API) を使用してポータルにアクセスするクライアントのタイプを検出します。次に、そのクライアントタイプを使用して、出力に使用するポータルテンプレート、JSP ファイル、および文字エンコーディングを選択します。

注 現在、Access Manager は、Internet Explorer や Netscape Communicator などのサポートされている HTML クライアントブラウザに対するクライアントデータのみを定義します。詳細については、Access Manager のマニュアルを参照してください。

Sun Java System Portal Server Mobile Access 6.3 ソフトウェアは、Portal Server プラットフォームのサービスと機能をモバイルデバイスへ拡張し、音声アクセスのためのフレームワークを提供します。このソフトウェアは、HTML ブラウザを使用してアクセスする場合と同じコンテンツをポータルサイトのユーザーが入手できるようにします。

Mobile Access ソフトウェアは、xHTML、cHTML、HDML、HTML、WML などのモバイルマークアップ言語をサポートします。このソフトウェアは、HTTP または HTTPS のどちらかのプロトコルを使用して、LAN または WAN 経由でワイヤレスネットワークに接続されているモバイルデバイスをサポートできます。実際に、Portal Server Mobile Access ソフトウェアは、オートモバイル、セットトップボックス、PDA、携帯電話、音声など任意の数のデバイスをサポートできます。

第 6 章

本稼働環境

この章では、Sun Java System Portal Server Secure Remote Access 製品を含む Sun Java™ System Portal Server ソフトウェアを監視およびチューニングする方法について説明します。

この章で説明する内容は次のとおりです。

本稼働環境への移行

本稼働環境への移行は、ポータルをよくテストしてから、試験的な配備で稼働させて設計をテストおよび改良したあとに行います。

監視とチューニング

ポータル配備の監視とチューニングは、常に定期的に実行される処理であり、この処理で障害やパフォーマンスに関するその他の問題を検出します。

ポータルを監視およびチューニングする場合、次の点を注意してください。

- 試験的なポータルから始めて、配備のパフォーマンスの基準を定義してから、プロジェクトの複雑な部分を追加していきます。
- この最初のベンチマークを使用して、短期および長期的に組織がサポートするトランザクションの量を定義します。
- 現在の物理的なインフラストラクチャーが、定義したトランザクションの量の要件をサポート可能であるかどうかを確認します。ポータルに対するアクティビティーが増えるにつれ、最初に限度に達するサービスを特定します。これにより、上限までの余裕が示され、また強化すべきところが特定されます。

- モデルを検証するためにトラフィックを定期的に測定および監視します。
- モデルを使用して、長期のシナリオを計画します。今後数年の総合的な成長予測に対応するために、配備をどのように大幅に変更する必要があるかを理解します。
- 本稼働システムでは、エラーロギングレベルを MESSAGE ではなく ERROR に保ちます。MESSAGE エラーレベルは冗長であり、ファイルシステムのディスク領域がすぐに不足する原因になります。ERROR レベルは、すべてのエラー状態と例外をログに記録します。

ポータルの文書化

ポータルの機能の包括的な文書は、システムをサポートしやすくする重要な手段です。サポート可能なソリューションを作成するために文書化する必要がある領域を次に示します。

- システムのアーキテクチャー
- ソフトウェアのインストールと設定
- 操作手順、「運用書」ともいう
- ソフトウェアのカスタマイズ
- カスタムコード
- サードパーティー製品の統合

運用書には、障害追跡の方法や配備のライフサイクルが要約されています。プロジェクトのトレーニングおよび知識の移譲段階でこのブックを利用できるようにします。

ヒント 通常、配備プロジェクトでは時間も費用も足りなくなるため、配備プロジェクトの終わりまで待たずに文書化を開始してください。ポータルの文書化は、配備過程全体を通して行う必要がある活動です。

Portal Server の監視

ここでは、ポータルのパフォーマンスに影響する可変要素、また実行可能なポータルの監視について説明します。監視対象には次のものが含まれます。

- Sun Java System Access Manager
- ポータルデスクトップ
- Sun Java System Directory Server
- Java 仮想マシン

次々と新しくなる技術によって Portal Server サービスの詳細な監視を実行できるようになりますが、ここではポータル配備の全体的なパフォーマンスを決定する基本的かつ広範なハードウェアおよびソフトウェアに焦点を合わせます。

特に、ポータルのパフォーマンスは、一定の期間にわたるスループットおよび応答時間の性能によって決まります。できるだけ早くパフォーマンスの基準の分析を行う必要があります。パフォーマンスの基準の分析では、ポータルが公開されたパフォーマンスの数値に実際に適合していることを確認します。パフォーマンスの基準の設定は、本稼働ポータルのパフォーマンスに重大な影響を及ぼすことがあるインフラストラクチャーの問題を理解するのに役立ちます。

それでもやはり、ポータルを継続して適切に稼働させるには、広範な問題を考慮する必要があります。次に、ポータルのパフォーマンスにおける可変要素の観点から問題について説明し、ポータルの効率を判断する際の指針を示します。

注 それらの規則は、パフォーマンス、スケーラビリティー、および負荷テストにも適用されます。

メモリーの消費とガベージコレクション

このセクションを読む前に、Java 仮想マシン、バージョン 1.4.2 でのガベージコレクションのチューニングについての次のドキュメントを読んでください。

`http://java.sun.com/docs/hotspot/gc1.4.2/index.html`

Portal Server では、可能なかぎり最高のスループットを実現するために、かなりの量のメモリーが必要になります。初期化時に、最大アドレス領域が実質的に確保されますが、必要でないかぎり物理メモリーは割り当てられません。オブジェクトメモリー用に確保したアドレス領域全体は新世代と旧世代に分割できます。

ほとんどのアプリケーションでは新世代用にヒープ全体のかなりの割合を使用するように勧めていますが、Portal Server では、Portal Server が使用するメモリーの大半は長期的に使用されるので、新世代用の領域の 1/8 のみを使用するのが適切です。メモリーを旧世代にコピーするのが早いほど、ガベージコレクション (Garbage Collection、GC) のパフォーマンスは向上します。

ヒープのサイズが大きい場合でも、ポータルインスタンスが中程度の負荷で数日間実行されたあとは、GC の遅れによりほとんどのヒープが使用されたように見えます。GC は、常駐セットサイズ (Resident Set Size、RSS) がヒープ領域全体の約 85 パーセントに達するまで完全なガベージコレクションを実行します。85 パーセントに達すると、ガベージコレクションがパフォーマンスにある程度の影響を及ぼすことがあります。

たとえば、900MHz UltraSPARCIII™ では、2G バイトのヒープに対するフル GC には 10 秒を超えることがあります。その間、システムは Web 要求に応答できません。信頼性のテスト時に、フル GC は応答時間の急激な上昇としてはっきり目に見えるようになります。フル GC のパフォーマンスに対する影響と頻度を理解する必要があります。本稼働時には、ほとんどの場合フル GC が認識されることはありませんが、システムのパフォーマンスを測定する監視スクリプトはフル GC が発生する可能性があることを考慮する必要があります。

フル GC の頻度を測定するのが、システムにメモリーリークが発生していることを確認する唯一の手段である場合があります。(基準システムの) 予測頻度を示す分析を行い、その結果を観測したフル GC の頻度と比較します。GC の頻度を記録するには、`vebose:gc` JVM™ パラメータを使用します。

CPU の使用率

構築モジュール概念 (第 5 章「ポータルの設計」 で説明) を使用して配備する場合、Portal Server のアーキテクチャーは高性能でスケーラブルな CPU アーキテクチャーになりますが、このアーキテクチャーでは高負荷時にパフォーマンスが徐々に低下します。

ただし、本稼働サイトを監視する場合は、CPU の使用率を一定の期間追跡します。通常、負荷は急に上昇し、上昇が発生する前に対策を講じるためには利用可能な機能の慎重な評価が必要です。

ほとんどの組織はポータルサイトが「スティッキー」な性質を持つことを認識しています。つまり、ユーザーのコミュニティーのサイズが固定されていても、ユーザーがそのサイトに慣れてくると、サイトの使用率が時間の経過とともに増加することを意味します。また、ユーザーのコミュニティーのサイズが時間の経過とともに増加する場合も、成功しているポータルサイトでは短期間に CPU 要件が高まる場合があります。

ポータルサーバーの CPU の使用率を監視する場合、負荷のピーク時における平均ページ待ち時間を確認し、それが平均の待ち時間とどれだけ異なるかを確認します。

ピーク時の負荷は、短期間ではありますが、平均の負荷の 4 ～ 8 倍の負荷であると予測します。

Access Manager のキャッシュとセッション

ポータルシステムのパフォーマンスは、Access Manager キャッシュのキャッシュヒット率の影響をかなり受けます。このキャッシュは高度にチューニング可能ですが、このキャッシュが使用するメモリーとヒープの残りの利用できるメモリーのどちらかを選択する必要があります。

amSSO および amSDKStats ログを有効にして、サーバー上のアクティブなセッションの数と Directory Server キャッシュの効率を監視できます。それらのログは、デフォルトで /var/opt/SUNWam/debug ディレクトリにあります。ロギング間隔を設定するには、com.iplanet.am.stats.interval パラメータを使用します。5 秒未満の値を使用しないでください。30 ～ 60 秒の値を使用すると、パフォーマンスに影響を与えずに良い結果が得られます。

com.iplanet.services.stats.directory パラメータを使用して、ファイルまたはコンソールのどちらかのログの場所を指定し、またログを無効にします。変更を有効にするには、サーバーを再起動する必要があります。ログは、システムがアクティビティーを検出するまで、作成されません。

注 複数の Web コンテナインスタンスが、同じファイルにログを書き込みます。

amSDKStats ファイルに表示されるキャッシュヒット率は、サーバーが起動されてからの内部の値と全体的な値の両方を示します。ユーザーがログインすると、ユーザーのセッション情報はキャッシュに無期限に、またはキャッシュが一杯になるまで残ります。キャッシュが一杯になると、もっとも古いエントリが先に削除されます。サーバーがユーザーのエントリを削除する必要がない場合は、数日後にログインするときに、ユーザーの情報がキャッシュから取得されることがあります。ヒット率が高いと、パフォーマンスがかなり向上します。最低 80 パーセントのヒット率を目標にします。この値は良い目標ですが、可能であればそれよりも高いヒット率を目標にすることが望まれます。

スレッドの使用

Web コンテナツールを使用して、要求を処理するために使用するスレッドの数を監視します。一般に、実際に使用されるスレッドの数は多くの場合予測した数よりも少なく、特に CPU の使用率が通常 100 パーセントよりもかなり低い本稼働サイトではそのようになります。

ポータルの使用情報

Portal Server には、ポータルのユーザーがポータルの使用情報を監視するための組込みの報告メカニズムがありません。これには、アクセスされるチャネル、チャネルがアクセスされた期間、またポータルのユーザーの行動様式を作成する能力が含まれます。ただし、Portal Server Desktop の要求を代行受信し、SSO トークンを抽出し、ユーザークセス情報をログに保存してから、ユーザーを目的の URL にリダイレクトする Java サーブレットを作成することはできます。そのような構成は、Access Mangaer スキーマへのカスタム属性の拡張に基づきます。

インストールされた製品のレイアウト

この付録では、設定および操作データの格納に使用する Sun Java™ System Portal Server ディレクトリ構造およびプロパティーファイルについて説明します。

Portal Server 用にインストールされるディレクトリ

表 A-1 は、Sun Java System Portal Server 用にインストールされるプラットフォーム固有のディレクトリ構造を示しています。

表 A-1 Portal Server のディレクトリ

説明	場所
デフォルトのインストールディレクトリ	*portal-server-install-root*/SUNWps
設定情報用のデフォルトのインストールディレクトリ	/etc/*portal-server-install-root*/SUNWps
SDK 用のデフォルトのインストールディレクトリ	*portal-server-install-root*/SUNWps/sdk
一時ファイル	/usr/tmp
デバッグファイル	/var/*portal-server-install-root*/SUNWam/debug
ログファイル	/var/*portal-server-install-root*/SUNWam/log
	/var/*portal-server-install-root*SUNWps*instance-directory*
検索エンジンのロギング、設定、およびデータディレクトリ	/var/*portal-server-install-root*/SUNWps/*instance-directory*/*log-directory*

表 A-1　Portal Server のディレクトリ (続き)

説明	場所
コンテナおよびチャネルディスプレイプロファイル	*portal-server-install-root*/SUNWps/samples/desktop/dp-org.xml
プロバイダディスプレイプロファイル	*portal-server-install-root*/SUNWps/samples/desktop/dp-providers.xml
HTML テンプレートファイル	/etc/*portal-server-install-root*/SUNWps/desktop/default/*channelname*.template
JSP テンプレートファイル	/etc/*portal-server-install-root*/SUNWps/desktop/default/*JSPchannelname*
コマンド行ユーティリティー	*portal-server-install-root*/SUNWps/bin/
タグライブラリ定義	/etc/*portal-server-install-root*/SUNWps/desktop/default/tld/*.tld
ディスプレイプロファイル DTD	*portal-server-install-root*/SUNWps/dtd/psdp.dtd
Java プロパティファイル	*portal-server-install-root*/SUNWam/locale

SRA 用にインストールされるディレクトリ

この付録では、設定および操作データの格納に使用する Sun Java™ System Secure Remote Access (SRA) ディレクトリ構造および設定ファイルについて説明します。

表 A-2 は、Secure Remote Access 用にインストールされるプラットフォーム固有のディレクトリ構造を示しています。

表 A-2　Portal Server、SRA ディレクトリ

説明	場所
デフォルトのインストールディレクトリ	*portal-server-install-root*/
Access Manager の実行可能ファイル、Web サーバー、および配備されるアプリケーション用のデフォルトのインストールディレクトリ	*portal-server-install-root*/SUNWam
設定情報用のデフォルトのインストールディレクトリ	/etc/*portal-server-install-root*/SUNWps
ログファイル	/var/*portal-server-install-root*/SUNWam/logs
デバッグログファイル	/var/*portal-server-install-root*/SUNWps/debug

設定ファイル

Portal Server および SRA のすべての設定データは、Sun Java System Access Manager Services Management 機能を使用して格納されます。Access Manager は、Sun Java System Directory Server を検索するために必要なブートストラップ設定ファイルを提供します。

`platform.conf` ファイルには、ゲートウェイが必要とする詳細情報が収められています。デフォルトでは、`platform.conf` ファイルは次の場所にあります。

`/etc/opt/SUNWps`

付録 B

分析ツール

Sun Java™ Enterprise System および SDK には、インストール後すぐに満足のいく体験ができるようにデフォルトの設定オプションが用意されています。ただし、デフォルトのオプションが、Sun Java System Portal Server 本稼働環境の Web アプリケーションのパフォーマンスを最高にするとはかぎりません。ここでは、いくつかの代わりのオプションと基本的なチューニング技術について説明します。

注 ここで説明するチューニング設定は、Solaris プラットフォームに存在する Portal Server に焦点を合わせています。ただし、この原則はその他の一般的な Unix タイプのオペレーティングシステムに適用できます。

次の表 B-1 は、Portal Server とその Web コンテナのチューニングのためのフィードバックの提供に役立つパフォーマンス分析ツールを示しています。パフォーマンスの問題以外にも、それらのツールの多くは、オペレーティングシステムレベル全体でのその他のタイプの障害を検出するのに使用できます。

多くのツールの説明には、出力の例、出力結果の解釈の仕方についての提案、出力結果の改善方法についてのヒント、および関連サイトへのリンクが含まれています。

表 B-1 パフォーマンス分析ツール

カテゴリ	タイプ	名前	パラメータ	用途
分析ツール	Solaris 8 と Solaris 9	mpstat		CPU の使用率
		iostat		ディスク入出力サブシステム
		netstat		ネットワークサブシステム
			-I hme) 10	インタフェース帯域幅

表 B-1　パフォーマンス分析ツール (続き)

カテゴリ	タイプ	名前	パラメータ	用途
			-sP tcp	TCP カーネルモジュール
			-a | grep *hostname* | wc-1	ソケット接続カウント
	アプリケーションサーバーコンテナ上の Portal Server	verbose:gc		ガーベジコレクション
チューニングパラメータ	Solaris 8 と Solaris 9	/etc/system	各種	パフォーマンス
		/etc/rc2.d/t チューニングパラメータファイル	各種	TCP カーネルチューニングパラメータ

mpstat

mpstat ユーティリティーは、CPU の使用率を監視するのに役立つツールです。特に、企業ソリューションによくある構成の、マルチプロセッサマシンで実行されるマルチスレッドアプリケーションの CPU の使用率の監視に役立ちます。

mpstat には、5 ～ 10 秒の引数を使用します。

5 ～ 10 秒よりも短い期間を分析するのは難しくなります。期間を長くすると、誤った結果につながる急激な値の上昇をなくしてデータを滑らかにすることができる場合があります。

出力

```
#mpstat 10

 CPU minf mjf xcal  intr ithr  csw icsw migr smtx  srw syscl  usr sys  wt idl
  0    1   0 5529   442  302  419  166   12  196    0   775   95   5   0   0
  1    1   0  220   237  100  383  161   41   95    0   450   96   4   0   0
  4    0   0   27   192  100  178   94   38   44    0   100   99   1   0   0
```

注意点

特定の CPU では intr および ithr の値がかなり大きいことに注意してください。Solaris はシステムの割り込みを処理するために CPU をいくつか選択します。選択される CPU および数は、システムに接続された入出力デイバス、それらのデバイスの物理的な場所、また CPU に対して割り込みが禁止されている (psradmin コマンド) かどうかによって決定されます。

- intr - 割り込み
- intr - スレッド割り込み (クロック割り込みは含まない)
  - csw - 任意コンテキストスイッチ。この数値が徐々に増加し、アプリケーションに入出力制約がない場合、相互排他競合を示す場合があります。
  - icsw - 強制コンテキストスイッチ。この数値が 500 を超える場合は、システムの負荷が高いことを示します。
  - smtx - smtx が急激に増加する場合。50 ～ 500 までの増加は、システムリソース (ネットワークやディスクなど) の障害の兆候を示します。
  - Usr、sys、および idl - これらの 3 つの列は、CPU の飽和を表します。最大の負荷がかかった状態 (0% アイドル) におけるよくチューニングされたアプリケーションは、usr が 80 ～ 90% 内に、sys が 20 ～ 10% に収まる必要があります。sys のパーセント値が小さいのは、ユーザーコードにより多くの時間が使用され、プリエンプションが少ないことを反映し、これは Portal アプリケーションのスループットを向上します。

考慮点

アプリケーションが、効率的に使用できるだけの数の CPU を利用できるようにします。たとえば、1 つのインスタンスに 2 つの CPU を使用するとパフォーマンスが最高になります。14 個の 2 CPU プロセッサセットを作成すると、パフォーマンスが最高になると予想できます。

csw 値の増加は、ネットワークの使用の増加を示します。一般的に、csw 値が高くなるのは、接続をプーリングしないため、または新しい接続を非効率的に扱うために、ソケット接続が多数作成された結果が原因です。この場合は、netstat -a | wc -l を実行すると、TCP 接続の数も多いことがわかります。netstat のセクションを参照してください。

icsw の増加がみられるなら、その一般的な原因はプリエンプションです。多くの場合、プリエンプションは CPU のタイムスライスが終了したことによって生じます。

iostat

iostat ツールは、ディスク入出力サブシステムの統計情報を提供します。iostat コマンドには、多数のオプションがあります。詳細は、マニュアルページを参照してください。次に示す典型的なオプションは、入出力の障害を特定するための情報を提供します。

出力

```
#iostat -xn 10
                    extended device statistics
    r/s    w/s   kr/s   kw/s wait actv wsvc_t asvc_t  %w  %b device
    0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0   0   0 fd0
    2.7   58.2   14.6 2507.0  0.0  1.4    0.0   23.0   0  52 d0
   47.3    0.0 2465.6    0.0  0.0  0.4    0.0    8.8   0  30 d1
```

注意点

- %b - このディスクが使用中である (進行中のトランザクションの) 割合。25 を超える平均 %b 値は障害を示している可能性があります。
- %w - トランザクションがサービスを待つ (キューが空でない) 時間の割合。
- asvc_t - アクティブなトランザクションのミリ秒単位の平均応答時間を報告します。このオプションには、誤って asvc_t という名前が付けられています。これは、ユーザープロセスが読み取りを発行してから、読み取りが完了するまでの時間を示します。値が常に 30ms を超える場合は、障害を示している可能性があります。

考慮点

ファイルシステムにさらにディスクを追加します。単一ディスクファイルシステムを使用している場合は、ハードウェアまたはソフトウェアの RAID へのアップグレードが次にとるべき手段です。ハードウェアの RAID は、ソフトウェアの RAID よりもかなり高速であり、強くお勧めします。ソフトウェアの RAID ソリューションは、システムの CPU にさらに負荷をかけます。

ストレージのハードウェアまたはソフトウェアの動作によっては、ufs のデフォルト値の 8192K バイト以外のブロックサイズを使用するほうがよい場合があります。『Solaris System 管理ガイド』を参照してください。

netstat

netstat ツールは、ネットワークサブシステムの統計情報を提供します。このツールは、ネットワークサブシステムのさまざまな側面を分析するのに使用できます。分析対象には TCP/IP カーネルモジュールとインタフェースの帯域幅の 2 つが含まれます。両方に対する分析の概要を次に示します。

netstat -I hme0 10

これらの netstat オプションは、インタフェースの帯域幅の分析に使用されます。現在のスループットの上限 (最大) は、出力から算出できます。netstat の出力はパケットのメトリックスを報告しますが、これは必ずしもパケットの最大サイズである必要はないので、上限が報告されます。帯域幅の上限は、次の式で求めることができます。

使用される帯域幅 = (パケットの総数) / (ポーリング間隔 (10)) * MTU (1500 デフォルト)。

インタフェースの現在の MTU は、ifconfig -a で確認できます。

```
netstat -I hme0 10 Output

#netstat -I hme0 10

 input   hme0       output            input  (Total)    output

packets errs  packets errs  colls  packets errs  packets errs  colls

122004816 272   159722061 0     0      348585818 2582  440541305 2
2

0       0     0       0     0      84144   0    107695  0     0

0       0     0       0     0      96144   0    123734  0     0

0       0     0       0     0      89373   0    114906  0     0

0       0     0       0     0      84568   0    108759  0     0

0       0     0       0     0      84720   0    108800  0     0
```

注意点

- colls- 衝突。ネットワークが交換ネットワークでない場合、低レベルの衝突が発生する可能性があります。ネットワークの飽和状態が増すと、衝突が増加し、最終的には障害になります。衝突の最善の解決策は、交換ネットワークです。
- errs - エラー。エラーの存在は、デバイスエラーを示す可能性があります。使用しているネットワークが交換ネットワークである場合、エラーはネットワークの帯域幅がほとんど使い果たされていることを示します。この問題の解決策は、システムの帯域幅を広くすることです。このためにはネットワークインタフェースを追加するか、またはネットワークの帯域幅をアップグレードします。これは、使用している特定のネットワークアーキテクチャーに大きく依存します。

考慮点

- ネットワークの飽和が早く発生する (100M ビットの Ethernet で稼働するアプリケーションサーバーの場合 8 個よりも少ない CPU で飽和) 場合、ネットワークの無駄な使い方をしないようにするための調査が最初にとるべき良い手段です。
- ネットワークの帯域幅を広げます。次の手段をとることができます。交換ネットワークへのアップグレード、ネットワークインタフェースの追加、またはネットワークトラフィックの負荷に対応できるより広い帯域幅へのアップグレード。

これらの netstat オプションは、TCP カーネルモジュールの分析に使用されます。報告されるフィールドの多くは、障害を示す、カーネルモジュール内のフィールドを表します。それらの障害には、ndd コマンドおよび /etc/inet で参照されるチューニングパラメータを使用して対処できます。

netstat -sP tcp の出力

```
#netstat -sP tcp


TCP    tcpRtoAlgorithm    =     4    tcpRtoMin           =   400


        <snip>


        tcpInDupSegs        =  1144    tcpInDupBytes       =132520
        tcpInPartDupSegs    =     1    tcpInPartDupBytes   =   416
        tcpInPastWinSegs    =     0    tcpInPastWinBytes   =     0
        tcpInWinProbe       =    46    tcpInWinUpdate      =    48
        tcpInClosed         =   251    tcpRttNoUpdate      =   344
        tcpRttUpdate       =1105386    tcpTimRetrans       =   989
        tcpTimRetransDrop   =     5    tcpTimKeepalive     =   818
        tcpTimKeepaliveProbe=   183    tcpTimKeepaliveDrop =     0
        tcpListenDrop       =     0    tcpListenDropQ0     =     0
        tcpHalfOpenDrop     =     0    tcpOutSackRetrans   =    56
```

注意点

- tcpListenDrop - このコマンドの出力を何回か観察したあとも tcpListenDrop が増加し続ける場合は、キューのサイズに関する問題を示す可能性があります。

考慮点

- tcpListenDrop の増加は、実行中のスレッドの数がアプリケーションのスループットの障害になっていることが原因である可能性があります。この段階では、アプリケーションスレッドの数を増やしてみると良い場合があります。
- キューのサイズを大きくします。ndd を使用して要求キューのサイズを増加します。その他の ndd コマンドの詳細は、『Solaris 管理ガイド』を参照してください。

```
ondd -set /dev/tcp tcp_conn_req_max_q <value>

ondd -set /dev/tcp tcp_conn_req_max_q0 <value>

netstat -a | grep <your_hostname> | wc -l
```

このコマンドを実行すると、システムのソケット接続のおおよその数がわかります。同時に開くことのできる接続の数は制限されています。このツールを使用して障害を調べることができます。

```
netstat -a | grep <your_hostname> | wc -l Output

#netstat -a | wc -l

34567
```

注意点

- socket count - 戻される値が 20,000 よりも大きい場合は、ソケット接続の数が障害である可能性があります。

次の点を考慮します。

- 匿名ソケット接続が開始される箇所の数を減らします。

```
ondd -set /dev/tcp tcp_smallest_anon_port <value>
```

- TCP 接続が TIME_WAIT になる時間を短縮します。

```
ondd -set /dev/tcp tcp_time_wait_interval <value>
```

/etc/system のチューニングパラメータ

表 B-2 は、パフォーマンスの調査時に使用される /etc/system のチューニングパラメータを示しています。変更は、それぞれを /etc/system ファイルに追加することによって適用されます。

表 B-2　/etc/system オプション

/etc/system オプション	説明
set rlim_fd_max=<value>	1 つのプロセスが開くことがある、ファイル記述子の「強力な」制限。この制限を無効にするには、スーパーユーザー権限が必要です。
set tcp:tcp_conn_hash_size=<value>	すべての TCP 接続用の TCP モジュール内のハッシュテーブルのサイズを制御します。
	tune_t_flushr とともに、autoup は各呼び出しでダーティーページの検索対象になるメモリーの量とファイルシステムの同期操作の頻度を制御します。
set autoup=<value>	autoup の値は、バッファーを空きリストから書き出すかどうかを制御するためにも使用されます。B_DELWRI フラグ (変更されたファイルコンテンツページ) でマークされたバッファーは、autoup 秒よりも長い間、空きリストにあると書き出されます。
	autoup の値を大きくすると、バッファーがメモリーにある時間をさらに長くできます。
set tune_t_fsflushr=<value>	fsflush 呼び出しの間隔を秒数で指定します。
set rechoose_interval=<value>	プロセスが最後に実行された CPU とのすべての関係がなくなったとみなされる前のクロック刻み数。この期間が過ぎると、どの CPU もスレッドをスケジュールするための候補とみなされます。このパラメータは、タイムシェアリングクラスのスレッドにだけ関係します。リアルタイムスレッドは、最初に利用できる CPU にスケジュールされます。

すべての /etc/system パラメータの説明は、『Solaris Tunable Parameters Reference Manual』を参照してください。

表 B-3 は、TCP カーネルチューニングパラメータの一覧です。それらは、Portal Server のほとんどのパフォーマンスに影響することがわかっている TCP チューニングパラメータです。それらのパラメータの推奨値は、『Identity Server Customization and API Guide』を参照してください。

表 B-3　TCP/IP オプション

TCP/IP オプション	説明
ndd -set /dev/tcp tcp_xmit_hiwat 65535 ndd -set /dev/tcp tcp_recv_hiwat 65535	バイト単位のデフォルトの送信ウィンドウサイズ。バイト単位のデフォルトの受信ウィンドウサイズ。
ndd -set /dev/tcp tcp_cwnd_max 65535	バイト単位の TCP 輻輳ウィンドウ (cwnd) の最大値。
ndd -set /dev/tcp tcp_rexmit_interval_min 3000	ミリ秒単位のデフォルトの最小再転送タイムアウト (Retransmission Timeout、RTO) 値。どの TCP 接続の計算して求めた RTO も、この値以上である必要があります。
ndd -set /dev/tcp tcp_rexmit_interval_max 10000	ミリ秒単位のデフォルトの最大再転送タイムアウト (Retransmission Timeout、RTO) 値。どの TCP 接続の計算して求めた RTO も、この値以下である必要があります。
ndd -set /dev/tcp tcp_rexmit_interval_initial 3000	ミリ秒単位のデフォルトの初期再転送タイムアウト (Retransmission Timeout、RTO) 値。
ndd -set /dev/tcp tcp_time_wait_interval 60000	TCP 接続が TIME-WAIT 状態にあるミリ秒単位の時間。詳細は、RFC 1122, 4.2.2.13 を参照してください。
ndd -set /dev/tcp tcp_keepalive_interval 900000	TCP 接続が KEEP-ALIVE 状態にあるミリ秒単位の時間。詳細は、RFC 1122, 4.2.2.13 を参照してください。
ndd -set /dev/tcp tcp_conn_req_max_q <value>	accept(SOCKET) によって受け入れられるのを TCP リスナーが待機している保留中の TCP 接続のデフォルトの最大数。
ndd -set /dev/tcp tcp_conn_req_max_q0 <value>	TCP リスナーが待機している未完了 (3 方向ハンドシェークがまだ終わっていない) の保留中の TCP 接続のデフォルトの最大数。
ndd -set /dev/tcp tcp_ip_abort_interval <value>	TCP 3 方向ハンドシェークの詳細は、RFC 793 を参照してください。
ndd -set /dev/tcp tcp_ip_abort_interval <value>	TCP 接続に関するミリ秒単位のデフォルトの合計再転送値。特定の TCP 接続で、TCP が tcp_ip_abort_interval の間再転送を行なっているが、この期間にもう一方の終端から肯定応答を受信しない場合、TCP はこの接続を閉じます。

Portal Server とアプリケーションサーバー

この付録では、Sun Java™ System Portal Server 製品およびそのアプリケーションサーバーのサポートの概要を述べます。

この付録で説明する内容は次のとおりです。

Portal Server でのアプリケーションサーバーのサポートについて

Sun Java System Portal Server 製品は、Java™ Web Server ソフトウェアのほかに、Web アプリケーションコンテナとして使用する次のアプリケーションサーバーをサポートします。

- Sun Java System Application Server Enterprise Edition
- BEA WebLogic Server™ Server 8.1 SP 2
- IBM WebSphere® Application Server 5.1

注 Portal Server は Web アプリケーションコンテナの環境で実行され、このコンテナは配備によって Web サーバーまたは前述のアプリケーションサーバーのいずれかにできます。この付録では、Web アプリケーションコンテナがアプリケーションサーバーであると仮定しています。

Portal Server をアプリケーションサーバーで実行すると、次のことが可能になります。

- アプリケーションサーバープラットフォームからポータルプラットフォームを切り離す。これにより、組織に最適の Portal Server とアプリケーションサーバーの組み合わせを選択できる
- アプリケーションサーバーコンテナで実行される Enterprise JavaBeans™ アーキテクチャーおよびその他の J2EE™ 技術を呼び出す
- スケーラビリティーと高可用性を実現するアプリケーションサーバークラスタを使用する
- クラスタでセッションのフェイルオーバーを使用する (現在は BEA WebLogic Server™ および Sun Java System Application Server Enterprise Edition でのみ利用可能)

アプリケーションサーバークラスタ上の Portal Server

ここでは、Application Server Enterprise Edition ソフトウェア、BEA WebLogic Server™、および IBM WebSphere® Application Server がアプリケーションサーバークラスタをどのように管理するかを説明します。アプリケーションサーバークラスタは、各サーバーがホストするサービスへの共有アクセスを可能にするために協力するアプリケーションサーバーの粗結合グループです。クラスタは、スケーラビリティーを実現するために、リソース要求、リソースの高可用性、およびアプリケーションロジックのフェイルオーバーのバランスをとることを目標にします。Portal Server および Access Manager は純粋な Web アプリケーションではありません。これらのアプリケーションは、マシンに存在するローカルファイルと、ポータル、amserver、および amconsole の 3 つの Web アプリケーションから構成されています。この 3 つの Web アプリケーションは Web アプリケーションコンテナで実行され、このコンテナはアプリケーションサーバーの Web アプリケーションコンテナで実行されます。

Java Enterprise System はローカルファイルをインストールして設定し、ローカルアプリケーションサーバーを設定してから、3 つの WAR ファイルをローカル Web アプリケーションコンテナに配備します。WAR ファイルは、自己完結型のファイルではありません。WAR ファイルは、サービスを提供するためにマシン上のローカルファイルおよびディレクトリを利用します。

アプリケーションサーバークラスタは、異なるマシンでホストされる可能性がある、多くのアプリケーションサーバーインスタンスをグループ化する論理エンティティーです。純粋の Web アプリケーションは、アプリケーションサーバー固有の配備ツールを使用してクラスタに配備されます。クラスタに配備されると、Web アプリケーションはクラスタを構成するすべてのサーバーインスタンスに配備され、集中管理されます。

Portal Server はローカルアプリケーションとして、また Web アプリケーションとしての 2 つの性質があるため、次の手順を実行して Portal Server をアプリケーションサーバーにインストールします。

1. 同じ構成の設定を使用してすべてのマシンに Portal Server をインストールします。
2. 3 つの Web アプリケーション (portal、amserver、および amconsole) をクラスタに配備します。

次のセクションでは、Portal Server をアプリケーションサーバークラスタで実行可能にするのにはどのような意味があるのかについて説明します。

Application Server Enterprise Edition の概要

Sun Java System Application Server Enterprise Edition 8 は、エンタープライズアプリケーションを開発、配備、および管理するための堅牢な J2EE プラットフォームを提供しています。主要な機能としては、トランザクション管理、パフォーマンス、スケーラビリティー、セキュリティー、統合性などが挙げられます。この Application Server は、Web パブリッシングから企業規模のトランザクション処理までのサービスをサポートします。

この Application Server には、Platform Edition と Enterprise Edition が用意されています。Platform Edition は無償で配布され、ソフトウェア開発および部門レベルの本稼動環境を構築するために使用できます。Enterprise Edition は、ミッションクリティカルなサービスと大規模な本稼動環境向けに設計されており、ロードバランサプラグインとクラスタ管理によって水平方向のスケーラビリティーとサービスの継続性をサポートしています。また、Enterprise Edition は、高可用データベース (HADB: Highly Available Database) によってセッションの継続性もサポートしています。詳細は、次の Application Server Enterprise Edition のマニュアルを参照してください。

http://docs.sun.com/db/coll/ApplicationServer8_ee_04q4

BEA WebLogic Server Cluster の概要

BEA WebLogic Server™ 製品は、次の定義を使用します。

- **ドメイン** : 1 つの単位として管理される WebLogic Server リソースの相互関係のある集合。ドメインには、1 つ以上の WebLogic Server が含まれ、WebLogic Server クラスタが含まれる場合もあります。
- **管理サーバー** : 管理サービスを実行する WebLogic Server。管理サービスは、ドメイン全体の設定および監視の集中制御を可能にします。管理サーバーは、ドメインに対する管理操作を実行するためには、そのドメインで稼働している必要があります。
- **管理対象サーバー** : 複数の WebLogic Server があるドメインでは、1 つのみが管理サーバーであり、他のサーバーは管理対象のサーバーと呼ばれます。WebLogic の各管理対象サーバーは、それぞれの設定を起動時に管理サーバーから入手します。

詳細については、次のマニュアルを参照してください。

`http://edocs.beasys.com/wls/docs61/cluster/index.html`

次のコマンドで管理サーバーを起動します。

install_dir`/config/domain_name/startWeblogic.sh`

ローカルサーバーは、その設定を *install_dir*`/config/domain_name/config.xml` ファイルから取得します。管理対象サーバーを起動するには、次のコマンドを使用します。

install_dir`/config/domain_name/startManagedWebLogic.sh` *servername admin_server_url*

管理対象サーバーは、その設定を *install_dir*`/config/domain_name/config.xml` ローカルファイルから取得するのではなく、HTTP を使用して管理サーバーから取得します。

注 Portal Server を BEA WebLogic Server™ にインストールするためにサポートされているデフォルトの構成は、ドメインの管理サーバーでもある単一サーバーです。

BEA クラスタは同じドメイン内の、WebLogic コンソールでクラスタとして宣言された管理対象サーバーの集合です。Web アプリケーションを配備するときには、個々のサーバーの名前ではなく、クラスタの名前を使用します。配備後、Web アプリケーションは、クラスタ内のすべてのマシンに同じように配備されます。

BEA でのセッションのフェイルオーバーについては、次のマニュアルに説明があります。

`http://edocs.beasys.com/wls/docs61/cluster/servlet.html#1009453`

HTTP セッション状態のためにメモリー内レプリケーションを使用するには、次の前提条件を満たす必要があります。

- Portal Server が、メモリー内セッションレプリケーションで WebLogic Server クラスタの使用をサポートする。それらのクラスタの設定方法については、BEA のマニュアルを参照してください。『Java Enterprise System インストールガイド』には、BEA に付属する HttpClusterServlet を使用する、そのようなクラスタ用のロードバランサの構成が記載されています。BEA のマニュアルに記載されているその他のロードバランスのハードウェアおよびソフトウェアも同様に設定できます。
- セッションのデータは直列化可能である必要がある。
- セッションの状態を変更するのに、`setAttribute` を使用する。

BEA クラスタをインストールするには、クラスタに参加する各マシンに対する BEA ライセンスは特別な BEA クラスタライセンスである必要があります。BEA ライセンスの取得および HttpClusterServlet を使用した BEA クラスタの設定手順については、BEA のマニュアルを参照してください。

IBM WebSphere Application Server の概要

IBM WebSphere Application Server 製品は、次の定義を使用します。

- **管理ドメイン** : WebSphere 環境のさまざまなオブジェクトの設定が存在する論理空間。1 つの管理ドメイン内からアプリケーションサーバーを起動します。これがデフォルトのインストールです。
- **サーバーグループ** : サーバーグループは、アプリケーションサーバー設定のほぼ同一の追加コピーを作成するためのテンプレートです。これは BEA クラスタに相当します。
- **クローン** : 同じマシン、または異なるマシンにあるサーバーグループのコピー。クローンは、BEA の管理対象サーバーに相当します。

詳細は、次の IBM WebSphere Application Server のマニュアルを参照してください。

http://www-3.ibm.com/software/webservers/appserv/doc/v40/ae/infocenter/was/welcome.html

WebSphere Advanced Server は、データベースを備えているため、より強力なクラスタを実現可能にします。Advanced Server では、すべてのサーバーが設定情報用のデータベースを使用します。WebSphere 管理コンソール、Swing Java アプリケーション、またはコマンド行ユーティリティーの `XMLConfig` および `wscpthen` を使用してサーバーを管理できます。

ポータルの配備の障害追跡

この付録では、Sun Java™ System Portal Server ソフトウェアおよび Sun Java System Portal Server Secure Remote Access (SRA) ソフトウェアの問題を解決する方法について説明します。

この付録で説明する内容は次のとおりです。

障害追跡 Portal Server

ここには、Sun Java System Portal Server の障害追跡情報が記載されています。

UNIX プロセス

ポータルが適切に機能するには、次の root 所有プロセスが実行されていることを確認します。この出力を表示するには、ps コマンドを使用します。

Sun Java System Directory Server:

```
/ns-slapd -D /usr/ldap/slapd-server -i /usr/ldap/slapd-server/logs/pid
```

Sun Java System Access Manager:

identity-server-install-root`/SUNWam/bin/doUnix -c 8946`

Sun Java System Portal Server:

```
./uxwdog -d portal-server-install-root/SUNWam/servers/https-server/config
ns-httpd -d portal-server-install-root/SUNWam/servers/https-server/config
```

Admin Web Server (オプションであるが、通常は実行される):

```
./uxwdog -d web-container-install-root/SUNWam/servers/https-admserv/config
ns-httpd -d web-container-install-root/SUNWam/servers/https-admserv/config
```

ログファイル

次のログファイルでエラーを調べます。

Sun Java System Web Server (errors および access):

web-container-install-root/SUNWam/servers/https-*server*/logs

Sun Java System Directory Server:

/var/opt/SUNWam/logs

検索データベースの回復

検索データベースには、回復可能なトランザクションログが保持されます。したがって、正常の状態では、データベースを回復するために何もする必要はありません。エラー状態、またディスクが一杯などの一時的な状態からの回復は簡単です。必要に応じて、検索データベースのアーカイブを保持し、データベース全体を失った場合、アーカイブから復元します。このシナリオでは、アーカイブを元のデータベースにコピーして復元します。

➤ データベースを回復するには

1. Portal Server インスタンスを含む、データベースにアクセスするすべてのプロセスを停止します。
2. rdmgr -R コマンドを使用して回復します。

ディスプレイプロファイルの操作

ポータルのディスプレイプロファイルの XML コンテンツを障害追跡する必要がある場合は、調査のためにコンテンツをファイルから抽出します。障害追跡の過程のある時点で、ディスプレイプロファイルを再読み込みすると役立つ場合があります。

➤ ディスプレイプロファイルを抽出するには

1. 管理者としてログインします。
2. dpadmin コマンドを使用して、ディスプレイプロファイルを抽出します。次に例を示します。

```
./dpadmin list -u "uid=amAdmin,ou=People,o=sesta.com,o=isp" -w
password -d "o=sesta.com,o=isp" > /tmp/displayxml
```

この例では、ディスプレイプロファイルのコンテンツを /tmp/displayxml ファイルに格納します。

➤ ディスプレイプロファイルを再読み込みするには

1. 管理者としてログインします。
2. dpadmin コマンドを使用して、ディスプレイプロファイルを再読み込みします。次に例を示します。

```
./dpadmin modify -u "uid=amAdmin,ou=People,o=sesta.com,o=isp" -w
password -d "o=sesta.com,o=isp" /tmp/updated_displayxml
```

この例は、ディスプレイプロファイルのコンテンツを /tmp/updated_displayxml ファイルから再読み込みします。

Portal Server インスタンスの高 CPU 使用率

Cisco Content Services Switch を使用する場合、Portal Server インスタンスの CPU の使用率が非常に高くなる場合があり、その場合 5 秒おきに Sun Java System Web Server エラーファイルに次のメッセージが表示されます。

```
[20/Jan/2003:16:53:36] failure ( 5926): Error accepting connection -5928,
oserr=130 (Connect aborted)
```

このエラーの原因は、Cisco Content Services Switch 内の「スティッキービット」の設定です。このロードバランサは、サーバーが稼働していることを確認するためにサーバーを定期的 (5 秒おき) に ping します。「スティッキービット」の設定をオフにすると、サーバーに対する 5 秒おきの ping が禁止されるので、Web Server 製品でエラーは発生しません。

HTTP プロキシを使用するための Sun Java System Portal Server インスタンスの設定

Portal Server ソフトウェアが、インターネットまたはイントラネットの特定の部分に直接アクセスできないホストにインストールされている場合、エラーが表示されることがあります。たとえば、SampleSimpleWebService プロバイダを使用する場合、プロキシが設定されていないと、次のエラーが表示されることがあります。

```
java.net.UnknownHostException: services.xmethods.net
```

➤ Portal Server インスタンスに対する HTTP プロキシの使用を設定するには

1. ディレクトリを、インスタンスの設定が含まれるポータルサーバーのインストールルートディレクトリに変更します。

   `cd` *portal-server-install-root*`/SUNWam/servers/https-`*servername*`/config`

2. このディレクトリ内の server.xml ファイルを編集し、次の行を追加します。

   `http.proxyHost=`*proxy-host*

   `http.proxyPort=`*proxy-port*

   `http.nonProxyHosts=`*portal-host*

   各表記の意味は次のとおりです。*proxy-host* は、プロキシホストの完全修飾ドメイン名であり、*proxy-port* はプロキシが実行されるポート、*portal-host* はポータルホストの完全修飾ドメイン名です。

SRA の障害追跡

ここでは、Sun Java System のサポート担当者が配備の問題の原因を特定するために必要な情報を収集する方法を説明します。

ゲートウェイのデバッグ

デバッグをオンまたはオフにするには、デバッグのレベルを設定するか、またはデバッグをオフに設定します。次の手順は、実行方法を示しています。

1. ゲートウェイマシンに root としてログインし、次のファイルを編集します。

   gateway-install-root/SUNWam/config/AMConfig-*instance-name*.properties

2. デバッグレベルを設定します。

   com.iplanet.services.debug.level=

   次のデバッグレベルがあります。

   error : 重要なエラーだけがデバッグファイルに記録されます。このようなエラーが発生すると、通常、リライタは機能を停止します。

   warning : 警告メッセージが記録されます。

   message : すべてのデバッグメッセージが記録されます。

   off : デバッグメッセージは記録されません。

3. AMConfig-*instance-name*.properties ファイルの次のプロパティーに、デバッグファイルのディレクトリを指定します。

   com.iplanet.services.debug.directory=/var/opt/SUNWam/debug

   この /var/opt/SUNWam/debug は、デフォルトのデバッグディレクトリです。

4. 端末ウィンドウからゲートウェイを再起動します。

   gateway-install-root/SUNWps/bin/gateway -n *gateway-profile-name* start

shooter について

shooter ツールは、開発およびサポートチームが Sun Java System Portal Server Secure Remote Access 製品の配備に関する問題の原因を特定するために必要なすべての情報を収集します。このツールは Portal Server マシンでも実行できます。

このツールは、次のデータを収集します。

- インストールの種類 - Sun Java System Portal Server with Sun Java System Secure Remote Access コア、つまり SRA を備えた Portal Server がインストールされているかどうかを確認します。
- システムの設定に関連する情報 - ホスト、ドメイン、オペレーティングシステム、バージョン、CPU の種類と速度、クロック速度、および利用できるメモリーを確認します。
- プロセッサ、プロセッサセット、およびそれらにバインドされた SRA プロセス
- SRA インストールログ
- platform.conf ファイル
- ヒープの使用を含む JVM™ 設定や、ライブラリパスなどのゲートウェイスクリプトの設定
- ゲートウェイサービスの設定
- Sun Java System Access Manager、Sun Java System Directory Server、および Sun Java System Web Server の設定に使用するさまざまなファイルに含まれるチューニングの設定
- ガベージコレクションの出力
- ゲートウェイを使用していたときのメモリーまたはプロセスのフットプリント
- フォーマットされたデバッグログファイル
- リライタルールセット

注 このツールは、インストール時に指定したゲートウェイのインスタンスのみの情報を収集します。

shooter の使用

shooter ツールは、次に説明する 5 つのファイルから構成されています。

shooter.sh

これはメインスクリプトです。このスクリプトは、SRA インストールでテスト後に、またはテストを開始する直前に実行します。

portal-server-install-root/bin/perf から、次のように入力します。

```
./shooter.sh
```

このツールは、一時フォルダ下にデータを収集し、フォルダ名を表示します。

gctool.pl

このスクリプトは、JVM からガベージコレクションの出力を収集しフォーマットします。

gctool を実行するには、ゲートウェイを起動し、次のように入力して出力をこのスクリプトにリダイレクトし、テスト時に収集を可能にします。

```
/etc/init.d/Gateway -n default start | gctool.pl
```

注 gctool を実行する前に、ゲートウェイスクリプトの "CMD" セクションに -verbose:gc を含めるようにします。ゲートウェイスクリプトは次のようになります。

```
-server -verbose:gc -Xms1G -Xmx2G
-XX:+OverrideDefaultLibthread -XX:ThreadStackSize=128
-XX:MaxPermSize=128M -XX:PermSize=128M
-XX:MaxNewSize=256M -XX:NewSize=256M
```

テスト期間の終わりに、shooter を実行して、gctool の出力をその他のデータとともに収集します。

memfoot.sh

このスクリプトは、プロセスのメモリーフットプリントを追跡します。ゲートウェイの起動後にこのスクリプトを開始し、テスト期間中実行されるようにします。指定された名前または PID を持つ最大のプロセスが、指定された秒数ごとに追跡されます。

memfoot を実行するには、次のように入力します。

```
./memfoot java 60
```

このスクリプトの出力は、タイムスタンプ付きのプロセスステータスファイルです。shooter ツールは、残りのデータとともにこの出力を収集します。

uniq.pl

このスクリプトは、shooter が一意の行とその数を見つけるために内部で使用します。隣接していない一意の行を見つけるという点が、システム uniq スクリプトよりも優れています。

GWDump.class

このクラスは、Access Manager 管理コンソールでゲートウェイの設定を入手するために shooter によって内部で呼び出されます。

SRA ログファイル

次のログファイルでエラーを調べます。

ゲートウェイ :

/var/opt/SUNWps/debug/srapGateway_*Gateway-hostname_Gateway-profile-name*

NetFile:

/var/opt/SUNWps/debug/srapNetFile

Netlet:

/var/opt/SUNWps/debug/srapNetlet_*Gateway-hostname_Gateway-profile-name*

ポータル配備ワークシート

この付録は、ポータルの配備処理に役立つワークシートを提供します。

この付録で説明する内容は次のとおりです。

ポータル評価ワークシート

このワークシートを使用して、組織の業務のニーズとポータルの配備の問題になる可能性のある部分についてさらに学びます。

表 E-1 一般的な質問

1. ポータルが必要な業務上の理由を確認します。該当するものにすべてチェックマークを付け、詳しく検討します。

- 調達コストを削減する
- 顧客、サプライヤ、またはパートナーとの情報共有のコストを削減する
- 多数の個別目的のソリューションを維持するためのコストをなくす
- サービスの対象の顧客ベースを拡大する
- 新しい業務サービスの配備の時間を削減する
- データおよびサービスへのアクセスをセキュリティー保護する
- インターネットによる顧客との取り引きを容易にする
- 業務サービスのサプライヤおよびパートナーとの統合の費用および時間を削減する
- 政府の規制に従う
- ユーザーの体感をパーソナライズする
- サービスの使用についてのビジネスインテリジェンスを収集する必要がある

2. 組織にはいくつのポータルがすでにありますか。

3. それらのポータルの種類は、企業対社員、企業対顧客、企業間、ISP のどれですか。

4. 複数のポータルがある場合は、数を減らす、統合する、または連携する必要がありますか。

5. 部門ごとのポータルがありますか。

6. どの程度 Web に参加していますか。Web サイトをいくつ持っていますか。

7. Portal Server を使用してパートナーにアクセス可能にする、価値のある上位 10 個のアプリケーションサービスを挙げてください。サプライヤ、顧客、従業員にアクセス可能にするアプリケーションサービスについても同様に挙げてください。

8. ポータルの対象コミュニティーはどこですか。

表 E-2 組織に関する質問

1. このポータルの関係者はだれですか。
2. ポータルを使用して所有するコンテンツまたはアプリケーションサービスを公開する、組織内の業務情報の所有者 (部門、組織、または個人) はだれですか。
3. このポータルを使用して公開されるアプリケーションサービスは、部門間の業務プロセスによって管理されるさらに小規模なアプリケーションから構成されていますか。
4. このポータル (インフラストラクチャー) を「所有」するのはだれですか。
5. コンテンツを所有するのはだれですか。
6. ポータルにコンテンツやアプリケーションを提供するように、組織内の業務情報の所有者をどのようにさらに募集する計画ですか。
7. このポータルの開発の支援に、どのプロジェクト管理リソース、アーキテクトリソース、および技術実装リソースを利用できますか。
8. 見た目と使い心地やプレゼンテーションなどの Web サイトの特性のポリシーを設定するのは誰ですか。

表 E-3 ビジネスサービスレベルの期待に関する質問

1. 開発プロジェクトに一貫性がありますか。開発プロジェクトのリスク管理を行いますか。
2. 開発チームはどのようにテストグループ、配備グループ、および運用グループと連携しますか。
3. 組織は現在いくつのプラットフォームをサポートしていますか。
4. 情報はどの程度保護されていますか。セキュリティーはどの程度整合性がとれていますか。
5. それらの課題は改善されていますか、あるいは悪化していますか。

表 E-3　ビジネスサービスレベルの期待に関する質問 (続き)

6. ポータルにコンテンツやアプリケーションを提供するように、組織内の業務情報の所有者をどのようにさらに募集する計画ですか。

7. このポータルの開発の支援に、どのプロジェクト管理リソース、アーキテクトリソース、および技術実装リソースを利用できますか。

8. 見た目と使い心地やプレゼンテーションなどの Web サイトの特性のポリシーを設定するのは誰ですか。

表 E-4　コンテンツの管理に関する質問

1. コンテンツまたはドキュメント管理システムがありますか。

2. コンテンツの開発および公開を管理するためのワークフローを定義していますか。

3. 分類を定義していますか。

4. 情報にどの程度適切にタグが付けられ、分類されていますか。

5. 企業コンテンツはどのように開発、管理、追跡、および公開されますか。

6. ポータルにシンジケートコンテンツが必要ですか。そうである場合、それはどのようなものですか。

7. コンテンツの動的な部分と静的な部分の割合はどのようになっていますか。

表 E-5　ユーザーの管理とセキュリティーに関する質問

1. ユーザーコミュニティーをどのように区分、分離、および関係付け (階層的に) ますか。
2. 現在および将来のセキュリティーポリシーは何ですか。
3. さまざまな部門において非公開の顧客の情報を所有または管理していますか。
4. 企業ディレクトリがありますか。

表 E-6　ビジネスインテリジェンスに関する質問

1. 企業の意思決定のための情報の収集、保存、分析、および提供を行う必要がありますか。
2. データ分析ツールまたは OLAP ツールをすでに使用していますか。
3. どのレベル (企業全体、部門、部、プロジェクト、1 回だけのイベント) で業務情報を収集する必要がありますか。

表 E-7　アーキテクチャーに関する質問

1. すでにアーキテクチャーの方針を決めていますか。

- 新しいアーキテクチャーソリューションを実装する能力がありますか。
- 現在どのような技術を使用していますか。
- 新しいアーキテクチャーソリューションを実装する担当者がいますか。

2. 新しい IT アーキテクチャーの実現の成功を妨げるような組織の問題がありますか。
3. ポータルを使用して配備する必要がある上位 10 個のサービスに、どのプラットフォームとアーキテクチャーをサポートする必要がありますか。
4. それらのサービスはどのようにユーザーを認証し、アクセス制御を管理しますか。
5. どのようにしてそれらのサービスにプログラムでアクセスできますか。
6. 現在および将来のメッセージング (電子メール) およびコラボレーションアーキテクチャーはどのようなものですか。
7. 現在および将来の企業ディレクトリアーキテクチャーはどのようなものですか。
8. アプリケーションの統合には、どのような技術を使用しますか。
9. 対象のユーザーコミュニティーの規模はどの程度ですか。
10. 同時に使用するユーザー数はどの程度ですか。
11. ポータルの使用範囲はどの程度ですか。

表 E-7　アーキテクチャーに関する質問 (続き)

12. ユーザーベースの地理的分布はどのようになっていますか。

13. Web でないアクセス (ワイヤレス、音声 /IVR) が現在または将来必要ですか。

14. 顧客ベースがコンテンツおよびサービスの国際化を必要としますか。

15. どのようなサーバープラットフォーム技術を使用しますか。

16. どのような開発環境、開発ツールを使用しますか。

17. どのような開発方法論を採用しますか。

ポータル設計作業リスト

表 E-8 は、ポータルの主な開発段階と設計作業を示しています。この作業リストを使用して、ポータルのプロジェクトの計画を立てます。

作業は組織や各配備の規模によって異なりますが、ワークシートはもっとも一般的な段階と作業を示します。

この表は、2 列からなります。最初の列は、主な作業を示しています。2 列目は、主な各作業を構成する個々の作業を示しています。

表 E-8　設計作業のリスト (1 / 7)

主な段階と作業	個々の作業
1. プロジェクトの開始と調整	
プロジェクトの計画	• 一般的なプロジェクトの管理を実行します。
プロジェクトの計画の見直し	• 事前実装を見直します。 • 業務要件を見直します。 • 技術要件を見直します。 • アーキテクチャーの文書を見直します。 • ハードウェアおよびインフラストラクチャーを見直します。

表 E-8 設計作業のリスト（２ / ７）

主な段階と作業	個々の作業
リソースの調整	• 必要なスキルを確認します。 • リソースを確認します。 • リソースをスケジュールします。 • プロジェクトチームのメンバーを集めます。 • プロジェクトチームのメンバーと作業計画を見直します。
要件の定義	• 業務要件を収集します。 • 要件を要約します。 • 機能要件を確認します。 • 技術要件を収集します。 • 技術要件を要約します。 • 技術要件を確認します。 • 要件をまとめた文書を準備します。 • 要件を伝えます。
2. 設計	
ソリューションアーキテクチャーの開発	• ソフトウェアのアーキテクチャーを設計します。 • サーバーのトポロジを設計します。 • アーキテクチャーの文書を作成します。
ポータルの統合方法の開発	• システムの統合について理解します。 • コンテナとチャネルレイアウトを定義します。 • コンテンツの集約を定義します。 • SSO の方法を定義します。 • カスタム Netlet モジュールおよびカスタム認証モジュールを開発します。
ユーザーインタフェースの設計	• ユーザーインタフェースの設計を準備または変更します。 • 画面の仕様を開発または更新します。 • ユーザーインタフェースモデルを見直し、承認します。

表 E-8　設計作業のリスト (3 / 7)

主な段階と作業	個々の作業
ディレクトリの設計	• 組織、サブ組織、ロール、およびユーザーを設計します。 • アクセス権限を定義します。 • 共有データの要件を見直します。 • データ転送プロトコルを設定します。 • 一時表または中間表を作成します。 • 一時表または中間表をテストします。 • 設計方法の文書を作成します。 • 設計の文書を配布します。 • 該当する関係者および組織の承諾を得ます。
3. 開発および統合	
テスト環境および開発環境へのソフトウェアのインストール	• Sun Java System Portal Server ソフトウェア、またオプションとして Sun Java System Portal Server Secure Remote Access ソフトウェアをインストールします (適切なサポートソフトウェアをインストールする)。 • 必要に応じて、アプリケーションサーバーをインストールします。 • その他のソフトウェアをインストールします。 • サーバーソフトウェアを設定します。 • サーバーソフトウェアのコンポーネントをテストします。 • テスト結果の文書を作成します。
開発環境へのサーバーソフトウェアのインストール	• Portal Server、またオプションとして Sun Java System Portal Server Secure Remote Access をインストールします。 • 必要に応じて、アプリケーションサーバーをインストールします。 • その他のソフトウェアをインストールします。 • サーバーソフトウェアのコンポーネントをテストします。 • テスト結果の文書を作成します。
ソフトウェアの設定	• 特定のソフトウェア設定要件を適用します。 • 製品設定マトリックスを作成します。

表 E-8 設計作業のリスト (4 / 7)

主な段階と作業	個々の作業
Sun Java System Portal Server、Sun Java System Application Server およびその他のソフトウェアの変更	• 組織の要件と期待を見直します。 • ソフトウェアの変更を定義します。 • ソフトウェアの変更の方法を定めます。 • ソフトウェアの変更計画を立てます。 • ソフトウェアの変更を設計します。 • ソフトウェアの変更チームを編成します。 • 変更を行います。 • 変更をテストします。 • 該当する関係者および組織に変更を見直してもらい承諾を得ます。
LDAP ディレクトリの設定	• 適切なスキーマを設定するために関係者と話し合います。 • ソフトウェアの変更を定義します。 • ソフトウェアの変更の方法を定めます。 • ソフトウェアの変更計画を立てます。 • ソフトウェアの変更を設計します。 • ソフトウェアの変更チームを編成します。 • スキーマを作成します。 • LDAP を設定します。 • データを受け取り、検証します。 • LDAP に要求されるようにマッピングを変更します。 • データ更新方法を定めます。 • ディレクトリをテストします。 • 更新方法についてのクライアントユーザー用の文書を作成します。
旧バージョンのソフトウェアの統合 (PeopleSoft、SAP など)	• 統合を行います。 • パッケージの統合テストの計画を準備します。 • 統合テストを実施します。 • パッケージの統合テストの結果を出します。

表 E-8　設計作業のリスト（５／７）

主な段階と作業	個々の作業
レポート	• 組織のレポートの要件を定めます。
	• レポートの計画を立てます。
	• レポートチームを編成します。
	• レポートを設計します。
	• レポートを作成します。
	• レポートをテストします。
	• レポートを顧客と見直します。
	• レポートツールの情報とトレーニングを提供します。
テスト	• テストの計画を立てます。
ユーザー受け入れテストの計画	• ユーザー受け入れテストのマネージャーを定めます。
	• ユーザー受け入れのテストの方針と手順を作成します。
	• 方針と手順を顧客と見直します。
	• 方針および手順の承諾を得ます。
	• ユーザー受け入れテストのロールと責任を定めます。
	• 統合テストのシナリオを入手します。
	• テスト条件と受け入れ基準を見直し、修正します。
	• ユーザー受け入れテストのスケジュールを立てます。
	• 受け入れテストのログを準備し、テストのシナリオの指定で更新します。
ユーザー受け入れテストの実施	• ユーザー受け入れテストを実施します。
	• ユーザー受け入れテストの矛盾を特定し、文書を作成します。
	• ユーザー受け入れテストの矛盾を解決します。
	• ユーザー受け入れテストを再度実行して、ユーザー受け入れテストの進捗状況を追跡します。
	• 既知の制限とテスト中に確認したプロセスの改善可能な部分を分類し、優先順位を割り当てます。
	• テストの結果を品質保証アドバイザと見直し、結果を要約して関係者に伝えます。
	• 関係者から受け入れテストの承認を得ます。

表 E-8 設計作業のリスト (6 / 7)

主な段階と作業	個々の作業
統合およびシステムのテストの実施	• 統合テスト環境を整備します。 • テストチームを指名し、テストのシナリオの所有権を割り当てます。 • チームに対して統合テスト手順、ロール、および責任のトレーニングを行います。 • 必要に応じて、統合テストの実施スケジュールを見直して修正します。 • 統合テストを実施します。 • 統合テストの矛盾を特定し文書を作成します。 • 統合テストの矛盾を解決し文書を作成します。 • 必要な変更 (設定の改善、インタフェース、レポートなど) を特定します。 • 統合テストを再度実施します。 • 必要に応じて更新します。 • テストの進捗状況を追跡します。 • テストの承認を得ます。 • 結果を要約し関係者に伝えます。
4. 配備の実施	
確認方法	• 実装の場所と構成を関係者と見直し、定めます。 • 実装の方法を定めます。 • 開発用ハードウェアおよびソフトウェアのインストール作業の中で該当するものを繰り返します。
配備の見直しと更新	• テスト結果の既存の文書を見直します。 • 範囲、目的、および重要な成功の要因を検証します。 • 配備の方法を更新します。 • 配備を見直し、承認します。

表 E-8 設計作業のリスト (7 / 7)

主な段階と作業	個々の作業
配備の実装	• システムのオペレーションを見直し、調整します。
	• 組織およびシステムの手順を見直します。
	• 本稼働に昇格させます。
	• 現在のオペレーションを更新します。
	• システムのリリースを改訂し、配備の資料も新しいものにします。
	• 移行を支援します。
トレーニング	• 組織の約束と期待を確認します。
	• すべての担当者のトレーニング要件を定めます。
	• トレーニングのスケジュールを立てます。
	• トレーニングの担当者を決定します。
	• トレーニングに必要なものを準備します。
	• 管理者をトレーニングします。
	• 保守プロバイダをトレーニングします。
	• トレーニングの参加者から意見を聞きます。
	• トレーニングの改善のために参加者の意見を反映させます。
ポータルの文書の作成	• システム管理者用の「運用書」を作成します。

Linux プラットフォームの Portal Server

Sun Java™ System Portal Server は RedHat 3.0 Linux プラットフォームをサポートしていますが、Solaris プラットフォームと Linax プラットフォームとの違いに注意する必要があります。

Linux の使用上の制限事項

次のことに注意してください。

- Portal Server と Access Manager は同じサーバーに存在する必要があります。
- サンプルのポータルは Linux プラットフォームをサポートしません。
- IBM および BEA の Web コンテナはサポートされていません。

設定ファイル、配備、およびアプリケーションプログラミングインタフェースについては、Solaris も Linux も同じです。

Solaris と Linux とのパス名の比較

表 F-1　Solaris と Linux とのパス名の比較

Solaris のパス名	Linux のパス名
/opt/SUNWps (デフォルト)	/opt/sun/portal (デフォルト)
/etc/opt/SUNWps (config)	/etc/opt/sun/portal (config)
/var/opt/SUNWps (データ)	/var/opt/sun/portal (データ)

用語集

このドキュメンテーションセットで使用されているすべての用語の一覧については、Java Enterprise System の用語集 (http://docs.sun.com/app/docs/doc/819-1933) を参照してください。

索引

記号

A

B

C

D

O

P

R

S

T

U

V

W

く

け

こ

さ

し

す

せ

そ

た

ち

つ

て

と

に

は

ひ

ふ

へ

ほ

ま

ゆ

よ

り

る

ろ

わ