



Sun Java™ System

Communications Services 6 企業向け配備計画ガイド

2004Q2

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

Part No: 817-7090

Copyright © 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Sun Microsystems, Inc. は、この製品に含まれるテクノロジーに関する知的所有権を保持しています。特に限定されることなく、これらの知的所有権は <http://www.sun.com/patents> に記載されている 1 つ以上の米国特許および米国およびその他の国における 1 つ以上の追加特許または特許出願中のものが含まれている場合があります。

このソフトウェアは SUN MICROSYSTEMS, INC. の機密情報と企業秘密を含んでいます。SUN MICROSYSTEMS, INC. の書面による許諾を受けることなく、このソフトウェアを使用、開示、複製することは禁じられています。

U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

この配布には、第三者が開発したソフトウェアが含まれている可能性があります。

本製品の一部は、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいていることがあります。UNIX は、X/Open Company, Ltd が独占的にライセンスしている米国およびその他の国における登録商標です。

Sun、Sun Microsystems、Sun のロゴマーク、Java、Solaris、JDK、Java Naming and Directory Interface、JavaMail、JavaHelp、J2SE、iPlanet、Duke のロゴマーク、Java Coffee Cup のロゴ、Solaris のロゴ、SunTone 認定ロゴマークおよび Sun ONE ロゴマークは、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標もしくは登録商標です。

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャに基づくものです。

Legato および Legato のロゴマークは Legato Systems, Inc. の商標であり、Legato NetWorker は同社の商標または登録商標です。

Netscape Communications Corp のロゴマークは Netscape Communications Corporation の商標または登録商標です。

OPEN LOOK および Sun Graphical User Interface は、米国 Sun Microsystems 社が自社のユーザおよびライセンス実施権者向けに開発しました。米国 Sun Microsystems 社は、コンピュータ産業用のビジュアルまたはグラフィカルユーザインタフェースの概念の研究開発における米国 Xerox 社の先駆者としての成果を認めるものです。米国 Sun Microsystems 社は米国 Xerox 社から Xerox Graphical User Interface の非独占的ライセンスを取得しており、このライセンスは米国 Sun Microsystems 社のライセンス実施権者にも適用されます。

この製品は、米国の輸出規制に関する法規の適用および管理下にあり、また、米国以外の国の輸出および輸入規制に関する法規の制限を受ける場合があります。核、ミサイル、生物化学兵器もしくは原子力船に関連した使用またはかかる使用者への提供は、直接的にも間接的にも、禁止されています。このソフトウェアを、米国の輸出禁止国へ輸出または再輸出すること、および米国輸出制限対象リスト (輸出が禁止されている個人リスト、特別に指定された国籍者リストを含む) に指定された、法人、または団体に輸出または再輸出することは一切禁止されています。

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、明示的であるか黙示的であるかを問わない、なんらの保証も行われぬものとします。

目次

図目次	7
表目次	9
はじめに	11
第 1 章 Communications Services について	15
Communications Services の概要	15
Messaging Server について	17
Calendar Server について	17
Instant Messaging について	18
同期について	18
Connector for Microsoft Outlook について	19
Communications Express について	19
Communications Services コンポーネント製品の依存性	20
Communications Services の企業のビジネスニーズへの対応方法について	20
Messaging Server のビジネスニーズへの対応方法について	21
Calendar Server のビジネスニーズへの対応方法について	21
Instant Messaging のビジネスニーズへの対応方法について	21
Communications Services の利点のまとめ	22
Communications Services 配備の高可用性の向上	23
Communications Services での Portal Server の使用	24
配備プロセスについて	25
配備アーキテクチャの設計	25
配備とカスタマイズ	26
プロトタイプとテスト	26
製品システムの展開	27

第 2 章 企業の要件の分析	29
配備目標の明確化	29
ビジネス要件	30
運用要件	30
企業文化と政治	30
技術要件	31
既存の使用パターンのサポート	31
サイトの分散	31
ネットワーク	31
既存のインフラストラクチャ	32
サポート要員	32
財務要件	32
サービス品質保証契約 (SLA)	33
プロジェクトの目標の特定	34
拡大のための計画	34
総所有コスト (TCO) の理解	34
第 3 章 製品の要件と考慮事項について	37
さまざまなコンポーネントの計画	38
サービスコンポーネントとサービス層の理解	39
LDAP ディレクトリ情報ツリーの要件	40
DIT 構造の変更	41
1 ツリー DIT 構造の利点	42
スキーマの要件	44
Directory Server の考慮事項	45
Directory Server と Tier (層) アーキテクチャの考慮事項	46
Directory Server トポロジの考慮事項	46
Directory Server の容量計画	46
Directory Server と Calendar Server の相互作用に関する考慮事項	47
Directory Server と個人アドレスブックに関する考慮事項	48
Directory Server と Communications Express の考慮事項	48
Messaging Server の考慮事項	49
Message Store の考慮事項	49
Message Transfer Agent (MTA) の考慮事項	50
MTA と LMTP の考慮事項	50
Mail Message Proxy (MMP) の考慮事項	51
Messaging Express Multiplexor (MEM) の考慮事項	51
Calendar Server の考慮事項	52
Instant Messaging の考慮事項	54
Portal Server の考慮事項	54
Connector for Microsoft Outlook の考慮事項	55
Connector for Microsoft Outlook コンポーネント製品の依存性	55
Sun ONE Calendar Server データの移行	56

Exchange Server データの移行	56
Communications Express の考慮事項	56
セキュリティの考慮事項	57
ネットワークのセキュリティ	58
オペレーティングシステムのセキュリティ	58
アプリケーションのセキュリティ	59
安全な接続の実装	59
2 つの異なる認証局 (CA) を使用する安全な接続の実装	60
第 4 章 Communications Services 論理アーキテクチャの開発	61
Communications Services Enterprise Deployment 論理アーキテクチャの概要	61
単一層、単一ホスト論理アーキテクチャ	62
単一層、複数ホスト論理アーキテクチャ	63
単一層、分散論理アーキテクチャ	64
2 層論理アーキテクチャ	66
エッジ論理アーキテクチャ	68
エッジアーキテクチャ設計の推奨事項	70
単一層アーキテクチャの利点	70
2 層アーキテクチャの利点	71
水平方向のスケーラビリティ戦略	73
その他の配備の課題	74
Messaging Server 用の LMTP の実装	74
Realtime Blackhole List (RBL) の実装	75
論理サービス名の使用	75
第 5 章 サービスの可用性の設計	79
高可用性ソリューションの概要	79
対称型 HA	80
非対称型 HA	80
システムの自動再設定 (ASR)	81
有効化テクニックとテクノロジーの使用	81
ロードバランサの使用	81
Directory Proxy Server の使用	81
複製ロールプロモーションの使用	82
Communications Services の高可用性ソリューション	83
Directory Sever の高可用性化	83
Messaging Server および Calendar Server の高可用性化	85
第 6 章 Communications Services ソフトウェアの機能	87
Communications Services コンポーネントの機能	87
Messaging Server ソフトウェアの概要	88
Messaging Server アーキテクチャの概要	89

Messenger Express による Web ベースのメールクライアントサービス (HTTP)	91
Messaging Server の詳細情報の入手先	91
Calendar Server ソフトウェアの概要	92
Calendar Server アーキテクチャの概要	92
Calendar Server の詳細情報の入手先	94
Instant Messaging ソフトウェアの概要	94
Instant Messaging アーキテクチャの概要	95
Instant Messaging の詳細情報の入手先	97
インフラストラクチャコンポーネントの機能	98
Directory Server ソフトウェアの概要	98
Directory Server アーキテクチャの概要	99
Directory Server の詳細情報の入手先	99
Identity Server ソフトウェアの概要	100
Identity Server アーキテクチャの概要	100
Identity Server の詳細情報の入手先	100
DNS の概要	101
第 7 章 Communications Services 配備の例	103
2 層配備の例	103
用語集	107
索引	109

図目次

図 3-1	Communications Services のコンポーネント	39
図 3-2	2 ツリー LDAP 構造と 1 ツリー構造との比較	41
図 3-3	aliasedDomainName と inetDomainBaseDN を持つ 2 ツリーエイリアス	42
図 3-4	inetCanonicalDomainName 属性を持つ 2 ツリーエイリアス	43
図 3-5	associatedDomain を持つ 1 ツリーエイリアス	43
図 4-1	単一層、1 ホスト論理アーキテクチャ	62
図 4-2	単一層、複数ホスト論理アーキテクチャ	64
図 4-3	単一層、分散論理アーキテクチャ	65
図 4-4	2 層論理アーキテクチャ	67
図 4-5	エッジ論理アーキテクチャ	69
図 6-1	Messaging Server の基本アーキテクチャ	90
図 6-2	Calendar Server 内部サブシステムの論理フロー	93
図 6-3	Instant Messaging の基本アーキテクチャ	96
図 7-1	Communications Services 2 層配備の例	104

表目次

表 1	書体の表記規約	12
表 2	プレースフォルダの表記規約	12
表 3	記号の表記規約	12
表 1-1	Communications Services が企業に提供する利点	22
表 2-1	総所有コストに関する検討事項	35
表 4-1	ユーザー側の論理名	75
表 4-2	保守レベルの論理名	76
表 4-3	保守レベル論理名のユーザーレベルへのマッピング	76
表 5-1	高可用性 Directory Server の設計	83
表 7-1	2 層配備の例で使用するプロトコルとポート	105

はじめに

『Sun Java System Communications Services 6 2004Q2 企業向け配備計画ガイド』には、企業環境に Sun Java™ System Communications Services 6 2004Q2 を配備するために必要な情報が含まれています。このガイドは、Communications Services の理解、サイトの評価と分析、企業のニーズに適合する配備アーキテクチャの設計プロセスに役立ちます。

この序文には、以下の節があります。

- [対象読者](#)
- [表記規約](#)
- [Web 上の情報源](#)
- [問題を報告する方法](#)
- [コメントをお寄せください](#)

対象読者

このガイドは、Communications Services を事前評価し、サイトに配備する次のような担当者向けに作成されています。

- 評価者
- 設計者
- システム管理者

このガイドは、読者が以下の事項に精通していることを前提としています。

- 企業レベルのソフトウェアのインストール方法
- IMAP、POP、HTTP、SMTP、WCAP、および LDAP プロトコル
- Solaris システム管理とネットワーク管理

表記規約

下表は、このガイドで使用する書体の表記規約について説明します。

表 1 書体の表記規約

書体	意味	例
AaBbCc123 (モノスペース)	API の言語要素、HTML タグ、Web サイトの URL、コマンド名、ファイル名、ディレクトリパス名、画面上的コンピュータ出力、サンプルコードを示す	.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表示します。 % You have mail
AaBbCc123 (太字のモノスペース)	ユーザーが入力する文字を、画面上的コンピュータ出力とは区別して表示す	% su Password:
AaBbCc123 (斜体)	新しい語句または用語を表示する 実際の名前または変数に置き換えられるコマンド行変数を表示する	これらを <i>class</i> オプションと呼びます。 ファイルは、 <i>ms_svr_base/bin</i> ディレクトリにあります。

下表は、このガイドで使用するプレースフォルダの表記規約について説明します。

表 2 プレースフォルダの表記規約

項目	意味	例
<i>product_base</i>	製品がインストールされるディレクトリのプレースフォルダを表示する	<i>ms_svr_base/bin</i> ディレクトリは、 <i>/opt/SUNWmgsrc</i> を表すことができます。

下表は、このガイドで使用する記号の表記規約について説明します。

表 3 記号の表記規約

記号	意味	表記	例
[]	コマンドオプションの選択肢が含まれる	O[n]	O4, O

表 3 記号の表記規約 (続き)

記号	意味	表記	例
{ }	必要なコマンドオプションの選択肢のセットが含まれる	d{y n}	dy
	コマンドオプションの選択肢を区切る		
+	グラフィカルユーザーインターフェイスで使用するキーボードショートカットで同時に実行するキーストロークを結び付ける		Ctrl+A
-	グラフィカルユーザーインターフェイスで使用するキーボードショートカットで連続的に実行するキーストロークを結び付ける		Esc-S
>	グラフィカルユーザーインターフェイスのメニューの選択を表示する		ファイル>新規 ファイル>新規>テンプレート

Web 上の情報源

このガイドのほかに、Sun Java™ System Calendar Server 6、Sun Java™ System Messaging Server 6、Sun Java™ System Instant Messaging Server 6 用のマニュアルが役立つ場合があります。これらのマニュアルを参照するには、以下の URL にアクセスしてください。

<http://docs.sun.com/db/prod/entsys?l=ja>

このマニュアルには、追加の関連情報を提供するために、サードパーティの URL も記載されています。

注 Sun は、このマニュアルに記載されているサードパーティ Web サイトの利用可能性について責任を負いません。Sun は、サードパーティのサイトやリソース上またはこれらを通じて利用できるコンテンツ、広告、製品、その他の素材について保証せず、いかなる責任も負いません。こうしたサイトやリソース上またはこれらを通じて利用できるコンテンツ、製品、またはサービスを利用または信用したことに伴って発生した (あるいは発生したと主張される) いかなる損害や損失についても、Sun は一切責任を負いません。

問題を報告する方法

Communications Services の使用上問題が発生した場合は、以下の方法のいずれかを使用して Sun のカスタマサポートに連絡してください。

- 次の Sun Software Support オンラインサービス

<http://www.sun.com/service/sunone/software>

このサイトには、ナレッジベース、オンラインサポートセンター、ProductTracker、保守プログラムおよびサポートの連絡先電話番号へのリンクがあります。

- 保守契約に関連する電話番号

お客様の問題を解決するため、テクニカルサポートに連絡する際はあらかじめ次の情報をご用意ください。

- 問題が発生した箇所や動作への影響など、問題の具体的な説明
- マシン機種、OS バージョン、および、問題の原因と思われるパッチやそのほかのソフトウェアなどの製品バージョン
- 問題を再現するための具体的な手順の説明
- エラーログやコアダンプ

コメントをお寄せください

弊社ではマニュアルの改善に努力しており、お客様からのコメントおよび提案を歓迎いたします。以下の Web ベースの様式を使用して Sun にフィードバックしてください。

<http://www.sun.com/hwdocs/feedback/>

マニュアルの完全なタイトルと Part-No を含めてください。Part No. は、マニュアルのタイトルページまたは上部に記載された 7 桁または 9 桁の番号です。たとえば、この『Communications Services 企業向け配備計画ガイド』の Part No. は 817-7090 です。

Communications Services について

この章では、Sun Java™ System Communications Services 6 2004Q2 の概要、Communications Services の配備に関する業務上の理由、および配備プロセスそのものについて説明します。

この章には、以下の節があります。

- [Communications Services の概要](#)
- [Communications Services の企業のビジネスニーズへの対応方法について](#)
- [配備プロセスについて](#)

Communications Services の概要

Sun Java System Communications Services 6 2004Q2 は、安全で、費用効率の高い通信とコラボレーションを提供します。Communications Services は、他の企業向けメッセージングソリューションに代わる、安全で、スケーラブルな、総所有コスト (TCO) を削減するソリューションを提供し、企業にとって懸念であるコスト、機能、および従来の企業通信インフラストラクチャのセキュリティなどの問題解決に取り組みます。

Communications Services は、企業の通信およびコラボレーションのニーズに対応するために必要な電子メール、カレンダー、およびインスタントメッセージングソリューションを提供します。Communications Services の製品とサービスは、一般的なビジネス要件に対する強力な対応策を提供します。あらゆる企業にとって、通信は不可欠です。そして多くの場合、大規模な範囲の多様で地理的に分散しているユーザーコミュニティに対して通信サービスを提供する必要があります。従来の通信ソリューションはコストがかかり、今日の企業のスケーラビリティとセキュリティ要件に対応するのに十分ではありません。Communications Services によって、企業は総所有コストの予算内でソリューションを配備することが可能になります。

また、Communications Services は、多様な顧客が必要とする独自のサービスとフル装備のコラボレーション機能を提供します。Communications Services 配備は、企業のファイアウォールの外側に拡大する通信と複数のデバイスを使用するモバイルユーザーに対して、ますます増大する企業のセキュリティニーズに対応します。

Communications Services のコアソリューションは、以下のコンポーネントで構成されます。

- Sun Java™ System Messaging Server 6 (従来の Sun™ ONE Messaging Server)
- Sun Java™ System Calendar Server 6 (従来の Sun™ ONE Calendar Server)
- Sun Java™ System Instant Messaging 6 (従来の Sun™ ONE Instant Messaging)

Communications Services ソリューションを拡張する追加製品には以下のものがあります。

- Sun ONE™ Synchronization 1.1
- Sun Java™ System Connector for Microsoft Outlook 6 2004Q2
- Sun Java™ System Communications Express 6

全体として、Communications Services は、何千ものユーザーを抱える企業向け配備のための標準ベースの統合された通信およびコラボレーション製品群を提供します。Communications Services は、あらゆる組織の多様な通信ニーズに対応する堅固で柔軟なプラットフォームを提供します。Communications Services は、遠隔地オフィス、分散ワークグループ、グローバルな企業拠点を接続するための最適なソリューションです。

注 現在、このガイドでは、最高 5,000 までの企業ユーザーで構成される企業の配備に対応しています。

Messaging Server について

Sun Java System Messaging Server 6 は、高性能かつ高い安全性を備えたメッセージングプラットフォームです。数千人から数万人規模のユーザーのスケールリングに対応する Messaging Server は、電子メールサーバーを統合し、通信インフラストラクチャの総所有コストを削減しようとする企業に適しています。Messaging Server は、ユーザー認証、セッションの暗号化、スパムとウィルスの防止に役立つ適切なコンテンツのフィルタリングを通し通信の統合を実現する幅広いセキュリティ機能を提供します。

Messaging Server によって、企業は社員、パートナー、および顧客からなるコミュニティ全体に対して安全で信頼性の高いメッセージングサービスを提供できます。

このガイドで取り扱わない Messaging Server の概念とその他の配備に関する詳細については、次の Web サイトの『Sun Java System Messaging Server 配備計画ガイド』を参照してください。

<http://docs.sun.com/db/prod/entsys?l=ja>

Calendar Server について

Sun Java System Calendar Server 6 は、ユーザーによるアポイントメント、予定、作業、リソースの管理、調整を可能にして、円滑なチームコラボレーションを可能にします。Calendar Server は、直観的な Web ベースのインタフェースによって、エンドユーザーが任意の時間、任意の場所で任意の Web 対応デバイスから、非公開、公開、またはグループカレンダーにアクセスできるようにします。企業配備は、Messaging Server および Instant Messaging とともに Calendar Server を使用して、総合的な通信と協調的な環境をユーザーに提供します。

Calendar Server の概念に関する詳細については、次の Web サイトの『Sun Java System Calendar Server 管理ガイド』を参照してください。

<http://docs.sun.com/db/prod/entsys?l=ja>

Instant Messaging について

Sun Java System Instant Messaging 6 は、安全で、リアルタイムの通信とコラボレーションを可能にします。Instant Messaging は、参加の確認をチャット、会議、アラート、ニュース、ポーリング、ファイル転送などのインスタントメッセージング機能と組み合わせて、機能の豊富なコラボレーション環境を形成します。これらの機能は、短時間の通信または会議室やニュースチャンネルなど継続的な場のいずれかを通じて、1対1のコラボレーションとともにグループコラボレーションを可能にします。Instant Messaging は、複数の認証メカニズムと安全な SSL 接続によって通信の統合を実現します。Sun Java™ System Portal Server 6 と Sun Java™ System Identity Server 6 への統合により、追加のセキュリティ機能、サービスベースのプロビジョニングアクセスポリシー、ユーザー管理、安全なリモートアクセスが可能になります。

Instant Messaging の概念と配備に関する考慮事項の詳細については、次の Web サイトの『Sun Java System Instant Messaging 配備計画ガイド』を参照してください。

<http://docs.sun.com/db/prod/entsys?l=ja>

注 Instant Messaging は Communications Services 製品の重要な一部ですが、このガイドでは Instant Messaging を配備する方法については説明していません。この情報については、『Sun Java System Instant Messaging 配備計画ガイド』を参照してください。

同期について

Sun ONE Synchronization 1.1 は、Windows パーソナルコンピュータ上で実行されるソフトウェア製品で、Calendar Server の予定および作業と、モバイルデバイスや Microsoft Outlook などの PIM (Personal Information Manager) との同期を可能にします。

詳細については、次の Web サイトにある Sun ONE Synchronization のマニュアルを参照してください。

http://docs.sun.com/db/coll/S1_Sync_11

Connector for Microsoft Outlook について

Sun Java System Connector 6 for Microsoft Outlook を使用して、Outlook を Sun Java Enterprise System のデスクトップクライアントとして使用できるようにします。

Connector for Microsoft Outlook は、エンドユーザーのデスクトップにインストールする Outlook のプラグインです。Connector for Microsoft Outlook は、Messaging Server にフォルダの階層と電子メールメッセージを照会します。次に、Connector for Microsoft Outlook は、この情報を Outlook で表示できる MAPI (Messaging API) プロパティに変換します。同様に、Connector for Microsoft Outlook では、WCAP を使用して Calendar Server に予定と作業を照会し、MAPI プロパティに変換します。このモデルによって、Connector for Microsoft Outlook では、Messaging Server のメールと Calendar Server のカレンダー情報の 2 つの別個の情報源からエンドユーザーの Outlook 表示を作成します。

詳細については、次の Web サイトにある Connector for Microsoft Outlook のマニュアルを参照してください。

<http://docs.sun.com/db/prod/entsys?l=ja>

Communications Express について

Sun Java System Communications Express 6 は、企業ユーザーのニーズに対応する、統合された Web ベースの通信およびコラボレーションクライアントです。

Communications Express は、Web ベースのクライアントとして、Web サーバーを使用してアクセスし、ブラウザを使用して表示します。Communications Express 製品は、Calendar、Address Book、Mail の 3 つのクライアントモジュールで構成されます。Calendar、Address Book、Mail クライアントモジュールは、任意の Web コンテナ上に単独のアプリケーションとして配備されます。

Communications Express のほとんどの機能をカスタマイズできます。詳細については、次の Web サイトにある Communications Express のマニュアルを参照してください。

<http://docs.sun.com/db/prod/entsys?l=ja>

Communications Services コンポーネント製品の依存性

Communications Services は、インフラストラクチャサービスを提供するほかの Sun Java System コンポーネント製品との依存関係もあります。これらのコンポーネント製品には、Sun Java™ System Directory Server と Sun Java™ System Identity Server が含まれます。さらに、Communication Services は、HTML コンテンツを提供し、HTML 接続を提供する Web サーバーに依存します。この機能を実行するために、Sun Java™ System Web Server (Sun™ ONE Web Server と呼ばれる) を使用できます。

また、Communications Services は DNS 機能にも依存します。Communications Services 製品をインストールするには、DNS サーバーが機能している必要があります。

製品の依存関係の詳細については、[第 3 章「製品の要件と考慮事項について」](#) を参照してください。

Communications Services の企業のビジネスニーズへの対応方法について

企業は、強力な機能を備えると同時に、コストを削減し管理を簡素化するためのサービスの配備が必要です。企業通信サービスのアーキテクチャには、ユーザーが日常業務の遂行に不可欠な情報に、複数の方法でのアクセスを可能にするためのセキュリティとスケーラビリティの要件を追加する必要があります。Communications Services では、企業の総所有コストの予算内でスケーラブルなメッセージング、カレンダー、インスタントメッセージングを提供することによりこれらのニーズに対応します。

Communications Services により、配備と保守が容易で、完全な機能を持つアーキテクチャの開発が可能になります。最も重要なことは、Communications Services アーキテクチャによって各サービス要素にセキュリティが組み込まれることです。これらの要素には、ネットワークインフラストラクチャ、動作環境、および Communications Service コンポーネント製品そのものが含まれます。

Messaging Server のビジネスニーズへの対応方法について

Messaging Server は、優れた信頼性と生産性の向上を促進するとともに、管理と運用コストを低減します。Messaging Server では、メッセージがディスク上のメッセージストアに格納されるまで受信として承認されないことを意味する格納済みトランザクションを使用します。この信頼性機能は、メールメッセージの損失や破損を防止します。さらに、Message Store は、卓越したパフォーマンスとデータ統合を実現するために、追記型データストアと 2 段階インデックスを採用するカスタム設計のデータベースを中心に構築されます。

Calendar Server のビジネスニーズへの対応方法について

Calendar Server は、業界で最もオープンで相互運用可能な、高パフォーマンスの時間管理および資源管理ソリューションを提供します。Calendar Server は、ほかの代替ソリューションよりも低い総所有コストで必要な機能を提供します。Calendar Server は、柔軟性のある拡張可能なアーキテクチャによって、垂直方向 (システムあたり CPU 数の増加) と水平方向 (ネットワークへのサーバーの追加) の両方向にスケールアップします。

Instant Messaging のビジネスニーズへの対応方法について

Instant Messaging ソフトウェアは、プロジェクトのライフサイクルを短縮し、新しいサービスを手ごろな価格で配備できるように Java Enterprise System と緊密に統合されています。さらに、Instant Messaging は、Portal Server、Identity Server、Messaging Server、Calendar Server と完全に統合されています。この統合によって、企業ユーザーは、安全かつスケーラブルなフル装備の通信およびコラボレーションサービスのプラットフォームを、単一のベンダーから入手できます。Instant Messaging に含まれる定評ある Java API は、複数のプラットフォームのサポート、プラットフォームの拡張性、リアルタイム通信およびコラボレーション機能のカスタマイズとともに、統合を容易にするオープンな標準を提供します。これらの機能は、既存の企業アプリケーションに組み込まれたり、あるいは新しいアプリケーションの基盤となります。

Communications Services の利点のまとめ

従来、Communications Services コンポーネントは、大規模な、通信事業者クラスの設備に使用されてきました。このガイドで説明されているとおり、大規模設備で要求されるのと同じ信頼性を企業で利用することができます。

以下の表に、Communications Services によって企業が得られる利点を要約します。

表 1-1 Communications Services が企業に提供する利点

主な機能	企業に提供する利点
高パフォーマンスとスケーラビリティ	効率的な通信を可能にし、企業と ISP の両者のサービス品質の向上を図る
幅広いセキュリティ機能	通信とデータの統合、社員、顧客、パートナーのプライバシーを保護し、業界の規制準拠に対応する
仮想ドメインのホスティングと委任管理	Messaging Server は 1 つのサーバーで複数の企業のメッセージをホストし、あるいは企業 IT が企業内の複数の部門をホストできるようにして、必要なサーバー数を削減し、TCO を低減する
スケーラブルで、堅固な、拡張性のあるコンポーネント	電話サービスと電子メール通知、ファックス、ページング、その他のテクノロジーを結合して、統一された通信サービスの配備を可能にする
予定管理、作業とリソースの管理のための拡張可能なコラボレーションプラットフォーム	Calendar Server は、時間管理と資源管理を改善し、ユーザーの生産性を向上させる
会議や予定のグループスケジューリング	Calendar Server は、企業全体のチームコラボレーションと通信を向上させる
予定または作業のハイパーリンクによる情報の共有	Calendar Server は、作業や予定に関連する情報の交換を通じてコラボレーションを促進する
複数クライアントのサポート	Ximian Evolution および Microsoft Outlook を含む多くの複数のクライアントに対して、統合された Web ベースのクライアントとサポートを提供する
オープンな、モジュラー化された標準ベースのアーキテクチャ	カスタマイズされた顧客独自のソリューションの配備を可能にする

Communications Services 配備の高可用性の向上

Messaging Server と Calendar Server は、両者とも Sun™ Cluster サービスと Veritas クラスタリングソリューションをサポートする高可用性オプションを提供します。このオプションによって、プライマリシステムが保守目的でオフラインとなっている場合、あるいは障害によりダウンしている場合に、Message Server または Calendar Server のセカンダリホストがユーザーにサービスを提供します。

Sun Cluster を使用しなくても、Messaging Server には、サーバープロセスとサービスの可用性の状態を継続的にチェックする組み込み監視機能が装備されています。Messaging Server は、必要に応じてプロセスとサービスを自動的に再起動することができます。レポートと分析を選択した場合、Messaging Server は障害と回復操作のログを記録します。

さらに、冗長コンポーネントを使用することにより、高度に可用性のある構成で Communications Services 製品を配備することができます。この種類の配備では、高レベルの作動時間のサービスが提供されます。このような高可用性配備では、サービスアーキテクチャにおけるすべてのコンポーネントに冗長性が要求されます。これらのコンポーネントには、重複データストアサーバー、重複ネットワークインタフェースカード、および重複システムストレージが含まれます。

注 このガイドでは、Communications Services の高可用性配備における Sun Cluster の利用に関する詳細は取り扱っていません。このトピックに関する詳細については、Sun Cluster、Messaging Server、Calendar Server のマニュアルを参照してください。

Communications Services での Portal Server の使用

Portal Server を含む Communication Services 製品のインストールでポータルページのメッセージングおよびカレンダーポートレットにアクセスできます。これらのポートレットは、メッセージング情報、カレンダースケジュール、アドレスブック情報の要約を提供します。Portal Server の統合には、Portal Server、Calendar Express Web クライアント、Messaging Express Web クライアント、Communications Express クライアント間のシングルサインオン機能が含まれます。

注 Sun Java™ System Schema 1 と Schema 2 の両方の環境で、Communications Express を実行できます。Schema 2 を使用している場合は、Identity Server 認証を使用して Communications Express にシングルサインオンすることができます。

また、Portal Server は Instant Messaging のメッセージアーカイブをサポートします。さらに、ユーザーは、Portal Server Desktop を使用して、Messaging Express、Calendar Express、Instant Messaging クライアントを利用することができます。

Portal Server の以下の 2 つのコンポーネントは、Communications Services の基本配備に対する追加機能を提供します。

- **Portal Server Desktop:** Portal Server にインストールされた Instant Messenger は、Instant Messaging クライアントを Instant Messaging チャネルから起動できるようにします。
- **Sun Java™ System Portal Server 6 Secure Remote Access:** リモートエンドユーザーが、インターネットを介して Secure Remote Access 組織のネットワークとそのサービスに安全に接続できるようにします。エンドユーザーは、Secure Remote Access ゲートウェイを経由して、Web ベースの Portal Server Desktop にログインすることにより、Secure Remote Access にアクセスします。Portal Server に設定された認証モジュールがエンドユーザーを認証します。Portal Server でエンドユーザーのセッションが確立され、エンドユーザーの Portal Server Desktop へのアクセスが可能になります。

注 このガイドでは、ポータル環境における Communications Services のポータル配備については取り扱っていません。詳細については、Portal Server のマニュアルを参照してください。

配備プロセスについて

Communications Services の配備プロセスは、以下の一般的な段階で構成されます。

- 配備設計
- 開発
- プロトタイプテスト
- 本稼動環境への展開

配備の段階は固定的なものではありません。実際には配備プロセスは繰り返し実行されますが、以下の項では各配備段階について個別に説明します。

Communications Services の配備プロセスの詳細については、次の Web サイトの『Sun Java Systems Messaging Server 配備計画ガイド』を参照してください。

<http://docs.sun.com/db/prod/entsys?l=ja>

配備アーキテクチャの設計

通常、配備設計の段階では、要件分析の段階で特定した配備シナリオに基づいて配備アーキテクチャを構築します。この目的は、配備シナリオで指定されたシステム要件に適合するように、論理構築ブロック（論理アーキテクチャ）を物理環境にマッピングすることです。

この設計の段階の 1 つに、負荷、可用性、およびパフォーマンス要件に適合するように物理環境のサイズを確定することがあります。配備アーキテクチャでは、環境内の処理ノードにシステムサーバーやアプリケーションコンポーネントを割り当てる際に、さまざまな処理ノードやネットワーク帯域幅の能力などの物理トポロジの詳細を考慮します。

配備とカスタマイズ

ライフサイクルの要件分析の段階で指定された論理アーキテクチャでは、ソリューションの実装に必要な開発作業の範囲を決定します。

API の使用によるサービスの拡張、または企業商標の設定などのルックアンドフィールのカスタマイズのいずれかの追加作業が必要になる場合があります。

一部のソリューションでは、新しいビジネスやプレゼンテーションサービスの開発が要求される場合、配備のカスタマイズが非常に広範囲にわたる場合があります。その他の場合には、必要な機能を実現するために、Portal Server デスクトップなど既存のユーザーインタフェースをカスタマイズするだけで十分な場合があります。

製品 API の使用や製品機能のカスタマイズの詳細については、以下の該当するコンポーネントのマニュアルを参照してください。

- 『Sun Java System Messaging Server Developer's Reference』
- 『Sun Java System Messaging Server Messenger Express Customization Guide』
- 『Sun Java System Communications Services Event Notification Service Guide』
- 『Sun Java System Calendar Server Developer's Guide』

プロトタイプとテスト

プロトタイプの段階では、配備アーキテクチャをテスト環境に実装することにより配備設計のプロトタイプを作成します。開発段階で作成した新しいアプリケーションロジックとサーバーのカスタマイズを使用して、上記の説明に従って POC (機能検証) 配備テストを実行します (26 ページの「[配備とカスタマイズ](#)」を参照)。この段階には、テスト環境への分散アプリケーションのインストール、設定、起動、および必要なインフラストラクチャサービスが含まれます。

プロトタイプテストにより配備アーキテクチャの欠陥が検出された場合は、アーキテクチャを修正し、再度プロトタイプを作成してテストします。この反復的なプロセスによって、本稼動環境に配備が可能な配備アーキテクチャが完成します。

製品システムの展開

製品展開の段階では、本稼働環境に配備アーキテクチャを実装します。この段階には、本稼働環境への分散アプリケーションのインストール、設定、起動、および必要なインフラストラクチャサービスが含まれます。通常は、限定的な配備から開始して、組織全体への実装に移行します。このプロセスでは、負荷を増大させ、システムの耐久度テストを行う試運転を実行します。

展開段階の一環として、ユーザーのプロビジョニング、シングルサインオンの実装、パフォーマンス目標に適合させるためのシステムの調整などの管理タスクを実行する必要があります。配備の検証と容量計画の実行もこの段階の一部です。システムの監視が重要な役割を果たす容量計画は、システムの長期的な拡大のニーズに対応するために必要になります。

配備プロセスについて

企業の要件の分析

Communications Services 配備の計画においては、まず企業のビジネスと技術的な要件を分析する必要があります。この章は、Communications Services 設計を決定するために使用する要件を収集し、評価するのに役立ちます。

この章には、以下の節があります。

- [配備目標の明確化](#)
- [プロジェクトの目標の特定](#)

配備目標の明確化

Communications Services ハードウェアまたはソフトウェアを購入または配備する前に、配備目標を明確にする必要があります。配備要件は企業内のあらゆるソースが関係しています。多くの場合、要件はあいまいな用語で表現されるため、特定の目標を確定できるようにそれらの用語を明確化する必要があります。

要件分析の結果は、配備が成功したかどうかを評価できるように明確かつ簡潔で、測定可能な目標でなければなりません。プロジェクトの利害関係者に受け入れられるような明確な目標を持たずに推進するのでは根拠の乏しい計画になってしまいます。

配備計画を立てる前に確認する必要がある要件の一部は以下のとおりです。

- ビジネス要件
- 技術要件
- 財務要件
- サービス品質保証契約 (SLA)

ビジネス要件

ビジネスの目的が配備の決定に影響を与えます。特に、ユーザーの動作、サイトの分散、および配備に影響を与える企業内の潜在的な政治的問題について理解する必要があります。これらのビジネス要件を理解していないと、容易に間違った推定に基づく決定をしてしまい、配備設計の正確性に悪影響を与える可能性があります。

運用要件

簡単でわかりやすい目標として、運用要件を一連の機能要件として表現します。通常、以下のような非公式の仕様があります。

- エンドユーザーの機能
- エンドユーザーの応答時間
- 可用性と作動時間
- 情報のアーカイブと保存

たとえば、「適切なエンドユーザーの応答時間」を、「適切な」とは何か、応答時間の測定方法などすべての利害関係者が理解できる測定可能な用語に変換します。

企業文化と政治

配備にあたっては企業文化と政治を考慮する必要があります。ビジネス要件を代表する部門からさまざまな要求が提出される場合があります。

例：

- 一部のサイトは、配備されるソリューションを自分自身で管理することを要求します。このような要求は、プロジェクトのトレーニング費用、複雑性などを増加させます。
- LDAP ディレクトリに人事データが含まれる場合、人事管理部門がそのディレクトリを所有し、制御することを要求する場合があります。

技術要件

技術要件（または機能要件）とは、組織のシステムニーズの詳細です。

既存の使用パターンのサポート

既存の使用パターンを、配備が実現する明確で測定可能な目標として表現します。以下の質問が、このような目標を特定するのに役立ちます。

- 現在のサービスをどのように活用するか。
- ユーザーを分類することができるか（たとえば、散発的ユーザー、常連ユーザー、ヘビーユーザーなど）。
- ユーザーが通常送信するメッセージのサイズはどの程度か。
- カレンダーのアポイントメントに招待される参加者の数は通常何人か。
- ユーザーが送信するメッセージの数は何通か。
- 通常、ユーザーは一日あたりまたは一時間あたり何件のカレンダー予定および作業を作成するか。
- ユーザーは企業のどのサイトにメッセージを送信するか。

サービスにアクセスするユーザーについて調べます。ユーザーが既存のサービスをいつ使用するかなどの要素は、配備要件すなわち目標を特定するための手がかりとなります。組織の経験からこのようなパターンが得られない場合は、別の組織の経験を調べて予測します。

利用率の高い組織の領域には、専用のサーバーが必要になる場合があります。通常、実際のサーバーから離れた場所のユーザーの場合、応答時間が遅くなります。容認可能な応答時間を検討します。

サイトの分散

以下の質問を使用して、サイトの分散が配備目標にどのように影響を与えるかについて理解します。

- サイトは地理的にどのように分散しているか。
- サイト間の帯域幅はどうなっているか。中央集中型の手法は、分散型よりも大きな帯域幅が必要です。基幹業務を扱うサイトには、専用のサーバーが必要になる場合があります。

ネットワーク

以下の質問がネットワークの要件を理解するのに役立ちます。

- 内部ネットワーク情報を難読化するか。
- ネットワークサービスの冗長性を装備するか。

- アクセスレイヤーホスト上の利用可能なデータを制限するか。
- エンドユーザーが変更する必要のない単一のメールホストに入るなど、エンドユーザーの設定を簡素化するか。
- ネットワークの HTTP トラフィックを軽減するか。

注 これらの質問に対する答えが「はい」の場合には、2層アーキテクチャをお勧めします。

既存のインフラストラクチャ

信頼性が高く、可用性の高い帯域幅がある場合は、サーバーを集中化することができます。

- 既存のインフラストラクチャや設備は、この配備を行うのに適しているか。
- DNS サーバーは負荷の上積みに対応できるか。ディレクトリサーバー、ネットワーク、ルーター、スイッチ、ファイアウォールはどうか。

サポート要員

一日 24 時間、一週 7 日間のサポートは、特定のサイトでのみ利用可能です。簡単なアーキテクチャで、サーバーの数が少ないほどサポートが簡単になります。

- この配備を円滑に推進するために、運用およびテクニカルサポートグループに対する十分な人的資源がありますか。
- 配備段階で、運用およびテクニカルサポートグループは増加する負荷に対応することができますか。

財務要件

財務要件は配備を構築する方法に影響を与えます。財務要件は、総合的な観点を通して明確に定義されるため、配備の制限または配備対象が絞られる傾向にあります。

明白なハードウェア、ソフトウェア、および保守費用のほかに、以下のような多くのコストがプロジェクトの全体的コストに影響を与えます。

- トレーニング
- たとえば、ネットワーク帯域幅またはルーターなど、その他のサービスや設備のアップグレード
- 配備の構想を実証するために必要な要員や資源などの配備コスト
- 配備したソリューションを管理する要員などの運用コスト

プロジェクトの要件に関連する多くの要素に対して十分な注意を払い、分析を行うことによって、プロジェクトに関連する財務問題を回避することができます。

サービス品質保証契約 (SLA)

作動時間、応答時間、メッセージ配信時間、障害回復などの配備に関連する SLA を開発する必要があります。SLA 自体で、システムの概要、サポート組織の役割と責任、応答時間、サービス水準の測定方法、変更の要求などの項目を明らかにする必要があります。

システムの可用性に関する組織の要求を確認することは、SLA の範囲を決定するためのかぎです。通常、システムの可用性は、システムの作動時間のパーセンテージで表されます。システムの可用性を算出する基本的な計算式は次のとおりです。

$$\text{可用性} = \text{作動時間} / (\text{作動時間} + \text{不稼動時間}) * 100$$

たとえば、サービス品質保証契約でフォーナイン (99.99%) の作動時間は、1 ヶ月で約 4 分間システムが利用できないことを意味します。

さらに、システムの不稼動時間とは、システムを利用できない合計時間を示します。この合計時間には、ハードウェアの故障やネットワークの停止などの予測できないダウンタイムだけでなく、計画されたダウンタイム、予防保守、ソフトウェアのアップグレード、パッチなどが含まれます。システムが一週 7 日間、一日 24 時間利用可能でなければならない場合は、計画されたダウンタイムや予測できないダウンタイムを回避して高可用性を確保するために、アーキテクチャに冗長性を備える必要があります。

プロジェクトの目標の特定

調査と分析によってプロジェクトの要件を明確にします。次に、はっきりと測定できる目標を特定することが必要になります。プロジェクトに直接関与しないスタッフが、目標と目標に対するプロジェクトの実績を測定する方法を理解できるようにこれらの目標を特定します。

利害関係者がプロジェクトの目標を受け入れる必要があります。実装後の審査でプロジェクトの目標を測定し、プロジェクトが成功したかどうかを確認する必要があります。

拡大のための計画

現在必要な容量を確定するだけでなく、計画できる期間内で将来必要になる容量を予測します。通常、拡大の時系列は6ヶ月から12ヶ月の範囲です。拡大に対応するために考慮しなければならない要素は、拡大の予測と利用特性の変化です。

ユーザーやメッセージ数の増加に応じて、容量計画の適切な指針の輪郭を描く必要があります。さまざまなサーバーのメッセージトラフィックの増大、全体のユーザー数の増加、メールボックスサイズの拡大、カレンダーのアポイントメントの増加などを計画に含める必要があります。ユーザー人口が増加すると、時間の経過に伴って利用特性が変化します。配備目標(すなわち配備設計)は、将来の成長に基づいていなければなりません。

理想的に言えば、将来の拡大に容易に対応できるようにアーキテクチャを設計する必要があります。たとえば、**Communications Services** 自体に論理名を使用します。詳細については、[75 ページの「論理サービス名の使用」](#)を参照してください。また、配備が本稼動段階に入ると、配備を拡大する時期と大きさを理解するために、配備の監視が非常に重要になります。

総所有コスト (TCO) の理解

総所有コスト (TCO) は、容量計画に影響を与えるもう一つの要素です。総所有コストには、**Communications Services** を配備するハードウェアの選択が含まれます。[35 ページの表 2-1](#) に、小規模なハードウェアシステムを数多く配備するか、または大規模なハードウェアシステムを数少なく配備するかを検討する要素の一部を示します。

表 2-1 総所有コストに関する検討事項

ハードウェアの選択	長所	短所
小規模な数多くのハードウェアシステム	<ul style="list-style-type: none"> • 通常、小規模なハードウェアシステムの場合はコストが低下する • 小規模な数多くのハードウェアシステムは、分散ビジネス環境をサポートする数多くの場所に配備することができる • 小規模な数多くのハードウェアシステムを配備する場合は、別のサーバーを保守している間にオンラインになっている別のサーバーにトラフィックを転送することができるため、システムの保守、アップグレード、移行のためのダウンタイムを短縮させることができる 	<ul style="list-style-type: none"> • 小規模なハードウェアシステムは、容量が制限されるため、数多くのハードウェアシステムが必要になる。ハードウェアシステムの数の増加に応じて、管理と保守費用が上昇する • さらに、小規模なハードウェアシステムは、保守対象が増加するため、多くのシステム保守が必要になる
大規模な数少ないハードウェアシステム	<ul style="list-style-type: none"> • 大規模な数少ないハードウェアシステムを配備する場合は、サーバーあたりの固定管理費が低下する。管理するハードウェアシステムの数が減少するため、内部またはISPからの月々の請求によって繰り返し発生する管理費が低減される • また、ハードウェアシステムの数が少ないと、保守するシステムの数が少なくなるため、システムの保守、アップグレード、移行が簡単になる 	<ul style="list-style-type: none"> • 通常、大規模なハードウェアシステムは初期コストが増加する • また、数少ないハードウェアシステムは、保守、アップグレード、移行のためのダウンタイムが増加することを意味する

プロジェクトの目標の特定

製品の要件と考慮事項について

この章では、配備設計に影響を与える要件と考慮事項について説明します。
Communications Services アーキテクチャを正確に決定するためには、これらの要件と考慮事項を理解する必要があります。

この章には、以下の節があります。

- [さまざまなコンポーネントの計画](#)
- [LDAP ディレクトリ情報ツリーの要件](#)
- [スキーマの要件](#)
- [Directory Server の考慮事項](#)
- [Messaging Server の考慮事項](#)
- [Calendar Server の考慮事項](#)
- [Instant Messaging の考慮事項](#)
- [Portal Server の考慮事項](#)
- [Connector for Microsoft Outlook の考慮事項](#)
- [Communications Express の考慮事項](#)
- [セキュリティの考慮事項](#)

さまざまなコンポーネントの計画

配備アーキテクチャを設計する場合、配備に使用するさまざまなコンポーネントの要件を考慮する必要があります。たとえば、**Communications Services** をほかの Java システム製品に統合するための技術要件がある場合は、対応するスキーマを選択する必要があります。同様に、たとえば、**Communication Service** が **Directory Server** にアクセスし、負荷を配置する方法などの製品間の依存性によって配備を選択する必要があります。

各製品の個々のコンポーネントを理解することにより、要件に最適なアーキテクチャの種類を計画することができます。配備に応じて、以下のコンポーネントの基本を理解し、計画する必要があります。

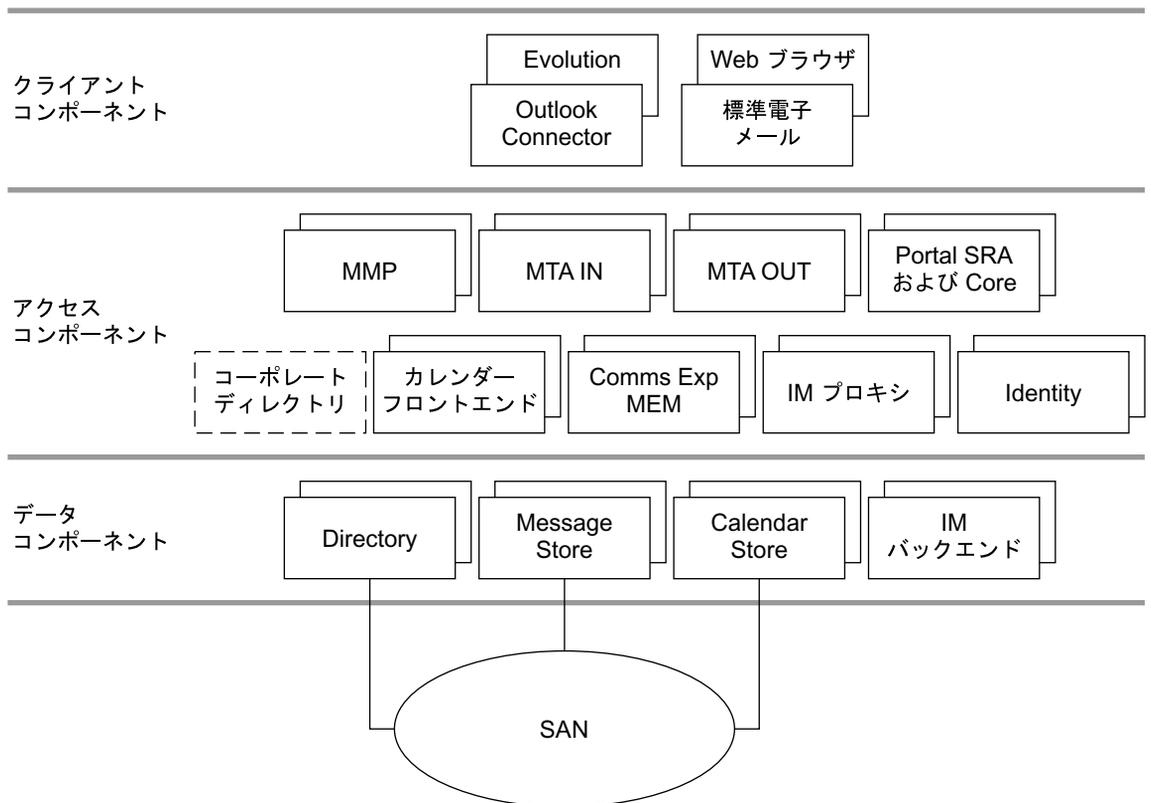
- LDAP ディレクトリの情報ツリー
- スキーマ (Schema)
- Directory Server (Identity Server)
- Messaging Server
 - Message Transfer Agent (MTA)
 - メッセージストア (Message Store)
 - MMP (Messaging Multiplexor)
 - MEM (Messaging Express Multiplexor)
- Calendar Server
 - フロントエンド (Front End)
 - カレンダーストア (Calendar Store)
- Instant Messaging
 - Instant Messaging プロキシ
 - Instant Messaging バックエンド
- Portal Server
- Connector for Microsoft Outlook
- Communications Express

サービスコンポーネントとサービス層の理解

複数のコンポーネント製品またはサービスを利用する Communications Services の配備を計画する場合、各コンポーネント製品（またはサービス）自体の構成を理解する必要があります。

39 ページの図 3-1 は、各サービスを独立したホストに配備できるコンポーネントに分割する方法と各コンポーネントが使用する特定の層を示しています。単一のホストにすべてのコンポーネントを配備したり、同一のホストに特定のサービスのコンポーネントを配備することもできますが、Tier (層) アーキテクチャに移行することを検討してください。Tier (層) アーキテクチャには、単一層の場合でも 2 層の場合でも多くの利点があります。詳細については、70 ページの「単一層アーキテクチャの利点」と 71 ページの「2 層アーキテクチャの利点」を参照してください。

図 3-1 Communications Services のコンポーネント



上図では、クライアントコンポーネントは Outlook Connector プラグイン、Evolution などのシッククライアント、ブラウザ、標準電子メールアプリケーションで構成されます。これらのコンポーネントは、エンドユーザーのクライアントコンピュータに配置されます。アクセス層のコンポーネントは、Messaging Server (MMP、MTA、MEM)、Calendar Server、Communications Express (MEM と連結する必要がある)、Instant Messaging (Instant Messaging Proxy)、Portal Server (SRA および Core)、Identity Server (認証用) およびアドレスブックの検索を提供するコーポレートディレクトリのフロントエンドサービスで構成されます。データ層のコンポーネントは、Directory Server (それ自体で、フロントエンドおよびバックエンドコンポーネントを構成できます)、Messaging Server (Message Store)、Calendar Server (Calendar Store)、および Instant Messaging のバックエンドサービスで構成されます。SAN (Storage Area Network) の「雲形模様」は物理データストレージを表します。

注 この図で示されるコーポレートディレクトリは、コンポーネント製品そのものではありません。これは、クライアントがアドレスブック形式の検索を行うために、企業が通常アクセス層に配備するコーポレートディレクトリの「コピー」を表しています。

以下の節では、これらのさまざまなコンポーネントを詳細に説明します。

LDAP ディレクトリ情報ツリーの要件

ディレクトリ情報ツリー (DIT) は、ディレクトリエントリをドメイン、サブドメイン、ユーザー、グループを表すノードを持つツリー構造またはスキーマに編成するための方法です。Sun Java Enterprise System では、1 ツリー構造を実装することにより、ディレクトリを構造化する方法の基本的な変更が行われています。

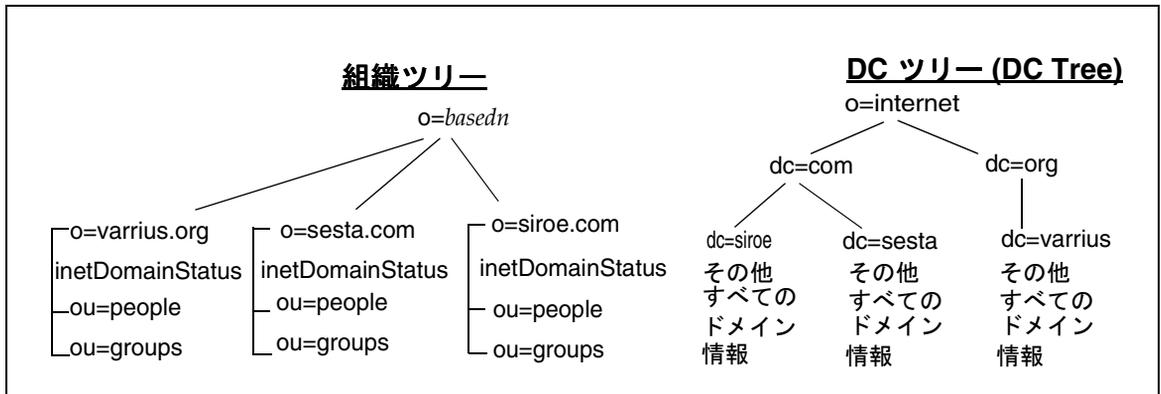
DIT 構造の変更

Messaging Server と Calendar Server は 1 ツリー構造を導入しており、ドメインコンポーネント (DC) ツリーがありません。すべてのドメイン情報が組織ツリーのドメインノードに保持されます。新しい 1 DIT 構造では、エイリアスはまったく異なる方法で処理されます。

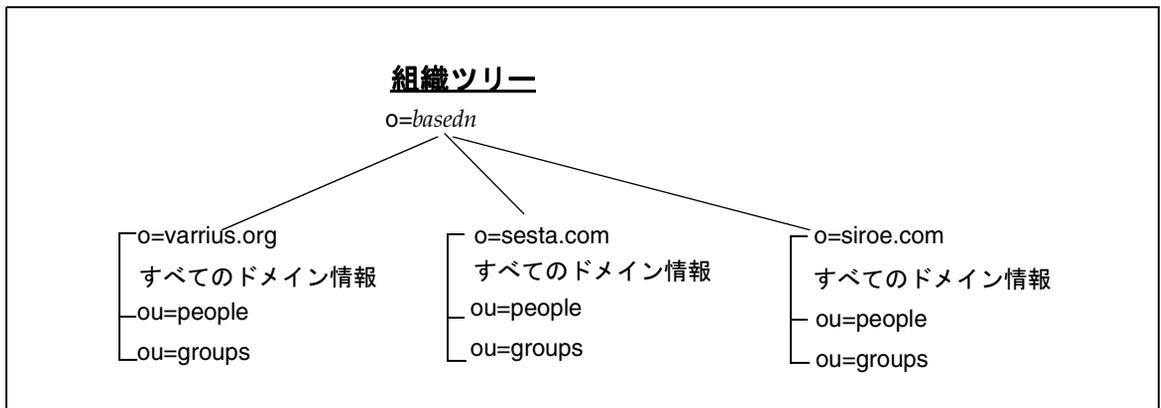
図 3-2 の下の半分の図は、1 ツリー LDAP 構造を示します。

図 3-2 2 ツリー LDAP 構造と 1 ツリー構造との比較

2 ツリー構造



1 ツリー構造



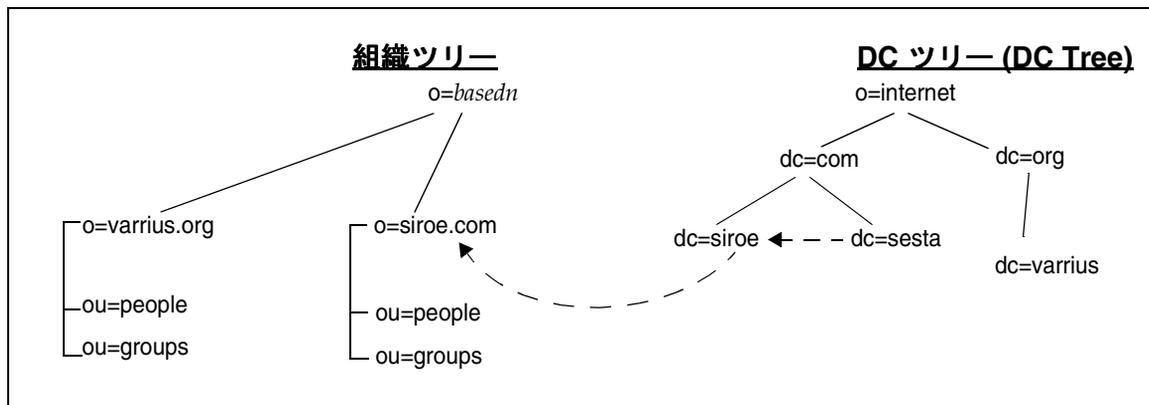
1 ツリー DIT 構造の利点

1 ツリー構造の Schema 2 ネイティブモードを使用する主な利点は以下のとおりです。

- 構造が Identity Server に統合される
- 構造が業界標準と緊密に整合される
- 構造が 2 ツリー構造よりも大幅に簡素化される

下図に示されているとおり、2 ツリー構造では一部のノードが組織ツリーのノードを直接指定しています (inetDomainBaseDN 属性を使用)。その他のノードは、組織ツリーノードを直接指定する代わりに、aliasedObjectName 属性を使用して別の DC ツリーノードを指定するエイリアスノードです。

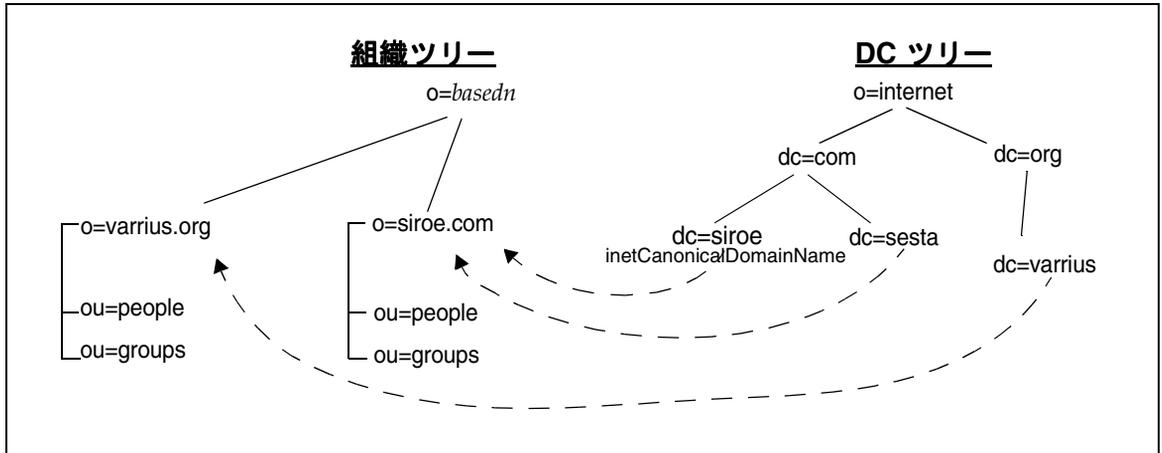
図 3-3 aliasedDomainName と inetDomainBaseDN を持つ 2 ツリーエイリアス



上図では、DC ツリーの sesta.com は、aliasedObjectName を使用して DC ツリーの siroe.com を指定し、siroe.com は、inetDomainBaseDN を使用して組織ノードのリンク名が付けられたノードを指定します。

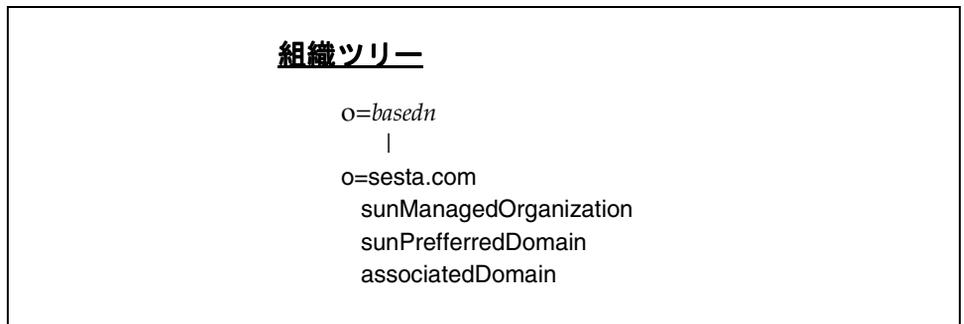
さらに、43 ページの図 3-4 に示されるように、DC ツリーには inetDomainBaseDN を使用して組織ツリーの同じノードを直接指定する 1 つまたは複数のノードが存在します。この場合、「実際」のドメイン名 (メールが実際に配置され、メールがルーティングされるドメイン) を指定するために、DC ツリーノードのいずれかに「連結遮断」属性 inetCanonicalDomainName が必要になります。

図 3-4 inetCanonicalDomainName 属性を持つ 2 ツリーエイリアス



それに対して、下図に示すように 1 ツリー構造は組織ツリーのみで構成されます。

図 3-5 associatedDomain を持つ 1 ツリーエイリアス



1 ツリー構造では、以前 DC ツリーにあったすべてのドメイン属性が組織ツリーのドメインノードに含まれます。各ドメインは、`sunManagedOrganization` オブジェクトクラスと DNS ドメイン名を含む `sunPreferredDomain` 属性によって識別されます。また、ドメインノードは、このドメインを識別するエイリアス名をリスト表示する 1 つまたは複数の `associatedDomain` 属性を持ちます。2 ツリー構造とは反対に、エイリアス名の重複ノードはありません。

1 ツリー DIT 構造は、組織固有のアクセス制御のためのデータを区分する方法として役立ちます。つまり、各組織は、ユーザーおよびグループエントリが配置される独立したサブツリーを DIT に持つことができます。このデータへのアクセスは、そのサブツリーの一部にあるユーザーに制限されます。これにより、ローカライズされたアプリケーションを安全に実行することができます。

さらに、Calendar Server または Messaging Server の新しい配備では、1 ツリー構造が既存の単一の DIT LDAP アプリケーションによりよくマッピングされます。

スキーマの要件

Communications Services 製品をインストールする前に、使用するスキーマについて理解する必要があります。スキーマとは、ディレクトリのエントリとして格納できる情報の種類を説明する一連の定義です。Communications Services では、Sun Java™ System LDAP Schema 1 と Sun Java™ System LDAP Schema 2 の 2 つのスキーマの選択肢が利用でき、サポートされています。以下の基準に従って、スキーマを選択します。

以下の場合に Schema 2 を使用します。

- ユーザープロビジョニングまたはシングルサインオン (SSO) のいずれかのために、Identity Server 6 2004Q2 (以前の Sun ONE Identity Server 6.1) を使用する場合
- はじめて、Communications Services コンポーネントをインストールする場合
- Communications Services を Portal Server などのほかの Java Enterprise System 製品と統合する場合

注 SSO を提供するために Identity Server 6 を使用する必要はありません。Schema 2 を選択すると、Identity Server 6 に依存しない信頼できるサークルタイプの SSO を使用することができます。

以下の場合に Schema 1 を使用します。

- たとえば、Messaging Server 5.2 をアップグレードする場合など、Communications Services コンポーネントの既存のバージョンを所有している場合
- ユーザーをプロビジョニングする必要はありません (Identity Server SSO が不要な場合)。
- ユーザーのプロビジョニングを行うために Delegated Administrator を使用する場合

スキーマの選択の詳細については、次の Web サイトの『Sun Java System Messaging Server 配備計画ガイド』を参照してください。

<http://docs.sun.com/db/prod/entsys?l=ja>

Directory Server の考慮事項

Sun Java System Directory Server は、イントラネット、ネットワーク、およびエクストラネット情報用に柔軟性のある、複数層データストレージを提供します。Directory Server は既存のシステムを統合し、社員、顧客、供給業者、およびパートナー情報を統合する中央リポジトリとして機能します。Directory Server を拡張して、ユーザープロフィールおよび設定とともに、エクストラネットユーザーの認証を管理することができます。

Portal Server、Identity Server、Messaging Server、Calendar Server、Instant Messaging Server のスキーマなど、すべてのカスタム LDAP スキーマは単一のディレクトリにインストールされます。

ビジネスの目的と予想される使用パターンに応じて、データ環境と考慮すべき多くの要素を設計する数多くの方法があります。ディレクトリ設計は以下の領域に対処する必要があります。

- ディレクトリスキーマとオブジェクトクラスの定義
- ディレクトリ情報ツリー (DIT)
- グループとメンバーの定義
- スタティックおよびダイナミックグループを含むアクセス制御リスト (ACL) の方針
- フェイルオーバー、レプリケーション、レフェラルを含む高可用性および高パフォーマンス対応のディレクトリアーキテクチャ
- ディレクトリの管理

データ環境を設計する方法に関するこれらの要素や提案の詳細な説明については、次の Web サイトの『Sun Java System Directory Server 配備計画ガイド』を参照してください。

<http://docs.sun.com/db/prod/entsys?l=ja>

Directory Server と Tier (層) アーキテクチャの考慮事項

単一層アーキテクチャから複数層アーキテクチャへの移行では、まず Directory Server を専用のマシンに「分割」する必要があります。負荷が一定の量を超えると、同じホストの Directory Server と Messaging Server は特有のパフォーマンスの影響を受けます。これは、Messaging Server が Directory Server で動作するように設計されることによります。Directory Server を専用のマシンに分離することが、配備のパフォーマンスを向上させる最初の手順です。

Tier (層) アーキテクチャの詳細については、次の Web サイトの『Sun Java System Messaging Server 配備計画ガイド』を参照してください。

<http://docs.sun.com/db/prod/entsys?l=ja>

Directory Server トポロジの考慮事項

単一マシンにインストールした Directory Server のインスタンスに配備を構築することは可能ですが、その他の Communications Services コンポーネントもコアサービスとして機能するディレクトリサーバーに依存します。したがって、通常の配備をするのではなく、冗長性のある可用性の高い構成の Directory Server の配備を計画する必要があります。

Directory Server の可用性を高めるための最初の手順は、マスターディレクトリサーバーのペアを確立することです。次に、複数マスターのレプリケーションを使用して、LDAP の書き込みスループットと可用性を向上させます。Sun Cluster を高可用性配備で使用する場合、2つの LDAP マスターがクラスター化されます。詳細については、[83 ページの「Directory Sever の高可用性化」](#)を参照してください。

Directory Server の容量計画

Directory Server の容量計画には確立されたルールはありませんが、パフォーマンス測定基準に適合するかどうかを確認するためにディレクトリサーバーを注意深く監視することは非常に重要です。システムがこれらの測定基準に適合しない場合は、増設ディレクトリコンシューマを追加します。通常、以下の点を監視します。

- 負荷のヒット数
- 負荷のキャッシュ数
- 秒あたりの要求数

目標応答時間 (10 ミリ秒) に対する上記測定値を評価します。IOWAIT は 10 ミリ秒を超えてはなりません。また、この層での CPU 利用率の合計が 70% を超えてはなりません。

Directory Server と Calendar Server の相互作用に関する考慮事項

Calendar Server は、Directory Server に格納されるユーザーエントリに対して複数の書き込みを行います。これらの大量の書き込みは、ユーザーが最初に Calendar Server にログインするときとユーザーが特定のアクションを実行するときに発生します。これらのアクションには、カレンダーの作成、カレンダーへの登録、設定の変更などがあります。これらのアクションを考慮しないと、Directory Master Server に大きな負荷を与える可能性があります。

ディレクトリのレプリケーションを使用すると、LDAP Master Server は LDAP レプリケーションサーバーにエントリを複製します。Calendar ユーザーがこれらのアクションのいずれかを実行する場合、Calendar Server は Master Directory Server への変更の書き込みだけを行うことができます。これは複製が読み取り専用のためです。

2 番目の相互作用に関する考慮事項は、これらの複製されたディレクトリ構造の中にあります。ユーザーが設定を変更する場合、変更が Master Directory Server から Calendar Server が使用する Directory Replica に正しく複製されるまでこれらの変更は正常に表示されません。この応答遅延を防止するために、Calendar Express (cshttpd) が変更をキャッシュするように設定して、この問題を回避することができます。

Directory Server と個人アドレスブックに関する考慮事項

Messenger Express クライアントは、個人アドレスブック (PAB) の概念をサポートします。これにより、ユーザーは Directory Server に個人用連絡先 (たとえば、業務用連絡先、友人、家族など) を格納することができます。ユーザーの PAB に新しい個人用連絡先が追加されるごとに、Directory Server に書き込みが行われます。これらのアクションを考慮しないと、ディレクトリレプリケーションの方針とは関係なく LDAP Master Server に大きな負荷を与える可能性があります。

User and Group Directory Server 上のパフォーマンスの問題を解決する 1 つの方法は、PAB 情報を別の Directory Server に配置することです。これにより、PAB の相互作用が LDAP Master Server に負荷を配置しなくても済むようになります。

Directory Server と Communications Express の考慮事項

Communications Express クライアントでは、Directory Server の設定の一部を変更する必要があります。Communications Express 設定には、Java Enterprise System 2004Q2 に付属して提供される `comms_dssetup.pl` スクリプトが必要です。このスクリプトは、Java Enterprise System 2003Q4 の一部として、Sun ONE Calendar Server 6.0 および Sun ONE Messaging Server 6.0 で提供されるものとは異なります。

Messaging Server の考慮事項

Communications Services の開発では、以下の Messaging Server コンポーネントのパフォーマンスを評価する必要があります。

- Message Store
- Message Transfer Agent (MTA)
- Mail Message Proxy (MMP)
- Messaging Express Multiplexor (MEM)

以下の節では、これらの各コンポーネントに対する考慮事項の概要を説明します。これらのコンポーネントのパフォーマンスと可能なハードウェアソリューションの詳細については、次の Web サイトの『Sun Java System Messaging Server 配備計画ガイド』を参照してください。

<http://docs.sun.com/db/prod/entsys?l=ja>

Message Store の考慮事項

メッセージストアのパフォーマンスは以下のさまざまな要素の影響を受けます。

1. ディスク I/O
2. 受信メッセージの頻度 (メッセージ配信頻度とも呼ばれる)
3. メッセージのサイズ
4. ログインの頻度 (POP/IMAP/HTTP)
5. トランザクションの頻度 (IMAP/HTTP)
6. さまざまなプロトコルの同時接続数
7. ネットワーク I/O

上記の要素は、Message Store に影響を与えるおおよその順番を示しています。Message Store の大部分のパフォーマンス問題は、不十分な I/O 容量から起ります。また、物理ディスクにストアを配置する方法もパフォーマンスに影響します。小規模なスタンドアロンシステムでは、ディスクの単純なストライプを使用して十分な I/O を装備することができます。大規模なシステムでは、ファイルシステムを分離して、ストアのさまざまな部分に I/O を装備します。

Message Transfer Agent (MTA) の考慮事項

MTA パフォーマンスは、以下のような数多くの要素の影響を受けます。

- ディスクパフォーマンス
- SSL の使用
- 送受信メッセージ数および接続数
- メッセージのサイズ
- 送信対象宛先およびメッセージの数
- MTA への接続および MTA からの接続速度と待ち時間
- スпамまたはウィルスのフィルタリングの必要性
- SIEVE 言語フィルタリングルールの使用およびほかのメッセージ解析を行う必要性 (変換チャネルの使用など)

MTA ルーターは CPU と I/O の両方を大量に使用します。MTA は、キューディレクトリとログ記録ディレクトリに 2 種類の異なるファイルシステムを使用します。MTA ルーターとして機能する小規模なホスト (4 プロセッサ以下) の場合、これらのディレクトリを別々のファイルシステムに分離する必要はありません。キューディレクトリは、かなり大量の書き込みが同期的に行われます。ログ記録ディレクトリは、一連の小規模かつ同期的な順次書き込みが行われます。

ほとんどの場合、ディスクサブシステムの MTA に冗長性を計画して、スピンドルが故障した場合にメールの恒久的消失を回避します (スピンドルの故障は、最も発生しやすいハードウェア障害)。これは、外部ディスクアレイまたは多くの内部スピンドルを持つシステムが最適であることを意味します。

さらに、MTA の配置に関しては、MTA は必ずファイアウォールに配置する必要があります。

MTA と LMTP の考慮事項

Messaging Server の LMTP サービスは、そのままでは、LMTP サーバーまたはほかの LMTP クライアントで動作するように設計されておりません。配備における LMTP と SMTP サーバーの混在の設定に関する詳細については、次の Web サイトの『Sun Java System Messaging Server リリースノート』を参照してください。

<http://docs.sun.com/db/prod/entsys?l=ja>

Mail Message Proxy (MMP) の考慮事項

MMP は、ログ記録以外にディスク I/O を使用することはありません。MMP は、完全に CPU およびネットワークにバインドされています。ほかのすべての Messaging Server コンポーネントと異なり、MMP はマルチプロセス、マルチスレッドではありません。主要な実行コードは、シングルプロセス、マルチスレッドです。したがって、MMP には十分なマルチプロセスがないため、ほかのコンポーネントと同様にスケールアップされることはありません。

MMP は 4 プロセッサを超えてスケールアップされることはなく、2～4 プロセッサの間で直線的にスケールアップされます。2 プロセッサ、ラックマウントマシンが MMP に適しています。

MMP と同じマシンにほかのコンポーネントソフトウェア (MEM、Calendar Server フロントエンド、Communications Express Web クライアント、LDAP プロキシなど) を配置するように選択した配備の場合は、大規模な 4 プロセッサ SPARC マシンを検討してください。この構成は、管理、パッチ、監視などが必要となるマシンの総数を低減します。

Messaging Express Multiplexor (MEM) の考慮事項

MEM は、Web メールクライアントにミドル層プロキシを提供します。このクライアントにより、ユーザーはメールにアクセスし、ブラウザを使用してメッセージを作成できるようになります。MEM の利点は、どのバックエンドサーバーがメールを格納しているかにかかわらず、エンドユーザーは MEM にのみ接続して電子メールにアクセスできることです。MEM は、ユーザーの LDAP 情報を介して HTTP セッション情報とユーザープロフィールを管理することにより、これを実現します。次の利点は、すべてのスタティックファイルと LDAP 認証の状態が Messaging Server のフロントエンドに配置されることです。この利点は、Message Store バックエンドからレンダリングされる Web ページに関連する追加 CPU 要件の一部を緩和することです。

MEM には、多くの MMP と同じ特性があります。MEM は 4 プロセッサを超えてスケールアップされますが、大部分の環境で、このようにスケールアップする特別な価値はありません。また、将来、Web メールコンポーネントは、Message Store から Web サーバーの Java サーブレットとして XML レンダリングを実行するアクセス層のマシンに負荷が移されることになっています。現在、Java サーブレットは 2 プロセッサを超えて拡大されません。したがって、ハードウェアの選択として SPARC または Intel の MEM 用 2 プロセッサマシンを計画するか、現在の 2 プロセッサ MEM ハードウェアに再度目的を持たせて、次世代ソリューションが発売されたときに小規模なマシンに置き換えることを考慮してください。

同じサーバーのセットに MMP と MEM を配置することができます。このようにする利点は、少数の MMP または MEM が必要になった場合に、冗長性のための追加ハードウェアの数が最小化されることです。同じサーバーのセットに MMP と MEM を配置する唯一の欠点は、あるプロトコルへのサービス拒否攻撃がほかのプロトコルに影響することです。

Calendar Server の考慮事項

Calendar Server は以下の 5 つの主要なサービスで構成されます。

- HTTP サービス (cshttpd) は HTTP 要求を待機する。ユーザー要求を受信して、呼び出し元にデータを返す
- 管理サービス (csadmin) は Calendar Server のインスタンスを必要とする。Calendar Server の認証と管理の単一ポイントを提供し、大部分の管理ツールを提供する
- 通知サービス (csnotify) は、電子メールまたは ENS (予定通知サービス) のいずれかを使用して予定と予定事項の通知を送信する
- ENS (予定通知サービス) (enpd) は、予定アラームのブローカとして機能する
- 分散データベースサービス (csdwpd) は、複数のデータベースサービスを同一の Calendar Server システムにリンクさせ、分散カレンダーストアを形成する

スケーラブルな Calendar Server 配備では、HTTP サービスと管理サービスのインスタンスをカレンダーのフロントエンドシステムとして一緒に配備します (マシンごとに 1 つの cshttpd インスタンスを配備します。各マシンでは、そのマシンの CPU ごとに 1 つの cshttpd プロセスを設定する)。通知サービス、予定通知サービス、分散データベースサービス、管理サービスを、カレンダーのバックエンドシステムとして一緒に配備します。

認証と XML/XSLT 変換は、大きな負荷を生成する 2 つのカレンダーサービスアクティビティです。サービス品質の要件に適合するために、増設 CPU を追加できます。

通常、カレンダーのバックエンドサービスは、カレンダーのフロントエンドサービスの 1/2 のサイズの CPU 数が必要です。カレンダーのフロントエンドシステムによるサービス品質をサポートするためには、カレンダーのバックエンドシステムがフロントエンド CPU の約 2/3 を使用する必要があります。

配備の初期段階で、カレンダーサービスをフロントエンドとバックエンドサービスに分割することを考慮します。

通常はフロントエンドサービスのコンポーネントである Calendar Server の HTTP プロセスが大半の CPU 時間を消費します。これは、カレンダーのピーク利用率を考慮し、予想される HTTP のピークセッションに対応する十分なフロントエンド処理能力を選択する必要があることを示しています。通常、Calendar Server フロントエンドは、冗長性、つまり複数のフロントエンドホストを配備することにより可用性を高めます。フロントエンドシステムはカレンダーの持続的データを保持しないので、Sun Cluster または Veritas のような HA ソリューションに適したシステムではありません。さらに、このようなハードウェアおよび管理諸経費の追加というソリューションは、Calendar Server フロントエンド用の HA の配備を、高コストで多くの時間を消費するものにしていきます。

注 真の HA ソリューションを保証できるカレンダーフロントエンドの唯一の構成は、Messaging Server MTA が配置されている同じホストにカレンダーフロントエンドを配備した場合です。ただし、この構成の場合でも、このようなソリューションの間接費が、わずかな利点に比べて不利にならないかどうか慎重に検討する必要があります。

Calendar Server フロントエンドの最適なハードウェアの選択肢は、シングルまたはデュアルプロセッサの SPARC または Intel サーバーです。マシンごとに 1 つの Calendar Server cshttpd プロセスを配備します。このような配備は、ある程度の初期クライアントの並行処理機能で開始し、既存の構成によるピーク利用率がわかった段階でクライアントのセッション容量の増加を可能にして、費用効率の高いソリューションを提供します。

複数のフロントエンドを配備する場合は、フロントエンドサービス全体に負荷を分散するために、ロードバランサ (スティッキで持続的な接続性のある) が必要になります。

注 Communications Express は 2 つのプロセッサを超えて拡大されません。以前に Calendar Server で説明した同じハードウェアの選択肢が、Communications Express の配備にも適用されます。

Calendar Server のバックエンドサービスはリソース消費のバランスがよくとれており、CPU または I/O (ディスクまたはネットワーク) のいずれにおいてもボトルネックは発生しません。したがって、バックエンド用ハードウェアに適した選択肢は、シングルストライプボリュームの SPARC サーバーとなります。このマシンは、大量のカレンダーのピーク負荷に対してかなりの容量を提供します。

要件に高可用性が含まれる場合は、バックエンドに持続的データが含まれるので、Sun Cluster または Veritas Cluster を使用し Calendar Server のバックエンドを配備するのが賢明です。

-
- 注 フロントエンドとバックエンド Calendar Server ホストの両方を持つ構成では、すべてのホストが以下を実行する必要があります。
- 同じオペレーティングシステム
 - パッチおよびホットフィックスリリースを含む Calendar Server の同一リリース
-

Instant Messaging の考慮事項

ほかの Communications Services コンポーネントと同様に、Instant Messaging をフロントエンド (Instant Messaging マルチプレクサ) とバックエンド (サーバーとストア) に分割するアーキテクチャを構成することができます。

Instant Messaging の考慮事項の詳細については、次の Web サイトの『Sun Java System Instant Messaging 配備計画ガイド』を参照してください。

<http://docs.sun.com/db/prod/entsys?l=ja>

Portal Server の考慮事項

Portal Server を持つ Communications Services 製品をインストールして、メッセージング、カレンダー、インスタントメッセージングアプリケーションにアクセスする「傘型」フロントエンドを提供します。Portal Server の統合には、Portal Server、Calendar Express Web クライアント、Messaging Express Web クライアント、Communications Express クライアント間のシングルサインオン機能が含まれます。さらに、ユーザーは、Portal Server デスクトップを使用して、Messaging Express、Calendar Express、Instant Messaging クライアントを利用することができます。

詳細については、次の Web サイトの『Sun Java System Portal Server Deployment Guide』と『Sun Java System Identity Server 配備計画ガイド』を参照してください。

<http://docs.sun.com/doc/817-5321>

<http://docs.sun.com/db/prod/entsys?l=ja>

Connector for Microsoft Outlook の考慮事項

この節では、Connector for Microsoft Outlook の配備の際に発生するいくつかの配備問題について説明します。詳細については、次の Web サイトの Connector for Microsoft Outlook のマニュアルを参照してください。

<http://docs.sun.com/db/prod/entsys?l=ja>

Sun Java System Connector for Microsoft Outlook は、Outlook を Sun Java Enterprise System のデスクトップクライアントとして使用できるようにします。Connector for Microsoft Outlook は Outlook のプラグインで、エンドユーザーのデスクトップにインストールする必要があります。Connector for Microsoft Outlook は、Messaging Server にフォルダの階層と電子メールメッセージを照会します。次に、この情報を Outlook が表示できる MAPI (Messaging API) プロパティに変換します。同様に、WCAP プロトコルを使用して Calendar Server に予定と作業を照会し、MAPI プロパティに変換します。このモデルによって、Connector for Microsoft Outlook は、Messaging Server のメールと Calendar Server のカレンダー情報の 2 つの別個の情報源からエンドユーザーの Outlook 表示を作成します。

Connector for Microsoft Outlook コンポーネント製品の依存性

Connector for Microsoft Outlook を使用するには、同一の配備に Sun Java System Messaging Server と Sun Java System Calendar Server が必要です。サポートされるバージョンに関する詳細については、これらの製品のリリースノートを参照してください。

Connector for Microsoft Outlook を正常に機能させるには、全体のパフォーマンスを向上させるために、以下の Sun Java System Directory Server の LDAP 属性に少なくとも実在と等価のインデックスを付ける必要があります。

- icsCalendar
- mail
- mailalternateaddress

製品の依存性の完全なリストについては、次の Web サーバーの『Sun Java System Connector for Microsoft Outlook リリースノート』を参照してください。

<http://docs.sun.com/db/prod/entsys?l=ja>

Sun ONE Calendar Server データの移行

Sun ONE Calendar Server 6.0 以前の Calendar Server のバージョンを使用している場合は、Sun Professional ServicesSM を利用して、データを新しい形式に変換し、移行する必要があります。この Calendar Server データの移行は Outlook を使用するために必要です。また、ストレージの基本的な変更と繰り返される予定の管理のために必要になります。新たな Sun Java System Calendar Server の顧客は、移行サービスは必要ありません。

Exchange Server データの移行

Connector for Microsoft Outlook は、Exchange Server の Microsoft Exchange Server メッセージを変換しません。Sun Professional Services を利用してデータを変換する必要があります。

Communications Express の考慮事項

Sun Java System Communications Express は、ブラウザを使用してアクセスできる Web サーバーベースのアプリケーションです。Communications Express 製品は、Calendar、Address Book、Mail の 3 つのクライアントモジュールで構成されます。Calendar、Address Book、Mail クライアントモジュールは、Web コンテナ上に単独のアプリケーションとして配備され、総称して Communications Express と呼びます。

Communications Express は以下の Sun Java System コンポーネント製品に依存します。

- Directory Server
- Identity Server (オプション)
- Calendar Server
- Messaging Server

Communications Express をフロントエンドサーバーとしてインストールします (複数層環境)。Communications Express を実行する同じホストに Messaging Server パッケージの完全なセットをインストールする必要があります。Messaging Server は、Message Store に直接アクセスする Messenger Express インタフェースを提供するか、または Messenger Express を実行するバックエンドに接続する MEM を提供するために使用されます。さらに、フロントエンドマシン上の Communications Express の

AddressBook サーバーを設定して、LDAP ディレクトリインフラストラクチャか Communications Express マシン以外の LDAP サーバーのいずれかにデータを格納することができます。詳細については、次の Web サイトの『Sun Java Communications Express 管理ガイド』を参照してください。

<http://docs.sun.com/db/prod/entsys?l=ja>

Communications Express は、Calendar Server のローカル (通常、同一マシン) の cshttp デーモンと通信します。次に、これらのデーモンは DWP を使用して Calendar Server のバックエンドと通信します。Communications Express は、Messenger Express (Web メール) を使用して Message Store と通信します。

ロードバランサまたはポートディレクタタイプのデバイスを使用する場合は、ユーザーがセッション中に同じフロントエンドサーバーに継続的にルーティングする「スティッキー」(持続的)な接続を使用してください。

セキュリティの考慮事項

Communications Services の企業配備のセキュリティは、「多重防御」手法を採用することによって管理されます。ネットワーク、ハードウェアのプラットフォーム、オペレーティングシステム、アプリケーション自体を個々に安全にすることによって、アーキテクチャの各層のセキュリティを確保します。セキュリティには、不必要なネットワークポートやアクセスメカニズムを閉鎖することによって各層を堅牢化する方法があります。また、インストールするソフトウェアパッケージの数を最小にして、システムが必要とするパッケージのみ利用できるようにします。最後に、ネットワーク内の意図しないアクセスから、層をセキュリティ保護するために層を隔離します。

Messaging Server のプロキシサーバーをインストールして、データセキュリティを補強することができます。背後にある Messaging Server のファイアウォールに配置されたプロキシサーバーは、Messaging Server 上の情報に対する攻撃を防御します。

Calendar Server は、数多くのセキュリティレベルを装備して、盗聴、無許可使用、または外部アタックからユーザーを保護します。セキュリティの基本的なレベルは認証の使用です。Calendar Server は、デフォルトで LDAP 認証を使用しますが、代替認証手段が必要な場合は認証プラグインの使用もサポートします。Identity Server との統合により、Calendar Server はシングルサインオン機能の利用が可能になります。

Instant Messaging は、複数の認証メカニズムと安全な SSL 接続によって通信の統合を可能にします。Portal Server と Identity Server との統合によって、追加セキュリティ機能、サービススペースのプロビジョニングアクセスポリシー、ユーザー管理、安全なリモートアクセスが可能になります。

セキュリティの詳細については、『Sun Java System Messaging Server 配備計画ガイド』、『Sun Java System Calendar Server 管理ガイド』、『Sun Java System Instant Messaging 配備計画ガイド』を参照してください。

注 完全なセキュリティ環境を確保するために、配備にはセキュリティ保護するホストの内部クロックを同期させる時間サーバーが必要です。

ネットワークのセキュリティ

水平方向のスケーラビリティとサービスのセキュリティの両方をサポートする推奨配備構成は、ファイアウォールの背後にアーキテクチャのアクセス層を配置することです。2層アーキテクチャでは、2つのファイアウォールを使用してDMZを作成します。これは、2番目のファイアウォールの背後にある内部ネットワークのメインサービス要素を保護しながら、情報配信要素、カレンダーおよびメッセージングフロントエンドへのアクセスを可能にします。また、このような構成は、アクセス層とデータ層の要素を個別にスケーリングして、トラフィックおよびストレージ要素に対応することができます。

オペレーティングシステムのセキュリティ

運用環境におけるセキュリティ違反の潜在的リスクを軽減するために、「システムの堅牢化」と呼ばれる以下の方法を実行します。

- 運用環境におけるインストールの最小化**：インターネットまたは信頼されないネットワークにさらされた環境で Sun サーバーをインストールする場合、ホストするアプリケーションのサポートに必要な最小パッケージ数に Solaris のインストールを減らします。サービス、ライブラリ、アプリケーションの最小化を実行することにより、保守が必要なサブシステムの数を減らして、セキュリティを強化することができます。

Solaris Security Toolkit は、Solaris 運用環境システムの最小化、堅牢化、セキュリティ保護を実行する柔軟性のある拡張可能なメカニズムを提供します。このツールキットの配備の背後にある主な目的は、Solaris システムのセキュリティを確保するプロセスを簡素化し、自動化することです。詳細については、次の Web サイト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の追跡と監視**：セキュリティの組み込みが必要なシステム内では、ファイル内の変更を追跡し、侵入を検出するためにファイル変更制御および監査ツールが不可欠です。Tripwire for Servers または Solaris Fingerprint Database (SunSolve オンラインで入手可能) などの製品を使用できます。

アプリケーションのセキュリティ

Communications Services 製品ポートフォリオは、業務用通信のセキュリティと統合を実現する機能を提供します。Communications Services は、以下のような幅広い「組み込み」セキュリティ機能を提供します。

- 認証
- メッセージとセッションの暗号化
- ウィルスとスパムの防護
- 通信のアーカイブと監査
- エンドユーザーが設定可能なプライバシーオプション

安全な接続の実装

Communications Services は、SSL/TLS、S/MIME、SAML などのセキュリティ標準をサポートします。SSL/TLS は、クライアントとサーバー間のすべての通信が暗号化されたセッション内で行われるようにします。Portal Server との統合によって、追加設定不要で使用可能な追加認証メカニズムとともにアプリケーション全体にわたってシングルサインオン機能を利用することができます。

公開鍵データセキュリティを実装する場合は、公開鍵インフラストラクチャと鍵の選択をサポートするメールクライアントを選択する必要があります。Communications Services 製品は、追加の設定なしで、このように暗号化されたメッセージの転送と保管に直ちに関わることができます。ただし、Web クライアントにはメッセージを作成し、デコードする機能はありません。

一般的に使用されるデータセキュリティのメカニズムは、さまざまなメッセージングエージェント間のデータ送信に使用する接続で SSL 暗号化を使用して、配線間（つまり、クライアントからサーバーまで）のデータだけを保護します。このソリューションは、公開鍵暗号化ほど完全ではありませんが、実装がはるかに容易で、数多くの製品とサービスプロバイダによってサポートされています。

どんな問題が解決するか。企業は、所有するコーポレートネットワークを制御し、そのネットワーク上で送信されるデータは社員以外の者からは安全であるとみなしています。コーポレートネットワークの外部から企業のインフラストラクチャを使用して送信されるメールは、暗号化接続を介して企業のネットワークにデータを送信します。同様に、コーポレートネットワークの外部の企業ユーザーが受信するすべてのメールは、暗号化接続を介して送信されます。したがって、内部ネットワークの安全に関する企業の仮定が正しく、社員が自分自身とほかの社員間の転送用に認定されたサーバーだけを使用する場合には、社員間のメールは外部攻撃から安全に保護されています。

このソリューションでは何が解決されないか。まず最初に、この方法は、企業の内部ネットワークにアクセスする意図しない受信者による意図しないデータの閲覧からデータを保護しません。次に、社員と外部パートナー、顧客、または供給業者間で送信されるデータの保護が行われません。データは、完全にセキュリティ保護されていない状態でパブリックインターネット間を移動します。

ただし、この問題は、企業と顧客の両方のネットワークの MTA ルーター間の SSL 暗号化設定によって修正することができます。この種類のソリューションは、使用する各プライベート接続の設定が必要になります。このためには、メールを介して送受信する顧客またはパートナーのデータにセキュリティのための重要な層を追加します。Communications Services の MTA と SSL を使用することにより、企業は送信手段としてパブリックインターネットを使用してコストの節約ができますが、MTA はパートナーの SSL を使用しなければなりません。このソリューションは、パートナーとの間のほかのトラフィックを考慮しておりません。ただし、通常、メールはトラフィックの大きな部分を占めており、企業は転送されるデータに基づいて料金を支払うことができるので、パブリックインターネットの使用はコストが少なく済みます。

2 つの異なる認証局 (CA) を使用する安全な接続の実装

サーバーとクライアント間、たとえば、Messaging Server から配備したほかのサーバー間に SSL 接続を実装することができます (Web Server、Calendar Server、Directory Server も同様)。必要に応じて、サーバー用とクライアント用に 2 つの認証局 (CA) を使用することができます。

このシナリオでは、ある CA を使用してサーバー証明書を発行し、別の CA を使用してクライアント証明書を発行します。クライアントにサーバーの証明書が真正なものであることを承認させたい場合は、サーバーの CA 証明書をクライアントの証明書 DB にロードする必要があります。サーバーにクライアントの証明書が真正なものであることを承認させたい場合は、クライアントの CA 証明書をサーバーの証明書 DB にロードする必要があります。

Communications Services 論理アーキテクチャ の開発

この章では、Communications Services 論理アーキテクチャを開発する方法について説明します。

この章には、以下の節があります。

- [Communications Services Enterprise Deployment 論理アーキテクチャの概要](#)
- [水平方向のスケーラビリティ戦略](#)
- [その他の配備の課題](#)

Communications Services Enterprise Deployment 論理アーキテクチャの概要

Communications Services を単一層または 2 層論理アーキテクチャのいずれかに配備することができます。論理アーキテクチャの特定によって必要なマシンのタイプと台数が決まるため、論理アーキテクチャは非常に重要です。

通常、一般企業への配備は単一層となり、ISP および電気通信会社への配備は 2 層となる傾向があります。ただし、一般的な場合と同じように例外が存在します。小規模な ISP が単一のマシンに配備することがあり、大規模な中央集中型の企業が、多くの場合 ISP が配備するのと同じ理由で 2 層アーキテクチャに配備することもあります。企業は、遠隔地で働く社員のアクセスを容易にしようとすればするほど、ISP と同様の配備を考慮することになります。

この節では、以下の Communications Services 論理アーキテクチャについて説明します。

- **単一層** : すべてのサービスが十分なメモリと CPU を装備した単一のホストに配置されるか、または各サーバーが特定のコンポーネント製品のすべてのサービスをホストする複数のサーバーに配置されます。
- **2 層** : コンポーネント製品のアクセス層とデータ層が別個のサーバーに分離されず。
- **エッジ** : 2 層アーキテクチャ上に構築され、インターネットを介してモバイル通信を利用する社員にセキュリティで保護された接続を提供します。

Communications Services を単一層に配備するか複数層に配備するかを決定する場合は、両方のモデルの長所と短所を理解する必要があります。

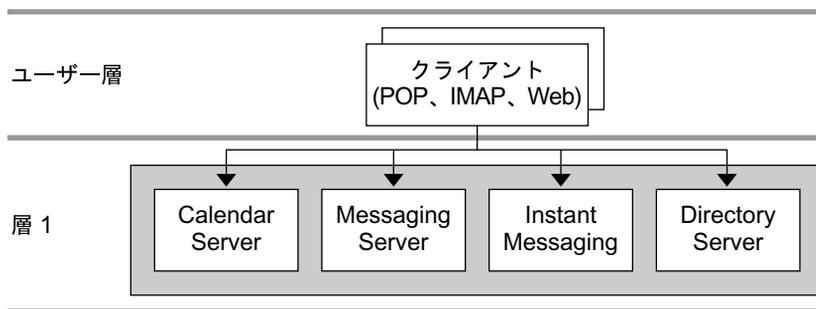
単一層、単一ホスト論理アーキテクチャ

その名前が示すように、単一層、1 ホスト論理アーキテクチャはすべてのサービスを単一のマシンに配置します。通常、このアーキテクチャは以下のような企業に最適です。

- ユーザー数が 500 人以下の場合
- 地理的に分散していない場合
- 少数の管理者によって管理される場合
- エントリレベルの構成が必要な場合

次の図は、単一層、1 ホスト論理アーキテクチャを示します。

図 4-1 単一層、1 ホスト論理アーキテクチャ



Outlook および Messenger Express などエンドユーザークライアントプログラムがユーザー層を形成します。層 1 は、メッセージング、カレンダー、インスタントメッセージング、ディレクトリなどすべてのサービスを実行する単一のマシンです。単一層配備の特徴は、エンドユーザーがプロキシやほかのエージェントを介さずに直接ストアと通信することです。

単一層、単一ホスト論理アーキテクチャでは、十分な CPU、メモリ、およびストレージを備えるマシンが必要になります。このタイプの配備に使用する企業のニーズに最も適合したマシンを決定するには、Sun の販売代理店と共同で作業を行う必要があります。

単一層、1 ホストアーキテクチャを実装する場合、サービスに論理名を割り当てることによって Tier (層) アーキテクチャに拡張できるように配備を構成することができます。この構成は、DNS マッピングを利用して、ユーザーが同一のフロントエンドプロセス (マシン) を使用するよう指示します。将来、サービスを Tier (層) 方式に分割するなど、拡大に対応するための変更が必要になった場合、ユーザーはクライアントアプリケーションを再設定する必要がありません。詳細については、[75 ページの「論理サービス名の使用」](#)を参照してください。

単一層、複数ホスト論理アーキテクチャ

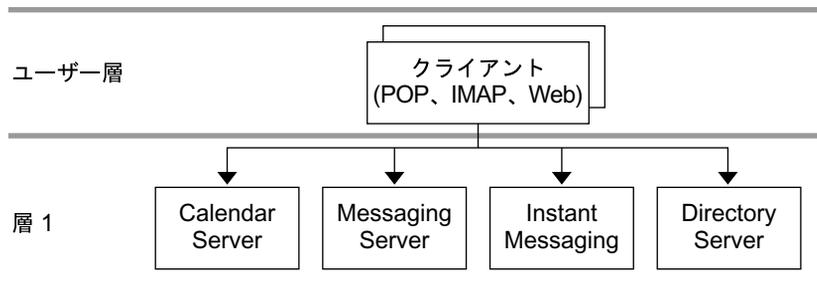
単一層、複数ホスト論理アーキテクチャは、各サーバーがコンポーネント製品に特有のサービスを実行するサーバーのセットです。たとえば、Messaging Server ホストはすべての Messaging Server サービスを実行するようにインストールおよび設定され、Calendar Server ホストはすべての Calendar Server サービスを実行するようにインストールおよび設定されます。ほかのホストも同様になります。このアーキテクチャは高可用性を確保するために設定される場合もあります。

単一層論理アーキテクチャの特徴は、エンドユーザーがプロキシやほかのエージェントを介さずに直接データストアと通信することです。たとえば、Messaging Server では、ユーザーは MMP または MTA を経由せずにルーティングされます。単一層論理アーキテクチャは、サーバー間のメールのルーティングまたは企業ネットワークへの出入りにスタンドアロン MTA ルーターを使用する場合がありますが、エンドユーザーはユーザーのメッセージストアの MTA にメールを送信します。MMP はメッセージストアへのイントラネット接続には関与しません。

同じ考え方が Calendar Server と Instant Messaging の両方に当てはまります。単一層論理アーキテクチャでは、フロントエンドプロセスは別個のマシンには配置されません。

[64 ページの図 4-2](#) は、Communications Services の単一層、複数ホスト論理アーキテクチャを示します。

図 4-2 単一層、複数ホスト論理アーキテクチャ



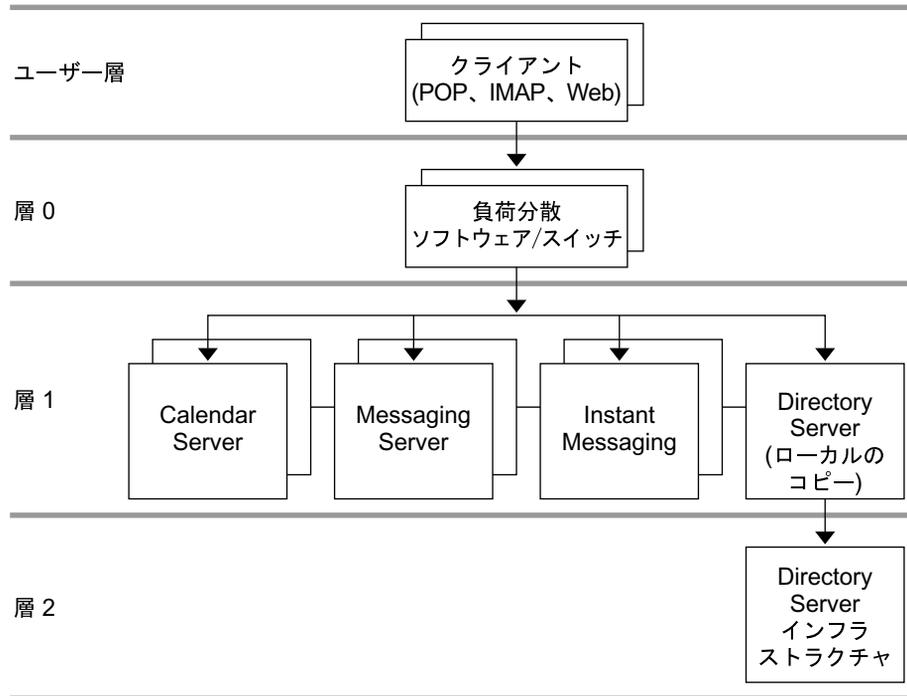
上図では、Outlook および Messenger Express などエンドユーザークライアントプログラムがユーザー層を形成しています。層 1 は 4 種類のサーバーのセットです。1 番目のサーバーは Calendar Server プロセスを実行し、2 番目は Messaging Server プロセスを実行し、3 番目は Instant Messaging プロセスを実行し、4 番目は Directory Server プロセスを実行します。

単一層、分散論理アーキテクチャ

単一層、分散論理アーキテクチャは、2つの層に Directory Server が配備された単一層の変形です。この形式の配備は、地理的に分散した小規模な部門または組織を持つ企業に適しています。各部門またはオフィスは、独自のサービス（メール、カレンダー、インスタントメッセージング）とローカルディレクトリインスタンス（コンシューマ）を持ちます。すべてのローカルディレクトリインスタンスがキャッシュされますが、集中化された企業マスターリポジトリとは同期します。この配備は、低帯域幅の接続を使用するオフィスでは一般的なシナリオです。ディレクトリは 2 層形式で設計され、データをローカルに保持するために低帯域幅を経由して複製されます。

65 ページの図 4-3 は、Communications Services の単一層、分散論理アーキテクチャを示します。

図 4-3 単一層、分散論理アーキテクチャ



上図では、Outlook および Messenger Express などエンドユーザークライアントプログラムがユーザー層を形成しています。層 0 は、層 1 を通じて負荷を分散するロードバランサを構成します。層 1 は、Communications Services プロセスの複数サーバーのセットです。複数のサーバーが Calendar Server サービスを実行し、複数のサーバーが Messaging Server サービスを実行し、複数のサーバーが Instant Messaging サービスを実行します。Directory Server は、ローカルの複製されたディレクトリのコピーを実行する層 1 のコンシューマサーバーとディレクトリのマスターコピーを持つ層 2 のもう 1 つのサーバーに分割されます。この配備形式では、クライアントの照会はマスターコピーではなく、ローカルのディレクトリコピーに送信されることに注意してください。ローカル Directory Server だけが、マスター Directory Server と通信します。

注 インターネット接続を使用する単一層を配備する場合は、別のアクセス層を使用します。たとえば、SSL を使用せずにイントラネット内部からデータストアにアクセスするように指示します。ただし、インターネットからデータストアにアクセスする場合は、SSL を介するアクセス層を経由するように指示します。これにより、インターネットから分離されたアクセス層に、データストアの SSL 負荷の大部分がオフロードされます。

この形式の配備の欠点は、企業のイントラネット上のサーバーを利用したり、インターネットからサーバーにアクセスしたりするユーザーが常時 SSL を使用するようにクライアントアプリケーションを設定する必要があることです。これは、SSL の有効、無効を切り替える手間が非常にかかるためです。このため、かなりの割合の SSL トラフィックがストアに直接接続されることとなります。イントラネット内部のアクセス層を利用することによって、この問題を解消し、接続の方向を制限することによって、不正なアクセスからイントラネットを保護することができます。

2 層論理アーキテクチャ

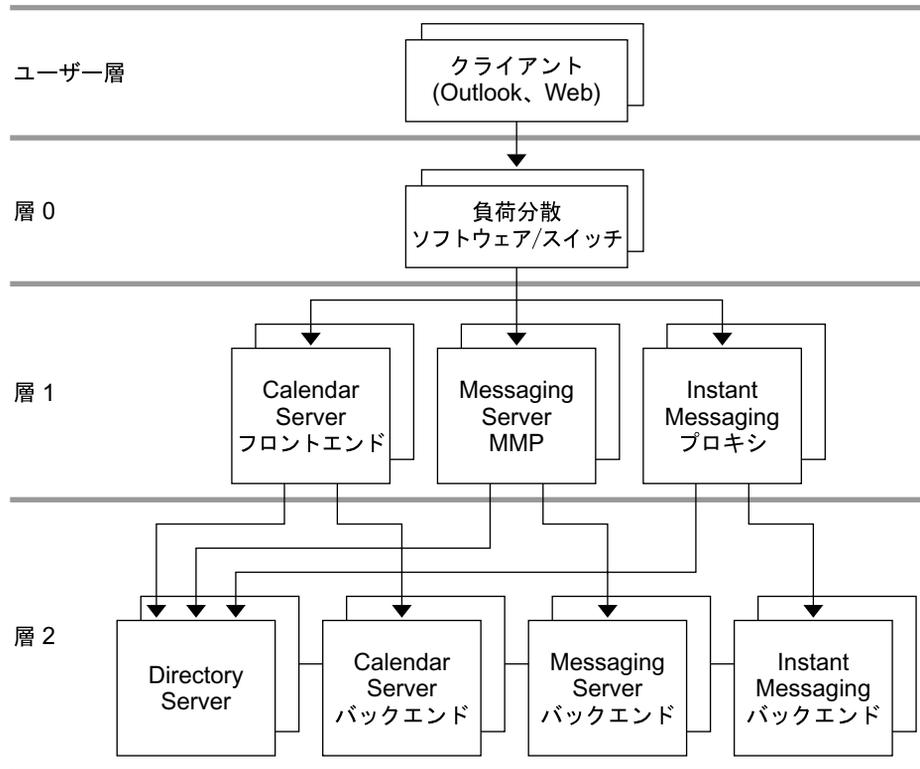
2 層論理アーキテクチャでは、データストアはフロントエンドプロセスを経由して通信します。つまり、Messaging Server の場合、MMP、MEM、および MTA がデータストアプロセスとは別のマシンに配置されることを意味します。この結果、メールストアは重要な共通タスクの負荷が軽減され、メールの受信と配信に専念することができます。Calendar Server の場合、HTTP サービスと管理サービスがストアプロセスと別のマシンに配置されることを意味します。Instant Messaging の場合、プロキシサービスがバックエンドプロセスと別のマシンに配置されることを意味します。

いくつかのレベルで、ほかのサーバーとの共存が可能です。たとえば、同一のマシンにカレンダーストアとメッセージストアを配置することができます。同様に、MMP マシンにカレンダーのフロントエンドを配置することができます。

2 層論理アーキテクチャでは、通常 Directory Server は、負荷分散された一連のコンシューマディレクトリに対する複数のマスターとレプリケーションを持つ、それ自体複雑な配備となります。

67 ページの図 4-4 は、Communications Services の 2 層論理アーキテクチャを示します。

図 4-4 2 層論理アーキテクチャ



上図では、Outlook および Messenger Express などエンドユーザークライアントプログラムがユーザー層を形成します。ロードバランサが層 0 を形成します。Calendar Server、Messaging Server、Instant Messaging フロントエンドが層 1 を形成します。最後に、Directory Server、Calendar Server、Messaging Server、Instant Messaging バックエンドが層 2 を形成します。このアーキテクチャでは、設計の全体的な柔軟性を高めるために、層 1 と層 2 要素を独立したインスタンスとして配備することができます。さらに、個々のインスタンスに別々の機能を割り当てることにより、システムのセキュリティを強化することができます。

通常の配備では、メッセージングとカレンダーフロントエンドをネットワークの非武装地帯 (DMZ) に配置し、ファイアウォールを経由してメインのメッセージングとカレンダーサービスに接続します。この構成により、層 1 の要素を個々にスケーリングすることができるため、システムの水平方向のスケーリングが可能になります。バックエンドサーバーの容量を超えて、これらの要素をスケーリングしてはなりません。

フロントエンドの要素がバックエンドサーバーの容量に到達した場合、バックエンドの層2の要素を拡大して、より多くのユーザーをサポートすることができます。通常、フロントエンドはトラフィックの関数としてスケーリングします。バックエンドはユーザー数の関数としてスケーリングされます。

注 単一層または2層アーキテクチャのコンポーネントのサイズ処理に関する具体的な手順については、サービスの専門技術者にお尋ねください。

エッジ論理アーキテクチャ

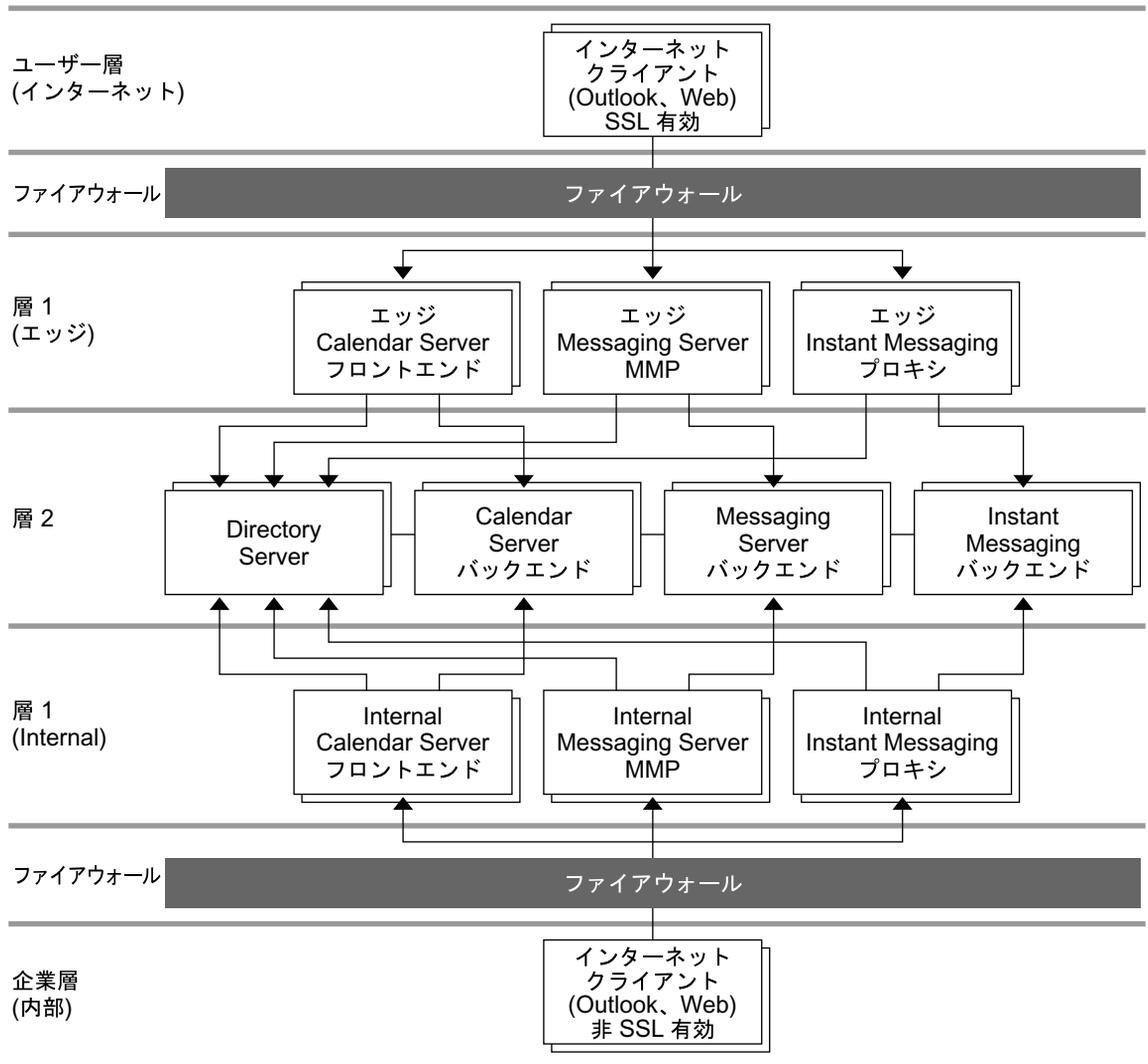
エッジ論理アーキテクチャは2層論理アーキテクチャへのリモートアクセスに対するセキュリティを向上させます。エッジ配備は、名前およびパスワード認証 (SMTPAuth) のみを使用することにより、遠隔地のモバイル通信を利用する社員にパブリックインターネットを経由するアクセスを許可します。メッセージは、パブリックインターネットを介して企業ネットワークと通信されるので、SSL を使用して暗号化されます。仮想プライベートネットワークは一切関与しません。通信の内部転送は、最大のパフォーマンスを発揮するために「平文」で行われます。アクセスは、無許可の侵入を防止するために、配備の「エッジ」に含まれます。

エッジを配備するビジネス上の理由は次のとおりです。

- 社員がモバイル通信を利用する遠隔地の社員で構成されていること
- すべてのリモートサイトに Communications Services サーバーをインストールし、維持することを避けたい

69 ページの図 4-5 は、Communications Services のエッジ論理アーキテクチャを示します。

図 4-5 エッジ論理アーキテクチャ



上図では、データストアは「エッジ」および「内部」フロントエンドサーバーにのみ接続される安全なプライベートネットワークの層 2 に配置されています。リモートクライアントは SSL を使用してフロントエンドサーバーに接続します。内部アクセスは本質的に安全であるとみなされるので、内部クライアントは SSL を使用して接続する必要はありません。

エッジアーキテクチャ設計の推奨事項

- エッジ層の容量計画を一般化することは困難です。容量計画を開発するには、配備用の機器を供給するハードウェアベンダーおよびソフトウェアベンダーと共同で行う必要があります。ただし、サイトのエッジ層に RBL (Realtime Blackhole List) を実装する必要があります。RBL は、スパムの拡散を防止するために、所有者が認証を拒否する IP アドレスのリストです。
- 最小応答時間 (エッジ層全体を通じて 1 ミリ秒以下) のエッジ層を設計します。
- CPU の利用率、またはアクティブな接続数によって負荷を検知する負荷分散アルゴリズムを使用します。ラウンドロビンを負荷モデルとして使用することはできません。MTA (状態を持たない) の例外を除いて、スティッキビット負荷分散を使用します。
- Web メールインタフェースが Web メールサーバー全体の状態を共有しないため、Web メールクライアントにはスティッキビットを管理できる負荷分散が必要になります。

単一層アーキテクチャの利点

論理アーキテクチャとして単一層を使用する利点は、追加ハードウェアの購入と維持が不要になることによるコストの低減です。

単一論理アーキテクチャが企業に最適かどうかを決定するには、以下の質問に対する回答が役立ちます。

- サービス拒否の脅威は最小か
- SSL は必要ないか
- 利用可能な相当数の保守ウィンドウがあるか

これらに該当する場合、企業が単一層論理アーキテクチャを使用できることを示します。

2 層アーキテクチャの利点

Communications Services 製品内のすべてのサービスはネットワーク機能に依存します。2 層アーキテクチャは、パブリック (ユーザー側) ネットワークとプライベート ネットワーク (データセンター側) の 2 つの独立したネットワークを持つネットワーク設計を提供します。アプリケーション層のアプリケーションホストによって、これらの 2 つのネットワークの分離が可能になります。

ネットワークを 2 層に分離することにより、以下の利点が提供されます。

- **内部ネットワークの非表示** : パブリック (ユーザー側) ネットワークとプライベート (データセンター側) ネットワークを分離して、データセンターの情報を非表示にすることにより、セキュリティが確保されます。この情報には、IP アドレスおよびホスト名などのネットワーク情報とメールボックスおよびカレンダー情報などのユーザーデータが含まれます。
- **ネットワーク情報の冗長性の提供** : 複数のフロントエンドマシンにわたるサービスへのアクセスを装備することにより、システムの冗長性を提供します。冗長性のあるメッセージングフロントエンドサーバーを追加して SMTP 要求を利用可能なメッセージングフロントエンドホストに負荷分散することにより、サービスの稼働時間を向上させることができます。
- **アクセス層のホスト上の利用可能なデータの制限** : アクセス層のホストを複合化することにより、攻撃者がアクセスホストの重要なデータを取得できないようにします。
- **タスクのアクセス層への軽減** : アクセス層が数多くのタスクを完全に所有できるようにすることにより、メッセージストア上のユーザーメールボックスの数が増加します。この方法が役立つのは、アクセス層マシン (第 2 層) よりもストアサーバーの購入および保守費用の方がコストが高くなるためです。通常、アクセス層マシンは小規模で大量のディスクが不要であり (50 ページの「[Message Transfer Agent \(MTA\) の考慮事項](#)」を参照)、ほとんどバックアップされることはありません。第 2 層によって軽減される機能の一部は次のとおりです。
 - サービス拒否の防護
 - SSL
 - 逆引き DNS
 - UBE (スパム) とウィルススキャン
 - 初期認証 : メッセージストアの認証が常に成功し、ディレクトリサーバーが最近のエントリをキャッシュしやすくなります。
 - LMTP: MTA リレーとメッセージストア間の LMTP のサポートにより、SMTP 処理が軽減され、メッセージストア上の MTA キューへのメッセージの追加書き込みが不要になります。

- クライアントアプリケーションのエンドユーザー設定の簡素化** : 2 層アーキテクチャの使用により、エンドユーザーはメッセージングとカレンダーアプリケーションが接続するホストの物理名を記憶する必要がありません。アクセス層のアプリケーションホストは、割り当てられたメッセージングまたはカレンダーデータセンターのホストにエンドユーザーを接続するプロキシを提供します。IMAP などのサービスは、ユーザーのメールボックスホストの名前を識別するために、LDAP 情報を使用してバックエンドサービスに接続されます。カレンダーサービスの場合、カレンダーのフロントエンドホストは割り当てられたユーザーのカレンダーストアホストへのバックエンド接続を構築するために、ディレクトリサーバーを使用してカレンダー検索を提供します。

この機能は、すべてのエンドユーザーがクライアント設定に同じホスト名を使用できるようにします。たとえば、ユーザーはメッセージストアが host-a であると記憶するのではなく、単に mail 設定を使用するだけで済みます。MMP は、ユーザーの割り当てられたメッセージストアへのプロキシサービスを提供します。すべてのメールの受信接続が 1 つ (または複数) の MMP をポイントするように DNS と負荷分散設定を装備する必要があります。

Calendar Server を 2 層に配置することにより、複数のカレンダーバックエンドサーバーを使用できます。Calendar Server のグループスケジューリングエンジンにより、ユーザーは任意のバックエンドカレンダーホストのカレンダーを使用するユーザーのアポイントメントを予定に入れることができます。

このプロキシ機能の追加の利点は、地理的に分散したユーザーが、物理的な場所に関係なく同じクライアントアプリケーションを利用できることです。ヨーロッパのユーザーがカリフォルニアを訪問したと仮定した場合、ユーザーはカリフォルニアのアクセスサーバーに即座に接続することができます。ユーザーの LDAP 情報は、ユーザーの代わりにヨーロッパにあるユーザーのメッセージストアへの別個の接続を確立するようにアクセスサーバーに指示します。

最後に、この機能は、ユーザーがブラウザを別に設定することなく、つまりユーザーのサポートを簡素化して大規模な環境の実行を可能にします。ユーザーに連絡せずに、あるいはデスクトップを変更せずに、ユーザーのメールボックスのあるメールボックスから別のメールボックス移動することができます。

- データセンターのネットワーク HTTP トラフィックの削減** : MEM (Messaging Express Multiplexor) とカレンダーフロントエンドはデータセンターネットワークへの HTTP トラフィックを大幅に削減します。HTTP はコネクションレス型サービスです。各 HTML の要素の場合、メールまたはカレンダーサービスに別々の HTTP 要求を送信する必要があります。これらの要求は、イメージ、スタイルシート、JavaScript™ ファイル、HTML ファイルなどのスタティックデータに対するものであることがあります。これらの要素をエンドユーザーの近くに配置することにより、バックエンドデータセンターのネットワークトラフィックが削減されます。

水平方向のスケラビリティ戦略

スケラビリティは、コンピュータ資源を最高の費用効率で利用し、ピーク時の作業負荷を処理し、ビジネスの拡大に対応してすばやくインフラストラクチャを拡張する必要がある企業に不可欠です。以下の点に注意してください。

- 増加する作業負荷に対するシステムへの影響：提供するパフォーマンスや作業負荷の増加に伴って、システムがクラッシュするか、パフォーマンスが徐々に低下する。
- 増大するユーザーの要求に応えるために、システムまたはネットワークにプロセッサ、CPU、ストレージ、I/O リソースを簡単に追加できるか。
- ローエンドシステムからミドルレンジサーバーおよびメインフレームクラスのシステムに拡大するときに、同一の環境でアプリケーションをサポートできるかどうか。

2層アーキテクチャに配備する場合、**Communications Services** 製品は水平方向に非常に効果的なスケリングが可能で、各機能要素は、特定の層に増設マシンを追加することにより、増大する負荷をサポートすることができます。

実際には、フロントエンドサービスとバックエンドサービスのスケリングは若干異なります。

層1の要素の場合、フロントエンドへのトラフィックが現在の容量を超えて増加したときにスケリングプロセスを開始します。比較的 low コストのマシンを追加し、これらのマシン全体の負荷分散を追加します。この結果、ロードバランサにシステムの全体負荷、サービスの分散、およびスケラビリティ要求の指示をすることにより、層1の各サービス機能に優先付けをすることができます。

層2の要素の場合、バックエンドサービスがユーザー容量またはデータ容量を超過したときにスケリングプロセスを開始します。一般的な基準としては、層1のサービスの負荷容量のちょうど2倍以下に対応できるように層2のサービスを設計します。

たとえば、5,000 ユーザー用に設計したアーキテクチャの場合、層1のフロントエンドサービスは5,000 ユーザーをサポートするように設計されています。バックエンドサービスは、この2倍の10,000 ユーザーに対応できるように設計します。システム容量が5,000 ユーザーを超えている場合は、フロントエンドサービスを水平方向に拡大することができます。全体容量が5,000 ユーザーに到達している場合は、バックエンドサービスに対応できるように拡大することができます。このような設計は、拡大がユーザーに関するものかスループットに関するものかによって、企業の拡大に柔軟に対応できるようにします。

その他の配備の課題

この節では、いくつかの共通の Communications Services 配備の最良の方法とその他の配備に関する考慮事項について説明します。

Messaging Server 用の LMTP の実装

最良の方法は、メッセージ配信のために SMTP の代わりに LMTP を実装することです。LMTP アーキテクチャは、システムの信頼性を高め、新しいサービスを実装するように配備を位置付ける堅牢かつ効率的なソリューションを提供すると同時に一部のストレージ IOP の要求を取り除きます。

MTA の LMTP サーバーがバックエンドメッセージストアへの配信に関して効率性が高い理由は次のとおりです。

- ストアにかかる負荷が軽減されます。リレーは水平方向に拡張できますが、ストアは水平方向に拡張できません。したがって、リレーの処理性能を可能な限り高めることが最善の方法です。
- MTA のキューをストアから引き出すことにより、IOP を 30% 削減することができます。
- LDAP サーバーにかかる負荷が軽減されます。
- 大規模なメッセージング展開では、LDAP インフラストラクチャが制限要因となることがよくあります。
- メッセージキューの数が減少します。

LMTP を実装するには 2 層アーキテクチャが必要です。LMTP の設定手順については、次の Web サイトの『Sun Java System Messaging Server 管理ガイド』を参照してください。

<http://docs.sun.com/db/prod/entsys?l=ja>

Realtime Blackhole List (RBL) の実装

Mail Abuse Protection System の Realtime Blackhole List (MAPS RBL) は、送信元の IP アドレスで識別される周知の UBE 送信元の動的な更新リストです。Messaging Server SMTP サーバーは RBL の使用をサポートし、不特定多数宛メール (UBE) またはスパムとして RBL が識別する送信元からの受信メールを反映することができます。

すべての配備において RBL の実装を考慮する必要があります。通常、MTA の前に配備された高品質の RBL は、最低でも 10%、場合によってはそれ以上 MTA に対するトラフィックを軽減します。

RBL と BrightMail などのアンチスパムまたはアンチウィルスサーバーは連携して動作することができます。たとえば、アンチスパムサーバーが特定の IP アドレスからの 100 通の電子メールから 95 ～ 99 通のメールを排除した場合、その IP アドレスを RBL に追加することができます。また、BrightMail 分析の実行時に、BrightMail の誤検知によって RBL を調整することができます。この結果、UBE の特定の変動を処理する上で、RBL を一層有効に活用できるようになります。

MTA ディスパッチャの ENABLE_RBL オプションの設定に関する詳細については、次の Web サイトの『Sun Java System Messaging Server Administration Reference』を参照してください。

<http://docs.sun.com/doc/817-6267>

論理サービス名の使用

Communications Services の論理名の使用する配備を設計します。将来の拡張や拡大を容易にするために、単一システムの配備においても論理名を使用する必要があります。論理名の使用は、DNS の格納以外に追加の配備設定費用は不要です。

これらの論理名を次の 2 つのカテゴリに分けて考えることができます。1 つは電子メールクライアントプログラムの設定などエンドユーザーに影響を与え、もう 1 つは受信 SMTP サーバーなどバックエンドの管理に影響を与えるものです。

次表でこれらの論理名について説明します。

表 4-1 ユーザー側の論理名

例	説明
mail.siroe.com	エンドユーザーが電子メールを収集するサーバーの名前
imap.siroe.com	エンドユーザーが電子メールを収集する IMAP サーバーの名前
pop.siroe.com	エンドユーザーが電子メールを収集する POP サーバーの名前

表 4-1 ユーザー側の論理名 (続き)

例	説明
smtp.siroe.com	送信メールサーバーとしてユーザーが指定する SMTP サーバーの名前
webcal.siroe.com または ce.siroe.com	Communications Express (以前の Calendar Express) サーバーの名前

表 4-2 保守レベルの論理名

例	説明
relay-in.siroe.com	受信 SMTP サーバーのバンクに相当
relay-out.siroe.com	送信 SMTP サーバーのバンクに相当
mmp.siroe.com	MMP サーバーのバンクに相当
mem.siroe.com	MEM サーバーのバンクに相当
storeAA.siroe.com	バックエンドメッセージストア。たとえば、storeAA.siroe.com ~ storeZZ.siroe.com のように、使用するトポロジで動作するネーミング方式を選択する
calstoreAA.siroe.com	バックエンドカレンダーストア。たとえば、calstoreAA.siroe.com ~ calstoreZZ.siroe.com のように、使用するトポロジで動作するネーミング方式を選択する

表 4-3 保守レベル論理名のユーザーレベルへのマッピング

保守レベル	ユーザーレベル
relay-in.siroe.com	なし
relay-out.siroe.com	smtp.siroe.com
mmp.siroe.com	mmp.siroe.com、pop.siroe.com、imap.siroe.com の 1 つまたは複数の名前
mem.siroe.com	webmail.siroe.com
storeAA.siroe.com ~ storeZZ.siroe.com	なし。エンドユーザーには隠されている

表 4-3 保守レベル論理名のユーザーレベルへのマッピング (続き)

保守レベル	ユーザーレベル
calstore_aa.siroe.com ~ calstore_az.siroe.com	なし。エンドユーザーには隠されている

その他の配備の課題

サービスの可用性の設計

論理アーキテクチャを決定したら、次の手順ではサイトに適したサービスの可用性レベルを特定します。サービスの可用性レベルは、選択したハードウェア、ソフトウェアインフラストラクチャ、使用する保守方法が関係しています。この章では、いくつかの選択肢、その利点、およびコストについて説明します。

この章には、以下の節があります。

- [高可用性ソリューションの概要](#)
- [Communications Services の高可用性ソリューション](#)

高可用性ソリューションの概要

Communications Services 製品は、2つの異なる高可用性ソリューションである Sun Cluster と Veritas Cluster Server (VCS) をサポートします。Messaging Server と Calendar Server は、これらのソリューションのそれぞれに対するエージェントを提供します。

Messaging Server と Calendar Server は、非対称型および対称型 HA の両方をサポートします。非対称型 HA では、ホットスタンバイサーバーを使用します。対称型 HA では、障害発生時に単一サーバー上のソフトウェアの複数のインスタンスを実行します。両方のソリューションに長所があります。ただし、ほとんどの場合、対称型 HA を選択します。以下の節では、各ソリューションの長所と短所について説明します。

HA 配備の詳細については、次の Web サイトの『Sun Java System Messaging Server 配備計画ガイド』を参照してください。

<http://docs.sun.com/db/prod/entsys?l=ja>

対称型 HA

対称型 HA では、各サーバーは現在アクティブなサーバーに障害が発生した場合に処理している負荷全体を引き受けるパートナーサーバーを持ちます。通常、対象 HA ではサーバーのペアを配備しますが、これは必要条件ではありません。ただし、複数のサーバーのペアを配備する場合、配備の管理はより複雑になります。

対称型 HA ソリューションを配備する場合、それ自体の負荷を実行するだけではなく、パートナーの負荷を実行できるようにサーバーのサイズを指定する必要があります。これは、必ずしもサーバーが通常のピーク時の負荷の 200% を高速応答時間で処理できるようにサイズを指定する必要があるということではありません。これは、通常のピーク時の 200% 負荷でサーバーが停止してはならないことを意味します。200% の負荷の状態でなんとか稼動するサーバーを配備する場合、ピーク時に障害が発生すると、ユーザーの応答時間が増加する (おそらく大幅に) ことを理解する必要があります。また、ストアへのメール配信の停滞が発生する可能性も高くなります。多くの企業 (および ISP) では、これは容認できるリスクです。

非対称型 HA

非対称型 HA は、1/1 として配備することができます。つまり、各ライブサーバーが対応するホットスタンバイを所有します。繰り返しになりますが、フェイルオーバーサーバーはメインサーバーと同じように強力である必要はありません。ただし、小さなボックスを使用した場合には、ピーク時に障害が発生するとサービス時間が増加し、メール配信の停滞が生ずる恐れがあります。また、非対称型 HA は、N/1、N+1、または N+M モードで配備することもできます。Sun Cluster は、標準 HA 構成で基本的に合計 4 ノードに制限されるため、N+M は実際的ではありません。ただし、3+1 は妥当でサポート可能なソリューションです。

Veritas は、より多くのノード数をサポートするので、顧客のニーズに応じて拡張でき、ソフトウェア固有の制限やサポート制限はありません。

システムの自動再設定 (ASR)

HA ソリューションを評価することのほかに、ASR を実行できるハードウェアの配備を検討する必要があります。

ASR は、ダウンタイムの原因となるハードウェア障害を最小化するプロセスです。サーバーが ASR を実行できる場合、ハードウェアの個々のコンポーネントに障害が発生した場合のダウンタイムを最低限に抑えることができます。

一般的には、サーバーがより多くの ASR 機能を持つとそのコストは高くなります。高可用性ソフトウェアがない場合は、法外にコストが高くなるかぎり、データストアに対するハードウェアの大規模な冗長性と ASR 機能を持つマシンを選択します。

Various Sun SPARC サーバーは、さまざまなレベルの ASR をサポートします。各製品の ASR 機能を理解するには、各製品のデータシートを参照してください。

有効化テクニックとテクノロジーの使用

前節で説明した高可用性ソリューションのほかに、有効化テクニックとテクノロジーを使用して可用性とパフォーマンスの両方を向上させます。これらのテクニックとテクノロジーには、ロードバランサ、Sun Java System Directory Proxy Server、複製ロールプロモーションなどがあります。

ロードバランサの使用

ロードバランサを使用して、エンドツーエンドのシステム全体に高可用性を提供することにより、アーキテクチャの各層の機能の可用性を保証することができます。ロードバランサは、専用のハードウェア機器または完全なソフトウェアソリューションです。

負荷分散は、単一のインスタンス、サーバー、またはネットワークが単一の障害ポイントになることを回避すると同時にサービスのパフォーマンスを向上させる最善の方法です。負荷分散の主な目的の 1 つは、サービスの水平方向の能力を拡大することです。たとえば、ディレクトリサービスの場合、ロードバランサは、ディレクトリサービスが処理する同時 LDAP 接続の総数および 1 秒あたり LDAP 操作の総数を増加させます。

Directory Proxy Server の使用

Sun Java System Directory Proxy Server (以前の Sun ONE Directory Proxy Server) は多くのプロキシ形式の機能を提供します。これらの機能の 1 つに LDAP 負荷分散があります。Directory Proxy Server は専用ロードバランサと同じ機能を実行できませんが、フェイルオーバー、レフェラルのフォロー、セキュリティ、マッピング機能のために、この機能の使用を検討します。

詳細については、次の Web サイトの Directory Proxy Server のマニュアルを参照してください。

<http://docs.sun.com/db/prod/entsys?l=ja>

複製ロールプロモーションの使用

Directory Server には、ディレクトリインスタンスの複製ロールを昇格させたり、降格させる方法があります。この機能により、複製ハブを複数のマスターサブライヤとして昇格させ、またはその逆を可能にします。また、コンシューマを複製ハブのロールに昇格させ、またはその逆を行うことができます。ただし、コンシューマを直接複数のマスターサブライヤとして昇格させたり、またはその逆を行うことはできません。この場合には、コンシューマはまず複製ハブとなり、次にハブから複数マスターの複製となることができます。逆の場合も同じように実行できます。

複製ロールプロモーションは分散配備に役立ちます。地理的に分散した 6 箇所のサイトがある場合について考えてみます。複数のマスターサブライヤを各サイトに配置したいと思いますが、最大 4 つのサイトに、サイトごとに 1 つ配置するだけに制限されています。ほかの 2 つの各サイトに少なくとも 1 つのハブを配置する場合は、ほかの複数のマスターサブライヤの 1 つがオフラインになっているか、または何らかの理由で運用されていない場合に、それらを昇格させることができます。

詳細については、次の Web サイトの『Sun Java System Directory Server 管理ガイド』を参照してください。

<http://docs.sun.com/db/prod/entsys?l=ja>

Communications Services の高可用性ソリューション

Directory Sever の高可用性化

Communications Services の見地からみると、ディレクトリサービスを計画する際の最も重要な要素は可用性です。インフラストラクチャサービスとして、ディレクトリは認証、アクセス、電子メールのルーティングなどの高レベルのアプリケーションに対して、可能なかぎり継続的なサービスを提供する必要があります。

高可用性を提供する Directory Server の重要な機能はレプリケーションです。レプリケーションは、ある Directory Server から別の Directory Server にディレクトリデータを自動的にコピーするメカニズムです。レプリケーションによって、可用性の高いディレクトリサービスを提供し、データを地理的に分散することが可能になります。実際的には、レプリケーションは以下の利点を提供します。

- 対障害性およびフェイルオーバー
- 負荷分散
- 高パフォーマンスと応答時間の短縮
- ローカルデータの管理

下表は、可用性のあるディレクトリを設計する方法を示します。

表 5-1 高可用性 Directory Server の設計

手法	説明
単一マスターレプリケーション	サブライヤとして機能するサーバーが1つまたは複数のコンシューマサーバーにマスターの複製をコピーする。この構成では、すべてのディレクトリの変更がサブライヤに格納されたマスターの複製に対して行われ、コンシューマには読み取り専用のデータの複製が含まれる
双方向、複数マスターレプリケーション	同一データの共有を担当する2つのサブライヤ間の複数マスター環境で、2つのレプリケーション定義を作成する。サブライヤ A とサブライヤ B がそれぞれ同一データのマスターの複製を保持し、この複数マスター構成のレプリケーションフローを制御する2つのレプリケーション定義が存在する

表 5-1 高可用性 Directory Server の設計 (続き)

手法	説明
4 方向、複数マスター	通常、2つの独立したデータセンターに Directory Server マスターのペアを提供する。この構成は、レプリケーションに 4 方向、複数マスターレプリケーション (MMR) を使用する。4 方向マスターフェイルオーバー構成により、この完全に接続されたトポロジがデータの統合を保証する高可用性ソリューションの提供が可能になる。レプリケーショントポロジのハブとともに使用する場合、負荷分散を容易にし、各データセンターの 4 つのコンシューマが、読み取り (検索) 操作のためにこのトポロジのスケーリングを可能にする
Directory Server 用の Sun Cluster エージェント	Sun Cluster ソフトウェアの使用により、ディレクトリの設定に最高水準の可用性が提供される。アクティブ Directory Server ノードの障害が発生した場合、Sun Cluster はバックアップノードにサービスの透過的なフェイルオーバーを提供する。ただし、このような環境におけるインストール、設定、保守などの管理 (およびハードウェア) コストは、通常 Directory Server のレプリケーション手法よりも高くなる

詳細については、次の Web サイトの『Sun Java System Directory Service Deployment Planning Guide』を参照してください。

<http://docs.sun.com/doc/817-5218>

Messaging Server および Calendar Server の高可用性化

Sun Cluster と Veritas テクノロジを使用して、Messaging Server と Calendar Server の両方の高可用性設定を行うことができます。Messaging Server と Calendar Server は、非対称型および対称型構成をサポートします。非対称型 (ホットスタンバイ) 構成では、サービスはプライマリノード上でのみ実行され、スタンバイセカンダリノードはアイドル状態のままになります。任意のリソース (ストレージ、ホストシステム、またはプロセス自体) で障害が検出されると、プライマリノードのサービスが停止され、セカンダリノードでサービスが開始されます。対称型構成では、複数のノードがアクティブサービスを同時に実行し、ノードは相互のバックアップとして機能します。フェイルオーバーの場合、障害が発生したノードのサービスはシャットダウンされ、指定されたバックアップノードで再開されます。この構成では、バックアップノードがすでにほかのアクティブサービスを実行しています。

フロントエンドとバックエンドコンポーネントが別々のマシンに分散された、Tier (層) Communications Services アーキテクチャでは、クラスタテクノロジーを使用してバックエンドコンポーネントを高可用性化する場合があります。これは、バックエンドが持続的データを保持する「ストア」となるからです。ディスクのサブシステムを保護することにより、Messaging Server MTA フロントエンドを高可用性化する場合があります。Calendar Server フロントエンドでクラスタテクノロジーを使用することは意味がありません。通常、Calendar Server フロントエンドは、冗長性、つまり複数のフロントエンドホストを配備することにより可用性を高めます。

詳細については、次の Web サイトの『Sun Java System Messaging Server 配備計画ガイド』を参照してください。

<http://docs.sun.com/db/prod/entsys?l=ja>

Communications Services ソフトウェアの機能

この章では、配備計画に影響のある Sun Java System コンポーネントの特定の機能の概要について説明します。

この章には、以下の節があります。

- [Communications Services コンポーネントの機能](#)
- [インフラストラクチャコンポーネントの機能](#)

Communications Services コンポーネントの機能

Communications Services は次の 3 つのコンポーネントで構成されます。

- Messaging Server
- Calendar Server
- Instant Messaging

Communications Services コンポーネントは、次の 3 つの追加コンポーネントと依存関係があります。

- Directory Server
- Identity Server
- Web サーバー (Sun Java System Web Server がこの機能を実行する)

この節では、コアコンポーネントのアーキテクチャの概要を説明します。

Messaging Server ソフトウェアの概要

通常、メッセージングサービスは、さまざまな規模の企業およびメッセージングホストの電子メールに関するニーズに対応するオープンな標準ベースのクライアントサーバーソリューションです。メッセージングサービスは、優れたディレクトリサービス、管理、スケーラビリティ、パフォーマンス、暗号化、リモート接続を提供します。

特に、メッセージングサービスは、音声、画像、ビデオファイル、HTML 形式、Java アプレット、およびデスクトップアプリケーションが埋め込まれた電子メールをすばやく配信するとともに、将来のアップグレードやスケーラビリティを備えています。

メッセージングサービスは、単純なレベルでは次のことを行います。

- 外部サイトからのメールの受信
- メッセージを配信し、ルーティングするユーザーのメールボックスの設定
- 内部ホストからのメールの受信
- メッセージを配信し、ルーティングするユーザーの宛先システムの設定

さらに、メッセージングサービスは、配達保証付きメール配信 (GMD)、ウィルス防止ソリューション、スパム制御を提供できます。

優れたメッセージングサービスにとって、SHARP (スケーラビリティ、高可用性、信頼性、高パフォーマンス) 標準に準拠するメッセージングサーバーが特に重要です。

メッセージングサーバーは、あるシステムから別のシステムへの電子メールの保存と配信をサポートする電子メール配信システムです。通常、Sun Java System Messaging Server を含むメッセージングサーバーには以下のコンポーネントが含まれています。

- **Mail User Agent:** エンドユーザーがメッセージの保存、作成、返信に使用するクライアント電子メールプログラムです。
- **Message Access/Transfer Protocols:** POP3、IMAP4、HTTP、および SMTP プロトコル。これらのプロトコルについては、「メッセージアクセスプロトコルとメッセージ転送プロトコル」の節で詳しく説明します。
- **Message Store:** アクセスサーバーによって取得されるか、または削除されるまでメールを保存するデータストア
- **Message Transfer Agent (MTA):** あるメールサーバーから別のメールサーバーにメッセージを転送します。さらに、MTA はそれ自体の Message Store にメッセージを転送します。
- **LDAP サーバー :** DAP (Directory Access Protocol) の「軽量」バージョンです。通常、LDAP サーバーはメッセージングサーバーのコンポーネントではありませんが、ユーザーの保守と管理のために不可欠です。通常は組織内のさまざまなアプリケーションのニーズに対応します。

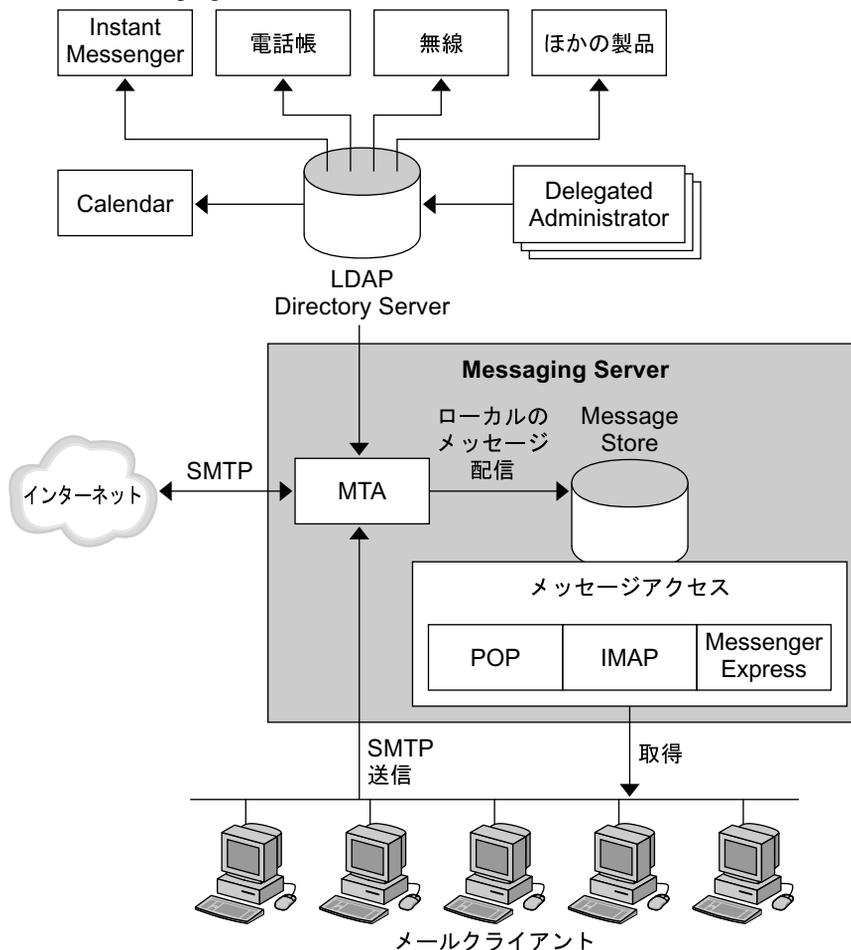
注 適切な Directory Server の実装は Messaging Server の配備を成功させるのに役立つので、このガイドのほかに『Sun Java System Directory Server 配備計画ガイド』も参照してください。

Messaging Server アーキテクチャの概要

Sun Java System Messaging Server には、Message Store、MTA、メッセージ転送およびメッセージアクセスプロトコルを実装するサーバーサイドソフトウェアが含まれます。また、HTTP メッセージアクセスクライアントである Messenger Express も含まれます。さらに、Directory Server ソフトウェアによって完全に設定可能です。

90 ページの図 6-1 に、追加設定不要で使用可能な Messaging Server の基本アーキテクチャを示します。

図 6-1 Messaging Server の基本アーキテクチャ



上図では、インターネットまたはローカルクライアントからの受信メッセージは SMTP を経由して MTA によって受信されます。アドレスが Messaging Server のドメイン内にある場合、MTA は Message Store にメッセージを配信します。メッセージのアドレスが Messaging Server の制御範囲外のドメインに宛てられている場合、MTA はインターネット上の別の MTA にメッセージを中継します。

旧式の UNIX /var/mail ファイルシステムにメールを配信することができますが、通常ローカルメッセージは、より最適化された Messaging Server の Message Store に配信されます。次に、これらのメッセージは、IMAP4 または POP3 クライアントによって取得されるか、または組み込まれた Messenger Express インタフェースを通じて取得されます。

Directory Server は、アドレス、代替メールアドレス、メールホスト情報などのローカルユーザーおよびグループの配信情報を取得し、格納します。MTA がメッセージを受信すると、Directory Server は、このアドレス情報を使用してメッセージの配信場所と配信方法を決定します。

メッセージの格納のほかに、Message Store は、Directory Server を使用してユーザーのログイン名とメールにアクセスするメールクライアントのパスワードを確認します。また、Directory Server は、制限容量、デフォルトのメッセージストアタイプなどの情報を格納します。

メールクライアントの送信メッセージは直接 MTA に送信され、MTA がこのメッセージをインターネット上の該当するサーバーに配信するか、アドレスがローカルの場合は、Message Store に配信します。

このディレクトリにユーザーとグループのエントリを追加して、新しいユーザーとグループを作成します。Delegated Administrator を使用するか、または LDAP を使用するディレクトリを変更して、エントリを作成または変更することができます。

Messaging Server のコンポーネントは、Messaging Server Administration Console (90 ページの図 6-1 には表示されていない) によって管理されます。さらに、一連のコマンド行インタフェースとテキストベースの設定ファイルが提供されます。Messaging Server ホストに接続された任意のマシンで管理タスクを実行することができます (管理者がパスワードを知っている場合)。一般的な管理タスクには、ユーザーおよびグループのメールシステムへの追加、変更およびメールシステムからの削除があります。また、管理者は MTA と Message Store の操作を設定することもできます。

Messenger Express による Web ベースのメールクライアントサービス (HTTP)

MEM (Messaging Express Multiplexor) は、Sun Java System Messenger Express のメールクライアントのミドル層を提供します。このクライアントにより、ユーザーがメールボックスのデータにアクセスし、Web ブラウザ (WUA) を使用して電子メールメッセージを作成できるようになります。MEM の利点は、どのバックエンドサーバーがメールを格納していても、エンドユーザーは MEM にのみ接続して電子メールにアクセスできることです。MEM は HTTP セッション情報を管理し、Directory Server に格納されたユーザーの LDAP 情報を使用してユーザーのプロファイルを作成します。次の利点は、スタティックファイルと LDAP 認証の状態が Messaging Server のフロントエンドサーバーに配置されることです。この利点は、Messaging Server ストアのバックエンドにおけるディスク要件を軽減します。

Messaging Server の詳細情報の入手先

Messaging Server の詳細情報については、次のマニュアルを参照してください。

<http://docs.sun.com/db/prod/entsys?l=ja>

Calendar Server ソフトウェアの概要

Sun Java System Calendar Server は、標準の基盤の上に構築されています。データストアは iCalendar テクノロジーをベースとしています。予定および作業に対するメッセージの iMIP Publish、Request、Reply、Cancel をサポートします。iCalendar 標準のネイティブサポートにより、ユーザーはインターネットを介して簡単に共有できる形式で予定をスケジュール化することができます。Calendar Server は、XML および XSLT (eXtensible Stylesheet Language Transformations) をベースとする Web client、Sun Java System Calendar Express を完全に装備して出荷され、特定の企業または ISP 用にカスタマイズできます。また、Calendar Server は、ほかの一般的なカレンダークライアントに対する接続機能も備えています。この結果、さまざまなクライアントとデバイスが Calendar Server と通信できるようになります。

さらに、Calendar Server は、Calendar Server と通信するクライアントの開発に使用する WCAP (Web カレンダーアクセスプロトコル) と呼ばれるオープンプロトコルを提供します。WCAP は、電話、PDA、またはほかのデバイスのような非ブラウザインタフェースにカレンダー情報を提供する必要がある場合にきわめて有用です。

WCAP コマンドの詳細については、次の『Sun Java System Calendar Server Developer's Guide』を参照してください。

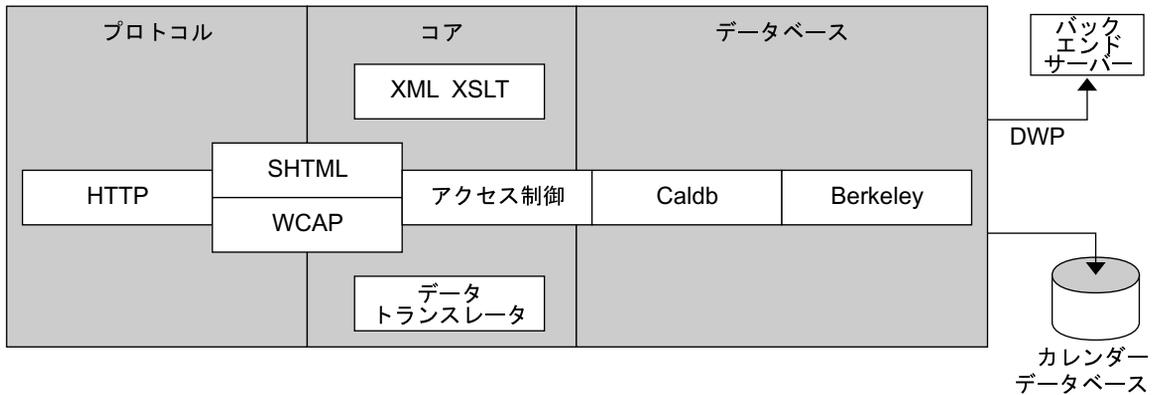
<http://docs.sun.com/doc/817-5698>

柔軟性に重点をおいて構築された Calendar Server は、LDAP ベースまたは ID ベースの環境で実行することができます。LDAP ベースの環境では、認証のために LDAP ディレクトリを使用し、プロビジョニングのために Calendar Server の管理ツールを使用します。ID ベースの配備では、認証のために Identity Server を使用し、プロビジョニングのために Identity Server の管理ツールを使用します。

Calendar Server アーキテクチャの概要

93 ページの [図 6-2](#) に、Calendar Server の内部アーキテクチャを示します。

図 6-2 Calendar Server 内部サブシステムの論理フロー



Calendar Server の一部は、Messaging Server と Directory Server の実績のあるコンポーネントの上に構築されています。設計は高度にモジュール化されています。Calendar Server は、共有サーバーを収集する方法で実装され、次の 3 つの主要コンポーネントに分類されます。

- プロトコルサブシステム**：コマンドと要求は、HTTP プロトコル層に入ります。これは、カレンダー要求のサポートを効率化するための最小の HTTP サーバー実装です。クライアントは SHTML または WCAP (Web カレンダーアクセスプロトコル) コマンドを使用して要求を送信します。
- コアサブシステム**：コアサブシステムには、アクセス制御サブシステム、ユーザーインタフェース (UI) ジェネレータサブシステム (XML と XSLT を使用する SHTML またはデータトランスレータを使用する WCAP のいずれか)、カレンダーデータベースサブシステム、CSAPI プラグインが含まれます。コアサブシステムはカレンダー要求を処理し、適切な UI 出力を生成します。また、コアサブシステムは、Calendar Server API (CSAPI) およびプロキシ認証 SDK (authSDK) を含むユーザー認証を処理します。
- データベースサブシステム**：データベースサブシステムは、Sleepycat Software の Berkeley DB (データベース API は未公開) を使用します。データベースサブシステムは、データベースとの間で予定、仕事 (作業)、アラームなどのカレンダーデータを取得、格納します。カレンダーデータは iCalendar 形式で、Calendar Server データのスキーマは iCalendar 標準のスーパーセットです。データベースサブシステムは低次の形式でデータを返し、それをコア UI ジェネレータ (SHTML または WCAP) が適切な出力に変換します。

プロトコル、コア、データベースサブシステムの共有ライブラリは、さまざまな組み合わせでバインドされ、cshttpd、csdwpd、csadmin、csnotifyd などの実行可能デーモンを生成します。さらに、enpd デーモンは、Calendar Server に搭載される予定通知サービスを提供します。

コマンドまたは要求は、プロトコルサブシステムを経由してサーバーに入力され、コアサブシステムに渡されて処理されます。データベースアクセスはデータベースサブシステムを通じて行われます。

Calendar Server の詳細情報の入手先

Calendar Server の詳細情報については、次のマニュアルを参照してください。

<http://docs.sun.com/db/prod/entsys?l=ja>

Instant Messaging ソフトウェアの概要

インスタントメッセージングサービスは、さまざまな規模の企業およびホストのインスタントメッセージングに関するニーズに対応するオープンな標準ベースのクライアントサーバーソリューションです。インスタントメッセージングサービスは、企業全体にわたって、またインターネット上で、優れた管理、スケーラビリティ、パフォーマンス、セキュリティ、接続性を提供します。

インスタントメッセージングサービスの単純化した段階は以下のとおりです。

- 外部サイトからのメッセージの受信
- メッセージを配信し、ルーティングするユーザーの特定
- 内部ホストからのインスタントメッセージの受信
- メッセージを配信し、ルーティングする宛先システムの設定

また、インスタントメッセージングサービスは、リアルタイム会議、ニュース、カレンダーアラート、およびオフラインユーザーに対しては電子メールメッセージの転送を行うことができます。

優れたインスタントメッセージングサービスにとって、SHARP (スケーラビリティ、高可用性、信頼性、高パフォーマンス) 標準に準拠するサービスが特に重要です。

Instant Messaging には、以下のコアコンポーネントが含まれています。

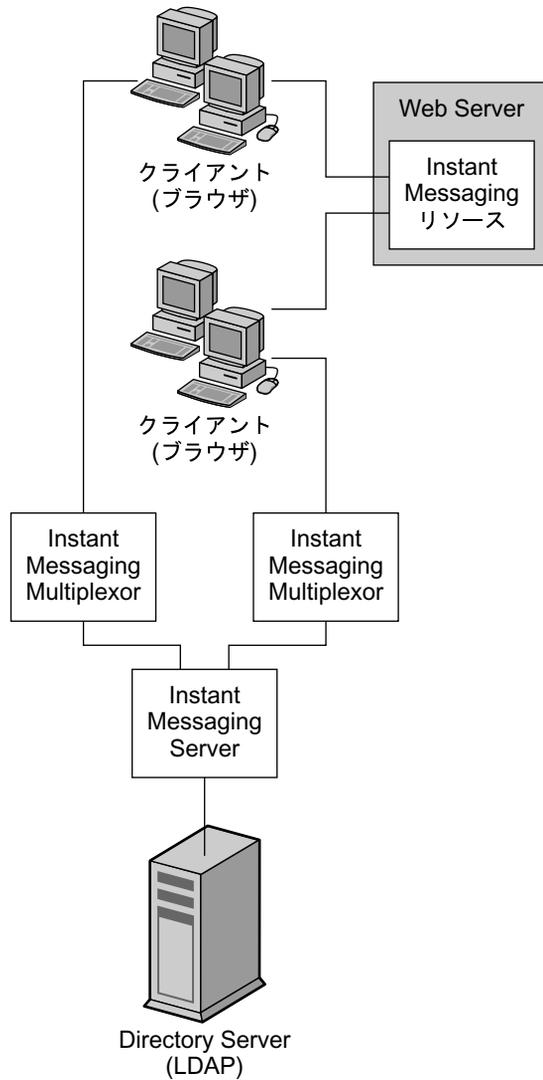
- **Instant Messenger リソース (クライアント):** エンドユーザーがメッセージの開始、作成、返信に使用するクライアントプログラムを構成する一連のファイルです。通常、ユーザーはクライアントを使用して会議に参加します。クライアントは Instant Messenger と呼ばれます。
- **Instant Messaging サーバー:** あるシステムから別のシステムへのインスタントメッセージの配信をサポートする電子メッセージ配信システムです。このサーバーは、Instant Messenger クライアントに存在情報を提供し、エンドユーザーがセッションを確立し、ポリシーを実行できるようにします。

- **Instant Messaging マルチプレクサ** : メッセージ接続を統合するスケーラビリティのあるコンポーネントです。たとえば、同時接続数が 40,000 に達するような大規模な配備をサポートするために、Instant Messaging は接続マルチプレクサを使用してサーバーのスケーラビリティを高めます。このコンポーネントは、Instant Messaging サーバーへの単一の接続を開きます。スケーラビリティに加えて、無許可の外部アクセスを防止するためにサーバーをファイアウォールの内側に配置したままファイアウォールの外側にマルチプレクサをインストールできます。また、Instant Messaging マルチプレクサは、単にマルチプレクサとも呼ばれます。
- **アクセス、通信、および転送プロトコル** : LDAP、HTTP、TCP/IP、および SMTP プロトコル。
- **Identity Server Instant Messaging サービスの定義** : Instant Messaging のポリシー管理、シングルサインオン機能を提供するために、Identity Server および Identity Server SDK が使用するサービスです。
- **Instant Messaging API** : カスタム Instant Messaging クライアントを作成可能にします。
- **LDAP サーバー** : DAP (Directory Access Protocol) の「軽量」バージョンです。通常、LDAP サーバーはインスタントメッセージングサーバーのコンポーネントではありませんが、ユーザーの保守と管理のために不可欠です。通常は組織内のさまざまなアプリケーションのニーズに対応します。

Instant Messaging アーキテクチャの概要

96 ページの図 6-3 に、追加設定不要で使用可能な Instant Messaging の基本アーキテクチャを示します。

図 6-3 Instant Messaging の基本アーキテクチャ



Web サーバー（または、Web サービスに埋め込まれたアプリケーションサーバー）は、ブラウザを使用して Instant Messaging リソースをクライアントにダウンロードします。リソースファイルはクライアントを構成します。クライアントは、Instant Messaging サーバーにメッセージを転送するマルチプレクサを通じて相互にメッセージを送信します。

Directory Server は、設定、場所、およびこのユーザーにメッセージをルーティングするマルチプレクサなどのローカルユーザーおよびグループの配信情報を格納し、取得します。Instant Messaging サーバーがメッセージを受信すると、Directory Server は、この情報を使用してメッセージの配信場所と配信方法を決定します。また、Directory Server には連絡先一覧および購読などのユーザー情報を含めることができます。

この基本的な設定によって、Instant Messaging は直接 Directory Server にアクセスし、Instant Messaging を使用するメールクライアントのユーザーログイン名とパスワードを検証します。

クライアントからの送信インスタントメッセージは直接マルチプレクサに送信されません。マルチプレクサは、該当する Instant Messaging サーバーにメッセージを送信し、順に別の Instant Messaging サーバーにメッセージを転送するか、またはメッセージがローカルの場合は受信者が関連するマルチプレクサにメッセージを転送します。

このディレクトリにユーザーのエントリを追加して、新しいユーザーを作成します。Directory Server に用意されているツールを使用してディレクトリを変更し、エントリを作成または変更することができます。

Instant Messaging コンポーネントは、一連のコマンド行インタフェースとテキストベースの設定ファイルを使用して管理されます。Instant Messaging ホストに接続された任意のマシンで管理タスクを実行することができます (管理者が必要な権限を持っている場合)。

Instant Messaging の詳細情報の入手先

Instant Messaging の詳細情報については、次のマニュアルを参照してください。

<http://docs.sun.com/db/prod/entsys?l=ja>

インフラストラクチャコンポーネントの機能

この節では、Communications Services コンポーネントが依存する Directory Server、Identity Server、DNS インフラストラクチャコンポーネントについて説明します。

Directory Server ソフトウェアの概要

ディレクトリサービスはソフトウェアの集合体であり、企業、登録者、またはその両方の格納情報を処理します。このマニュアルでは、ディレクトリサービスは少なくとも 1 つの Directory Server と 1 つ以上のディレクトリクライアントプログラムで構成されます。クライアントプログラムは、設定されたアクセス権に応じて、名前、電話番号、住所、ディレクトリに格納されたその他のデータにアクセスすることができます。

Directory Server には、ディレクトリ自体、LDAP プロトコルを実装するサーバーサイドソフトウェア、ユーザーがディレクトリ内を検索し、エントリを変更できるグラフィカルユーザーインターフェースが含まれています。また、Server Console のディレクトリマネージャを含むその他の LDAP クライアントを使用することもできます。さらに、別の LDAP クライアントプログラムを購入するか、Directory Server 製品に含まれる LDAP クライアント SDK を使用して独自のプログラムを記述することができます。

ほかのクライアントプログラムを追加しなくても、Directory Server はイントラネットまたはエクストラネットの基盤として使用できます。すべての Sun Java System コンポーネントサーバーは、社員、顧客、供給業者、パートナーデータなど共有サーバー情報の中央リポジトリとしてディレクトリを使用します。

Directory Server を使用して、エクストラネットのユーザー認証の管理、アクセス制御の作成、ユーザー設定のセットアップ、ユーザー管理の集中化を行うことができます。ホストしている環境では、パートナー、顧客、供給業者は、管理コストを低減するためにディレクトリの固有の領域を管理することができます。

Directory Server は、以下のコンポーネントで構成されます。

- プラグインインターフェースを持つ LDAP サーバー
- 管理サーバー
- サーバーを管理する Server Console
- サーバーの起動と停止、データベースのデータのインポートとエクスポート、データベースのインデックスの再作成、アカウントの無効化と機能停止、LDIF マージ、カーネルの調整、レプリケーションの管理などを実行するコマンド行ツール
- SNMP エージェント

- DSML (Directory Services Markup Language)

Directory Server アーキテクチャの概要

Directory Server の基本アーキテクチャは、以下のコンポーネントで構成されます。

- **ネットワーク通信を処理するサーバーのフロントエンド**：Directory Server のサーバーフロントエンドは、ディレクトリクライアントプログラムとの通信を管理します。Directory Server はデーモンとして機能します。複数のクライアントプログラムが、TCP/IP 上の LDAP または HTTP 上の DSML を使用してサーバーと通信できます。接続のために、クライアントが TLS の使用をネゴシエーションするかどうか応じて、接続は SSL/TLS (Transport Layer Security 上の Secure Socket Layer) を使用して保護することができます。
- **サーバー機能のプラグイン**：アクセス制御およびレプリケーションなどの実行に関して、Directory Server はプラグインに依存します。プラグインはコアサーバーに機能を追加する 1 つの方法です。たとえば、Uid Uniqueness プラグインは、ユーザー ID (uid) 属性に与えられた値が、ディレクトリをインストールするときに設定されるサフィックス内で一意のものであるかどうかを確認するために使用されます。
- **サーバー関連のデータを含む基本ディレクトリツリー**：ディレクトリ情報ツリーまたは DIT とも呼ばれるディレクトリツリーは、階層の最上部に表示されるツリーのルート、または最初のエントリによって、大部分のファイルシステムで使用されるツリーモデルを反映します。

Directory Server の詳細情報の入手先

Directory Server の詳細情報については、次のマニュアルを参照してください。

<http://docs.sun.com/db/prod/entsys?l=ja>

Identity Server ソフトウェアの概要

Identity Server は、急速に拡大する企業のニーズに対応するために設計された ID 管理ソリューションです。Identity Server を使用して、社員、パートナー、供給業者の ID を 1 つのオンラインディレクトリに取り込むことができます。これにより、ユーザーが企業内の情報にアクセスするためのポリシーとアクセス権を確立することができます。Identity Server は、すべてのデータ、サービス、およびどの情報に誰がアクセスできるかを決定するための重要な機能を提供します。

企業ユーザーまたは外部アプリケーションが企業内の Web サーバーに格納されたコンテンツにアクセスしようとする、ポリシーエージェントが要求をインターセプトして Identity Server に転送します。Identity Server は、ユーザー名およびパスワードなどの証明書の提示をユーザーに要求します。証明書が中央の Directory Server に格納された証明書と一致する場合、Identity Server はユーザー ID を検証します。次に、Identity Server はユーザー ID に関連付けられたポリシーを評価して、ユーザーが要求する情報を表示できるかどうかを決定します。最後に、Identity Server はユーザーの情報へのアクセスを許可するか、または拒否します。

Identity Server は、単一の管理コンソールで表示できるように、以下の主要な 4 つの機能を 1 つの製品に統合します。

- ID 管理
- アクセス管理
- サービス管理
- 連携管理

Identity Server アーキテクチャの概要

Identity Server は、スケーラビリティ、パフォーマンス、および簡単な配備を提供するために、Java テクノロジベースのアーキテクチャを使用します。また、以下のような業界標準を活用します。

- HTTP (HyperText Transfer Protocol)
- XML (eXtensible Markup Language)
- SOAP (Simple Object Access Protocol)
- SAML (Security Assertions markup Language) 仕様

Identity Server の詳細情報の入手先

Identity Server の詳細情報については、次のマニュアルを参照してください。

<http://docs.sun.com/db/prod/entsys?l=ja>

DNS の概要

ローカルネットワーク上の高品質キャッシュ DNS (Domain Name System) サーバーは、Communications Services の製品配備の要件です。

DNS は、標準 TCP/IP プロトコル群の一部であるアプリケーション層のプロトコルです。このプロトコルは、インターネット上で使用される名前サービスである DNS 名前サービスを実装します。

DNS は、インターネット上の複雑で世界的な規模のコンピュータの階層をサポートしますが、実際の基本的な機能は TCP/IP ベースのネットワークに対して名前とアドレス解決を提供する非常に簡単な機能です。「マッピング」とも呼ばれる名前とアドレス解決は、ホスト名をインデックスとして使用してデータベース内のコンピュータの IP アドレスを検出するプロセスです。

DNS は 2 つの主要なサービスを提供します。DNS は名前をアドレスにマッピングします (また、アドレスをホスト名にマッピングする)。また、DNS は、sendmail、POP などインターネットを介してメールを配信するメール配信サービスに役立ちます。

インターネットを介してメールを配信するために、DNS はメール交換レコード (MX レコード) を使用します。多くの組織は、組織内のホスト宛のメールの直接配信を許可していません。代わりに、中央のメールホスト (または、メールホストのセット) を使用して、受信メールメッセージをインターセプトして、受信者にルーティングします。

メール交換レコードは、ドメイン内の各マシンにサービスを提供するメールホストを識別します。このため、メール交換レコードは、リモート組織の DNS ドメイン名、および対応するメールホストの IP アドレスまたはホスト名をリスト表示します。次の表について検討します。

DNS ドメイン	メールホスト
International.com	129.44.1.1
sales.example.com	SalesExampleMailer
eng.example.com	EngExampleMailer
siroe.com	SiroeMailer

メールエージェントが別のドメインへのメールの送信要求を受信すると、受信者のアドレスを右から左に解析して、表内で一致するアドレスを検索します。

neverhome.sales.example.com へのメール送信要求を受信した場合、メールエージェントは最初に最上位ラベル com を抽出します。メール交換レコードを調べて、エントリ com の有無を確認します。エントリがない場合、解析を続行します。次のラベルを抽出して、エントリ example.com を検索します。エントリがない場合、検索を続行します。次に、エントリ sales.example.com を検索します。上表で確認できるとおり、このドメインのメールホストは SalesExampleMailer です。これがホスト名であることがわかったので、メールエージェントは DNS にホスト名の解決を要求します。DNS からこのメールホストのアドレスが提供されると、メールエージェントがメッセージを送信します。

メール交換レコードがメールホスト名ではなく IP アドレスを指定した場合には、DNS による名前解決が不要のため、メールエージェントは直接そのアドレスにメッセージを送信します。

Communications Services 配備の例

この章では、Communications Services の企業向け配備の例を説明します。配備に実装する機能に応じて、異なるホストのセットおよびその他のネットワークインフラストラクチャをインストールする必要があります。

この章には、以下の節があります。

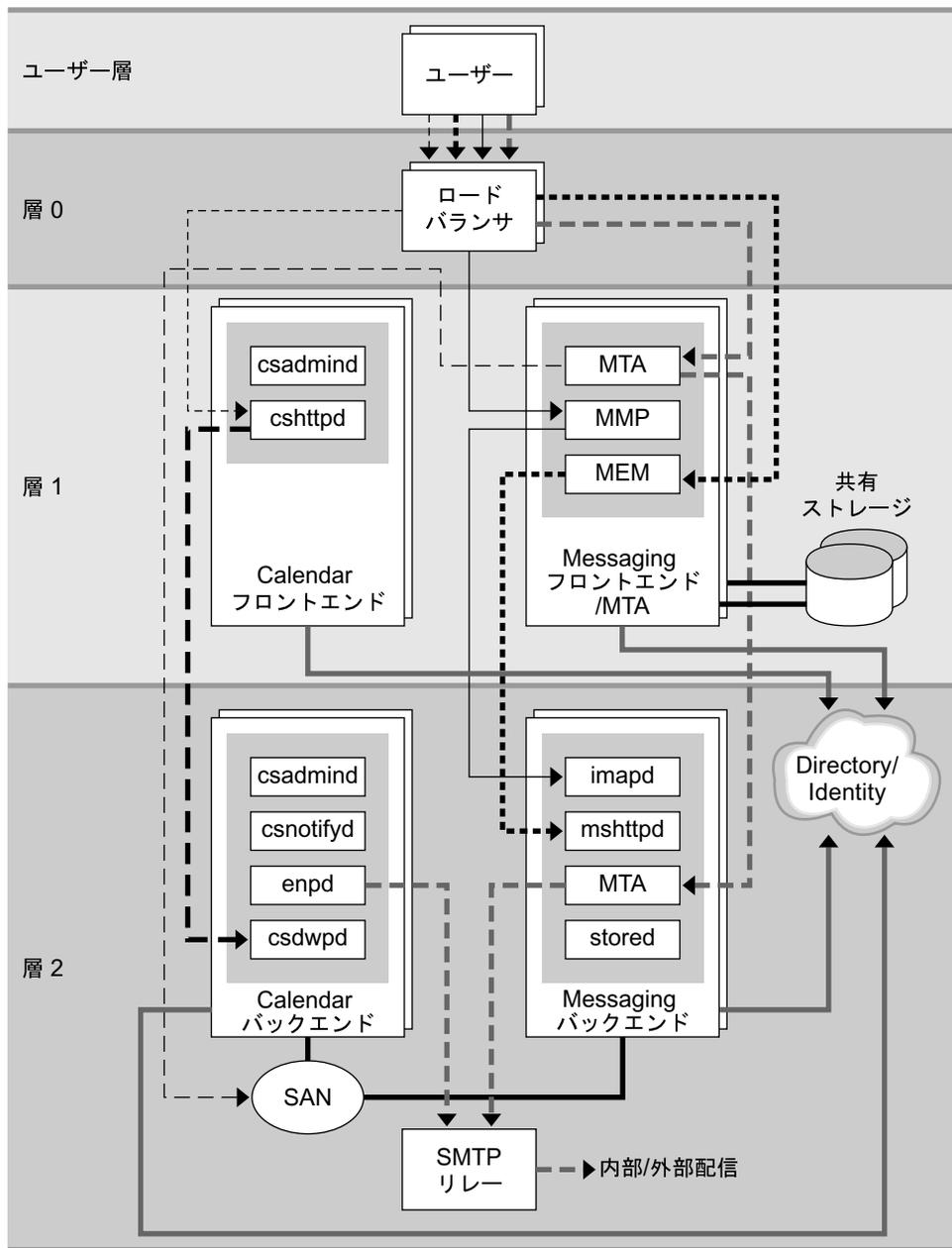
- [2 層配備の例](#)

2 層配備の例

[104 ページの図 7-1](#) は、Messaging Server と Calendar Server の 2 層配備の例を示します。層 0 はロードバランサで構成されます。層 1 は Calendar Server と Messaging Server のフロントエンドで構成されます。Calendar Server と Messaging Server のバックエンドストアが層 2 を形成します。

Directory Server と Identity Server は、それ自体が複雑な配備です。この図では、これらのコンポーネントが「雲形模様」で示されています。

図 7-1 Communications Services 2層配備の例



- 8081 ——— 143 ——— 389 -.-.- 9779
- 80 -.-.- 25 -.-.- 225

上図の例では、ロードバランサが層 0 を形成し、フロントエンドサービスへのアクセスをユーザーに指示します。

フロントエンドサービスは 4 台のマシンで構成されます。2 台のマシンに Calendar Server フロントエンドコンポーネントがインストールされます。これらの Calendar Server フロントエンドマシンは、1 台または 2 台の CPU サーバーと固有の内部ディスクストレージで構成されます。ほかの 2 台のマシンは Messaging Server プロキシと MTA として設定され、外部ディスクアレイを共有します。これらの Messaging Server マシンは 4 台の CPU サーバーで構成されます。

バックエンドも 4 台のマシンで構成されます。2 台のマシンは、メールストアとして機能し、Messaging Server プロセスを実行します。ほかの 2 台のマシンは、カレンダーストアとして機能し、Calendar Server プロセスを実行します。ストアマシンは SAN (Storage Area Network) に接続されます。これらのバックエンドマシンは、CPU のニーズに基づいて、さまざまな方法で配備することができます。CPU の総数を決定後、垂直および水平方向の構成を最適化することができます。たとえば、アーキテクチャが合計 12 個の CPU を必要とする場合、3 台の 4 方向サーバー、2 台の 6 方向サーバー、または 1 台の 12 方向サーバーを使用することができます。

別のマシンは、Calendar Server 通知と Messaging Server 電子メールの両方の SMTP リレーとして機能します。

下表は、この配備で使用するプロトコルとポート番号について説明します。

表 7-1 2 層配備の例で使用するプロトコルとポート

プロトコル	ポート	用途
HTTP	80	Messaging Server MEM と Web メール (httpd) コンポーネントと通信するクライアント
SMTP	25	Messaging Server MTA コンポーネント、フロントエンドおよびバックエンドの MTA コンポーネント、および電子メール通信用 Calendar Server (csendp) コンポーネントと通信するクライアント
IMAP	143	Messaging Server MMP と imapd コンポーネントと通信するクライアント
LMTP	225	バックエンド MTA を迂回して、フロントエンドからバックエンドの Message Store に電子メールをルーティングする MTA
LDAP	389	LDAP ディレクトリと通信するフロントエンドとバックエンド
HTTP	8081	カレンダーフロントエンド (cshttpd) と通信するクライアント
DWP	9779	カレンダーバックエンド (csdwpd) と通信するカレンダーフロントエンド (cshttpd)

2 層配備の例

用語集

このマニュアルで使用する用語の完全なリストについては、『Java Enterprise System Glossary』(<http://docs.sun.com/doc/816-6873>)を参照してください。

数字

2 層アーキテクチャ, 32, 61, 62, 71

2 層論理アーキテクチャ, 66

C

Calendar, 52

Calendar Express, 92

Calendar Server

Directory Server との相互作用, 47

アーキテクチャ, 92

概要, 17, 92

コアサブシステム, 93

高可用性, 53, 85

考慮事項, 52

データの移行, 56

データベースサブシステム, 93

ビジネスニーズへの対応, 21

標準プロトコルのサポート, 92

プロトコルサブシステム, 93

comms_dssetup.pl スクリプト, 48

Communications Express

Directory Server の考慮事項, 48

概要, 19

考慮事項, 56

スケーリング, 53

Communications Services

Portal Server, 24

概要, 15

高可用性 (High Availability), 23

コンポーネント, 40

コンポーネント製品の依存性, 20

配備プロセス, 25

ビジネスニーズへの対応, 20

利点のまとめ, 22

例, 103

Connector for Microsoft Outlook

概要, 19

考慮事項, 55

製品の依存性, 55

D

DC ツリー (DC Tree), 41, 43

Delegated Administrator, 44

Directory Proxy Server, 81

Directory Server

Communications Express の考慮事項, 48

Tier (層) アーキテクチャの考慮事項, 46

アーキテクチャ, 99

高可用性, 83

考慮事項, 45

個人アドレスブックに関する考慮事項, 48

トポロジの考慮事項, 46

容量計画, 46

DNS, 32, 63, 101

DWP, 57, 105

I

Identity Server, 100

アーキテクチャ, 100

概要, 100

Instant Messaging

アーキテクチャ, 95

概要, 18, 94

考慮事項, 54

ビジネスニーズへの対応, 21

Instant Messaging プロキシ, 40

L

LDAP Schema 1, 44

LDAP Schema 2, 44

LDAP サーバー, 88

LMTP, 50, 71, 74, 105

M

Mail Message Proxy, 40, 51, 63, 66

Message Store, 40, 49, 88

Message Transfer Agent, 40, 50, 63, 66, 88

Messaging Express Multiplexor, 51, 66, 72, 91

Messaging Server

アーキテクチャ, 89

概要, 17, 88

高可用性, 85

考慮事項, 49

ビジネスニーズへの対応, 21

Messenger Express, 65, 67

Microsoft Exchange Server データ, 56

Microsoft Outlook, 19, 40, 55, 65, 67

P

Portal Server, 24

Desktop, 24

Secure Remote Access, 24

考慮事項, 54

R

Realtime Blackhole List, 75

S

SSL, 59, 60, 66, 69, 71

Storage Area Network, 40

Sun ONE Synchronization, 18

W

WCAP (Web カレンダーアクセスプロトコル), 92

あ

アクセス制御リスト (access control list), 45

アクセス層, 40, 71

アドレスブック検索, 40

アプリケーションのセキュリティ, 59

安全な接続, 59

安全な接続の実装, 59

う

運用要件, 30

え

エッジ論理アーキテクチャ, 62, 68, 70

か

概要

Calendar Server, 17
Communications Express, 19
Communications Services, 15
Connector for Microsoft Outlook, 19
Instant Messaging, 18
Messaging Server, 17
Sun ONE Synchronization, 18
論理アーキテクチャ, 61

カレンダーストア (Calendar Store), 40

き

技術要件, 31

逆引き DNS, 71

こ

高可用性, 79

高可用性 (High Availability), 23

コーポレートディレクトリ, 40

個人アドレスブック, 48

さ

サービス拒否, 71

サービス層, 39

サービス層の理解, 39

サービス品質保証契約, 33

サイトの分散, 31

財務要件, 32

作動時間の計算, 33

し

システムの自動再設定, 81

使用パターン, 31

シングルサインオン (single sign-on), 44

す

水平方向のスケラビリティ戦略, 73

スキーマの要件, 44

スパム, 71

せ

製品システム, 27

セキュリティ

アプリケーション, 59

オペレーティングシステム, 58

考慮事項, 57

ネットワーク, 58

そ

総所有コスト (TCO), 34

組織ツリー, 41

た

対称型 HA, 80

単一層、1 ホスト論理アーキテクチャ, 62

単一層アーキテクチャ, 61, 62, 70

単一層、複数ホスト論理アーキテクチャ, 63

単一層、分散アーキテクチャ, 64

て

ディレクトリ情報ツリー (DIT), 40, 45

データ層, 40

に

認証局 (Certificate Authority), 60

は

配備アーキテクチャの設計, 25

配備のカスタマイズ, 26

配備のテスト, 26

配備の例, 103

配備プロセス, 25

配備目標, 29

ひ

ビジネス要件, 30

非対称型 HA, 80

ふ

複製ロールプロモーション, 82

プロジェクトの目標の特定, 34

め

メール交換レコード (mail exchange record), 101

メールユーザーエージェント, 88

メッセージアクセスおよび転送プロトコル, 88

よ

要件

運用, 30

企業文化と政治, 30

技術, 31

ザイム, 32

スキーマ (schema), 44

ビジネス, 30

容量計画, 34

り

利点

2層アーキテクチャ, 71

単一層アーキテクチャ, 70

ろ

ロードバランサ, 81

論理アーキテクチャ, 61

論理サービス名, 75