Sun Java System Communications Services 6 2005Q4 配備計画ガイド

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

Part No: 819–3543
2005 年 10 月

051219@13215

目次

表目次

図目次

はじめに

『Sun Java System Communications Services 6 2005Q4 配備計画ガイド』には、Sun Java™ System Communications Services 6 2005Q4 を配備するために必要な情報が記載されています。このガイドは、Communications Services の理解、サイトの評価と分析、組織のニーズに適合する配備アーキテクチャーの設計プロセスに役立ちます。

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

お読みになる前に

このマニュアルをお読みなる前に、次の概念について理解しておく必要があります。

- エンタープライズレベルのソフトウェア製品の設計およびインストール方法
- IMAP、POP、HTTP、SMTP、WCAP、および LDAP プロトコル
- Solaris™ オペレーティングシステム (Solaris OS) のシステム管理とネットワーキング

内容の紹介

このマニュアルの第 I 部では、Communications Services 製品の全体像を示し、配備の各項目について概説します。第 II 部では、Sun Java™ System Messaging Server の配備に関する詳細情報を提供します。第 III 部では、Sun Java™ System Calendar Server の配備に関する詳細情報を提供します。第 IV 部では、Sun Java™ System Instant Messaging の配備に関する詳細情報を提供します。第 V 部では、Sun Java™ System Communications Express の配備に関する詳細情報を提供します。第 VI 部では、配備の例を示します。次の表は、このマニュアルの内容を一覧にまとめたものです。

表 **P–1** 内容の紹介

章	説明
第 1 章	Communications Services の概要を示します。
第 2 章	組織のビジネス要件と技術要件を分析する方法について説明します。
第 3 章	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。
第 4 章	ネットワークインフラストラクチャーの構成要素とインフラストラクチャーレイアウトの計画方法について説明します。
第 5 章	Communications Services の論理アーキテクチャーの開発方法について説明します。
第 6 章	サービス可用性に関する選択肢とそのメリットおよびコストについて説明します。
第 7 章	セキュリティ手法の概要、一般的なセキュリティ脅威、およびセキュリティニーズ分析手順の概要について説明します。
第 8 章	Communications Services のスキーマおよびプロビジョニングのオプションについて説明します。
第 9 章	Messaging Server ソフトウェアの概要を説明します。
第 10 章	Messaging Server 配備のサイズ決定の基礎について説明し、正しいサイズ決定データを得て配備上の判断ができるようにすることを目的としています。
第 11 章	Messaging Server 配備のアーキテクチャーの設計方法について説明します。

表 P–1 内容の紹介　(続き)

章	説明
第 12 章	メッセージングトポロジの設計方法について説明します。メッセージングトポロジとは、ネットワーク化されたメッセージングシステムの物理レイアウトと論理レイアウトのことです。
第 13 章	Messaging Server 配備のさまざまなコンポーネントに対する計画を立案し、それらのコンポーネントを保護する方法について説明します。
第 14 章	さまざまなスパム防止、ウィルス対策用のツールと利用可能な戦略について説明します。
第 15 章	Messaging Server をインストールする前に検討しなければならない考慮事項と、実行しなければならない手順について説明します。
第 16 章	Calendar Server ソフトウェアの概要を説明します。
第 17 章	Calendar Server 配備の基本的なアーキテクチャーについて説明します。
第 18 章	Calendar Server 配備のさまざまな構成要素に関する計画を立案し、それらの構成要素を保護する方法について説明します。
第 19 章	Calendar Server サービスに関する、Calendar Server 配備の追加考慮事項について説明します。
第 20 章	Calendar Server のインストール前に考慮が必要な事項について説明します。
第 21 章	Instant Messaging ソフトウェアの概要を説明します。
第 22 章	Instant Messaging 配備のサイズ決定の基礎について説明し、正しいサイズ決定データを得て配備上の判断ができるようにすることを目的としています。
第 23 章	さまざまな Instant Messaging アーキテクチャーについて説明します。
第 24 章	Instant Messaging のインストール前に考慮が必要な事項について説明します。
第 25 章	Communications Express ソフトウェアの概要を説明します。
第 26 章	Communications Express の基本的なアーキテクチャーについて説明します。
第 27 章	Communications Express のインストール前に考慮が必要な事項について説明します。
第 28 章	Communications Services の配備例を示します。

表 **P–1** 内容の紹介　(続き)

章	説明
用語集	Java™ Enterprise System の用語集へのリンクを提供します。

Communications Services マニュアルセット

次の表は、Communications Services のコアマニュアルセットに含まれるマニュアルを一覧にまとめたものです。

表 **P–2** Communications Services マニュアル

マニュアルタイトル	内容
『Sun Java System Communications Services 6 2005Q4 Schema Reference』	LDAP を使用する Sun Java System Communication Services 製品 (具体的には Messaging Server と Calendar Server) のスキーマ情報のリファレンスマニュアルです。
『Sun Java System Communications Services 6 2005Q4 Schema Migration Guide』	Sun Java System Communications Services (具体的には Messaging Server と Calendar Server) の Sun Java™ System LDAP Directory データを LDAP スキーマ 1 から LDAP スキーマ 2 に移行する方法について説明しています。
『Sun Java System Communications Services 6 2005Q4 Delegated Administrator 管理ガイド』	Sun™ Java System Communications Services Delegated Administrator を設定および管理する方法について説明しています。

関連マニュアル

Communications Services の配備に関連するほかのサーバーのマニュアルについては、次を参照してください。

- Access Manager のマニュアル:
  `http://docs.sun.com/app/docs/coll/1292.1`
- Calendar Server のマニュアル:
  `http://docs.sun.com/app/docs/coll/1313.1`

- Communications Express のマニュアル:
  `http://docs.sun.com/app/docs/coll/1312.1`
- Directory Server のマニュアル:
  `http://docs.sun.com/app/docs/coll/1316.1`
- Instant Messaging のマニュアル:
  `http://docs.sun.com/app/docs/coll/1309.1`
- Messaging Server のマニュアル:
  `http://docs.sun.com/app/docs/coll/1312.1`

デフォルトのパス名とファイル名

次の表は、このマニュアルで使用されているデフォルトのパス名とファイル名について説明したものです。

表 **P–3** デフォルトのパス名とファイル名

Placeholder	説明	デフォルト値

product_base	Messaging Server のベースインストールディレクトリを表します。Messaging Server 6 2005Q4 のデフォルトのベースインストール / 製品ディレクトリは、次のようにプラットフォームごとに異なります。	Solaris システム:
		`/opt/SUNWmgsr`
		Linux システム:
		`/opt/sun/messaging`

表記上の規則

このマニュアルでは、次のような字体や記号を特別な意味を持つものとして使用します。

表 **P–4** 表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.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。 machine_name% you have mail.
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	machine_name% **su** Password:
aabbcc123	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、rm *filename* と入力します。
『 』	参照する書名を示します。	『コードマネージャ・ユーザーズガイド』を参照してください。
「 」	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第 5 章「衝突の回避」を参照してください。 この操作ができるのは、「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	sun% **grep '^#define ** **XV_VERSION_STRING'**

コード例は次のように表示されます。

- C シェル

  machine_name% **command y|n** [*filename*]

- C シェルのスーパーユーザー

  machine_name# **command y|n** [*filename*]

- Bourne シェルおよび Korn シェル

  $ **command y|n** [*filename*]

- Bourne シェルおよび Korn シェルのスーパーユーザー

  # **command y|n** [*filename*]

[ ] は省略可能な項目を示します。上記の例は、*filename* は省略してもよいことを示しています。

| は区切り文字 (セパレータ) です。この文字で分割されている引数のうち 1 つだけを指定します。

キーボードのキー名は英文で、頭文字を大文字で示します (例: Shift キーを押します)。ただし、キーボードによっては Enter キーが Return キーの動作をします。

ダッシュ (-) は 2 つのキーを同時に押すことを示します。たとえば、Ctrl-D は Control キーを押したまま D キーを押すことを意味します。

コマンド例のシェルプロンプト

次の表は、デフォルトのシェルプロンプトおよびスーパーユーザープロンプトです。

表 **P–5** シェルプロンプト

シェル	プロンプト
UNIX システムおよび Linux システム上の C シェル	machine_name%
UNIX システムおよび Linux システム上の C シェルのスーパーユーザー	machine_name#
UNIX システムおよび Linux システム上の Bourne シェルおよび Korn シェル	$
UNIX システムおよび Linux システム上の Bourne シェルおよび Korn シェルのスーパーユーザー	#
Microsoft Windows のコマンド行	C:\

記号の表記規則

次の表は、このマニュアルで使用されている記号を説明しています。

表 **P–6** 記号の表記規則

記号	説明	例	意味
[ ]	省略可能な引数およびコマンドオプションを含みます。	ls [-l]	-l オプションは必須ではありません。
{ | }	必須コマンドオプションの選択肢のセットを含みます。	-d {y|n}	-d オプションには、y 引数、n 引数のいずれかを指定する必要があります。

表 **P–6** 記号の表記規則　*(続き)*

記号	説明	例	意味
${ }	変数の参照を表します。	${com.sun.javaRoot}	com.sun.javaRoot 変数の値を参照します。
-	同時に押下する複数のキーを連結します。	Control-A	Ctrl キーを押しながら A キーを押します。
+	続けて押下する複数のキーを連結します。	Ctrl+A+N	Ctrl キーを押して離したあと、後続のキーを順次押します。
→	グラフィカルユーザーインタフェースで、メニュー項目の選択を示します。	ファイル → 新規 → テンプレート	「ファイル」メニューから「新規」を選択します。「新規」サブメニューから「テンプレート」を選択します。

マニュアル、サポート、およびトレーニング

Sun のサービス	**URL**	内容
マニュアル	http://jp.sun.com/documentation/	PDF 文書および HTML 文書をダウンロードできます。
サポートおよびトレーニング	http://jp.sun.com/supportraining/	技術サポート、パッチのダウンロード、および Sun のトレーニングコース情報を提供します。

パート I 配備計画の概要

この部には、次の章があります。

第 1 章

Communications Services 配備の紹介

この章では、Sun JavaTM System Communications Services 6 2005Q4 の概要、Communications Services の配備に関する業務上の根拠、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、次の節があります。

Communications Services の概要

Sun Java System Communications Services 6 2005Q4 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、他の通信およびコラボレーション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、顧客が懸念するコスト、機能、および従来の通信インフラストラクチャーのセキュリティーなどの問題解決に取り組みます。

Communications Services は、企業と ISP 双方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、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。あらゆる組織にとって、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ー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、今日のスケーラビリティーとセキュリティー要件に対応するのに十分ではありません。Communications Services によって、組織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。最後に、Communications Services の配備は、企業のファイアウォールの外側に通信を拡張する際や、複数のデバイスを使用するモバイルユーザーに対して必要とされるようになった高いセキュリティーを提供します。

Communications Services のコアソリューションは、次のコンポーネント製品から構成されています。

- 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 7 (従来の Sun™ ONE Instant Messaging)

Communications Services ソリューションは次の追加機能によって拡張されます。

- Sun Java™ System Communications Express 6
- Sun ONE™ Synchronization 1.1
- Sun Java™ System Connector for Microsoft Outlook 7

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および数十万ものユーザーを抱える ISP 配備ための、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。

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。数千人から数百万人規模のユーザーのスケーリングに対応する 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ー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ー機能を提供します。

Messaging Server によって、組織は社員、パートナー、および顧客からなるコミュニティー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。

Messaging Server は現在、次の 2 つのクライアント向けユーザーインタフェース (UI) をサポートしています。

- Messenger Express
- Communications Express

今後、Messenger Express ユーザーインタフェースに新機能が追加されることはありません。Messaging Server は非推奨となり、代わって Communications Express が推奨のユーザーインタフェースとなりました。Sun Microsystems, Inc. は後日、Messenger Express の生産中止スケジュールを発表する予定です。

Messaging Server の概念やその他の配備に関する詳細については、パート II「Messaging Server の配備」を参照してください。

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 ブラウザから、非公開、公開、またはグループカレンダにアクセスできるようにします。配備は、Messaging Server および Instant Messaging とともに Calendar Server を使用して、包括的な通信およびコラボレーション環境をユーザーに提供します。

Calendar Server は現在、次の 2 つのクライアント向けユーザーインタフェース (UI) をサポートしています。

- Calendar Express
- Communications Express

Calendar Server は非推奨となり、代わって新しい Communications Express が推奨のユーザーインタフェースとなりました。今後、Calendar Server ユーザーインタフェースに新機能が追加されることはありません。Sun Microsystems, Inc. は後日、Calendar Server の生産中止スケジュールを発表する予定です。

Calendar Server の概念やその他の配備に関する詳細については、パート III「Calendar Server の配備」を参照してください。

Instant Messaging について

Sun Java System Instant Messaging 7 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、1 対 1 だけでなくグループによる共同作業にも対応し、短期間の通信のほか、会議室やニュースチャネルなどの持続的な場を利用することができます。Instant Messaging を Calendar Server、Messaging Server と組み合わせて使用すれば、包括的な通信およびコラボレーション環境をユーザーに対して提供できます。

Instant Messaging は、複数の認証メカニズムとセキュリティー保護された SSL 接続によって通信の統合を可能にします。Sun Java™ System Portal Server 6 と Sun Java™ System Access Manager 6 との統合により、セキュリティー機能、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、セキュリティー保護されたリモートアクセスが強化されます。さらに、Instant Messaging は XMPP (Extensible Messaging and Presence Protocol) をサポートします。XMPP を使用すると、ユーザーは、公衆ネットワークからの接続を集約するサードパーティー製の一部のクライアントが使用できるようになります。1 つのクライアント内に、AIM、Yahoo、MSN、Sun、およびその他の XMPP ベースのサーバーからの接続を収容できます。

Instant Messaging の概念や配備に関する詳細については、パート IV「Instant Messaging の配備」を参照してください。

Communications Express について

Sun Java System Communications Express 6 は、通信およびコラボレーション用の Web ベースの統合クライアントです。Communications Express は Messaging Server と Calendar Server の共通ソフトウェアであり、カレンダ情報、メール、およびアドレス帳に対する Web インタフェースをエンドユーザーに対して提供します。

Communications Express の概念や配備に関する詳細については、パート V「Communications Express の配備」を参照してください。

Synchronization 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 for Microsoft Outlook 7 は、Outlook を Messaging Server と Calendar Server 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) プロパティーに変換します。同様に、Connector for Microsoft Outlook は、Calendar Server に予定と作業を照会し、それらを MAPI プロパティーに変換します。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。

同様に、Connector for Microsoft Outlook では、WABP (Web Address Book Protocol) を使用して Address Book Server に連絡先を照会し、それらを MAPI プロパティーに変換します。このモデルによって、Connector for Microsoft Outlook は、Messaging Server のメール、Calendar Server のカレンダ情報、Address Book Server の連絡先という、3 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/app/docs/coll/1312.1`

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ー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があります。これらのコンポーネント製品には、Sun Java™ System Directory Server と、オプションで Sun Java System Access Manager 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 (従来の Sun™ ONE Web Server) または Sun Java™ System Application Server 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 章を参照してください。

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

組織は、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。サービスのアーキテクチャーには、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ーとスケーラビリティー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、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ーの開発が可能になります。最も重要なことは、Communications Services アーキテクチャーによって各サービス要素にセキュリティー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ー、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 は、確定したトランザクションを使用するため、メッセージはディスクに格納されるまで受信済みとして認識されません。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、オープンで相互運用可能かつ高性能な、業界最高レベルの時間管理およびリソース管理ソリューションです。Calendar Server によって、ほかのソリューションに比べて低い総所有コストで、必要な機能を得ることができます。Calendar Server のアーキテクチャーは、柔軟で拡張可能なので、垂直方向 (システムごとの CPU の数を増大させる) と水平方向 (ネットワークにサーバーを追加する) の両方向で拡張性があります。

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、Access Manager、Messaging Server、および Calendar Server と連携して動作します。この統合によって、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。また、XMPP による相互運用性は、パートナー企業や顧客との間でリアルタイム通信を実現したいと考えている企業に大きなメリットをもたらします。というのも、それらのパートナー企業や顧客の多くは、それぞれ独自のインスタントメッセージングシステムを構築しているからです。

Communications Express のビジネスニーズへの対応方法について

Communications Express は通信およびコラボレーション用の Web ベースの統合クライアントであり、インターネットサービスプロバイダ、企業、および OEM のニーズを満たします。Communications Express はカレンダ、メール、およびアドレス帳に対する統合ユーザーインタフェースを備えており、あるクライアントモジュールから別のクライアントモジュールへとアクセス先を変更しても、ユーザー資格の再認証を行う必要がありません。メールとカレンダ間の通信は、Access Manager または Messaging Server のシングルサインオンメカニズムを使って確立されます。カレンダアプリケーションとメールアプリケーションは、同一のアドレス帳を共有します。Communications Express の「オプション」タブで指定されたユーザー設定を、すべてのモジュールが共有します。

Communications Services の利点の概要

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配備に使用されてきました。大規模配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。

次の表に、Communications Services の利点をまとめます。

表 **1–1** Communications Services が組織に対して提供する利点

主な機能	利点
高いパフォーマンスおよびスケーラビリティー	効率的な通信が可能になり、企業と ISP の両方のサービス品質が向上します。
豊富なセキュリティー機能	通信とデータの整合性および従業員、顧客、パートナーのプライバシーを保護し、業界の規則を遵守します。
仮想ドメインのホスティングと委任管理	Messaging Server、Calendar Server、および Instant Messaging 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 が組織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 を低減します。
スケーラブル、堅牢、さらに拡張可能なコンポーネント	一体化した通信サービスの配備が可能になり、電話サービスとともに、電子メール通知、ファックス、ポケットベルなどの技術も提供可能になります。
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 で時間とリソースの管理が改善され、ユーザーの生産性が向上します。
会議や予定のグループスケジュール機能	Calendar Server で組織全体のチームコラボレーションや通信が改善します。

表 1–1 Communications Services が組織に対して提供する利点　(続き)

主な機能	利点
ハイパーリンクで予定または作業の情報を共有	Calendar Server では作業または予定に関連した情報の交換によってコラボレーションを促進します。
複数クライアントのサポート	Ximian Evolution や Microsoft Outlook などの複数のリッチ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します。
オープン、モジュール方式、および標準ベースのアーキテクチャー	顧客は、カスタマイズされ、パーソナライズされたソリューションを配備できます。

Communications Services 配備の高可用性の向上

クラスタソフトウェアを使用すると、Messaging Server、Calendar Server、および Instant Messaging で高可用性が実現できるように設定できます。Messaging Server は、Sun™ Cluster および Veritas Cluster Server の両方のソフトウェアをサポートしています。Calendar Server および Instant Messaging は、Sun Cluster ソフトウェアをサポートしています。クラスタソフトウェアの使用時に、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、Messaging Server、Calendar Server、または Instant Messaging 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。この種の配備により、サービスの稼働時間を高レベルにすることができます。このように可用性の高い配備を行うには、サービスアーキテクチャーの各コンポーネントで冗長性が必要になります。このようなコンポーネントには、二重のデータストアサーバー、二重のネットワークインタフェースカード、および二重のシステム記憶装置が含まれます。

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、および Instant Messaging 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、カレンダスケジュール、アドレス帳情報の要約を提供します。Portal Server の統合には、Portal Server、Calendar Express、Messaging Express、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 スキーマ 1 とスキーマ 2 の両方の環境で、Communications Express を実行できます。スキーマ 2 を使用している場合は、Access Manager 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 デスクトップを使用して、Messenger Express、Calendar Express、Instant Messenger クライアントを利用することができます。

Portal Server の次の 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** デスクトップ**:** ユーザーがポートレットから Communications Services アプリケーションにアクセスし、起動できるようにします。
- **Sun Java™ System Portal Server Secure Remote Access:**これにより、リモートエンドユーザーは、インターネットを介して特定の組織のネットワークやそのサービスに安全に接続できます。エンドユーザーは、Secure Remote Access ゲートウェイを介して、Web ベースの Portal Server デスクトップにログインすることで 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で、エンドユーザーが認証されます。エンドユーザーのセキュリティー保護されたセッションが Portal Server との間で確立されると、エンドユーザーの Portal Server デスクトップへのアクセスが有効になります。

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、次の基本フェーズから構成されています。これらのフェーズをソリューションライフサイクルといいます。

- ビジネス要件の分析
- 技術要件の分析
- 論理アーキテクチャーの設計
- 配備アーキテクチャーの設計
- 配備の実行
- 配備の運用

配備フェーズは固定的なものではなく、配備プロセスは反復して行われます。ただし次の各節では、配備フェーズをそれぞれ個別に説明しています。

Communications Services や Java Enterprise System コンポーネントの配備プロセスの詳細については、『Sun Java Enterprise System 2005Q4 Deployment Planning Guide』を参照してください。

ビジネス要件の分析

ビジネス分析フェーズでは、配備プロジェクトのビジネス目標を定義し、その目標を達成するために満たす必要のあるビジネス要件を記述します。ビジネス要件を記述する際には、ビジネス目標の達成に影響する可能性のある、あらゆるビジネス制約を考慮してください。ビジネス分析フェーズの成果物であるビジネス要件文書は、後続の技術要件フェーズで使用されます。ライフサイクル全体を通じて、このビジネス分析フェーズで実施した分析結果に基づいて、配備計画と最終的な配備済みシステムの成功度合いを測定します。

技術要件の分析

技術要件フェーズでは、ビジネス分析フェーズで定義されたビジネス要件とビジネス制約の内容を確認し、それらを配備アーキテクチャー設計時に使用可能な技術仕様書に変換します。技術仕様書には、パフォーマンス、可用性、セキュリティーといったサービス品質に関する基準を記述します。

技術要件フェーズで準備する情報は、次のとおりです。

- ユーザーの作業と使用パターンの分析
- ユーザーと計画中の配備との相互作用をモデル化したユースケース
- ビジネス要件に基づいて作成されたサービス品質要件 (ユーザーの作業と使用パターンの分析結果も考慮)

成果物である使用分析文書、ユースケース文書、およびシステム要件文書が、ソリューションライフサイクルの論理設計フェーズに提供されます。また、技術要件分析フェーズではサービスレベル要件も特定します。これらの要件が、配備済みシステムの障害を解決し、システム要件を満たすべく顧客サービスを提供する上での条件となります。サービスレベル要件は、プロジェクト承認時に締結されるサービスレベル契約の基礎となります。

論理アーキテクチャーの設計

論理設計フェーズでは、配備に必要なサービスを特定します。サービスの特定が完了したら、それらのサービスを提供する論理的に区別されたコンポーネントを、論理アーキテクチャー内にマッピングします。論理アーキテクチャーには、コンポーネント間の依存関係も記載します。論理アーキテクチャーと技術要件フェーズで作成された技術要件仕様書によって、「配備シナリオ」が特徴づけられます。

論理アーキテクチャーには、配備シナリオ実施時に必要となる実際のハードウェアは規定されていません。しかしながら、論理アーキテクチャーは、コンポーネント間の相互関係の視覚化に役立ち、ユースケースと特定された使用パターンをさらに分析するための土台を提供し、配備設計フェーズの開始点となります。

API を使用してサービスを拡張する際や、たとえば企業のブランド設定を導入してルックアンドフィールをカスタマイズする際には、追加の作業が必要なこともあります。

ソリューションによっては、配備やカスタマイズにかなりのコストがかかり、新しいビジネスおよびプレゼンテーションサービスの開発が必要になる場合もあります。他のソリューションの場合、Portal Server デスクトップなどの既存のグラフィカルユーザーインタフェースをカスタマイズすることによって必要な機能の実現が可能な場合もあります。

製品 API の使用や製品機能のカスタマイズについては、適切なコンポーネント製品のマニュアルを参照してください。

- 『Sun Java System Calendar Server 6 2005Q4 Developer's Guide』
- 『Sun Java System Communications Services 6 2005Q4 Event Notification Service Guide』
- 『Sun Java System Messenger Express 6 2005Q4 Customization Guide』
- 『Sun Java System Messaging Server 6 2005Q4 MTA Developer's Reference』

配備アーキテクチャーの設計

設計フェーズでは、論理アーキテクチャー内に指定された論理コンポーネントを、配備アーキテクチャー内の物理コンポーネントにマッピングします。また、配備実施時に役立つ設計文書も作成します。配備設計がうまくいくと、次の成果物が得られます。

- プロジェクトの承認

  プロジェクトの承認は通常、このフェーズで作成された設計文書に基づいて行われます。プロジェクト承認時には配備コストが評価され、承認された場合には、配備実施契約が締結され、プロジェクトを立ち上げるためのリソースが確保されます。実際の承認がどの時点でなされるかは、設計した配備の種類と、その配備を要求している会社の社内方針によって決まります。

- 配備アーキテクチャー

  配備アーキテクチャーとは、論理コンポーネントからネットワークのハードウェアとソフトウェアへのマッピングを表現した、高レベルの設計文書のことです。

- 実施仕様書

  実施仕様書とは、次の文書を含む一連の設計文書のことです。

  - 配備実施時のブループリントとして使用される詳細な設計仕様書
  - ディレクトリサービスを設計および実装するための手順と、システムサービスにアクセスするユーザーのプロビジョニングに必要なデータ構造について概説したユーザー管理計画書
  - 配備の分散インストール手順について概説したインストール計画書
  - 配備の段階的な実施方法やエンドユーザーおよび管理者へのトレーニング方法を記した追加計画書、および配備のスムーズな導入に関係するその他の計画書

配備の実行

実行フェーズでは、配備設計時に作成された設計文書に基づいて配備アーキテクチャーを構築し、配備を実施します。このフェーズでは、個々の配備プロジェクトの特性に応じて次の手順の一部または全部を実行します。

- テスト環境内で、パイロット配備またはプロトタイプ配備、あるいはその両方を作成および配備します
- 機能テストを設計および実行し、システム要件への準拠度を測定します
- 負荷テストを設計および実行し、ピーク負荷時のパフォーマンスを測定します
- 本稼働用の配備を作成します (本稼働環境に段階的に導入してもよい)

配備の本稼働後も引き続き、配備の監視、テスト、および調整を行い、ビジネス目標が確実に達成されるようにする必要があります。

第 2 章

Communications Services の要件の分析

Communications Services 配備の計画においては、まず組織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、次の節があります。

Communications Services や Java Enterprise System コンポーネントの配備プロセスの詳細については、『Sun Java Enterprise System 2005Q4 Deployment Planning Guide』を参照してください。

配備目標の確認

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。組織内のさまざまなソースから、配備の要件があがってきます。多くの場合、要件はあいまいな言葉で表現されますが、それを特定の目標に向けた明確な定義に変える必要があります。

要件分析の結果は、明確で簡潔な言葉で定義し、配備による成果を評価できる目標としてまとめる必要があります。プロジェクト関係者からの同意を得た明確な目標がなければ、先に進んでも成功するのは困難です。

配備を計画する前に検討の必要がある要件には、次のものがあります。

- ビジネス要件
- 技術の要件
- 財務の要件
- サービスレベル契約 (SLA)

ビジネス要件の定義

ビジネスの目標は、配備の決定に大きく影響します。具体的には、ユーザーの行動、サイトの配布、配備に影響を与える潜在的な政治的要因について理解しておく必要があります。これらの業務上の要件を理解していない場合は容易に想定を誤り、配備設計の精度に影響を与えることになりかねません。

運用の要件

直接的な目標を持った一連の機能上の要件として、運用要件を明確にします。通常、次の項目が該当します。

- エンドユーザー機能
- エンドユーザー応答時間
- 可用性 / 稼働時間
- 情報の保存と保持

たとえば、「適切なエンドユーザー応答時間」という要件を評価可能な用語で言い換えて、関係者全員が何が「適切」で応答時間がどのように評価されるかを理解できるようにします。

カルチャーとポリティクス

配備を考える場合、企業のカルチャーとポリティクスを考慮する必要があります。需要というものは、結局はビジネス要件そのものから生み出されてくるものです。例:

- サイトの中には、配備されたソリューションを独自に管理する必要があるものもあります。そのような需要が、プロジェクトのトレーニング費用、複雑さなどを発生させる元となります。
- LDAP ディレクトリに個人情報が含まれている場合、人事部門はそのディレクトリを自己の管理下におきたいと考えるはずです。

技術要件の定義

技術の要件 (または機能の要件) は、組織のシステムニーズの詳細です。

既存の利用率パターンのサポート

既存の利用率パターンを、配備実現のための明確で評価可能な目標として定義します。そのような目標を定義する際に参考となる質問を、次に示します。

- 現在のサービスはどのように利用されていますか。
- ユーザーは分類可能ですか (一時的なユーザー、常用ユーザー、ヘビーユーザーなど)。

- ユーザーはどのような方法でサービスにアクセスしますか (自身のデスクトップから、共有 PC または工場の現場から、ローミングラップトップからなど)。
- ユーザーが通常送信するメッセージのサイズはどのくらいですか。
- カレンダのアポイントには、通常、何人くらいの招待が掲載されていますか。
- ユーザーはメッセージをいくつ送信しますか。
- 通常、ユーザーが日ごとまたは時間ごとに作成するカレンダ予定および作業はいくつですか。
- ユーザーがメッセージを送信するのは、社内のどのサイトですか。
- 必要とされる並行性 (任意の時刻に接続可能なユーザーの数) はどの程度ですか。

サービスにアクセスするユーザーについて調査します。ユーザーはいつ既存のサービスを使うのかといった要素が、配備の要件、ひいては配備の目標を定める重要なポイントとなります。組織の今までの事例からこれらのパターンを得ることができない場合は、他の組織の事例を研究し、推測します。

利用率のきわめて高い部署では、専用のサーバーが必要になる場合もあります。一般に、ユーザーが実際のサーバーから遠く離れており、回線速度も遅い場合、応答時間が長くなります。応答時間が適切であるかどうかを検討する必要があります。

サイトの分散

次の質問を検討して、サイトの分散が配備目標に与える影響を理解します。

- サイトは地理的にどのように分散されていますか。
- サイト間の帯域幅はどれだけありますか。

  集中化方式を採用する場合は、分散化方式よりも広い帯域幅が必要です。ミッションクリティカルなサイトには、専用サーバーが必要です。

ネットワーク

ネットワーク要件の理解に役立つ質問を、次に示します。

- 内部ネットワーク情報をわかりにくくしたいと考えますか。
- ネットワークサービスに冗長性を持たせたいと考えていますか。
- レイヤーホストにアクセスする場合に、利用可能なデータを制限したいと考えていますか。
- エンドユーザーの設定を簡略化したいと考えていますか (たとえば、エンドユーザーを移動する場合に変更が不要な単一のメールホストをエンドユーザーに入力させるなど)。
- ネットワークの HTTP トラフィックを削減したいと考えていますか。

注 – これらの質問に「はい」と答えた場合は、2 層アーキテクチャーをお勧めします。

既存のインフラストラクチャー

より信頼性の高い高可用性帯域幅が利用できる場合は、集中化サーバーを採用できます。

- 既存のインフラストラクチャーと設備で、この配備が可能ですか。
- DNS サーバーは追加の負荷を処理できますか。ディレクトリサーバーはどうですか。ネットワークはどうですか。ルーターはどうですか。スイッチはどうですか。ファイアウォールはどうですか。

サポート要員

24時間、週に 7 日 (24 x 7) 体制のサポートは、特定のサイトでのみ提供されます。少数のサーバーによる簡単なアーキテクチャーの場合は、サポートが容易です。

- 運用グループと技術サポートグループに十分な能力があり、この配備を促進できる状況にありますか。
- 運用グループと技術サポートグループは、配備期間中に増大する負荷に対処できますか。

財務要件の定義

財務上の制約は、配備の構築方法に影響を与えます。財務上の要件は全体的な視点から明確に定義される場合が多く、配備の限界や目標が明確になります。

ハードウェア、ソフトウェア、および保守のための明確なコスト以外に、次のような他のコストがプロジェクト全体に影響を与えます。

- トレーニング
- ネットワーク帯域幅やルーターなどのサービスや設備のアップグレード
- 配備のコンセプトを検証するのに必要な人員やリソースのような配備コスト
- 配備されたソリューションを管理する人員のような運用コスト

プロジェクトの要件に関連する数多くの要素を注意深く分析することで、プロジェクトに関連する財務上の問題を回避することができます。

サービスレベル契約 (SLA) の定義

サービスレベル契約には、稼働時間、応答時間、メッセージ配信時間、および障害回復のような領域に関連する配備を盛り込む必要があります。サービスレベル契約自体には、システムの概要、サポート組織の役割と責任、応答時間、サービスレベルの評価方法、要求の変更などの項目が網羅されています。

サービスレベル契約の範囲を決定する際には、システムの可用性に対する組織の予測が重要なポイントとなります。システムの可用性は、システム稼働時間に対するパーセンテージで表されます。システムの可用性を表す公式は次のとおりです。

```
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で稼働時間が 99.99 パーセントと規定されている場合、1 か月に許されるシステムが使用できない時間は、約 4 分間となります。

さらに、システムの停止時間とは、システムが使用できない時間の合計を意味します。この合計には、システム障害やネットワークの停止などの予期しない停止時間だけでなく、計画された停止時間、予防的保守、ソフトウェアのアップグレードやパッチを当てる時間なども含まれます。システムが 7x24 (週 7 日、24 時間) 稼働を前提としている場合、アーキテクチャーに冗長性を持たせて計画された停止や予期しない停止に備え、高可用性を確保する必要があります。

プロジェクト目標の決定

まず、調査と分析を行なって、プロジェクトの必要要件を明確にする必要があります。次に、明確で評価可能な目標を決定します。プロジェクトに直接関与しない人員でも理解可能な形で目標を設定し、プロジェクトの評価方法も明確にしておきます。

プロジェクト目標は、すべての利害関係者によって承認される必要があります。プロジェクト目標は、プロジェクトの成功を見きわめるために、実装後の検査で計測される必要があります。

拡大のための計画

現在要求されている許容量を決定するだけでなく、計画できる時間枠内で将来必要とされる能力も算出しておく必要があります。拡張のスケジュールは、通常 12 か月から 18 か月です。拡張の例外と利用率特性の変化を考慮して、拡張を検討する必要があります。

ユーザー数とメッセージの数の増加に対応して、容量計画のガイドラインを策定する必要があります。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。収容ユーザー数の増加に伴い、その間の利用率特性も変化します。配備目標 (そして配備設計) は、将来に向けても実現可能なように、状況に応じて対応できるものでなければなりません。

アーキテクチャーが将来の拡張を容易に吸収できるよう設計しておくのが理想的です。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、84 ページの「論理サービス名の使用」を参照してください。稼働段階に入ったら、配備状態を監視して、配備ニーズがいつどのように増加しているかを認識することも重要です。

総所有コスト (TCO) の理解

総所有コスト (TCO) もまた、許容量の計画に影響を与える要素です。これには、Communications Services の配備で選択するハードウェアが含まれます。次の表で、数を多くした小規模なハードウェアシステム、または少数の大規模ハードウェアシステムのどちらを配備するかに関しての検討項目をまとめています。

表 **2–1** 総所有コスト (TCO) の検討

ハードウェアの選択	利点	欠点
数を多くした小規模なハードウェアシステム	■ 小規模なハードウェアシステムは一般にコストが低くなります。 ■ 数を多くした小規模なハードウェアシステムは多くの拠点に配備が可能で、分散型ビジネス環境をサポートします。 ■ 数を多くした小規模なハードウェアシステムでは、サーバーが保守のため停止している場合でも、トラフィックを別のサーバーにルーティングすることでシステム保守やアップグレード、移行のための停止時間を短縮することが可能です。	■ ハードウェアシステムは小規模であればあるほど能力が限定され、必要な数が増えます。維持、管理、および保守のコストはハードウェアシステムの数が増えるにつれて増大します。 ■ 数を多くした小規模なハードウェアシステムでは管理台数が多いため、管理の手間が増大します。
少数の大規模ハードウェアシステム	■ 少数の大規模ハードウェアシステムでは、サーバーごとの固定管理コストが少なくなります。管理コストが、内部または ISP からに関係なく、毎月定期的に請求される場合、管理するハードウェアシステムが少数なのでコストが低くなります。 ■ ハードウェアシステムの数が少ないということは、保守が必要なシステムの数が少ないため、保守、アップグレード、移行の作業が容易になります。	■ 大規模なハードウェアシステムでは、通常、導入時のコストが大きくなります。 ■ 少数のハードウェアシステムでは、保守、アップグレード、移行のための停止時間も長くなります。

第 3 章

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 アーキテクチャーを的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、次の節があります。

さまざまなコンポーネントの計画

Communications Services の配備アーキテクチャーを設計する場合、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ーの種類を計画することができます。配備に応じて、次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 (Access Manager)
- Messaging Server
  - メッセージ転送エージェント (MTA)
  - メッセージストア
  - Messaging マルチプレクサ (MMP)
  - Messaging Express マルチプレクサ (MEM)
- Calendar Server
  - フロントエンド (Front End)
  - カレンダストア (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。

図 3–1 は、別個のホストに配備できるコンポーネントに各サービスを分割する方法と、各コンポーネントが占め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したりすることもできますが、Tier (層) アーキテクチャーに移行することを検討してください。Tier (層) アーキテクチャーには、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、80 ページの「単一層アーキテクチャーの利点」および 80 ページの「2 層アーキテクチャーの利点」を参照してください。

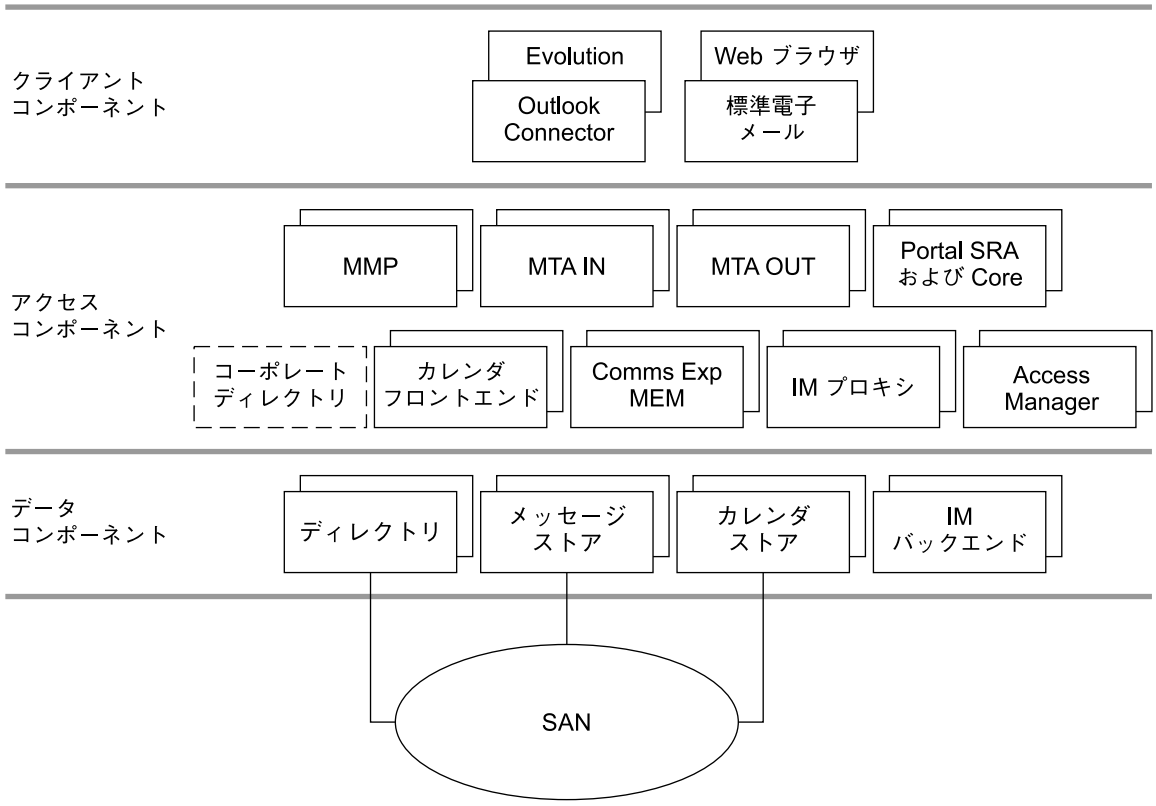

図 **3–1** Communications Services のコンポーネント

この図では、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 thick クライアント、ブラウザ、および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)、認証用の Access Manager、およびアドレス帳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 (本来はフロントエンドおよびバックエンドのコンポーネントから構成できる)、Messaging Server (メッセージストア)、Calendar Server (カレンダストア)、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帳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。

次の節では、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、ドメイン、サブドメイン、ユーザー、およびグループを表すノードを使用して、ディレクトリエントリを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、この構造には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 の下半分では、1 ツリー LDAP 構造を示しています。

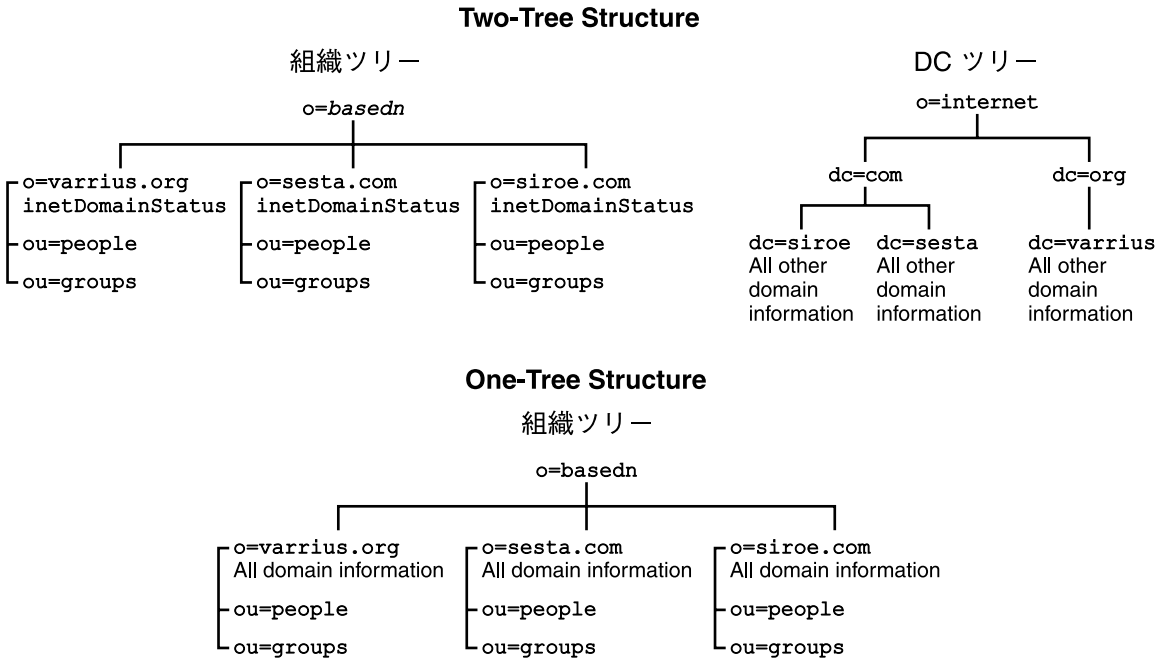

図 **3–2** 2 ツリー LDAP 構造と 1 ツリー構造との比較

1 ツリー DIT 構造の利点

1 ツリー構造のスキーマ 2 ネイティブモードを使用する主な利点は次のとおりです。

- 構造が Access Manager に統合されます。
- 構造が業界標準と緊密に整合されます。
- 構造が 2 ツリー構造よりも大幅に簡素化されます。

次の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)。そのほかのノードは、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。

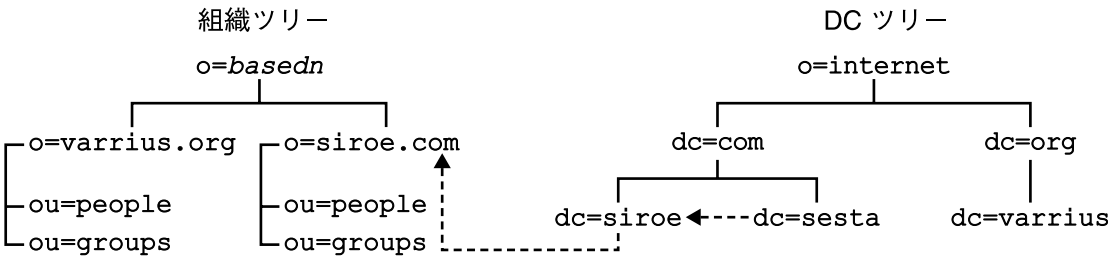

図 **3–3** aliasedDomainName と inetDomainBaseDN を持つ 2 ツリーエイリアス

この図では、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 を使用して組織ツリーの同様の名前が付けられたノードを指定します。

さらに、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内の同じノードを直接指す 1 つまたは複数のノードが存在する可能性があります。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。

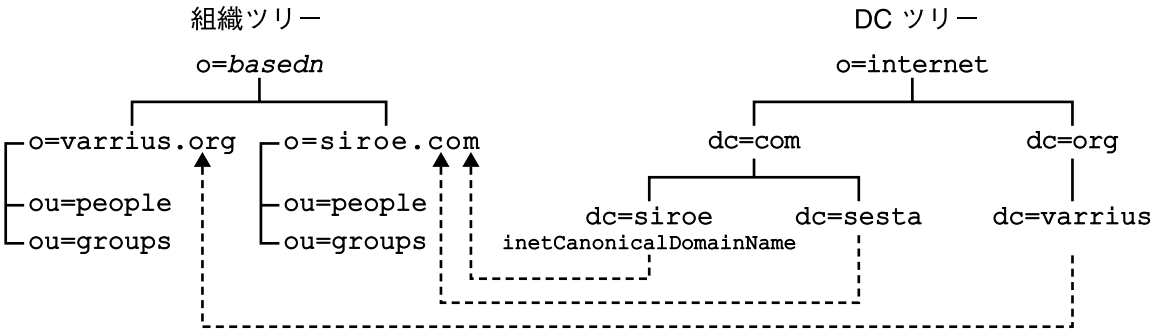

図 **3–4** inetCanonicalDomainName 属性を持つ 2 ツリーエイリアス

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。

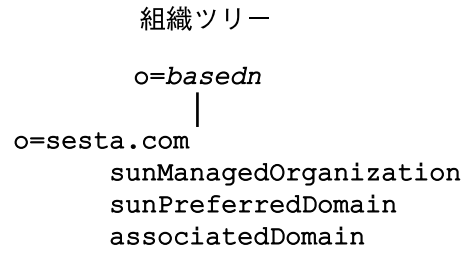

図 **3–5** associatedDomain を持つ 1 ツリーエイリアス

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。各ドメインノードは、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 属性を持つことができます。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 ツリー構造の方が、既存の単一の DIT LDAP アプリケーションに、より適切にマッピングされます。

スキーマの要件

どの Communications Services 製品をインストールする場合でも、使用するスキーマについて事前に理解しておく必要があります。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 スキーマ 1 と Sun Java™ System LDAP スキーマ 2 の 2 つのスキーマが選択肢としてあり、サポートされています。次の基準に従って、スキーマを選択します。

次の場合にスキーマ 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) のために Access Manager 6 2005Q4 (以前の Identity Server) を使用する場合
- Communications Services コンポーネントをはじめて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 Access Manager 6 を使用する必要はありません。スキーマ 2 を選択しても、Access Manager 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。

次の場合にスキーマ 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
- Access Manager SSO も要件である場合を除き、Access Manager を通じてユーザーをプロビジョニングする必要がない場合
- ユーザーのプロビジョニングを行うために Sun ONE Delegated Administrator (以前の iPlanet™ Delegated Administrator) を使用する場合

スキーマ選択の詳細については、第 8 章を参照してください。

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、ユーザープロファイ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、Access Manager、Messaging Server、Calendar Server、Instant Messaging 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、データ環境を設計する数多くの方法と、考慮すべき数多くの要素があります。ディレクトリ設計は次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
- 静的および動的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、リ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ー
- 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、『Sun Java System Directory Server 5 2005Q1 Administration Guide』を参照してください。

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

単一層アーキテクチャーから複数層アーキテクチャーへの移行では、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。

層アーキテクチャーの詳細については、第 5 章を参照してください。

注 – Directory Server は、ディレクトリユーザー管理とソフトウェアアプリケーション設定とを明確に区別してインストールすることが可能です。このアーキテクチャーには 2 つのディレクトリが存在します。1 つはディレクトリホスト上のユーザーおよびグループ用ディレクトリ、もう 1 つは別のホスト上の設定ディレクトリです。ソフトウェアアプリケーション設定の部分を削除する必要がある場合、こうした区別をしておいたほうが、Directory Server から情報を削除する作業が容易になります。

Directory Server の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、マルチマスター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化されます。詳細については、88 ページの「Directory Server と高可用性」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規則はありませんが、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、次の点を監視します。

- 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、カレンダの作成、カレンダへの登録、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場合、LDAP Master Server は LDAP 複製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。これはレプリカ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、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`) がローカルに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。詳細については、249 ページの「Calendar Server LDAP データキャッシュの計画」を参照してください。

Directory Server と個人アドレス帳に関する考慮事項

Messenger Express クライアントは、個人アドレス帳 (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。

注 – 現在の Communications Express クライアントと、非推奨の Messenger Express Web メールインタフェースの両方を実行している場合、これら 2 つのクライアントが使用するアドレス帳は情報を共有しません。2 つのクライアントインタフェースがエンドユーザーによって切り替えられる場合、2 つのアドレス帳には異なるエントリが含まれます。

Messaging Server の考慮事項

Communications Services の開発では、次の 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。

- メッセージストア
- メッセージ転送エージェント (MTA)
- Mail Message Proxy (MMP)
- Messaging Express マルチプレクサ (MEM)

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、165 ページの「Messaging Server アーキテクチャーのパフォーマンスの考慮事項」を参照してください。

Calendar Server の考慮事項

Calendar Server は次の 5 つの主要サービスから構成されています。

- HTTP サービス (`cshttpd`)。HTTP 要求を待機します。HTTP サービスはユーザー要求を受け取り、データを呼び出し元に返します。
- 管理サービス (`csadmind`)。Calendar Server のそれぞれのインスタンスに必要とされます。管理サービスは Calendar Server の認証および管理を 1 ヵ所で行い、また、ほとんどの管理ツールを提供します。
- 通知サービス (`csnotify`)。電子メールまたは予定通知サービスのいずれかを使用して、予定および作業の通知を送信します。
- 予定通知サービス (`enpd`)。予定およびアラーム通知のブローカとして機能します。
- 分散データベースサービス (`csdwpd`)。同じ Calendar Server システム内の複数のデータベースサーバー間でリンクを張り、分散型のカレンダストアを形成します。
- バックアップサービス (`csstored`)。自動バックアップ (アーカイブバックアップとホットバックアップの両方) を実行します。最初のバックアップはログファイルを使用したスナップショットであり、2 番目のバックアップは適用済みログファイ

ルを使用したスナップショットです。このサービスは、`start-cal` コマンド実行時に自動的に起動されます。ただし、インストール時には有効化されないため、このサービスが機能するように設定する必要があります。バックアップサービスを設定しなかった場合、このサービスが設定されていない旨の通知メッセージが、24 時間ごとに管理者に送信されます。

スケーラブルな Calendar Server の配備の場合、フロントエンドシステムをバックエンドサーバーとともに配備することがあります。この場合、フロントエンドシステムにはプロセッサごとに `cshttpd` デーモンのインスタンスが 1 つと、単一の管理サービスが含まれます。バックエンドサーバーには、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、および管理サービスのインスタンスが含まれます。

カレンダサービスのアクティビティのうち、認証と XML/XSLT 変換の 2 つは多大な負荷を生じさせます。サービス品質の要件を満たすために CPU を追加することができます。スケーラブルな環境の場合、このような負荷の高いアクティビティはフロントエンドシステムで実行され、サービス品質の要件に対応するために、個々のフロントエンドシステムに CPU を追加、またはフロントエンドシステムを追加できるようになっています。

注 – 上記は、Communications Express Calendar クライアントを使ってカレンダにアクセスする場合には当てはまりません。Communications Express は WCAP プロトコルを使用して Calendar Server データにアクセスするため、Calendar Server インフラストラクチャは XML/XSLT 変換を行いません。Communications Express の配備の詳細については、パート V「Communications Express の配備」を参照してください。

Calendar バックエンドサービスには、通常、Calendar フロントエンドサービスの CPU の半数が必要とされます。Calendar フロントエンドシステムによってサービス品質をサポートするには、フロントエンドの CPU の 2/3 前後を Calendar バックエンドシステムで使用する必要があります。

カレンダサービスをフロントエンドサービスとバックエンドサービスに分割することを、配備計画の初期の段階で考慮する必要があります。

通常、フロントエンドサービスのコンポーネントである Calendar Server HTTP プロセスは、CPU 時間を多く使用します。したがって、カレンダのピーク使用率を考慮して、予測されるピーク HTTP セッションに対応するため、十分なフロントエンドの処理能力を選択するようにします。通常、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って、Calendar Server フロントエンドの使用可能性が向上します。フロントエンドシステムはカレンダ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、そのようなソリューションを使用する際のハードウェアの追加や管理オーバーヘッドにより、HA の Calendar Server フロントエンドへの配備のコストと時間がかかります。

注 – 本来の HA ソリューションを保証する Calendar フロントエンドの唯一の構成は、Messaging Server MTA ルーターを含む同じホストに Calendar フロントエンドを配備している場合です。ただし、この構成でも、そのようなソリューションのオーバーヘッドについては、利点がわずかなことからして、注意深く比較検討する必要があります。

Calendar Server フロントエンドのハードウェアの適切な選択は、シングルプロセッササーバーまたはデュアルプロセッササーバーです。プロセッサごとに Calendar Server `cshttpd` プロセスのインスタンスを 1 つ配備します。そのような配備によってコスト効率の良いソリューションが提供され、一定レベルの初期のクライアント並行性機能から開始し、ピーク使用率レベルがわかるにつれ、既存の構成にクライアントセッション機能を追加していくことができます。

複数のフロントエンドを配備する場合、フロントエンドサービス全体に負荷を分散するにはスティッキー接続や持続的接続を備えるロードバランサが必要です。

注 – Communications Express は 2 つのプロセッサを超えては拡大されません。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 バックエンド サービスは、リソースの消費で十分にバランスが取れているので、CPU あるいはディスクまたはネットワークなどの I/O のいずれにおいても、ボトルネックが形成されるという証拠はありません。このため、バックエンドのハードウェアな適切な選択は、1 つのストライプボリュームを備える SPARC サーバーになります。そのようなマシンはピーク時の大量のカレンダ負荷に対してかなりの容量を提供します。

要件の中に高可用性がある場合、バックエンドには持続的データが含まれているので、Calendar Server バックエンドを Sun Cluster で配備するのが妥当です。

注 – フロントエンドおよびバックエンドの Calendar Server ホストの両方を持つ構成では、すべてのホスト上で次のソフトウェアが動作している必要があります。

- 同じオペレーティングシステム
- 同じリリースの Calendar Server (パッチやホットフィックスのリリースを含む)

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ーを構成することができます。詳細については、284 ページの「Instant Messaging アーキテクチャー戦略の構築」を参照してください。

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、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、および 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、『Sun Java System Portal Server 6 2005Q4 Deployment Planning Guide』および『Sun Java System Access Manager 7 2005Q4 Deployment Planning Guide』を参照してください。

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/app/docs/coll/1312.1`

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 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。情報源の 1 つは Messaging Server のメールで、もう 1 つは Calendar Server のカレンダ情報です。

ユーザーが Outlook を使用して項目を作成および変更する場合、Connector for Microsoft Outlook は新しいメッセージをメッセージ形式に応じて適切なサーバーに渡します。送信電子メールは SMTP メールサーバーに送信されて配信され、変更された電子メールメッセージはユーザーの IMAP フォルダに返送されて格納されます。新しいカレンダの予定および作業は標準的な形式に変換されて、Calendar Server データベースに格納されます。

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 Messaging Server と 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、次の 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、『Sun Java System Communications Services 2005Q4 リリースノート』の第 6 章「Microsoft Outlook 版 Sun Java System Connector 7 2005Q4 リリースノート」を参照してください。

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 Client Services 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 Client 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、通信およびコラボレーション用の Web ベースの統合クライアントです。Communications Express は Messaging Server と Calendar Server の共通ソフトウェアであり、カレンダ情報、メール、およびアドレス帳に対する Web インタフェースをエンドユーザーに対して提供します。

Communications Express は、カレンダ、アドレス帳、およびメールの 3 つのクライアントモジュールで構成されます。カレンダとアドレス帳のクライアントモジュールは、Web コンテナ上の単一アプリケーションとして配備されます。Messenger Express は、Messaging Server の HTTP サービスを使用する、スタンドアロンの Web ベースのメールアプリケーションです。Messenger Express は、Communications Express と同じシステム上に配備する必要があります。

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

- Directory Server
- Access Manager (Sun Java System LDAP スキーマバージョン 2 を使用する場合)
- Calendar Server
- Messaging Server
- Web Server または Application Server (Web コンテナとして)

S/MIME の考慮事項

Communications Express Mail では、S/MIME (Secure/Multipurpose Internet Mail Extension) のセキュリティ機能が利用可能です。S/MIME を使用するように設定された Communications Express Mail ユーザーは、ほかの Communications Express Mail ユーザーや Microsoft Outlook メールシステムのユーザーと、署名または暗号化されたメッセージを交換できます。

詳細については、330 ページの「Communications Express メールで S/MIME を使用するための要件」を参照してください。

第 4 章

ネットワークインフラストラクチャーに対するニーズの決定

ネットワークインフラストラクチャーはシステムの根本的な基盤です。ネットワークインフラストラクチャーが形成する各サービスによってネットワークの動作構造が作り出されます。Communications Services の配備で、プロジェクトの目標を基準にネットワークインフラストラクチャーを決定することで、拡大縮小や拡張が可能なアーキテクチャーを確保できます。

この章には、次の節があります。

既存ネットワークの理解

既存のネットワークインフラストラクチャーを理解して、それが配備の目標をどの程度満たすものであるかを判断する必要があります。既存のインフラストラクチャーを調査することで、既存のネットワークコンポーネントをアップグレードしたり、新規のコンポーネントを購入したりする必要があるかどうかがわかります。次の領域を調査して、既存のネットワークの完全な全体像を構築する必要があります。

1. ケーブルの長さ、グレードなどの物理的な通信リンク
2. アナログ、ISDN、VPN、T3 のような通信リンクと、サイト間で利用可能な帯域幅と待ち時間
3. 次のサーバーの基本情報
   - ホスト名
   - IP アドレス
   - ドメインメンバー用のドメインネームシステム (DNS) サーバー
4. 次に挙げるデバイスのネットワーク上の場所

- ハブ
- スイッチ
- モデム
- ルーターとブリッジ
- プロキシサーバー

5. モバイルユーザーを含むそれぞれのサイトのユーザー数

この調査結果一覧を完成させた後で、プロジェクトの目標に照らしてその情報を再検討し、配備を成功させるにはどのような変更が必要であるかを判断します。

ネットワークインフラストラクチャーの理解

次の代表的なネットワークインフラストラクチャーコンポーネントは、配備の成否に直接影響します。

- ルーターとスイッチ
- ファイアウォール
- ロードバランサ
- ストレージエリアネットワーク (SAN)
- DNS

ルーターとスイッチ

ルーターはインフラストラクチャーのネットワークを接続して、システム間の通信を可能にします。配備後のルーターの能力には余力を持たせて、プロジェクトの拡大とそれに伴う処理の増加に備える必要があります。

スイッチは、血管のようにネットワーク内のシステムを接続します。

フル稼働状態のルーターやスイッチはボトルネックとなる可能性があり、クライアントが別のネットワーク上にあるサーバーにメッセージを送信するのにかなり長い時間がかかる結果となります。そのような場合には、先見性の欠如や、ルーターやスイッチをアップグレードする資金の欠乏から、そのコスト以上に個人の生産性が低下してしまうこともあります。

ファイアウォール

ファイアウォールは、ルーターとアプリケーションサーバーの間に位置し、アクセス制御を行います。ファイアウォールは本来、信頼されていないネットワーク (インターネット) から信頼済み (内部) ネットワークを保護するものです。現在ではより一般的に、外部ネットワークやインターネットなどの信頼されていないネットワークから、信頼済みまたは隔離された自己のネットワーク上のアプリケーションサーバーを保護する目的で使われています。

ルーターの設定を行うことで、ファイアウォールを通過するデータのスクリーニングを行い、ファイアウォール全体の機能が強化されます。ルーターの設定により、NFS や NIS のような好ましくないサービスをブロックし、パケットレベルのフィルタリングを使用して信頼されていないホストやネットワークからの通信をブロックできます。

さらに、インターネットまたは信頼されていないネットワークに開放されている環境に Sun サーバーをインストールするときに、アプリケーションをホストするのに必要な最小限の数まで、Solaris ソフトウェアのインストールパッケージを減らすことができます。サービス、ライブラリ、およびアプリケーションの数を最小化することにより、保守が必要なサブシステムの数が減少し、セキュリティーの向上につながります。Solaris™ Security Toolkit は、Solaris システムを最小化し、強化し、セキュリティー保護されたシステムにするための、柔軟性と拡張性に富んだメカニズムを提供します。

サイトのセキュリティーポリシーで、このような問題に対する対策を考慮する必要があります。

ロードバランサ

ロードバランサを使用して、Web サーバーまたはアプリケーションサーバー全体の負荷を分散するか、実行するタスクの種類に基づいて要求を分散します。さまざまな専用アプリケーションを異なるアプリケーションサーバーで使用しているような場合は、ユーザーが要求するアプリケーションの種類に応じてロードバランサを使用します。

データセンターが複数ある場合は、ロードバランサの地理的な分散も考慮する必要があります。地理的な負荷分散により、要求やサイトの能力、ユーザーとの距離に基づいて負荷の分散が行われます。1 つのセンターがダウンした場合は、地理的なロードバランサによりフェイルオーバー機能が提供されます。

Web ファーム上のロードバランサでは、サーバーの前とルータの後ろにロードバランサを配置して、トラフィックを適切なサーバーにルーティングします。ソフトウェア負荷分散ソリューションは、Web サーバーにインストールします。ソフトウェアによるソリューションでは、サーバーの 1 つが通常はトラフィックスケジューラとして機能します。

負荷分散ソリューションでは、受信したパケットのヘッダと内容を読み取ることができます。これにより、ユーザーや要求の種類を含むパケット内の情報の種類別に負荷分散を行うことができます。パケットヘッダを読み取る負荷分散ソリューションにより、権限のあるユーザーを識別し、特定のタスクを処理するサーバーに要求を送ることができます。

サービスを提供しているすべてのサーバーとの間で、ロードバランサが動的な通信をどの程度行なっているかを調査する必要があります。スケジューラはそれぞれのサーバーに ping を実行するか「ライブ」なエージェントをサーバー上で作成してロードデータを確認していますか。ロードバランサが TCP パケットをどのように解析しているかも調査する必要があります。そして、ロードバランサがパケットを処理するスピードにも注目します。ロードバランサの中には、他のロードバランサより効率性の高いものもあります。ロードバランサの効率性は、通常スループットで測定されます。

ストレージエリアネットワーク (SAN)

配備を成功に導くためには、ストレージシステムのデータ要件を理解することが必要です。SAN のシステムでは、ストレージをそれが使用されているサーバーから独立した形で配備することが多くなってきています。SAN のシステムを配備することで、ストレージデバイスを再配置することなくマシンを交換することができるため、機能しなくなったサーバーの回復に要する時間を短縮することができます。

次の質問を参考にして、SAN の導入により配備するストレージの要件が適切に達成されているかどうかを評価します。

- 読み取りと書き込みは効果的に行われていますか。
- より高速な I/O ストレージが必要ですか。ストライピングの採用は選択として最適ですか。
- 高い稼働時間率を必要としていますか。ミラーリングの採用は選択として最適ですか。
- データのバックアップはどのような方法で行いますか。バックアップはどのタイミングで行いますか。

DNS (Domain Name System、ドメインネームシステム)

DNS クエリの使用頻度が高いサーバーにはローカルキャッシング DNS サーバーを用意して、ルックアップによる待ち時間を短縮し、ネットワークトラフィックを減らします。

要件を決定する際には、メールストア、メールリレーイン、メールリレーアウトなどの機能別にホスト名を割り当てるようにします。すべてのホスト名が現在 1 台のマシン上でホストされている場合でも、このポリシーを考慮する必要があります。サービスをそのように構成しておくと、そのサービスを別のハードウェアに移すときに、変更に伴う影響をかなり小さくすることができます。

ネットワークインフラストラクチャーレイアウトの計画

インフラストラクチャーのトポロジを考えるときに、次の視点から検討を行う必要があります。

- DMZ
- イントラネット
- 内部ネットワーク
- プロキシ
- ファイアウォールの設定
- モバイルユーザー

非武装地帯 (DMZ)

今日、ほとんどの企業ネットワークで DMZ が取り入れられています。DMZ により、企業ネットワークがインターネットから分離されます。DMZ は厳重に保護された領域で、Web サーバーのようなインターネットサービスと機能を提供するサーバーが配置されます。これらのマシンは、直面する攻撃に耐えられるように強化されています。そのような攻撃によりセキュリティーが破られた場合のエクスポージャーを制限するために、通常これらのサーバーには内部ネットワークに関する情報が含まれていません。たとえば、ネームサーバー機能には、インターネットに接続されたサーバーとルーターしか含まれていません。

さらに進んだ DMZ では、ファイアウォールのセキュリティーと機能がより強固になったことから、DMZ がファイアウォールの後ろのセグメントに移動されています。しかし、DMZ は依然として内部ネットワークからは分離されています。Web サーバー、FTP サーバー、メールサーバー、および外部 DNS をホストするすべてのマシンは、必ず DMZ セグメントに配置する必要があります。

単純なネットワーク設計では、インターネットサービス、VPN アクセス、およびリモートアクセスのための個別の DMZ セグメントだけを定義します。ただし、VPN アクセスとリモートアクセスのトラフィックにはセキュリティー上の問題が存在します。したがって、これらのタイプのトラフィックについては、それ以外のネットワークから分離された適切な接続が必要となります。

DMZ セグメントを提供するファイアウォールは、対応するサービスポートと DMZ 内でそのサービスを提供しているホストに宛てられたインバウンドパケットだけを許可するものでなければなりません。また、DNS やメールのようなサービスを提供するマシンは、そのサービスのためにインターネットにアクセスする必要がありますが、これらのマシンに対するインターネットへのアウトバウンドトラフィックを制限します。要求された接続のタイプにより、DMZ をインバウンド専用とアウトバウンド専用に分けることも 1 つの方法です。しかし、サービス拒否攻撃により DNS や電子メールサービスが妨害される可能性を考えると、インバウンドとアウトバウンド専用

のサーバーに分けてこれらのサービスを提供することには検討の余地があります。電子メールベースのトロイの馬やワームにより、アウトバウンドメールサーバーが制御不能に陥り、オーバーランが発生した場合でも、インバウンドメールは受け取ることができます。DNS サーバーと同じアプローチを適用します。

イントラネット

DMZ は、インターネットへのサービスを提供するホストのためのネットワークセグメントを提供します。この設計により、内部ホストは外部からの攻撃にさらされるホストとは別のセグメントに置かれるため、保護されます。内部的には、内部ユーザーに限定された同様のサービス (Web、ファイルサーバー、内部 DNS など) を提供しています。インターネットサービスをセグメント化するのと同様に、内部サービスもセグメント化します。このような方法によるサービスの分離により、ルーターのフィルタリングでより緊密な制御を行うことができます。

インターネットに向けたサービスを DMZ で分離してセキュリティーを確保したように、私設内部サービスも独自の内部 DMZ 内に配置するべきです。また、ネットワークのサービスとサイズによっては複数の DMZ が有用なように、複数のイントラネットも同様に有用です。

セグメントを提供するファイアウォールの規則は、DMZ のファイアウォールに使用されるものと同様に構成する必要があります。インバウンドトラフィックは、内部メールサーバーに渡されるインバウンドメールのような DMZ からの情報をリレーするマシンと、内部ネットワーク内にあるマシンだけから送られてくるものでなければなりません。

内部ネットワーク

残りのセグメントが内部ネットワークセグメントを構成します。これらのセグメントには、ユーザーのマシンや部署で使用するワークステーションが含まれます。これらのマシンは、イントラネット内のホストからの情報を要求します。開発、ラボ、およびテストネットワークセグメントもこれに含まれます。各内部ネットワークセグメント間のファイアウォールを使用してトラフィックのフィルタリングを行い、部門間のセキュリティーをさらに強化します。これらのセグメント上で使用される内部ネットワークトラフィックとサービスのタイプを識別して、内部ファイアウォールが有効であるかどうかを判断します。

内部ネットワーク上のマシンは、インターネット上のマシンと直接通信してはいけません。これらのマシンでは、DMZ 内のマシンとの直接通信を避けた方が賢明です。これらのマシンが要求するサービスがイントラネット上のホストにあれば理想的です。一方で、イントラネット上のホストは DMZ 内のホストと通信を行なって、電子メールのアウトバウンドや DNS などのサービスを完了することができます。このような間接的な通信であれば問題はありません。

プロキシ

DMZ 内には、インターネット上のマシンと直接通信を行うマシンだけを配置する必要があります。ユーザーがインターネットへのアクセスを要求した場合は、以前のトポロジに基づいた問題が発生します。このような場合は、プロキシが有効です。内部ネットワークセグメントに、またはさらに望ましいのはイントラネットセグメントにプロキシを配置します。インターネットにアクセスする必要のあるマシンは、要求をプロキシに渡し、プロキシがそのマシンに代わって要求を実行します。インターネットへのこのリレーにより、マシンが直面する可能性のある危険を防ぐことができます。

プロキシはインターネット上のマシンと直接通信を行うため、DMZ 内に配置する必要があります。ただしこれは、内部のマシンが直接 DMZ 内のマシンと通信を行うのを防ぐという意図と矛盾します。この通信を間接的なものにするために、二重のプロキシシステムを使用します。イントラネット内の二次プロキシは、内部マシンの接続要求を DMZ 内のプロキシに渡し、そこでインターネットへの直接接続が行われます。

ファイアウォールの設定

通常のパケットフィルタリング機能のほかに、ほとんどのファイアウォールには IP スプーフィングを防ぐ機能もあります。可能な限り IP スプーフィング保護機能を使用してください。

たとえば、インターネットから内部ネットワークへのエントリポイントが 1 つだけで、インターネットからのパケットに内部マシンの発信元アドレスがある場合、それはおそらくスプーフされたものです。ネットワークのトポロジに基づいて、内部マシンの発信元アドレスを持つパケットは、インターネットからではなく内部ネットワークから発信されたものでなければなりません。IP スプーフィングを防ぐことでこのような可能性はほとんどなくなり、IP アドレスベースの認証をすり抜けることも困難になるため、他のファイアウォールの規則を減らすことができます。内部ファイアウォールにも同様の IP スプーフィング対策を行います。

モバイルユーザー

リモートユーザーまたはモバイルユーザーに対しては、どのようにアクセス手段を提供するかを検討する必要があります。そのようなユーザーがアクセスできない手段があるでしょうか。どのようなタイプのセキュリティーポリシーを必要としていますか。SSL による認証が必要ですか。また、モバイルユーザーの数にほとんど変化がないか、今後増加するのかについても検討します。

第 5 章

Communications Services 論理アーキテクチャーの開発

この章では、Communications Services 論理アーキテクチャーを開発する方法について説明します。論理アーキテクチャーとは、Communications Services コンポーネントとそれらをサポートするために必要なインフラストラクチャーサービスの論理構築ブロックを記述した設計のことです。

この章には、次の節があります。

Communications Services 配備の論理アーキテクチャーの概要

Communications Services を単一層または 2 層論理アーキテクチャー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ーの特定は、必要なマシンのタイプと台数を左右するため、非常に重要です。

通常、一般企業への配備では単一層アーキテクチャーが使用され、インターネットサービスプロバイダ (ISP) や電気通信会社への配備では 2 層アーキテクチャーが使用されます。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 層アーキテクチャーに配備することもあります。遠隔地で働く社員のアクセスを容易にする企業が増えるほど、企業の配備も ISP と同様のものになっていきます。

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

- 単一層アーキテクチャー:すべてのサービスが、十分なメモリーと CPU をプロビジョニング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備されます。
- 2 層アーキテクチャー: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ます。
- エッジアーキテクチャー: 2 層アーキテクチャー上に構築され、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ー保護された接続を提供します。

Communications Services を単一層アーキテクチャーと複数層アーキテクチャーのどちらで配備するかにかかわらず、両方のモデルの長所と短所を理解する必要があります。

単一ホスト用の単一層論理アーキテクチャー

その名前が示すように、単一ホスト用の単一層論理アーキテクチャーは、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ーは次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、単一ホスト用の単一層論理アーキテクチャーを示します。

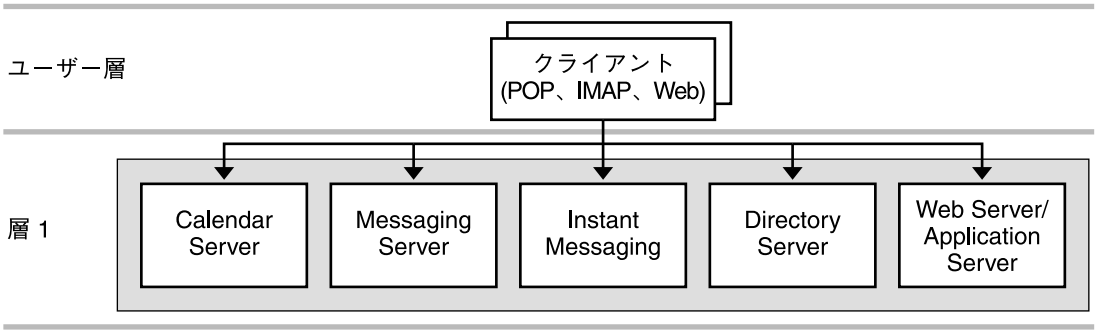

図 **5–1** 単一ホスト用の単一層アーキテクチャー

Outlook、Messenger Express などの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、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。Communications Express を配備する場合、Web サーバー (またはアプリケーションサーバー) も、この単一マシン上で実行されます。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。

単一ホスト用の単一層論理アーキテクチャーでは、十分な CPU、メモリー、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対する組織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。

単一ホスト用の単一層論理アーキテクチャーを実装する場合、サービスに論理名を割り当てることによって複数層アーキテクチャー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) を使用するように指示します。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、84 ページの「論理サービス名の使用」を参照してください。

複数ホスト用の単一層論理アーキテクチャー

複数ホスト用の単一層論理アーキテクチャーは複数のサーバーから構成され、各サーバーがコンポーネント製品に特定のサービスを実行します。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ー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ーの特徴は、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ーは、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ーでは、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。

図 5–2 は、複数ホスト用の単一層論理アーキテクチャーを表しています。

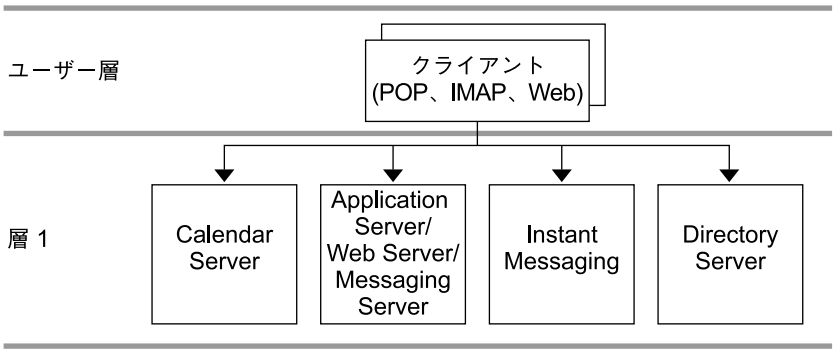

図 **5–2** 複数ホスト用の単一層アーキテクチャー

この図では、Outlook、Messenger Express などの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 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。Communications Express を配備する場合は、Messaging Server のホストに Web メール用の Web サーバー (Web Server または Application Server) も含まれます。

単一層分散論理アーキテクチャー

単一層分散論理アーキテクチャーは、単一層アーキテクチャーの変形で、2 つの層に Directory Server が配備されています。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、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。

図 5–3 は、単一層分散論理アーキテクチャーを表しています。

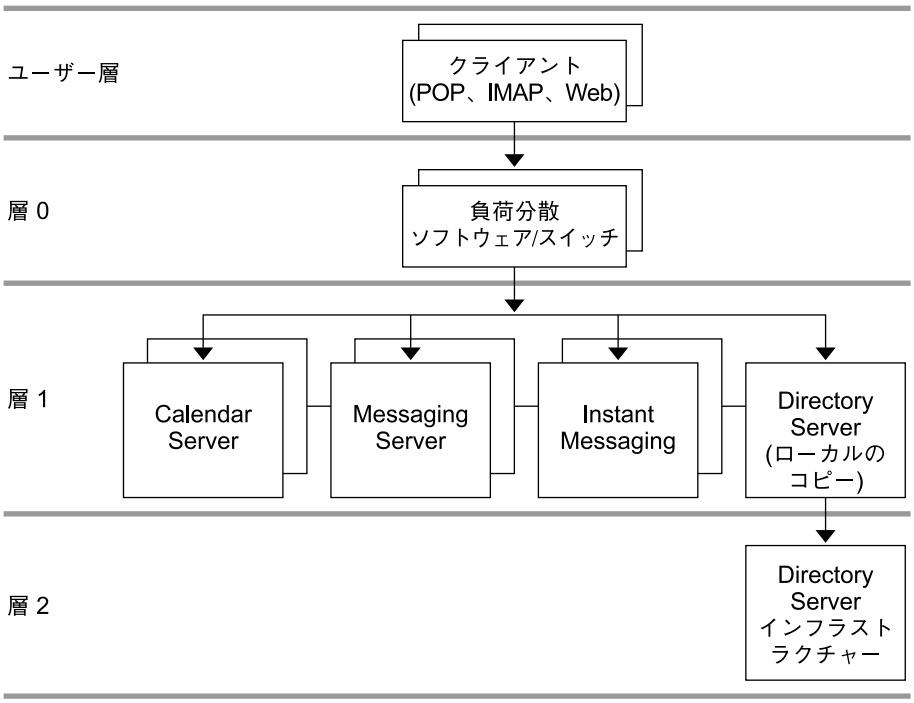

図 **5–3** 単一層分散アーキテクチャー

この図では、Outlook、Messenger Express などの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 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アーキテクチャーを配備する場合は、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になります。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ー

2 層論理アーキテクチャーでは、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。2 層アーキテクチャーを使えば、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集中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ーでは、Directory Server は通常、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マルチマスターとレプリケーションを持つため、それ自体が複雑な配備となります。

図 5–4 は、2 層論理アーキテクチャーを表しています。

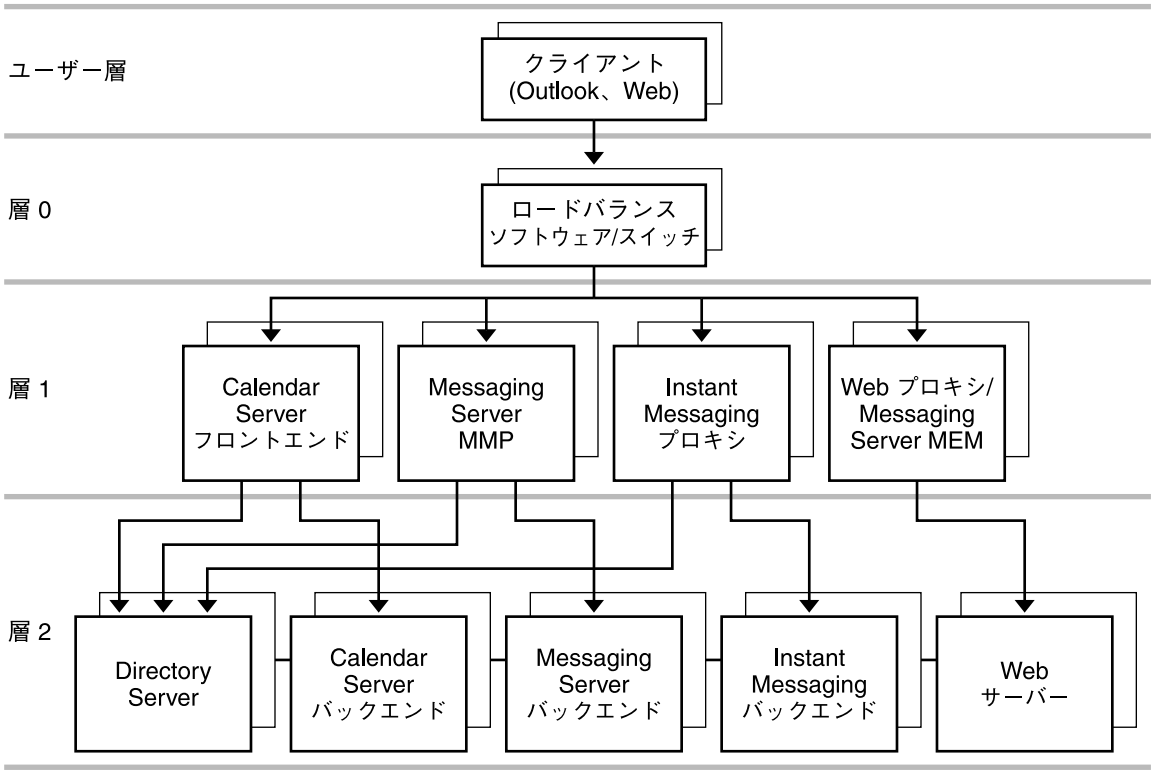

図 **5–4** 2 層アーキテクチャー

上図では、Outlook、Messenger Express などの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、Web プロキシ/MEM の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 のバックエンドが層 2 を形成します。Communications Express を配備する場合は、Web サーバーも層 2 に配置できます。

2 層アーキテクチャーでは、層 1 と層 2 の要素を独立したインスタンスとして配備できるため、設計全体の柔軟性が高まります。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ー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。これらの要素に、バックエンドサーバーの容量を超えるようなスケーリングはしないでください。

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 – 単一層アーキテクチャーおよび 2 層アーキテクチャーにおけるコンポーネントのサイズ決定に関する特別な手順については、クライアントサービス担当者に連絡してください。

エッジ論理アーキテクチャー

エッジ論理アーキテクチャーは 2 層論理アーキテクチャーへのリモートアクセスに対するセキュリティーを向上させます。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公衆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、公衆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。仮想私設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、配備の「エッジ」に含まれ、データストアを無許可の侵入から保護します。

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場合。

図 5–5 は、エッジ論理アーキテクチャーを表しています。

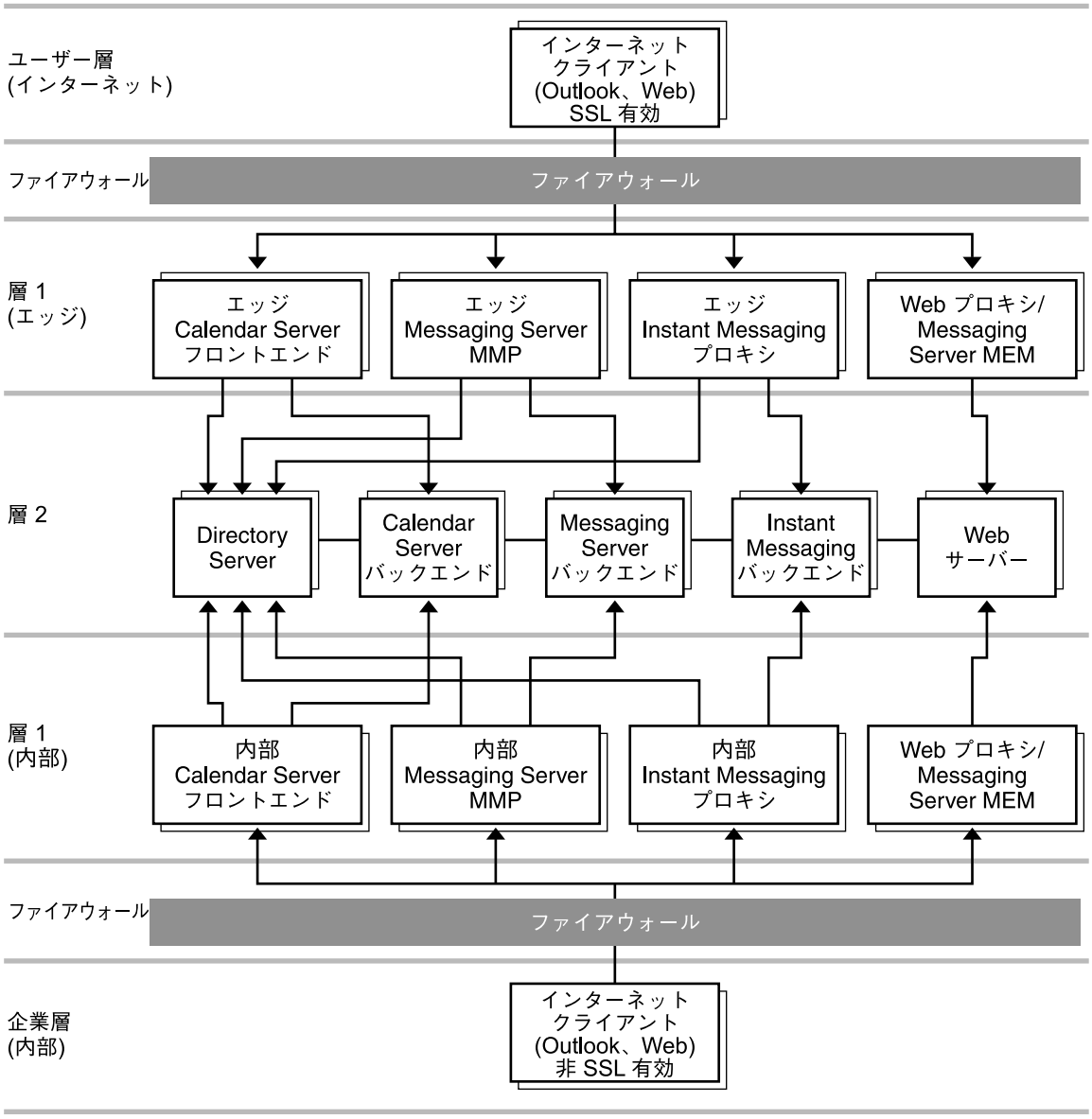

図 **5–5** エッジアーキテクチャー

この図では、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セキュリティー保護された私設ネットワークの層 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ー設計の推奨事項

- 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で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ーの利点

単一層アーキテクチャーを使用する利点は、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。

単一層アーキテクチャーがその企業に最適かどうかを決定するには、次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ですか。
- SSL は必要ありませんか。
- 保守のためにかなりの停止時間を見込むことができますか。

これらの質問に対する回答が「はい」である場合、その企業は単一層アーキテクチャーを使用することを示しています。

2 層アーキテクチャーの利点

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 層アーキテクチャーは、公衆 (ユーザー側) ネットワークと私設 (データセンター側) ネットワークの 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。

ネットワークを 2 層に分離することにより、次の利点が提供されます。

- 内部ネットワークの非表示**:** 公衆 (ユーザー側) ネットワークと私設 (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、セキュリティーが確保されます。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プロビジョニングすることで、システムの冗長性が実現されます。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が危険にさらされている場合も、攻撃者はアクセスホストから重要なデータを取得できません。
- 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 (171 ページの「MTA パフォーマンスの考慮事項」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 層アーキテクチャーの使用により、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。カレンダサービスの場合、カレンダ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ストアホストへのバックエンド接続を構築するために、Directory Server を使用してカレンダ検索を提供します。

  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。すべてのメールの着信接続が 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 Calendar Server 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 Calendar Server ホストのカレンダ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をあるメールボックスから別のメールボックスに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 マルチプレクサ) と Calendar Server 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 HTTML の要素の場合、メールまたはカレンダサービスに別々の 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 ファイルなどの静的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。

水平方向のスケーラビリティー戦略

スケーラビリティーは、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ーを拡張する必要がある組織に不可欠です。次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の反応: システムがどのようなパフォーマンスを提供するか。また、作業負荷の増加に伴って、システムがクラッシュするか、それともパフォーマンスが正常に低下するか。
- 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 層アーキテクチャーに配備する場合、Communications Services 製品は水平方向に非常に効果的なスケーリングが可能です。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。

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ーリング

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ー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ーリングプロセスを開始します。比較的低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、およびスケーラビリティー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ー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ーの場合、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を対応できるように拡大することができます。このような設計により、ユーザーに関してでも、スループットに関してでも、柔軟に拡大できるようになります。

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 に対する LMTP (Local Message Transfer Protocol) 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削除することにより、IOP を 30% 程度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ー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 層アーキテクチャーが必要です。LMTP の設定手順については、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』の「LMTP 配信の設定」を参照してください。

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、一方的に送られてくる大量電子メール (UBE: Unsolicited bulk email) の既知の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、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拒否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 ディスパッチャーの `ENABLE_RBL` オプションの設定に関する詳細については、『Sun Java System Messaging Server 6 2005Q4 Administration Reference』を参照してください。

論理サービス名の使用

Communications Services サーバーの論理名を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 つはインバウンド 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 **5–1** 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
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 **5–2** 保守レベルの論理名

例	説明
relay-in.siroe.com	インバウンド SMTP サーバーのバンクに相当します。
relay-out.siroe.com	アウトバウンド SMTP サーバーのバンクに相当します。
mmp.siroe.com	MMP サーバーのバンクに相当します。
mem.siroe.com	MEM サーバーのバンクに相当します。
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します。
calstoreAA.siroe.com	バックエンドカレンダストア。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します。

表 **5–3** ユーザーレベルの保守レベル論理名へのマッピング

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 のうちの 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
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

第 6 章

サービスの可用性の設計

論理アーキテクチャーを決定したら、次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ー、使用する保守方法が関係しています。この章では、いくつかの選択肢とそのメリットおよびコストについて説明します。

この章には、次の節があります。

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は、製品ごとに異なります。

たとえば、Messaging Server 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 は、これら各ソリューションに対するエージェントを提供します。

Messaging Server と Calendar Server は、異なるクラスタトポロジをサポートしています。詳細については、対応するクラスタ製品のマニュアルを参照してください。

Application Server を Web コンテナとして使用する場合は、その高可用性、負荷分散、およびクラスタ管理機能を利用できます。

Instant Messaging は Sun Cluster エージェントも提供しますが、VCS はサポートしません。冗長性のある Instant Messaging マルチプレクサの配備によって、さらに可用性の高い配備を実現できます。そのような配備では、あるマルチプレクサで障害が発生しても、それとは別の利用可能なマルチプレクサ経由で、Instant Messaging クライアントはバックエンドサーバーと通信できます。

また、Directory Server などのインフラストラクチャーコンポーネントの可用性を高くすることで、Communications Services 配備の可用性をさらに高めることができます。

この章の次の節では、各コンポーネントで使用できるオプションについて説明します。

システムの自動再設定 (ASR)

単に高可用性 (HA) ソリューションを評価するだけでなく、ASR を可能にするハードウェアの配備についても検討する必要があります。

ASR は停止時間に関連するハードウェア障害を最小限にするためのプロセスです。サーバーに ASR 機能がある場合は、ハードウェアの個別のコンポーネントに障害が発生しても、停止時間を最小限にとどめることが可能になります。ASR により、サーバーの自動再起動と、障害の発生したコンポーネントが交換されるまでそれを停止しておくことが可能になります。欠点は、障害の発生したコンポーネントがサービスから排除される結果、システムのパフォーマンスが低下することです。たとえば、CPU に障害が発生すると、マシンは残りの CPU を使用して再起動されます。システムの I/O ボードまたはチップに障害が発生した場合は、システムの I/O ボードが減少するか、代わりの I/O パスが使用されます。

さまざまな Sun SPARC システムが、さまざまなレベルの ASR をサポートしています。ASR をまったくサポートしていないシステムもあれば、非常に高レベルの ASR をサポートしているシステムもあります。当然ながら、高い ASR 機能を持つサーバーはその分コストも高くつきます。ソフトウェアに高可用性がない場合、コストには制約がないものとすれば、データ格納用にはハードウェアに高い冗長性と ASR 機能を持たせたマシンを選択します。

Directory Server と高可用性

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ーサービスとして、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次の利点を提供します。

- 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 **6–1** 高可用性 Directory Server の設計

手法	説明
シングルマスターレプリケーション	サプ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レプリカを直接コピーします。この構成では、すべてのディレクトリの変更がサプライヤに格納されたマスターのレプリカに対して行われ、コンシューマには読み取り専用のデータのコピーが含まれます。
双方向、マルチ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 つのサプライヤ間のマルチマスター環境で、2 つのレプリケーションアグリーメントを作成します。サプライヤ A とサプライヤ B がそれぞれ同一データのマスターのレプリカを保持し、このマルチ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アグリーメントが存在します。
4 方向、マルチマスター	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します。この構成は、レプリケーションに 4 方向、マルチマスターレプリケーション (MMR) を使用します。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完全性を保証する高可用性ソリューションを提供しています。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します。
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ます。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します。ただし、クラスタのインストール、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ります。

詳細については、『Sun Java System Directory Server 5 2005Q1 Deployment Plannning Guide』を参照してください。

Application Server と高可用性

Communications Express は Web コンテナ (Web Server または Application Server) 内に配備されるため、Web コンテナの高可用性化を検討してください。

たとえば、Application Server Enterprise Edition はコアアプリケーションサーバープラットフォームの機能強化版であり、高可用性化機能、負荷分散機能、およびクラスタ管理機能を備えています。Enterprise Edition では、Platform Edition の管理機能が拡張されており、複数インスタンス、複数マシンの配備にも対応できるようになっています。

Application Server のクラスタリングサポートには、設定の容易なクローンアプリケーションサーバーインスタンスグループが含まれます。このグループを使えば、クライアント要求の負荷分散を実現できます。このエディションでは、外部ロードバランサと負荷分散機能を備えた Web 層ベースプロキシの両方が、サポートされています。Application Server EE は、HADB (高可用性データベース) を使って HTTP セッションとステートフルセッション Bean に対するフェイルオーバーを実現します。

詳細については、次の Application Server Enterprise Edition 8.1 2005Q2 のマニュアルを参照してください。

`http://docs.sun.com/app/docs/coll/1310.2`

Messaging Server、Calendar Server と高可用性

クラスタソフトウェアを使用すると、Messaging Server および Calendar Server で高可用性が実現できるように設定できます。Messaging Server は、Sun Cluster および Veritas Cluster Server の両方のソフトウェアをサポートしています。Calendar Server は、Sun Cluster ソフトウェアをサポートしています。

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 アーキテクチャーでは、バックエンドが持続的データを保持する「ストア」となるため、クラスタテクノロジ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。クラスタテクノロジは、持続的データを保持していないため、通常は Messaging Server または Calendar Server のフロントエンドでは保障されていません。通常、Messaging Server の MTA、MMP、MEM と Calendar Server のフロントエンドの可用性を高めるには、システムを冗長化します。つまり、複数のフロントエンドホストを配備します。また、RAID テクノロジによって MTA のディスクサブシステムを保護することで MTA を高可用性化することも可能です。

Sun Cluster トポロジの詳細については、『Sun Cluster Concepts Guide for Solaris OS』の第 2 章「Key Concepts for Hardware Service Providers」を参照してください。

Messaging Server の高可用性の設定について詳しくは、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』の第 3 章「高可用性の構成」を参照してください。

Calendar Server の高可用性の設定について詳しくは、『Sun Java System Calendar Server 6 2005Q4 Administration Guide』の第 7 章「Configuring for High Availability (Failover Service)」

Instant Messaging と高可用性

Instant Messaging は Sun Cluster エージェントを提供しますが、Veritas Cluster Service をサポートしていません。Instant Messaging マルチプレクサの配備を冗長化したり、Instant Messaging ウォッチドッグプロセスを活用したりすることで、より可用性の高い環境を実現できます。

Instant Messaging の高可用性の概要

Sun Cluster エージェントを使用して高可用性 (HA) を実現するために Instant Messaging を設定すると、ソフトウェアとハードウェアの障害の監視および復旧機能が提供されます。高可用性機能はスケーラブルサービスではなくフェイルオーバーデータサービスとして実装され、現時点では Solaris 上でのみサポートされています。

注 – 同じ SMTP サーバーを使用することで、1 つの HA 環境内に複数の Instant Messaging ノードを配置することができます。

Sun Cluster エージェントを使用して Instant Messaging の HA 環境を実装する前に、次のどの HA 配備がもっともニーズに適しているかを決定します。

- 混合 **HA** 環境**:** この配備はローカル設定とバイナリ、およびグローバル実行時ファイルから構成されます。この設定の利点は、Instant Messaging がオフラインであるノード上でアップグレードを行えることにより、最小限の停止時間で Instant Messaging をアップグレードできることです。欠点は、クラスタ内のすべてのノード上で Instant Messaging の設定とバージョンの統一を保証しなければならないことです。加えて、このオプションを選択する場合、グローバル実行時ファイル用に HAStoragePlus またはクラスタファイルシステムのどちらを使用するのかを決定する必要があります。
- グローバル **HA** 環境**:** この配備はグローバル設定、バイナリ、および実行時ファイルから構成されます。この設定は管理が容易ですが、アップグレードの前に、クラスタ内のすべてのノード上で Instant Messaging を停止させる必要があります。

複数の Instant Messaging マルチプレクサの使用

複数のマルチプレクサを含む Instant Messaging 配備では、あるマルチプレクサで障害が発生しても、それとは別の利用可能なマルチプレクサ経由で、Instant Messaging クライアントはバックエンドサーバーと通信できます。現時点では、複数のマルチプレクサが単一の Instant Messaging サーバーインスタンスと通信するようにしか設定できません。複数のマルチプレクサが複数の Instant Messaging インスタンスと通信するように設定することはできません。

Instant Messaging ウォッチドッグプロセスの使用

Instant Messaging にはウォッチドッグプロセスが含まれています。このプロセスは Sun Cluster エージェントを監視し、サーバーのロックアップやクラッシュなど、何らかの理由により利用不可能になったサービスを再開します。ウォッチドッグプロセスを設定した場合、ある Instant Messaging コンポーネントの機能が停止すると、ウォッチドッグプロセスが、そのコンポーネントをシャットダウンしてから再起動します。

有効化テクニックとテクノロジの使用

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には、ロードバランサ、Sun Java System Directory Proxy Server 、レプリカ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、アーキテクチャー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、単一のアプリケーションインスタンス、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可能な同時 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/app/docs/coll/1317.1`

レプリカ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レプリカ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、レプリカハブをマルチマスターサプライヤに昇格させる、またはその逆を行うことができます。コンシューマをレプリカハブのロールに昇格させる、またはその逆を行うこともできます。ただし、コンシューマを直接マルチマスターサプライヤとして昇格させること、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レプリカハブとなり、次にハブからマルチマスターのレプリカ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。

レプリカ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。マルチマスターサプ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、ほかのマルチマスターサプライヤの 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、『Sun Java System Directory Server 5 2005Q1 Administration Guide』を参照してください。

高可用性製品の参照情報

高可用性モデルの詳細については、次の製品マニュアルを参照してください。

Sun Cluster

- 『Sun Cluster Concepts Guide for Solaris OS』
- 『Sun Cluster Data Services Developer's Guide for Solaris OS』
- 『Sun Cluster Overview for Solaris OS』

- 『Sun Cluster System Administration Guide for Solaris OS』

Veritas Cluster Server

- 『Veritas Cluster Server User's Guide』`http://seer.support.veritas.com/docs/275725.htm`

リモートサイトフェイルオーバーの理解

リモートサイトフェイルオーバーは、プライマリサイトに致命的な障害が発生した場合に、そのプライマリサイトに WAN で接続されているサイトでサービスを開始する機能です。リモートサイトフェイルオーバーにはいくつかの形式があり、それぞれにコストが異なります。

リモートサイトフェイルオーバーでは、すべてのケースでサーバーとストレージを追加して、リモートサイトにインストールおよび設定された、サービスのユーザー負荷のすべてまたは一部を処理する能力を持つようにする必要があります。すべてまたは一部というのは、顧客によっては優先するユーザーとそうでないユーザーがいることを意味します。ISP でも企業でも、そのような状況が起こります。ISP には、この機能のために割増料金を支払うユーザーがいます。企業では、全従業員に電子メールの機能を提供している部門内で、ユーザーによってはそのサポートが高くついている場合があるかもしれません。たとえば、カスタマサポートに直接関わるユーザーのメールに対してリモートサイトフェイルオーバーを選択した場合でも、製造ラインで勤務する従業員には、リモートサイトフェイルオーバーを用意しないというケースが考えられます。リモートハードウェアは、このようにリモートサイトフェイルオーバーメールサーバーにアクセスを許可されたユーザーの負荷を処理できます。

ユーザーベースの使用率だけを制限すると、必要な冗長サーバーとストレージハードウェアの数を減らすことができますが、フェイルバックの設定と管理も複雑になります。そのようなポリシーはまた、長期的には予期しない別の影響をユーザーに与えます。たとえば、ドメインメールルーターが 48 時間にわたって利用不能になった場合、インターネット上の他の MTA ルーターがそのドメイン宛てのメールを保持します。ある時点でサーバーがオンラインに戻った際に、メールが配送されます。さらに、すべてのユーザーをフェイルオーバーリモートサイトに設定していない場合は、MTA が起動して設定されていないユーザーに対して永続的なエラー (バウンス) が返されます。最後に、すべてのユーザーを受け入れるようにメールを設定している場合は、すべてのユーザーをフェイルバックするか、フェイルオーバーがアクティブな間使用できないアカウント宛てのメールを保持し、フェイルバックが起こったらそれを本来の配信の流れに戻すように MTA ルーターを設定する必要があります。

考えられるリモートサイトフェイルオーバーのソリューションには、次のようなものがあります。

- 単純でコストのかからないシナリオ: リモートサイトの接続に広帯域幅の大規模ネットワークを使用しません。十分な規模のハードウェアを必ずしも使用する必要はありません。実際のところ、ハードウェアは当面の間はほかの目的に使用できま

す。プライマリサイトからのバックアップがリモートサイトに対して定期的に提供されますが、必ずしも復元の必要はありません。予想される問題点としては、古いデータをオンラインに戻す際に重要なデータが失われたり、かなりの遅れが生じることです。プライマリサイトで障害が発生したときには、手動でネットワークを変更してサービスを開始します。サービスを開始したら、続いて `imsrestore` プロセスを開始します。最後にファイルシステムの復元が開始されると、続いてサービスが開始されます。

- より複雑で、よりコストのかかるソリューション**:** Veritas および Sun の市販ソフトウェアソリューションでは、ローカル (プライマリ) ボリュームで発生するすべての書き込みがリモートサイトにも書き込まれます。通常の製品では、リモートサイトはプライマリサイトとともにロックステップかそれに近い状態になります。プライマリサイトで障害が発生した場合は、セカンダリサイトがネットワーク設定をリセットし、データをほとんど失うことなくサービスを提供できます。このシナリオでは、テープから復元する意味はありません。プライマリサイトの障害の前に切り替えられなかったデータは、少なくともフェイルバックが起きるか、MTA キューデータの場合には手動による介入が行われるまで、失われることになります。Veritas Site HA ソフトウェアは、プライマリサイトの障害を検出し、ネットワークをリセットしてサービスを起動する用途でよく使われますが、これはより高いレベルのデータ保管には必要ありません。サーバーがデータをコピーするための負荷と待ち時間が大きく増加するため、このソリューションでは、プライマリサイトで必要なハードウェアの台数が大幅に増加します。
- 最も実現性の高いソリューション**:** このソリューションは、データのコピーがメッセージストアサーバーで行われることを除いて、ソフトウェアによるリアルタイムデータコピーソリューションと本質的には同じものです。Message Store サーバーが、リモートのレプリケーションをサポートするストレージアレイに接続されている場合、リモートサイトに対するデータコピーはストレージアレイのコントローラそれ自体によって処理されます。リモートのレプリケーション機能を提供するストレージアレイは大容量になりがちなため、このソリューションを導入するための基本コストは、ローエンドストレージ製品を使用する場合よりも高くつきます。

ハードウェアやソフトウェアをはじめ、管理コスト、電力費、光熱費、ネットワークコストまで、これらのソリューションでは、さまざまなコストが発生します。これらのコストはすべてそのまま計算に入れて、数字をはじき出します。そうしなければ、いくつかのコストを算出するのが困難になります。そのようなコストには、めったに実施しない一連の手順を実施するときのミスによるコスト、停止時間による直接のコスト、データ損失によるコストなどがあります。このような種類のコストを正確に算定するのは不可能です。顧客によっては、停止時間とデータの損失は代償が高くつくか、まったく受け入れられません。別の顧客にとっては、それは単なる不愉快にすぎないかもしれません。

リモートサイトフェイルオーバーを行う場合には、リモートディレクトリが少なくとも最新のもので、メッセージデータの復元が可能な状態にあることも必要です。リモートサイトにリストアメソッドを使用する場合は、ディレクトリが完全に復元されてからメッセージを復元する必要があります。また、ユーザーをシステムから削除した場合、ディレクトリ内でそのユーザーに無効のタグがつけられるだけなのはやむをえません。ユーザーのデータがあるメッセージバックアップテープが使用されるかぎり、それらのユーザーをディレクトリから削除してはなりません。

リモートサイトフェイルオーバーについての質問

次の質問を参考にして、リモートサイトフェイルオーバーの計画を立ててください。

- サイトで必要とする応答性のレベルはどの程度か

  組織によっては、プライマリサイトで障害が発生したときに、手動処理のスクリプトセットで十分対応可能な場合もあります。短時間 (数分間) のうちにリモートサイトがアクティブになる必要がある組織もあります。そのような組織では、Veritas リモートサイトフェイルオーバーソフトウェアかそれに相当する機能を持ったその他のソフトウェアが必要です。

注 – ローカル HA 用の Sun Cluster とリモートサイトフェイルオーバー用の Veritas ソフトウェアを併用しないでください。Sun Cluster は現時点ではリモートサイトフェイルオーバーをサポートしていません。

また、プライマリサイトからバックアップサイトへの自動フェイルオーバーをソフトウェアに許可しないでください。その場合、セカンダリサイトからプライマリサイトの障害が誤って検出される可能性がかなり高くなります。このようなケースでは、ソフトウェアにプライマリサイトを監視させ、障害を検出したときに警告を出させるように設定します。次に、バックアップサイトへの自動フェイルオーバープロセスを開始する前に、障害が実際に発生していることを確認します。

- どれぐらいのデータを保存し、どの程度の速さで利用可能にする必要があるか

  これは単純な質問のようですが、細分化されて回答の幅は広くなります。シナリオには、簡単なものからほとんど完全なものまであり、ハードウェア、ネットワーク、データインフラストラクチャー、保守のコストの面でも大きな違いがあります。

第 7 章

セキュリティーの設計

この章では、セキュリティーの手法の概要、一般的なセキュリティー脅威、およびセキュリティーニーズ分析の手順の概要について説明します。

この章には、次の節があります。

製品のセキュリティーの詳細については、第 13 章と第 18 章を参照してください。

Communications Services セキュリティーの概要

Communications Services 配備のセキュリティーは、「多重防御」手法を採用することによって管理します。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、アーキテクチャーの各層のセキュリティーを確保します。セキュリティーには、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、層をセキュリティー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ー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、ユーザーを盗聴、不許可の使用、または外部からの攻撃から保護するためにいくつかのセキュリティーレベルを提供します。基本レベルのセキュリティーは認証によるものです。Calendar Server は、デフォルトの設定で LDAP 認証を使用していますが、代替の認証方法が必要とされる場合、認証プラグインの使用もサポートしています。Access Manager と統合すれば、Calendar Server はそのシングルサインオン機能を利用することができます。

Instant Messaging は、複数の認証メカニズムとセキュリティー保護された SSL 接続によって通信の統合を可能にします。Portal Server と Access Manager との統合によって、追加セキュリティー機能、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、セキュリティー保護されたリモートアクセスが可能になります。

注 – 完全かつセキュリティー保護された環境を確保するために、配備にはセキュリティー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。

セキュリティー戦略の作成

セキュリティー戦略の作成は、配備計画のなかで最も重要なステップの 1 つです。セキュリティー戦略は、組織のセキュリティーに対するニーズを満たし、ユーザーに不便を強いることなくセキュリティーが確保されたメッセージ環境を提供するものでなければなりません。

さらに、セキュリティー戦略は単純なものにして、管理を容易に行えるようにしておく必要があります。複雑なセキュリティー戦略を用いると、ユーザーがメールにアクセスできなかったり、ユーザーや権限のない侵入者によってアクセスされては困る情報が変更されたり、収集されたりする問題が生じます。

RFC 2196『Site Security Handbook』に記載されたセキュリティー戦略を構築するための 5 つのステップを、次に示します。

1. 何を保護するのかをはっきりさせます。

   たとえば、保護対象のリストにはハードウェア、ソフトウェア、データ、従業員、文書、ネットワークインフラストラクチャー、または組織の評判などが含まれます。

2. 何から保護するのかを判断します。

   例: 権限のないユーザー、スパマー、またはサービス拒否攻撃

3. システムに対する脅威の可能性を推測します。

   大規模なサービスプロバイダの場合、セキュリティーが脅威に晒される可能性は小規模な組織よりもはるかに高いといえます。さらに、組織の性格がセキュリティーに対する脅威を誘発することも考えられます。

4. 費用対効果の高い方法で資産を守る対策を導入します。

   たとえば、SSL 接続を設定する際のオーバーヘッドによって、メッセージング配備のパフォーマンスに対する負荷が発生する可能性があります。セキュリティー戦略を設計するうえで、セキュリティーニーズとサーバーの能力のバランスを取る必要があります。

5. 戦略を常時見直し、弱点が発見されるたびに戦略を練り直して、よりすぐれたものに改善します。

   定期的な監査を行い、セキュリティーポリシーの全体的な有効性を検証します。監査は、ログファイルと SNMP エージェントが記録した情報を調査することで行います。SNMP の詳細については、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』を参照してください。

セキュリティー戦略では、次の項目についても計画する必要があります。

- 99 ページの「物理的なセキュリティー」
- 99 ページの「サーバーセキュリティー」
- 99 ページの「オペレーティングシステムのセキュリティー」
- 100 ページの「ネットワークセキュリティー」
- 101 ページの「メッセージングセキュリティー」
- 101 ページの「アプリケーションのセキュリティー」

物理的なセキュリティー

インフラストラクチャーの重要な部分への物理的なアクセスを制限します。たとえば、ルーター、サーバー、配線クローゼット、サーバールーム、データセンターを、窃盗、改竄、その他の悪用から保護するために、物理的な制限を設けます。権限を持たない人物にサーバールームへの侵入を許し、ルーターの配線を抜かれることがあるようでは、ネットワークとサーバーのセキュリティーも無意味なものとなります。

サーバーセキュリティー

重要なオペレーティングシステムアカウントとデータへのアクセスを制限することも、セキュリティー戦略の一部となります。この保護は、オペレーティングシステムで利用できる認証とアクセス制御のメカニズムにより行われます。

さらに、最新のオペレーティング環境のセキュリティーパッチをインストールし、数ヶ月ごとに、またベンダーからのセキュリティー警告に対応して、パッチを更新する必要があります。

オペレーティングシステムのセキュリティー

運用環境におけるセキュリティー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」と呼ばれる次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ていないネットワークに開放されている環境に Sun サーバーをインストールするときに、アプリケーションをホストするのに必要な最小限の数まで、Solaris ソフトウェアのインストールパッケージを減らすことができます。サービス、ライブラリ、およびアプリケーションの数を最小化することにより、保守が必要なサブシステムの数が減少し、セキュリティーの向上につながります。

  Solaris Security Toolkit は、Solaris システムを最小化し、強化し、セキュリティー保護されたシステムにするための、柔軟性と拡張性に富んだ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ー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://wwws.sun.com/software/security/jass`

- ファイルシステムの変更の追跡と監視**:** セキュリティー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、危険性のある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。

ネットワークセキュリティー

水平方向のスケーラビリティーとサービスのセキュリティーの両方をサポートするには、ファイアウォールの背後にアーキテクチャーのアクセス層を配置する配備構成をお勧めします。2 層アーキテクチャーでは、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拡大縮小して、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。

ネットワークへのアクセスを制限することは、セキュリティー戦略の重要なポイントとなります。通常は、ファイアウォールを使用してネットワークへの全般的なアクセスを制限します。ただし、電子メールはサイト外から使用できるようにしておく必要があります。SMTP がそのサービスの 1 つに該当します。

ネットワークのセキュリティーを確保するには、次の条件が必要となります。

- 使用しないポート上で待機している、オペレーティングシステムが提供するすべてのサービスを停止します。
- 可能な場合は、`telnet` を `sshd` に置き換えます。
- パケットフィルタで内部発信元 IP アドレスを持つ外部パケットを拒否し、その背後にアプリケーションサーバーを配置します。パケットフィルタは、明示的に指定したポート以外に向けたすべての外部接続を遮断します。

メッセージングセキュリティー

Messaging Server には、次のセキュリティー機能があります。

- 配備におけるメッセージングコンポーネントの保護

  このオプションセットにより、MTA リレー、メッセージストア、Messenger Express メールクライアント、および多重化サービスのセキュリティーが確保されます。さらに、サードパーティーのスパムフィルタオプションについてもわかります。

- ユーザー認証の計画

  これらのオプションを使用して、メールサーバーでユーザーが認証される仕組みを決定し、権限を持たないユーザーがシステムにアクセスするのを防ぐことができます。

- セキュリティーに関する誤解

  このオプションセットを使用すると、認証された SMTP とデジタル署名の証明書、暗号、SSL (Secure Sockets Layer) によるユーザー認証とメッセージの保護を行うことができます。

詳細については、第 13 章を参照してください。

アプリケーションのセキュリティー

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

- 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

セキュリティー保護された接続の実装

Communications Services は、SSL/TLS、S/MIME 、SAML などのセキュリティー標準をサポートします。SSL/TLS を使えば、クライアントとサーバー間のすべての通信を暗号化されたセッション内で行えます。Portal Server と統合すれば、追加の認証メカニズムがデフォルトで利用可能になるほか、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できるようになります。

注 – Web サーバー内の Communications Express アプリケーションと Calendar Server の `cshttpd` デーモン間では、SSL はサポートされていません。

公開鍵データセキュリティーを実装する場合は、公開鍵インフラストラクチャー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。Communications Express Mail クライアント上では、S/MIME (Secure/Multipurpose Internet Mail Extension) を利用できます。S/MIME を使用するように設定された Communications Express Mail ユーザーは、Communications Express Mail、Microsoft Outlook Express、および Mozilla メールシステムを使用するほかのユーザーと、署名または暗号化されたメッセージを交換できます。

注 – 以前のバージョンの Messaging Server に含まれる Web メールクライアントは、暗号化されたメッセージを生成したり復号化したりできません。

一般的に使用されるデータセキュリティーのメカニズムは、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。

クライアントからサーバーまで SSL を使用することにより、どんな問題が解決するでしょうか。組織は、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されています。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ー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どんな問題が解決されないか。まず最初に、この方法では、組織の内部ネットワークにアクセスした受信者以外のユーザーが偶然にもデータを参照してしまうことを防げません。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ー保護されていない状態で公衆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、使用する各私設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ー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公衆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、公衆インターネットの使用はコストが少なくて済みます。

2 つの異なる認証局 (CA) を使用するセキュリティー保護された接続の実装

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本物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本物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。

セキュリティーに関する誤解

この節では、配備のセキュリティーニーズに対して逆効果になる、典型的な誤解について説明します。

- 製品名とバージョンを隠す:

  製品名とバージョンを隠しても、即席の攻撃者の邪魔をする程度でしかありません。最悪の場合は、管理者にセキュリティーに関する誤った感覚を与えることになり、本当のセキュリティー問題の追跡を怠るという結果になりかねません。

  事実、製品情報とバージョン番号が削除されると、ソフトウェアの識別ができなくなるため、ベンダーのサポート部門がソフトウェアの問題を検証するのが困難となります。

  ハッカーが選択的な行動を取ることはほとんどありません。特に、SMTP サーバーに既知の脆弱性があれば、彼らはあらゆる SMTP サーバーにアクセスを試みるかもしれません。

- 内部マシン名を隠す:

  内部 IP アドレスとマシン名を隠すことで、次のことが困難になります。

  - 悪用またはスパムの追跡
  - メールシステムの設定エラーの診断
  - DNS 設定エラーの診断

  知識のある攻撃者であれば、一度ネットワークに侵入する方法を見つければ、マシンのマシン名と IP アドレスを簡単に見つけ出します。

- **SMTP** サーバーの **EHLO** をオフにする:

  EHLO がない場合、次のことができなくなります。

  - NOTARY

- TLS ネゴシエーション
- メッセージサイズのプリエンプティブ制御

EHLO を使用すると、SMTP クライアントは、制限の有無と、この応答を受けるとすぐに制限を超えたメッセージの送信を停止するかどうかを判断します。ただし、EHLO がオフになっているため HELO を使用しなければならない場合は、送信側の SMTP サーバーはメッセージデータ全体を送信し、その後メッセージサイズが制限を超えているため拒否されたことを通知されます。その結果、処理サイクルとディスク容量の無駄が発生します。

- **Network Address Translation (NAT)**

  NAT を一種のファイアウォールとして使用する場合は、システム間でエンドツーエンドの接続を行うことはできません。その代わり、中間に第三のノードを置くことになります。この NAT システムは仲介役として機能し、潜在的なセキュリティーホールの原因になります。

その他のセキュリティーリソース

セキュリティー保護された Communications Services 配備の設計について、詳しくは Computer Emergency Response Team (CERT) Coordination Center のサイトを参照してください。

`http://www.cert.org`

第 8 章

スキーマとプロビジョニングのオプションについて

この章では、Communications Services のスキーマおよびプロビジョニングのオプションについて説明しています。Communications Services のプロビジョニングは複雑なので、製品をインストールする前にオプションについて理解しておく必要があります。

この章には、次の節があります。

スキーマの選択について

この節では、Communications Services で使用可能なサポートされているスキーマオプションと、どちらを使用するかを決定する方法について説明しています。

Messaging Server スキーマの選択について

Messaging Server では、2 つのスキーマオプションが使用可能で、サポートされています。Sun Java System LDAP スキーマバージョン 1 と Sun Java System LDAP スキーマバージョン 2 の 2 つです。

注 – Sun Java System LDAP スキーマバージョン 1 からSun Java System LDAP スキーマバージョン 2 への移行方法については、『Sun Java System Communications Services 6 2005Q4 Schema Migration Guide』の「`commdirmig` command」を参照してください。

スキーマ 1 のインストールとプロビジョニングのサポートは非推奨になり、今後のリリースからは削除される予定です。ただし、独自のプロビジョニングツールを持つ顧客は、LDAP スキーマ 1 を引き続き使用できます。

Messaging Server で使用するスキーマの選択

プロビジョニングのニーズにより、Messaging Server のインストールにふさわしいスキーマを選択します。

- Portal Server や Access Manager など、シングルサインオン機能を提供するほかの Java Enterprise System コンポーネント製品と Messaging Server を統合しますか。

  答えが「はい」の場合は、スキーマ 2 を使用する必要があります。
- Messaging Server をはじめてインストールしますか、それとも古いバージョンからのアップグレードですか。

  Messaging Server をはじめてインストールする場合は、スキーマ 2 を使用します。

  Messaging Server の古いバージョンからのアップグレードの場合は、スキーマ 1 または 2 のどちらも使用できます。

LDAP スキーマ 1 と Messaging Server

LDAP スキーマ 1 は、組織ツリーと DC ツリーの両方で構成されるプロビジョニングスキーマです。当時は単に「スキーマ」と呼ばれた、このスキーマのセットは、以前の Messaging Server 5.x バージョンでサポートされていました。

スキーマ 1 では、Messaging Server がユーザーエントリまたはグループエントリを検索するときは、DC ツリーのユーザーまたはグループのドメインノードを見て、`inetDomainBaseDN` 属性の値を抽出します。この属性には、実際のユーザーまたはグループエントリの組織サブツリーへの DN 参照があります。

以前のバージョンの Messaging Server がインストールされているサイトでのみ、スキーマ 1 を使用します。

注 – 将来 Messaging Server にその他の Sun Java System 製品を統合してインストールする場合は、スキーマ 2 への移行が必須となります。

LDAP スキーマ *1* と *Messaging Server* でサポートされているプロビジョニングツール

スキーマ 1 は、Sun™ ONE Delegated Administrator for Messaging (旧称 iPlanet Delegated Administrator) と LDAP プロビジョニングツールをサポートしています。詳細については、111 ページの「プロビジョニングツールについて」を参照してください。

LDAP スキーマ 2 (ネイティブモード) と Messaging Server

LDAP スキーマ 2 は一連のプロビジョニング定義で、Directory Server LDAP を使用してエントリとして格納できる情報のタイプを定義しています。

ネイティブモードでは、検索テンプレートを使用して LDAP Directory サーバーを検索します。ドメイン検索テンプレートによりドメインが検索されると、次にユーザーまたはグループ検索テンプレートにより、特定のユーザーまたはグループが検索されます。

Communications Services をはじめてインストールし、2 ツリープロビジョニングモデルに依存するその他のアプリケーションをマシンにインストールしていない場合は、ネイティブモードの使用をお勧めします。Java Enterprise System 製品群にその他の製品をインストールする場合も、このモードを使用するとよいでしょう。

スキーマ 1 を使用する既存の Communications Services 5.x があり、Communications Services をほかの Java Enterprise Server 製品と統合する場合は、Communications Services 6 にアップグレードしたあとでディレクトリをスキーマ 2 に移行させることをお勧めします。LDAP スキーマバージョン 1 から LDAP スキーマバージョン 2 への移行方法の詳細については、『Sun Java System Communications Services 6 2005Q4 Schema Migration Guide』を参照してください。

注 – Java Enterprise System 製品群のすべての Sun Java System 製品で、スキーマ 2 ネイティブモードをプロビジョニングモデルとして使用するようお勧めします。

LDAP スキーマ *2* と *Messaging Server* でサポートされているプロビジョニングツール

スキーマ 2 は、Sun Java System Communications Services Delegated Administrator をサポートしています。詳細については、111 ページの「プロビジョニングツールについて」を参照してください。

LDAP スキーマ 2 互換モードと Messaging Server

スキーマ 2 互換モードは、スキーマ 1 とスキーマ 2 ネイティブモードとの中間のモードです。スキーマ 2 互換モードは両方のスキーマをサポートしており、すでに保有している既存の 2 つのツリー設計を維持できます。また、スキーマ 2 互換モードは、Messaging Server をインストールする前に Access Manager をインストールしていることが前提となっています。

スキーマ 1 を必要とする既存のアプリケーションがあるが、Access Manager やシングルサインオン機能などのように、スキーマ 2 を要求する機能も必要な場合に、スキーマ 2 互換モードを使用します。

注 – スキーマ 2 互換モードは、スキーマ 2 ネイティブモードへの移行の便宜を提供するためのものです。最終的なスキーマ選択では、スキーマ 2 互換モードを使用しないでください。スキーマ 1 からスキーマ 2 互換モードへ移行してから最終的にスキーマ 2 ネイティブモードへと移行するプロセスは、スキーマ 1 からスキーマ 2 ネイティブモードへの単純な移行よりも複雑です。詳細については、『Sun Java System Communications Services 6 2005Q4 Schema Migration Guide』を参照してください。

Calendar Server スキーマの選択について

Calendar Server では、2 つのスキーマオプションが使用可能で、サポートされています。Sun Java System LDAP スキーマバージョン 1 と Sun Java System LDAP スキーマバージョン 2 の 2 つです。

注 – Sun Java System LDAP スキーマバージョン 1 からSun Java System LDAP スキーマバージョン 2 への移行方法については、『Sun Java System Communications Services 6 2005Q4 Schema Migration Guide』を参照してください。

スキーマ 1 のインストールとプロビジョニングのサポートは非推奨になり、今後のリリースからは削除される予定です。ただし、独自のプロビジョニングツールを持つ顧客は、LDAP スキーマ 1 を引き続き使用できます。

Calendar Server で使用するスキーマの選択

プロビジョニングのニーズに基づき、Calendar Server のインストールにふさわしいスキーマを選択します。

- Portal Server や Access Manager など、シングルサインオン機能を提供するほかの Java Enterprise System コンポーネント製品と Calendar Server を統合しますか。

  答えが「はい」の場合、スキーマ 2 ネイティブモードを使用する必要があります。
- はじめて Calendar Server をインストールするのですか、それとも旧バージョンからアップグレードするのですか。

Calendar をはじめてインストールする場合はスキーマ 2 ネイティブモードを使用します。

Calendar Server の旧バージョンからアップグレードする場合は、スキーマ 1 またはスキーマ 2 のネイティブモードまたは互換モードのいずれでも使用することができます。

- プロビジョニングまたはシングルサインオンのいずれかに、Access Manager CLI ユーティリティーの使用を計画していますか。

  答えが「はい」の場合、スキーマ 2 のネイティブモードまたは互換モードを使用します。

- Calendar Server `csdomain` ユーティリティーを使用してドメインをプロビジョニングする予定ですか。

  答えが「はい」の場合、スキーマ 2 のネイティブモードまたは互換モードを使用します。`csdomain` ユーティリティーを使用する予定がなく、Calendar Server がすでにインストールされている場合は、スキーマ 1 を使用します。

- プロビジョニングに Access Manager や Calendar Server CLI のいずれのユーティリティーも使用しない場合、新規のインストールにはスキーマ 2 ネイティブモード、既存の Calendar Server のインストールにはスキーマ 1 またはスキーマ 2 互換モードのいずれかが使用できます。

LDAP スキーマ 1 と Calendar Server

LDAP スキーマ 1 は、組織ツリーと DC ツリーの両方で構成されるプロビジョニングスキーマです。当時は単に「スキーマ」と呼ばれた、このスキーマのセットは、以前の Calendar Server 5.x バージョンでサポートされていました。

Calendar Server はユーザーやグループのエントリを検索する場合、DC ツリーのユーザーまたはグループのドメインノードを調べ、`inetDomainBaseDN` 属性の値を抽出します。この属性には、実際のユーザーまたはグループエントリの組織サブツリーへの DN 参照があります。

Calendar Server の旧バージョンをインストール済みのサイトだけが、スキーマ 1 を使用する必要があります。

注 – 将来、ほかの Sun Java System 製品とともに Calendar Server をインストールすることを計画している場合、スキーマ 2 への移行は必須です。

LDAP スキーマ *1* と *Calendar Server* でサポートされているプロビジョニングツール

スキーマ 1 は LDAP プロビジョニングツールをサポートしています。詳細については、111 ページの「プロビジョニングツールについて」を参照してください。

LDAP スキーマ 2 (ネイティブモード) と Calendar Server

スキーマ 2 は一連のプロビジョニング定義で、Directory Server LDAP を使用してエントリとして格納できる情報のタイプを定義しています。

ネイティブモードでは、検索テンプレートを使用して LDAP Directory サーバーを検索します。ドメイン検索テンプレートによりドメインが検索されると、次にユーザーまたはグループ検索テンプレートにより、特定のユーザーまたはグループが検索されます。

Communications Services をはじめてインストールし、2 ツリープロビジョニングモデルに依存するその他のアプリケーションをマシンにインストールしていない場合は、ネイティブモードの使用をお勧めします。Java Enterprise System 製品群にその他の製品をインストールする場合も、このモードを使用するとよいでしょう。

スキーマ 1 を使用する既存の Communications Services 5.x があり、Communications Services をほかの Java Enterprise Server 製品と統合したい場合は、Communications Services 6 に移行したあとで、ディレクトリをスキーマ 2 に移行させることをお勧めします。LDAP スキーマバージョン 1 から LDAP スキーマバージョン 2 への移行方法の詳細については、『Sun Java System Communications Services 6 2005Q4 Schema Migration Guide』を参照してください。

注 – Java Enterprise System 製品群のすべての Sun Java System 製品で、スキーマ 2 ネイティブモードをプロビジョニングモデルとして使用するようお勧めします。

LDAP スキーマ *2* と *Calendar Server* でサポートされているプロビジョニングツール

スキーマ 2 は、Sun Java System Communications Services Delegated Administrator をサポートしています。詳細については、111 ページの「プロビジョニングツールについて」を参照してください。

LDAP スキーマ 2 互換モードと Calendar Server

スキーマ 2 互換モードは、スキーマ 1 とスキーマ 2 ネイティブモードとの中間のモードです。スキーマ 2 互換モードは両方のスキーマをサポートしており、すでに保有している既存の 2 つのツリー設計を維持できます。また、スキーマ 2 互換モードは、Messaging Server をインストールする前に Access Manager をインストールしていることが前提となっています。

スキーマ 1 を必要とする既存のアプリケーションがあるが、Access Manager やシングルサインオン機能などのように、スキーマ 2 を要求する機能も必要な場合に、スキーマ 2 互換モードを使用します。

注 – スキーマ 2 互換モードは、スキーマ 2 ネイティブモードへの移行の便宜を提供するためのものです。最終的なスキーマ選択では、スキーマ 2 互換モードを使用しないでください。スキーマ 1 からスキーマ 2 互換モードへ移行してから最終的にスキーマ 2 ネイティブモードへと移行するプロセスは、スキーマ 1 からスキーマ 2 ネイティブモードへの単純な移行よりも複雑です。詳細については、『Sun Java System Communications Services 6 2005Q4 Schema Migration Guide』を参照してください。

プロビジョニングツールについて

この節では、サポートされているプロビジョニングツールについて説明します。これらのツールを使って、LDAP ディレクトリ内のユーザー、グループ、およびドメインのエントリ情報の問い合わせ、変更、追加、または削除を行うことができます。

Messaging Server プロビジョニングツールの理解

サポートされている Messaging Server プロビジョニングツールを使って、LDAP ディレクトリ内のユーザー、グループ、およびドメインのエントリ情報の問い合わせ、変更、追加、または削除を行うことができます。この節では、これらの Messaging Server プロビジョニングツールを検証します。

106 ページの「Messaging Server で使用するスキーマの選択」にある質問の他に、表 8–1 を使用して、スキーマとプロビジョニングツールオプションを評価します。

注 – Messaging Server のインストールと設定を行う前に、Messaging Server エントリのプロビジョニングのためのスキーマモデルとツールを決定する必要があります。

次の節で、サポートされているプロビジョニングツールに関する高度な情報について説明します。

Sun ONE Delegated Administrator for Messaging

Sun ONE Delegated Administrator for Messaging (旧称 iPlanet Delegated Administrator) には、ユーザーおよびグループのプロビジョニングを行うためのコマンド行ユーザーインタフェースとグラフィカルユーザーインタフェースがあります。Delegated Administrator は、プロビジョニング定義の Messaging Server 5.x バージョンである Sun LDAP スキーマ 1 を使用します。

Messaging Server 用 LDAP プロビジョニングツール

スキーマ 1 のユーザーとグループに関しては、LDAP Directory ツール (スキーマ 2 はサポートされていない) を使用してプロビジョニングを行います。Delegated Administrator のグラフィカルおよびコマンド行インタフェースとは異なり、ユーザーインタフェースを使用せずに、LDAP を通じて LDIF レコードの追加、削除、変更を行うことで、ダイレクトにユーザーとグループのプロビジョニングを行います。

Delegated Administrator と Messaging Server

Access Manager は スキーマ 2 を使用します。Java Enterprise System 製品群に含まれる Sun Java System コンポーネント製品がスキーマ 2 を使用するため、Communications Services 6 Delegated Administrator を使用します。Java Enterprise System 製品を複数使用する場合や Calendar Server の新規インストールを実行する場合には、特にそのようにする必要があります。

インストールの詳細については、『Sun Java System Communications Services 6 2005Q4 Delegated Administrator 管理ガイド』を参照してください。

Messaging Server プロビジョニングツールオプションの比較

表 8–1 に、サポートされているさまざまなスキーマ、プロビジョニングツール、プロビジョニングの制限、および詳細情報についての推奨マニュアルを示します。

表 **8–1** Messaging Server のプロビジョニングメカニズム

サポートされているプロビジョニングツール	プロビジョニングツールの機能	プロビジョニングツールの制限	詳細情報
Sun ONE Delegated Administrator for Messaging グラフィカルユーザーインタフェース 使用スキーマ: スキーマ 1	ユーザー、グループ、ドメイン、およびメーリングリストの管理者のためのグラフィカルユーザーインタフェースを提供します。エンドユーザーは不在メッセージと Sieve フィルタを管理できます。	■ Messaging Server 6 にアップグレードしている既存の Messaging Server 5.x の顧客だけが使用可能です。 ■ Sun ONE Web Server 6.0 (Messaging Server 5.2 バンドルとしてのみ入手可能) でのみ使用可能です。Sun ONE Web Server 6.1 では使用できません。 ■ Sun スキーマ 2 およびほかの Java Enterprise System 製品との互換性がありません。 ■ Sun Java System Messenger Express 経由でメールフィルタを使用できません。Delegated Administrator 経由でフィルタを使用する必要があります。 ■ Messaging Server 5.2 製品でのみ使用可能なオートリレーチャネルを使用する必要があります。	Sun ONE Delegated Administrator for Messaging 1.3 のマニュアルを参照してください。 Sun ONE Delegated Administrator インタフェースのインストールと管理方法を説明しています。

表 **8–1** Messaging Server のプロビジョニングメカニズム　(続き)

サポートされているプロビジョニングツール	プロビジョニングツールの機能	プロビジョニングツールの制限	詳細情報
Sun ONE Delegated Administrator for Messaging コマンド行インタフェース 使用スキーマ: スキーマ 1	ユーザー、グループ、ドメイン、およびメーリングリストの管理者のためのコマンド行インタフェースを提供します。	■ Sun スキーマ 2 およびほかの Java Enterprise System 製品との互換性がありません。	Sun ONE Delegated Administrator for Messaging 1.3 のマニュアルを参照してください。 Sun ONE Delegated Administrator コマンド行ユーティリティーの構文と使用方法を解説しています。
LDAP プロビジョニングツール 使用スキーマ: スキーマ 1	LDAP エントリを直接変更するツールまたはカスタムプロビジョニングツールを作成するツールを提供します。	■ Sun スキーマ 2 およびほかの Java Enterprise System 製品との互換性がありません。	『iPlanet Messaging Server 5.2 Provisioning Guide』および『iPlanet Messaging and Collaboration Schema Reference』を参照してください。 Sun LDAP スキーマ 1 プロビジョニングモデルについて説明しています。 さらに、LDAP プロビジョニングツールと特定の属性およびオブジェクトクラスの使用法についても説明しています。
Sun Java System Console 使用スキーマ: スキーマ 1	Sun Java System Console にプロビジョニング機能が含まれていますが、Messaging ユーザーとグループのプロビジョニングには推奨しません。代わりに、割り当て、ログファイル、その他の関連するメッセージストア項目などのサーバー設定の管理に Sun Java System Console を使用してください。	■ Sun スキーマ 2 およびほかの Java Enterprise System 製品との互換性がありません。 ■ Console ではユーザーとグループを適切に追加したり変更したりできないため、プロビジョニングツールとしては推奨しません。	『Sun Java System Messaging Server 6 2005Q4 管理ガイド』および対応する Sun Java System Console オンラインヘルプを参照してください。

表 8–1 Messaging Server のプロビジョニングメカニズム　(続き)

サポートされているプロビジョニングツール	プロビジョニングツールの機能	プロビジョニングツールの制限	詳細情報
Delegated Administrator 使用スキーマ: スキーマ 2	ユーザー、グループ、ドメイン、およびメーリングリストの管理者のためのグラフィカルインタフェースとコマンド行インタフェースを提供します。 ほかの Java Enterprise System 製品と互換性があります。	■ Sun スキーマ 1 との下位互換性がありません。 ■ Sun Java System Access Manager では GUI プロビジョニングツールを使用できません。 ■ Sun Java System Access Manager をインストールしてこのインタフェースを有効にする必要があります。	『Sun Java System Communications Services 6 2005Q4 Delegated Administrator 管理ガイド』を参照してください。 コマンド行ユーティリティーの構文と使用法を解説しています。

Calendar Server プロビジョニングツールについて

サポートされている Calendar Server プロビジョニングツールを使って、LDAP ディレクトリ内のユーザー、グループ、およびドメインのエントリ情報の問い合わせ、変更、追加、または削除を行うことができます。この節では、これらの Calendar Server プロビジョニングツールについて説明します。

108 ページの「Calendar Server で使用するスキーマの選択」にある質問のほかに、表 8–2 を使用して、スキーマとプロビジョニングツールオプションを評価します。

注 – Calendar Server のインストールおよび設定に先立って、Calendar Server エントリをプロビジョニングするためのスキーマおよびツールを決定する必要があります。

次の節で、サポートされているプロビジョニングツールに関する高度な情報について説明します。

- 112 ページの「Messaging Server 用 LDAP プロビジョニングツール」
- 116 ページの「Delegated Administrator と Calendar Server」
- 112 ページの「Messaging Server プロビジョニングツールオプションの比較」

Calendar Server 用 LDAP プロビジョニングツール

スキーマ 1 のユーザーとグループに関しては、LDAP Directory ツール (スキーマ 2 はサポートされていない) を使用してプロビジョニングを行います。ユーザーインタフェースを使用せずに LDAP を通じて LDIF レコードの追加、削除、変更を行うことで、ユーザーとグループのプロビジョニングを直接行えます。

Delegated Administrator と Calendar Server

Access Manager はスキーマ 2 を使用します。Java Enterprise System 製品群に含まれる Sun Java System コンポーネント製品がスキーマ 2 を使用するため、Communications Services 6 Delegated Administrator ユーティリティーを使用します。Java Enterprise System 製品を複数使用する場合や Calendar Server の新規インストールを実行する場合には、特にそのようにする必要があります。

インストール方法については、『Sun Java System Communications Services 6 2005Q4 Delegated Administrator 管理ガイド』を参照してください。

Calendar Server プロビジョニングツールオプションの比較

次の表に、サポートされているさまざまなスキーマ、プロビジョニングツール、プロビジョニングの制限、および詳細情報についての推奨マニュアルを示します。

表 **8–2** Calendar Server のプロビジョニングメカニズム

サポートされているプロビジョニングツール	プロビジョニングツールの機能	プロビジョニングツールの制限	詳細情報
LDAP プロビジョニングツール 使用スキーマ: スキーマ 1	LDAP エントリを直接変更するツールまたはカスタムプロビジョニングツールを作成するツールを提供します。	Sun スキーマ 2 およびほかの Java Enterprise System 製品との互換性がありません。	『iPlanet Messaging Server 5.2 Provisioning Guide』および『iPlanet Messaging and Collaboration Schema Reference』を参照してください。 Sun LDAP スキーマ 1 プロビジョニングモデルについて説明しています。 さらに、LDAP プロビジョニングツールと特定の属性およびオブジェクトクラスの使用法についても説明しています。
Delegated Administrator 使用スキーマ: スキーマ 2	ユーザー、グループ、ドメイン、およびリソースを管理する管理者のためのグラフィカルインタフェースとコマンド行インタフェースを提供します。 ほかの Java Enterprise System 製品と互換性があります。	■ Sun スキーマ 1 との下位互換性がありません。 ■ Sun Java System Access Manager をインストールしてこのインタフェースを有効にする必要があります。	『Sun Java System Communications Services 6 2005Q4 Delegated Administrator 管理ガイド』を参照してください。 コマンド行ユーティリティーの構文と使用法を解説しています。

パート II Messaging Server の配備

この部には、次の章があります。

第 9 章

Messaging Server ソフトウェアの紹介

Sun Java System Messaging Server は、企業とサービスプロバイダの両方で要求される大容量で信頼性の高いメッセージング処理のために設計された、強力なインターネットメッセージングサーバーです。サーバーはモジュール化された、個別に構成可能な複数のコンポーネントから成り立っています。これらのコンポーネントは、さまざまな電子メールプロトコルをサポートしています。

Messaging Server は、ユーザー、グループ、およびドメインについての情報を格納するために一元化された LDAP データベースを使用します。サーバー設定についてのいくつかの情報は LDAP データベースに格納されます。また、ローカル設定ファイルに格納される情報もあります。

Messaging Server 製品群には、ユーザーのプロビジョニングやサーバーの構成をサポートするツールが含まれています。

この章には、次の節があります。

メッセージングシステムとは

すぐれた電子メールシステムアーキテクチャーでは、埋め込まれたサウンド、画像、ビデオファイル、HTML 形式とともに電子メールが迅速に配信され、将来のアップグレードへの対応とスケーラビリティを提供します。単純化すると、Messaging Server アーキテクチャーは次の機能を備える必要があります。

- 外部サイトからのメールを受信する
- これらのメッセージが配信されるユーザーメールボックスを判断し、そこにルーティングする

- 内部ホストからのメールを受信する
- これらのメッセージの配信先となるシステムを判断し、そこにルーティングする

電子メールシステムアーキテクチャーの中心はメッセージングサーバー自体で、これはメッセージの送信と配信に使用されるコンポーネントの集合体です。Messaging Server で提供されるコンポーネントとは別に、電子メールシステムでは LDAP サーバーと DNS サーバーも必要となります。DNS サーバーは、電子メールシステムを配備する前に配置しておく必要があります。

効率性とスケーラビリティ以外にも、いくつかの要素が Messaging Server アーキテクチャーに影響を与えます。これらの要素を次に示します。

- 負荷分散
- ファイアウォール
- 高可用性

これらのトピックの詳細については、第 11 章を参照してください。

Messaging Server がサポートする標準と機能

この節では、Messaging Server がサポートする標準について説明するほか、Messaging Server がサポートするその他の機能についても説明します。

標準プロトコルのサポート

Messaging Server は、電子メッセージングに関連するほとんどの国内規格、国際規格、および業界規格をサポートしています。完全なリストは、『Sun Java System Messaging Server 6 2005Q4 Administration Reference』の付録 A「Supported Standards」を参照してください。

ホストされているドメインのサポート

Messaging Server は、ISP にアウトソースされた電子メールドメインのようなホストされているドメインを完全にサポートしています。つまり、ISP は組織の電子メールサービスをリモートで操作および管理することにより組織をホスティングする電子メールドメインを提供します。ホストしているドメインは、ほかのホストしているドメインと同じ Messaging Server ホストを共有することができます。初期の LDAP ベースの電子メールシステムでは、1 つのドメインが 1 つまたは複数の電子メールサーバーホストによってサポートされていました。Messaging Server では、複数のド

メインを単一のサーバーでホストできます。ホストされている各ドメインには、そのドメインのユーザーとグループのコンテナを指し、さまざまなドメイン固有のデフォルト設定を提供する LDAP エントリがあります。

ドメインを定義する場合、そのドメインに対応するドメインエントリがディレクトリ内に存在する必要があります。つまり、そのドメインに対する LDAP エントリを作成する必要があります。`mailAlternateAddress` や `mailEquivalentAddress` などの属性は、ディレクトリ内のドメインエントリの存在に依存します。これは、バニティドメインの場合とは対照的です。バニティドメインは、特定のサーバーやホストされたドメインに関連付けられるのではなく、特定のユーザーに関連付けられたドメイン名です。バニティドメインの場合、そのドメイン名に対する LDAP エントリは存在しません。

注 – バニティドメインを使用すると処理時のオーバーヘッドが増大します。したがって、その使用はお勧めできません。

ユーザーのプロビジョニングのサポート

Messaging Server は、ユーザー、グループ、およびドメインについての情報を格納するために一元化された LDAP データベースを使用します。現在、Messaging Server は Sun Java System LDAP スキーマバージョン 1 (スキーマ 1) と Sun Java System LDAP スキーマバージョン 2 (スキーマ 2) の 2 つのスキーマオプションをサポートしています。プロビジョニングオプションは、選択されたスキーマにより異なります。詳細については、第 15 章を参照してください。

スキーマ 2 の Messaging Server プロビジョニングは、Delegated Administrator を使って行います。これについては、『Sun Java System Communications Services 6 2005Q4 Delegated Administrator 管理ガイド』を参照してください。

スキーマ 1 は、メッセージング用 iPlanet Delegated Administrator 製品によってサポートされています。この製品には、組織内のユーザー、グループ、およびドメインを管理するために、グラフィカルユーザーインタフェースとコマンド行ユーティリティーが用意されています。スキーマ 1 におけるユーザー、グループ、およびドメイン管理については、以前のリリースのソフトウェアに関する次のマニュアルを参照することもできます。

- 『iPlanet Messaging Server 5.2 Provisioning Guide』 - LDAP を使ってドメイン、ユーザー、グループ、または管理者のエントリを作成する方法を説明しています。
- 『iPlanet Messaging and Collaboration Schema Reference』 - Communications Services のスキーマ 1 について説明しています。
- 『iPlanet Messaging Server 5.2 Reference Manual』 - ユーザー、グループ、およびドメインを管理するための iPlanet Delegated Administrator コマンド行ユーティリティーについて説明しています。
- iPlanet Delegated Administrator オンラインヘルプ

注 – Access Manager コンソールは、Messaging Server と Calendar Server の LDAP ユーザーエントリに対し、Access Manager サービスによる最小限のプロビジョニング機能を提供します。インタフェースには入力を確認する機能がないため、電子メールを受け取ることができないユーザーエントリや動作しないユーザーエントリが、エラーが報告されることなく作成されてしまいます。そのため、このインタフェースはデモの目的でだけ使用します。

『Sun Java System Communications Services 6 2005Q4 Delegated Administrator 管理ガイド』で説明している Delegated Administrator は、Communications Services ユーザーをプロビジョニングするための推奨メカニズムです。

統一されたメッセージングのサポート

Messaging Server は完全な、統一されたメッセージングソリューションの基盤となります。統一されたメッセージングとは、電子メール、ボイスメール、FAX、ビデオ、およびそのほかの通信形態に関して単一のメッセージストアを使用するという概念です。

Web メールのサポート

Messaging Server は現在、次の 2 つのクライアント向けユーザーインタフェース (UI) をサポートしています。

- Messenger Express
- Communications Express

今後、Messenger Express ユーザーインタフェースに新機能が追加されることはありません。Messaging Server は非推奨となり、代わって Communications Express が推奨のユーザーインタフェースとなりました。Sun Microsystems, Inc. は後日、Messenger Express の生産中止スケジュールを発表する予定です。

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

`http://docs.sun.com/app/docs/coll/1312.1`

Messaging Server のセキュリティとアクセス制御

Messaging Server には、次のセキュリティとアクセス制御の機能があります。

- POP、IMAP、HTTP、または SMTP へのパスワードによるログインおよび証明書に基づくログインのサポート
- 標準セキュリティプロトコルのサポート: TLS (Transport Layer Security)、SSL (Secure Sockets Layer)、および SASL (Simple Authentication and Security Layer)

- Delegated Administrator
- POP、IMAP、SMTP および HTTP へのクライアント IP アドレスアクセスのフィルタ
- システム全体、ユーザーごと、およびサーバー側の Sieve 規則による、大量な迷惑メールのフィルタリング

Messaging Server の管理ユーザーインタフェース

Messaging Server はモジュール化された、個別に構成可能な複数のコンポーネントから成ります。これらのコンポーネントは、電子メールの転送とアクセスプロトコルをサポートしています。

メッセージ転送エージェント (MTA) を設定するために、Messaging Server には、サーバー上にローカルに格納されたコマンド行ユーティリティーと設定ファイルの完全なセットが用意されています。また、メッセージストアおよびメッセージアクセスサービスを設定するために、コンソールグラフィカルユーザーインタフェースとコマンド行ユーティリティーの完全なセットが用意されています。

詳細については、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』を参照してください。

Messaging Server のソフトウェアアーキテクチャー

図 9–1 は、スタンドアロンの Messaging Server を簡略化して示しています。この特別な配備は、スケーラビリティが低いため、大規模配備にはお勧めできませんが、Messaging Server の個々のコンポーネントを示しています。

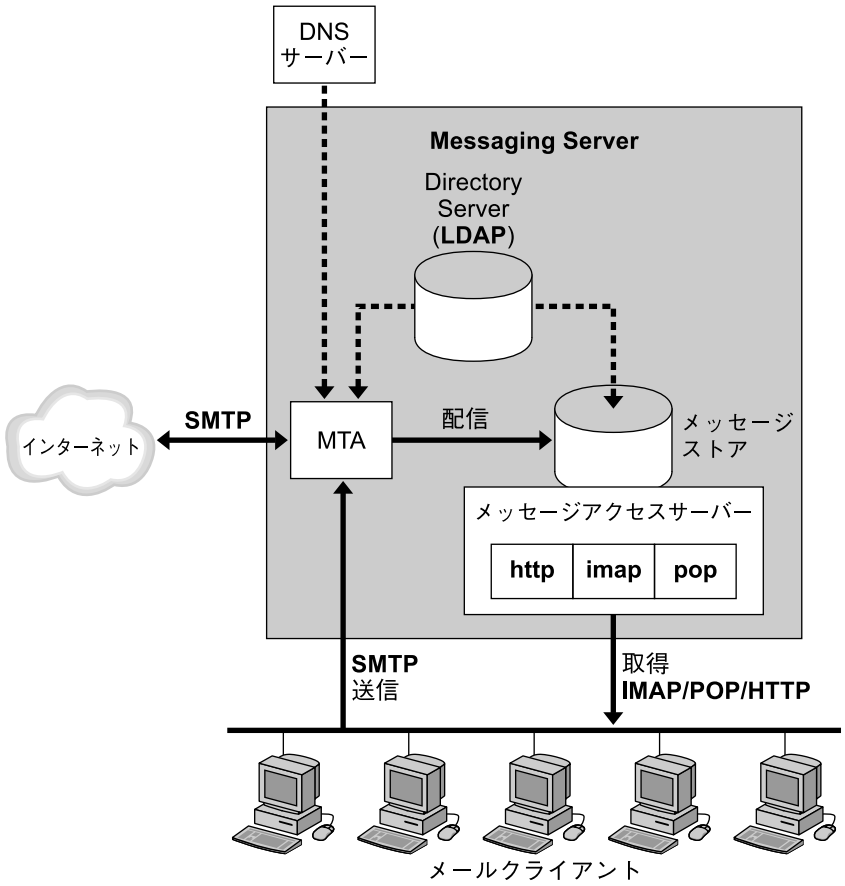

図 **9–1** スタンドアロンの Messaging Server の簡略化したコンポーネント表示

この図は、次の Messaging Server ソフトウェアコンポーネントを示しています。

- メッセージ転送エージェント **(MTA):** SMTP プロトコルを使用して、メールメッセージの受信、ルーティング、転送、および配信を行います。MTA は、ローカルのメールボックスか別の MTA にメッセージを配信します。
- メッセージストア**:** メールクライアントのメッセージの格納、取得、および操作を行う一連のコンポーネントで構成されます。メールは POP クライアント、IMAP クライアント、または HTTP クライアントにより取得されます。POP クライアントは、メッセージをクライアントマシンにダウンロードして、読み取りと保管を行います。IMAP クライアントと HTTP クライアントは、サーバー上のメッセージの読み取りと操作を行います。

- **LDAP** ディレクトリ**:** Messaging Server のメールディレクトリ情報の保管、取得、および配信を行います。これには、ユーザーのルーティング情報、配信リスト、設定データ、および電子メールの配信とアクセスのサポートに必要なその他の情報などがあります。また、MTA またはメッセージストアがユーザー認証時に必要とするパスワードなどの情報も、LDAP ディレクトリ内に格納されます。

  メッセージを格納するだけでなく、メッセージストアはディレクトリサーバーを使用して、メールクライアントがメールにアクセスする場合のユーザーのログイン名とパスワードの検証も行います。ディレクトリには、割り当て制限、デフォルトのメッセージストアタイプなどの情報も格納されます。
- **DNS** サーバー**:** ドメイン名を IP アドレスに変換します。このコンポーネントは Messaging Server をインストールする前に必要となります。

簡略化した Messaging Server システムを通したメッセージパス

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、Simple Mail Transport Protocol (SMTP) を通じて MTA によって受信されます。内部アドレスの場合、すなわち Messaging Server ドメイン内の場合は、MTA はメッセージをメッセージストアに配信します。メッセージが外部宛て、すなわち Messaging Server の制御外のドメイン宛ての場合、MTA はメッセージをインターネット上の別の MTA にリレーします。

UNIX システムの場合に限り、`/var/mail` ファイルシステムにメールを配信することも可能ですが、ローカルメッセージは通常、より最適化された Messaging Server メッセージストアに配信されます。次に、IMAP4、POP3 、または HTTP メールクライアントプログラムがメッセージを取得します。

メールクライアントからの送信メッセージは MTA に直接送られ、そこでインターネット上の適切なサーバーに送信されます。アドレスがローカルの場合は、MTA はメッセージをメッセージストアに送信します。

新しいユーザーとグループは、ディレクトリにユーザーとグループのエントリを追加することで作成されます。Communications Services Delegated Administrator ユーティリティーを使用するか、LDAP を使用してディレクトリを変更することで、エントリを作成または変更することができます。

Messaging Server コンポーネントは、管理サーバーコンソールを使って管理できます。さらに、Messaging Server には一連のコマンド行インタフェースと設定ファイルも用意されています。より一般的な管理タスクとしては、メールシステムへのユーザーやグループの追加、変更、削除や、MTA、ディレクトリサーバー、およびメッセージングストアの操作の設定があります。

メッセージ転送エージェント (MTA)

MTA は、Messaging Server に宛てられたインターネットメールメッセージのルーティング、転送、および配信を行います。メールは、「チャネル」と呼ばれるインタフェース内を通過します。各チャネルは、1 つまたは 1 組のエージェントプログラムと一連の設定情報とで構成されます。エージェントプログラムには、チャネルに入ってきたメールを処理する「スレーブプログラム」と、チャネルを出ていくメールを処理する「マスタープログラム」があります。任意のチャネルに関連付けられた 1 つ以上のインタフェースに送られるメッセージを格納するためのメッセージキューがあります。Messaging Server には、次のような数多くのチャネルがデフォルトで用意されています。

- **SMTP** チャネル**:** TCP/IP ベースのメッセージ配信と受信を有効にします。マスターチャネルとスレーブチャネルが用意されます。
- **LMTP** チャネル**:** 2 層構成における MTA から メッセージストアへのメッセージの直接ルーティングを有効にします。これらのチャネルは、SMTP ではなく LMTP を使用して別のシステム上のメッセージストアと通信を行います。マスターチャネルとスレーブチャネルが用意されます。
- パイプチャネル**:** 代替メッセージ配信プログラムで使用します。メッセージをユーザーの受信箱に直接送るのではなく、メールソーターのようなプログラムへの配信を行います。マスターチャネルが用意されます。
- ローカルチャネル**:**メールを `/var/mail` に配信します。古い UNIX メールクライアントとの互換性を提供します。マスターチャネルが用意されます。
- 再処理チャネル**:** 再送信されたメッセージの処理に役立ちます。マスターチャネルが用意されます。
- 再組立チャネル**:** 不完全なメッセージを再度組み立て、MIME の Message/Partial Content-type をサポートする元の完全なメッセージにします。マスターチャネルが用意されます。
- 変換チャネル**:** メッセージを本文ごとに変換します。アドレスの再書き込みまたはメッセージの再フォーマットに役立ちます。マスターチャネルが用意されます。
- メッセージストアチャネル**:** メッセージストアへのローカル配信を行います。

図 9–2 は、このプロセスを示したものです。チャネルを個別に設定し、アドレスに基づいてメールを特定のチャネルに送ることもできます。

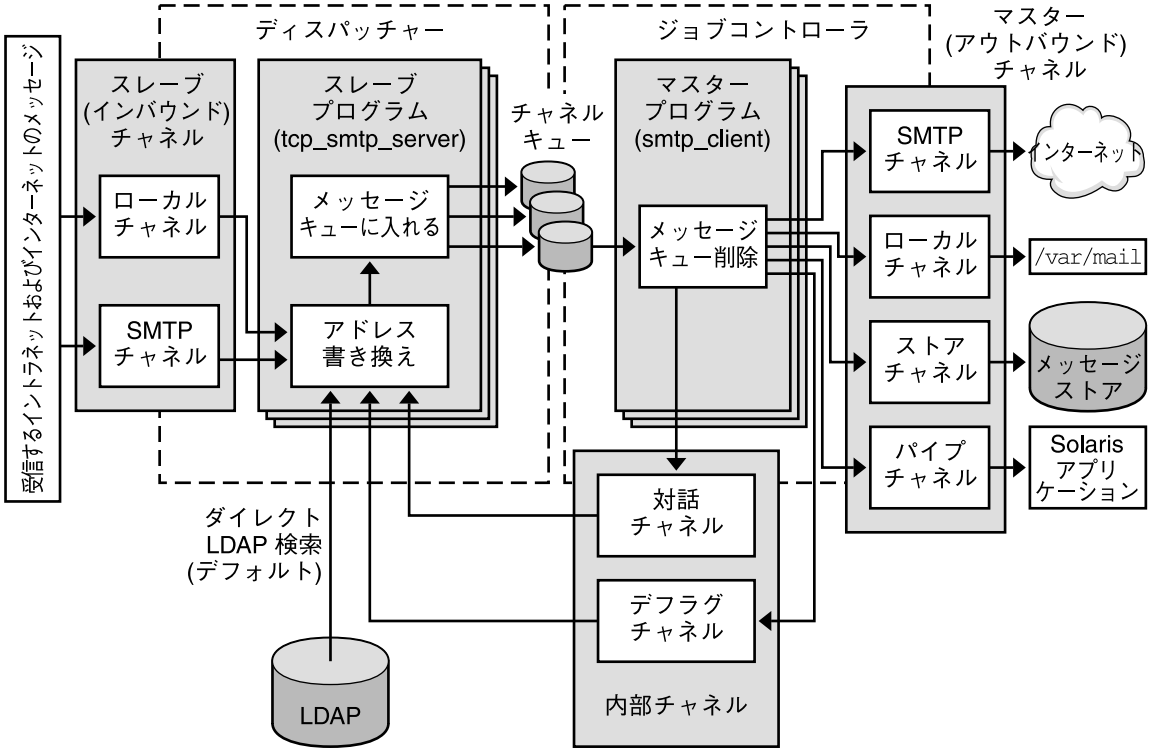

図 **9–2** チャネルアーキテクチャー

チャネルプログラムは、次の 2 つの機能の 1 つを実行します。

- SMTP スレーブプログラムは、メッセージをメッセージキューに入れて MTA による次の処理に備えるか、システムに受け入れることのできないメッセージを拒否し、他のインタフェースからのメッセージを受け入れます。
- マスタープログラムは、キュー領域からのメッセージを処理し、それを同じシステムのキューに入れて、別のチャネルによる処理に備えます。または、他のインタフェースに送信し、送信後にキューから削除します。あるいは、そのメッセージをメッセージストアのようなシステム上の最終送信先に配信します。

チャネルの設定は、`imta.cnf` 設定テキストファイルを使用して行います。チャネル設定を通じて、さまざまな「チャネルキーワード」を設定してメッセージの処理方法を制御できます。チャネルキーワードは、パフォーマンスの調整とシステムのレポーティング面に影響を与えます。たとえば、複数のチャネルを定義してトラフィックを送信先別に分類し、メッセージサイズを制限してトラフィックを制限し、業務のニーズに応じて配信状態通知規則を定義します。診断属性もチャネル単位で設定可能です。かなりの数の設定パラメータが、チャネルベースで設定可能です。

MTA の概念の詳細については、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』の第 8 章「MTA の概念」を参照してください。

LDAP 直接検索

MTA は、情報の検索を LDAP サーバーに対して直接行います。直接検索により、MTA と LDAP サーバーとの関係がスケーラブルで高速かつ設定可能になります。LDAP クエリの結果は設定可能なサイズと時間でプロセスにキャッシュされるため、パフォーマンスの調整が可能です。詳細については、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』を参照してください。

書き換え規則

メールは、「ドメイン書き換え規則」(略して「書き換え規則」) が適用された送信先アドレスの実行結果に基づいて、チャネルにルーティングされます。書き換え規則は、アドレスを真のドメインアドレスに変換し、それに対応するチャネルを決定するために使用されます。これらの規則は、「トランスポート層」と「メッセージヘッダー」の両方に表示されるアドレスを書き換えるために使用されます。トランスポート層は、メッセージのエンベロープです。ルーティング情報はユーザーには見えない形で含まれていますが、実際の情報はメッセージを適切な受信者に配信するのに使用されます。

書き換え規則とチャネルのテーブルは、協力してそれぞれのアドレスの処置を決定します。書き換えプロセスの結果により、アドレスとルーティングシステム、すなわちメッセージが送信またはキューイングされるシステム (チャネル) が書き換えられます。ネットワークのトポロジ次第で、ルーティングシステムはメッセージが送信先までにたどるパスの最初のステップである場合もあれば、最終の送信先システムである場合もあります。

書き換えプロセスが終了すると、`imta.cnf` ファイルのチャネル部分に対してルーティングシステムの検索が行われます。それぞれのチャネルには、チャネルに関連付けられた 1 つ以上のホスト名があります。ルーティングシステム名がそれぞれのホスト名と比較されて、メッセージがどのチャネルのキューに入れられるかが決定されます。次に簡単な書き換え規則を示します。

```
example.com     $U%example.com@tcp_siroe-daemon
```

この規則は、ドメイン `example.com` のアドレスだけを検索します。一致したアドレスは、次に示すテンプレート`$U%$D` を使用して書き換えられます。

`$U`　アドレスのユーザーの部分またはアドレスの左側 (@ の前) を示します

`%`　@ 符号を示します

`$D`　アドレスのドメインの部分またはアドレスの右側 (@ の後ろ) を示します

このように、`wallaby@thor.example.com` の形式のメッセージが `wallaby@example.com` に書き換えられ、`tcp_siroe-daemon` をチャネルホスト名に持つチャネルに送信されます。

書き換え規則は、マッピングテーブル、LDAP ディレクトリ検索、およびデータベース参照に基づいて、高度な置換を行うこともできます。暗号のようなわかりにくいものになる場合もありますが、書き換え規則が低レベルで動作し、処理サイクルへの直

接のオーバーヘッドがほとんどない点が便利です。書き換え規則の詳細と書き換えプロセスで利用できる機能については、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』の第 11 章「書き換えルールの設定」を参照してください。

ジョブコントローラ

マスターチャネルプログラムは、ジョブコントローラの制御下で実行されます。ジョブコントローラは、メッセージキューを制御し、実際のメッセージ配信を行うチャネルプログラムを呼び出すプログラムです。ジョブコントローラはマルチスレッドプロセスであり、Messaging Server システムに常駐している数少ないプロセスの 1 つです。チャネル処理ジョブ自体は、ジョブコントローラにより作成されますが、一時的なジョブで、実行する作業がない場合は存在しなくなります。

ジョブコントローラの設定により、チャネル処理プログラムのインスタンスが常に少なくとも 1 つ存在するかどうかが決まります。多くの場合は、すぐに実行する作業がなくてもサービスプログラムのインスタンスが少なくとも 1 つは常に存在するように設定されます。それ以外の場合は、現在行うべき作業がなくなってから一定期間の間インスタンスが存在することになります。

外部メッセージを受け入れたスレーブチャネルは、メッセージをキューイングすることにより、新しいメッセージファイルが作成されたことをジョブコントローラに通知します。ジョブコントローラは、この情報を内部データ構造に入力し、必要に応じてそのキュー内のメッセージを処理するマスターチャネルジョブを作成します。ジョブコントローラで、既存のチャネルジョブが新しくキューイングされたメッセージファイルを処理できるように設定されている場合は、このジョブを作成する必要はありません。マスターチャネルジョブは、ジョブが開始されると、ジョブコントローラからメッセージ割り当てを取得します。メッセージの処理を終了すると、マスターチャネルはその処理のステータスに応じてジョブコントローラを更新します。そのステータスは、メッセージが正常にキューから削除されたか、メッセージの再配信スケジュールが組まれたかのいずれかになります。

ジョブコントローラは、メッセージの優先度と失敗した配信に関する情報を維持し、チャネルジョブに優先的なスケジュールを許可します。ジョブコントローラは、各ジョブの状態の追跡も行います。ジョブの状態は、アイドル、アイドルの時間、ジョブがビジーであるかどうかです。状態の追跡により、ジョブコントローラはチャネルジョブの最適なプールを維持できます。

注 – 現時点で存在しているスレーブチャネルは、SMTP スレーブと LMTP スレーブの 2 つだけです。次に、これらのプログラムを制御するディスパッチャーについて説明します。

ディスパッチャー

ディスパッチャーは、Messaging Server システムに常駐しているもう 1 つのプロセスです。これはマルチスレッドのトラフィックディスパッチャーであり、着信した SMTP 接続または LMTP 接続を、プロトコル固有の処理が行えるように SMTP サーバースレッドまたは LMTP サーバースレッドのプールへと振り分けます。SMTP サーバープログラムと LMTP サーバープログラムは、ディスパッチャーによって制御されるワークスレッドのプールを提供します。メッセージ処理 (メッセージの拒否または送信先チャネルへのメッセージのキューイング) が完了すると、ワークスレッドは、ディスパッチャーから別の作業を受け入れられる状態になります。

ディスパッチャーは IP アドレスに基づいて着信トラフィックをブロックできるため、サービス拒否攻撃を回避することができます。また、ディスパッチャーは、負荷と設定に基づく SMTP サーバープロセスまたは LMTP サーバープロセスの作成およびシャットダウンも行います。このように、SMTP または LMTP のスレーブチャネルプログラムは、ジョブコントローラの制御下ではなく、ディスパッチャーの制御下にあります。

Local Mail Transfer Protocol (LMTP)

Messaging Server 6.0 リリースでは、複数階層配備におけるメッセージストアに配信を行う LMTP 設定が可能になりました。インバウンドリレーとバックエンドメッセージストアが使用されるこのような環境では、アドレス拡張、自動返信や転送などの配信方法、およびメーリングリストの拡張などに関してリレーが重要な役割を果たします。

バックエンドストアへの配信はこれまで SMTP 上で行われてきました。SMTP では、バックエンドシステムで LDAP ディレクトリの受取人アドレスを再度調べる必要があるため、MTA の全機能が使用されます。速度と効率性を向上するために、MTA では SMTP ではなく LMTP を使用してバックエンドストアにメッセージを配信できます。詳細については、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』の第 15 章「LMTP 配信」を参照してください。

注 – LMTP は、複数階層配備で使用されるように設計されています。LMTP を単一システム配備で使用することはできません。Messaging Server に実装されている LMTP サービスは、ほかの LMTP サーバーまたは LMTP クライアントと連携して動作するように設計されていません。

メッセージストア

メッセージストアは、インターネットメールメッセージの配信、取得、および操作のための専用のデータストアです。メッセージストアは IMAP4 および POP3 クライアントアクセスサーバーとともに動作し、メッセージへの柔軟で容易なアクセスを提供します。また、メッセージストアは HTTP サーバー (`mshttpd`) 経由でも動作します。これにより、Web ブラウザ内の Communications Express に対してメッセージング機能が提供されます。詳細については、この節のほかに、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』を参照してください。

メッセージストアは、一連のフォルダまたはユーザーメールボックスとして構成されます。フォルダまたはメールボックスは、メッセージのコンテナです。それぞれのユーザーには、新しく受信したメールが入る INBOX があります。それぞれの IMAP ユーザーまたは Web メールユーザーには、メールを格納できる 1 つ以上のフォルダがあります。フォルダには、他のフォルダを階層構造で含めることができます。個別のユーザーが所有するメールボックスは非公開フォルダです。非公開フォルダは、所有者の判断で、同じメッセージストア内のほかのユーザーと共有できます。Messaging Server は、IMAP プロトコルによる複数ストア間でのフォルダ共有をサポートします。

メッセージストアには、ユーザーファイルとシステムファイルの 2 つの一般領域があります。ユーザー領域では、それぞれのユーザーの INBOX の位置が 2 階層ハッシングアルゴリズムを使用して決定されます。それぞれのユーザーのメールボックスまたはフォルダは、その親フォルダ内の別のディレクトリとして表されます。各メッセージは 1 つのファイルとして格納されます。フォルダ内に大量のメッセージがある場合は、システムによりフォルダのハッシュディレクトリが作成されます。
ハッシュディレクトリを使用することで、フォルダに大量のメッセージがある場合にファイルシステムが抱える負担が軽減されます。メッセージストアでは、メッセージ自体のほかに、メッセージヘッダー情報の索引とキャッシュ、およびその他の頻繁に使用されるデータが維持されるため、クライアントはメールボックスの情報を迅速に取得し、個別のメッセージファイルにアクセスすることなく一般的な検索を実行できます。

メッセージストアには、多くのユーザーファイル用メッセージストアパーティションを含めることができます。メッセージストアパーティションは、ファイルシステムボリュームに格納されます。ファイルシステムがいっぱいになると、追加のファイルシステムボリュームを作成し、それらのファイルシステムボリューム上に新しいユーザーを格納するためのメッセージストアパーティションを作成できます。

注 – パーティションがいっぱいになると、そのパーティション上のユーザーは、新たなメッセージを格納できなくなります。この問題を解決するには、次の方法があります。

- ユーザーのメールボックスのサイズを縮小する
- ボリューム管理ソフトウェアを使用している場合、別のディスクを追加する
- 別のパーティションを作成し、その新しいパーティションにメールボックスを移動する

詳細については、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』の第 18 章「メッセージストアを管理する」を参照してください。

メッセージストアは、パーティションごとにメッセージそれぞれのコピーを 1 つずつだけ維持します。これは、シングルコピーメッセージストアとも呼ばれます。メッセージストアが複数のユーザー、グループ、または配信リストに宛てられたメッセージを受信した場合、それぞれのユーザーの INBOX にそのメッセージへの参照を追加します。メッセージストアでは、メッセージのコピーをそれぞれのユーザーの INBOX に保存するのではなく、同じデータを重複して保存しないようにしています。既読、返信済み、削除などの個別メッセージステータスのフラグは、それぞれのユーザーのフォルダごとに維持されます。

システム領域には、メッセージストア全体の情報が特定のデータベース形式で格納されており、高速なアクセスを実現しています。システム領域内の情報は、ユーザー領域から再構築できます。Messaging Server にはデータベーススナップショット機能が含まれています。必要な場合には、データベースを既知の状態に迅速に回復できます。Messaging Server には高速回復機能もあり、データベースが破損した場合には、データベース再構築のために長い時間待つことなく、メッセージストアをシャットダウンして、すぐに元の状態に戻すことができます。

メッセージストアはユーザー単位の割り当てをサポートします。割り当ての拡張は有効にすることも無効にすることもできます。ユーザー割り当ては、バイト数またはメッセージ数を使用して設定できます。しきい値を設定して、割り当てがしきい値に達した場合には、ユーザーに警告を出すこともできます。ユーザーが割り当てを超過した場合は、猶予期間中の新規メッセージは保留され、再試行されます。猶予期間の後で、割り当てを超過したユーザーに送信されたメッセージは、未送信通知と共に送信者に返されます。

割り当てを使用する特別なアプリケーションで、ユーザーの割り当てステータスに関係なくメッセージが配信されなければならない場合には、保証メッセージ配信チャネルがあります。このチャネルは、割り当てステータスに関係なくすべてのメッセージを配信するのに使用できます。割り当て使用率のレポートと割り当て警告の送信を行うユーティリティーも用意されています。

Messaging Server とディレクトリサービス

Messaging Server は、Sun Java System Directory Server にバンドルされています。Directory Server は、LDAP (Lightweight Directory Access Protocol) ディレクトリサービスです。Directory Server は、Messaging Server の運用に不可欠な情報のための中央リポジトリを提供します。この情報には、ユーザープロファイル、配信リスト、およびその他のシステムリソースなどが含まれます。

ディレクトリ情報ツリー

ディレクトリは、ディレクトリ情報ツリー (DIT) として知られるツリー形式でデータを格納します。DIT は、ツリーの最上部に 1 つの主要ブランチがあり、その下にブランチおよびサブブランチがある階層構造です。DIT は、組織のニーズに合わせた配備の設計を可能にする柔軟性を備えています。たとえば、実際の業務組織構造に従った DIT の配置を選択することも、業務の地理的なレイアウトに従って選択することもできます。また、使用する DNS レイヤーに 1 対 1 でマッピングした DIT を設計することもできます。実稼働後の DIT の変更は大変な作業となるため、DIT の設計は慎重に行なってください。

DIT は、幅広い管理シナリオに適応する柔軟性も備えています。DIT は、集中型でも分散型でも管理できます。集中型の管理では、1 つの権限で DIT 全体を管理します。集中型管理の場合は、DIT 全体を 1 つのメールサーバー上に配置して使用します。分散型管理では、複数の権限で DIT を管理します。通常は、DIT がいくつかの部分、サブツリー、または異なるメールサーバーに分割された場合に分散型管理を用います。

DIT が大規模な場合、またはメールサーバーが地理的に分散されている場合は、DIT の一部の管理を委託することも検討します。通常は、DIT のそれぞれのサブツリーを管理する権限を割り当てます。Messaging Server では、1 つの権限で複数のサブツリーの管理が可能です。ただしセキュリティ上の理由で、権限は、その権限が所有する DIT のサブツリーの変更だけが可能となっています。

Access Manager が使用されない場合に Messaging Server が使用するデフォルトのスキーマは、Access Manager が使用するスキーマとは異なります。Messaging Server は、Sun Java System LDAP スキーマ 1 および 2 をサポートしており、スキーマの切り替えと移行も可能です。

ディレクトリのレプリケーション

Directory Server はレプリケーションをサポートしており、冗長性と効率性を実現するさまざまな設定が可能です。1 つのホストから別のホストへの DIT の全部または一部をレプリケーションすることで、次の設定機能が利用できます。

- ディレクトリ情報が 1 つのサーバー上にだけあるのではなく、複数のサーバーにレプリケートされるため、ディレクトリ情報へのアクセスがより容易になります。

- ディレクトリ情報はローカルディレクトリサーバーにキャッシュされ、リモートディレクトリサーバーから情報にアクセスする手間を省いています。ディレクトリ情報のキャッシュにより、特に中央ディレクトリへのネットワーク帯域幅が限られている配備では、パフォーマンスが向上します。
- 実際の設定次第で、複数のディレクトリサーバーは単独の集中型サーバーよりも、メールクライアントの要求をより高速に処理できます。

ディレクトリのレプリケーション、ディレクトリパフォーマンスの調整、DIT 構造と設計の詳細については、Sun Java System Directory Server のマニュアルを参照してください。

`http://docs.sun.com/app/docs/coll/1316.1`

メッセージングユーザーのプロビジョニング

Messaging Server ユーザーに対するスキーマとプロビジョニングのオプションについては、第 8 章を参照してください。

第 10 章

Messaging Server サイズ決定戦略の計画

配備を計画する場合には、Messaging Server の設定方法を検討して、パフォーマンス、スケーラビリティー、および信頼性を最適化する必要があります。

サイズ決定はそのための重要な要素の 1 つです。サイズ決定のプロセスを実行することで、Messaging Server ユーザーへの作業負荷の見積もりを踏まえた、希望するレベルのサービスまたは応答時間を実現するために必要となるハードウェアとソフトウェアを確認できます。サイズ決定は反復的な作業です。

この章は、Messaging Server 配備のサイズ決定の基礎について説明し、正しいサイズ決定データを得て配備上の判断ができるようにすることを目的としています。また、Messaging Server のサイズ決定プロセスの背景と理論的根拠についても説明します。

この章には、次の節があります。

注 – 配備にはそれぞれに固有の特徴があるため、この章では特定のサイトに関するサイズ決定情報の詳細な説明はしていません。代わりにここでは、サイズ決定計画を構築する場合には何を考慮しなければならないのかを説明します。配備のハードウェアとソフトウェアのニーズを決定する場合には、ご購入先のテクニカルサポート担当者とともに作業を行なってください。

Messaging Server サイズ決定データの収集

この節の説明を読んで、Messaging Server 配備のサイズ決定に必要なデータを確認してください。この節には、次の項目があります。

メッセージングのピークボリュームの判断

「ピークボリューム」は、1 日の特定の時間帯でメッセージングシステムにトランザクションがもっとも集中したときのトランザクション数です。このボリュームは、サイト間やユーザークラスの違いにより大きく異なります。たとえば、ある中規模企業のマネージャークラスでは、朝の 9 時から 10 時の間、昼の 12 時から 1 時の間、夕方の 5 時から 6 時の間にピークボリュームが発生します。

ピークボリュームの分析には、次の 3 つの基本処理が含まれます。

1. ピークがいつ発生し、どのくらい継続するかを判断します
2. ピークボリューム負荷を前提として配備のサイズを決定します

   パターンの分析が終了すれば、システムの負荷を処理しやすくし、ユーザーの求めるサービスを提供するための選択を行えます。
3. ユーザーが決定したピークボリュームを Messaging Server 配備がサポートできることを確認します

メッセージングの使用率プロファイルの作成

正確なサイズ決定には、負荷の測定が不可欠です。「使用率プロファイル」により、Messaging Server ホスト上のプログラムとプロセスが実行する要素が決定されます。

この節では、使用率プロファイルを作成して、配備で発生する負荷の量を測定する方法について説明します。

使用率プロファイルを作成するには、次の質問に答えてください。

1. システムのユーザー数は何人ですか。

   システムのユーザー数を数える時には、メールアカウントを持ちメールシステムにログインできるユーザーだけでなく、メールアカウントを持っているが、現在システムにはログインしていないユーザーも含めます。特に、次に示しているアクティブなユーザーとアクティブでないユーザーとの相違点に注意してください。

ユーザー	説明
アクティブなユーザー	POP、IMAP、または HTTP のようなメールアクセスプロトコルを使用してメールシステムにログインしているユーザー。アクセスプロトコルの種類により、アクティブなユーザーはメールサーバーに接続していたり、接続していなかったりします。 たとえば、POP ユーザーはメールアカウントを開きますが、メールクライアントからメールサーバーに対して確立される POP 接続は短時間で、断続的です。 ここで説明しているアクティブなユーザーは、`mailuserstatus` や`inetuserstatus` のような `active` ステータスを持ったメール属性ではありません。メール属性の詳細については、『Sun Java System Communications Services Schema Reference 』を参照してください。
アクティブでないユーザー	メールアカウントを持っているが、現在はメールシステムを使用していないユーザー。

ユーザー数が 300 以下のきわめて小規模な配備の場合は、サイズ決定戦略の計画でこのプロセスを実行する必要はありません。クライアントサービス担当者と作業を行い、個別のニーズについて判断します。

2. POP、IMAP、および Messenger Express クライアントがサービスにアクセスするピークボリューム時に、システムへの接続数はどのくらいになりますか。

   特に、サポートするそれぞれのクライアントアクセスサービスの並行接続、アイドル接続、ビジー接続の数に注意します。

接続	説明
並行接続	ある時間にメールシステム上で確立される、固有の TCP 接続またはセッション (HTTP、POP、または IMAP) の数。 アクティブなユーザーは複数の並行 IMAP セッションを行うことができます。一方、POP クライアントまたは Messenger Express クライアントを使用するユーザーは、クライアントごとに 1 つの接続しか確立できません。さらに、POP 接続と Messenger Express 接続は、サーバーに接続してデータを取得し、サーバーへの接続を切断して、データの表示、ユーザー入力の受け入れを行い、そしてメールサーバーへの再接続を行うため、POP および Messenger Express クライアントアクセスサービスのアクティブなユーザーは、ある時点においてはアクティブな接続を行わずにサービスにアクセスすることも可能です。
アイドル接続	確立された IMAP 接続で、時々送信される `check` または `noop` コマンドを除き、メールクライアントと Messaging Server との間で情報送信を行わないもの。
ビジー接続	進行中の接続。ビジー接続の例としては、メールクライアントが送信したばかりのコマンドを処理中、つまり、メールクライアントに応答を送り返している状態のメールサーバーがあります。

配備における「並行接続」の数は、次のいずれかの方法で決定します。

a. UNIX プラットフォームで `netstat` コマンドを使用して、確立された TCP 接続数をカウントします。

b. Messenger Express または IMAP のユーザーの、最後のログイン時刻とログアウト時刻を取得します。詳細については、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』を参照してください。

3. 大規模な配備を行う場合には、ユーザーをどのように組織化しますか。

次の選択肢が考えられますが、これに限られません。

- アクティブなユーザーとアクティブでないユーザーをそれぞれのマシンから集めて、アクティブユーザーを集めたマシンと非アクティブユーザーを集めたマシンとに分けます。

  アクティブでないユーザーがアクティブなユーザーになる場合は、そのユーザーをアクティブなユーザーのマシンに移動します。このアプローチを採用すると、アクティブなユーザーとアクティブでないユーザーを同じマシンに置いた場合よりも、必要なハードウェアを減らすことができます。

- ユーザーをサービスのクラス別に分けます。

  コントリビュータ、マネージャ、エグゼクティブのユーザーを、それぞれのサービスのクラス、権限、専門サービスに応じたメールストレージ容量の割り当てを提供するマシンに分けます。

4. それぞれのメールボックスで使用されるストレージの量はどのくらいですか。

   メールボックスあたりのストレージの容量を測定するときには、指定した割り当てではなくメールボックスの実際の使用率で見積もります。ごみ箱内のメッセージもディスク容量と割り当てを消費します。

5. インターネットからどれぐらいの数のメッセージがメッセージングシステムに送信されますか。

   メッセージの数は、ピークボリューム時の 1 秒あたりのメッセージ数で測定します。

6. ユーザー別ではどれぐらいの数のメッセージが送信されますか。

   - メールシステムのエンドユーザーに対して送信される数
   - インターネットに対して送信される数

   このメッセージの数も、ピークボリューム時の 1 秒あたりのメッセージ数で測定します。

7. 異なるサイズ範囲では、配信分布状態はどのようになっていますか。

   例:

   - 5K バイト未満
   - 5K バイト以上 10K バイト未満
   - 10K バイト以上 100K バイト未満
   - 100K バイト以上 500K バイト未満
   - 500K バイト以上 10M バイト未満
   - 10M バイト以上

   配信されるメッセージのサイズがわからない場合は、メールシステムの平均のメッセージサイズを使用しますが、これはサイズの範囲がわかる場合ほど有効ではありません。

   メッセージのサイズは、MTA の配信レート、メッセージストアへの配信レート、メッセージ取得のレート、およびウィルス対策用またはスパム防止用のフィルタの処理に影響を与えるため、特に重要なものです。

8. SSL/TLS を使用しますか。使用する場合は、ユーザーの何パーセントが、またどのようなタイプのユーザーが使用しますか。

   たとえば、ある組織では、ピーク時間中に IMAP 接続の 20 パーセントで SSL が使用されます。

9. 何らかの SSL 暗号化アクセラレータハードウェアを使用する予定がありますか。

10. ウィルススキャニングまたはその他の専用のメッセージ処理を使用し、その処理をすべてのユーザーに適用しますか。

    Messaging Server の設定により、MTA は専用の処理で指定された基準に一致するすべてのメッセージをスキャンする必要があり、その結果システムの負荷が増大します。

11. POP ユーザーに対し、メールへのアクセス可能頻度を制限するポリシーを適用しますか。適用する場合、どのくらいの頻度にしますか。
12. IMAP ユーザーに対し、標準のクライアントを強制しますか。それとも、各ユーザーが選択できるようにしますか。

    IMAP クライアント数が増加するとサーバーへの同時接続数も増加します。したがって、多くのフォルダを開いているパワーユーザーは、多くの同時接続を使用している可能性があります。
13. ユーザーがフォルダを共有できるようにしますか。共有できるようにする場合、すべてのユーザーに許可しますか。それとも一部のユーザーにだけ許可しますか。

これらの質問に答えることで、配備のための、準備段階としての使用率プロファイルが完成します。Messaging Server のニーズの変更に応じて、この使用率プロファイルにも修正を加えます。

その他の質問

次の質問は使用率プロファイルの作成に使用できるものではありませんが、配備のサイズ決定戦略には重要なものです。これらの質問にどのように答えるかによって、ハードウェアの追加を検討しなければならなくなる場合もあります。

1. 配備にどの程度の冗長性を持たせますか。

   たとえば、高可用性の実現を考えている場合です。どれくらいの停止時間であれば許容範囲であるかを検討してください。また、クラスタリングテクノロジが必要かどうかも検討してください。
2. どのようなバックアップ戦略と回復戦略 (障害回復、メールボックスの復元、サイトのフェイルオーバーなど) を実行しますか。回復タスクが完了するまでにどのくらいの時間を予想しますか。
3. DMZ を使って内部ネットワークと外部ネットワークを分離する必要がありますか。すべてのユーザーが内部ネットワークを使用していますか。それとも、一部のユーザーはインターネットを使って接続しますか。

   プロキシサーバー (MMP、MEM) と独立した MTA 層が必要になる可能性があります。
4. 応答時間の要件を記述してください。スループットの要件を記述してください。
5. リソースの使用条件を具体的に記述してください。CPU 使用率は平均 80 % でかまいませんか。それとも、80 % はピーク時のみですか。
6. メッセージングサーバーをいくつかの地理的に異なる場所に設置しますか。ユーザーのメールが地理的に分散配置される可能性はありますか。
7. アーカイブを使ってメールメッセージをある一定期間保管しておく必要がありますか。
8. すべてのメッセージをロギングする法律上の必要性がありますか。送受信されたすべてのメッセージのコピーを保存しておく必要がありますか。

メッセージングユーザーベースの定義

使用率プロファイルの作成が完了したら、次にそれをこの節で説明されている定義済みのユーザーベースの例と比較してみます。「ユーザーベース」は、ユーザーが送受信するメッセージサイズの範囲と、ユーザーが実行するメッセージング操作のタイプで構成されます。メッセージングユーザーは、5 つのユーザーベースに分類されます。

- 143 ページの「軽量級の POP ユーザー」
- 143 ページの「重量級の POP ユーザー」
- 144 ページの「軽量級の IMAP ユーザー」
- 144 ページの「標準的な IMAP ユーザー」
- 144 ページの「標準的な Messenger Express/Communications Express ユーザー」

この節のユーザーベースの例では、ユーザーの行動を幅広く一般化しています。特定の使用率プロファイルは、このユーザーベースとは多少異なるかもしれません。これらの差異は、負荷シミュレータを実行するとき (145 ページの「Messenger Express 負荷シミュレータの使用」を参照) に調整できます。

軽量級の POP ユーザー

軽量級の POP ユーザーベースは、一般に、簡単なメッセージング要件を持つ家庭のダイアルアップユーザーで構成されます。それぞれの並行クライアント接続は、1 時間あたり約 4 件のメッセージを送信します。これらのユーザーは、1 回のログインセッション中にすべてのメッセージの読み取りと削除を行います。さらに、これらのユーザーは 1 回の受信では、自分のメッセージの作成と送信をほとんど行いません。メッセージの約 80 パーセントが 5K バイト以下のサイズで、約 20 パーセントが 10K バイト以上です。

重量級の POP ユーザー

重量級の POP ユーザーベースは、高速ブロードバンドのユーザーか小規模な企業のアカウントであるのが一般的で、軽量級の POP ユーザーベースよりメッセージに関して高度な要件を持っています。このグループは、ケーブルモデムか DSL を使用してサービスプロバイダに接続します。それぞれの並行クライアント接続は、1 時間あたり約 6 件のメッセージを送信します。メッセージ受信者数の平均は 1 メッセージあたり約 2 人です。メッセージの 65 パーセントが、5K バイト以下のサイズです。このユーザーベースのメッセージの 30 パーセントが、5K バイトから 10K バイトの間のサイズです。5 パーセントのメッセージが 1M バイトを超えるサイズです。ユーザーのうち、85 パーセントが読んだ後ですべてのメッセージを削除しています。ただし、15 パーセントのユーザーは、メッセージをサーバー上に残したまま数回のログインを行なってから、メッセージを削除しています。メールは、これらのメールボックスのわずかな割合を占めるだけです。同じメッセージがサーバーから数回取得される場合もあります。

軽量級の IMAP ユーザー

軽量級の IMAP ユーザーベースは、高速なブロードバンドインターネットサービスを利用するユーザーに代表されます。このユーザーが利用するサービスには、メッセージ検索やクライアントフィルタのような高度なメッセージングシステム機能のほとんどが含まれます。このユーザーベースは、メッセージのサイズ、受信者の数、それぞれの並行接続別の送受信メッセージ数に関して、重量級の POP ユーザーベースに類似しています。軽量級の IMAP ユーザーは一般的に、一度のログインでセッションを数時間継続し、ログアウトする前にほとんどまたはすべてのメールを削除します。その結果、ログインセッション中にメールが蓄積されますが、通常はメールボックスに 20 から 30 件以上のメッセージが蓄積されることはありません。ほとんどの受信ボックスで、残っているメッセージの数は 10 件以下です。

標準的な IMAP ユーザー

標準的な IMAP ユーザーベースは、高度な企業ユーザーに代表され、営業日にはログインセッションがほぼ 8 時間継続します。これらのユーザーは、大量のメールの送受信と保管を行います。さらに、これらのユーザーの場合、メッセージの割り当ては無制限か、またはかなり大きなものとなります。受信ボックスには大量のメールが 1 日中蓄積されていき、溢れそうになったときにはすべて、または一部が消去されます。メッセージは定期的にフォルダに整理され、1 時間に何度かの割合で検索されます。それぞれの並行クライアント接続は、1 時間あたり約 8 件のメッセージを送信します。このカテゴリのユーザーの場合、送信する 1 件のメッセージの平均受信者数は 4 人で、同じメッセージサイズを重量級の POP および軽量級の IMAP のユーザーベースとして混在させます。

標準的な Messenger Express/Communications Express ユーザー

標準的な Messenger Express/Communications Express ユーザーベースは、標準的な IMAP に似ています。このユーザーベースのメッセージのサイズは、標準的な IMAP、軽量級の IMAP、および 重量級の POP ユーザーと同じです。メッセージの配信頻度も標準的な IMAP ユーザーと同じです。

組織内で、特に複数のクライアントアクセス手段を提供する場合は、おそらく複数のユーザーベースを持つことになります。ユーザーベースをこれらのカテゴリの中から決定したら、使用率プロファイルと145 ページの「Messenger Express 負荷シミュレータの使用」で説明されている負荷シミュレータを使用して、そのユーザーベースのテストを行います。

Messenger Express 負荷シミュレータの使用

Messaging Server のパフォーマンスを測定するには、メッセージングユーザーベース (143 ページの「メッセージングユーザーベースの定義」を参照) およびメッセージングの使用率プロファイル (138 ページの「メッセージングの使用率プロファイルの作成」を参照) を負荷シミュレータへの入力として使用します。

負荷シミュレータは、ピークボリューム環境を作り出し、サーバーにかかる負荷の量を調整します。これにより、システムに過負荷をかけることなく希望する応答時間を実現するには、ハードウェア、スループット、または配備のアーキテクチャーを変更する必要があるかどうかを判断できます。

▼ 負荷シミュレータを使用するには

手順

1. テストするユーザーベース **(**軽量級の **IMAP** など**)** を定義します。
   必要に応じて、使用率プロファイルに最適化するように個別のパラメータを調整します。
2. テストするハードウェアを定義します。
3. 負荷シミュレータを実行し、ユーザーベースを使用してテストされたハードウェアの最大並行接続数を測定します。
4. 結果を記録して、稼働中の配備の結果と比較します。
5. ピーク負荷状態の応答時間が組織で容認されるレベルになるまで、さまざまなユーザーベースとハードウェアを使用してこのプロセスを繰り返します。

注 **–** 推奨負荷シミュレータとサポートについては、ご購入先のクライアントサービス担当者に連絡してください。

Messaging Server システムパフォーマンスの評価

負荷シミュレータを使用してハードウェアとユーザーベースの評価を行ったら、システムパフォーマンスを測定する必要があります。次のトピックで、システムの全体的なパフォーマンスを向上させる方法について説明します。

Messaging Server のメモリー使用率

配備で使用するそれぞれのマシンに、適切な量の物理メモリーが搭載されていることを確認してください。物理メモリーを追加するとパフォーマンスが向上し、ピークボリューム時でもサーバーが適切に動作するようになります。メモリーが不足していると、Messaging Server で過剰なスワッピングが発生し、効率的に動作しません。

少なくとも、1 つの CPU あたり 1G バイトのメモリーを用意してください。ほとんどの配備で、UltraSPARC® III システムの CPU 1 つにつき 2G バイトのメモリーが必要になります。

Messaging Server のディスクスループット

ディスクのスループットとは、システムでメモリからディスクに、またはディスクからメモリに転送されるデータ量のことです。このデータ転送レートは、Messaging Server のパフォーマンスに重大な影響を及ぼします。システムのディスクスループットを向上させるには、次のことを考慮します。

- 保守作業を検討し、バックアップのための十分な帯域幅があることを確認します。特にリモートバックアップの場合は、ネットワーク帯域幅にも影響を与えます。私設バックアップネットワークは、より効率的な代替バックアップ手段となります。
- ストアのパーティションと、`tmp` や `db` のようなストアデータ項目の分割を慎重に行なって、スループットを向上させます。
- 大規模な配備では、ユーザーベースが必ず RAID (Redundant Array of Independent Disks) 環境全体に分散されるようにします。
- ディスクからデータを取得する操作のスピードを向上させるために、データを複数のディスクでストライピングします。
- RAID がハードウェア上に存在しない場合は、RAID のサポートに十分な CPU リソースを割り当てます。

ディスク I/O を、帯域幅ではなく IOPS (1 秒あたりの I/O の合計) で測定することをお勧めします。システムがきわめて短い応答時間 (10 ミリ秒未満) で処理できる、個別のディスクトランザクションの数を測定する必要があります。

Messaging Server のディスク容量

サーバーシステムのディスク容量を計画する際には、環境ソフトウェア、Messaging Server ソフトウェア、およびメッセージの内容を運用するための容量、さらにトラッキングのための容量を確実に含める必要があります。可用性が要求される場合には、必ず外部ディスクアレイを使用します。ほとんどのシステムで、内部システムディスクでは 4 台までのディスクしかサポートされないため、パフォーマンスを向上させるには外部ディスクが必要となります。

メッセージストアパーティションでは、全メッセージの合計サイズに 30 % のオーバーヘッドを加えたストレージサイズが必要です。

さらに、ユーザーディスク容量を割り当てます。この容量は、通常、サイトのポリシーに従って決定されます。

注 – 配備計画には、障害回復のためのメッセージストアのバックアップ方法を含める必要があります。Messaging Server では、Solstice Backup (Legato Networker)、`imsbackup` ユーティリティ、およびファイルシステムスナップショットバックアップがサポートされています。バックアップメディアを遠隔地に格納した方がよい場合もあります。バックアップは、サーバーの処理に影響しない範囲で、できるだけ頻繁に実行することをお勧めします。

MTA メッセージキューのディスクサイズ決定

Messaging Server の MTA キューの動作は、配信されるのを待機しているメッセージ用の一時記憶域を提供することです。保証されたサービス提供を維持するために、メッセージは持続的方式でディスクに書き込まれます。メッセージを配信できない場合、MTA は最終的に断念してメッセージを送信者に戻すまで再試行します。

メッセージキューのパフォーマンス

MTA キューのディスクサイズ決定は、MTA のパフォーマンスを改善するための重要なステップです。MTA のパフォーマンスは、ほかのどのシステムリソースにもましてディスク I/O に直接影響されます。したがって、複数のディスクスピンドルで構成されるディスクボリュームを計画することをお勧めします。これらのディスクスピンドルは、ディスク RAID システムを使用して連結およびストライプ化されます。

MTA のパフォーマンスに問題があると、エンドユーザーはすぐにその影響を受けます。ユーザーが電子メールクライアントの「送信」ボタンを押すとき、MTA はメッセージがメッセージキューにコミットされるまで、メッセージの受信を完全には受理しません。したがって、メッセージキューのパフォーマンス改善は、エンドユーザーの立場から見ると応答時間の短縮につながります。

メッセージキューの可用性

SMTP サービスは、保証されたメッセージ配信サービスであると考えられます。これは、サービスが配信しようとしているメッセージを途中で失わないことを、メッセージングサーバーがエンドユーザーに対して保証するという意味です。MTA キューシステムを設計するときは、メッセージが失われないことを保証するためにあらゆる努力を払う必要があります。この保証は通常、さまざまな RAID 技術を使用して、冗長化されたディスクシステムを実装することによって達成されます。

メッセージキューの利用可能なディスクサイズの決定

次のいずれかの条件が発生した場合、キューのサイズは非常に大きくなります。

- サイトにおいてネットワーク接続に重大な問題がある
- MTA の設定で、メッセージの保持期間が長すぎる
- メッセージに関して、このマニュアルで扱っていない何らかの問題がある

次節以降で、これらの問題について説明します。

ネットワーク接続の問題に備えた計画

ネットワーク接続の問題により、MTA がメッセージを配信できない場合があります。そのような場合、定義された再試行間隔に従って MTA が配信を再開できるようになるまで、メッセージはキューに格納されます。

そのような中断に備えたディスク領域の計画は、「メッセージキューのサイズ決定の一般則」という単純なルールに基づいて行います。

1. 配信が予測される 1 分あたりの平均メッセージ数を決定します (N)。
2. メッセージの平均サイズ (K バイト) を決定します (S)。
3. 典型的なネットワーク接続障害の最大持続時間 (分) を決定します (T)。

ディスクキューサイズを見積もるための式は次のようになります。

ディスクキューサイズ（K バイト）= N x S x T

MTA の配信再試行設定の調整

システムがメッセージをまったく配信できなくなることがあります。この状態では、MTA は配信を再試行するまでの一定期間 (再試行間隔として定義された期間)、メッセージをメッセージキューに保留します。この状態は、MTA が配信を断念してメッセージを送信者に戻すまで続きます。メッセージが配信できなくなる理由の多くは予測不可能です。メッセージが配信できない理由には、ネットワーク接続の問題、送信先サーバーのビジー状態、ネットワークの混雑などさまざまなものがあります。

ビジー状態のサーバー上では、高ボリュームアクティビティの期間中、このような一時的に格納されるメッセージが大量に蓄積される可能性があります。大量の蓄積によって、ディスク容量の問題が発生する可能性があります。そのような蓄積を防ぐには、より短い間隔で配信を再試行するように MTA を調整します。

再試行間隔は、imta.cnf ファイルの「Channel Block」設定で設定します。このファイルの構造としては、書き換えルールとチャネルブロックの 2 つの部分から構成されています。チャネルブロックは、特定のディスクキューと、それに関連するプロセスの動作を定義します。ここでの説明に関係するのは tcp_local チャネルです。tcp_local チャネルは、企業のローカルネットワークの外部のサイト、すなわち、インターネットを経由した場所への配信を提供します。

tcp_local チャネルの再試行間隔は当初、デフォルトチャネルブロックによって設定されます。デフォルトチャネルブロックを使用すると、設定を複製することによって設定の繰り返しを防ぐことができます。

デフォルトチャネルブロックの例を次に示します。

```
defaults notices 1 2 4 7 copywarnpost copysendpost postheadonly
noswitchchannel immnonurgent maxjobs 7 defaulthost
red.siroe.com red.siroe.com
```

チャネルブロックの構造の先頭はチャネル名です。前述の例において、これは、これらの設定を持たないチャネルに適用されるデフォルトチャネルブロックです。2 番目の部分はチャネルキーワードのリストです。

notices キーワードは、メッセージ配信通知 (MDN) が送信者に返送されるまでに経過可能な時間を指定します。このキーワードは notices キーワードで始まり、それに続けて、再試行期間を設定する一連の数値を指定します。デフォルトでは、MTA は配信を再試行し、送信者に通知を返送します。これらの通知は「ポストマスター」からエンドユーザーの受信箱に送られます。

この例では、MTA は 1 日、2 日、および 4 日を経過するタイミングで配信を再試行します。7 日経過すると、MTA はメッセージを送信者に戻し、そのメッセージを配信失敗とみなします。

多くの場合、MTA のデフォルト設定で適切なパフォーマンスが得られます。場合によっては、メッセージキュー用のディスク領域不足などのリソース枯渇の可能性を回避するため、MTA を調整する必要があります。これは製品の制限ではなく、ハードウェアおよびネットワークリソースを含めた Messaging Server システム全体の制限です。

ディスクサイズに関して起こりうるこれらの問題を考慮すると、ユーザー数の多い配備では、あまり短い間隔でメッセージ配信を試みるのは好ましくない場合があります。そのような状況に該当する場合は、次に示すドキュメントを参考にしてください。

参考資料

詳細については、次のマニュアルを参照してください。

- 『Sun Java System Messaging Server 6 2005Q4 管理ガイド』の「通知メッセージの配信間隔を設定するには」
- 『Sun Java System Messaging Server 6 2005Q4 管理ガイド』の第 12 章「チャネル定義を設定する」

Messaging Server のネットワークスループット

ネットワークスループットは、一定時間内にクライアントアプリケーションとサーバー間のネットワークで転送可能なデータ量のことです。ネットワークに接続されたサーバーがクライアントからの要求に応答できない場合、通常クライアントは要求の再送信を何度も行います。再送信のたびに、システムにはオーバーヘッドと余分なネットワークトラフィックが生じます。

データの完全性とシステムのパフォーマンスを向上させて、ネットワークの混雑を解消することで、再送信の数を減らすことができます。

- ボトルネックを解消するには、ネットワークインフラストラクチャーが負荷を処理する能力を確保します。
- ネットワークを分割します。たとえば、クライアントアクセスに 100Mbps のイーサネットを、バックボーンに 1G バイトイーサネットを使用します。
- 将来の拡張に備えて十分な容量を確保するには、ネットワークを構築するときに理論最大値を使用してはなりません。
- トラフィックのフローを異なるネットワークパーティションに分割して衝突を減らし、帯域幅の使用を最適化します。

Messaging Server の CPU リソース

メッセージストア、MTA、および複合サービス (MMP および Messenger Express マルチプレクサ) だけを実行しているシステムに対して、十分な CPU を用意します。さらに、使用を計画している RAID システムにも十分な CPU を用意します。

Messaging Server アーキテクチャー戦略の構築

システムパフォーマンスのニーズを確認したあと、Messaging Server 配備のサイズ決定で次のステップは、アーキテクチャーの決定に基づいて特定のコンポーネントのサイズを決定することです。

以降の節で、2 層と 1 層のアーキテクチャーを配備する場合のサイズ決定で考慮しなければならないことについて説明します。

注 – アーキテクチャー計画の詳細については、第 11 章を参照してください。

---

2 層 Messaging Server アーキテクチャー

2 層アーキテクチャーでは、Messaging Server 配備を 2 つの層に分割します。1 つはアクセス層、もう 1 つはデータ層です。簡略化した 2 層アーキテクチャーでは、MMP と MTA をアクセス層に追加します。MMP が POP と IMAP メールリーダーのプロキシとして機能し、MTA が送信されたメールのリレーを行います。データ層には、メッセージストアと Directory Server を配置します。図 10–1 は簡略化した 2 層アーキテクチャーを示しています。

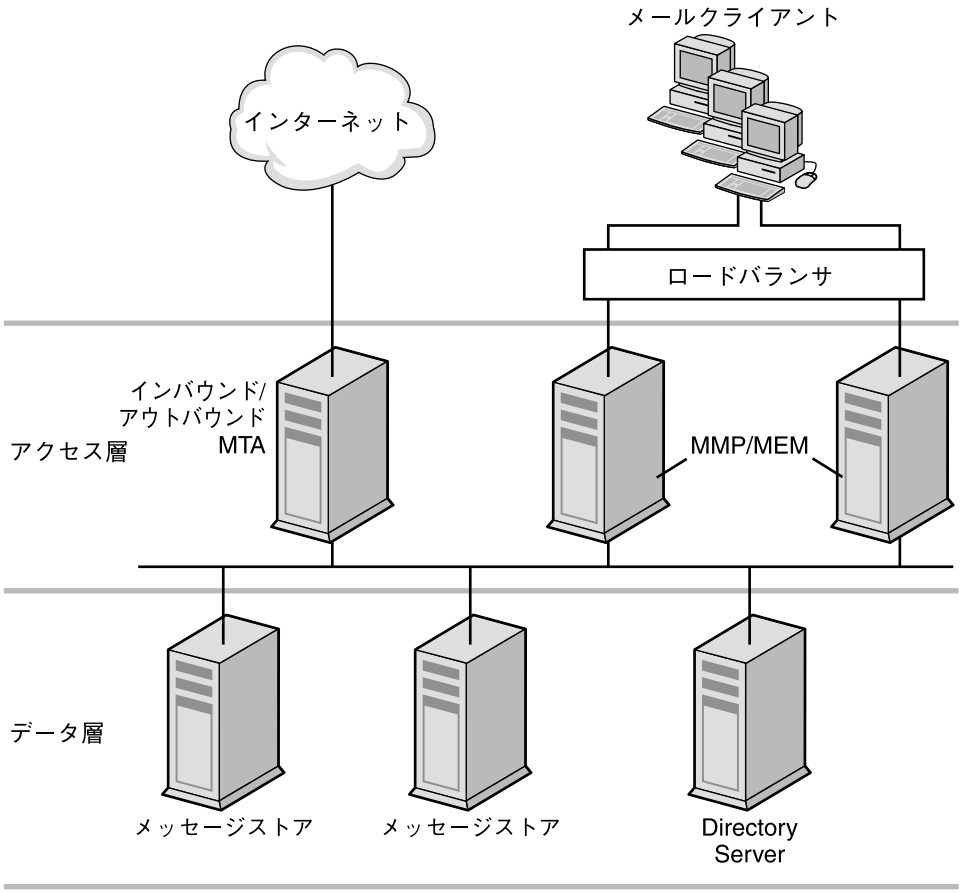

図 **10–1** 簡略化した Messaging Server の 2 層アーキテクチャー

1 層アーキテクチャーに対して、2 層アーキテクチャーには、サイズの決定に影響を与えるかもしれないいくつかのメリットがあります。2 層アーキテクチャーでは次のことが可能です。

- 1 層アーキテクチャーより簡単な保守

- SSL、ウィルススキャニング、メッセージ再処理、サービス拒否攻撃のような負荷の高いプロセスのオフロード
- 限られた停止時間内でのより簡単な拡張管理とシステムのアップグレード

次のいくつかの節で、2 層配備における特定のコンポーネントのサイズ決定方法について説明します。

▼ メッセージストアのサイズ決定

メッセージストアのサイズ決定の目的は、ストアが処理可能な最大並行接続数を確認し、1 秒間にストアに配信されるメッセージの数を決定することです。

手順

1. 負荷シミュレータを使って集めた数値をもとに、マシン **1** 台あたりのストアマシン数と並行接続数を決定します。サイズ決定ツールの詳細については、**145** ページの「**Messenger Express** 負荷シミュレータの使用」を参照してください。
2. それぞれのストアマシンに必要なストレージの容量を決定します。
3. バックアップとファイルシステム回復時の復元で必要な場合は、複数のストアパーティションまたはストアマシンを使用します。

   ご購入先のクライアントサービス担当者に、メッセージストアのユーザーの推奨最大数を尋ねてください。推奨数値を得るには、次の点を理解しておく必要があります。

   - 使用率のパターン (145 ページの「Messenger Express 負荷シミュレータの使用」を参照)。
   - 配備内のハードウェアすべてのアクティブなユーザーの最大数。
   - バックアップ、復元、回復に要する時間。これらの時間は、メッセージストアのサイズが大きくなるにつれて長くなります。

▼ インバウンド MTA とアウトバウンド MTA のサイズを決定するには

一般的には、MTA サービスはインバウンドサービスとアウトバウンドサービスとに分けます。次に、同じ方法でそれぞれのサイズを決定できます。MTA のサイズ決定の目的は、1 秒間にリレーできるメッセージの最大数を決定することです。

インバウンド MTA のサイズを決定するには、実稼働環境でのインバウンド MTA の raw パフォーマンスを知る必要があります。

手順

1. インバウンド **MTA** の **raw** パフォーマンスをもとに、**SSL**、ウィルススキャニングプロセス、その他の臨時のメッセージ処理を追加します。
2. **1** 日のピークボリューム時のサービス拒否攻撃についても考慮します。

3. 十分な量の **MTA** を追加して、負荷分散と必要に応じた冗長性を確保します。
   冗長性を持たせることで、1 つ以上のタイプのマシンで、スループットや応答時間に実質的な影響を与えることなくピーク負荷を処理できます。

   さらに、一時的なメッセージの量に対して十分なディスク容量を計算して、ネットワーク上の問題やリモート MTA の機能停止に備えます。

▼ 複合サービスのサイズを決定するには

MMP と MEM のサイズを決定する場合には、システム負荷、特に MMP に対する POP と IMAP の並行接続数と、MEM に対する HTTP 接続数に基づいて計算を行います。

注 – ここでの手順は、MEM と MMP が同じマシンにインストールされていることを前提にしています。

さらに、次のことを実行する必要があります。

手順
1. 必要に応じて、**MMP** と **MMP** の**SSL** 用に **CPU** またはハードウェアアクセラレータを追加します。
2. マシンに **MEM** を設定している場合は、メモリを追加します。
3. **MMP** の **SMTP** プロキシにディスクを追加します。
4. サービス拒否攻撃について考慮します。
5. 必要に応じて、負荷分散と冗長性の能力を追加します。
   インバウンド MTA ルーターの場合と同様に、配備に冗長性を持たせることでスループットや応答時間に実質的な影響を与えることなく、各タイプの 1 台以上のマシンでもピーク負荷を処理します。

単一層 Messaging Server アーキテクチャー

単一層アーキテクチャーは、アクセス層とデータ層に分割されていません。MTA、メッセージストア、そして場合によっては Directory Server が 1 つの層にインストールされます。図 10–2 は単一層アーキテクチャーを示します。

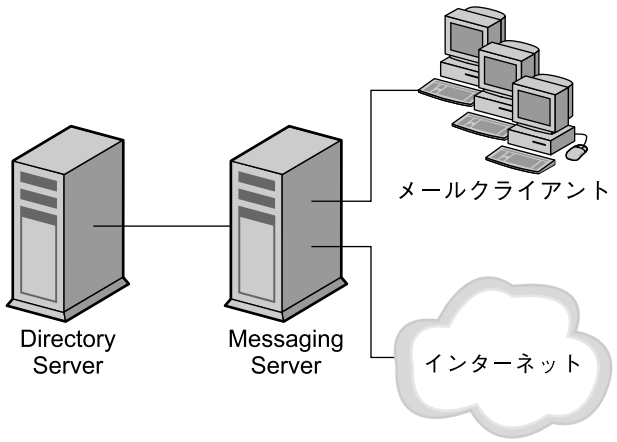

図 **10–2** 簡略化した Messaging Server の単一層アーキテクチャー

2 層アーキテクチャーと比較して、単一層アーキテクチャーはハードウェアに対する初期投資が少なくて済みます。しかし、1 層アーキテクチャーを選択した場合は、保守のためにかなりの停止時間を見込んでおく必要があります。

▼ Messaging Server の単一層アーキテクチャーのサイズを決定するには

手順 **1.** 151 ページの「**2** 層 **Messaging Server** アーキテクチャー」でのサイズ決定と同様に、メッセージストアのサイズを決定します。

2. 必要に応じて **SSL** 用の **CPU** を追加します。

3. サービス拒否攻撃について考慮します。

4. **SMTP** 接続数の増加に対応してディスクを追加します。

5. アウトバウンド **MTA** ルーター用のディスクを追加します。

注 **–** 単一層アーキテクチャーおよび 2 層アーキテクチャーにおけるメッセージングコンポーネントのサイズ決定に関する特別な手順については、ご購入先のクライアントサービス担当者に連絡してください。

第 11 章

Messaging Server アーキテクチャーの開発

この章では、Messaging Server のアーキテクチャーの設計方法について説明します。このアーキテクチャー設計により、ハードウェアリソースとソフトウェアリソースに Messaging Server コンポーネントをどのように配置するかが決定されます。

この章には、次の節があります。

2 層メッセージングアーキテクチャーの理解

2 層アーキテクチャーにより、スケーラビリティーと信頼性のために最適化された設計が可能になります。単独のホストでメッセージングシステムのすべてのコンポーネントを実行する代わりに、2 層アーキテクチャーでは、コンポーネントを異なるマシンに分割します。このようなコンポーネントの分割により、特別な専用機能が実行されます。たとえば、追加のメッセージストレージが必要になったり、より多くのアウトバウンドリレーが必要になったりするなど、特定の機能コンポーネントの負荷が増大すると、サーバーを追加して増大する負荷に対応できます。

2 層アーキテクチャーは、「アクセス層」と「データ層」で構成されます。アクセス層は、アーキテクチャーの中で、配信、メッセージアクセス、ユーザーログイン、および認証を処理する部分です。データ層は、すべてのデータを維持する部分です。これには、LDAP マスターサーバーと、ユーザーメッセージを格納するよう設定された Messaging Server マシンが含まれます。

図 11–1 は、2 層アーキテクチャーの例を示しています。

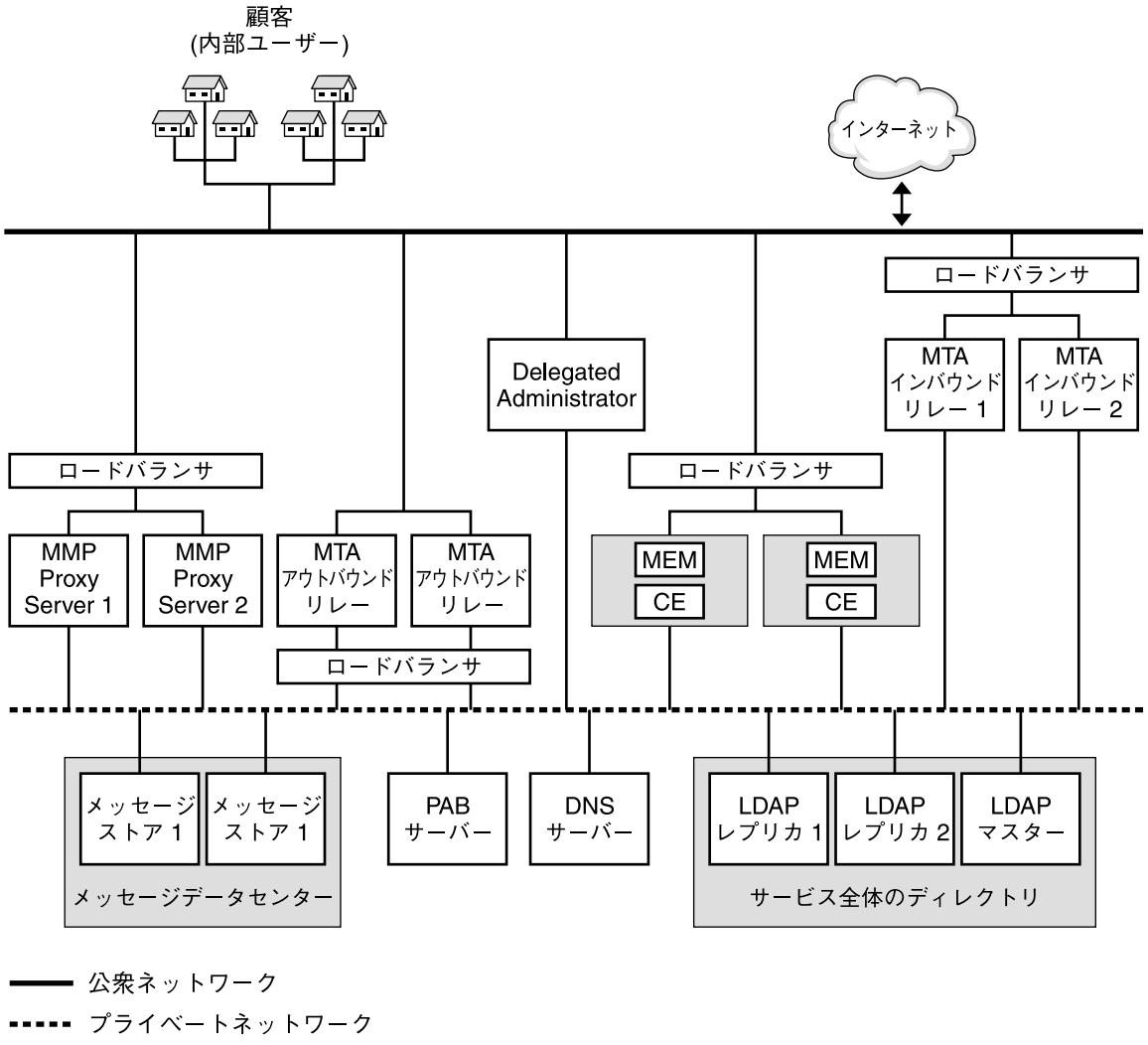

図 **11–1** 2 層 Messaging Server アーキテクチャー

次でこれらの機能部分について説明します。

公衆アクセスネットワーク: Messaging Server を内部ユーザーとインターネットに接続するネットワークです。それぞれの配備で独自のネットワーク要件が定義されますが、基本的な Messaging Server の要件は、SMTP、POP、IMAP、および HTTP のような標準のプロトコルを使用したエンドユーザーとインターネットへの接続性です。

私設データネットワーク: このネットワークは、公衆アクセスネットワークと Messaging Server データの間で、セキュリティー保護された接続を提供します。セキュリティー保護されたアクセス層と、サービス全体のディレクトリ、メッセージデータセンター、および個人アドレス帳 (PAB) サーバーが含まれるデータ層で構成されます。

LDAP ディレクトリサーバー: ユーザーベースに関する情報の格納と取得に使用されるディレクトリサーバーです。ユーザーとグループエイリアス、メールホスト情報、配信設定などを格納します。設計の要件次第で、システムの同一ディレクトリを複数格納することも可能です。図 11–1 は、マスターディレクトリと 2 つのレプリカを示します。LDAP ディレクトリサーバーは、Messaging Server 製品の一部として提供されます。必要であれば、既存の Sun Java System Directory Server ディレクトリのデータを使用することも可能です。既存のディレクトリのデータ形式は、Messaging Server スキーマに準拠している必要があります。

メッセージストア: ユーザーメールを保持し、格納します。「バックエンド」と呼ばれることもあります。メッセージストアは、IMAP サーバー、POP サーバー、および Web メール (Messenger Express) サーバーのようなメッセージアクセスコンポーネントを指すこともあります。図 11–1 は、2 つのメッセージストアを持つ配備を示します。必要に応じて、さらにストアを追加することもできます。

個人アドレス帳 **(PAB)** サーバー: LDAP サーバー内のユーザーのアドレスを保存および取得します。このサーバーは、前述の LDAP サーバーと同一であってもなくてもかまいません。

DNS サーバー: ホスト名を IP アドレスにマップします。DNS サーバーは、メッセージを外部ドメインにルーティングするときに、どのホストに接続するかを判断します。内部的には、DNS は実際のサービスをマシン名にマップします。DNS サーバーは、Messaging Server 製品の一部ではありません。Messaging Server をインストールする前に、稼働状態の DNS サーバーをインストールする必要があります。

ロードバランサ: ネットワーク接続について、均一にバランスを取るか、複数のサーバーにアルゴリズムを適用してバランスを取ります。ロードバランサを使用すると、1 つのネットワークアドレスで多数のサーバーを表すことができるため、トラフィックのボトルネックを解消し、トラフィックフローの管理と高いレベルのサービス保証が可能になります。図 11–1 には、MMP 用、MTA 用、および MEM 用のロードバランサが含まれています。ロードバランサは Java Enterprise System 製品の一部ではありません。ロードバランサをメッセージストアまたはディレクトリマスター上で使用することはできません。ロードバランサは、MMP、MEM、Communications Express、MTA、ディレクトリコンシューマへの接続用として使用したり、Messaging Server の MTA が Brightmail 製品を使用する状況で使用したりします。

MTA インバウンドリレー: 外部 (インターネット) サイトからのメッセージを受信し、それを内部ホストおよびローカルのメッセージストアサーバーにルーティングする専用MTA です。この MTA は外部からの最初のコンタクトポイントとなるため、MTA インバウンドリレーには、権限のないリレーを防ぎ、スパムをフィルタリングし、サービス拒否攻撃に対抗する機能が追加されます。MX レコードを使えば、着信メールトラフィックの負荷を分散できます。詳細については、163 ページの「MX (メール交換) レコード」を参照してください。

MTA アウトバウンドリレー: 内部または承認されたユーザーからのメールだけを受け取り、それをその他の内部ユーザーまたは外部 (インターネット) ドメインにルーティングする MTA です。単独のマシンをインバウンドリレーとアウトバウンドリレーとして使用できますが、インターネットに接続された大規模な配備では、これらの機能を 2 つの別のマシンに分割します。このようにすると、内部クライアントは、外部サイトからインバウンドするメールと競合することなく、メールを送信できます。

Delegated Administrator サーバー: 管理者に GUI 管理コンソールを提供し、管理者がユーザーの追加や削除など、より高度な管理作業を行えるようにします。

Messaging マルチプレクサ または *MMP*: ユーザーのメールボックスを含む特定のマシンと、関連付けられた DNS 名との結合を解除して、複数の物理マシンにわたるメッセージストアの拡大縮小を可能にします。クライアントソフトウェアは、メッセージストアのある物理的なマシンを知る必要はありません。このようにすると、ユーザーは、メールボックスが新しいマシンに移動されるたびにホストメッセージストアの名前を変更する必要がなくなります。POP クライアントまたは IMAP クライアントがメールボックスへのアクセスを要求すると、MMP はディレクトリサービスを参照してユーザーのメールボックスがある場所を検索し、そのメールボックスがある Messaging Server システムに要求を転送します。複数の MMP を使用する場合、それらをロードバランサの背後に配置する必要があります。

MEM (**Messenger Express** マルチプレクサ): Web メール用の HTTP アクセスサービスへの単一の接続ポイントとして機能する特別なサーバーです。すべてのユーザーがこのメッセージングプロキシサーバーに接続し、該当するメールボックスに導かれます。このため、メールユーザーには複数の Messaging Server が単一のホスト名であるかのように表示されます。Messaging Multiplexing Proxy (MMP) は POP および IMAP サーバーに接続しますが、Messenger Express マルチプレクサは HTTP サーバーに接続します。つまり、Messenger Express マルチプレクサと Messenger Express との関係は、MMP と POP や IMAP との関係と同じです。複数の MEM を使用する場合、それらをロードバランサと組み合わせて使用する必要があります。Communications Express 配備の場合、Communications Express ソフトウェアも、MEM を含む同じホストに配備されます。

2 層アーキテクチャー — メッセージングデータフロー

この節では、メッセージングシステム経由のメッセージフローについて説明します。メッセージフローがどのように機能するかは、実際のプロトコルとメッセージパス次第です。

メールの送信: 内部ユーザーから別の内部ユーザーへ

機能説明: 内部ユーザー > ロードバランサ > MTA アウトバウンドリレー 1 または 2 > MTA インバウンドリレー 1 または 2 > メッセージストア 1 または 2

注 – アウトバウンドリレーからストアにメールを直接配信させるために、LMTP を使用するのが一般的になってきています。2 層配備では、この方法を選択できます。

内部ユーザーから別の内部ユーザー (すなわち同じ電子メールシステムのユーザー) へ宛てられたメッセージは、最初にロードバランサへ送られます。ロードバランサは、電子メールユーザーを基盤となるサイトアーキテクチャーから切り離し、高可用性電子メールサービスを提供します。ロードバランサは、その接続を MTA アウトバウンドリレー 1 または 2 のいずれかに送信します。アウトバウンドリレーはアドレスを読み取り、メッセージが内部ユーザー宛てのものかどうかを判断します。アウトバウンドリレーは、MTA インバウンドリレー 1 または 2 にメッセージを送信するか、設定によっては適切なメッセージストアに直接送信します。MTA インバウンドリレーは、そのメッセージを適切なメッセージストアに配信します。メッセージストアはそのメッセージを受け取り、メールボックスに配信します。

メールの取得: 内部ユーザー

機能説明: 内部ユーザー > ロードバランサ > MMP/MEM/Communications Express Proxy Server 1 または 2 > メッセージストア 1 または 2

メールは POP、HTTP、または IMAP のいずれかを使用して取得されます。ユーザー接続がロードバランサに受信され、MMP サーバー、MEM/Communications Express サーバーのいずれかに転送されます。次にユーザーは、接続したアクセスマシンにログイン要求を送信します。アクセス層のマシンは、ログイン要求とパスワードを検証し、ユーザー接続で指定された同じプロトコルを使用して、適切なメッセージストア (1 または 2) に要求を送信します。そしてアクセス層のマシンは、クライアントとサーバー間の残りの接続を仲介します。

メールの送信: 内部ユーザーから外部 (インターネット) ユーザーへ

機能説明: 内部ユーザー > ロードバランサ > MTA アウトバウンドリレー 1 または 2 > インターネット

内部ユーザーから外部ユーザー (すなわち異なる電子メールシステムのユーザー) へ宛てられたメッセージは、ロードバランサに送られます。ロードバランサは、電子メールユーザーを基盤となるサイトアーキテクチャーから切り離し、高可用性電子メールサービスを提供します。ロードバランサは、そのメッセージを MTA アウトバウンドリレー 1 または 2 のいずれかに送信します。アウトバウンドリレーはアドレスを読み取り、メッセージが外部ユーザー宛てのものかどうかを判断します。アウトバウンドリレーは、インターネット上の MTA にメッセージを送信します。

メールの送信: 外部 (インターネット) ユーザーから内部ユーザーへ

機能説明: 外部ユーザー > MTA インバウンドリレー 1 または 2 > メッセージストア 1 または 2

外部ユーザー (インターネット) から内部ユーザーへ宛てられたメッセージは、MTA インバウンドリレー 1 または 2 へ送られます (ロードバランサは不要)。インバウンドリレーはアドレスを読み取り、メッセージが内部ユーザー宛てのものかどうかを判断します。インバウンドリレーは LDAP 検索を使用して メッセージストア 1 または 2 のいずれに送信するかを判断し、それに従って配信します。メッセージストアはそのメッセージを受け取り、適切なメールボックスに配信します。

Messaging Server における水平スケーラビリティーと垂直スケーラビリティーの理解

「スケーラビリティー」は、メッセージングサービスの利用拡大に対応する配備の能力です。スケーラビリティーにより、ユーザー数の急激な拡大をシステムがどのくらいうまく受け入れられるかが決まります。またスケーラビリティーにより、たとえば 1 か月の間にユーザーの多くが SSL の使用を希望するなどというような、ユーザーの行動の大きな変化にシステムがどのくらいうまく適応できるかも決まります。

この節では、個別のサーバーとサーバー全体で、利用の拡大に対応するためにアーキテクチャーに追加する機能について確認します。次のトピックについて説明しています。

- 160 ページの「水平的スケーラビリティーの計画」
- 164 ページの「垂直スケーラビリティーの計画」

水平的スケーラビリティーの計画

水平スケーラビリティーは、アーキテクチャーにサーバーを追加することがどの程度容易であるかを示します。ユーザー数が拡大する、またはユーザーの行動が変化するにつれて、やがては既存の配備のリソースが過負荷状態になります。慎重に計画を立てて、配備のスケールを適切に拡張する方法を決めます。

配備の水平的拡張を行う場合には、リソースを複数のサーバーに分散します。水平スケーラビリティーでは、2 つの方法が使用されます。

- 161 ページの「複数サーバーへのメッセージングユーザーベースの分散」

■ 162 ページの「冗長コンポーネントへのメッセージングリソース分散」

複数サーバーへのメッセージングユーザーベースの分散

負荷を複数のサーバーに分散するには、クライアントのメールをいくつかのバックエンドメッセージストアに均等に分割します。ユーザーをアルファベット順に分けたり、サービスのクラス別、部門、または物理的な場所別に分けたりして、特定のバックエンドメッセージストアホストに割り当てます。

MMP と MEM は、管理を容易にする目的でしばしば同じマシンに置かれます。図 11–2 に、ユーザーが複数のバックエンドサーバーに分散され、着信クライアント接続の処理にマルチプレクサを使用するサンプルアーキテクチャーを示します。

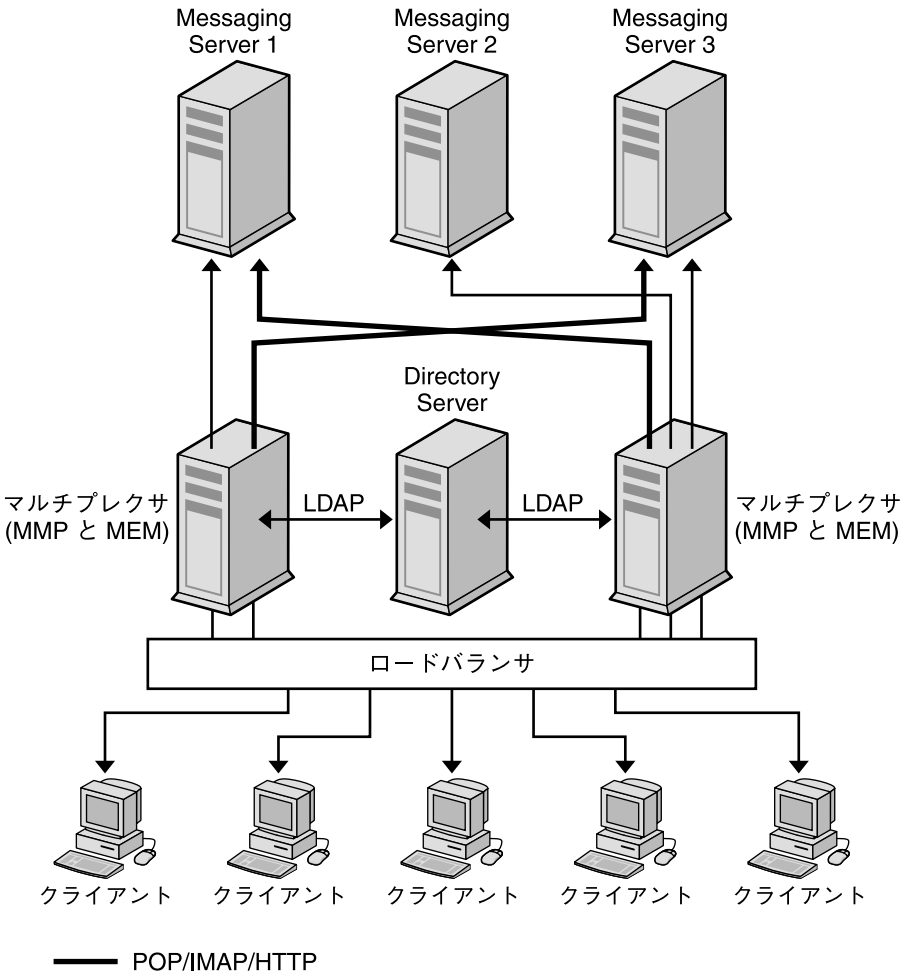

図 **11–2** 複数サーバーへのユーザーベースの分散

ユーザーをバックエンドサーバーに分散することで、MMP または MEM を使用するかぎり、ユーザーの管理が簡単になります。ユーザーは、メールがある 1 つのバックエンドサーバーに接続するため、すべてのユーザーに対して設定を標準化できます。この設定により、複数のサーバーの管理も容易になります。また、Messaging Server ホストへの要求増加に対応して、ホストをシームレスに追加できます。

冗長コンポーネントへのメッセージングリソース分散

電子メールが日常業務に重要な地位を占める場合は、メッセージングシステムが常に運用可能な状態であることを確実にするために、ロードバランサ、MX (メール交換) レコード、リレーのような冗長コンポーネントが必要になります。

冗長 MTA を使用することで、あるコンポーネントが動作不能に陥っても、別のコンポーネントが使用可能になります。また、リソースを冗長 MTA に分散することで、負荷の分散も行われます。この冗長性により、Messaging Server システムにフォールトトレランスも提供されます。それぞれの MTA リレーが、他の MTA リレーの機能を受け持ちます。

冗長ネットワーク接続をサーバーと MTA にインストールすることで、ネットワークの問題に対するフォールトトレランスが実現します。メッセージング配備が組織にとってより重要なものであるほど、フォールトトレランスと冗長性の検討もより重要になります。

163 ページの「MX (メール交換) レコード」および 163 ページの「インバウンド MTA とアウトバウンド MTA」の詳細については、次の節で説明します。

*MX (*メール交換*)* レコード

MX レコードは、1 つのホスト名を別のホスト名にマップする DNS レコードの一種です。等しい優先度の MX レコードにより、冗長化されたインバウンド MTA にメッセージがルーティングされます。たとえば、インターネットからの送信 MTA は、`siroe.com` に対する MX レコードが、`MTAA.siroe.com` および `MTAB.siroe.com` に対応していることを検出します。優先度が同じためにこれらの MTA の 1 つがランダムに選択され、SMTP 接続が開かれます。最初に選択された MTA が応答しなかった場合は、メールは別の MTA に送信されます。次の MX レコードの例を参照してください。

```
siroe.com in MX 10 MTAA.siroe.com
siroe.com in MX 10 MTAB.siroe.com
```

インバウンド *MTA* とアウトバウンド *MTA*

Messaging Server ホストがそれぞれ多数のユーザーをサポートしており、SMTP メールの送信負荷が高い場合、独立したインバウンド MTA とアウトバウンド MTA を使用することで Messaging Server ホストはルーティングタスクから開放されます。送信メッセージと受信メッセージの処理に異なる MTA を指定して、さらに負荷の分散を図ることもできます。

インバウンド MTA とアウトバウンド MTA の両方が、1 つの送受信 SMTP ホストとして組み合わされる場合もあります。1 つまたは複数の MTA ホストが必要であるかどうかを判断するには、アーキテクチャー全体のインバウンドおよびアウトバウンドメッセージのトラフィック特性を確認します。

ロードバランサ

負荷分散を使用すると、負荷を複数のサーバーに分散して、どれか 1 つのサーバーが過負荷状態にならないようにします。ロードバランサは、クライアントからの要求を受けて、各サーバーの CPU とメモリーの使用率を追跡するようなアルゴリズムを使用して、利用可能なサーバーに要求をリダイレクトします。ロードバランサは、共通サーバーで実行されるソフトウェアとして、純粋に外部のハードウェアソリューションとして、またはハードウェアとソフトウェアを組み合わせたパッケージとして使用可能です。

垂直スケーラビリティーの計画

垂直スケーラビリティーは、CPU の追加など、個々のサーバーマシンへのリソースの追加に関係があります。それぞれのマシンには、一定の負荷を処理できる能力があります。一般には、リソースに制限があるか、配備の拡大に応じて追加のハードウェアを購入できない場合に、配備における垂直スケーラビリティーを検討します。

配備の垂直的スケールを行うには、次のことが必要です。

- 各メッセージングコンポーネントのサイズを決定する

  150 ページの「Messaging Server アーキテクチャー戦略の構築」を参照してください。
- システムのプロトタイプの負荷をテストする

  145 ページの「Messenger Express 負荷シミュレータの使用」を参照してください。
- システムパフォーマンスを監視し、それに従って配備を調整する

高可用性の Messaging Server 配備の計画

高可用性は、計画された停止時間と予期しない停止時間を短時間にとどめるための配備の設計です。通常、高可用性設定は緩やかに結合された 2 つ以上のシステムで構成されたクラスタです。各システムがそれぞれのプロセッサ、メモリー、オペレーティングシステムを維持しています。ストレージはシステム間で共有されます。特別なソフトウェアがシステムをバインドし、単一点での障害からシステムが完全に自動的に回復できるようにします。Messaging Server には、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のオプションが用意されています。

高可用性に向けた計画を作成する場合は、可用性とコストとのバランスを検討する必要があります。一般に、より可用性の高い配備では、設計と運用のコストも高くなります。

高可用性は、アプリケーションサービスの中断や停止時間によるデータアクセス機会の損失に対する保険です。アプリケーションサービスが利用不能になった場合、組織は収入、顧客、その他の機会を失うことになります。組織にとっての高可用性の価値は、停止時間のコストに直接関係します。停止時間のコストが高くなるほど、高可用性のための追加コストを正当化するのも容易になります。また、一定のレベルの可用性を保証するサービスレベル契約を組織が結んでいる場合もあります。可用性の目標を達成できない場合、財務的な打撃を直接受ける可能性があります。

詳細については、第 6 章を参照してください。

Messaging Server アーキテクチャーのパフォーマンスの考慮事項

この節では、Messaging Server コンポーネントのパフォーマンス特性を評価して、的確にアーキテクチャーを開発する方法について説明します。

この節では次の項目について説明します。

メッセージストアのパフォーマンスの考慮事項

メッセージストアのパフォーマンスは、次のようなさまざまな要素に影響を受けます。

1. ディスク入出力
2. インバウンドメッセージレート (メッセージ挿入レートとも呼ばれる)
3. メッセージサイズ
4. ログインレート (POP/IMAP/HTTP)
5. IMAP および HTTP のトランザクションレート
6. さまざまなプロトコルの並行接続数
7. ネットワーク入出力
8. SSL の使用

前の要素リストは、メッセージストアに影響を与えるおおよその順序で記載されています。メッセージストレージに関するパフォーマンス問題のほとんどは、ディスクの入出力能力が不十分なことに原因があります。さらに、物理ディスク上のストアのレイアウトもパフォーマンスに影響を与えます。より小規模のスタンドアロンシステムでは、単純なディスクのストライピングでも十分な入出力が得られます。ほとんどの大規模システムでは、ファイルシステムを分離し、ストアのさまざまな部分に入出力を提供します。

メッセージングサーバーのディレクトリ

Messaging Server は 6 つのディレクトリを使用して大量の入出力活動に対応しています。これらのディレクトリは高頻度でアクセスされるため、ディレクトリごとにディスクを用意するか、より理想的には、ディレクトリごとに RAID を用意します。次の表で、これらのディレクトリについて説明します。

表 **11–1** アクセス頻度の高い Messaging Server ディレクトリ

高入出力ディレクトリ	説明とパラメータの定義
MTA キューディレクトリ	このディレクトリでは、MTA チャネルを通る各メッセージについて 1 つずつのファイルが、大量に作成されます。ファイルが次の目的地に送信されると、そのファイルは削除されます。ディレクトリの場所は、`imta_tailor` ファイルの `IMTA_QUEUE` オプションにより制御されます。MTA キューディレクトリを変更する前に、『Sun Java System Messaging Server 6 2005Q4 Administration Reference』にあるこのオプションについての説明を参照してください。 デフォルトの場所: `/var/opt/SUNWmsgrsr/queue`
Messaging Server ログディレクトリ	このディレクトリには、新しいログ情報が常に追加されるログファイルがあります。変更の回数は、ログレベルの設定によります。ディレクトリの場所は、`configutil` のパラメータ `logfile.*.logdir` によって制御されます。ここで「*」は、admin、default、http、imap、pop など、ログを生成するコンポーネントを示します。MTA ログファイルは `imta_tailor` ファイルの IMTA_LOG オプションを使用して変更できます。 デフォルトの場所: `/var/opt/SUNWmsgsr/log`
メールボックスデータベースファイル	これらのファイルはキャッシュの同期と継続的な更新を必要とします。このディレクトリは最も高速なディスクボリュームに配置します。これらのファイルは常に `/var/opt/SUNWmsgsr/store/mboxlist` ディレクトリに格納されます。
メッセージストアインデックスファイル	これらのファイルにはメールボックス、メッセージ、ユーザーに関するメタ情報が含まれます。デフォルトではこれらのファイルはメッセージファイルと共に格納されます。`configutil` パラメータの `store.partition.*.path` はディレクトリの場所を制御します。ここで「*」はパーティション名です。リソースに余裕がある場合は、これらのファイルを 2 番目に高速なディスクボリュームに配置します。 デフォルトの場所: `/var/opt/SUNWmsgsr/store/partition/primary`
メッセージファイル	これらのファイルにはメッセージが含まれており、メッセージごとに 1 つのファイルとなっています。ファイルは頻繁に作成され、変更されることはなく、最終的には削除されます。デフォルトでは、これらのファイルはメッセージストアインデックスファイルと同じディレクトリに格納されます。ディレクトリの場所は、`configutil` のパラメータ `store.partition.` *partition_name* `.messagepath` で制御されます。ここで *partition_name* はパーティション名です。 サイトによっては、`store.partition.primary.path` によって指定される、`primary` と呼ばれる単独のメッセージストアパーティションがあります。大規模なサイトの中には、`store.partition.` *partition_name* `.messagepath` により指定される追加パーティションを持つものもあります。ここで *partition_name* はパーティション名です。 デフォルトの場所: `/var/opt/SUNWmsgsr/store/partition/primary`

表 **11–1** アクセス頻度の高い Messaging Server ディレクトリ　(続き)

高入出力ディレクトリ	説明とパラメータの定義
メールボックスリスト データベース一時ディレクトリ	すべての一時ファイル格納用としてメッセージストアによって使用されるディレクトリ。パフォーマンスを最大化するには、このディレクトリを最も高速なファイルシステムに配置する必要があります。Solaris の場合は、`configutil` コマンドを使用して `tmpfs` の下のディレクトリ (たとえば、 `/tmp/mboxlist`) に `store.dbtmpdir` 変数を設定します。 デフォルトの場所: `/var/opt/SUNWmsgsr/store/mboxlist`

次の節では、Messaging Server の高頻度アクセスディレクトリについてさらに詳しく説明します。

MTA キューディレクトリ

LMTP 以外の環境では、メッセージストアシステム内の MTA キューディレクトリもかなりの頻度で使用されます。LMTP は、インバウンドメッセージが MTA キューに置かれず、ストアに直接挿入されるように機能します。このメッセージの挿入により、メッセージストアマシンの全体的な入出力要件が少なくなり、メッセージストアマシンの MTA キューディレクトリの使用頻度が大きく減少します。システムがスタンドアロンの場合、または Web メール 送信のためのローカル MTA を使用する場合は、アウトバウンドメールトラフィックのためのこのディレクトリに、まだかなりの入出力が発生します。LMTP を使用した 2 層環境では、このディレクトリが使用されることがあったとしても、ごくまれです。Messaging Server の前のバージョンでは、大規模なシステムではこのディレクトリをそれ自身のストライプまたはボリューム上に設定する必要がありました。

MTA キューディレクトリは通常、専用のファイルシステム上に配置し、メッセージストア内のメッセージファイルから分離すべきです。メッセージストアには、ディスク容量がある定義済みのしきい値を下回った場合にメッセージの配信と追加を停止するメカニズムが備わっています。しかしながら、ログディレクトリとキューディレクトリがどちらも同一ファイルシステム上に存在しており、かつそれらのサイズが増大し続けた場合、ディスク容量不足によりメッセージストアの動作が停止する可能性があります。

ログファイルディレクトリ

ログファイルディレクトリでは、設定されているログのレベルにより、さまざまな量の入出力が要求されます。メッセージストアのその他の高入出力要求とは異なり、ログディレクトリへの入出力は非同期です。典型的な配備シナリオでは、LUN 全体をログ専用には使用しません。かなり規模の大きなストア配備、または大量のログが必要な環境では、専用の LUN を使用するのが理に適っています。

ほとんどすべての環境で、メッセージストアをデータ喪失から守る必要があります。要求される損失からの保護と継続的な可用性のレベルは、RAID5 のような単純なディスク保護から、ミラーリング、日常的なバックアップ、データのリアルタイムレ

プリケーション、リモートデータセンターまで、さまざまです。データの保護に関しても、Automatic System Recovery (ASR) が可能なマシンから、ローカル HA 機能、自動リモートサイトフェイルオーバーまで、さまざまなものがあります。これらの決定は、ハードウェアの量とサービスの提供に必要なサポート要員の数に影響します。

mboxlist ディレクトリ

`mboxlist` ディレクトリには入出力が非常に集中しますが、特にサイズが大きいというわけではありません。`mboxlist` ディレクトリには、ストアとトランザクションログで使用されるデータベースがあります。高頻度の入出力があり、データベースを構成する複数のファイルを複数のファイルシステム間で分割できないことから、大規模な配備では `mboxlist` ディレクトリをそれ自身のストライプかボリューム上に配置する必要があります。これは、メッセージストアの多くの操作がデータベースにアクセスするため、垂直的スケーラビリティーの喪失の原因にもつながります。アクセスが激しいシステムでは、これがボトルネックになります。`mboxlist` ディレクトリの入出力パフォーマンスのボトルネックによって、ストアの raw パフォーマンスと応答時間が悪くなるだけでなく、垂直的スケーラビリティーも減少します。バックアップから高速に復旧することが要求されるシステムでは、このディレクトリを Solid State Disks (SSD) 上に配置するか、パフォーマンスの高いキャッシングアレイを使って、ファイルシステム上でサービスを継続したまま復元処理を進行できるような高い書き込みレートを許可します。

複数のストアパーティション

メッセージストアは、複数のストアパーティションをサポートしています。各パーティションを、それ自身のストライプまたはボリューム上に配置します。ストア上に配置するパーティションの数は、さまざまな要素により決定されます。明確な要素としては、サーバーのピーク負荷時の入出力要件があります。追加のストアパーティションとしてファイルシステムを追加することで、メールの配信や取得のためにサーバーで可能な IOPS (1 秒あたりの総入出力) を引き上げます。ほとんどの環境で、大きくて数が少ないストライプあるいは LUN よりも、多数の小さなストライプあるいは LUN のほうが、より大きな IOPS が得られます。

いくつかのディスクアレイを使用すると、アレイを 2 つの異なる方法で設定できます。それぞれのアレイを LUN として設定し、それをファイルシステムにマウントします。または、それぞれのアレイを LUN として設定し、それをサーバー上でストライプします。どちらも有効な設定です。ただし、複数のストアパーティション (小さいアレイでは 1 つのパーティション、または LUN のストライプセットをサーバーボリュームにした大きなアレイ上の多数のパーティション) は最適化と管理が容易です。

ただし、通常は raw パフォーマンスは、ストアパーティションの数を決定する場合の優先事項とはなりません。企業環境では、IOPS よりも容量のほうが重要となる場合が多いでしょう。また、LUN をソフトウェアストライプで設定し、1 つの大きなストアパーティションとすることも可能です。ただし、複数の小さなパーティションのほうが、一般に管理は容易です。ストアパーティションの数を決定する際に適切な最優先事項は、一般的には回復時間です。

ストアパーティションの回復時間は、いくつかのカテゴリに分類されます。

- 最初に、電源、ハードウェア、またはオペレーティングシステムの障害によるクラッシュからの回復と並行して、`fsck` コマンドが複数のファイルシステム上で動作します。HA プラットフォームで強く推奨され、必須となっているジャーナリングファイルシステムを使用している場合は、この要素は小さなものとなります。
- 次に、バックアップおよび回復手順が複数のストアパーティション上で並行して実行されます。メッセージストアではすべてのストアパーティションで単独のデータベースが使用されているため、この並行動作は `mboxlist` ディレクトリの垂直的スケーラビリティーにより制限されます。ストアパーティションあたりの 1 つのスレッドの実行と並行して、ストアクリーンアップ手順 (`expire` および `purge`) が実行されます。
- 最後に、ミラーリングまたは RAID 再同期手順が、小さな LUN で高速に実行されます。ここでは厳密な規則はありませんが、ほとんどの場合はストアパーティションを構成するスピンドルを 10 個までにすることをお勧めします。

ストレージアレイで使用されるドライブのサイズは、容量要件に対する IOPS 要件という問題になります。ほとんどの家庭用 ISP POP 環境では、「より小さなドライブ」を使用します。大規模な割り当てによる企業配備では、「より大きな」ドライブを使用します。繰り返しになりますが、すべての配備は異なっており、一連の要件を個別に検討する必要があります。

メッセージストアのプロセッサスケーラビリティー

マルチプロセスとマルチスレッドにより、メッセージストアは良好なスケール化がなされています。実際には、メッセージストアは 1 つのプロセッサから 4 つのプロセッサまで、一次直線形の比率を上回るスケール化が行われています。これは、4 つのプロセッサシステムは、1 つのプロセッサシステムを 4 つ合わせたものよりも大きな負荷を処理できることを意味します。メッセージストアは 4 から 12 のプロセッサ数についてもかなり直線形でスケール化されます。12 から 16 のプロセッサ数では、能力は増強されますが、直線形ではなくなります。LMTP を使用すると、同じサイズのストアシステムでサポートされるユーザー数は大きく増加しますが、メッセージストアの垂直的スケーラビリティーはより制限されます。

メールボックスデータベースキャッシュサイズの設定

Messaging Server は、メールボックスデータベースの呼び出しを頻繁に行います。 そのため、そのデータができるだけ迅速に返されることが重要です。メールボックスデータベースの部分をキャッシュ化すると、メッセージストアのパフォーマンスが改善されます。最適なキャッシュサイズを設定することで、メッセージストア全体のパフォーマンスを大きく向上させることができます。キャッシュのサイズは、`configutil` のパラメータ `store.dbcachesize` を使用して設定します。

メールボックスデータベースの場所を `/tmp`、つまり `/tmp/mboxlist` に定義し直すには、`configutil` のパラメータ `store.dbtmpdir` の使用をお勧めします。

メールボックスデータベースは、データページに格納されます。さまざまなデーモンにより stored、imapd、 popd などのデータベースが呼び出されると、指定されたページがキャッシュに格納されているかどうかが、システムによりチェックされます。ページがキャッシュ内に存在する場合は、それがデーモンに渡されます。存在しない場合は、システムは 1 ページをキャッシュからディスクに書き戻し、指定されたページを読み込んでそれをキャッシュに書き込む必要があります。ディスクの書き込みと読み取り回数を減らすことはパフォーマンスの向上につながりますが、それだけに、キャッシュサイズを最適に設定することが重要となります。

キャッシュサイズが小さすぎる場合は、指定されたデータをディスクから必要以上の頻度で読み込む必要があります。キャッシュサイズが大きすぎる場合は、ダイナミックメモリー (RAM) が浪費され、ディスクとキャッシュの同期に余計な時間がかかります。これら 2 つの状況の中では、キャッシュが大きすぎる場合よりも小さすぎる場合の方が、より大きなパフォーマンスの低下を招きます。

キャッシュ効率は、「ヒットレート」により測定されます。ヒットレートは、データベースがキャッシュにより処理される回数の割合のことです。最適化されたサイズのキャッシュでは、ヒットレートは 99 パーセントに達します。すなわち、要求されたデータベースページの 99 パーセントが、ディスクから取得されることなくデーモンに返されます。要求されたデータの 95 パーセント以上を返せるページ数をキャッシュが保持することを目標にしてキャッシュを設定します。キャッシュから返されるページが 95 パーセント未満の場合は、キャッシュサイズを大きくする必要があります。

キャッシュのヒットレートは、データベースコマンド db_stat を使用して測定できます。次の例では、configutil のパラメータ store.dbtmpdir を使用して、メールボックスデータベースの場所を /tmp、つまり /tmp/mboxlist に定義し直しています。db_stat コマンドは、次の場所に対して実行されます。

```
# /opt/SUNWmsgsr/lib/db_stat -m -h /tmp/mboxlist

2MB 513KB 604B  Total cache size.
1                   Number of caches.
2MB 520KB           Pool individual cache size.
0                   Requested pages mapped into the process' address space.
55339               Requested pages found in the cache (99%).
```

この例では、ヒットレートは 99 パーセントです。これは、キャッシュサイズが最適であるか、大きすぎることを示します。これをテストするには、ヒットレートが 99 パーセント以下になるまでキャッシュサイズを小さくしていきます。ヒットレートが 98 パーセントになったら、データベースキャッシュサイズが最適化されたことを意味します。逆に、db_stat が 95 パーセント未満のヒットレートを示した場合は、store.dbcachesize パラメータを使用してキャッシュサイズを大きくします。最大サイズは、store/mboxlist ディレクトリ内のすべての *.db ファイルを合計したものになります。キャッシュサイズは、store/mboxlist ディレクトリに格納されるすべての .db ファイルの合計サイズを超えてはいけません。

注 – ユーザーベースが変化すると、ヒットレートも変化します。このパラメータを定期的にチェックして、必要に応じて調整します。このパラメータの上限はデータベースの制約による 2G バイトです。

ディスクストライプ幅の設定

ディスクストライピングを設定するときには、システムを通過するメッセージの平均サイズにストライプ幅を合わせます。128 ブロックのストライプ幅は、通常の使用には大きすぎて、パフォーマンスに悪影響を与えます。代わりに、8、16、32 ブロック (それぞれ 4、8、16K バイトのメッセージサイズの場合) の値を使用します。

MTA パフォーマンスの考慮事項

MTA のパフォーマンスは、次の項目を含む多くの要素に影響されます。

- ディスクパフォーマンス
- SSL の使用
- インバウンドおよびアウトバウンドのメッセージ数および接続数
- メッセージのサイズ
- 対象送信先数およびメッセージ数
- MTA との接続スピードと接続待ち時間
- スパムフィルタリングまたはウィルスフィルタリングの必要性
- SIEVE 規則とその他のメッセージ解析 (変換チャネルの使用など) の使用

MTA は CPU と入出力を集中的に使用します。MTA は、キューディレクトリとロギングディレクトリという異なる 2 つのディレクトリに対し、読み書きを行います。MTA として機能する小規模なホスト (4 プロセッサ以下) では、これらのディレクトリを別のファイルシステムに分ける必要はありません。キューディレクトリでは、かなり大きい量で同期書き込みが行われます。ログディレクトリでは、小さな量の非同期書き込みが連続的に行われます。トラフィック量の多いシステム上では、これら 2 つのディレクトリを分離し、それぞれ異なるファイルシステム上に配置することを検討してください。

ほとんどのケースで、ディスクサブシステムの MTA で冗長性を導入して、ディスクの障害時にメールデータが永久に失われることを回避したいと考えるでしょう。ディスクの障害は、ハードウェアの障害で最も起こる可能性の高いものです。これは、多くの内部ディスクを持つ外部ディスクアレイやシステムが最適だということを意味します。

MTA と RAID のトレードオフ

外部 RAID コントローラデバイスとソフトウェアミラーによる JBOD アレイの使用との間にはトレードオフの関係があります。JBOD によるアプローチは、ハードウェアの購入という点では安価な場合がありますが、より多くのラックスペースと電力を必

要とします。JBOD アプローチは、ソフトウェアによるミラーリングを行うことでサーバーのパフォーマンスを少し低下させ、一般的には保守コストも高くなります。ソフトウェア RAID5 は、パフォーマンスへの影響が非常に大きいため、代わりに使うことができません。そのため、RAID5 を使用する場合は、RAID5 キャッシングコントローラアレイを使用します。

MTA とプロセッサスケーラビリティー

MTA の処理能力は 8 プロセッサを超えても直線的に向上します。また、メッセージストアと同様に、1 プロセッサから 4 プロセッサまでは飛躍的にアップします。

MTA と高可用性

MTA を HA の制御のもとに置くのはあまりお勧めできません。しかし、それが保証されている環境では例外です。ハードウェアの障害時にも、メールの配信を短時間で指定した時間枠内で実行しなければならないという要件がある場合は、MTA を HA のソフトウェア制御のもとに配置します。ほとんどの環境では、ピーク負荷要件に対応できるように MTA の数を単純にいくつか増やします。これにより、1 つの MTA で障害が発生した場合でも、または大規模な配備環境で何らかの理由で複数の MTA の接続が遮断された場合でも、適切なトラフィックフローが生み出されます。

さらに、MTA の配置に関しては、MTA を常にファイアウォールの内側に配置するよう配慮します。

MMP パフォーマンスの考慮事項

MMP は、マルチスレッドの単一プロセスとして動作し、CPU とネットワークに強く依存します。MMP がディスクリソースを使用するのは、ロギング時だけです。MMP のスケーラビリティーは、2 プロセッサマシンでもっとも効率がよく、2 プロセッサから 4 プロセッサまでは直線形を下回る比率になり、4 プロセッサを超えると大きく低下します。MMP には、2 つのプロセッサを備えたラックマウントのマシンが適しています。

その他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 コンテナ、LDAP プロキシなど) を MMP と同じマシンに配置する配備の場合は、大型の 4 プロセッサ SPARC マシンの配備を検討します。そのような構成を行うことにより、管理、パッチの導入、監視などが必要なマシンの総数を減らすことができます。

MMP のサイズは、接続レートとトランザクションレートにより決まります。POP のサイズ決定は、POP 接続がほとんどアイドル状態にならないため、きわめて明快です。POP 接続では、接続が行われ、作業が行われ、そして接続が遮断されます。IMAP のサイズ決定はより複雑です。IMAP では、ログインレート、並行レート、接続のビジー状態の起こり方について確認する必要があります。MMP も、接続の待ち時間と帯域幅に多少影響を受けます。MMP はメッセージストアからクライアントに送信されるデータのバッファとして機能するため、ダイアルアップ環境では、ブロードバンド環境の場合よりも並行して処理できるユーザーの数が少なくなります。

SSL の使用率が接続のかなりの割合を占める場合は、ハードウェアアクセラレータをインストールします。

MMP と高可用性

決して MMP を HA の制御のもとに配備しないでください。個別の MMP には静的データはありません。可用性の高い環境では、1 つ以上の MMP マシンを追加して、1 つ以上の MMP が停止してもピーク負荷に対して十分な能力を確保します。Sun Fire Blade™ Server ハードウェアを使用する場合は、Blade ラックユニット全体が停止する可能性を考慮して、適切な冗長性の配備を計画します。

MEM パフォーマンスの考慮事項

MEM では、Web メール (Messenger Express) クライアントに対して中間層プロキシが提供されます。このクライアントを使用して、ユーザーはブラウザを通じてメールにアクセスし、メールを作成できます。MEM のメリットは、メールを格納しているのはバックエンドサーバーであるにもかかわらず、エンドユーザーは MEM にだけ接続して、自分の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 情報を通じて HTTP セッション情報とユーザープロファイルを管理することで、この機能を実現しています。2 番目のメリットは、すべての静的ファイルと LDAP 認証の状態が Messaging Server のフロントエンドに存在することです。このメリットにより、メッセージストアバックエンドからの Web ページレンダリングに関連した、CPU の追加要件が相殺されます。

MMP と MEM は同じサーバーセット上に配置できます。そうすることのメリットとして、少数の MMP または MEM が必要な場合に、冗長性確保のために必要なハードウェアの追加を最小限に抑えることができます。MMP と MEM を同じサーバーセット上に配置することで生じる唯一のデメリットの可能性は、1 つのプロトコルに対するサービス拒否攻撃が別のプロトコルにも影響を与えることです。

Messaging Server と Directory Server のパフォーマンスの考慮事項

Access Manager、Messaging Server、および LDAP スキーマ 2 ディレクトリを使用した大規模なインストールでは、使用するディレクトリに ACI (アクセス制御命令) を統合した方がよい場合があります。

Messaging Server を使用して Access Manager をインストールするときには、多数の ACI がディレクトリに最初にインストールされます。デフォルトの ACI の多くは、Messaging Server では不要であり使用されません。ディレクトリ内のデフォルト ACI を統合して数を減らすことにより、Directory Server のパフォーマンス、ひいては Messaging Server ルックアップのパフォーマンスを向上させることができます。

使用されていない ACI を統合および破棄する方法については、『Sun Java System Communications Services 6 2005Q4 Delegated Administrator 管理ガイド』の付録 E「Directory Server パフォーマンスのための ACI 統合」を参照してください。

第 12 章

Messaging Server トポロジの設計

この章では、「メッセージングトポロジ」の設計方法について説明します。メッセージングトポロジは、ネットワーク化されたメッセージングシステムの物理的および論理的なレイアウトを示すものです。とくに、トポロジは、ネットワーク上でデバイスがどのように配置され、互いにどのようにやり取りするかを示します。さらに、ネットワークを経由してデータを配信する方法も示します。トポロジは、データフローを規定するネットワークプロトコルに結びつけられています。

この章には、次の節があります。

地理的ニーズの理解

メッセージングトポロジ設計の最初のステップは、地理的ニーズを確認することです。特に、組織内のそれぞれの場所に必要なメッセージングサービスを決定します。

1. 配備の目標を確認したら、次に配備内のそれぞれの場所に必要な機能を決定します。
2. 組織の物理的な制約、特に次の項目について理解します。
   - 使用可能な帯域幅
   - 組織内の物理的な場所間の距離
   - それぞれの物理的な場所におけるメールトランザクションレートとメールストレージの量

メッセージングトポロジの設計

トポロジを開発する前に、組織内のどこにメッセージングサービスを配置するかを決定するための戦略が必要です。目標により、組織に適用可能なトポロジには次の 4 つがあります。

- 176 ページの「集中トポロジ」

  ほとんど、またはすべての主要システムコンポーネントとメッセージングサーバーを 1 箇所で一元管理します。
- 178 ページの「分散トポロジ」

  ほとんどまたはすべてのシステムコンポーネントとメッセージングサーバーを複数のサイトに分散します。
- 180 ページの「ハイブリッドトポロジ」

  一部のシステムコンポーネントを一元管理し、その他のコンポーネントは複数箇所に分散します。
- 182 ページの「サービスプロバイダトポロジ」

  複数のドメインをホストして、より大きなカスタマベースを処理します。集中トポロジと同様に、ほとんどのシステムコンポーネントを 1 箇所で一元管理します。

集中トポロジ

集中トポロジでは、ほとんどまたはすべての、主要なシステムコンポーネントおよびメッセージングプロセスを 1 つのサイトに配置します。リモートサイトのクライアントは、Wide Area Network (WAN) により中央メッセージングサーバーと通信を行います。図 12–1 は集中トポロジを示します。

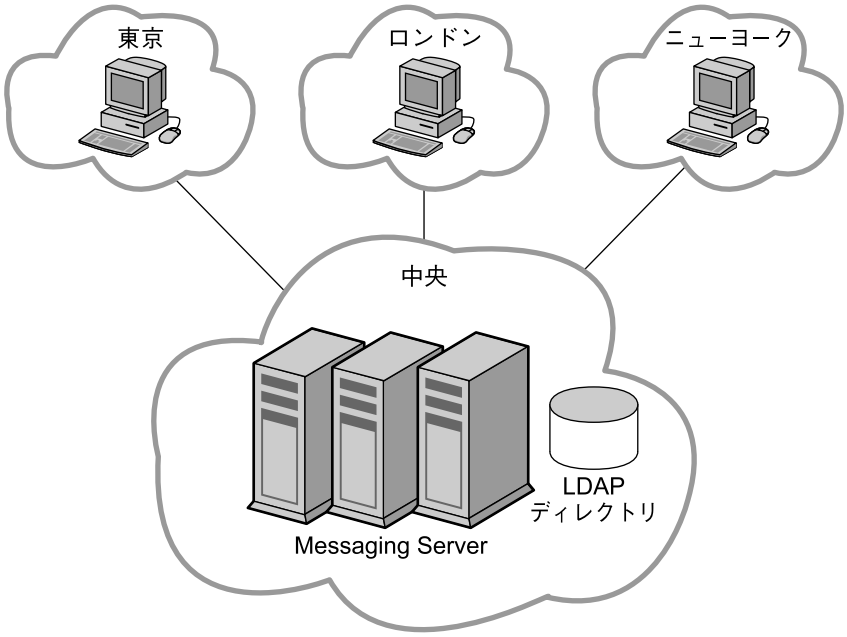

図 **12–1** 集中トポロジ

次のような場合に、集中トポロジの導入を検討します。

- リモートサイトでのメッセージングがミッションクリティカルなものではない。
- 小さなサイズのテキストメッセージの送受信を行うユーザーが多い。
- 組織が 1 つの物理的な場所にあるか、または小人数のユーザーが複数の場所に分散している。
- リモートサイトのサポート要員がいない。
- リモートサイトと中央サイト間で、少なくとも ISDN 以上の良質な帯域幅が使用可能。

集中トポロジの導入にはいくつかのメリットがあります。一般に、集中トポロジでは、ハードウェアとサポートのコストが低くなります。単純なメッセージングアーキテクチャーと少数のレプリカ契約によるディレクトリレプリカ構造のため、集中トポロジにすると管理がより容易になります。単純なアーキテクチャーと地理的に離れたサイト間でインストールを調整する必要がないため、集中トポロジでは迅速な配備が可能です。

ただし、集中トポロジの実施にはメリットと等しくデメリットもあります。集中化アプローチは WAN に大きく依存しています。ネットワークが正しく機能しなくなると、同じサイトのユーザーもリモートサイトのユーザーも、共に電子メールの送信ができなくなります。ネットワークの帯域幅とトラフィックにより、使用率がピークに

達したときはサービスの処理が遅くなる場合があります。同じドメイン内にメッセージを送信するユーザーにとって、集中トポロジは非効率的となります。たとえば、図 12–1 では、東京サイトのあるユーザーが送信したメッセージは、同じ東京サイトの別のユーザーに配信される前にまず中央サイトに送られます。

分散トポロジ

分散トポロジでは、ほとんどまたはすべてのシステムコンポーネントとメッセージングプロセスを、複数のサイト (通常は各リモートサイト) に分散配置します。図 12–2 は分散トポロジを示します。

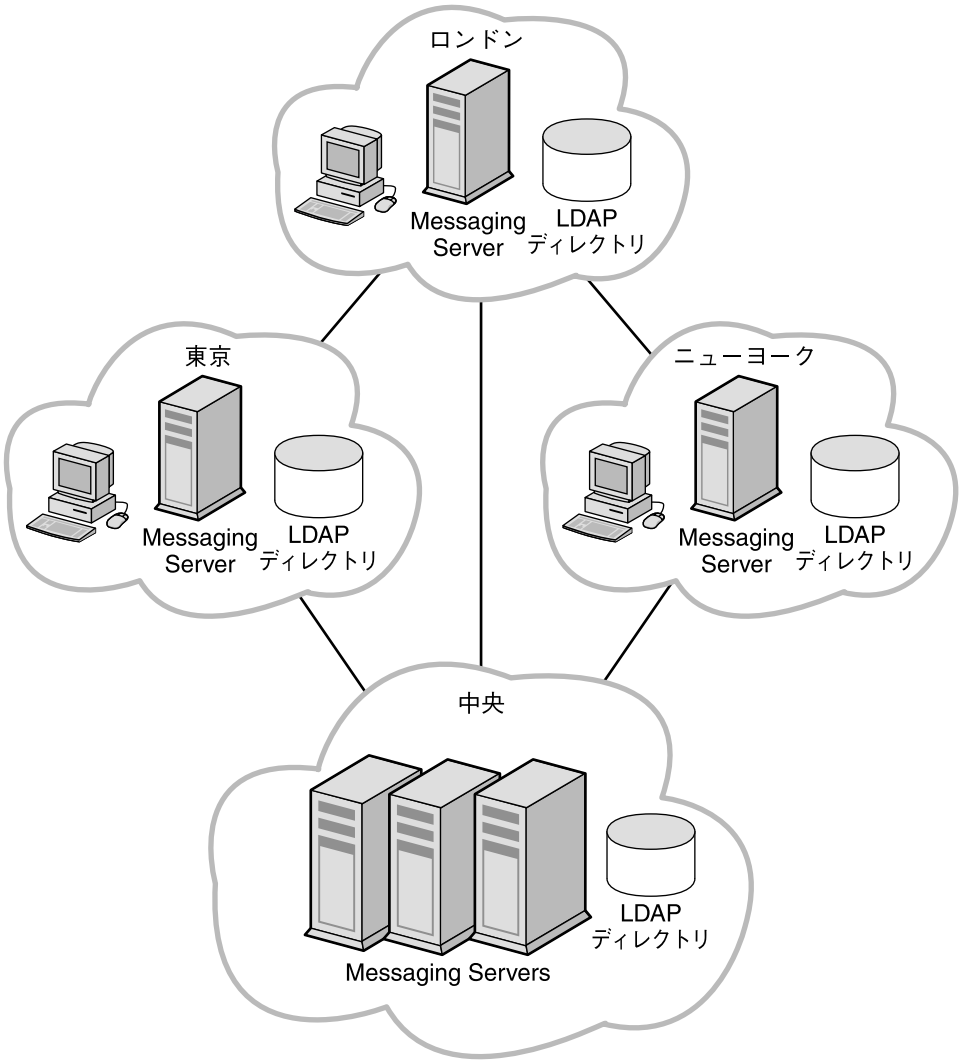

図 **12–2** 分散トポロジ

次のような場合には、分散トポロジの導入を検討することをお勧めします。

- リモートサイトでのメッセージングがミッションクリティカルなものである。
- ユーザーが大量のメッセージの送受信を行う。
- リモートサイトに大量のユーザーを抱えている。
- リモートサイトにサポート要員がいる。
- リモートサイトへの帯域幅が貧弱。

帯域幅がトポロジ戦略に大きな影響を及ぼす場合は、帯域幅のアップグレードを検討します。一般に、帯域幅は比較的安価です。Virtual Private Networking (VPN) の導入についても検討します。VPN ではファイアウォールで保護された専用線ではなく、既存の広帯域幅インターネット網を使用します。

分散トポロジの導入にはいくつかのメリットがあります。メッセージを WAN 経由で取得する必要がないため、地域サイトのユーザーはメッセージに迅速にアクセスできます。さらに、特定地域内で送信されるメッセージに起因するメッセージングトラフィックは、集中トポロジの場合よりも少なくなります。ただし、遠隔オフィスは WAN に依存します。したがって、大量のメッセージトラフィックが遠隔オフィスで生成される場合、WAN をアップグレードする必要が出てきます。

分散トポロジを導入することのデメリットは、多くの場所で多くのハードウェアを保守しなければならないため、一般にハードウェアとサポートのコストが高くなることです。分散トポロジは複雑なため、サポートのコストも高くなります。たとえば、分散トポロジにおけるフェイルオーバーは、集中トポロジの場合よりも難しくなります。さらに、複数のサーバーを複数のサイトに分散するため、Messaging Server の初期配備に時間がかかります。

Messaging Server は LDAP ディレクトリにアクセスするため、メール配信処理においては、LDAP サーバーへの接続が不可欠となります。リモートの LDAP レプリカを使用しない場合、中央の LDAP がダウンすると、メッセージングサービスが使用できなくなります。

ハイブリッドトポロジ

ハイブリッドトポロジでは、集中トポロジと分散トポロジを組み合わせて、組織のニーズを満たします。図 12–3 はハイブリッドトポロジを示します。

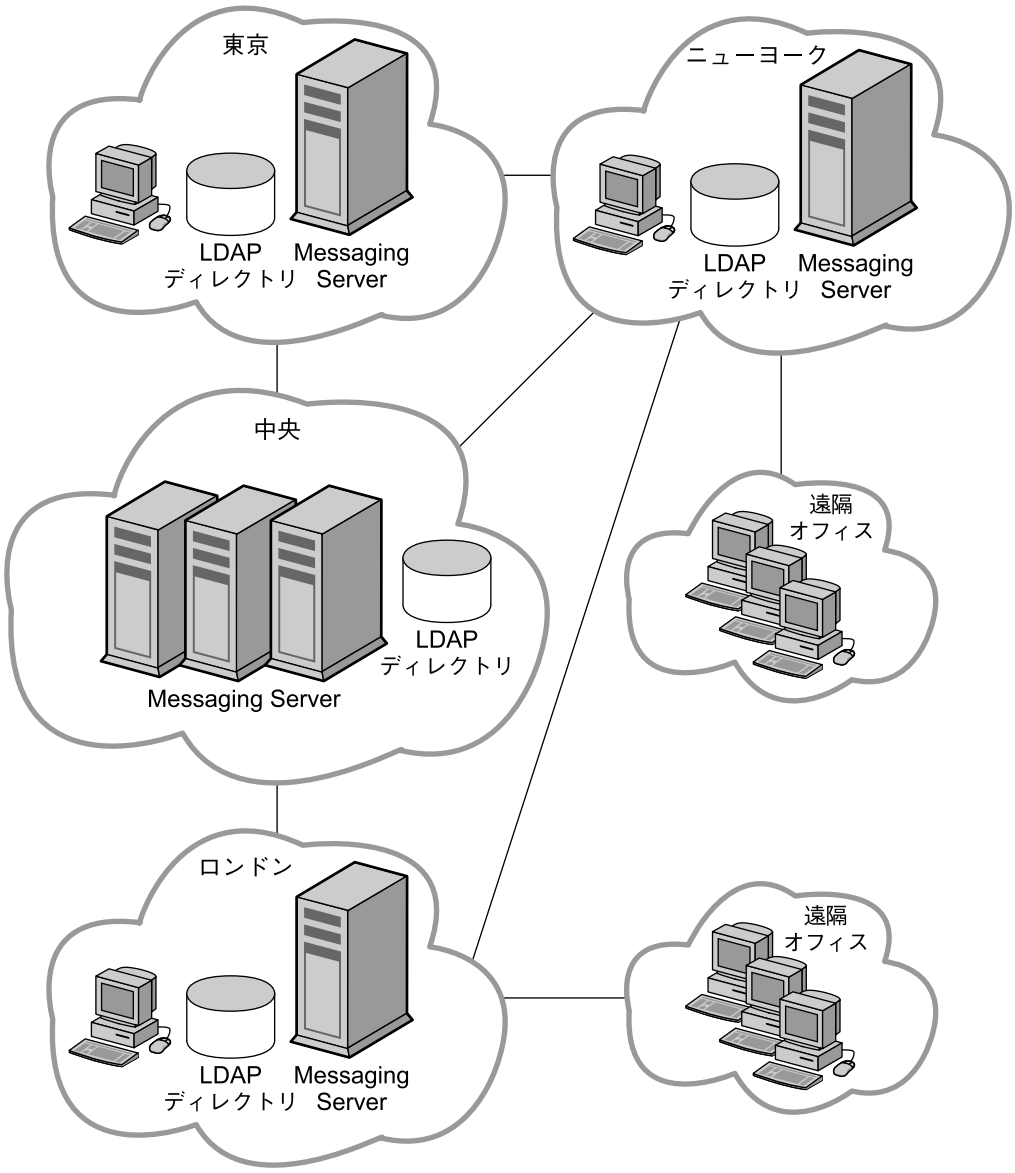

図 **12–3** ハイブリッドトポロジ

ハイブリッドトポロジからメリットを得られる組織として、大規模なユーザーベースをサポートできるサイトを数多く持つ組織があげられます。大規模なユーザーベースをサポートするサイトは、メッセージングサーバーを独自に保有できます。これらの大規模なサイトには、その近くに小規模な遠隔オフィスを持つ場合もあります。ただし、これらの遠隔オフィスには固有のメッセージングサーバーは必要ありません。代わりに最寄りの主要オフィスが、遠隔オフィスのためのサービスの中央ロケーションとして機能します。

サービスプロバイダトポロジ

サービスプロバイダトポロジは、本質的には大規模な集中トポロジです。通常、サービスプロバイダは複数のドメインをホストしており、企業よりも大規模なカスタマベースを抱えています。システムは集中化されており、ピーク時でも複数のユーザーをサポートする能力があります。図 12–4 はサービスプロバイダトポロジを示します。

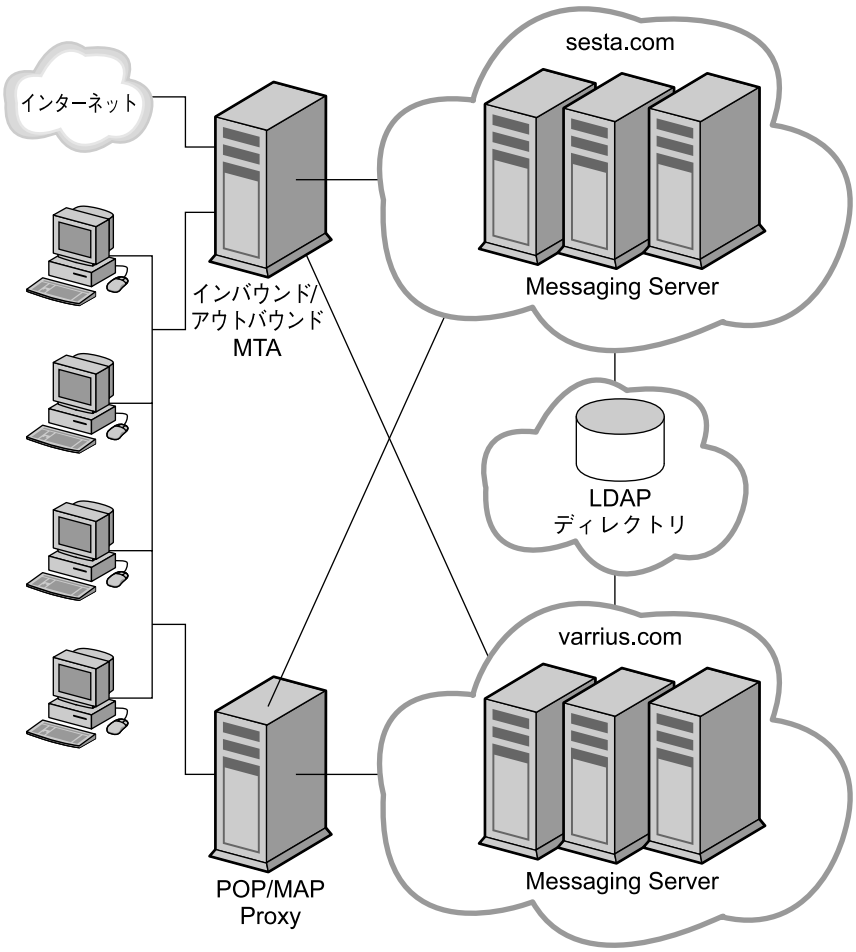

図 **12–4** サービスプロバイダトポロジ

メッセージングトポロジ要素の理解

この節では、メッセージングトポロジにおける最も一般的な要素について説明します。基本的な要素について理解を深めることで、独自のトポロジの設計が容易になります。

次のトピックについて説明しています。

- 184 ページの「メッセージングトポロジのコンポーネント」

メッセージングトポロジのコンポーネント

176 ページの「メッセージングトポロジの設計」で、メッセージングトポロジのコンポーネントである Messaging Server、Directory Server、およびクライアントの 3 つについて簡単に説明しました。この節では、基本的なメッセージングトポロジにおけるその他のコンポーネントについて説明します。

Messaging Server: ユーザーのメールボックスを収容して管理します。また、「インターネット接続 MTA」と「MTA リレー」で説明されているように、Messaging Server の MTA 部分だけを含むサーバーとしても機能します。

クライアント**:** 多くの場合 Messaging マルチプレクサを通じて、Messaging Server からメッセージングサービスにアクセスします。

Directory Server: Messaging Server により名前とエイリアスの検索に使用されます。ダイレクト LDAP 検索によりメッセージがどこにルーティングされるかが決められます。

Messaging マルチプレクサ**:** メッセージ取得のために適切なメッセージングサービスにクライアントを接続します。

インターネット接続 **MTA:** インターネットからのメッセージをルーティングし、ファイアウォールを越えてリレーします。通常、Messaging Server ホストはこの機能を実行するように設定されます。

MTA リレー**:** インバウンド MTA は、着信メッセージを適切な Messaging Server 内の有効なアドレスにルーティングします。発信 MTA はクライアントからの発信メッセージを受け取り、LDAP にクエリを行なって送信先を検索し、メッセージを適切なサーバーに送信するか、ファイアウォールを越えてインターネットに向けて送信します。通常、Messaging Server ホストはこの機能を実行するように設定されます。

DNS サーバー**:** サーバー名を IP アドレスに解決し、ネットワーク内の適切なアドレスにメッセージが届くようにします。

ファイアウォール**:** 内部サイトのインターネットアクセスを制限します。組織内の部門間にもファイアウォールを設置することが考えられます。

MTA によるメッセージングシステムの保護

MTA を使えば、Messaging Server 配備を保護できるほか、サイトに出入りするメッセージトラフィックのフローを制御することができます。

インターネット接続 MTA は組織外のサイトからのメッセージを受信する単一窓口です。インターネット接続 MTA は、ファイアウォールを越えてインバウンド MTA、通常は別の Messaging Server に着信メッセージを送信します。

次に、インバウンド MTA はディレクトリのクエリを行なって、組織内のメッセージの送信先を判断します。インターネット接続 MTA は、ファイアウォールの外部ウォールと内部ウォールの間に位置するファイアウォールの非武装地帯 (DMZ) に配置され、インバウンド MTA 以外のサーバーについての情報にはアクセスしません。

アウトバウンド MTA は、クライアントから送信されたメッセージを受け取ります。送信 MTA は LDAP のクエリを行なって送信先を検索し、メッセージを適切なサーバーに送信するか、ファイアウォールを越えてインターネットに向けて送信します。これにより、ユーザーのためにメッセージを取得するというメッセージングサーバーとしての機能から MTA が解放されます。図 12–5 にこの概念を示します。

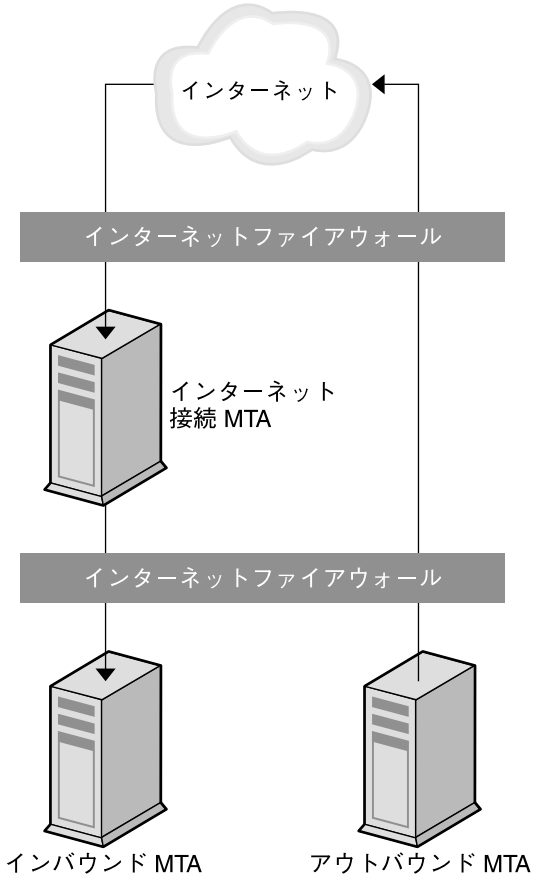

図 **12–5** メッセージングトポロジ内の MTA

MMP と MEM の使用

MMP により、Messaging Server ホストのレイアウトをエンドユーザーから隠すことができます。その結果、メールボックスが配置されているサーバーを特定することなく、ユーザーに汎用的な MMP またはロードバランサを割り当てることができます。メッセージアクセスクライアントは、受信メッセージを取得するときに MMP を指定します。

そのようなクライアント接続と認証の際に、MMP はディレクトリ内のユーザー情報の検索を行い、ユーザーのメッセージがどこにあるかを判断します。次に、MMP はクライアントを特定のサーバーに接続します。次の図は、Messaging Server に対する IMAP4 と POP3 接続のプロキシとして MMP が機能する仕組みを示します。MEM 機能を使用することで、Messenger Express のような 複合 HTTP サービスを利用できます。図 12–6 は、Messaging Server 環境においてマルチプレクサがどのように機能するかを示します。

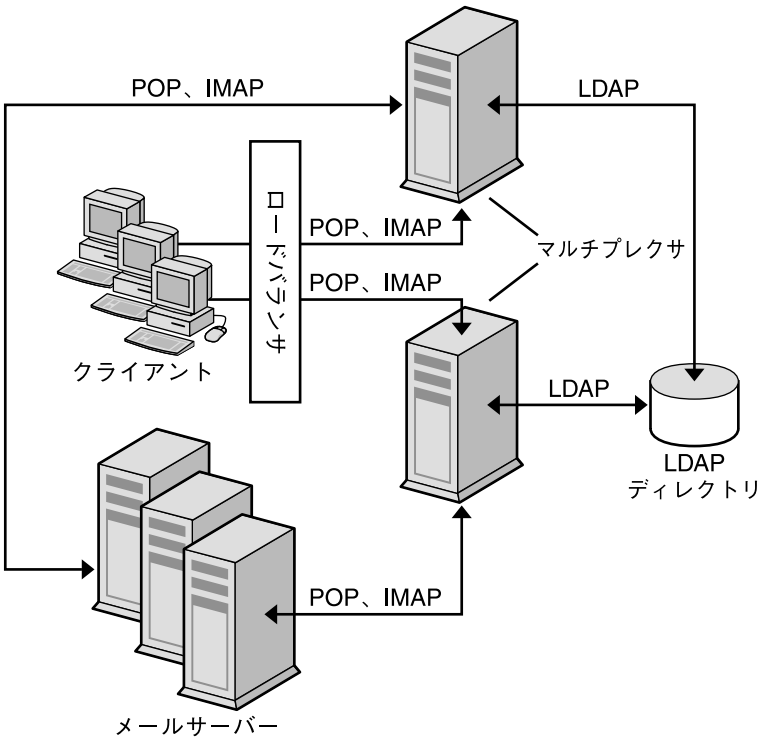

図 **12–6** MMP の概要

複数の MMP の手前にロードバランサを配置します。MMP は通常、複数個存在します。

ゲートウェイの使用

組織には、旧バージョンのメッセージングシステムがメッセージング処理の専用メソッドとして存在する場合があります。ユーザーを移行させるまで、両方のメッセージング戦略を残しておかなければなりません。これらの旧バージョンのシステムにアクセスする場合には、SMTP ゲートウェイを使用できます。これは、新規のシステムと旧バージョンのシステム間で SMTP 接続を有効にするものです。通常、旧バージョンのシステムは、インバウンド MTA がメッセージをルーティングできるように、SMTP 接続をサポートしています。

メッセージングトポロジ例の作成

トポロジ上のニーズ、戦略、トポロジ要素について基本的な部分を理解すれば、メッセージングトポロジを作成できます。メッセージングトポロジの作成方法を示すために、この節では Siroe Corporation の例を使用します。

Siroe Corporation は、ニューヨークに本社を置くマルチメディア企業です。ロサンジェルスとシカゴに小さなオフィスを持ち、サンディエゴとミネアポリスに遠隔オフィスがあります。

ステップ 1: メッセージング目標の確認

トポロジ作成の最初のステップは、組織の目標を確認することです。第 2 章で行なったように、Siroe のメッセージング目標を、ビジネス目標、技術的および財務的制約に分類します。

Siroe のビジネス目標

財務、マーケティング、法務、IT、エンジニアリングの各グループがニューヨークにあります。クリエイティブグループはロサンジェルスとサンディエゴにあります。テクニカルサポートグループはシカゴとミネアポリスにあります。メッセージのほとんどは、シカゴ、ロサンジェルス、ニューヨーク間で送信されています。

Siroe Corporation の従業員は、通信の主要手段を電子メールに依存しています。平均すると、従業員は 1 日に約 15 件のメッセージを送信しており、スプレッドシート、プレゼンテーション、またはアニメーション形式の添付ファイルを送信しています。

配備の計画者は、メッセージングサーバーのホストをシカゴ、ロサンジェルス、ニューヨークに配置することを決定しました。サンディエゴとミネアポリスの電子メールトラフィックは比較的少ないため、これらの遠隔オフィスは、シカゴとロサンジェルスのサーバーに接続するメールクライアントを持つだけになります。

Siroe の財務的および技術的制約

予算上の制約により、Siroe は稼働中の既存のインフラストラクチャーとハードウェアを使用し、サーバーをクリティカルなニーズのある場所に移動する予定です。24 時間年中無休のサポートは、ニューヨーク、シカゴ、ロサンジェルスのオフィスでのみ実施します。すべてのオフィスは T3 回線でインターネットに接続されます。

ステップ 2: トポロジ戦略の選択

メッセージングトポロジ作成の 2 番目のステップは、176 ページの「メッセージングトポロジの設計」で説明されているトポロジ戦略の選択です。Siroe Corporation は、ビジネス目標と財務的および技術的制約の評価を行いました。その結果、次の判断を下しました。

- Messaging Server ホストを遠隔オフィスに配置する必要はなく、メールクライアントだけとします。
- 遠隔オフィスには、T3 回線による高品質の帯域幅が存在します。
- 場所にかかわらず、メールユーザーは会社全体に対して大量のメッセージの送受信を行います。
- ニューヨーク、ロサンジェルス、シカゴのユーザー数が多く、ミネアポリスとサンディエゴのユーザー数は少数です。
- ニューヨーク、ロサンジェルス、シカゴにはサポート要員が存在します。

次に、Siroe Corporation は目標と制約を一般的な設計戦略にマップしました。図 12–7 は Siroe Corporation がハイブリッドトポロジを選択したことを示します。

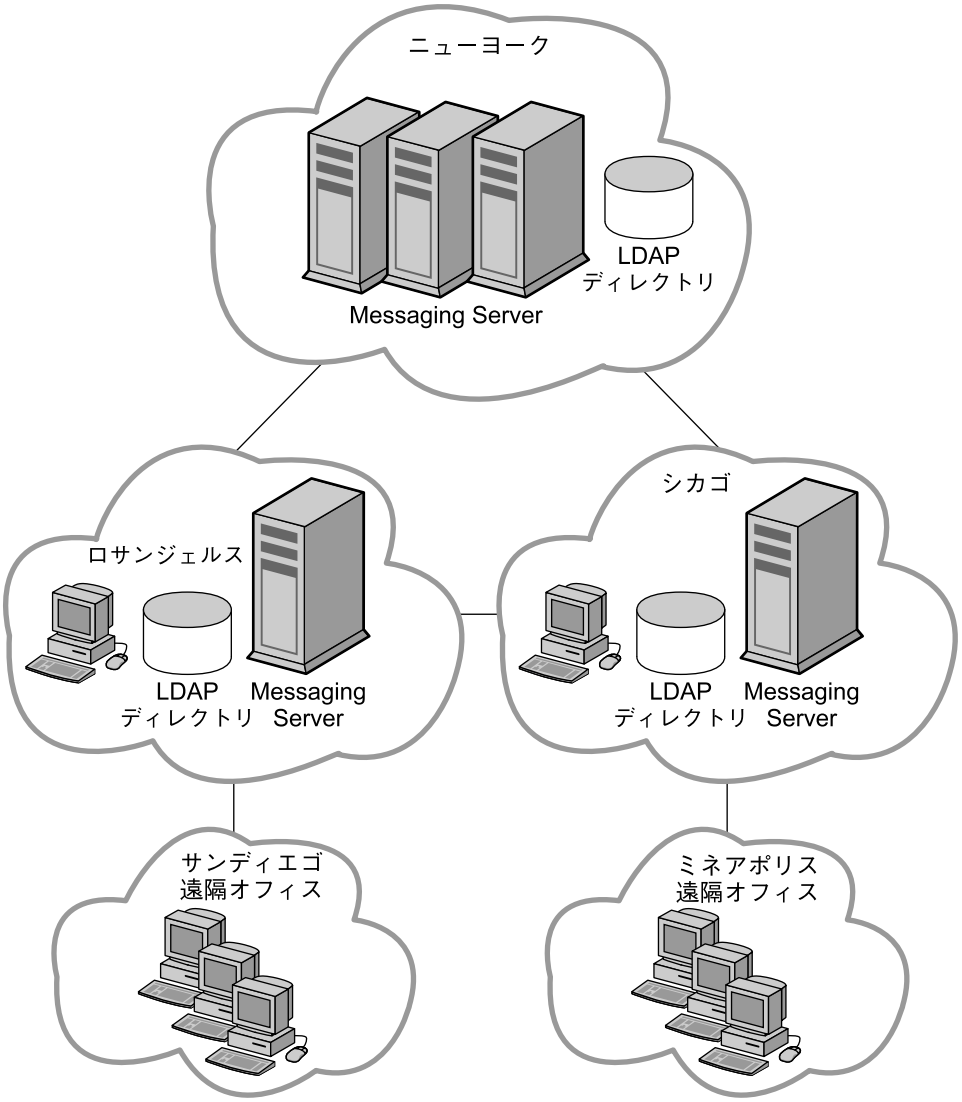

図 **12–7** Siroe Corporation のハイブリッドトポロジ

システムに対して送受信されるメッセージトランザクションのレートはニューヨークが最も高いため、Messaging Server を最も多く配置します。ニューヨークより小規模のロサンジェルスとシカゴは、サンディエゴとミネアポリスもサポートします。ただし、これらの遠隔オフィスには固有のメッセージングサーバーは必要ありません。代わりに、シカゴとロサンジェルスが遠隔オフィスのためのサービスの中央ロケーションとして機能します。

ステップ 3: トポロジ要素の計画

メッセージングトポロジ作成の最後のステップは、183 ページの「メッセージングトポロジ要素の理解」で説明されているように、実際の配備におけるトポロジ要素を計画することです。次の図は、シカゴとミネアポリスオフィスのトポロジ要素を示します。

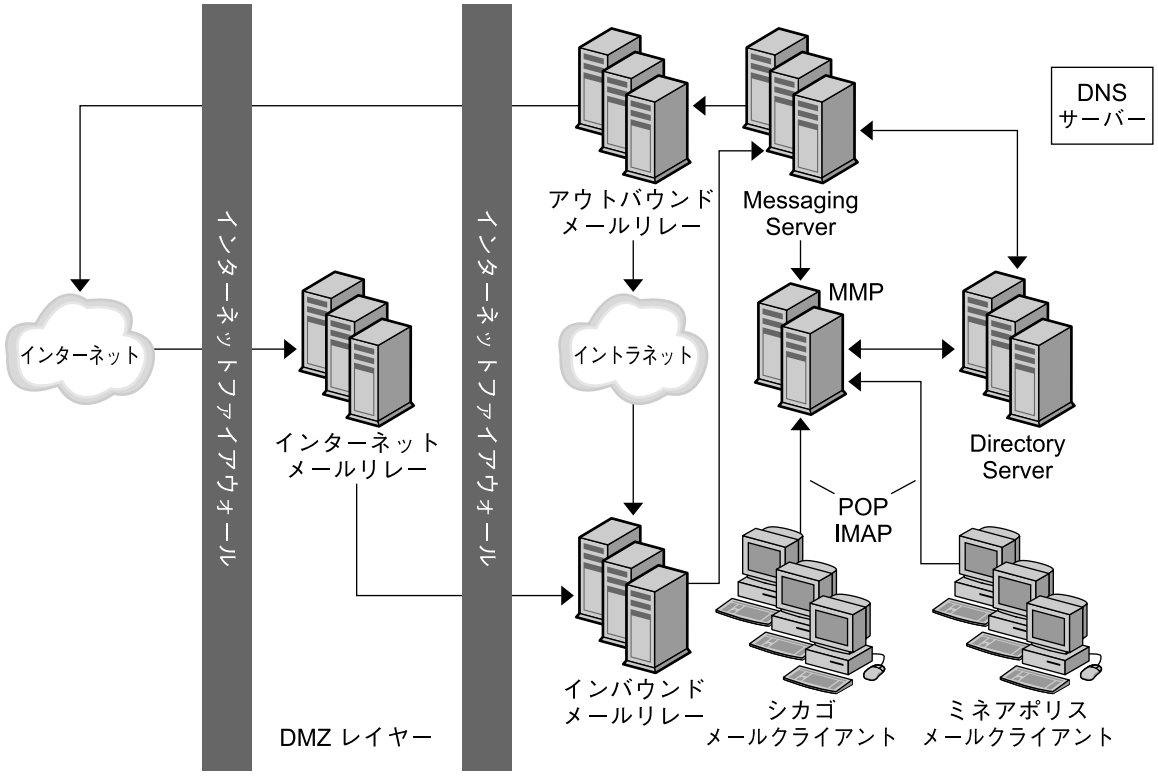

図 **12–8** シカゴとミネアポリスオフィスのための Siroe のメッセージング配備におけるトポロジ要素

作業要員の 30 パーセントがサードパーティーのベンダと請負業者で構成されるため、トポロジ内で外部ファイアウォールに内部ファイアウォールを追加して、社内の場所へのアクセスを制限します。インターネット MTA をトポロジ内に配置し、インターネットからのメッセージをルーティングし、ファイアウォールを越えてリレーします。MTA が追加され、着信メッセージと発信メッセージがルーティングされます。受信メッセージと送信メッセージを分離することにより、大量のメッセージトラフィックに対応できます。MMP は、従業員の POP および IMAP メールクライアントをMessaging Server 内のそれぞれのメールボックスに接続します。MMP を使用することで、従業員はログイン時に特定のメールホストを知る必要がなく、管理者は従業員のメールボックスを別のメールサーバーにシームレスに移動できます。

メッセージングトポロジを作成することで、配備におけるすべての要素の物理的および論理的配置を考慮できます。また、導入のやり直しを最小限にとどめることが可能になります。

第 13 章

Messaging Server セキュリティーの計画

この章では、Messaging Server 配備のさまざまなコンポーネントに対する計画を立案し、それらのコンポーネントを保護する方法について説明します。

この章には、次の節があります。

配備におけるメッセージングコンポーネントの保護

この節では、メッセージング配備でコンポーネントをセキュリティーで保護する方法について説明します。

注 – それぞれのコンポーネントで、`chroot` 機能を使用して、各マシンで使用できるコマンドの数を制限します。

MTA の保護

MTA をセキュリティーで保護し、処理リソースやサーバー可用性を保護します。権限を持たないユーザーからメッセージがリレーされた場合、または大量のスパムが配信された場合には、応答速度が遅くなり、ディスク容量が圧迫され、エンドユーザーのための処理リソースが消費されます。スパムはサーバーのリソースを浪費するだけでなく、エンドユーザーを煩わせるものでもあります。

注 – 権限を持たない外部のユーザーから配備を保護するだけでなく、内部ユーザーからシステムを保護する必要もあります。

次の表で、MTA に対する最も一般的な脅威について説明します。

表 13–1 MTA に対する一般的なセキュリティー脅威

脅威	説明
UBE (Unsolicited Bulk Email) またはスパム	多数のユーザーに迷惑メールを送りつける行為のことを言います。
不正なリレー	別の会社の SMTP サーバーを使用してメールをリレーします。スパムの送信者は、証拠を残さないようにするためにこのテクニックを多用します。エンドユーザーは、スパム送信者ではなく、送信したリレーにクレームをつけていることもあります。
メール爆弾	同じメッセージを特定のアドレスに繰り返し送るような行為。大量のメッセージにより、メールボックスの容量を超過させるのが狙いです。
電子メールスプーフィング	別の発信元からの電子メールを、ある発信元からのものにみせかけます。
サービス拒否攻撃	あるサービスの正規ユーザーがそのサービスを利用できないようにします。たとえば、攻撃者がネットワークを占有し、正規のユーザーのトラフィックを妨害します。

MTA リレーに関するこの節では、配備で使用できるセキュリティーオプションについて説明します。

- 194 ページの「アクセス制御」
- 198 ページの「変換チャネルとサードパーティーのフィルタリングツール」
- 199 ページの「RBL チェック」
- 200 ページの「クライアントアクセスの制御」
- 201 ページの「セキュリティー戦略の監視」

アクセス制御

アクセス制御を使用して、特定のユーザーから (へ) のメッセージをシステムレベルで拒否できます。また、特定のユーザー間でより複雑なメッセージトラフィックの制限を構成することもできます。さらに、ユーザーに独自の受信メッセージのフィルタ設定を許可し、メッセージヘッダの内容に基づいてメッセージを拒否することなどができます。

エンベロープレベルでアクセス制御を行うときは、マッピングテーブルを使用してメールのフィルタリングを行います。ヘッダベースでアクセス制御を行う場合、またはユーザー独自の制御を行う場合は、サーバー側の規則とともに一般的なメールボックスフィルタを使用します。

マッピングテーブルの概要

特定の「マッピングテーブル」を設定することにより、メールサービスへのアクセスを制御できます。MTA の多くのコンポーネントでは、テーブル検索指向の情報を利用しています。この種類のテーブルは、変換、すなわち入力文字列を出力文字列へ「マップ」するのに使用されます。マッピングテーブルは通常、2 つの列として表示されます。1 列目 (左側) には、マッチングの対象となる可能性のある入力文字列 (パターン)、2 列目 (右側) には入力文字列のマッピングの結果である出力文字列 (テンプレート) が表示されます。

次の表で、マッピングテーブルの使用により、だれがメールを送信または受信できるのか、あるいは送受信ともにできるのかを制御する方法を説明します。詳細については、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』を参照してください。

表 **13–2** アクセス制御マッピングテーブル

マッピングテーブル	説明
SEND_ACCESS	エンベロープ From: アドレス、エンベロープ To: アドレス、ソースおよび送信先チャネルに基づいて、着信接続をブロックする場合に使用します。書き換え、エイリアスの展開などが実行されたあとで To: アドレスが調べられます。
ORIG_SEND_ACCESS	エンベロープ From: アドレス、エンベロープ To: アドレス、ソースおよび送信先チャネルに基づいて、着信接続をブロックする場合に使用します。 To: アドレスは、書き換えのあとに、しかしエイリアスの展開より先に調べられます。
MAIL_ACCESS	SEND_ACCESS および PORT_ACCESS の各テーブル内の情報の組み合わせに基づいて着信接続をブロックするために使用します。SEND_ACCESS 内のチャネルおよびアドレス情報と、PORT_ACCESS 内の IP アドレスおよびポート番号情報を組み合わせた情報が基準となります。
ORIG_MAIL_ACCESS	ORIG_SEND_ACCESS および PORT_ACCESS の各テーブル内の情報の組み合わせに基づいて着信接続をブロックするために使用します。ORIG_SEND_ACCESS 内のチャネルおよびアドレス情報と、PORT_ACCESS 内の IP アドレスおよびポート番号情報を組み合わせた情報が基準となります。
FROM_ACCESS	エンベロープ From: アドレスに基づいてメールをフィルタリングする場合に使用します。このテーブルは、To: アドレスが不適切な場合に使用します。
PORT_ACCESS	IP 番号に基づいて着信接続をブロックする場合に使用します。

図 13–1 は、メール受信プロセスの中でマッピングテーブルが使用される場所を示したものです。

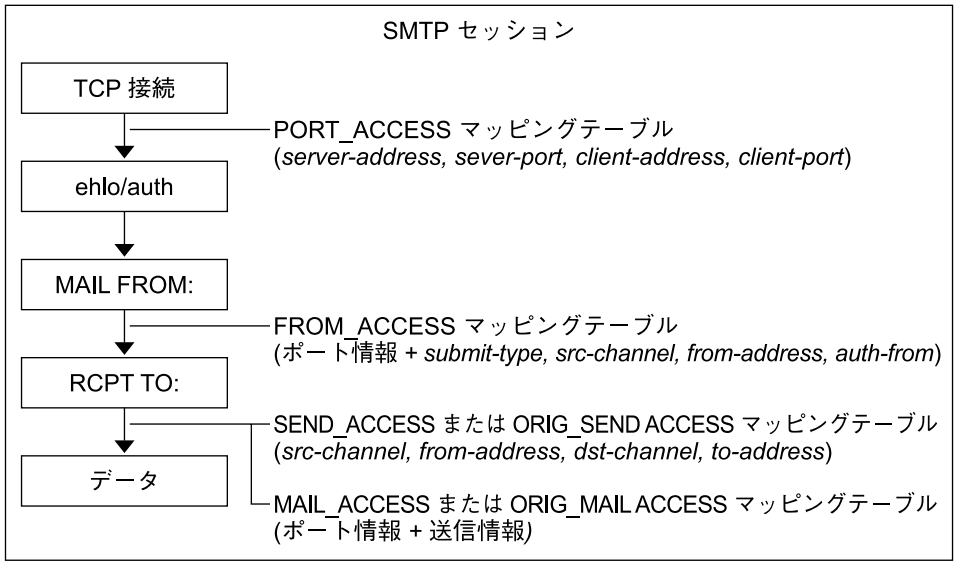

図 **13–1** マッピングテーブルとメール受信プロセス

MTA サービスディスパッチャーが制御するすべてのネットワークポートで、`PORT_ACCESS` 拒否応答が保証されている場合は、リモートホストから最初の接続が行われた時点で、それが実行されます。`FROM_ACCESS` による拒否は、送信側が受信者情報またはメッセージデータを送信する前に、`MAIL FROM:` コマンドへの応答として行われます。`SEND_ACCESS` または `MAIL_ACCESS` による拒否は、送信側がメッセージデータを送信する前に、`RCPT TO:` コマンドへの応答として行われます。SMTP メッセージが拒否された場合は、Messaging Server がメッセージデータを受信せずメッセージデータを確認しないため、そのような拒否を処理するためのオーバーヘッドが最小になります。複数のアクセス制御マッピングテーブルが存在する場合、Messaging Server はそれらをすべて調べます。

注 – メッセージが受け入れられた場合は、さらに変換チャネルとユーザー定義のフィルタによりフィルタリングされます。

マッピングテーブルによるリレー防止設定

アクセス制御マップを使うことによって、Messaging Server システムが SMTP メールのリレーに利用されるのを防ぐことができます。たとえば、あるユーザーが迷惑メールのリレーにメールシステムを使用して、システム上の多数のメールボックスに送信しようとするような場合です。

Messaging Server のデフォルトでは、ローカルの POP メールクライアントおよび IMAP メールクライアントによるリレーを含むすべての SMTP リレー操作が防止されます。204 ページの「認証された SMTP を有効にする」で説明しているように、クラ

イアントが SMTP AUTH を使用して認証せず、Messaging Server の SMTP サーバーを介して外部アドレスにメッセージを送信しようとした場合、その送信は拒否されます。このため、内部システムとリレーを許可するサブネットを認識するように設定を変更した方がよいでしょう。

▼ 外部ホストからのリレーを防止するには

ドメイン外にあるホストからドメイン外の別のホストにメッセージがリレーされるのを防ぐには、次の方法を取ります。

手順 **1.** 受信メールをいくつかのチャネルに分けます。例:

- ドメイン内の IP アドレスは `tcp_internal` チャネルに送られます。
- 認証されたセッションは `tcp_auth` チャネルに送られます。
- その他のすべてのメールは、`tcp_local` チャネルに送られます。

2. 『**Sun Java System Messaging Server 6 2005Q4** 管理ガイド』の「メールのフィルタリングとアクセス制御」の章で詳しく説明されているように、**`INTERNAL_IP`** マッピングテーブルを使用して、**POP** クライアントと **IMAP** クライアントからのメールを識別し、処理を許可します。

メールボックスフィルタの使用

フィルタは、メッセージに適用される 1 つ以上の条件付き処理で構成されています。Messaging Server フィルタはサーバーに保存され、サーバーによって評価されます。そのため、それらはサーバー側規則 (SSR) と呼ばれることもあります。

チャネルレベルのフィルタと MTA 全体のフィルタを作成し、不正メールの配信を防止できます。また、フィルタテンプレートを作成し、Messenger Express を使用してそれをエンドユーザーに使わせることもできます。エンドユーザーはテンプレートを使用して個人のメールボックスフィルタを構築し、不要なメールメッセージが自分のメールボックスに配送されないようにすることができます。サーバーは、次の優先順位に従ってフィルタを適用します。詳細については、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』を参照してください。

1. ユーザー単位のフィルタ

   ユーザー単位のフィルタは、特定ユーザーのメールボックスに送信されるメッセージに適用されます。フィルタテンプレートを作成し、Messenger Express クライアントを使用してそれをエンドユーザーに使わせることができます。エンドユーザーはテンプレートを使用して個人のサーバーフィルタを構築し、自分のメールボックスへのメールメッセージの配送を管理できます。フィルタにより、不要なメッセージ、リダイレクトメールなどの拒否や、メールボックスフォルダに配信されるメッセージのフィルタリングなどが行われます。

   個人用メールボックスフィルタにメッセージの許可あるいは拒否が定義されている場合は、メッセージに対してそのフィルタ処理が行われます。

   フィルタテンプレートは、Sieve スクリプトの「ハードコード」された要素をプロンプトと入力フィールドに置き換えることで、Sieve スクリプトを一般化します。Java サーブレットは、Sieve テンプレートを解析し、ブラウザ内でユーザーインタ

フェースを生成するのに使用されます。エンドユーザーが入力フィールドに値を入力すると、サーブレットがその値を取得して、ユーザーのディレクトリにあるプロファイルエントリ内の Sieve スクリプトに保存します。Messenger Express インタフェースを通じて、プロンプトと入力フィールドがエンドユーザーに提示されます。

しかし、受取人がメールボックスフィルタを設定していない場合、またはユーザーのメールボックスフィルタが明示的に適用されないメッセージの場合、Messaging Server によってチャネルレベルのフィルタが適用されます。

2. チャネルレベルのフィルタ

   チャネルレベルのフィルタは、チャネルのキューに入った各メッセージに適用されます。この種のフィルタの一般的な用途は、特定のチャネルから入ってくるメッセージをブロックすることです。

   チャネルレベルのフィルタを作成するには、Sieve を使用してフィルタを書く必要があります。Sieve を使用してフィルタを作成する場合の指示の詳細については、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』の第 17 章「メールのフィルタリングとアクセス制御」を参照してください。

   チャネルレベルのフィルタにメッセージの許可あるいは拒否が定義されている場合は、メッセージに対してそのフィルタ処理が行われます。それ以外の場合は、Messaging Server によって MTA 全体のフィルタが適用されます (該当する場合)。

3. MTA 全体のフィルタ

   MTA 全体のフィルタは、MTA のキューに入るすべてのメッセージに適用されます。この種のフィルタの一般的な用途は、メッセージの送信先とは関係なく、ダイレクトメールや受信したくないメッセージをブロックすることです。

   MTA 全体のフィルタを作成するには、Sieve を使用してフィルタを書く必要があります。Sieve を使用してフィルタを作成する場合の指示の詳細については、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』の第 17 章「メールのフィルタリングとアクセス制御」を参照してください。

   デフォルト設定を使用した場合、それぞれのユーザーはメールボックスフィルタを所有していません。ユーザーが Messenger Express インタフェースにアクセスして 1 つまたは複数のフィルタを作成すると、そのフィルタが LDAP ディレクトリに保存されます。

変換チャネルとサードパーティーのフィルタリングツール

変換チャネルは、MTA を通じて配信されるメッセージを本文部分ごとに変換します。この処理は、サイトで提供されるプログラムかコマンドにより行われます。変換チャネルは、テキストや画像のフォーマット変換、ウイルスのスキャン、言語の変換などを行うことができます。MTA で通信するさまざまなメッセージ形式を変換することができ、特定の処理やプログラムをメッセージの本文部分に指定することができます。変換チャネルをウイルススキャニングプログラムと併用する場合は、ウイルスの除去、メッセージの保留または拒否を選択できます。特別な変換チャネル設定を使用すると、それぞれのメッセージ本文に対する適切な変換を選択できます。詳細については、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』の第 13 章「定義済みチャネルを使用する」を参照してください。

注 – 変換チャネルのような特別な処理を行うと、システムに余分の負荷がかかります。戦略のサイズを検討する場合には、この点を考慮してください。

変換チャネルを使用すると、サードパーティーのスパム防止およびウイルス対策ソフトウェアソリューションを利用できます。また、MTA API を使用してチャネルを作成し、リモートスキャニングエンジンを起動することもできます。MTA API の詳細については、『Sun Java System Messaging Server 6 2005Q4 Administration Reference』を参照してください。

一般に、サードパーティーのソリューションは外部サイトから保護して、バックエンドまたは中間のリレーのみで使用するのが最も適した使い方です。

Brightmail ソリューションは、Brightmail サーバーと、リアルタイムのスパム防止およびウイルス対策 (サービスプロバイダ向けのみ) 規則アップデートで構成されており、規則はメッセージングサーバーにダウンロードされます。Brightmail Logistics and Operations Center (BLOC) が電子メールプローブからスパムを受信すると、オペレータがただちに適切なスパム防止規則を作成します。次に、これらの規則が Brightmail カスタママシンにダウンロードされます。同様に、Symantec Security Response のリアルタイムのウイルス規則が Brightmail から送信されます。これらの規則は顧客の Brightmail サーバーでスパムやウイルスを検出するために使用されます。

Messaging Server では、SpamAssassin の使用もサポートされています。SpamAssassin はフリーウェアのメールフィルタで、スパムの特定に使用されます。SpamAssassin では、すべてのメッセージのスコアが計算されます。スコアは、メッセージヘッダーや本文の情報に対して一連のテストを実行することによって計算されます。各テストに成功するか失敗するかによってスコアは調整されます。スコアは正または負の実数です。スコアが一定のしきい値を超えると、スパムであるとみなされます。

Brightmail および SpamAssassin の Messaging Server に対する設定の詳細については、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』の第 14 章「スパムとウィルスのフィルタ処理プログラムを Messaging Server に統合する」を参照してください。

RBL チェック

Mail Abuse Protection System の Real-time Blackhole List (MAPS RBL) は、スパムの発信やリレーを行なったり、スパムのサポートサービスを提供したりしてホストやネットワークを悪用している者に好意的、あるいは中立的な立場を取っていると判断されたホストとネットワークのリストです。

外部からの MAPS RBL に対する接続の比較を行うように、MTA を設定することができます。また、DNS ベースのデータベースを使用して、不特定多数宛のメールを送る可能性のある受信 SMTP 接続を判別できます。

詳細については、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』の第 17 章「メールのフィルタリングとアクセス制御」を参照してください。

クライアントアクセスの制御

Messaging Server は、POP、IMAP、および HTTP について、サービスごとの高度なアクセス制御機能をサポートしています。Messaging Server のアクセス制御機能は、TCP デーモンと同じポートで待機するプログラムです。アクセス制御機能では、アクセスフィルタによるクライアントの識別情報の検証が行われ、そのクライアントがフィルタリング処理を通過した場合は、デーモンへのアクセスが許可されます。

大企業やサービスプロバイダのメッセージングサービスを管理する場合、これらの機能を使用して、スパム (大量メール送信) や DNS スプーフィングを行うユーザーをシステムから除外したり、ネットワークの全般的なセキュリティーを強化したりできます。

Messaging Server の TCP クライアントアクセス制御システムは、必要な場合、その処理の一部として、次のようなソケットの終端アドレスの分析を行います。

- 両方の終端の逆引き DNS 検索 (名前に基づくアクセス制御を行うため)
- 両方の終端の正引き DNS 検索 (DNS スプーフィングを検出するため)
- `Identd` コールバック (クライアントエンドのユーザーがクライアントホストに認識されていることを調べるため)

システムは、この情報を「フィルタ」と呼ばれるアクセス制御文と比較して、アクセスの許可または拒否を決定します。サービスごとに、個別の許可フィルタと拒否フィルタのセットを使用して、アクセスを制御します。許可フィルタは明示的にアクセスを許可し、拒否フィルタは明示的にアクセスを禁止します。

クライアントがサービスへのアクセスを要求すると、アクセス制御システムは、そのクライアントのアドレスまたは名前情報を、次の条件を使用して順番に対象のサービスのフィルタと比較します。

1. 検索は、最初の一致項目が見つかった時点で終了する。許可フィルタは、拒否フィルタより先に処理されるため、許可フィルタが優先される。
2. クライアント情報が対象のサービスの許可フィルタに一致した場合は、アクセスが許可される。
3. クライアント情報がそのサービスの拒否フィルタに一致した場合は、アクセスが拒否される。
4. どの許可または拒否フィルタにも一致しなかった場合、アクセスが許可される。例外は、許可フィルタは存在しているが拒否フィルタが存在しない場合で、その場合にはフィルタに一致しなかったアクセスは拒否される。

ここで説明するフィルタの構文は柔軟性に富んでいるため、わかりやすい簡単な方法で、さまざまなアクセス制御ポリシーを実装できます。許可フィルタと拒否フィルタは自由に組み合わせて使用できますが、大半のアクセスを許可するフィルタまたは大半のアクセスを拒否するフィルタを使用すると、ほとんどのポリシーを実装できます。

クライアントアクセスフィルタは、問題のあるドメインの数が把握できる場合に特に有効です。UBE の場合、Messaging Server はすべてのスパムメッセージを格納し処理する必要がありますが、クライアントアクセスフィルタの場合はスパムメッセージを処理する必要がありません。クライアントアクセスフィルタはドメイン全体からのメールをブロックするため、この機能は慎重に使用する必要があります。

クライアントアクセスフィルタには、次の制限があります。

- メッセージをリレーする前に、SMTP クライアントがログインする必要があります。
- クライアントアクセスフィルタは、大規模な配備には向いていません。

クライアントアクセスフィルタの詳細については、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』の第 19 章「セキュリティーとアクセス制御を設定する」を参照してください。

セキュリティー戦略の監視

サーバーの監視は、セキュリティー戦略で重要な位置を占めます。システムに対する攻撃を識別するには、メッセージキューのサイズ、CPU の使用率、ディスクの空き容量、ネットワークの使用率を監視します。メッセージキューのサイズが異常に大きくなったり、サーバーの応答時間が長くなったりするのは、MTA リレーへの攻撃の可能性があります。また、通常とは異なるシステムの負荷パターンや接続についても調査します。ログを毎日チェックして、異常な活動がないか調べます。

メッセージストアの保護

メッセージングサーバーで最も重要なデータは、メッセージストア内のユーザーのメールです。メールメッセージは、暗号化されない個別のファイルとして格納されることに留意してください。したがって、物理的なアクセスや `root` アクセスからメッセージストアを保護する必要があります。

メッセージストアをセキュリティーで保護するには、ストアがインストールされているマシンへのアクセスを制限します。暗号化されないプレーンテキストのパスワードの代わりに、CRAM-MD5 パスワードまたは Digest-MD5 パスワードを使用できます。パスワードの詳細については、203 ページの「メッセージングユーザー認証の計画」を参照してください。

ストアマシンの認証にパスワードを作成するだけでなく、VPN アクセス、`ssh`、または `pam` のような、マシンへのログインが許可された有効なユーザーを一覧表示するツールを使用することもあります。

また、1 層のアーキテクチャーよりも 2 層のアーキテクチャーをお勧めします。メッセージストアは、メッセージングシステムのコンポーネント中で最もディスクに負担をかける作業を行うため、フィルタリング、ウイルススキャニング、およびその他のディスクに負担をかけるセキュリティー処理を同じマシンで行わないようにしま

す。2 層のアーキテクチャーでは、システムに余分な負荷がかかるメッセージストアと同じマシンで UBE フィルタ、リレー防止機能、およびクライアントアクセスフィルタを使用せずにすみます。代わりに、MTA がその処理を行います。さらに、ストアへのユーザーアクセスが 2 階層配備の MMP または MEM により制限され、実質的にメッセージストアにセキュリティー層を追加したことになります。

1 層のアーキテクチャーで配備を行う場合は、セキュリティー処理の追加と、SSL やウイルススキャニングなどに必要となる負荷を考慮してください。詳細については、第 10 章を参照してください。

メッセージストアにセキュリティー処理を追加する場合は、ユーザーごとにディスク割り当てを行なって、ディスクの使用率を制限します。また、空き容量が制限に近づいたときには管理アラームを出すようにします。さらに、MTA の場合と同様に、サーバーの状態、ディスク容量、サービスの応答時間を監視します。詳細については、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』の第 18 章「メッセージストアを管理する」を参照してください。

MMP と MEM の保護

MMP はメッセージストアのプロキシとして機能するため、エンドユーザーデータへのアクセスを防ぎ、権限のないアクセスから保護する必要があります。ユーザー ID とパスワードは、基本的な認証機能となります。さらに、クライアントアクセスフィルタを使用すれば、ユーザーが特定のドメインや特定の IP アドレスの範囲にアクセスするのを制限できます。SMTP リレーサーバーのセキュリティーを提供する方法としては、SMTP 認証または SMTP AUTH (RFC 2554) をお勧めします。SMTP AUTH は、認証済みのユーザーだけに MTA を介したメール送信を許可します。詳細については、204 ページの「認証された SMTP を有効にする」を参照してください。

POP サービスまたは IMAP サービスの前に、MMP を別のマシンまたは別のユーザー ID のもとに配置します。フロントエンドマシンには MMP と MTA のみを配置してから、フロントエンドマシン、メールストア、および LDAP サーバー間で、物理的にセキュリティー保護されたネットワークを構築できます。

ユーザーがインターネットからログインする場合は、Messenger Express からメッセージストアへのアクセスのセキュリティーには特に配慮が必要となります。一般的には、ストアはファイアウォールにより外部と分離します。さらに、HTTP アクセスサービスへの単一の接続ポイントとして機能する特別なサーバーとして、Messenger Express マルチプレクサ (MEM) を使用することも考えられます。MMP と同様に、MEM は、メールクライアントとの間で、暗号化されていない通信と暗号化された (SSL) 通信の両方をサポートしています。MEM は、エンドユーザーデータへのアクセスと権限のないアクセスからの保護も行う必要があります。

ログファイルを定期的に監視することで、権限のないアクセスを防ぐことができます。

メッセージングユーザー認証の計画

ユーザー認証を行うことで、ユーザーはメールクライアントへのログインとメールメッセージの取得が可能になります。ユーザー認証の方法には次のものがあります。

プレーンテキストと暗号化されたパスワードによるログイン

ユーザー ID とパスワードは、LDAP ディレクトリに保存されます。最低限必要な長さなど、パスワードのセキュリティー基準は、ディレクトリポリシー要件によって決定されます。パスワードのセキュリティー基準は、Messaging Server 管理の一部ではありません。ディレクトリサーバーのパスワードポリシーについては、『Sun Java System Directory Server 5 2005Q1 Deployment Plannning Guide』を参照してください。

管理者は、メッセージング設定パラメータを設定して、プレーンテキストのパスワードを許可するかどうか、パスワードの暗号化を必須とするかどうかを決めることができます。詳細については、『Sun Java System Messaging Server 6 2005Q4 Administration Reference』の `service.`*xxx*`.plaintextminciper` パラメータ (*xxx* は `http`、`pop`、`imap` のいずれか) を参照してください。

プレーンテキストによるログインと暗号化されたパスワードによるログインは、どちらも POP、IMAP、および Messenger Express ユーザーアクセスプロトコルで使用できます。

Simple Authentication and Security Layer (SASL) による認証

SASL (RFC 2222) は、POP、IMAP、および SMTP ユーザーアクセスプロトコルの追加認証メカニズムとして機能します。Messaging Server は、表 13–3 に一覧表示されているユーザーアクセスプロトコルの SASL をサポートしています。

表 **13–3** SASL 認証のユーザーアクセスプロトコルのサポートマトリックス

	プレーン	ログイン	CRAM-MD5	Digest-MD5	証明書	APOP
SMTP AUTH	Yes	Yes	Yes	Yes	-	-
POP	Yes	-	Yes	Yes	-	Yes
IMAP	Yes	-	Yes	Yes	-	-
HTTP (Messenger Express)	Yes	-	-	-	Yes	-

注 –

- CRAM-MD5 を使用する場合は、パスワードをプレーンテキスト形式で LDAP ディレクトリサーバーに保存する必要があります。
- Digest-MD5 は MMP ではまだサポートされていませんが、MMP を使用しないようにすればサポートされます。
- POP を使用する場合は、パスワードをプレーンテキストで LDAP ディレクトリサーバーに保存する必要があります。

SASL を使用する場合、セッションで SSL を使用しないと、ユーザー名とパスワードは暗号化されません。SSL の詳細については、207 ページの「SSL による暗号化」を参照してください。SASL メカニズム、PLAIN および LOGIN は、認証情報を復号化しますが、情報が捕捉された場合には容易に解読されてしまいます。このような限界があるにもかかわらず、SASL は SMTP AUTH (204 ページの「認証された SMTP を有効にする」を参照) と組み合わせて、システムで認証されたユーザーにだけシステムを経由したメールのリレーを許可できるため、便利です。たとえば、正当なユーザーが SMTP サーバーへの認証を受けると、SMTP サーバーで別のチャネルへの切り替えを設定できます。このようにすると、認証されたセッションからのメッセージは、認証されていないユーザーとは別の TCP チャネルから送られてくるメッセージとなります。内部ネットワークのユーザーからのメッセージも、着信接続の IP アドレスに基づいて、その他の発信元からのメッセージとは別のチャネルに切り替えできます。

SASL の詳細については、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』の第 19 章「セキュリティーとアクセス制御を設定する」を参照してください。

認証された SMTP を有効にする

デフォルトでは、ユーザーは、メッセージ送信時に Messaging Server の SMTP サービスに接続する際に、パスワードを送信する必要はありません。ただし、SMTP へのパスワードログインを有効にすれば、認証された SMTP を使用できるようになります。

認証された SMTP (SMTP AUTH とも呼ばれる) は、SMTP プロトコルを拡張したものです。認証された SMTP を使用すると、サーバーへのクライアント認証が可能になります。認証は、メッセージの送受信時に実行されます。認証された SMTP の主な用途は、悪用される可能性のあるオープンリレーを作成することなく、オフィス外のローカルユーザーがメールを送信するのを可能にすることです。クライアントは、AUTH コマンドを使用してサーバーに対する認証を行います。

認証された SMTP は、SMTP プロトコルによるメッセージの送信をセキュリティーで保護します。認証された SMTP を使用する場合に、証明書に基づいたインフラストラクチャーを用意する必要はありません。証明書による認証については、205 ページの「Secure Sockets Layer (SSL) による証明書ベースの認証」を参照してください。

認証された SMTP を使用すると、クライアントは認証メカニズムをサーバーに提示し、認証プロトコルの交換を行うことができます。さらに任意で、後続のプロトコル相互対話で使用するセキュリティー層とネゴシエーションを行うこともできます。

メールの送信に SMTP AUTH の使用を要求している場合は、適切なログを記録してメールが悪用されたケースを追跡できます。

認証された SMTP の詳細については、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』の MTA に関する章を参照してください。

Secure Sockets Layer (SSL) による証明書ベースの認証

Messaging Server は、SSL プロトコルを使用して、暗号化通信とクライアントおよびサーバーの証明書ベースの認証を行います。この節では、証明書ベースの SSL 認証について説明します。SSL 暗号化の詳細については、207 ページの「SSL による暗号化」を参照してください。

SSL は公開鍵暗号法の概念に基づいています。TLS (Transport Layer Security) は SSL のスーパーセットとして機能しますが、名前が混同されて使われています。

SSL をサポートしているサーバーには、証明書、公開鍵、非公開鍵、証明書、鍵、およびセキュリティーデータベースが高レベルで必要となります。これにより、メッセージの認証、機密、完全性が確保されます。

表 13–4 で、各クライアントアクセスプロトコルによる SSL 認証のサポートについて説明します。

表 13–4 SSL 認証のサポートマトリックス

	MMP による **SSL**	代替ポートでの **MMP** による **SSL**	**SSL**	代替ポートでの **SSL**
SMTP	Yes	Yes	Yes	Yes

表 **13–4** SSL 認証のサポートマトリックス　(続き)

	MMP による **SSL**	代替ポートでの **MMP** による **SSL**	**SSL**	代替ポートでの **SSL**
POP	-	Yes	-	Yes
IMAP	Yes	Yes	-	Yes
Messenger Express (HTTP)	Yes (Messenger Express マルチプレクサによる)	Yes (Messenger Express マルチプレクサによる)	-	Yes

SMTP、POP、および IMAP プロトコルは、クライアントとサーバーが SSL なしで通信を開始したあと、"start TLS" と同等のコマンドを使用して SSL 通信に切り替える方法を提供します。"start TLS" を実装していないクライアントの場合は、SMTP、POP、および IMAP サーバーが SSL を代替ポートで使用するように設定することもできます。

SSL による認証を行うには、メールクライアントはサーバーとの SSL セッションを確立し、ユーザーの証明書をサーバーに提出します。サーバーは、提出された証明書が本物であるかどうかを評価します。証明書の信頼性が確認されると、そのユーザーは認証済みであるとみなされます。

SSL を認証用途で使う場合、Messaging Server 用のサーバー証明書を入手する必要があります。この証明書は、クライアントやほかのサーバーに対して、そのユーザーのサーバーを特定します。サーバーは、クライアントの認証に使用する、信頼できる認証局 (CA) の証明書をいくつでも持つことができます。

SSL の詳細については、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』の第 19 章「セキュリティーとアクセス制御を設定する」を参照してください。

メッセージ暗号化戦略の計画

この節では、暗号化とプライバシーソリューションについて説明します。次のトピックについて説明しています。

- 207 ページの「SSL による暗号化」
- 208 ページの「署名され暗号化された S/MIME」

SSL による暗号化

SSL は、IMAP、HTTP、および SMTP のアプリケーション層の下のプロトコル層として機能します。Messaging Server とそのクライアント間、および Messaging Server とほかのサーバー間におけるメッセージの転送が暗号化される場合は、通信が盗聴される危険性はほとんどありません。また、接続しているクライアントとサーバーが認証済みの場合は、侵入者がそれらのクライアントになりすます (スプーフィングする) 危険性もほとんどありません。

メッセージ送信でエンドツーエンドの暗号化を行うには、S/MIME を使用する必要があります。詳細については、208 ページの「署名され暗号化された S/MIME」を参照してください。

注 – SSL 接続の設定によりパフォーマンスのオーバーヘッドが生じると、サーバーへの負担となります。メッセージングシステムの設計とパフォーマンスの分析を行う際には、セキュリティー要件とサーバーの容量のバランスをとる必要があります。

暗号化の用途で SSL を使用する場合は、ハードウェア暗号化アクセラレータをインストールすることでサーバーのパフォーマンスを向上させることができます。一般的に、暗号化アクセラレータは、サーバーマシンに常設されたハードウェアボードとソフトウェアドライバで構成されます。

HTTP/SSL (HTTPS) を使用したクライアントとサーバー間の SSL 接続プロセスは、次のようになります。

1. クライアントが HTTPS を使用して接続を開始します。クライアントが、使用する秘密鍵アルゴリズムを指定します。
2. サーバーが認証のための証明書を送り、使用する秘密鍵アルゴリズムを指定します。クライアントと共通の最も強力なアルゴリズムが指定されます。秘密鍵が一致しない場合 (たとえば、クライアントの鍵が 40 ビットのみで、サーバーが 128 ビットの鍵を要求している場合)、その接続は拒否されます。
3. サーバーがクライアント認証を要求するように設定されている場合、この時点でクライアントに証明書が要求されます。
4. クライアントは、サーバーの証明書の正当性をチェックし、次の内容を確認します。
   - 期限が切れていない
   - 既知の署名された認証局
   - 有効な署名
   - 証明書のホスト名が HTTPS 要求のサーバーのホスト名と一致している

SSL 暗号化方式

暗号化方式とは、暗号化プロセスでデータの暗号化と復号化に使用されるアルゴリズムのことです。各暗号化方式によって強度が異なります。つまり、強度の高い暗号化方式で暗号化されたメッセージほど、承認されていないユーザーによる解読が困難になります。

暗号化方式では、キーをデータに適用することによってデータを操作します。一般的に、暗号化方式で使用するキーが長いほど、適切な解読キーを使わずにデータを解読することが難しくなります。

クライアントは、Messaging Server と SSL 接続を開始するときに、サーバーに対して、希望する暗号化用の暗号化方式とキー長を伝えます。暗号化された通信では、両方の通信者が同じ暗号化方式を使用する必要があります。一般的に使用される暗号化方式とキーの組み合わせは数多くあります。そのため、サーバーが柔軟な暗号化方式をサポートしている必要があります。暗号化方式の詳細については、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』の第 19 章「セキュリティーとアクセス制御を設定する」を参照してください。

署名され暗号化された S/MIME

署名され、暗号化されたメッセージは、Secure/Multipurpose Internet Mail Extensions (S/MIME) メッセージと呼ばれます。S/MIME は、クライアント間の通信をセキュリティーで保護する手段です。

S/MIME を使用すると、送信者は送信する前にメッセージを暗号化できます。受信者は、受信した暗号化されたメッセージを保存し、あとで読むときだけそれを解読することができます。

Communications Express Mail に S/MIME のセキュリティー機能が追加されました。S/MIME を使用するように設定された Communications Express Mail ユーザーは、ほかの Communications Express Mail ユーザーや、Microsoft Outlook メールシステムなどの S/MIME をサポートするメールクライアントのユーザーと、署名または暗号化されたメッセージを交換できます。詳細については、330 ページの「Communications Express メールで S/MIME を使用するための要件」を参照してください。

S/MIME をサポートするその他のクライアントについては、各クライアントのマニュアルで S/MIME の設定方法を確認してください。

第 14 章

Messaging Server スパム防止およびウイルス対策戦略の計画

Messaging Server は、一方的に送られてくる大量電子メール (UBE、または「スパム」) とウイルスに対処するためのツールを数多く提供しています。この章では、利用可能なさまざまなツールと対策について説明します。

この章には、次の節があります。

スパム防止およびウイルス対策ツールの概要

インターネットに接続されるコンピュータの数が増加し、オンラインでのビジネスが容易となるにつれて、スパムやウイルスなどを含めたセキュリティに関する問題もいっそう増加しつつあります。そのため、これらの問題に対処するための Messaging Server 配備を計画する必要があります。

Messaging Server を経由して送受信されるメールトラフィックは、さまざまな基準に基づいて異なるチャネル別に分類できます。この基準には、発信元および送信先電子メールアドレスや発信元 IP アドレスまたはサブネットが含まれます。これらのさまざまな電子メールフローやチャネルに異なる処理特性を適用できます。その結果、これらのチャネル上で、さまざまなアクセス制御、メールフィルタ、処理の優先順位、およびツールをさまざまな方法と組み合わせで使用できます。たとえば、ドメイン内から発信されたメールを配備の外部から発信されたメールと区別して処理できます。

チャネルベースのメッセージフロー以外の便利な分類方法として、メーリングリストトラフィックがあります。Messaging Server に送信されてくる特定のメーリングリストのトラフィックは、数多くのチャネルを経由して受信し、また数多くの異なる

チャネルに分けて送信できます。メーリングリストを使用すると、チャネルではなく、リスト自体を基準に考えるのが有用であることがわかります。Messaging Server はこの分類を認識し、数多くのチャネル固有のスパム対策ツールをメーリングリスト固有の方法で適用できます。

Messaging Server で使用できるスパム防止およびウイルス対策ツールの概要を次で説明します。

- アクセス制御**:** 既知のスパム発信元からのメールを排除し、組織内でメールを送受信可能なユーザーを制御できるようにします
- メールボックスフィルタリング**:** ユーザーが Web インタフェースを通じて独自のスパムフィルタを管理し、メールボックスに配信されるメールの特性を制御できるようにします
- アドレス検証**:** 不正な発信者アドレスを持つメールを拒否します
- **Real-time Blackhole List:** 既知のスパム発信元のリストを責任を持って管理し、常に更新を行う Mail Abuse Protection System の Real-time Blackhole List (MAPS RBL) に基づき、スパムの発信元として認識された発信者からのメールを拒否します
- リレーブロッキング**:** メールシステムの悪用者が、メールシステムをリレーとして使用して大量の受信者にスパムを送信しようとするのを防ぎます
- 認証サービス**:** Simple Authentication and Security Layer (SASL) プロトコルを使用して、SMTP サーバー内でのパスワード認証を有効にします
- サイドライニング**:** スパムの可能性のあるメッセージを保留するか、場合によっては削除します
- 総合追跡**:** 信頼性の高いメカニズムを使用してメッセージの発信元を特定します
- 変換チャネル**:** サードパーティーのウイルス対策およびスパム防止製品を統合します

これらのツールは個別に使用したり、組み合わせて使用したりできます。単独ですべてのスパムを防ぐことのできるツールはありません。しかし、組み合わせて使用することで、これらのツールはメールシステムの不正使用を防ぐ効果的な手段となります。次の節で、これらのツールについてより詳しく説明します。詳細については、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』を参照してください。

アクセス制御

Messaging Server には、さまざまな検証基準に従ってメールを拒否できる汎用の機能が備わっています。この基準には、メッセージの発信元および送信先電子メールアドレスや発信元 IP アドレスが含まれます。たとえば、このメカニズムを使用して、特定の発信者やドメイン全体 (たとえば `spam@public.com` というドメイン全体) からのメールを拒否できます。スクリーニング情報のために大量のリストが必要な場合は、アクセス基準を格納したデータベースでリストを拡張することもできます。UBE 関連以外でも、これと同じアクセス制御のメカニズムは、特定のチャネルからのメール送信を許可または禁止された内部ユーザーのデータベースを管理するのに適しています。たとえば、インターネットメールの送受信を許可するか禁止するかを、ユーザー別に制限できます。

詳細については、194 ページの「アクセス制御」を参照してください。

メールボックスフィルタリング

Messaging Server には、ユーザー別、チャネル別、およびシステム全体で使用できるメールフィルタがあります。ユーザー別チャネルは、Messenger Express のどの Web ブラウザからでも管理が可能です。これらのフィルタを使用して、ユーザーは自分のメールボックスに配信されるメールを制御できます。たとえば、「簡単に儲かる」式の UBE をユーザーが受け取りたくない場合、そのような件名のメールを拒否するよう指定できます。Messaging Server のメールフィルタリング機能は、Internet Engineering Task Force (IETF) により開発された Sieve フィルタリング言語 (RFC 3028 および 3685) に基づいています。

詳細については、197 ページの「メールボックスフィルタの使用」を参照してください。

Brightmail や SpamAssassin のようなサードパーティーのコンテンツフィルタリングソフトウェアを使用して、コンテンツベースなウイルススキャニングのフィルタリングを実装することも可能です。詳細については、214 ページの「スパム防止およびウイルス対策の考察」を参照してください。

アドレス検証

UBE メッセージは、しばしば不正発信者のアドレスを使用します。Messaging Server SMTP サーバーは、メッセージを不正な発信者のアドレスと照合させることで、これを利用できます。発信者のアドレスが DNS サーバーに対するクエリにより有効なホストネームに対応していないと判断された場合、そのメッセージは拒否されます。ただし、DNS をこのように使用する場合は、パフォーマンスが低下する可能性があります。

『Sun Java System Messaging Server 6 2005Q4 管理ガイド』で説明されているチャネルキーワード `mailfromdnsverify` を使用して、チャネル別ベースのアドレス検証を有効にします。

Real-time Blackhole List

Mail Abuse Protection System の Real-time Blackhole List (MAPS RBL) は、発信元 IP アドレスによって識別された既知の UBE 発信元のリストを動的に更新します。Messaging Server SMTP サーバーは MAPS RBL をサポートしており、MAPS RBL が UBE の発信元として認識した IP アドレスからのメッセージ受信を拒否できます。MAPS RBL は、インターネット DNS を使用した無料サービスです。

詳細については、次を参照してください。

```
http://mail-abuse.org/rbl
```

MTA Dispatcher の ENABLE_RBL オプションを使用すると、Messaging Server SMTP サーバーで RBL を有効にできます。

リレーブロッキング

総合的な UBE 対策としては、アクセス制御、メールボックスフィルタリング、アドレス検証、RBL を使用して UBE を受け取らないようにする対策と、システムが不正に利用されてメールをほかのシステムにリレーしてしまうことを防ぐ対策が必要です。後者は、リレーブロッキングと呼ばれます。リレーブロッキングの最も単純な方法は、非ローカルシステムからのリレーを拒否しながら、ローカルユーザーとシステムにはメールのリレーを許可することです。IP アドレスを選別の基準として使用すると、ローカルと非ローカルを簡単かつ安全に判断できます。デフォルトでは、Messaging Server はインストール時にリレーブロッキングを行うように設定されます。詳細については、196 ページの「マッピングテーブルによるリレー防止設定」を参照してください。

認証サービス

Messaging Server の SMTP サーバーには Simple Authentication and Security Layer (SASL、RFC2222) が実装されています。SASL は POP クライアントと IMAP クライアントで使用することができ、SMTP サーバーへのパスワードベースのアクセスを提供しています。SASL の一般的な使用方法は、認証を受けた外部ユーザーにメールのリレーを許可することです。これにより、自宅からまたは出張中に ISP を使用するローカルユーザーに共通の問題が解決されます。そのようなユーザーは、メールシステムに接続するときに、ローカルとは異なる IP アドレスを使用します。発信元 IP アドレスのみを考慮するリレーブロックでは、これらのユーザーのメールはリレーされません。この問題は、SASL を使用してこれらのユーザーの認証を可能にすることで解決できます。一度認証を受けたユーザーは、メールのリレーが許可されます。

サイドライニング

前述したアクセス制御のメカニズムでは、疑わしいメッセージの処理を保留しておき、あとで手動で検査することもできます。あるいは保留する代わりに、送信先アドレスを変更して疑わしいメールを特定のメールボックスに配信したり、警告なしで削除したりすることもできます。この対策は、UBE が既知の固定された発信元から送られてきたものである場合に有効で、これを完全に受信拒否してしまうと、悪用者が発信元を変更してしまうだけの結果となってしまいます。Messaging Server のメーリングリストでも同様の機能を使用できます。警告なしでメールを削除する場合には、正当な送信者が影響を受けないよう慎重に行う必要があります。

総合追跡

Messaging Server の SMTP サーバーは、すべての受信メールメッセージに関する重要な発信元情報を検出し、記録します。この情報には、発信元 IP アドレスとそれに対応するホスト名が含まれます。検出されたすべての情報は、設定によりログファイルとともにメッセージの追跡フィールド (たとえば、Received: ヘッダー行) に記録されます。そのような信頼性の高い情報を利用できることは、ヘッダが詐称されることの多い UBE の発信元を突き止めるのに重要なことです。各サイトでは任意のレポートツールを使用して、プレーンテキストで保存されているこの情報にアクセスできます。

変換チャネル

変換チャネルは非常に汎用的な目的で使われるインタフェースです。チャネル上でスクリプトやプログラムを呼び出して、電子メールメッセージの本文を任意に処理できます。変換プログラムは、それぞれの MIME のメッセージ全体ではなく本文をプログラムまたはスクリプトに渡し、その本文をプログラムまたはスクリプトの出力に置き換えます。変換チャネルは、テキスト形式から PostScript 形式へというようにファイル形式を変換したり、ある言語を別の言語に変換したり、会社の機密情報のためにコンテンツフィルタリングを実行したり、ウイルスを検索したり、メッセージを別のものに置き換えたりするのに使用できます。

サードパーティー製品との統合

Messaging Server の変換チャネルを使用すると、サードパーティーの供給元が提供するコンテンツフィルタリングソフトウェアを配備に統合できます。チャネルキーワードは、Brightmail または SpamAssassin のようなスパム防止およびウイルス対策製品を使用したメールフィルタリングを行うのに使用されます。MTA を設定して、すべてのメッセージまたは特定のチャネルを経由するメッセージのフィルタリングを行なったり、ユーザー別のレベルでフィルタの精度を設定したりできます。スパム防止とウイルス対策のいずれか、または両方の使用を選択できます。SpamAssassin はスパムのフィルタリングのみを行います。

Sieve の広範囲なサポートにより、スパムやウイルスであると判定されたメッセージの処理設定に大きな柔軟性を持たせることが可能となりました。ウイルスとスパムの削除をデフォルトの動作とするか、スパムを特定のフォルダに集めることができます。ただし、Sieve を使用する場合は、メッセージのコピーを特別なアカウントに転送するか、カスタムヘッダーを追加するか、spamtest Sieve 拡張を使用して、SpamAssassin から返されるレイティングに基づいて異なる動作を行うことができます。

スパム防止およびウイルス対策の考察

この節では、スパム防止またはウイルス対策の技術を使用した配備を計画する場合の留意事項について説明します。

スパム防止およびウイルス対策を配備する場合のアーキテクチャー上の問題

Messaging Server MTA は、Brightmail や SpamAssassin のようなメールフィルタリングシステムと同じシステムでも、別のシステムでも使用することができます。MTA をメールフィルタリングサーバーから分離することのメリットは、ハードウェアを追加してサーバーのクローンを使用すれば、簡単にフィルタリングの処理能力を上げられることです。システムの能力に余裕があり、過負荷状態になっていない場合は、メールフィルタリングサーバーソフトウェアを MTA と同じサーバーに置くことができます。

一般には、MTA がメールのフィルタリングに使用する Brightmail サーバーの「ファーム」を配備することを検討します。MTA が Brightmail サーバー名のリストを使用するように設定すると、MTA の負荷を分散できます。この負荷分散機能は、Brightmail SDK により可能になります。Brightmail サーバーのファームを導入するメリットは、より多くの処理パワーが必要な場合に、Brightmail サーバーを追加するだけで対応できることです。

メールフィルタリング製品は、一般に高い CPU 占有率を要求します。MTA とメールフィルタリング製品をそれぞれ専用のマシンに分けるアーキテクチャーを構築することで、メッセージング配備の全体的なパフォーマンスを向上させることができます。

注 – メールフィルタリングサーバーの CPU 占有率が高い傾向にあるため、フィルタリングの対象となる MTA ホスト以上の数のメールフィルタリングシステムを使用するアーキテクチャーに行き着く場合もあります。

大規模な配備では、それぞれのインバウンドメールとアウトバウンドメールの MTA プールに対応する、サーバーのインバウンドおよびアウトバウンドフィルタリングプールを構築することも検討します。また、「スイング」プールを構築して、必要とされる状況に応じてインバウンド、アウトバウンドのいずれかのプールとして機能させることもできます。

その他の配備全般と同様に、メールフィルタリング層を常時監視する必要があります。経験上、CPU 占有率 50% をしきい値とするのが良い指針です。このしきい値に達したら、メールフィルタリング層の能力増強を検討する必要があります。

RBL の実装

一般的には、RBL を実装するとすぐにスパムを減らすことができます。MTA によって RBL が実装されれば、スパムを少なくとも 10% 以上、すぐに減らすことができます。この数字が 50% にまで達する場合もあります。

RML と Brightmail とは併用できます。Brightmail が一定の時間内に特定の IP アドレスの電子メール 100 件のうち 95 件を処理した場合、その IP アドレスを RBL に追加する必要があります。Brightmail の分析を行う際、Brightmail のメール判定基準のために RBL を調整できます。この調整により、RBL はスパムが集中する場合の処理に十分に備えられます。

スパム防止およびウイルス対策配備の一般的なシナリオ

この節では、Brightmail および SpamAssassin の一般的な配備例について説明します。詳細については、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』を参照してください。

Symantec Brightmail の使用

Symantec Brightmail には、次の一般的な配備シナリオがあります。

- ローカルメッセージストア (`ims-ms` チャネル) に届く受信メッセージの処理
- インターネット (`tcp-local` チャネル) に送られるメッセージの処理
- インターネット (`tcp-local` チャネル) から届くメッセージの処理
- 特定のドメインに送られるメッセージの処理 (`per-domain` オプション)
- 特定のユーザーに送られるメッセージの処理 (`per-user` オプション)
- Class-of-Service オプションとしての Brightmail 処理の設定

Brightmail がスパムとウイルスの両方のチェックを実行する場合、MTA のメッセージスループットは 50% ほど低下する可能性があります。MTA のスループットを維持するには、各 MTA につき 2 台の Brightmail サーバーが必要です。

SpamAssassin の使用

Messaging Server では、SpamAssassin の使用がサポートされています。SpamAssassin はフリーウェアのメールフィルタで、スパムの特定に使用されます。SpamAssassin は、Perl や一連のアプリケーションで記述されたライブラリと、SpamAssassin のメッセージングシステムへの統合に使用するユーティリティーで構成されています。

SpamAssassin では、すべてのメッセージのスコアが計算されます。スコアは、メッセージヘッダーや本文の情報に対して一連のテストを実行することによって計算されます。各テストに成功するか失敗するかによってスコアは調整されます。スコアは正または負の実数です。スコアが一定のしきい値 (通常 5.0) を超えると、スパムであるとみなされます。

SpamAssassin には高い設定性があります。テストはいつでも追加したり削除したりでき、既存テストのスコアは調整できます。これらはすべてさまざまな設定ファイルを通じて実行されます。SpamAssassin の詳細については、SpamAssassin の Web サイトを参照してください。

`http://www.spamassassin.org`

Brightmail のスパムおよびウイルススキャンライブラリに接続する場合と同じ方法で SpamAssassin `spamd` サーバーに接続できます。

Symantec AntiVirus Scan Engine (SAVSE) の使用

Messaging Server は SAVSE の使用をサポートします。SAVSE は、TCP/IP サーバーアプリケーションおよび通信用の API であり、高性能なウイルススキャニングを提供します。SAVSE は、ネットワークインフラストラクチャーデバイス経由で送受信されるトラフィックやそれらのデバイス上に格納されているトラフィックを保護するように設計されています。

スパム防止およびウイルス対策のためのサイトポリシーの開発

スパムとスパムのリレーを防止するためのポリシーを開発する場合には、スパムの防止機能と電子メールがサイトにタイムリーに配信されることのバランスを取る必要があります。したがってベストのポリシーは、処理時間があまり長くない判定基準をコアとして最初に配置して、スパムの大半を捕捉することです。最終アーキテクチャーでストレステストを行なったあとに、この判定基準のコアセットを定義できます。最初の判定基準は次の内容で始めます。システムを配備したら、捕捉されたスパムと捕捉されなかったスパムを分析してシステムの微調整を行い、必要に応じて機能を入れ替えたり、新機能を追加したりします。

サイトのスパム防止およびウイルス対策ポリシーの開始点として、次の判定基準を使用します。

- リレー防止は、`ORIG_SEND_ACCESS` の設定により行います。この構造により、登録者とパートナーのユーザーだけがアクセスを許可され、SMTP 経由でメールを外部に送信できます。

- 認証サービスを使用して、ローミングユーザーを検証します。これらのユーザーは、識別情報が確認された後で SMTP 経由の外部への送信を許可されます。
- システム全体のメールボックスフィルタを使用して、件名行をチェックしてスパムに共通の言い回しをチェックする機能を実装します。
- `holdlimit` キーワードを使用して、メール受信者の最大数を設定します。これにより、スパムの可能性のあるトラフィックを保留にできます。受信者数の初期値を 50 に設定しておいて、ある程度の期間監視を続けてから、必要に応じて増減します。
- ポストマスターがマニュアルで利用する、専用のダミーアカウントを設定して、そのアカウントに送られてくるスパムを監視して新しいスパムサイトをつきとめます。
- ウイルスが検出されたメッセージを送信者に返送すべきではありません。そして、受信対象となっているユーザーにも転送すべきではありません。それが無意味である理由は、このようなメッセージでは、ウイルスがメールを生成し、送信者アドレスを偽造しているからです。ウイルスに感染しているメッセージが重要なものであることはきわめて稀です。
- 感染しているメッセージは、ウイルスに関する情報を収集してリスト化するウイルス対策エンジンに送ります。そのような情報を利用して、システム管理者に新しいウイルスやワームの発生を通知するレポートを作成できます。

第 15 章

Messaging Server インストール前の考慮事項と手順について

この章では、Messaging Server をインストールする前に検討しなければならない考慮事項と、実行しなければならない手順について説明します。Java Enterprise System インストーラの実行手順については、『Sun Java Enterprise System 2005Q4 Installation Guide for UNIX』を参照してください。

この章には、次の節があります。

Messaging Server インストールの考慮事項

この節では、Messaging Server のインストールの準備のための考慮事項について説明します。

- リソースの競合**:** サーバー間のリソースの競合を回避するには、Messaging Server をインストールするホストとは別のホストに Directory Server をインストールすることを検討してください。
- インストール権限**:** Messaging Server をインストールするには、ルート としてログオンする必要があります。
- **Messaging Server** ベースディレクトリ**:** Messaging Server は、*msg_svr_base*と呼ばれるディレクトリ (たとえば `/opt/SUNWmsgsr`) にインストールされます。このディレクトリは、既知のファイル配置構造 (ファイルディレクトリパス) を持っています。

- サーバーのアップグレード**:** Messaging Server ホストにその他のコンポーネント (Web Server、Directory Server、Access Manager、および管理サーバー) をインストールしない場合は、これらのコンポーネントのアップグレードは不要で、Messaging Server は問題なく動作します。同じマシンにその他のコンポーネントがインストールされている場合は、Messaging Server とともにそのコンポーネントをアップグレードする必要があります。
- ポート番号の競合**:** 同じマシンに特定の製品をインストールすると、ポート番号の競合が起こる場合があります。次の表は、ポート番号が競合する可能性についてまとめたものです。

表 **15–1** 可能性のあるポート番号の競合

ポート番号の競合	コンポーネント	コンポーネント
143	IMAP サーバー	MMP IMAP プロキシ
110	POP3 サーバー	MMP POP3 プロキシ
993	SSL を使用した IMAP	SSL を使用した MMP IMAP プロキシ
80	Access Manager (Web サーバーのポート)	Messenger Express

可能であれば、ポート番号が競合する製品は別のホストにインストールします。それができない場合は、競合する製品のいずれかでポート番号を変更する必要があります。ポート番号を変更するには、`configutil` ユーティリティーを使用します。手順については、『Sun Java System Messaging Server 6 2005Q4 Administration Reference』を参照してください。

次の例では、`configutil` の `service.http.port` パラメータを使用して、Messenger Express の HTTP ポート番号を `8080` に変更しています。

```
configutil -o service.http.port -v 8080
```

Messaging Server インストール用ワークシート

Messaging Server をインストールするときに、次のインストールワークシートを使用して記録をつけておくと、インストールプロセスで役立ちます。これらのインストールワークシートは、Messaging Server を何度もインストールしたり、アンインストールしたり、アップグレードのためにアップグレードしたりする際に再使用できます。

ヒント – インストール中に指定したすべてのポート番号と、そのポート番号を使用する特定のコンポーネントを記録しておきます。

ワークシートには次のものがあります。

- 221 ページの「Directory Server インストール用ワークシート」
- 222 ページの「管理サーバー初期実行時設定用ワークシート」

Directory Server インストール用ワークシート

Java Enterprise System インストーラ、または以前の Directory Server インストールにより、Directory Server をインストールできます。Directory Server のインストール情報と設定パラメータを表 15–2 に記録します。管理サーバーと Messaging Server のインストールと設定を行うときや、Messaging Server の実行時初期設定を行うときに、これらのパラメータが必要となります。その他の情報については、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』を参照してください。

表 **15–2** Directory Server インストールパラメータ

パラメータ	説明	例	設定対象	実際の設定値
Directory Installation Root	Directory Server ホストにあるディレクトリで、サーバープログラム、設定、保守、および情報のファイルの格納専用に使用されます。	`/var/opt/mps/serverroot`	`comm_dssetup.pl` Perl スクリプト	
ホスト	完全修飾ドメイン名。完全修飾ドメイン名は、ホスト名とドメイン名の 2 つの部分から構成されます。	`svr1.west.sesta.com`	管理サーバー設定	
LDAP Directory Port Number	LDAP ディレクトリサーバーのデフォルトは `389` です。	`389`	管理サーバー設定と Messaging Server 設定	

表 **15–2** Directory Server インストールパラメータ　(続き)

パラメータ	説明	例	設定対象	実際の設定値
Administrator ID and Password	設定情報を担当する、または設定情報に責任を持つ管理者。 管理者のパスワード。	`AdminPaSsWoRd`	管理サーバー設定	
User and Group Tree Suffix	ユーザーとグループのデータが格納されるディレクトリツリーの最上部の LDAP エントリの識別名。	`o=usergroup`	`comm_dssetup.pl` Perl スクリプト	
Directory Manager DN and Password	UNIX の ルート に相当する権限を持つディレクトリ管理者。通常この管理者は、ユーザーとグループのデータに責任を持ちます。 ディレクトリマネージャのパスワード	`cn=Directory Manager` `pAsSwOrD`	`comm_dssetup.pl` Perl スクリプトと Messaging Server 設定	
Administration Domain	管理制御の対象範囲。	`System Lab`	管理サーバー設定	

管理サーバー初期実行時設定用ワークシート

Java Enterprise System インストーラを通じて管理サーバーの初期実行時設定プログラムを実行するときは、インストールパラメータを次の表に記録します。これらのパラメータの中には、Messaging Server 初期実行時設定に必要なものがあります。いくつかの質問項目については、221 ページの「Directory Server インストール用ワークシート」も参照してください。

表 **15–3** 管理サーバー初期実行時設定プログラムのパラメータ

パラメータ	説明	例	実際の設定値
Fully Qualified Domain Name	ホストマシンの完全修飾ドメイン名。	`svr1.west.sesta.com`	

表 **15–3** 管理サーバー初期実行時設定プログラムのパラメータ　(続き)

パラメータ	説明	例	実際の設定値
Server Root Definition	管理サーバーがインストールされる root ディレクトリで、サーバープログラム、設定、保守、および情報ファイルの格納専用に使用されます。	`/var/opt/mps/serverroot`	
UNIX System User	システムユーザーに特定の権限を指定することで、ユーザーが実行するプロセスに適切な許可を与えることができます。値は常に `root` となります。	`root`	
UNIX System Group	特定の UNIX ユーザーが属するグループ。値は常に `other` となります。	`other`	
Configuration Directory Server	221 ページの「Directory Server インストール用ワークシート」で指定されるホストとポート。	Host `svr1.west.sesta.com` Port `390`	
Configuration Directory Server Administrator and Password	221 ページの「Directory Server インストール用ワークシート」で指定される管理者 ID。 管理者 ID のパスワード。	`Admin` `PaSsWoRd`	
Administration Domain	管理制御の対象範囲。 Messaging Server と Directory Server を同じマシンにインストールした場合は、221 ページの「Directory Server インストール用ワークシート」で同じ管理ドメインを選択する必要があります。	System Lab2	
Administrative Server Port	管理サーバー専用の固有のポート番号。	`5555`	

設定する Messaging Server コンポーネントの選択

Messaging Server ソフトウェアをインストールするときに、Java Enterprise System インストーラによりすべての Messaging Server がインストールされます。次に、Messaging Server 設定プログラムを使用して、Messaging ホスト上で適切なMessaging Server コンポーネント (MTA、メッセージストア、Messenger Express、MMP) を選択します。

次の表は、それぞれのタイプの Messaging ホストで設定する必要のあるコンポーネントを示します。

表 **15–4** Messaging Server で設定するコンポーネントの選択

設定するメッセージングホストのタイプ	設定プログラムで選択されるコンポーネント
MTA	メッセージ転送エージェント
メッセージストア (バックエンド)	メッセージ転送エージェント、メッセージストア、Messenger Express 注: 設定の終了後、MEM プロキシの保存を設定する必要があります。
Messenger Express (フロントエンドのみ、保存または SMTP の機能なし)	Messenger Express、Messaging マルチプレクサ 注: Messenger Express だけを設定する場合は、メッセージストアと MTA を選択するか、少なくとも既存の MTA を指定します。
Messenger マルチプレクサ (フロントエンドのみ、保存または SMTP の機能なし)	Messaging マルチプレクサ

注 – LMTP 配信メカニズムを設定するには、MTA とバックエンドストアの両方の設定が必要です。LMTP の設定手順については、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』の第 15 章「LMTP 配信」を参照してください。

sendmail デーモンを無効にする

Messaging Server のインストールに先立ち、もしも sendmail デーモンが実行中であれば無効にしておくことをお勧めします。Messaging Server SMTP サーバーが実行する Dispatcher には、ポート 25 を割り当てる必要があります。ポート 25 で sendmail デーモンが実行されていると、Dispatcher をポート 25 に割り当てることができません。

▼ sendmail デーモンを無効にするには

手順

1. **/etc/init.d** ディレクトリに移動します。

   ```
   cd /etc/init.d
   ```

2. **sendmail** が実行されている場合は、停止します。

   ```
   ./sendmail stop
   ```

3. **/etc/default/sendmail** に **MODE=""** を追加します。

   sendmail ファイルが存在しない場合は、ファイルを作成し、MODE="" を追加します。

   この修正の追加は、ユーザーが誤って sendmail start を実行したり、パッチにより sendmail が再起動されたりする場合に、sendmail がデーモンモードで起動するのを防ぎます。

注 – 場合によっては (特に Solaris 10 上において)、/etc/init.d/sendmail stop コマンドを実行したあとでも、sendmail が自動的に再起動されます。その場合は、次のコマンドを使って sendmail プロセスを停止します。

```
svcadm disable network/smtp:sendmail
```

パート III Calendar Server の配備

この部には、次の章があります。

第 16 章

Calendar Server ソフトウェアの紹介

この章では、Sun Java System Calendar Server の概要、Calendar Server の配備が役立つビジネス上の理由、および配備プロセスについて説明します。

この章には、次の節があります。

Calendar Server の概要

Sun Java System Calendar Server (旧称 Sun™ ONE Calendar Server) は、高性能な、インターネット標準ベースのカレンダサーバーで、中規模および大規模な企業から、さらに非常に大規模な電気通信およびインターネットのサービスプロバイダまで各ユーザーの必要に対応したスケーラビリティーを考慮し設計されています。ネイティブな Web ブラウザインタフェースまたはコネクタを使用して、Microsoft Outlook などの他のカレンダクライアントに接続することで Calendar Server は、家庭または職場のコンシューマにグループスケジュール機能および個人用のカレンダ機能を提供すると同時に、インターネットを介して他のユーザーとのカレンダ情報の共有を可能にします。ユーザーインタフェース (UI) をカスタマイズして、電子商取引用の Web リンク、バナー広告、ロゴ、またはカレンダサーバーユーザーのブランドなどを含めることができます。

Calendar Server は、オープンで相互運用可能かつ高性能な、業界最高レベルの時間管理およびリソース管理ソリューションです。そのスケーラビリティー、パフォーマンス、信頼性によって、ほかのソリューションに比べて低い総所有コストで、必要な機能を得ることができます。iCalendar 標準のネイティブサポートによって、ユーザーは簡単にインターネットで共有できる形式で予定をスケジュールすることができます。Calendar Server は次のような標準規格とプロトコルを採用しています。

- Internet Calendaring (iCalendar)

- iCalendar Transport-Independent Interoperability Protocol (iTIP)
- iCalendar Message-based Interoperability Protocol (iMIP)
- Extensible Markup Language (XML)
- Lightweight Directory Access Protocol (LDAP)
- HyperText Transport Protocol (HTTP)

Calendar Server のアーキテクチャーは、垂直方向 (システムごとの CPU の数を増大させる) と水平方向 (ネットワークにサーバーを追加する) の両方向で、柔軟性に富み、拡張可能で、スケーラブルです。その結果、Calendar Server は、さまざまなニーズに対応した設定が可能なサーバーから構成されるシステムと見なすことができます。スタンドアロンのカレンダサーバーとして単独で使用することもでき、さまざまなサービスをサーバー間で重複または分割させる、多くのインスタンスで設定することもできます。

Calendar Server は、プラグインを利用して外部のサービスを取得します。さらに、Calendar Serverは、LDAP ベースおよび ID ベースの配備もサポートしており、Sun Java System Access Manager (旧称 Identity Server)、Sun Java System Portal Server (旧称 Sun ONE Portal Server)、および Sun Java System Instant Messaging (旧称 Sun ONE Instant Messaging) と統合して追加の機能を提供します。

Calendar Server は次の 利点を提供します。

- Web ベースのグループスケジュール機能、空き時間 - 予定ありの検索、および企業のディレクトリの検索
- 会議室、プロジェクタ、および他のリソースの Web ベースのリソーススケジュール機能
- XML ベースのカスタマイズ機能 (配色、ログイン、ユーザーインタフェース、ロゴ、ブランド設定など)
- XML または iCalendar 形式で配信される標準ベースの予定およびカレンダデータフィードのサポート。これによって通信機能が強化され、商取引リンクやバナー広告から新しい収入の可能性が提供される
- 他のディレクトリサービス用の API を含む、ネイティブ LDAP のサポート
- Microsoft Outlook などの追加カレンダクライアントへのコネクタ。これによって、クライアントは Calendar Server 上でスケジュール機能を実行することができる
- ホストしているドメインのサポート
- システム管理、オンラインのバックアップと復元、およびデータベース全体のバックアップと復元の簡略化
- 仕事、家族、友人などの複数のカレンダのサポート
- 公開および非公開のカレンダのサポートのほか、公開、非公開、および機密の個々の予定のサポート
- カレンダの階層化表示のサポート。これによってユーザーは 2 つ以上のカレンダを 1 つの表示に統合でき、コミュニケーションや生産性を向上することができる
- 選択した受信者にアポイント、出席依頼、およびアラームの電子メール通知が自動的に送信され、Sun Java System Instant Messaging との統合でポップアップアラームが自動的に提供される

- 各カレンダで複数の所有者がサポートされ、プロジェクトチームやコミュニティーグループでコミュニケーションが簡単になり生産性が向上する
- 一次所有者の代わりに行使する他者にカレンダの所有権を委託する機能
- 日ごと、週ごと、月ごと、年ごと、および比較の各ビュー
- スケーラブルで、ネットワーク化された、サーバー間、クライアントサーバーアーキテクチャーによって、何十万ものユーザーをサポート
- Secure Sockets Layer (SSL) 暗号化、LDAP 認証、認証プラグイン、および Access Manager でアイデンティティ対応のシングルサインオン (SSO) をサポート

Calendar Server の概念の詳細については、『Sun Java System Calendar Server 6 2005Q4 Administration Guide』を参照してください。

Calendar Server 配備の設計

配備プロセスは、次の基本フェーズから構成されており、ソリューションライフサイクルと呼ばれます。

- ビジネス要件の分析
- 技術要件の分析
- 論理アーキテクチャーの設計
- 配備アーキテクチャーの設計
- 配備の実行
- 配備の運用

配備フェーズは固定的なものではなく、配備プロセスは反復して行われます。

Calendar Server やその他の Java Enterprise System コンポーネントの配備プロセスの詳細については、『Sun Java Enterprise System 2005Q4 Deployment Planning Guide』を参照してください。

Calendar Server の配備目的

Calendar Server の配備計画を開始する前に、次のことを確認してください。

組織が Calendar Server を配備するのはなぜでしょうか

次のようにいくつかの理由が考えられます。

- コストの削減:ユーザーごとの総所有コストが市販されている他のカレンダ製品を使用するより低くなります。
- 生産性の向上:カレンダユーザーは、自分の予定や作業を管理できるほか、組織内の他の職員との会議や約束を予定することができます。また、ユーザーはカレンダグループと、会議室や機器などのリソースを管理できます。さらに、カレンダを

PDA などのモバイルデバイスや Microsoft Outlook と同期させることもできます。

- スケーラビリティーおよび可用性の向上:Calendar Server は垂直と水平の両方向で拡大縮小します。組織が拡大すると、サーバーをアップグレードしたり、さらにサーバーを追加したりすることによって簡単に設定をアップグレードできます。
- セキュリティーの向上:Solaris システムに Calendar Server を配備すると、その組織は、Windows 環境でよく見られる多くのウィルスや他のセキュリティー上の危険を回避できます。
- 高可用性 **(HA)** 設定:Sun Cluster ソフトウェアとの統合によって、Calendar Server で高可用性サービスが行えるように設定できます。ソフトウェアやハードウェアの障害が発生すると、Calendar Server は二次サーバーへフェイルオーバーします。

Calendar Server 配備チーム

通常、Calendar Server の配備には、それぞれが異なる役割と責任を受け持つ何人かの人員が必要です。小規模な組織では、1 人でいくつかの役割を兼任することがあります。考慮すべき役割の中には次のものがあります。

- プログラムマネージャは、Calendar Server 配備全体を監督し、その成功や失敗に責任を持ちます。
- Calendar Server 管理者は、Calendar Server を管理するための毎日の管理業務を行い、さらに Calendar Server のインストールやアップグレードの責任者となる場合もあります。
- パフォーマンスエンジニアは、試験的配備および本稼働における配備の Calendar Server のパフォーマンスをテストおよび監視し、配備条件を満たしているかを確認します。
- 開発エンジニアリングでは、Calendar Server のアプリケーションやプラグインを記述し、必要に応じて、Calendar Server のユーザーインタフェース (UI) をカスタマイズします。
- マニュアルスペシャリストは、管理者やエンドユーザー向けにカスタマイズされたあらゆるマニュアルを執筆します。
- 教育およびトレーニングでは、実務講習と教材を開発します。
- サポートスペシャリストは、試験的配備および本稼働における配備の両方をサポートします。

Calendar Server のエンドユーザー

エンドユーザーは、Calendar Express Web クライアント、Communications Express Web クライアント、または Sun Java System Connector for Microsoft Outlook を使用することによって Calendar Server に接続できます。

サイトのエンドユーザーについて、次のことを確認します。

- サイトには、全部で何人の Calendar Server のエンドユーザーがいますか。

- エンドユーザーはどのようにして Calendar Server に接続しますか。Calendar Express を使用しますか。Communications Express を使用しますか。Microsoft Outlook を使用しますか。それともそれらのクライアントを組み合わせますか。
- 地理的な場所はいくつ含まれていますか。エンドユーザーはすべて同じまたは違うタイムゾーンにいますか。
- エンドユーザーは、毎日、同じ時間に Calendar Server にログインしますか。
- 配備では、ピーク使用時のアクティブなエンドユーザーは何人いますか。
- エンドユーザーベースはどのくらいの速さで拡張しますか。
- Calendar Server エンドユーザーに特有のパフォーマンス要件は何ですか。
- シングルサインオン (SSO) 要件は何ですか。
- ユーザーの中に、Netscape™ Calendar 4.x から移行しているユーザーがいますか。
- エンドユーザーは Sun ONE Synchronization の使用を計画していますか。
- エンドユーザー向けに Calendar Server UI のカスタマイズを計画していますか。
- サイトにプロキシサーバーの使用を計画していますか。
- サイトに負荷分散機能の使用を計画していますか。

必要とされる Calendar Server エンドユーザーのパフォーマンス

エンドユーザーに特有のパフォーマンス要件は何でしょうか。例:

- どのくらいのエンドユーザーの応答時間が許容されますか。
- ピークロード時に予想されるパフォーマンスの低下を許容できますか。

配備で使用することを計画しているのはどのような構成ですか。Calendar Server の構成シナリオには、次のものが含まれます。

- 1 つの Calendar Server インスタンス
- 単一のフロントエンドサーバーと単一のバックエンドデータベースサーバー
- LDAP CLD プラグインを用いる複数のバックエンドデータベースサーバーを持つ複数のフロントエンドサーバー
- LDAP CLD プラグインによる複数のフロントエンド/バックエンドサーバー
- 高可用性 (HA) 構成

複数のフロントエンドサーバーの設定を計画している場合、どのようにエンドユーザーの分散を計画するでしょうか。

複数のバックエンドデータベースサーバーの設定を計画している場合、どのようにデータベースの分散を計画するでしょうか。たとえば、サーバーを地理的に分散するなどの方法があります。

どのような拡張計画があるでしょうか。フロントエンドとバックエンドの両方についてはどうでしょうか。

第 17 章

Calendar Server アーキテクチャーの開発

この章では、3 つの基本的な Calendar Server 配備アーキテクチャーについて説明しています。それらは使用しているサイトに特有の要件に応じて変更することができます。

この章には、次の節があります。

単一サーバー Calendar Server アーキテクチャー

図 17–1 に、単一サーバーアーキテクチャーを示します。この配備では、すべての Calendar Server サービス (プロセス) が、1 つのサーバーの 1 つの CPU (プロセッサ) または複数の CPU で稼働します。Directory Server と Access Manager のプロセスは、同じサーバー上でも異なるサーバー上でも実行できます。

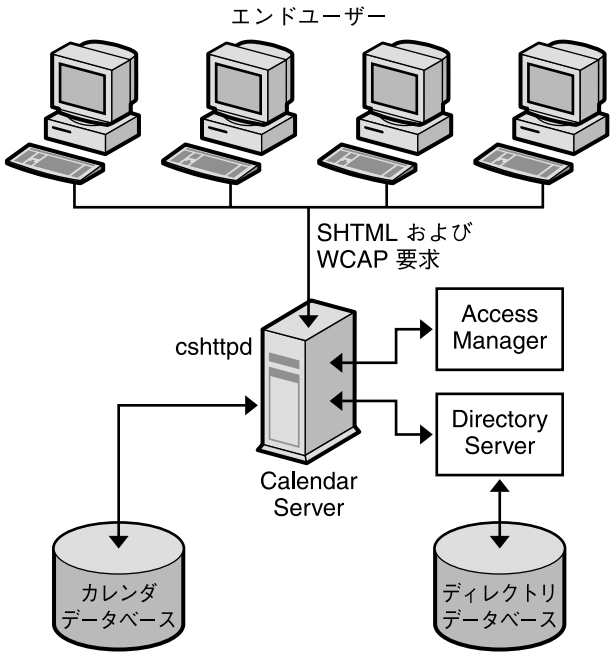

図 **17–1** 単一サーバー Calendar Server アーキテクチャー

単一サーバー上の Calendar Server インスタンスには、次のサービスが含まれます。

- 管理サービス (`csadmind` プロセス)。Calendar Server の起動と停止、カレンダユーザーまたはリソースの作成と削除、カレンダの取得と格納を行うコマンドなど、管理機能をサポートします。
- HTTP サービス (`cshttpd` プロセス)。着信した SHTML および WCAP 要求を処理します。
- 予定通知サービス (`enpd`)。予定およびアラーム通知のブローカとして機能します。
- バックアップサービス (`csstored`)。自動バックアップ (アーカイブバックアップとホットバックアップの両方) を実行します。

Calendar Server サービスの詳細については、『Sun Java System Calendar Server 6 2005Q4 Administration Guide』を参照してください。

最小構成ではデータベースは同じサーバーに配置されるため、カレンダデータベースが別のサーバーに配置されている環境でネットワーク機能を提供する DWP (データベースワイヤプロトコル) サービス (`csdwpd` プロセス) は必要ありません。

Calendar Server は、ユーザーの認証とユーザー設定の格納に使用するディレクトリサーバーを必要とします。通常は、Sun Java System Directory Server などの LDAP ディレクトリサーバーを使用します。ただし、Calendar Server API (CSAPI) を使用して、LDAP 以外のディレクトリサーバーを使用するためのプラグインを記述することもできます。この API については、『Sun Java System Calendar Server 6 2005Q4 Developer's Guide』を参照してください。

ディレクトリサーバーは、Calendar Server が稼働しているサーバーに配置することも、リモートサーバーに配置することもできます。

Sun Java System Access Manager (リリース 2003Q4 (6.1) 以降) には次の機能があります。

- **Communications Services Delegated Administrator** ユーティリティ**:**Calendar Server を含む Sun Java System コミュニケーションサーバーの、ホスト (仮想) ドメイン、ユーザー、グループ、組織、リソース、ロールをプロビジョニングおよび管理するときは、この CLI ユーティリティ (`commadmin`) を使用します。

  Communications Services Delegated Administrator ユーティリティの詳細については、『Sun Java System Communications Services 6 2005Q4 Delegated Administrator 管理ガイド』を参照してください。

- シングルサインオン **(SSO):**Access Manager の使用または信頼できるサークルテクノロジによって、Calendar Server や Messaging Server を含む Sun Java Enterprise System サーバーに SSO を実装することができます。Access Manager は、Java Enterprise System サーバーの SSO ゲートウェイとして機能します。ユーザーは Access Manager にログインすると、すべてのサーバーで SSO が適切に設定されている限り、その他のサーバーにもアクセスできます。

- **Sun Java System LDAP** スキーマ **2:** このバージョンのスキーマを利用するには、Access Manager (リリース 2003Q4 以降) が必要です。

これらのトピックの詳細については、『Sun Java System Calendar Server 6 2005Q4 Administration Guide』を参照してください。

Access Manager は、Calendar Server が稼働しているサーバーで実行することも、リモートサーバーで実行することもできます。

エンドユーザーは、2 つの Web ユーザーインタフェース (UI) の 1 つ、つまり Sun Java System Calendar Express または Sun Java System Communications Express のいずれかを使用して、クライアントマシンから Calendar Server に接続します。これらのインタフェースの使い方については、各インタフェースのオンラインヘルプを参照してください。

2 層 Calendar Server アーキテクチャー

Calendar Server は、複数のフロントエンドサーバーとバックエンドサーバーに設定を分配することにより、スケーラビリティーを実現します。各サーバーでは、Calendar Server サービスを複数の CPU に分散させることもできます。

次の図に示す 2 層アーキテクチャーはネットワークフロントエンド / データベースバックエンド構成とも呼ばれ、ユーザーはフロントエンドサーバーにログインし、DWP (データベースワイヤプロトコル) サービス (`csdwpd` プロセス) を使用してバックエンドサーバーに接続します。カレンダデータベースは、バックエンドサーバーだけに接続されています。

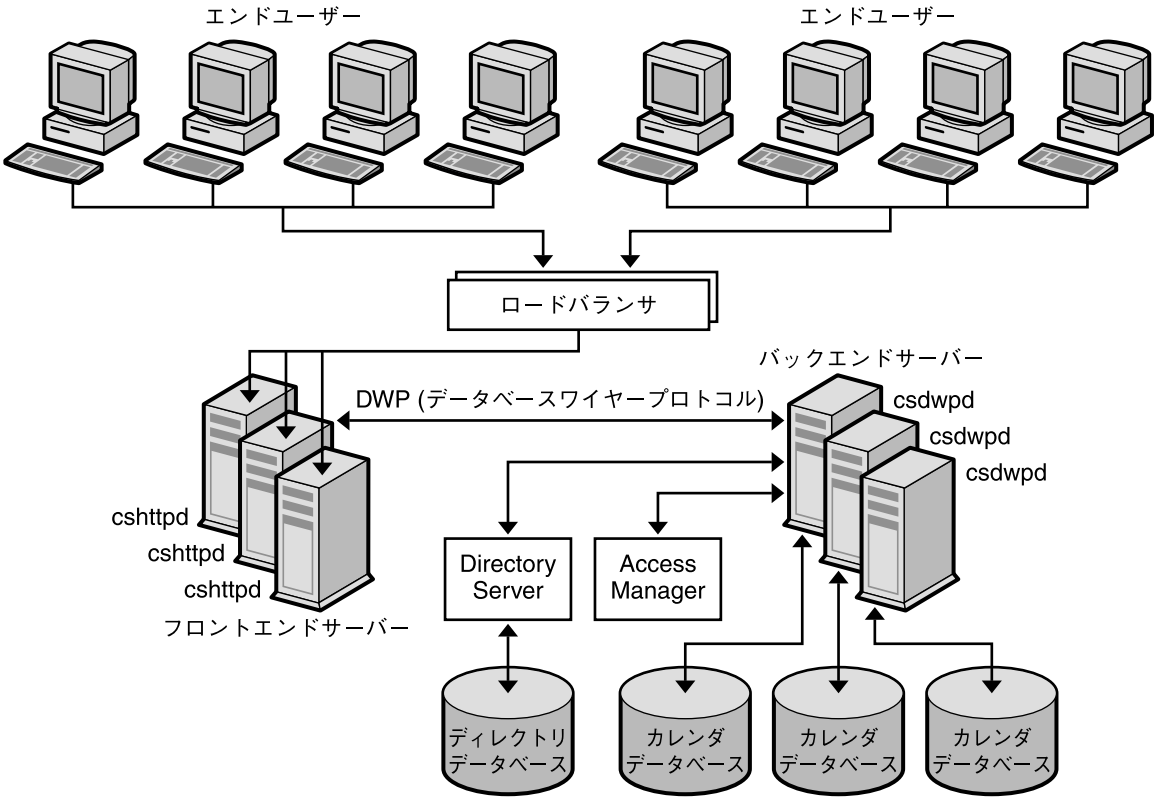

図 **17–2** 2 層 Calendar Server アーキテクチャー

フロントエンドサーバーとバックエンドサーバーの両方で実行される Calendar Server プロセスは次のとおりです。

- ユーザーはロードバランサによってフロントエンドサーバーに誘導され、そこでログインします。それぞれのフロントエンドサーバーは次のサービスを必要とします。
  - 管理サービス (`csadmind` プロセス)
  - HTTP サービス (`cshttpd` プロセス)
- 各バックエンドサーバーにはカレンダデータベースが接続されるため、各バックエンドサーバーは次のサービスを必要とします。
  - 管理サービス (`csadmind` プロセス)
  - バックアップサービス (`csstored`)
  - 予定通知サービス (`enpd` および `csnotifyd` プロセス)
  - カレンダデータベース用にフロントエンドサーバーにネットワーク機能を提供する DWP (データベースワイヤプロトコル) サービス (`csdwpd` プロセス)

この構成では、ユーザーはバックエンドサーバーにログインしないため、HTTP サービス (`cshttpd` プロセス) は必要ありません。

Calendar Server サービスの詳細については、『Sun Java System Calendar Server 6 2005Q4 Administration Guide』を参照してください。

スケーラブルな Calendar Server の構成には、ユーザーの認証とユーザー設定の格納に使用するディレクトリサーバーが必要です。

Access Manager (リリース 6.1 (リリース 6 2003Q4) 以降) を使用して、シングルサインオン (SSO) の実装、Sun Java Enterprise System LDAP スキーマ 2 の使用、ホスト (仮想) ドメイン、ユーザー、グループ、組織、リソース、ロールのプロビジョニングと管理を行うことができます。

エンドユーザーは、2 つの Web ユーザーインタフェース (UI) の 1 つ、つまり Sun Java System Calendar Express または Sun Java System Communications Express のいずれかを使用して、クライアントマシンから Calendar Server に接続します。これらのインタフェースの使い方については、各インタフェースのオンラインヘルプを参照してください。

複数サーバーの 2 層 Calendar Server アーキテクチャー

複数のフロントエンドサーバーやバックエンドサーバーを使用した 2 層 Calendar Server アーキテクチャー (図 17–3) では、ユーザーは特定のサーバーにログインし、各サーバーはカレンダデータベースに接続されています。この構成では、カレンダを物理的に配布することができます。各サーバーにはカレンダが配置され、その所有者が Calendar Server にログインします。

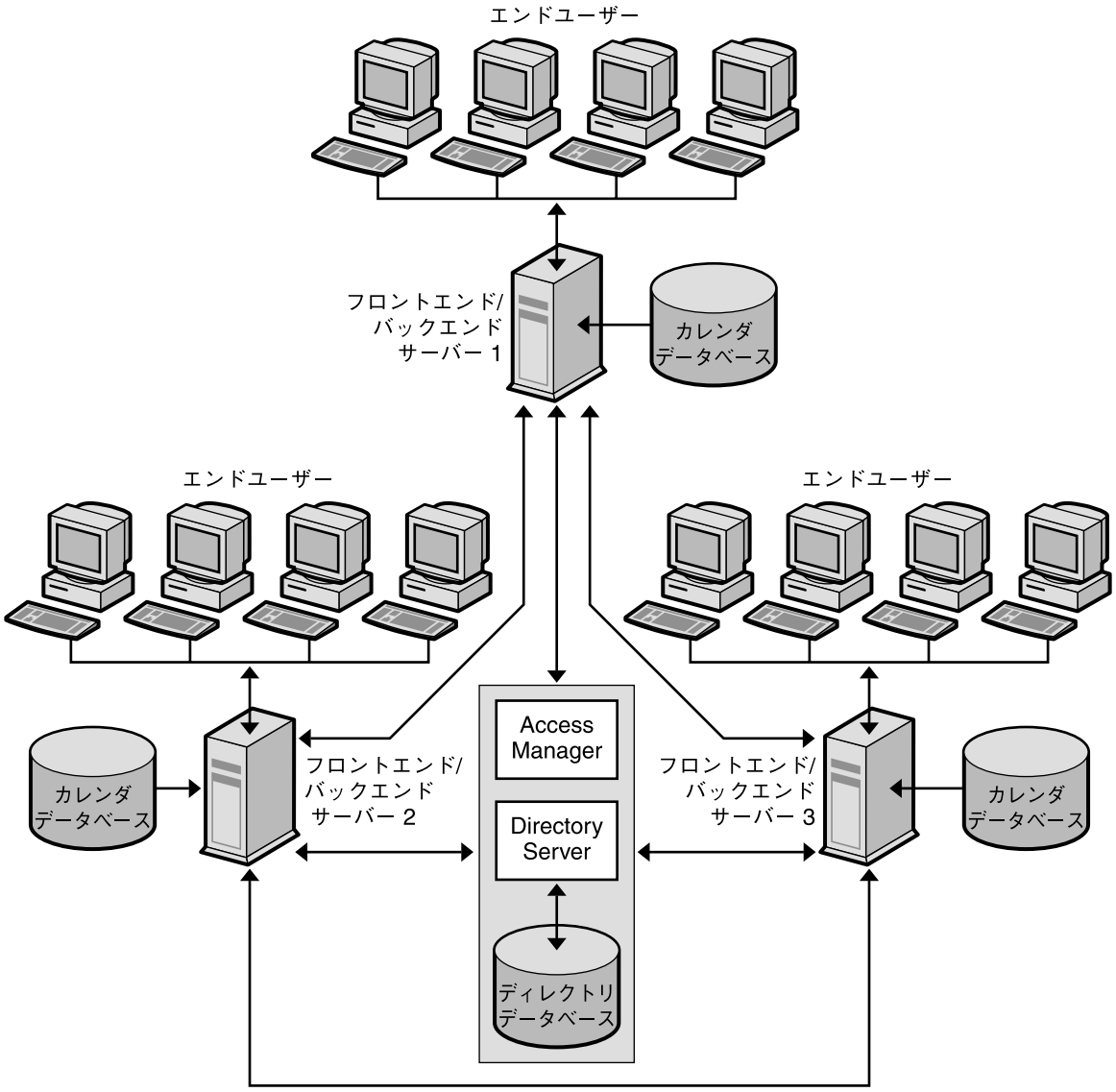

図 **17–3** 複数サーバーの 2 層 Calendar Server アーキテクチャー

このアーキテクチャーでは、どのサーバーもフロントエンドとしても、バックエンドとしても機能し、すべての Calendar Server サービス、つまり、管理サービス (`csadmind` プロセス)、HTTP サービス (`cshttpd` プロセス)、予定通知サービス (`enpd` プロセスおよび `csnotifyd` プロセス)、およびDWP (データベースワイヤプロトコル) サービス (`csdwpd` プロセス) を必要とします。

Calendar Server サービスの詳細については、『Sun Java System Calendar Server 6 2005Q4 Administration Guide』を参照してください。

注 – このアーキテクチャーでは、フロントエンドサービスをバックエンドサービスとは別のマシンに配置することも可能で、LDAP Calendar Lookup Database (CLD) を使用してフロントエンドがどのバックエンドからデータを取得する必要があるかを判別できます。詳細については、『Sun Java System Calendar Server 6 2005Q4 Administration Guide』を参照してください。

複数のフロントエンド / バックエンドサーバーの配備には、ユーザーの認証とユーザー設定の格納に使用するディレクトリサーバーが必要です。

Access Manager (リリース 6.1 (リリース 6 2003Q4) 以降) を使用して、シングルサインオン (SSO) の実装、Sun Java Enterprise System LDAP スキーマ 2 の使用、またはホスト (仮想) ドメイン、ユーザー、グループ、組織、リソース、ロールのプロビジョニングと管理を行うことができます。

エンドユーザーは、2 つの Web ユーザーインタフェース (UI) の 1 つ、つまり Sun Java System Calendar Express または Sun Java System Communications Express のいずれかを使用して、クライアントマシンから Calendar Server に接続します。これらのインタフェースの使い方については、各インタフェースのオンラインヘルプを参照してください。

第 18 章

Calendar Server セキュリティーの計画

この章では、Calendar Server 配備のさまざまなコンポーネントに対する計画を立案し、それらのコンポーネントを保護する方法について説明します。

この章には、次の節があります。

Calendar Server セキュリティーの概要

セキュリティーは、今日のビジネスにおける日々の業務の中で重要な役割を果たしています。セキュリティーの侵害は、企業秘密を危険にさらす可能性だけでなく、停止時間、データの破損、および運用コストの増大を招く可能性もあります。Calendar Server は、ユーザーを盗聴、不許可の使用、または外部からの攻撃から保護するために多くのセキュリティーレベルを提供します。基本レベルのセキュリティーは認証によるものです。Calendar Server は、デフォルトの設定で LDAP 認証を使用していますが、代替の認証方法が必要とされる場合、認証プラグインの使用もサポートしています。さらに、Access Manager と統合する場合、Calendar Server はそのシングルサインオン機能を利用することができます。

セキュリティーにはユーザーが本人であることを確かめる以上のことが関係しています。セキュリティーにはデータの機密保護を確実にすることも含まれます。このため、Calendar Server はログインまたはログインとデータに対して、SSL 暗号化技術の使用をサポートしています。つまり、Web クライアントからサーバーへ、ログインのみが暗号化される場合と、ログインを含むセッション全体が暗号化される場合があります。

Secure Remote Access との統合によっても SSL 暗号化が可能になりますが、その場合、プロキシゲートウェイを介することになります。さらに、ポータルゲートウェイとの統合により提供される URL 書き換え機能により、外部のエンティティからさらに Calendar Server を隔離することが可能になります。Calendar Server は、ゲート

ウェイを介さない Calendar Server と直接接続ができないようにする、ポータルゲートウェイとともに配備することができます。この場合、すべての URL が書き換えられるため、Calendar Server の本当の URL の特定が困難になります。ユーザーが認証される場合でも、それによって、そのユーザーに他のカレンダユーザーのデータへのアクセス権があるわけではありません。

カレンダドメインの中には、認証されたユーザーが他の認証されたユーザーのカレンダデータへの不正にアクセスするのを防ぐ、他のセキュリティー層があります。セキュリティーの方策の 1 つに、Calendar Server アクセス制御のエントリによる方法があります。アクセス制御によって、カレンダユーザーは、自分のカレンダを閲覧可能な人、予定を自分のカレンダにスケジュール設定可能な人、自分のカレンダを変更可能な人、自分のカレンダから予定を削除可能な人を指定することができます。さらにアクセス制御によって、ユーザーは、自分の代わりに出席依頼に応答することのできる人、予定のスケジュール設定または変更ができる人、予定を削除できる人を選択することができます。最後に、アクセス制御を使用すると、ユーザーのドメインの範囲を調整できるので、あるドメインのユーザーが別のドメインのユーザーによる予定のスケジュール設定を防止したり、または可能にしたりすることができます。

ただし、アクセス制御に加えて、Calendar Server は、カレンダフロントエンドとデータベースバックエンドを分割する配備に対して、データベースプロトコルレベルにおける追加レベルのセキュリティーを提供します。このセキュリティーレベルは、DWP (データベースワイヤプロトコル) 認証と呼ばれ、ユーザーの名前とパスワードのペアを利用して DWP 接続を認証します。DWP 接続が認証されるためには、ユーザー名とパスワードのペアが、フロントエンドとデータベースバックエンドの両方で同じである必要があります。

セキュリティー戦略の監視

サーバーの監視は、セキュリティー戦略で重要な位置を占めます。システムに対する攻撃を識別するには、メッセージキューのサイズ、CPU の使用率、ディスクの空き容量、ネットワークの使用率を監視します。メッセージキューサイズの異常な増大やサーバー応答時間の異常な減少から、攻撃のいくつかは識別できます。また、通常とは異なるシステムの負荷パターンや接続についても調査します。ログを毎日チェックして、異常な活動がないか調べます。

カレンダユーザー認証の計画

ユーザー認証によって、ユーザーはカレンダクライアントを介してログインし、自分のカレンダ情報を取得できます。ユーザー認証の方法には次のものがあります。

- 245 ページの「プレーンテキストと暗号化されたパスワードによるログイン」
- 245 ページの「Secure Sockets Layer (SSL) による証明書ベースの認証」

プレーンテキストと暗号化されたパスワードによるログイン

ユーザー ID とパスワードは、LDAP ディレクトリに保存されます。「最小の長さ」のようなパスワードのセキュリティー基準は、ディレクトリのポリシー要件で決定されます。パスワードのセキュリティー基準は Calendar Server の管理範囲外のものです。ディレクトリサーバーのパスワードポリシーについては、『Sun Java System Directory Server 5 2005Q1 Deployment Plannning Guide』を参照してください。

プレーンテキストパスワードと暗号化パスワードの両方のログインが使用可能です。

Secure Sockets Layer (SSL) による証明書ベースの認証

Calendar Server は SSL プロトコルを使用して、暗号化通信やクライアントとサーバー の証明書ベースの認証を行います。この節では、証明書ベースの SSL 認証について説明します。

SSL は公開鍵暗号法の概念に基づいています。TLS (Transport Layer Security) は SSL のスーパーセットとして機能しますが、名前が混同されて使われています。

高いレベルでは、SSL をサポートしているサーバーには、証明書、公開鍵、非公開鍵、証明書、鍵、およびセキュリティーデータベースが必要となります。これにより、メッセージの認証、機密、完全性が確保されます。

SSL で認証するため、カレンダクライアントはサーバーと SSL セッションを確立し、ユーザーの証明書をサーバーに送信します。その後、サーバーが、提出された証明書の信頼性を評価します。証明書の信頼性が確認されると、そのユーザーは認証済みであるとみなされます。

認証に SSL を使用する場合は、Calendar Server のサーバー証明書を取得する必要があります。この証明書は、使用するサーバーの識別情報をクライアントや他のサーバーに提供します。サーバーには複数のサーバー証明書を用意しておき、証明書自身を識別することができます。サーバーには、信頼できる認証局 (CA) の証明書を必要な数だけインストールして、クライアントの認証に使用できます。

SSL の詳細については、『Sun Java System Calendar Server 6 2005Q4 Administration Guide』を参照してください。

第 19 章

Calendar Server サービスの計画

この章では、Calendar Server サービスの計画について説明します。

この章には、次の節があります。

Calendar Server のフロントエンドサービスとバックエンドサービスの計画

Calendar Server は次の主な 6 つのサービスから構成されています。

- HTTP サービス (`cshttpd`)。HTTP 要求を待機します。HTTP サービスはユーザー要求を受け取り、データを呼び出し元に返します。
- 管理サービス (`csadmind`)。Calendar Server のそれぞれのインスタンスに必要とされます。管理サービスは Calendar Server の認証および管理を 1 ヵ所で行い、また、ほとんどの管理ツールを提供します。
- 通知サービス (`csnotify`)。電子メールまたは予定通知サービスのいずれかを使用して、予定および作業の通知を送信します。
- 予定通知サービス (`enpd`)。予定アラームのブローカとして機能します。
- 分散データベースサービス (`csdwpd`)。同じ Calendar Server システム内の複数のデータベースサーバー間でリンクを張り、分散型のカレンダストアを形成します。
- バックアップサービス (`csstored`)。自動バックアップ (アーカイブバックアップとホットバックアップの両方) を実行します。最初のバックアップはログファイルを使用したスナップショットであり、2 番目のバックアップは適用済みログファイルを使用したスナップショットです。このサービスは、`start-cal` コマンド実行

時に自動的に起動されます。ただし、インストール時には有効化されないため、このサービスが機能するように設定する必要があります。バックアップサービスを設定しなかった場合、このサービスが設定されていない旨の通知メッセージが、24 時間ごとに管理者に送信されます。

スケーラブルな Calendar Server の配備の場合、フロントエンドシステムをバックエンドサーバーとともに配備することがあります。この場合、フロントエンドシステムにはプロセッサごとに `cshttpd` デーモンのインスタンスが 1 つと、単一の管理サービスが含まれます。バックエンドサーバーには、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、および管理サービスのインスタンスが含まれます。

認証および XML と XSLT の変換の 2 つは、多大な負荷を生じさせるのカレンダサービスのアクティビティです。サービス品質の要件を満たすために CPU を追加することができます。スケーラブルな環境の場合、このような負荷の高いアクティビティはフロントエンドシステムで実行され、サービス品質の要件に対応するために、個々のフロントエンドシステムに CPU を追加、またはフロントエンドシステムを追加できるようになっています。

注 – 上記は、Communications Express Calendar クライアントを使ってカレンダにアクセスする場合には当てはまりません。Communications Express は WCAP プロトコルを使用して Calendar Server データにアクセスするため、Calendar Server インフラストラクチャーは XML/XSLT 変換を行いません。Communications Express の配備の詳細については、パート V「Communications Express の配備」を参照してください。

Calendar バックエンドサービスには、通常、Calendar フロントエンドサービスの CPU の半数が必要とされます。Calendar フロントエンドシステムによってサービス品質をサポートするには、フロントエンドの CPU の 2/3 前後を Calendar バックエンドシステムで使用する必要があります。

カレンダサービスをフロントエンドサービスとバックエンドサービスに分割することは、配備の初期の段階で考慮する必要があります。フロントエンドサービスとバックエンドサービスに別々のホスト名を割り当てることによって、カレンダサービスの機能をホストごとに分割するときに、変更が基本的には内部的に行われ、ユーザーが操作方法を変更する必要がないようにします。

通常、フロントエンドサービスのコンポーネントである Calendar Server HTTP プロセスは、CPU 時間を多く使用します。カレンダのピーク使用率を考慮して、予測されるピーク HTTP セッションに対応するため、十分なフロントエンドの処理能力を選択するようにします。通常、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って、Calendar Server フロントエンドの使用可能性が向上します。フロントエンドシステムはカレンダの持続的データを保持しないので、Sun Cluster 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、そのようなソリューションを使用する際のハードウェアの追加や管理オーバーヘッドにより、HA の Calendar Server フロントエンドへの配備のコストと時間がかかります。

注 – 本来の HA ソリューションを保証する Calendar フロントエンドの唯一の構成は、Messaging Server MTA を含む同じホストに Calendar フロントエンドを配備している場合です。ただし、この構成でも、そのようなソリューションのオーバーヘッドについては、利点がわずかなことからして、注意深く比較検討する必要があります。

Calendar Server フロントエンドのハードウェアの適切な選択は、シングルプロセッササーバーまたはデュアルプロセッササーバーです。プロセッサごとに Calendar Server `cshttpd` デーモンのインスタンスを 1 つ配備します。そのような配備によってコスト効率の良いソリューションが提供され、一定レベルの初期のクライアント並行性機能から開始し、ピーク使用率レベルがわかるにつれ、既存の構成にクライアントセッション機能を追加していくことができます。

複数のフロントエンドを配備する場合、フロントエンドサービス全体にロードを分散するにはスティッキ接続や持続的接続を備えるロードバランサが必要です。

Calendar Server バックエンド サービスは、リソースの消費で十分にバランスが取れているので、CPU あるいはディスクまたはネットワークなどの I/O のいずれにおいても、ボトルネックが形成されるという証拠はありません。このため、バックエンドのハードウェアな適切な選択は、1 つのストライプボリュームを備える SPARC サーバーになります。そのようなマシンはピーク時の大量のカレンダロードに対してかなりの容量を提供します。

要件の中に高可用性がある場合、バックエンドには持続的データが含まれているので、Calendar Server バックエンドを Sun Cluster で配備するのが妥当です。

注 – フロントエンドおよびバックエンドの Calendar Server ホストの両方を持つ構成では、すべてのホスト上で次のソフトウェアが動作している必要があります。

- 同じオペレーティングシステム環境とバージョン (つまり、Solaris SPARC、Solaris x86、Linux Red Hat などをそれぞれ実行するシステムの混在は不可)。
- 同じリリースの Calendar Server (パッチやホットフィックスのリリースを含む)。

Calendar Server LDAP データキャッシュの計画

LDAP データキャッシュオプションを使うと、LDAP データのコミット直後にそのデータが利用可能になります。LDAP ディレクトリサーバーの設定によっては、(リモート) マスターサーバーに更新を参照した後、そのマスターサーバーからローカルの LDAP ディレクトリに更新をレプリケートする必要があります。このような種類の設定では、ローカルの LDAP サーバーのコミット済みデータが利用可能になるまでに遅延が生じる可能性があります。

たとえば、サイト上にマスター / スレーブ LDAP 構成が配備されており、Calendar Server がマスター LDAP ディレクトリにスレーブ LDAP ディレクトリサーバー経由でアクセスする仕組みになっている場合、コミット済み LDAP データが利用可能になるまでにいくらかの遅延が発生しますが、LDAP データキャッシュを使えば、Calendar Server クライアントが正確な LDAP データにアクセスできるようになります。

この節では次の内容について説明します。

LDAP データキャッシュの使用に関する考慮事項

サイトで LDAP データキャッシュを設定すべきかどうかを決定するには、次のガイドラインに従います。

- サイトの Calendar Server がマスター (またはルート) LDAP ディレクトリサーバーに直接アクセスし、コミット済み LDAP データが利用可能になるまでの遅延が発生しない場合、LDAP データキャッシュを設定する必要はありません。`local.ldap.cache.enable` パラメータがデフォルトの「no」に設定されていることを確認してください。
- サイト上に250 ページの「マスター / スレーブ LDAP 構成」が配備されており、Calendar Server がマスター LDAP ディレクトリにスレーブ LDAP ディレクトリサーバー経由でアクセスする仕組みになっている場合、コミット済み LDAP データが利用可能になるまでにいくらかの遅延が発生しますが、LDAP データキャッシュを設定すれば、エンドユーザーが最新データにアクセスできるようになります。

マスター / スレーブ LDAP 構成

マスター / スレーブ LDAP 構成には、1 つのマスター (ルート) ディレクトリサーバーと、1 つ以上のスレーブ (コンシューマまたはレプリカ) ディレクトリサーバーが含まれます。Calendar Server からマスター LDAP ディレクトリサーバーへのアクセスは、直接行うことも、スレーブディレクトリサーバー経由で行うことも可能です。

- Calendar Server がマスター LDAP ディレクトリサーバーに直接アクセスする場合、LDAP データは正確であるはずなので、LDAP データキャッシュを設定する必要はありません。
- Calendar Server がマスター LDAP ディレクトリサーバーにスレーブディレクトリサーバー経由でアクセスする場合、LDAP データの変更結果は通常、LDAP レフェラル経由でマスターディレクトリサーバーに透過的に書き込まれたあと、そのデータが各スレーブディレクトリサーバーに複製されます。

上記の 2 番目のタイプの構成では、コミット済み LDAP データがスレーブディレクトリサーバー上で利用可能になるまでにいくらかの遅延が発生するため、LDAP データが不正確になるという問題が発生する可能性があります。

たとえば、Calendar Server がある LDAP データの変更をコミットしても、その新しいデータはある一定期間利用可能になりません。なぜなら、マスターディレクトリサーバーが各スレーブディレクトリサーバーを更新するのに一定の時間がかかるからです。後続の Calendar Server クライアント処理では、古い LDAP データが使用され、ユーザーに古いデータが表示されます。

スレーブディレクトリサーバーの更新遅延が短い場合 (ほんの数秒程度である場合)、クライアント側で大きな問題は生じません。しかしながら、その遅延が長い場合 (数分または数時間の場合)、その遅延時間の間、不正確な LDAP データがクライアント上に表示されてしまいます。

次の表は、マスター / スレーブ LDAP サーバー構成で Calendar Server がマスター LDAP ディレクトリサーバーにスレーブ LDAP ディレクトリサーバー経由でアクセスする場合に、遅延の影響を受ける LDAP 属性の一覧です。

表 19–1 遅延の影響を受ける Calendar Server LDAP 属性

処理	影響を受ける **LDAP** 属性
自動プロビジョニング	`icsCalendar`、`icsSubscribed`、`icsCalendarOwned`、`icsDWPHost`
カレンダグループ	`icsSet`
カレンダ作成	`icsCalendarOwned`、`icsSubscribed`
カレンダ登録	`icsSubscribed`
ユーザーオプション	`icsExtendedUserPrefs`、`icsFirstDay`、`icsTimeZone`、`icsFreeBusy`
カレンダ検索	`icsCalendarOwned`

エンドユーザーが常に最新の LDAP データにアクセスするようにするには、次の節251 ページの「マスター / スレーブ遅延問題の解決」で説明する手順に従って LDAP データキャッシュを設定します。

マスター / スレーブ遅延問題の解決

マスター / スレーブ LDAP 構成の問題は、LDAP データキャッシュを使えば解決します。なぜなら、マスターディレクトリサーバーが各スレーブディレクトリサーバーを更新し終わっていなくても、Calendar Server クライアントに最新の LDAP データが提供されるようになるからです。

ユーザーが LDAP データキャッシュを有効にした場合、Calendar Server は、コミット済み LDAP データをキャッシュデータベース (`ldapcache.db` ファイル) に書き込みます。LDAP キャッシュデータベースはデフォルトで `/var/opt/SUNWics5/csdb/ldap_cache` ディレクトリに格納されますが、必要であれば、これを別の場所に設定してもかまいません。

クライアントがある単一ユーザーの LDAP データを変更した場合、Calendar Server はその変更データを LDAP キャッシュデータベース (とスレーブディレクトリサーバー) に書き込みます。後続のクライアント処理では、キャッシュデータベースから LDAP データが取得されます。こうしたデータ取得は、単一ユーザーに対する次の処理に適用されます。

- ユーザーのログイン時の属性
- ユーザーのオプション (カラースキームやタイムゾーンなど)
- ユーザーのカレンダグループ
- ユーザーのカレンダ登録リスト

したがって、LDAP データキャッシュデータベースで実現可能な機能は、次のとおりです。

- 単一システム上のプロセス間におけるデータ整合性の維持: このデータベースは、マルチプロセッサシステム上のすべての Calendar Server プロセスから利用可能です。
- ユーザーセッションをまたがるデータの持続性: このデータベースは永続的であり、更新を必要としません。LDAP データキャッシュエントリの TTL (Time To Live) やデータベースクリーンアップ間隔を設定できます。

LDAP データキャッシュの制限

LDAP データキャッシュで実現不可能な機能は、次のとおりです。

- 複数エントリの一致が予想されるような検索におけるキャッシュ読み取り (特定の会議への出席者を検索する場合など)。このタイプの検索では、LDAP 遅延が発生します。たとえば、LDAP 検索オプションが有効になっており、かつ新しいカレンダが作成されてから遅延期間内にカレンダ検索が実行された場合、その新しいカレンダは検索結果に含まれません。
- 複数フロントエンドサーバーにまたがるキャッシュの読み書き。各フロントエンドサーバーはそれぞれ独自のキャッシュを持ち、ほかのキャッシュ内のデータにアクセスすることはありません。
- 常に同じサーバーにログインするとはかぎらないユーザーを処理する機能。そのようなユーザーに対しては、各サーバーのキャッシュ内にそれぞれ異なる LDAP データが生成されます。

LDAP データキャッシュの設定

LDAP データキャッシュを設定するには、 `ics.conf` ファイル内の対応するパラメータを設定します。詳細については、『Sun Java System Calendar Server 6 2005Q4 Administration Guide』を参照してください。

注意 – Calendar Server またはその稼働元のサーバーが正しくシャットダウンされなかった場合、`ldap_cache` ディレクトリ内のすべてのファイルを手動で削除してください。そうしないと、次回の再起動時にデータベースが破損し、問題が発生する可能性があります。

第 20 章

Calendar Server のインストール前の考慮事項について

この章では、Calendar Server のインストール前に考慮が必要な事項について説明します。

この章には、次の節があります。

Calendar Server のインストール考慮事項

Calendar Server のインストールと設定は、以前の Calendar Server リリース (2003Q4 以前のバージョン) に比べ大幅に変更されました。Calendar Server 用のスタンドアロンのインストーラはなくなりました。

まだ Calendar Server をインストールしていない場合は、Sun Java Enterprise System インストーラを使用して、2005Q4 バージョンを取得する必要があります。このインストーラを使用すると、他の Sun Java System コンポーネント製品およびパッケージもインストールできます。Java Enterprise System インストーラについては、『Sun Java Enterprise System 2005Q4 Installation Guide for UNIX』を参照してください。

Calendar Server 6 2003Q4 から Calendar Server 6 2005Q4 にアップグレードする場合のアップグレードプロセスについては、『Sun Java Enterprise System 2005Q4 アップグレードガイド』の「Java Enterprise System 2003Q4 からのアップグレード」で説明されています。

Calendar Server の旧バージョン (バージョン 5.x まで) からの移行については、『Sun Java System Calendar Server 6 2005Q4 Administration Guide』の第 4 章「Database Migration Utilities」を参照してください。

5.x 以降のバージョンからの移行については、Sun サポート担当者に連絡してください。

設定が必要な Calendar Server コンポーネント

Calendar Server ソフトウェアをインストールする際、Java Enterprise System インストーラは Calendar Server パッケージをすべてインストールします。 ついで、Calendar Server 設定プログラムを使い、適切な Calendar Server コンポーネントを Calendar ホストに設定します。

次の表では、それぞれのタイプの Calendar ホストで、設定が必要なコンポーネントを示しています。

表 **20–1** 設定が必要な Calendar Server コンポーネント

構成対象の **Calendar** ホストのタイプ	設定プログラムで選択されるコンポーネント
フロントエンド	HTTP サービスおよび管理サービス
バックエンド	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、および管理サービス

分散データベースサービス (`csdwpd`) は、バックエンドサーバー、つまりカレンダデータベースのあるサーバーにのみ必要とされ、ユーザーアクセスサービス (`cshttpd`) を提供しません。これは、カレンダデータベースのないフロントエンドサーバーには必要とされません。`csdwpd` サービスを使用することで、同じ Calendar Server 設定内のフロントエンドとバックエンドのサーバーをリンクし、分散型のカレンダストアを形成することができます。

Calendar Server の管理者の計画

Calendar Server の管理者には、次の管理者が含まれます。

- 257 ページの「Calendar Server 管理者 (calmaster)」
- 257 ページの「Calendar Server ユーザーおよびグループ」
- 257 ページの「スーパーユーザー (root)」

Calendar Server 管理者 (calmaster)

Calendar Server 管理者とは、ユーザー名とそれに関連付けられたパスワードの組み合わせのうち、Calendar Server の管理権限を付与されているユーザーのことです。たとえば、Calendar Server 管理者は Calendar Server サービスの起動と停止、ユーザーの追加と削除、カレンダの作成と削除などを実行できます。このユーザーは Calendar Server の管理権限を持ちますが、ディレクトリサーバーの管理権限を持つとは限りません。

Calendar Server 管理者のデフォルトのユーザー ID は `calmaster` ですが、Calendar Server の設定時に別のユーザーを指定することもできます。インストール後に別のユーザーを指定する場合は、`ics.conf` ファイルの `service.admin.calmaster.userid` パラメータの設定を変更します。

Calendar Server 管理者として指定するユーザー ID は、ディレクトリサーバー内の有効なユーザーアカウントである必要があります。Calendar Server の設定時に Calendar Server 管理者のユーザーアカウントがディレクトリサーバーに存在していない場合には、設定プログラムがアカウントを自動的に作成します。

`ics.conf` ファイルの Calendar Server 管理者用設定パラメータの完全なリストについては、『Sun Java System Calendar Server 6 2005Q4 Administration Guide』を参照してください。

Calendar Server ユーザーおよびグループ

Solaris システムでは、これらの特別なアカウントは Calendar Server の実行に使用されるユーザー ID とグループ ID を示しています。特別なアカウントが存在しないときは、設定プログラムによって自動的に作成されるデフォルト値 `icsuser` および `icsgroup` を使用してください。ただし、Calendar Server 設定プログラムの実行時に `icsuser` および `icsgroup` 以外の値を指定することもできます。これらの値は、それぞれ `ics.conf` ファイルの `local.serveruid` パラメータおよび `local.servergid` パラメータに格納されます。

スーパーユーザー (root)

Solaris ソフトウェアが稼働するマシン上では、Calendar Server をインストールするには `superuser` (`root`) としてログインするか、あるいは superuser になる必要があります。コマンド行ユーティリティを使用して `superuser` として実行し、Calendar Server を管理することもできます。ただし、一部のタスクについては、Calendar Server ファイルへのアクセスの問題を回避するために、`superuser` としてではなく `icsuser` および `icsgroup` (または選択した値) として実行することをお勧めします。

Calendar Server のホストしているドメインの計画

Calendar Server はホストしている (または仮想) ドメインをサポートしています。ホストしているドメインのインストールでは、各ドメインが Calendar Server の同じインスタンスを共有するため、1 つのサーバーに複数のドメインが存在できます。各ドメインはネームスペースを定義し、1 つのネームスペースではすべてのユーザー、グループ、リソースが一意です。各ドメインには、変更可能な属性とユーザー設定もあります。

ホストしているドメインのインストールおよび設定には、スキーマ 2 だけを使用してください。

ホストしているドメインをサーバーにインストールおよび設定するには、次の高レベルの手順を実行します。

1. Directory Server をインストールおよび設定します。
2. Web Server または Application Server をインストールおよび設定します。
3. Access Manager をインストールおよび設定します。

   Access Manager とともに Delegated Administrator がインストールされます。
4. Calendar Server をインストールします。
5. `comm_dssetup.pl` スクリプトを実行します。

   このスクリプトの実行手順については、『Sun Java System Calendar Server 6 2005Q4 Administration Guide』の第 2 章「Directory Preparation Script (comm_dssetup.pl)」を参照してください。
6. Communications Services Delegated Administrator を設定します。

   Communications Services Delegated Administrator ユーティリティの設定手順と使用手順については、『Sun Java System Communications Services 6 2005Q4 Delegated Administrator 管理ガイド』を参照してください。
7. デフォルトドメインおよびサイト管理者 (`calmaster`) を作成します。

   デフォルトドメインは `commadmin` の設定時に作成されます。ただし、ドメインエントリを変更して、Calendar または Mail サービスを追加する必要があります。また、サイトカレンダ管理者 (`calmaster`) を設定する必要があります。これらの 2 つのタスクの実行方法の手順については、『Sun Java System Calendar Server 6 2005Q4 Administration Guide』のパート II「Postinstallation Configuration」を参照してください。
8. Calendar Server を設定します。

   `csconfiguratior.sh` プログラムの実行手順については、『Sun Java System Calendar Server 6 2005Q4 Administration Guide』の第 3 章「Calendar Server Configuration Program (csconfigurator.sh)」を参照してください。
9. Calendar Server のホストしているドメインの設定パラメータを設定します。

設定パラメータおよびその値のリストについては、『Sun Java System Calendar Server 6 2005Q4 Administration Guide』の「Hosted Domain Configuration」を参照してください。

10. commadmin ユーティリティを使用して、ホストしているドメインをサイトに作成します。
11. commadmin ユーティリティを使用して、ホストしているドメインにユーザーおよびリソースを配置します。
12. Calendar Server サービスを起動します。

    手順については、『Sun Java System Calendar Server 6 2005Q4 Administration Guide』の「Starting and Stopping Calendar Server」を参照してください。

注 – 常時 Communications Services Delegated Administrator インタフェースを使用して、スキーマ 2 のプロビジョニングを行なってください。

スキーマ 1 のプロビジョニングツールはホストしているドメインをサポートしていません。

Calendar Server のインストール後の設定

Calendar Server ソフトウェアをインストールしたら、その設定をする必要があります。この手順は、従来インストールプロセスの一部として行われましたが、現在インストーラから分離されています。

Calendar Server をインストールしたあと、次のように Calendar Server を設定する必要があります。

1. Directory Server 設定スクリプト (comm_dssetup.pl) を実行し、Sun Java System Directory Server を設定します。
2. Calendar Server 設定プログラム (csconfigurator.sh) を実行してサイト固有の要件を設定し、新しい ics.conf 設定ファイルを作成します。ics.conf ファイルのパラメータの説明については、『Sun Java System Calendar Server 6 2005Q4 Administration Guide』の「Configuration Parameters (ics.conf) File」を参照してください。

   comm_dssetup.pl スクリプトは /opt/SUNWcomds/sbin ディレクトリに、csconfigurator.sh ユーティリティは /opt/SUNWics5/cal/sbin ディレクトリに格納されています。

   Java Enterprise System インストーラおよび Calendar Server 設定ユーティリティ (csconfigurator.sh) が実行しない、構成の設定や変更がいくつかあります。次の項目は手動で変更する必要があります。

- **DWP** および **CLD** の設定**:** `ics.conf` ファイルを編集して、CLD キャッシュオプションを有効にしてください。このキャッシュがカレンダユーザーの DWP ホストサーバー情報を格納することで、LDAP ディレクトリサーバーに対する呼び出しを減らすことができます。
- デフォルトのタイムゾーン**:**デフォルトのタイムゾーンがアメリカのニューヨークでない場合、 `ics.conf` ファイルを編集して変更してください。さらに、 `/opt/SUNWics5/cal/bin/html/default_user_prefs.xml` ファイルも変更して、`ics.conf` ファイルと同期するようにする必要があります。
- クライアント側のレンダリング**:** Calendar Server では、XSLT 処理をエンドユーザーのブラウザにダウンロードすることで、クライアント側のレンダリングを実行します。このため、Calendar Server が実行する必要のある処理は減少します。Calendar Server は、ブラウザが XSLT 処理のレンダリングに対応している場合にだけ XSLT 処理をダウンロードします。現在のリリースでは、この機能は Internet Explorer 6.0 以降だけに適用されます。`ics.conf` ファイルを編集して、クライアント側のレンダリングへのこのようなパフォーマンスの改善を行なってください。
- **tmpfs** の設定**:**`tmpfs` の設定を編集してパフォーマンスを向上させてください。

これらの変更の詳細については、『Sun Java System Calendar Server 6 2005Q4 Administration Guide』を参照してください。

パート IV Instant Messaging の配備

この部には、次の章があります。

第 21 章

Instant Messaging ソフトウェアの紹介

Instant Messaging では、チャット、会議室、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在席確認機能とを組み合わせることで豊かな共同作業環境を形成し、セキュリティー保護された、リアルタイムの通信や共同作業を行えます。これらの機能は、グループによる共同作業だけでなく 1 対 1 にも対応し、短期間の通信のほか、会議室やニュースチャネルなどの持続的な場の利用を可能にします。

Instant Messaging では、複数の認証メカニズムと SSL (Secure Sockets Layer) 接続を使用することで、通信の整合性が保たれます。Portal Server と Access Manager との統合によって、さらなるセキュリティー機能、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、セキュリティー保護されたリモートアクセスが可能になります。

この章には、次の節があります。

Instant Messaging サービスとは

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次のとおりです。

- 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、ルーティングする送信先システムの設定

さらに、Instant Messaging サービスはリアルタイムの会議室、ニュース、およびカレンダアラート機能を提供し、オフラインユーザーには電子メールメッセージの転送機能を提供します。

優れた Instant Messaging サービスには、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の実現が不可欠です。

Instant Messaging コア製品コンポーネント

Instant Messaging には、次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、ユーザーは会議室への参加にもこのクライアントを使用します。クライアントは Sun Java System Instant Messenger とも呼ばれます。
- **Instant Messaging Server:** 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。サーバーは、在籍情報を Instant Messenger クライアントに提供し、エンドユーザーによるセッションの確立を可能にし、ポリシーを実施します。
- **Instant Messaging** マルチプレクサ**:** メッセンジャー接続を統合するスケーラビリティーのあるコンポーネントです。たとえば、同時接続数が数千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ーを高めます。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。スケーラビリティーに加え、ファイアウォールの外にマルチプレクサをインストールし、サーバーをファイアウォール内に残すことで、承認されていない外部アクセスからサーバーを保護することができます。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、SMTP などのプロトコルについては、267 ページの「Instant Messaging でサポートされている標準」で説明します。
- **Access Manager Instant Messaging** サービス定義**:** Instant Messaging は、Access Manager SDK を使って Access Manager にサービス定義を提供することで、Access Manager 管理ポリシーと SSO 機能をサポートします。
- **Instant Messaging API:** カスタム Instant Messaging クライアントを作成可能にします。

Instant Messaging の関連コンポーネント

この節で説明するソフトウェアコンポーネントは Instant Messenger サーバーで使用されますが、インストールは個別に行われます。これらのサーバーと Instant Messenger の詳細なやり取りについては、第 23 章を参照してください。

Web サーバー

(必須) どのような配備においても、Sun Java System Web Server や Sun Java System Application Server などの Web サーバーをインストールする必要があります。また、Apache などのオープン標準に準拠した Web サーバーを使用することもできます。いずれの場合も、Instant Messenger リソースは Web サーバーホストに存在する必要があります。

Instant Messaging では、Web サーバーが Instant Messenger リソースを処理します。Instant Messenger リソースには次のものが存在します。

- Instant Messenger によって提供されている `index.html` ファイルまたは Instant Messenger 起動用リンクを含むホームページ
- Instant Messenger jar ファイル (`messenger.jar`、`imres.jar`、`imbrand.jar`、`imdesktop.jar`、`imnet.jar`、および `imjni.jar`)
- Instant Messenger オンラインヘルプ

Instant Messenger リソースは、Web サーバーと同じホストにインストールする必要があります。Access Manager の配備では、これらのリソースを Access Manager のホスト、または別の Web サーバーホストにインストールできます。多くの場合、リソースは Instant Messaging サーバーソフトウェアと同じホストにインストールされます。ただし、Instant Messenger リソースを Instant Messaging サーバーまたはマルチプレクサと別のホストに置くこともできます。

注 – Instant Messaging を設定する前に Web サーバーをインストールしてください。

LDAP サーバー

(必須) Instant Messaging は、エンドユーザーの認証と検索に Directory Server などの LDAP サーバーを使用します。Portal Server を実装する配備では、Instant Messaging は Portal Server と同じ LDAP サーバーを使用します。LDAP ディレクトリがまだインストールされていない場合は、インストールする必要があります。

Instant Messaging サーバーには Instant Messenger のエンドユーザー認証情報は格納されません。この情報は LDAP サーバーに格納されます。

デフォルトでは、エンドユーザーとグループ情報の検索に Instant Messaging サーバーは共通エンドユーザー属性 `cn` および `uid` を使用します。サーバーが別の属性を使用して検索を行うように設定することもできます。また、連絡先リストやその登録情報などの Instant Messaging のプロパティーは、Instant Messaging サーバー上のファイル、または LDAP サーバーに格納できます。

デフォルト以外の属性を使用してユーザー検索を行うようにサーバーを設定する方法については、『Sun Java System Instant Messaging 7 2005Q1 Administration Guide』を参照してください。

注 – Instant Messaging の配備を成功させるには適切な Directory Server 実装が前提となるため、このマニュアルのほかに『Sun Java System Directory Server 5 2005Q1 Deployment Plannning Guide』も参照してください。

SMTP サーバー

(省略可能) インスタントメッセージを電子メールとしてオフラインのエンドユーザーに送信するときは、Messaging Server などの SMTP メッセージングサーバーが使用されます。SMTP サーバーは、Instant Messaging サーバーと同じホスト上に存在する必要はありません。

Calendar Server

(省略可能) カレンダベースの予定をユーザーに通知するときは、Calendar Server が使用されます。

Access Manager と Access Manager SDK

(省略可能) Access Manager と Access Manager SDK は、エンドユーザーとサービスの管理に認証サービスとシングルサインオンサービスを提供します。さらに、Portal Server を含む配備では、Access Manager と Access Manager SDK が必須となります。どちらの配備でも、Instant Messaging サーバーと同じホストに SDK をインストールする必要があります。

Portal Server

(省略可能) Portal Server は、メッセージアーカイブをサポートし、Instant Messaging をセキュリティー保護されたモードで実行できるようにします。また、Portal Server デスクトップによりエンドユーザーは Instant Messenger クライアントを利用することができます。次の 2 つの Portal Server コンポーネントは追加機能を提供します。

Portal Server デスクトップ

Portal Server 環境にインストールした Instant Messenger は、Portal Server デスクトップのエンドユーザーが使用できる Instant Messaging チャネルから起動できます。

Secure Remote Access

Secure Remote Access を使えば、リモートエンドユーザーは所属する組織のネットワークとサービスに、インターネット経由で安全にアクセスできます。エンドユーザーは、ポータルゲートウェイ経由で Web ベースの Portal Server デスクトップにログインし、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で、エンドユーザーが認証されます。エンドユーザーのセッションが Portal Server との間で確立されると、エンドユーザーの Portal Server Desktop へのアクセスが有効になります。

Portal Server 環境では、Instant Messenger をセキュリティー保護されたモードにも、セキュリティー保護されていないモードにも設定できます。セキュリティー保護されたモードでは、通信内容は Portal Server の Netlet によって暗号化されます。セキュリティー保護されたモードで Instant Messenger にアクセスすると、Instant Messenger の「状態」領域に鍵のアイコンが表示されます。セキュリティー保護されていないモードでは、Instant Messenger セッションは暗号化されません。Netlet の詳細については、『Sun Java System Portal Server 6 2005Q4 Secure Remote Access 管理ガイド』を参照してください。

Instant Messaging でサポートされている標準

Instant Messaging はネイティブのインターネットテクノロジに対応しているので、顧客やパートナー企業と共同作業を行う場合でも、組織の内外をまとめて 1 つのアーキテクチャーとして維持することができます。また、特定のシステムに束縛されることもありません。Instant Messaging の主要コンポーネントは、すでに定着しているオープンなインターネット標準に基づいています。次に、代表的な標準を示します。

- **LDAP:** エンタープライズディレクトリ情報へのアクセスを提供し、正確でセキュリティー保護された Instant Messaging システムを実現します。
- **HTML:** クライアントに Web ブラウザアクセスを提供するためのフォーマット言語。

- **HTTP:** クライアントに Web ブラウザアクセスを提供するためのハイパーテキストトランスポートプロトコル。
- **SMTP:** インターネットメールメッセージ経由でインスタントメッセージを確実に配信するためのメール転送プロトコル。
- **TCP/IP:** 実績のある世界規模のネットワークプロトコル。
- **XMPP:** オープンソースゲートウェイ経由で公衆ネットワークと相互運用するための、拡張可能なメッセージングおよびプレゼンス用のプロトコル (Extensible Messaging and Presence Protocol)。

インスタントメッセージの構造フォーマット

インスタントメッセージのフォーマットとしては、XMPP プロトコルが使用されます。メッセージの本文自体は HTML 内に格納できます。

アクセスプロトコル

Instant Messaging では、ユーザーの情報と設定は LDAP ディレクトリから取得されます。このディレクトリは、Instant Messaging 専用でもかまいませんし、Access Manager や Portal Server など、ほかのコンポーネントと共用でもかまいません。ユーザーデータは通常は LDAP 検索機能によって取得されます。Access Manager と Portal Server を使用する Instant Messaging 配備では、同一の LDAP サーバーが使用されます。

通信プロトコルとメッセージ転送プロトコル

Instant Messaging のサーバー対サーバーおよびクライアント対サーバーの通信は、TCP/IP を通じて行われます。

Instant Messaging は、SMTP を使ってオフラインユーザーにメッセージを送信します。

ブラウザは、Web サーバーからの Instant Messenger リソースファイルの取得に HTTP を使用します。ブラウザは、取得したリソースファイルから HTML を読み取り、ファイルのコンテンツを表示します。

Instant Messaging 7 は、XMPP (Jabber) 対応のクライアントサーバーソリューションであり、XMPP に準拠したサーバー、クライアント、およびゲートウェイと通信を行えます。オープンソースコミュニティーでゲートウェイが利用でき、Jabber と AOL や Yahoo、およびその他の Instant Messaging システムとの通信が可能です。

Instant Messaging のソフトウェアアーキテクチャー

図 21–1 は、Instant Messaging ソフトウェアアーキテクチャーを示しています。

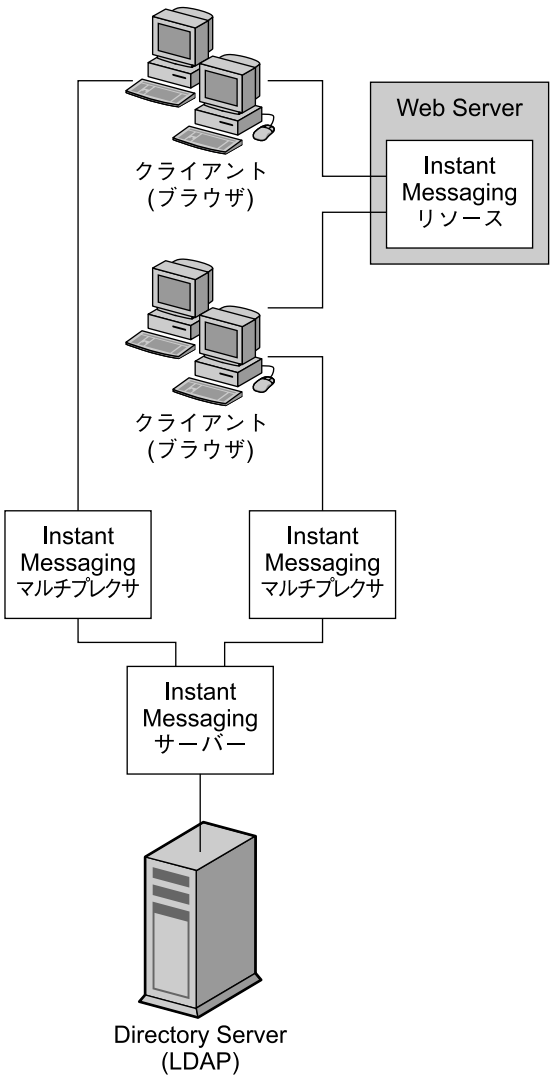

図 **21–1** Instant Messaging のソフトウェアアーキテクチャー

Web サーバー (または Web サービスが組み込まれたアプリケーションサーバー) は、ブラウザ経由でクライアントに Instant Messaging リソースをダウンロードします。クライアントはリソースファイルから構成されます。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。

ディレクトリサーバーは、設定情報、位置、メッセージのルーティング先マルチプレクサなどの、ユーザーおよびグループの配信情報を格納、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、連絡先リストやその登録情報などのユーザー情報がディレクトリサーバーに格納されることもあります。

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。

クライアントから送信されるインスタントメッセージは、直接マルチプレクサにルーティングされます。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。この処理の図については、304 ページの「Instant Messaging の物理的な配備例」を参照してください。

新規ユーザーを作成するときは、ディレクトリにユーザーエントリを追加します。ディレクトリ内のエントリを作成または変更するときは、Directory Server に付属するツールを使用します。

Instant Messaging コンポーネントの管理には、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ます。管理者が必要な権限を持っている場合は、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。

注 – Instant Messaging 配備は通常、単一マシン上にはインストールされません。また、そうした配備には、多重化や高可用性化などの追加機能も搭載されます。詳細については、第 23 章を参照してください。

次に、Instant Messaging の 3 つの主要コンポーネントについて、さらに詳しく説明します。

- 271 ページの「Instant Messaging Server」
- 271 ページの「Instant Messaging マルチプレクサ」
- 272 ページの「Instant Messenger クライアント」

Instant Messaging Server

Instant Messaging サーバーは、Instant Messenger の権限やセキュリティーの制御、アラートの送信による Instant Messenger クライアントどうしの通信の実現、チャットの開始、および使用可能なニュースチャネルへのメッセージの投稿などのタスクを処理します。また、Instant Messaging サーバーは、アーカイブ、カレンダアラート、およびオフライン電子メール通知も処理します。

Instant Messaging は、接続を 1 つのソケットに統合するマルチプレクサの接続をサポートしています。マルチプレクサについては、271 ページの「Instant Messaging マルチプレクサ」を参照してください。

エンドユーザー、ニュースチャネル、および会議室の管理には、アクセス制御ファイルと Access Manager ポリシーが使用されます。

Instant Messaging サーバーは、Instant Messaging 製品のインスタントメッセージをルーティング、転送、配信します。

LDAP 直接検索

サーバーは、LDAP サーバーの情報を直接検索できます。LDAP クエリの結果は、事前に設定可能な有効期限に達するまでプロセスにキャッシュされます。詳細については、『Sun Java System Directory Server 5 2005Q1 Administration Guide』を参照してください。

メッセージ配信

サーバーは、作成されたメッセージをメッセージ配信経路の次の配信先に送信します。送信先は、受信者のマルチプレクサまたは別のサーバーです。マルチプレクサが受信すると、メッセージは適切な受信者に直接ルーティングされます。この処理の図については、290 ページの「Instant Messaging の基本アーキテクチャー」を参照してください。

Instant Messaging マルチプレクサ

Instant Messaging マルチプレクサコンポーネントは、複数のインスタントメッセンジャー接続を 1 つの TCP (Transmission Control Protocol) 接続にまとめ、この TCP 接続を Instant Messaging サーバーに接続します。マルチプレクサは Instant Messenger からのデータを読み取り、それをサーバーに書き込みます。反対に、サーバーが Instant Messenger にデータを送信すると、マルチプレクサはそのデータを読み取り、適切な接続にそれを書き込みます。マルチプレクサは、エンドユーザーの認証やクライアントサーバー間のプロトコル (IM プロトコル) 解析は行いません。各マルチプレクサは、1 つの Instant Messaging サーバーにだけ接続されます。

Instant Messaging マルチプレクサは、必ずしもインストールする必要はありません。つまり、マルチプレクサを使用しない Instant Messaging 構成にすることも可能です。ただし、本稼働配備ではマルチプレクサを使用する構成にすることをお勧めします。

配備環境の要件に応じて、複数のマルチプレクサをインストールできます。詳細については、第 23 章を参照してください。

Instant Messenger クライアント

Instant Messenger は、Java プラグインを使用してブラウザベースのアプレットとして設定したり、Java™ Web Start を使用してスタンドアロンの Java アプリケーションとして設定したりできる Instant Messaging のクライアントです。

Solaris または Linux 上で Instant Messenger クライアントを実行するには、Java Web Start を使用する必要があります。Microsoft Windows では、アプレットまたは Java Web Start アプリケーションとして Instant Messenger を実行できます。ほとんどの場合、Java Web Start アプリケーションとして Instant Messenger を実行します。

Instant Messenger のカスタマイズについては、『Sun Java System Instant Messaging 7 2005Q1 Administration Guide』を参照してください。

Instant Messenger には、次の通信モードがあります。

- チャット:Instant Messenger バージョンの Instant Messaging 会議室がチャットと呼ばれています。チャットはリアルタイムの対話機能で、エンドユーザーはこれを利用してプロジェクトを遂行したり、顧客の質問に答えたり、即時性が要求されるその他の業務を遂行したりすることができます。チャットセッション (参加者は 2 名以上) は、必要に応じて作成されるチャット室で保持されます。
- 会議室:会議室は通常のチャットセッションと同様に機能する持続的なチャットルームで、次の機能が追加されています。
  - アクセス制御
  - モデレートチャット
- アラート:アラートにより、エンドユーザーに対する情報配信や応答が Instant Messenger インタフェースを通じて行えるようになります。アラートは、エンドユーザーに緊急情報を配信できます。アラートメッセージの送信者には、メッセージの配信時と受信者によってそのメッセージが開かれた時に通知が送信されます。また、アラートを特定の電子メールアドレスに転送するように Instant Messaging を設定することもできます。
- ポーリング:ポーリング機能により、質問に対する回答をエンドユーザーに要求できます。ポーリングの受信者には質問と選択式の回答を送信し、受信者は回答を選択してそれに返信します。
- ニュース:ニュースチャネルは、情報を投稿し、それを共有するためのフォーラムです。エンドユーザーは興味のあるニュースチャネルに登録し、ニュースチャネルの URL にアクセスして更新を確認したり、静的なメッセージを表示してニュースチャネルの更新を確認できます。管理者は、エンドユーザーに必要なニュースチャネルを割り当て、ニュースチャネルの情報を表示したり、情報を提示したりできるユーザーを決定することでニュースチャネルへのアクセスを制御します。

注 – インスタントメッセージには URL を埋め込むことができます。プロキシサーバーを使用している場合は、このような URL を解決できるように、Java Web Start を使用するクライアントでプロキシ設定の修正が必要になることがあります。

プロキシ設定の手動変更については、『Sun Java System Instant Messaging 7 2005Q1 Administration Guide』を参照してください。

Instant Messaging 配備の設計

配備プロセスは、次の基本フェーズから構成されており、ソリューションライフサイクルと呼ばれます。

- ビジネス要件の分析
- 技術要件の分析
- 論理アーキテクチャーの設計
- 配備アーキテクチャーの設計
- 配備の実行
- 配備の運用

配備フェーズは固定的なものではなく、配備プロセスは反復して行われます。

Instant Messaging やその他の Java Enterprise System コンポーネントの配備プロセスの詳細については、『Sun Java Enterprise System 2005Q4 Deployment Planning Guide』を参照してください。

第 22 章

Instant Messaging サイズ決定戦略の計画

この章では、Instant Messaging 配備のサイズ決定に関する概念、背景、および理論的根拠を紹介します。

この章には、次の節があります。

Instant Messaging サイズ決定戦略の概要

配備を計画する場合には、Instant Messaging サーバーの設定方法を検討して、パフォーマンス、スケーラビリティー、および信頼性を最適化する必要があります。

サイズ決定はそうした設計作業の重要な要素の 1 つです。サイズ決定のプロセスを実行することで、Instant Messaging サーバーユーザーへの作業負荷の見積もりを踏まえた、希望するレベルのサービスまたは応答時間を実現するために必要となるハードウェアとソフトウェアを確認できます。サイズ決定は反復的な作業です。

配備にはそれぞれに固有の特徴があるため、この章では特定のサイトに関する Instant Messaging サイズ決定情報の詳細な説明はしていません。また、この章では、LDAP や SMTP など、Instant Messaging と連携して動作するサーバーに対するサイズ決定情報も提供しません。代わりにここでは、サイズ決定計画を構築する場合には何を考慮しなければならないのかを説明します。また、Instant Messaging コンポーネントに

関する一般的なガイドラインも提示します。このガイドラインは、ユーザーのサイトのニーズに合わせて変更してかまいません。配備のハードウェアとソフトウェアのニーズを決定する場合には、ご購入先のテクニカルサポート担当者と共に作業を行なってください。

Instant Messaging サイズ決定データの収集

この節の説明を読んで、Instant Messaging のサイズ決定に必要なデータを確認してください。この節には、次の項目があります。

一意 Instant Messaging ログインのピークボリュームの決定

ピークボリュームは、1 日の特定の時間帯で Instant Messaging システムにトランザクションが最も集中したときの一意ログイン数です。このボリュームは、サイト間やユーザークラスの違いにより大きく異なります。たとえば、グループ間のピークボリュームは業務時間内のコアタイムに発生することが考えられますが、コアタイムはタイムゾーンによって異なります。

ピークボリュームの分析には、次の 3 つの基本処理が含まれます。

1. ピークがいつ発生し、どのくらい継続するかを判断します
2. ピークボリューム負荷を前提として配備のサイズを決定します

   パターンの分析が終了すれば、システムの負荷を処理しやすくし、ユーザーの求めるサービスを提供するための選択を行えます。
3. ユーザーが決定したピークボリュームを Instant Messaging 配備がサポートできることを確認します

Instant Messaging の使用率プロファイルの作成

正確なサイズ決定には、負荷の測定が不可欠です。「使用率」プロファイルは、プログラムとプロセスが Instant Messaging サーバーおよびマルチプレクサに及ぼす負荷要因を特定します。

この節では、使用率プロファイルを作成して、配備で発生する負荷の量を測定する方法について説明します。

使用率プロファイルを作成するには、次の質問に答えてください。

1. システムの合計ユーザー数は何人ですか。

   システムのユーザー数を数えるときは、アカウントを持ち、システムにログインできるユーザーだけを対象とするだけでなく、アカウントを持ち、現在システムにログインしていないユーザーも対象に含めます。次の表は、全ユーザーの種類別の内訳を示しています。

接続	説明
アクティブでないユーザー	Instant Messaging のアカウントを持ち、システムに現在ログインしていないユーザー。接続していないユーザーは、ディスク領域を消費しますが、CPU またはメモリーは消費しません。
接続非アクティブユーザー	ログインはしているが、インスタントメッセージを現在送受信していないユーザー。
接続アクティブユーザー	システムにログインし、メッセージの送信、連絡先リストなどのユーザー情報の更新、会議室への参加などの処理を一日中、活発に行なっているユーザー。

   次の 3 つの一般的なプロファイルで、ユーザーを分類します。これらのユーザーの合計から、サポートが必要な同時接続の総数を想定することができます。

   小規模な配備であれば、デフォルトの設定でもサイトのニーズを満たすことができます。このため、配備の規模がごく小規模 (たとえば 300 ユーザー未満) であれば、サイズ設定については考慮する必要がない場合もあります。クライアントサービス担当者と作業を行い、個別のニーズについて判断します。

2. システムのピークボリューム時の接続はいくつですか。

   システムで維持する必要がある同時接続ユーザーの最大数を正確に算定しておくことが、リソースの条件を計画する上で重要です。配備では設定済みユーザーの最大数を想定しますが、計画では、アクティブかどうかにかかわらず接続されている同時接続ユーザーの最大数を想定するほうが重要です。同時接続ユーザー数は、安全な見積もりとして、1 対 10 で算出できます。つまり、50,000 ユーザーが設定されている配備では、同時接続ユーザーは 5,000 人と算定します。

   具体的には、同時平行接続、アイドル接続、ビジー接続の数に注意してください。

接続	説明
並行接続	ある時点にシステムで確立されている一意の TCP 接続またはセッションの数。
アイドル接続	クライアントとマルチプレクサ、またはサーバーとマルチプレクサの間で情報が送信されていない接続。
ビジー接続	進行中の接続。クライアントとマルチプレクサ、またはマルチプレクサとサーバーの間で確立され、情報の送信が行われている接続。

配備の「同時接続数」を決定するには、Solaris プラットフォームの `netstat` コマンドを使って確立済み TCP 接続の数をカウントします。

サポートできる同時接続数を決定するには、マルチプレクサのパフォーマンス調整に使用される `iim.conf` ファイルから 2 つのパラメータの値を取得する必要があります。

a. `iim_mux.numinstances` - マルチプレクサインスタンスの数を指定する
b. `iim_mux.maxsessions` - 1 つのマルチプレクサプロセスが処理できる最大クライアント数を指定する。デフォルトは 1000。

これらの値を取得したら、`numinstances` の値と `maxsessions` の値を掛け合わせます。これにより、配備でサポートされる同時接続の総数が算出されます。`iim.conf` ファイルについては、『Sun Java System Instant Messaging 7 2005Q1 Administration Guide』を参照してください。

3. 大規模な配備を行う場合には、ユーザーをどのように組織化しますか。

   たとえば、アクティブユーザーと非アクティブユーザーをそれぞれ異なるサーバーに配置することを検討します。

4. 1 ユーザーあたりのストレージ容量

   連絡先リストなどのエンドユーザーデータを LDAP に格納しない場合、このデータの格納に必要な容量を計画する必要があります。このデータを LDAP 外に格納するようにサーバーを設定した場合、サーバーはこれをフラットファイルに格納します。詳細については、『Sun Java System Instant Messaging 7 2005Q1 Administration Guide』を参照してください。

5. インターネットからどれぐらいの数のメッセージが Instant Messaging システムに送信されますか。

   メッセージの数は、ピークボリューム時の 1 秒あたりのメッセージ数で測定します。

6. ユーザー別ではどれぐらいの数のメッセージが送信されますか。

   - システム上のエンドユーザー
   - インターネットに対して送信される数

   このメッセージの数も、ピークボリューム時の 1 秒あたりのメッセージ数で測定します。

7. SSL を使用しますか。使用する場合は、ユーザーの何パーセントが、またどのようなタイプのユーザーが使用しますか。

   たとえばある組織では、ピーク時の 20% の接続が SSL で保護されます。

これらの質問に答えることで、配備のための、準備段階としての使用率プロファイルが完成します。Instant Messaging のニーズの変更に応じて、この使用率プロファイルにも修正を加えます。

その他の質問

次の質問は使用率プロファイルの作成に使用できるものではありませんが、配備のサイズ決定戦略には重要なものです。これらの質問にどのように答えるかによって、ハードウェアの追加を検討しなければならなくなる場合もあります。

1. 配備にどの程度の冗長性を持たせますか。

   たとえば、高可用性が重要であると考えられますか。

2. バックアップと復元 (障害回復やサイトフェイルオーバーなど) はどのように計画されていますか。回復タスクが完了するまでにどのくらいの時間を予想しますか。

   通常は、サーバー設定ファイル、データベース、カスタマイズされたリソースファイルのバックアップが必要です。

Instant Messaging のユーザーベースまたはサイトプロファイルの定義

ユーザープロファイルの作成が完了したら、次にそれをこの節で説明されているユーザーベースの例と比較してみます。ユーザーベースは、ユーザーが実行する Instant Messaging オペレーションの種類から構成されます。Instant Messaging ユーザーは、次のいずれかのユーザーベースに分類されます。

- 279 ページの「一般ユーザー」
- 280 ページの「ヘビーユーザー」

この節のユーザーベースの例では、ユーザーの行動を幅広く一般化しています。実際の使用率プロファイルがこのユーザーベースと一致するとは限りません。負荷シミュレータ (280 ページの「Instant Messaging 負荷シミュレータの使用」を参照) の実行時に、差異を調節してください。

一般ユーザー

一般に、軽量なユーザーベースは、シンプルな Instant Messaging 要件を持つユーザーから構成されます。これらのユーザーがチャットセッションを開始したり、出席依頼を受け取ったりすることはほとんどありません。Instant Messaging を在席確認ツールとしてだけ使用する場合もあります。

ヘビーユーザー

ヘビーユーザーは、一般ユーザーとは比較にならないほど多くのシステムリソースを消費します。これらのユーザーの一般的なリソース使用状況は、たとえば次のようなものです。

- 1 日のうち 20 回以上、在席の更新がある。
- 連絡リストに約 30 の連絡先がある。
- 連絡リスト内のすべての連絡先の在席更新の通知を受け取っている。
- 1 日あたり 4 つの会議室またはチャットを設定し、各会議室の平均参加者は 3 名、持続時間は 10 分、1 ～ 15秒ごとにメッセージが会議室に追加される。

Instant Messaging 負荷シミュレータの使用

Instant Messaging アーキテクチャーのパフォーマンスを測定するには、負荷シミュレータの入力としてユーザーベース (279 ページの「Instant Messaging のユーザーベースまたはサイトプロファイルの定義」を参照) と使用率プロファイル (276 ページの「Instant Messaging の使用率プロファイルの作成」を参照) を使用します。

負荷シミュレータは、ピークボリューム環境を作り出し、サーバーにかかる負荷の量を調整します。これにより、システムに過負荷をかけることなく希望する応答時間を実現するには、ハードウェア、スループット、または配備のアーキテクチャーを変更する必要があるかどうかを判断できます。負荷シミュレータを使用するには、次の 5 つの基本手順に従います。

1. テストするユーザーベース (たとえば、一般ユーザー) を定義します

   必要に応じて、使用率プロファイルに最適化するように個別のパラメータを調整します。
2. テストするハードウェアを定義します
3. 負荷シミュレータを実行し、ユーザーベースを使用してテストされたハードウェアの最大同時接続数を測定します
4. 結果を記録して、稼働中の配備の結果と比較します
5. ピーク負荷状態の応答時間が組織で容認されるレベルになるまで、さまざまなユーザーベースとハードウェアを使用してこのプロセスを繰り返します

注 – 推奨負荷シミュレータとサポートについては、ご購入先のクライアントサービス担当者に連絡してください。

Instant Messaging のシステムパフォーマンスガイドラインについて

負荷シミュレータを使用してハードウェアとユーザーベースの評価を行うと、システムパフォーマンスを測定する必要があります。この節の各トピックでは、システムの全体的なパフォーマンスを向上させる方法について説明します。

Instant Messaging のメモリー使用率

配備で使用するそれぞれのマシンに、適切な量の物理メモリーが搭載されていることを確認してください。物理メモリーを追加するとパフォーマンスが向上し、ピークボリューム時でも Instant Messaging サーバーが適切に動作するようになります。メモリー容量が十分であれば、Instant Messaging は過度のスワッピングをすることなく効率的に動作できます。

ほとんどの配備では、256M バイト以上の RAM が必要です。RAM の必要容量は、同時並行クライアント接続の数、およびサーバーとマルチプレクサが同じホストに配備されているかどうかによって異なります。同時接続については、276 ページの「Instant Messaging の使用率プロファイルの作成」を参照してください。サーバーとマルチプレクサの同一ホスト上でのホスティングについては、284 ページの「Instant Messaging アーキテクチャー戦略の構築」を参照してください。

Solaris システムでは、`iim.conf` ファイルの `iim.jvm.maxmemorysize` パラメータを変更して、サーバーに割り当てるメモリーの容量を設定できます。このパラメータは、サーバーを実行する JVM (Java Virtual Machine) が使用できる最大メモリー数を M バイト単位で指定します。デフォルトの設定は 256M バイト、最大設定は 500M バイトです。このパラメータの設定方法については、『Sun Java System Instant Messaging 7 2005Q1 Administration Guide』を参照してください。

Windows NT システムでは、現時点ではこの値を変更できません。

Instant Messaging のディスクスループット

ディスクのスループットとは、システムでメモリーからディスクに、またはディスクからメモリーに転送されるデータ量のことです。このデータ転送レートは、Instant Messaging のパフォーマンスに重大な影響を及ぼします。システムのディスクスループット効率を向上させる方法は、次のとおりです。

- 保守作業を検討し、バックアップのための十分な帯域幅があることを確認します。バックアップも、特にリモートバックアップがネットワーク帯域幅に影響します。プライベートバックアップネットワークの利用が一層効率的です。
- スループット効率が向上するようにデータストアを慎重にパーティションで区切ります。

- 大規模な配備では、ユーザーベースが必ず RAID (Redundant Array of Independent Disks) 環境全体に分散されるようにします。通常、この決定は、ディレクトリサーバーの配備計画プロセスの一部として行います。
- ディスクからデータを取得する操作のスピードを向上させるために、データを複数のディスクでストライピングします。

Instant Messaging のディスク容量

サーバーシステムのディスク容量を計画するときは、オペレーティング環境ソフトウェア、Instant Messaging ソフトウェア、Instant Messaging をサポートするためにインストールが必要で、現在ネットワーク内に存在しないサーバー (LDAP など) の容量を考慮してください。必ず外部ディスク配列を使用してください。さらに、ユーザーディスク容量を割り当てます。この容量は、通常、サイトのポリシーに従って決定されます。一般的なインストールでは、次の容量が必要です。

- サーバーまたはマルチプレクサごとに約 300M バイトのディスク空き容量
- 1 ユーザーごとに約 5K バイトのディスク容量
- Instant Messaging アーカイブ用の追加容量

  このアーカイブでは、インスタントメッセージが取り込まれ、Portal Server 検索データベース内にアーカイブされます。エンドユーザーは、アーカイブされたメッセージを Portal Server デスクトップの検索ページから検索し、取得することができます。

表 22–1 は、アーカイブ機能を有効または無効にした場合のサーバーおよびマルチプレクサのディスク容量のサイズ設定を示しています。この表に示す値は、400MHz の Ultra SPARC II Processor を使用して算出したものです。

表 **22–1** 同時接続ユーザーを考慮した、Instant Messaging サーバーとマルチプレクサのメモリーディスク容量のサイズ設定

	接続/非アクティブユーザーのサーバーメモリー消費量	接続/アクティブユーザーのサーバーメモリー消費量	接続/非アクティブユーザーのマルチプレクサメモリー消費量	接続/アクティブユーザーのマルチプレクサメモリー消費量
アーカイブ無効	ユーザーあたり 8M バイト + 20K バイト	ユーザーあたり 120M バイト +20K バイト	ユーザーあたり 8M バイト +20K バイト	ユーザーあたり 8M バイト + 28K バイト
SSO/ ポータル / アーカイブ有効	ユーザーあたり 100M バイト + 25K バイト	ユーザーあたり 120M バイト + 30K バイト	ユーザーあたり 8M バイト + 35K バイト	ユーザーあたり 8M バイト + 40K バイト

Instant Messaging のネットワークスループット

ネットワークスループットは、一定時間内にクライアントアプリケーションとサーバー間のネットワークで転送可能なデータ量のことです。ネットワークに接続されたサーバーがクライアントからの要求に応答できない場合、通常クライアントは要求の再送信を何度も行います。再送信のたびに、システムにはオーバーヘッドと余分なネットワークトラフィックが生じます。

データの完全性とシステムのパフォーマンスを向上させ、ネットワークの混雑を解消することで、再送信の数を減らすことができます。それには、次の手順に従います。

- ボトルネックを解消し、ネットワークインフラストラクチャーが負荷を処理できるようにします
- ネットワークを分割します
- ネットワーク構築時には理論上の最大値を使用しないようにします。それにより、将来の拡張にも対応できるだけの容量を確保できます
- トラフィックのフローを異なるネットワークパーティションに分割して衝突を減らし、帯域幅の使用を最適化します

Instant Messaging の CPU リソース

サーバーとマルチプレクスサービス用に十分な数の CPU を用意します。さらに、使用を計画している RAID システムにも十分な CPU を用意します。配備でアーカイブ機能を利用する場合は、ディスク容量の要件についても考慮する必要があります。

表 22–2 は、アーカイブが有効または無効な場合のインストールの最適なパフォーマンスに必要な CPU 数を示しています。この表に示す値は、400MHz の Ultra SPARC II Processor を使用して算出したものです。

表 **22–2** Instant Messaging の CPU の使用に関する数値

	接続/非アクティブユーザーのサーバー **CPU** 使用率	接続/アクティブユーザーのサーバー **CPU** 使用率	接続/非アクティブユーザーのマルチプレクサ **CPU** 使用率	接続/アクティブユーザーのマルチプレクサ **CPU** 使用率
アーカイブ無効	1 CPU あたり数十万のユーザー	1 CPU あたり 30,000 ユーザー	1 CPU あたり 50,000 ユーザー	1 CPU あたり 5,000 ユーザー

Instant Messaging マルチプレクサの最適設定

マルチプレクサの配備を計画するときは、次に提案する一般的な値を参考にしてください。ここで説明するパラメータは、`iim.conf` ファイルで設定できます。

- `iim_mux.maxthreads` の値は、サーバー上の CPU の数を超えないようにする必要があります。

  これにより、リソースの使用率を最大にし、処理速度を最適化することができます。
- `iim_mux.maxsessions` の値は、接続拒否を防ぐため十分な大きさに設定する必要がありますが、マルチプレクサプロセスに負荷がかかりすぎない適切な値にする必要があります。
- 同時接続するクライアントの予想数が、安全基準による最大可能数よりも小さくなるようにします。
- スレッドまたは同時セッションの数を必要以上に大きく設定しないようにします。必要以上のサイズに設定すると、システムリソースを不必要に消費することになります。

- `iim_mux.numinstances` は、最初はシステムの CPU 数に設定することをお勧めします。

これらのパラメータの詳細については、『Sun Java System Instant Messaging 7 2005Q1 Administration Guide』を参照してください。

Instant Messaging アーキテクチャー戦略の構築

システムパフォーマンスのニーズを確認した後、Instant Messaging 配備のサイズ決定では次に、アーキテクチャーの決定に基づいて特定のコンポーネントのサイズを決定します。

この節の各トピックでは、2 層と 1 層のアーキテクチャーを配備する場合のサイズ決定で考慮しなければならないことについて説明します。Instant Messaging と共にロードバランサを使用する方法についても説明します。

2 層 Instant Messaging アーキテクチャー

2 層アーキテクチャーでは、Instant Messaging サーバー配備をアクセス層とデータ層の 2 層に分割します。単純な 2 層配備では、アクセス層に 1 つまたは複数のマルチプレクサとサーバーを追加します。ユーザーにとってマルチプレクサはプロキシのように機能し、メッセージを Instant Messaging サーバーにリレーします。データ層には、Instant Messaging サーバーデータベースとディレクトリサーバーが保持されます。図 22–1 は、簡略化した 2 層 Instant Messaging アーキテクチャーを示しています。

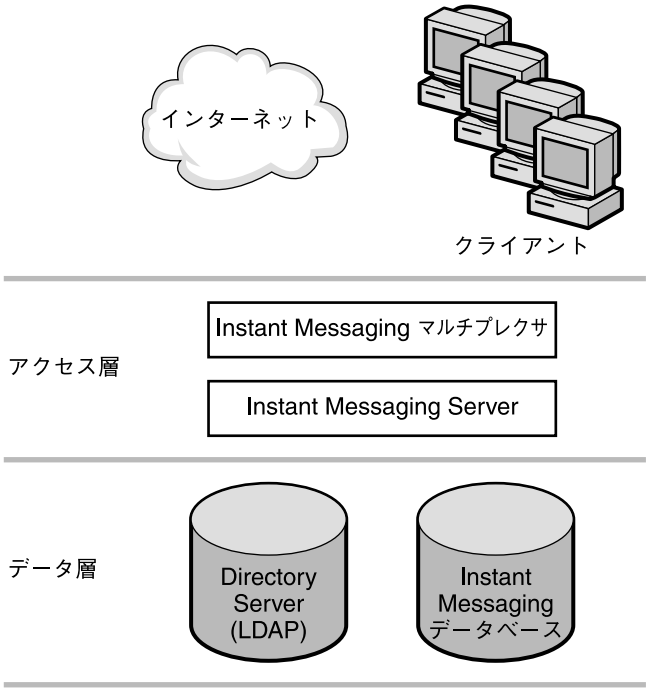

図 **22–1** 簡略化した 2 層 Instant Messaging アーキテクチャー

1 層アーキテクチャーと比較して、2 層アーキテクチャーにはサイズ設定上の利点があります。2 層アーキテクチャーの特徴は、次のとおりです。

- 1 層アーキテクチャーよりも管理が容易です
- SSL やメッセージの再処理など、負荷の高いプロセスをオフロードできます
- サイズの拡張が容易で、短いダウン時間でシステムをアップグレードできます

多重化サービスのサイズ決定

マルチプレクサのサイズを設定する場合、システムの負荷、特にマルチプレクサが処理する同時接続の数に基づいて計算を行います。

さらに、次のことを実行する必要があります。

1. 必要であれば、SSL 用の CPU またはハードウェアアクセラレータを追加します。
2. マルチプレクサを設定するマシンにメモリーを追加します。
3. サービス拒否について考慮します。
4. 必要に応じて、負荷分散と冗長性の能力を追加します。

配備に冗長性を持たせる場合、それぞれのマシンがスループットや応答時間を大きく損なわずにピーク負荷を処理できるようにする必要があります。

1 層 Instant Messaging アーキテクチャー

1 層アーキテクチャーは、アクセス層とデータ層に分割されません。Instant Messaging サーバー、マルチプレクサ、場合によってはディレクトリサーバーが 1 つの層にインストールされます。次の図は、この概念を示したものです。

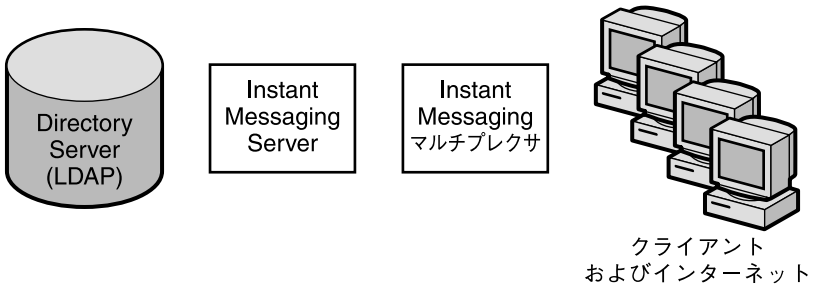

図 **22–2** 簡略化した 1 層 Instant Messaging アーキテクチャー

2 層アーキテクチャーと比較して、1 層アーキテクチャーはハードウェアへの初期投資が少なくて済みます。しかし、1 層アーキテクチャーを選択した場合は、保守のためにかなりの停止時間を見込んでおく必要があります。

1 層アーキテクチャーのサイズを設定するには、次の事項を考慮する必要があります。

1. 必要に応じて SSL 用の CPU を追加します。
2. サービス拒否攻撃について考慮します。
3. クライアント接続数の増加に対応するためのディスクを追加します。
4. 各マルチプレクサ用にディスクを追加します。

1 層アーキテクチャーおよび 2 層アーキテクチャーにおける Instant Messaging コンポーネントのサイズ決定に関する特別な手順については、ご購入先のクライアントサービス担当者に連絡してください。

Instant Messaging と共にロードバランサを使用する

Instant Messaging は、Instant Messaging マルチプレクサの前に配置されているロードバランサの使用をサポートしています。ただし、現在のところ、Instant Messaging マルチプレクサと Instant Messaging サーバーの間ではロードバランサを使用できません。

Instant Messaging を Portal Server/Secure Remote Access 配備の一部として配備する場合、Secure Remote Access ゲートウェイと Instant Messaging マルチプレクサの間にロードバランサを配置できます。

注 – クライアント接続にセキュリティーが必要で、HTTP トンネリングには不要である場合、Secure Remote Access の代わりに SSL の使用を検討します。SSL をマルチプレクサで使用可能にして、ファイアウォールの外側に配置することにより、セキュリティー保護された Instant Messaging クライアント接続を構成できます。

Instant Messaging リソース要件の例

ここでは、次の 2 種類の Instant Messaging 配備に必要なリソース分散の例と推奨サイズに関する情報を紹介します。

- 287 ページの「小規模配備のリソース要件の具体例」
- 287 ページの「大規模配備のリソース要件の具体例」

小規模配備のリソース要件の具体例

サーバーとマルチプレクサが単一サーバーにインストールされ、次のプロファイルの 10,000 ユーザーを持つ小規模の Instant Messaging 配備の場合

- 30 % の接続アクティブユーザー
- 20 % の接続非アクティブユーザー
- 50 % の非接続ユーザー

メモリー要件として、1 つまたは 2 つの CPU で、それぞれについて 300 ～ 500M バイトの RAM が必要です。

大規模配備のリソース要件の具体例

次のプロファイルの 1,000,000 ユーザーを持つ大規模の Instant Messaging 配備の場合

- 5 % の接続アクティブユーザー
- 20 % の接続非アクティブユーザー
- 75 % の非接続ユーザー

サーバーのメモリー要件として、2 つの CPU で、合計 4G バイトの RAM が必要です。マルチプレクサには、16 個の CPU で、合計 4G バイトの RAM が必要です。

第 23 章

Instant Messaging アーキテクチャーの開発

この章では、さまざまな Instant Messaging アーキテクチャーについて説明します。インストールする必要のあるコンポーネントは、個々の配備ごとに異なります。たとえば、電子メール通知をサポートする場合は、SMTP サーバーをインストールする必要があります。電子メール通知をサポートしない場合は、SMTP サーバーをインストールしないでください。

Instant Messaging と相互運用するコンポーネントの詳細については、265 ページの「Instant Messaging の関連コンポーネント」を参照してください。

この章には、次の節があります。

注 – 現在、Solaris プラットフォームではすべての配備オプションが利用できます。Linux および Windows オペレーティングシステムでは一部の配備オプションが利用できません。

Instant Messaging の基本アーキテクチャー

図 23–1 は、Instant Messaging の基本アーキテクチャーを示したものです。

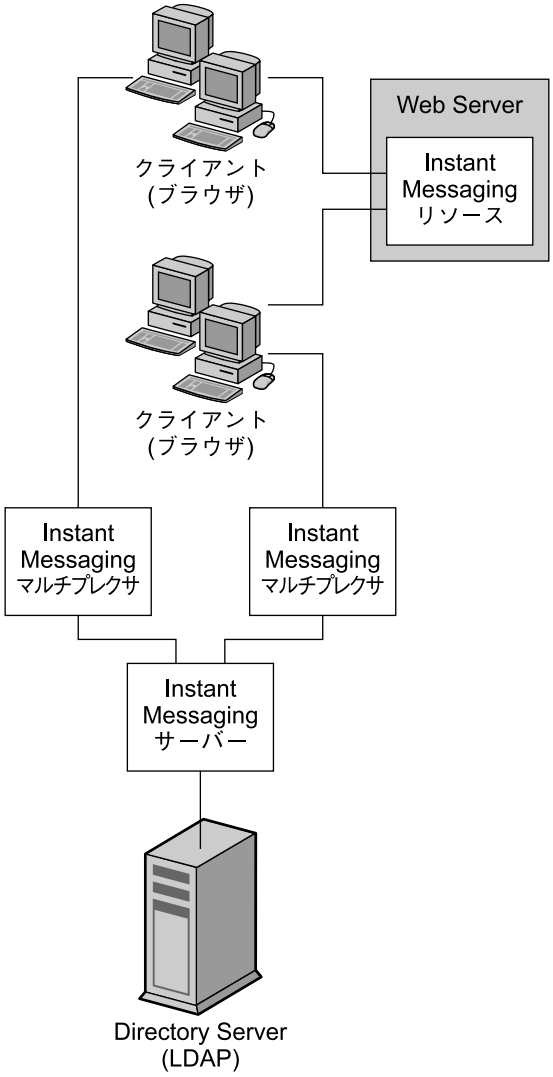

図 **23–1** Instant Messaging の基本アーキテクチャー

この Instant Messaging 基本アーキテクチャーでは、チャット、ニュースアラート、会議室などの機能が提供されます。この基本機能を利用するには、次のコンポーネントをインストールする必要があります。

- Instant Messaging サーバーと 1 つ以上の Instant Messaging マルチプレクサ
- Instant Messaging リソース
- Sun Java System Web Server などの Web サーバー
- Sun Java System Directory Server などの LDAP サーバー

この例では、次のようにします。

- 認証と検索用のユーザーエントリは LDAP サーバーに保持されます。
- クライアントは Web サーバーまたは Sun Java System Application Server から Instant Messaging リソースをダウンロードします。
- クライアントは常に Instant Messaging マルチプレクサ経由で Instant Messaging サーバーに接続します。

基本アーキテクチャーにおける認証

図 23–2 は、Instant Messaging の基本アーキテクチャーで行われる認証プロセスで、ソフトウェアコンポーネントがどのように連携するかを示しています。認証要求のフローに注目しています。このプロセスの各段階の説明は、図の後に記載しています。

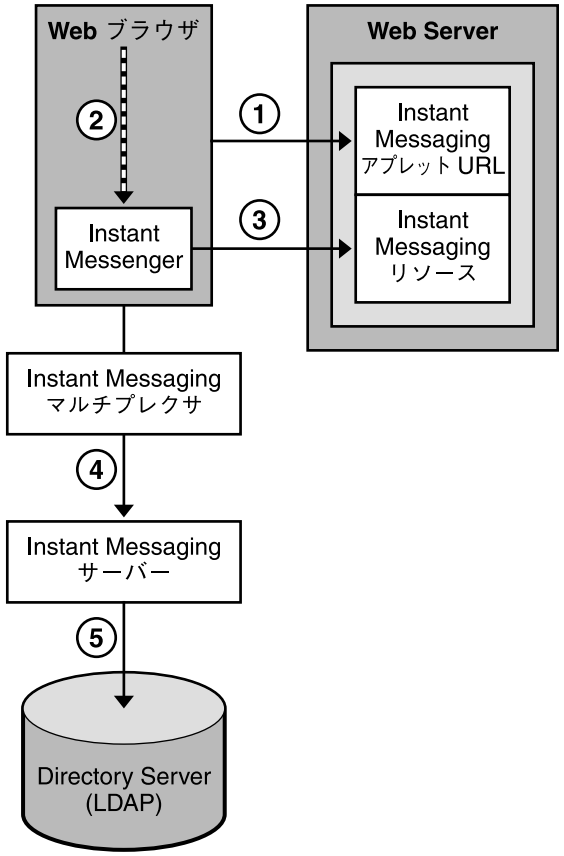

図 **23–2** Instant Messaging の基本アーキテクチャーにおける認証要求のフロー

基本アーキテクチャーにおける認証プロセスは、次のように処理されます。

1. エンドユーザーはブラウザから Instant Messenger アプレット URL にアクセスし、クライアントを呼び出すメソッドを選択します。
2. ブラウザが Java Web Start または Java プラグインを起動します。
3. Java Web Start または Java プラグインは、適切な Instant Messenger リソースファイルをダウンロードし、Instant Messenger を起動します。
4. ログインウィンドウが表示され、エンドユーザーはログイン名とパスワードを入力します。ログインデータはマルチプレクサ経由で Instant Messaging サーバーに送信されます。
5. Instant Messaging サーバーは LDAP サーバーと通信してエンドユーザーを認証し、連絡先リストやその登録情報などのエンドユーザー情報を要求します。

エンドユーザーの認証が完了すると、Instant Messaging のメインウィンドウが表示され、そのエンドユーザーの連絡先リストが表示されます。これにより、エンドユーザーは他のエンドユーザーとの Instant Messaging セッションに参加できるようになります。

Instant Messaging 電子メール通知 (カレンダアラート) アーキテクチャー

オフラインユーザーへの電子メール通知とユーザーへのカレンダの予定の Instant Messaging ベース通知をサポートするように、Instant Messaging を配備することができます。

電子メール通知とカレンダアラートをサポートする Instant Messaging アーキテクチャー は、290 ページの「Instant Messaging の基本アーキテクチャー」と同じ機能を提供します。この機能を提供するには、290 ページの「Instant Messaging の基本アーキテクチャー」に記載されたコンポーネントを含める必要があります。電子メールアラートをサポートするには、Sun Java System Messaging Server などの SMTP サーバーもインストールする必要があります。カレンダアラートをサポートするには、Sun Java System Calendar Server もインストールする必要があります。

電子メール通知を有効にする場合、Instant Messaging のインストール時に使用する SMTP サーバーの指定が必要となります。SMTP サーバーがインストールされていない場合は、Instant Messaging ソフトウェアのインストール前にインストールを完了する必要があります。図 23–3 は、ネットワーク経由の電子メール通知に対応した Instant Messaging を示しています。

Calendar Server がインストールされていない場合は、Instant Messaging ソフトウェアのインストール前にインストールを完了する必要があります。図 23–4 は、ネットワーク経由のカレンダ通知に対応した Instant Messaging を示しています。

このアーキテクチャーの認証フローは基本配備と同じです。詳細については、291 ページの「基本アーキテクチャーにおける認証」を参照してください。

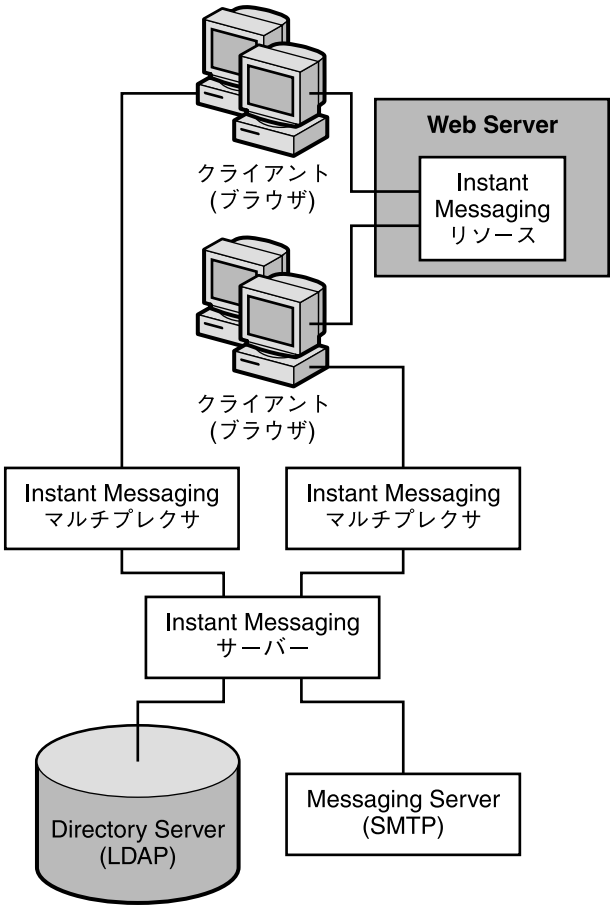

図 **23–3** 電子メール通知を使用する Instant Messaging アーキテクチャー

この例では、次のようにします。

- 認証と検索用のユーザーエントリは LDAP サーバーに保持されます。
- Instant Messaging サーバーは、オフラインユーザー宛てのメッセージを SMTP サーバーへ転送します。SMTP サーバーは、そのメッセージを電子メールとしてユーザーのメールボックスに送信します。
- クライアントは Web サーバー (またはアプリケーションサーバー) から Instant Messaging リソースをダウンロードします。
- クライアントは常に Instant Messaging マルチプレクサ経由で Instant Messaging サーバーに接続します。

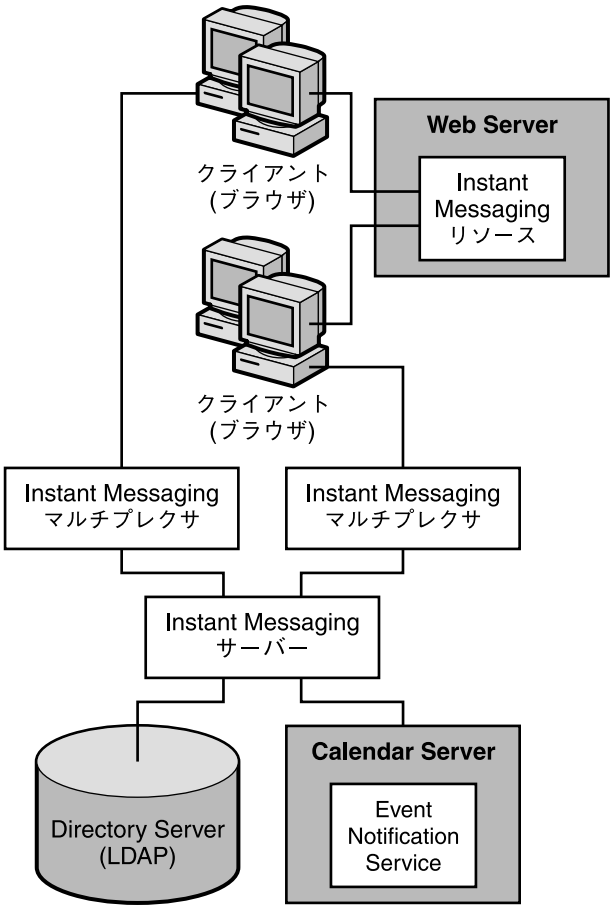

図 **23–4** カレンダアラートを使用する Instant Messaging アーキテクチャー

この例では、次のように動作します。

- 認証と検索用のユーザーエントリは LDAP サーバーに保持されます。
- ENS (Event Notification Server) がカレンダの予定の通知を Instant Messaging サーバーに送信し、そこから適切なエンドユーザーに通知が転送されます。
- クライアントは Web サーバー (またはアプリケーションサーバー) から Instant Messaging リソースをダウンロードします。
- クライアントは常に Instant Messaging マルチプレクサ経由で Instant Messaging サーバーに接続します。

Access Manager または SSO を使用する Instant Messaging アーキテクチャー

Access Manager のポリシー機能とシングルサインオン (SSO) を使用するように Instant Messaging を配備することができます。Access Manager を使用する Instant Messaging アーキテクチャーは、290 ページの「Instant Messaging の基本アーキテクチャー」と同じ機能を提供します。この機能を利用するには、290 ページの「Instant Messaging の基本アーキテクチャー」に記載されたコンポーネントのほかに、Access Manager もインストールする必要があります。さらに、Instant Messaging サーバーホスト上に Access Manager SDK をインストールする必要があります。

このアーキテクチャーの場合、Instant Messaging はユーザーの検索にディレクトリを使用しますが、ユーザーの認証または承認には使用しません。ユーザーの認証と承認は Access Manager 側で行われます。

Access Manager で SSO を使用する場合、Access Manager と Instant Messaging が同じ Web コンテナを使用するように構成する必要があります。

図 23–5 は、Access Manager を使用する Instant Messaging アーキテクチャーを示しています。

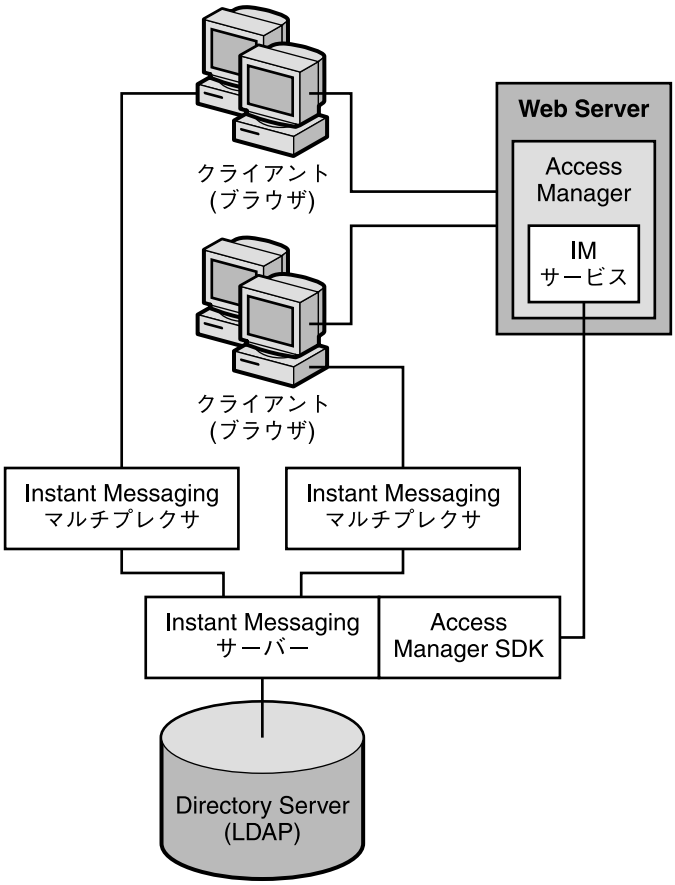

図 **23–5** Access Manager ベースのサーバーポリシー管理またはシングルサインオンを使用する Instant Messaging アーキテクチャー

この例では、次のように動作します。

- ユーザーエントリは LDAP サーバーに保持されます。
- Web サーバー (または Web サーバーが組み込まれたアプリケーションサーバー) は、ブラウザ経由でクライアントに Instant Messaging リソースをダウンロードします。リソースは、基本的にはクライアントです。
- クライアントは常にマルチプレクサ経由で接続します。
- Access Manager が提供する Instant Messaging 関連サービスには、在席確認サービスとインスタントメッセージングサービスがあります。
- Instant Messaging 配備で ID ベースのサービスを管理する Access Manager 管理インタフェースには、Web サーバーを使用してアクセスすることができます。Access Manager の Web サーバーは、Instant Messaging リソースのサーバーと同じであってもかまいません。詳細については、Access Manager のマニュアルを参照してください。

- Access Manager SDK は、Instant Messaging サーバーが Access Manager との通信時に使用する API を提供します。

Access Manager のみを使用するアーキテクチャーにおける認証

図 23–6 は、シングルサインオン環境において、コンポーネント Portal Server および Access Manager と連携する Instant Messaging ソフトウェアによって使用される認証プロセスを示したものです。図 23–2 と同様に、この図も認証要求のフローを示しています。このプロセスの各段階の説明は、図の後に記載しています。

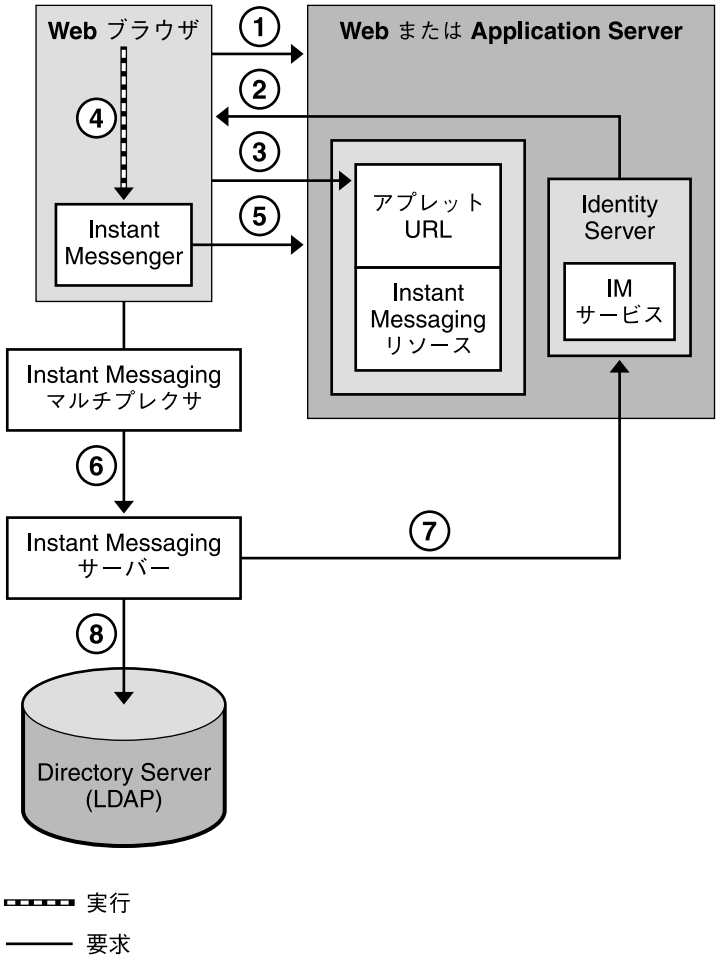

図 **23–6** Access Manager を伴う構成での認証要求のフロー

シングルサインオン環境において、この配備の Instant Messaging サーバーの認証プロセスは、次のように機能します。

1. ユーザーは、Web ブラウザに適切な URL を入力し、Access Manager にログインします。
2. Access Manager ソフトウェアはエンドユーザーを認証し、セッショントークンを返します。

   シングルサインオンが機能するには、セッショントークンが必要です。このトークンはアプレットパラメータとして提供され、認証プロセス全体で使用されます。セッショントークンがある限り、資格の再入力はエンドユーザーに求められません。
3. エンドユーザーはブラウザから Instant Messenger アプレットにアクセスし、クライアントを呼び出すメソッドを選択します。
4. ブラウザが Java Web Start または Java プラグインを起動します。
5. Java Web Start または Java プラグインは、適切な Instant Messenger リソースファイルをダウンロードし、Instant Messenger を起動します。
6. Instant Messenger は、セッショントークンを使用して Instant Messaging サーバーへの認証を要求します。
7. Instant Messaging サーバーは、セッショントークンの検証を Access Manager に求めます。セッションが有効であれば、Instant Messenger はエンドユーザーの連絡先リストを表示し、エンドユーザーはチャット、アラート、ポーリングなどの Instant Messenger サービスを利用できるようになります。
8. Instant Messaging サーバーは、連絡先リストやその登録情報などのエンドユーザー情報を取得または設定するときに、LDAP に直接照会する必要があります。

ポータルベースまたはアーカイブを使用する Instant Messaging アーキテクチャー

メッセージアーカイブをサポートするとともに Instant Messaging がセキュリティー保護されたモードで実行されるように、Instant Messaging を配備することができます。この機能を提供する Instant Messaging アーキテクチャーは、290 ページの「Instant Messaging の基本アーキテクチャー」と同じ機能も提供します。また、Portal Server デスクトップによりエンドユーザーは Instant Messenger クライアントを利用することができます。この機能を利用するには、290 ページの「Instant Messaging の基本アーキテクチャー」に記載されたコンポーネントのほかに、Portal Server と Access Manager もインストールする必要があります。

このアーキテクチャーでは、Access Manager がアクセスするディレクトリと Web サーバーが使用されます。これらのサーバーの追加インスタンスをインストールする必要はありません。また、Access Manager が必要となるこのアーキテクチャーでは、296 ページの「Access Manager または SSO を使用する Instant Messaging アーキテクチャー」で説明したすべての機能も利用できます。

図 23–7 は、ポータルベース Instant Messaging アーキテクチャーを示したものです。

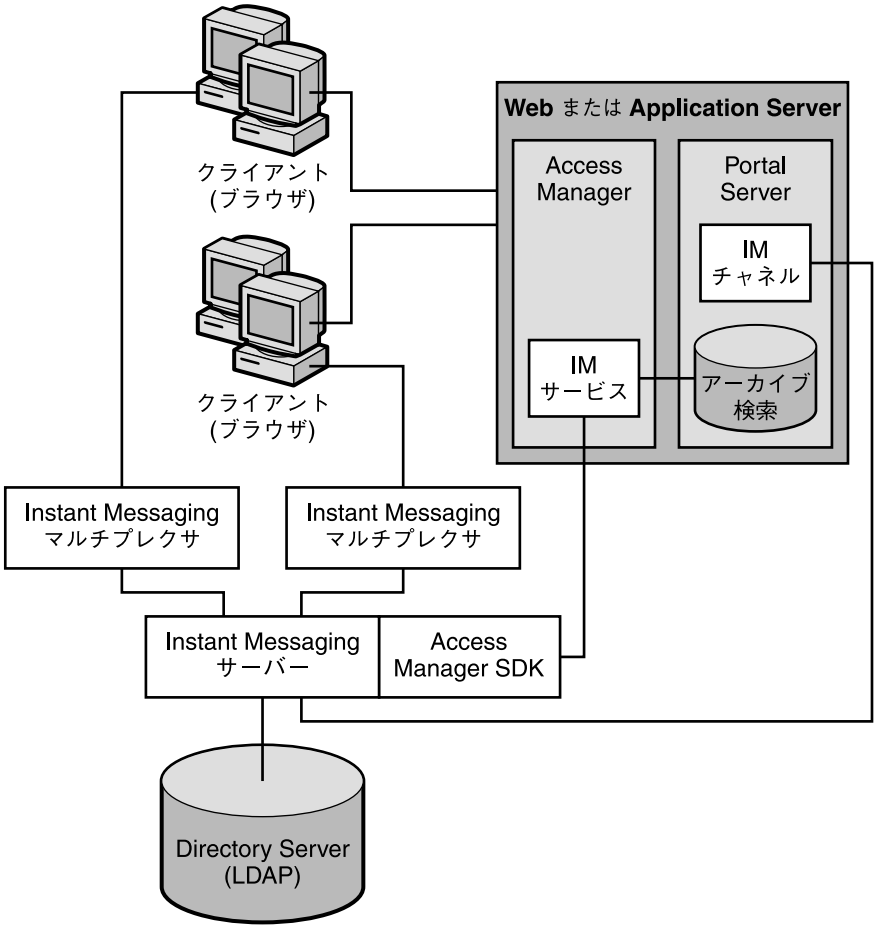

図 **23–7** ポータルベースのセキュリティー保護されたモードまたはアーカイブを使用する Instant Messaging アーキテクチャー

この例では、次のようにします。

- ユーザーエントリは LDAP サーバーに保持されます。
- Web サーバー (または Web サーバーが組み込まれたアプリケーションサーバー) は、ブラウザ経由でクライアントに Instant Messaging リソースをダウンロードします。リソースは、基本的にはクライアントです。

- Instant Messaging クライアントは常にマルチプレクサ経由で接続します。
- Access Manager が提供する Instant Messaging 関連サービスには、在席確認サービスとインスタントメッセージングサービスがあります。
- Instant Messaging 配備で ID ベースのサービスを管理する Access Manager 管理インタフェースには、Web サーバーを使用してアクセスすることができます。Access Manager と Portal Server の両方に対する Web サーバーは、Instant Messaging リソースのサーバーと同じであってもかまいません。詳細については、Sun Java System Access Manager と Sun Java System Portal Server のマニュアルを参照してください。
- Access Manager SDK は、Instant Messaging サーバーが Access Manager との通信時に使用する API を提供します。
- Portal Server は Instant Messaging チャネルをサポートし、ユーザーは Portal デスクトップから Instant Messenger にアクセスできます。
- Portal Server は、この配備で送信されるインスタントメッセージを保存するためのアーカイブ機能を提供します。

Portal Server アーキテクチャーにおける認証

図 23–8 は、シングルサインオン環境において、Portal Server および Access Manager と連携する Instant Messaging ソフトウェアによって使用される認証プロセスを示したものです。図 23–2 と同様に、この図も認証要求のフローを示しています。このプロセスの各段階の説明は、図の後に記載しています。

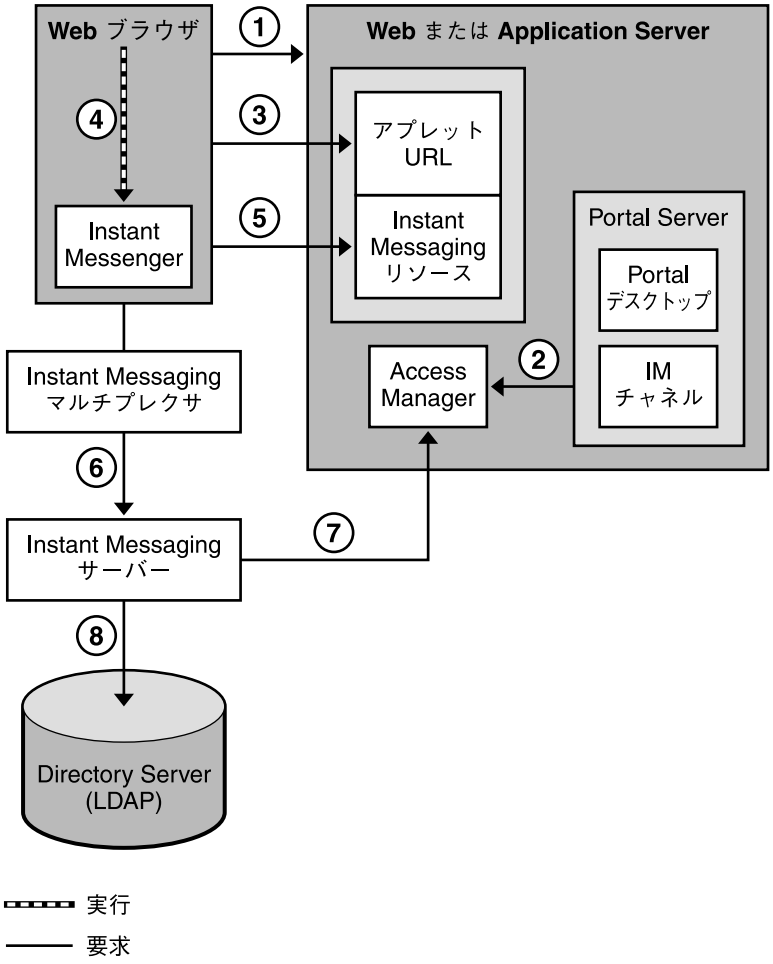

図 **23–8** Portal Server と Access Manager を伴う構成における認証要求のフロー

シングルサインオン環境において、この配備の Instant Messaging サーバーの認証プロセスは、次のように機能します。

1. ユーザーは、Web ブラウザに適切な URL を入力し、Portal Server にログインします。
2. Access Manager ソフトウェアはエンドユーザーを認証し、セッショントークンを返します。Portal Server によりエンドユーザーはデスクトップをダウンロードすることができます。Portal Server デスクトップは、エンドユーザーのブラウザに表示されます。セッショントークンの説明については、手順 6 を参照してください。
3. エンドユーザーは、デスクトップの Instant Messaging チャネルで Instant Messenger URL リンクをクリックします。
4. ブラウザが Java Web Start または Java プラグインを起動します。

5. Java Web Start または Java プラグインは、適切な Instant Messenger リソースファイルをダウンロードし、Instant Messenger を起動します。
6. Instant Messenger は、セッショントークンを使用して Instant Messaging サーバーへの認証を要求します。

   シングルサインオンが機能するには、セッショントークンが必要です。このトークンはアプレットパラメータとして提供され、認証プロセス全体で使用されます。セッショントークンがある限り、資格の再入力はエンドユーザーに求められません。
7. Instant Messaging サーバーは、セッショントークンの検証を Access Manager に求めます。セッションが有効であれば、Instant Messenger はエンドユーザーの連絡先リストを表示し、エンドユーザーはチャット、アラート、ポーリングなどの Instant Messenger サービスを利用できるようになります。
8. Instant Messaging サーバーは、連絡先リストやその登録情報などのエンドユーザー情報を取得または設定するときに、LDAP に直接照会する必要があります。

すべての機能が有効な Instant Messaging

Instant Messaging を配備し、この節で説明してきたすべての機能を有効にするには、次のようにします。

- Instant Messaging をインストールする前に次のコンポーネントをインストールします。
  - Directory Server (Access Manager インストール時)
  - Web Server (Access Manager インストール時)
  - Access Manager
  - Portal Server
  - Calendar Server
  - Messaging Server
- Instant Messaging リソースを Web Server ホストにインストールします。
- Access Manager SDK を Instant Messaging サーバーホストにインストールします。

また、Access Manager ホスト上で Access Manager Instant Messaging サービスを設定する必要もあります。

Instant Messaging の物理的な配備例

ここでは、290 ページの「Instant Messaging の基本アーキテクチャー」で説明した配備シナリオのバリエーションを説明します。たとえば、必要となる各種サーバーおよびコンポーネントを次の物理構成にインストールすることができます。

- 304 ページの「Instant Messaging の物理的な配備例: Web Server を別ホストにインストール」
- 305 ページの「Instant Messaging の物理的な配備例: マルチプレクサを別ホストにインストール」
- 306 ページの「Instant Messaging の物理的な配備例: 複数の Instant Messaging ホスト」
- 上記の一部またはすべての組み合わせ

これらのバリエーションは、この章で説明したすべてのアーキテクチャーに適用できます。配備要件に合わせて選択してください。

Instant Messaging の物理的な配備例: Web Server を別ホストにインストール

図 23–9 は、Instant Messaging サーバーとマルチプレクサが同一ホスト上にインストールされる構成を示したものです。Web サーバーは別ホスト上にインストールされます。Web サーバーホストには、Instant Messaging リソースも格納されます。Web サーバーと LDAP サーバーのインスタンスがすでに存在し、これらのホストに他のアプリケーションをインストールしない場合は、この構成を採用します。

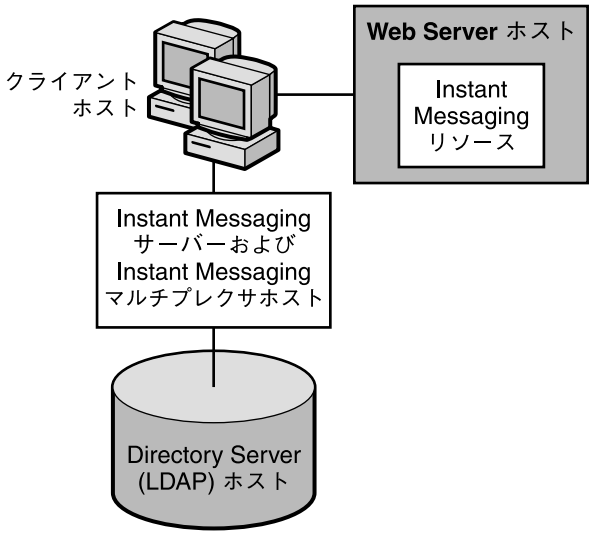

図 **23–9** Web サーバーと Instant Messaging サーバーを別々のホストにインストール

Instant Messaging の物理的な配備例: マルチプレクサを別ホストにインストール

図 23–10 は、2 つのマルチプレクサが 2 つの異なるホスト上にインストールされる構成を示したものです。Instant Messaging サーバーは別のホスト上にインストールされます。この構成では、企業のファイアウォールの外にマルチプレクサを置くことができます。複数のホストにマルチプレクサをインストールすると、Instant Messaging サーバーの負荷は複数のシステムに分散されます。

注 – マルチプレクサはリソースを大量に消費する場合があるので、別のホストに置くことでシステム全体のパフォーマンスを向上させることができます。

Windows 環境では、1 つのホストでサポートされるマルチプレクサは 1 つだけです。

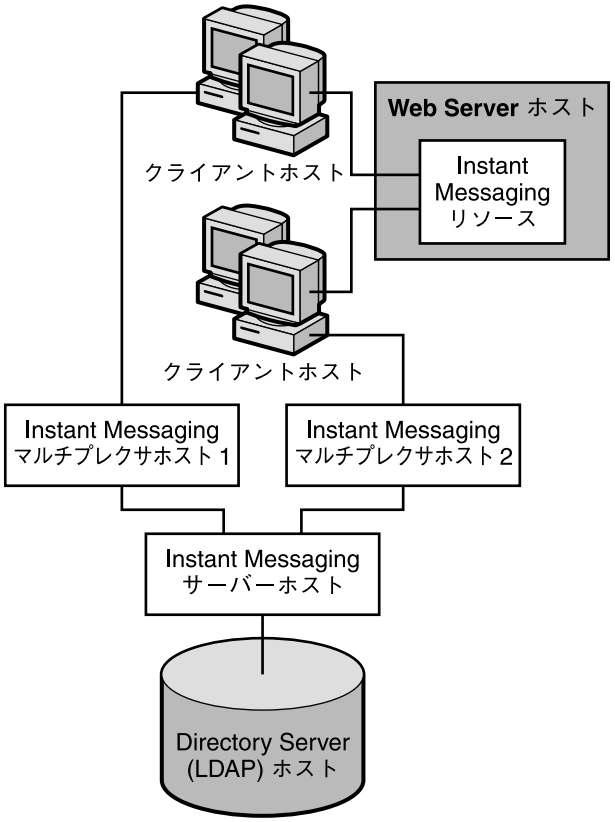

図 **23–10** Instant Messaging マルチプレクサの別ホストへのインストール

Instant Messaging の物理的な配備例: 複数の Instant Messaging ホスト

図 23–11 は、2 つの Instant Messaging サーバーによる構成を示しています。この構成は、管理ドメインが複数ある場合に採用されます。Instant Messaging サーバーの各ホストでは、一方の Instant Messaging サーバーのエンドユーザーが、もう一方の Instant Messaging サーバーのエンドユーザーと通信できるようにサーバーを設定する必要があります。

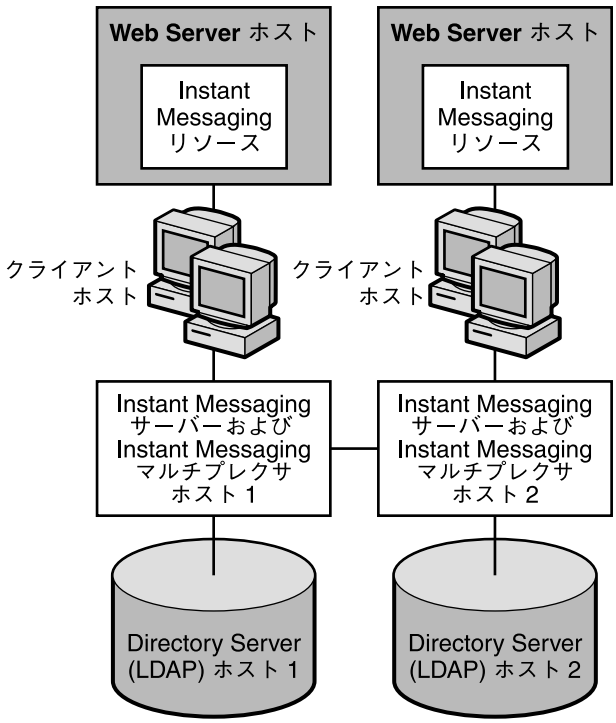

図 **23–11** 複数の Instant Messaging サーバーホスト

第 24 章

Instant Messaging のインストール前の考慮事項について

この章では、Instant Messaging のインストール前に考慮が必要な事項について説明します。Java Enterprise System インストーラの実行手順については、『Sun Java Enterprise System 2005Q4 Installation Guide for UNIX』を参照してください。

この章には、次の節があります。

Instant Messaging のインストールの概要

Solaris システム上で Instant Messaging をインストールするには、Java Enterprise System インストーラを使用します。Linux および Windows システム上では、それぞれのメディアキット CD に含まれるセットアッププログラムを使用します。また、このソフトウェアは、次のサイトからダウンロードすることもできます。

```
http://www.sun.com/software/download
```

Java Enterprise System と Instant Messaging のマニュアルには、インストールやアップグレード、サーバーの設定、クライアントの設定などに関する手順やツール情報が記載されています。そうした詳しいインストール手順や設定手順については、次のマニュアルを参照してください。

『Sun Java Enterprise System 2005Q4 Installation Guide for UNIX』

『Sun Java Enterprise System 2005Q4 アップグレードガイド』

『Sun Java System Instant Messaging 7 2005Q1 Administration Guide』

インストールを始める前に、『Sun Java System Communications Services 2005Q4 リリースノート』の第 3 章「Sun Java System Instant Messaging 7 2005Q4 リリースノート」でハードウェア要件、ソフトウェア要件、およびサポートされているバージョンを確認してください。

Instant Messaging をインストールする前に、Directory Server、Web Server、および必要に応じて Messaging Server をインストールする必要があります。さらに、Solaris システム上で Access Manager と Portal Server が提供する機能を Instant Messaging から使用する場合には、それらのサーバーもインストールする必要があります。ほかのサーバーとの連携については、265 ページの「Instant Messaging の関連コンポーネント」を参照してください。さらに、第 23 章では、Instant Messaging の各種機能を活用するうえで参考になるアーキテクチャーを、いくつか紹介しています。

Instant Messaging ワークシート

ユーザーは、インストールまたはアップグレード時に基本的な設定情報の入力を求められます。このような情報は事前に収集しておいてください。ユーザーはそうした情報の一部またはすべての入力を求められますが、そのどちらになるかは、ユーザーがどのコンポーネントをインストール対象として選択するかによります。

表 24–1 を印刷し、配備時の値を空白部分に記入してください。このインストール用ワークシートは、複数のインストールやアンインストール、アップグレードなどに再利用できます。この表にはパスワードなどの機密情報が含まれています。したがって、この情報を安全な場所に保管することをお勧めします。

表 **24–1** Instant Messaging インストールパラメータ

パラメータ	説明	実際の設定値
Installation Directory	*instant-messaging-install-dir* または *installation directory*。 Instant Messaging のインストール先ディレクトリ。 デフォルト: Solaris システム: `/opt/SUNWiim` Linux システム: `/opt/sun/im` Windows システム: `C:\Program Files\Sun\Instant Messaging`	

表 **24–1** Instant Messaging インストールパラメータ　(続き)

パラメータ	説明	実際の設定値
Instant Messaging Server Host and Domain Name	Instant Messaging のインストール先ホスト名と、そのホストに関連付けられたドメイン名。例: ホスト名: `instantmessaging.siroe.com` ドメイン名: `siroe.com`	
Instant Messaging Server Port Number	Instant Messenger クライアント以外から着信した要求に対する Instant Messaging サーバーの待機ポートの番号。 デフォルト: 49999	
Multiplexor Port Number (マルチプレクサ構成のみ)	Instant Messenger クライアントから着信した要求に対する Instant Messaging サーバーの待機ポートの番号。 デフォルト: 49909	
Disable Server	インストールしたインスタンスをサーバーとしてではなくマルチプレクサとして動作させる場合に、このオプションを選択します。このオプションを選択した場合、Remote Instant Messaging Server Host Name (マルチプレクサ構成のみ) の値を入力する必要があります。	
Remote Instant Messaging Server Host Name (マルチプレクサ構成のみ)	このマルチプレクサがメッセージをルーティングする Instant Messaging サーバーのホスト名。設定対象のインストール済みインスタンスが、マルチプレクサではなく Instant Messaging サーバーである場合には、このパラメータの値を入力しないでください。 依存関係: Disable Server パラメータを選択し、サーバー機能を無効にする必要があります。	
Assign Instant Messaging Services to existing users (省略可能)	このオプションを選択した場合、既存の Access Manager ユーザーに対して Instant Messaging が有効になります。 依存関係: Portal Server と Access Manager。	

表 **24–1** Instant Messaging インストールパラメータ　(続き)

パラメータ	説明	実際の設定値
Secure Mode (省略可能)	これを選択した場合、Portal Server Secure Remote Access との統合化が有効になります。 Secure Remote Access は、イントラネット内のリモートユーザーにセキュアアクセスを提供します。ユーザーは、ポータルゲートウェイを介して Web ベースの Portal Server Desktop にログインすることで、Secure Remote Access にアクセスできます。 依存関係: Portal Server と Access Manager が必要です。 Instant Messaging をセキュリティー保護されたモードで実行できるのは、Secure Remote Access が設定されている場合だけです。手順については、『Sun Java System Instant Messaging 7 2005Q1 Administration Guide』および『Sun Java System Portal Server 6 2005Q4 Secure Remote Access 管理ガイド』を参照してください。 この機能を有効にした場合、次にパラメータの値を入力する必要があります。 ■ Netlet Instant Messaging Port Number (省略可能) ■ Messenger Secure Download Port (省略可能)	
Netlet Instant Messaging Port Number (省略可能)	Secure Mode (省略可能) を有効にした場合、これが着信要求に対する Netlet の待機ポートの番号になります。 デフォルト: 49917 依存関係: Secure Mode (省略可能) の有効化、Portal Server、および Access Manager。	

表 **24–1** Instant Messaging インストールパラメータ　(続き)

パラメータ	説明	実際の設定値
Messenger Secure Download Port (省略可能)	Secure Mode (省略可能) を有効にした場合、これが、 Instant Messenger リソースを Netlet 経由でダウンロードする際のポート番号になります。 デフォルト: 49916 依存関係: Secure Mode (省略可能) の有効化、Portal Server、および Access Manager。	
Enable Instant Messaging Archive (省略可能)	これを選択した場合、Instant Messaging に対する Portal Server 検索ベースアーカイブが有効になります。 依存関係: Portal Server と Access Manager。	
LDAP Host Name	Instant Messaging に対するユーザーとグループの情報が格納された LDAP サーバーのホスト名。たとえば、`directory.siroe.com` など。 依存関係: Directory Server などの LDAP サーバー。	
LDAP Port Number	着信した要求に対するディレクトリサーバーの待機ポート番号。たとえば、`389` など。 依存関係: Directory Server などの LDAP サーバー。	
Base DN	Instant Messaging のユーザーとグループの情報が格納されているディレクトリツリー内のベース識別名。たとえば、`o=siroe.com` など。 依存関係: Directory Server などの LDAP サーバー。	

表 **24–1** Instant Messaging インストールパラメータ　(続き)

パラメータ	説明	実際の設定値
Bind DN	インストール中に、ディレクトリマネージャの バインド DN とパスワードを使用する必要があります。この情報に基づき、インスタントメッセージングと在席確認サービスのテンプレートと属性のみを使ってディレクトリスキーマが更新されます。これにはディレクトリマネージャのアクセス権が必要となります。インストールや初期設定の終了後に、ディレクトリマネージャのバインド DN とパスワードが保存または使用されることはありません。 サーバー構成の場合、Instant Messaging はこのバインド DN を使ってディレクトリ内のユーザーとグループを検索します。匿名でのディレクトリ検索が可能である場合は、これを空白のままにしてください。 依存関係: Directory Server などの LDAP サーバー。	
Bind Password	Bind DN のパスワード。	
SMTP Server Host Name (省略可能)	オフラインユーザーにメッセージ通知を電子メールで送信する際に使用する SMTP サーバーのホスト名。たとえば、`mail.siroe.com` など。SMTP サーバーが 25 以外のポートを使用する場合、ホスト名のほかにそのポートも指定します。たとえば、SMTP サーバーがポート `1025` を使用する場合、次のように指定します。 `mail.siroe.com:1025` 依存関係: Messaging Server などの SMTP サーバー。	
Database, Logs, and Runtime File Pathname	実行時ファイル、データベース、およびログの格納先。 デフォルト: Solaris システム: `/var/opt/SUNWiim/default` Linux システム: `/var/opt/sun/im` Windows システム: `C:\Program Files\Sun\Instant Messaging`	

表 **24–1** Instant Messaging インストールパラメータ　(続き)

パラメータ	説明	実際の設定値
Resources and Help Files Pathname	*instant-messaging-resource-directory* または *resource directory* リソースファイルとオンラインヘルプファイルのインストール先ディレクトリ。 デフォルト: Solaris システム: `/opt/SUNWiim/html` Linux システム: `/opt/sun/im/html` Windows システム: `C:\Program Files\Sun\Instant Messaging\html`	

表 **24–1** Instant Messaging インストールパラメータ　(続き)

パラメータ	説明	実際の設定値

Code Base	Instant Messenger がリソースをダウンロードする URL。
	
	リソースは、Web サーバーのドキュメントルート内にインストールします。たとえば、Web サーバー `www.example.com` の待機ポートが `89`、ドキュメントルートが `/opt/web/` であり、Instant Messenger リソースを `/opt/web/im` にインストールする場合は、Instant Messenger リソースのコードベースは次のようになります。
	
	`http://www.example.com:89/im/`
	
	インストール時に正しい `codebase` を入力しなかった場合、Instant Messenger の起動ページ *codebase* `/`*lang*`/im[ssl].html` と *codebase*`/`*lang*`/im[ssl].jnlp` 内の URL を、正しい値に更新する必要があります。
	
	UNIX の場合、リソースを任意のディレクトリにインストールし、シンボリックリンクを使ってそのリソースを Web サーバーから見えるようにする、といったことも可能です。
	
	たとえば、前述した例で、`/opt/SUNWiim/html` 内に Instant Messenger リソースをインストールした場合、そのリソースを Web サーバーから見えるようにするには、次のようなシンボリックリンクを作成します。
	
	`ln -s /opt/SUNWiim/html /opt/web/im`
	
	詳細については、『Sun Java System Instant Messaging 7 2005Q1 Administration Guide』と Web サーバーのマニュアルを参照してください。	

パート V Communications Express の配備

この部には、次の章があります。

第 25 章

Communications Express ソフトウェアの紹介

Communications Express は、通信およびコラボレーション用の Web ベースの統合クライアントです。Communications Express は Messaging Server と Calendar Server の共通ソフトウェアであり、カレンダ情報、メール、およびアドレス帳に対する Web インタフェースをエンドユーザーに対して提供します。

Communications Express は、カレンダ、アドレス帳、メールの 3 つのクライアントモジュールで構成されます。

この章には、次の節があります。

Communications Express の概要

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

- Directory Server
- Access Manager (Sun Java System LDAP スキーマバージョン 2 を使用する場合)
- Calendar Server
- Messaging Server
- Web Server または Application Server (Web コンテナとして)

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。また、Communications Express と Messenger Express の両方を、同一 IP アドレス上で実行する必要があります。Messaging Server パッケージは、Messenger Express として動作するように構成することもできますし、Messenger Express が実行されているバックエンドストアに接続する MEM として動作するように構成することもできます。

さらに、フロントエンドマシン上の Communications Express のアドレス帳を設定して、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が格納されるようにすることができます。詳細については、『Sun Java System Communications Express 6 2005Q4 管理ガイド』を参照してください。

Communications Express は、Calendar Server との通信に Calendar Server HTTP サービスを、Messaging Server との通信に `mshttpd` デーモンを、アドレス帳との通信に LDAP サービスを、それぞれ使用します。`cshttpd` デーモンはローカル、リモートのいずれかに、`mshttpd` デーモンはローカル Web メールサーバー、ローカル MEM のいずれかに、LDAP サービスはローカル、リモートのいずれかに、それぞれ設定できます。

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」(持続的) な接続を使用してください。

Communications Express の機能

- Communications Express はカレンダ、メール、およびアドレス帳に対する統合ユーザーインタフェースを備えており、あるクライアントモジュールから別のクライアントモジュールへとアクセス先を変更しても、ユーザー資格の再認証を行う必要がありません。
- メールとカレンダ間の通信は、Access Manager または Messaging Server のシングルサインオンメカニズムを使って確立されます。
- カレンダアプリケーションとメールアプリケーションは、同一のアドレス帳を共有します。
- Communications Express の「オプション」タブで指定されたユーザー設定を、すべてのモジュールが共有します。
- アドレス帳ストアは水平方向のスケーラビリティを提供します。詳細については、『Sun Java System Communications Express 6 2005Q4 管理ガイド』を参照してください。
- Communications Express は仮想ドメインをサポートします。

Communications Express の高レベルのアーキテクチャー

カレンダクライアントモジュールとアドレス帳クライアントモジュールは、Web コンテナ、つまり Sun Java Systems Web Server または Sun Java Systems Application Server のいずれかに単一の Web アプリケーションとして配備されます。メールモジュールは、Messenger Express によってレンダリングされます。Messenger Express は、Messaging Server の HTTP サービスを使用する、スタンドアロンの Web ベースのメールアプリケーションです。

Messenger Express または MEM は、Communications Express の配備先と同じシステム上に配備する必要があります。

図 25–1 は、Communications Express ソフトウェアのアーキテクチャーを示したものです。

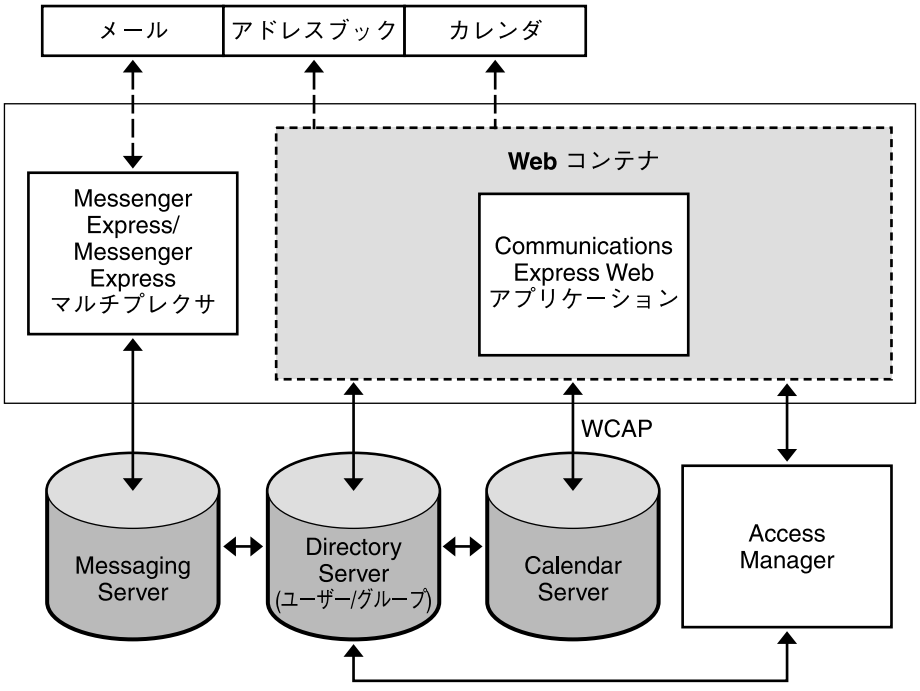

図 **25–1** Communications Express ソフトウェアの高レベルのアーキテクチャー

Communications Express は、次のモジュールで構成されます。

- メール:メールコンポーネントは、クライアントによって読み取られ、解釈される JavaScript 言語を使用します。JavaScript ファイルはサーバーに配置され、クライアントにダウンロードされます。クライアントは JavaScript コードからデータを抽出し、Communications Express 機能をカスタマイズします。すべての変更およびカスタマイズは、サーバー上で行われます。
- カレンダ:カレンダモジュールのプレゼンテーション層は、JavaServer Pages™ に基づいています。これらの JavaServer Pages ページは、クライアントの要件に合わせてカスタマイズできます。データ層は、JCAPI (Java API for Calendar) にアクセスすることで、HTTP ベースプロトコル経由での Calendar Server とのデータ交換を可能にします。
- アドレス帳:アドレス帳コンポーネントは、XSL タグ、静的 HTML、および `.js` スクリプトを含む XML/XSL ファイルを使用します。XSL および JavaScript コードは、動的なデータの表示に使用します。これらの XSL ファイルは、アドレス帳コンポーネントをカスタマイズするように編集できます。

第 26 章

Communications Express アーキテクチャーの開発

この章では、Communications Express の基本的な配備アーキテクチャーについて説明し 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ーをインストールする必要があります。

この章には、次の節があります。

Communications Express 基本アーキテクチャー

この Communications Express 基本アーキテクチャーでは、カレンダモジュール、アドレス帳モジュール、およびメールモジュールが、単一ホスト上の Web コンテナ内に配置されます。Messenger Express は、Messaging Server の HTTP サービスを使用する、スタンドアロンの Web インタフェースメールアプリケーションです。Messenger Express は、カレンダモジュールとアドレス帳モジュールと同じシステム上に配備されます。

この基本機能を利用するには、次のコンポーネントをインストールする必要があります。

- Directory Server
- Access Manager (Sun Java System LDAP スキーマバージョン 2 を使用する場合)
- Calendar Server
- Messaging Server
- Web Server または Application Server (Web コンテナとして)

この例では、次のようにします。

- Communications Express を実行するホスト上に Messaging Server パッケージの完全なセットをインストールします。
- Communications Express のアドレス帳サーバーを設定し、そのデータが LDAP ディレクトリインフラストラクチャー内に格納されるようにします。
- SSL は設定しません。

図 26–1 は、Communications Express の基本アーキテクチャーを示したものです。

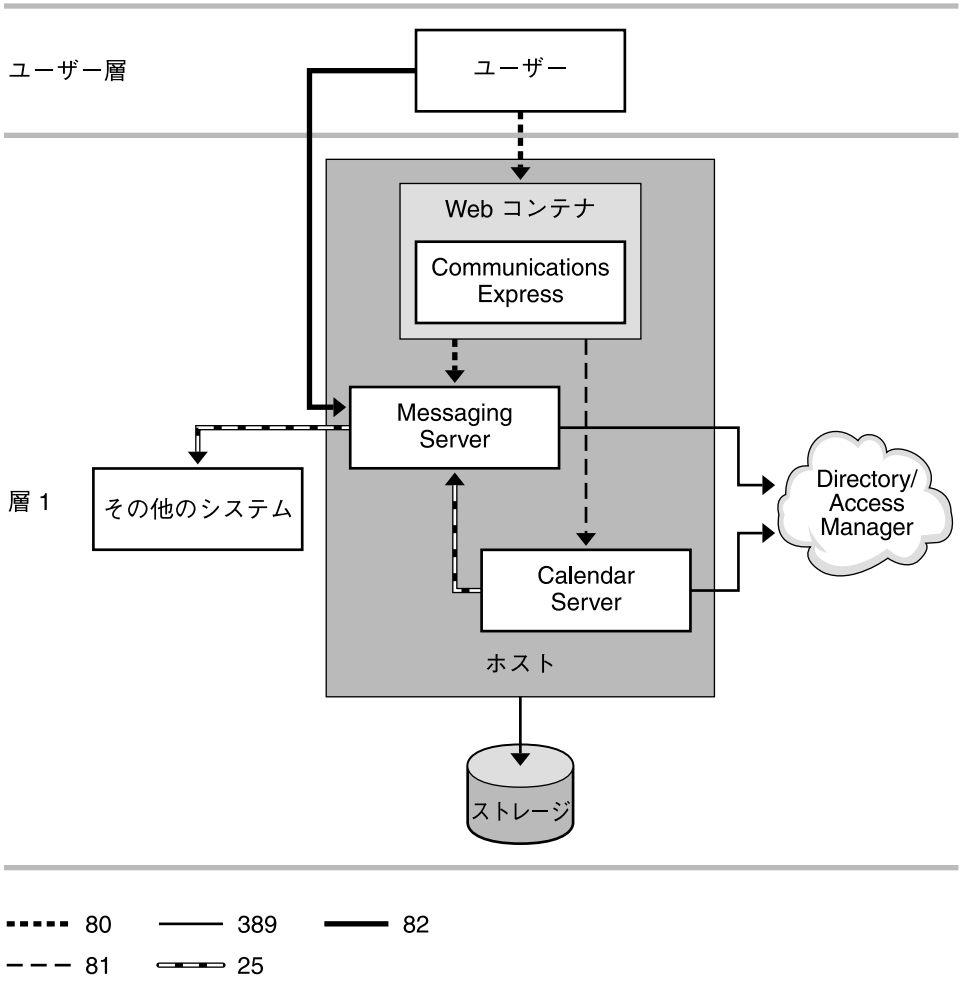

図 **26–1** Communications Express 基本アーキテクチャー

次の表は、このアーキテクチャーで使用するプロトコルとポート番号について説明しています。

表 **26–1** Communications Express の基本配備アーキテクチャーで使用されるプロトコルとポート

プロトコル	ポート	用途
SMTP	25	他のシステムと通信する Messaging Server MTA コンポーネント、および電子メール通知用 Calendar Server (`csenpd`) コンポーネント
HTTP	80	Communications Express フロントエンドと通信するインターネットユーザー、および Messaging Server と通信する Communications Express
HTTP	81	Calendar Server と通信する Communications Express 上の Calendar Express
MSHTTP	82	Messenger Express と通信するインターネットユーザー
LDAP	389	LDAP ディレクトリと通信する Messaging Server と Calendar Server

リモートホストアーキテクチャーの Communications Express

図 26–2 は、イントラネットユーザーとインターネットユーザーの双方に対応した Communications Express アーキテクチャーを示しています。イントラネットユーザーは、Communications Express バックエンドホストにログオンします。インターネットユーザーは、DMZ 内の Communications Express フロントエンドホストにログオンします。すると、そのフロントエンドホストがバックエンドホストと通信します。シングルサインオンはバックエンドホスト上で有効化されます。

フロントエンドホストには、次のコンポーネントをインストールします。

- Communications Express
- Web コンテナ
- Messaging Express マルチプレクサ
- Access Manager SDK

バックエンドには、次のコンポーネントをインストールします。

- Communications Express
- Web コンテナ
- Messaging Server (Messenger Express)
- Calendar Server
- Directory Server
- Access Manager

図 26–2 は、リモートホストアーキテクチャーの Communications Express を示しています。

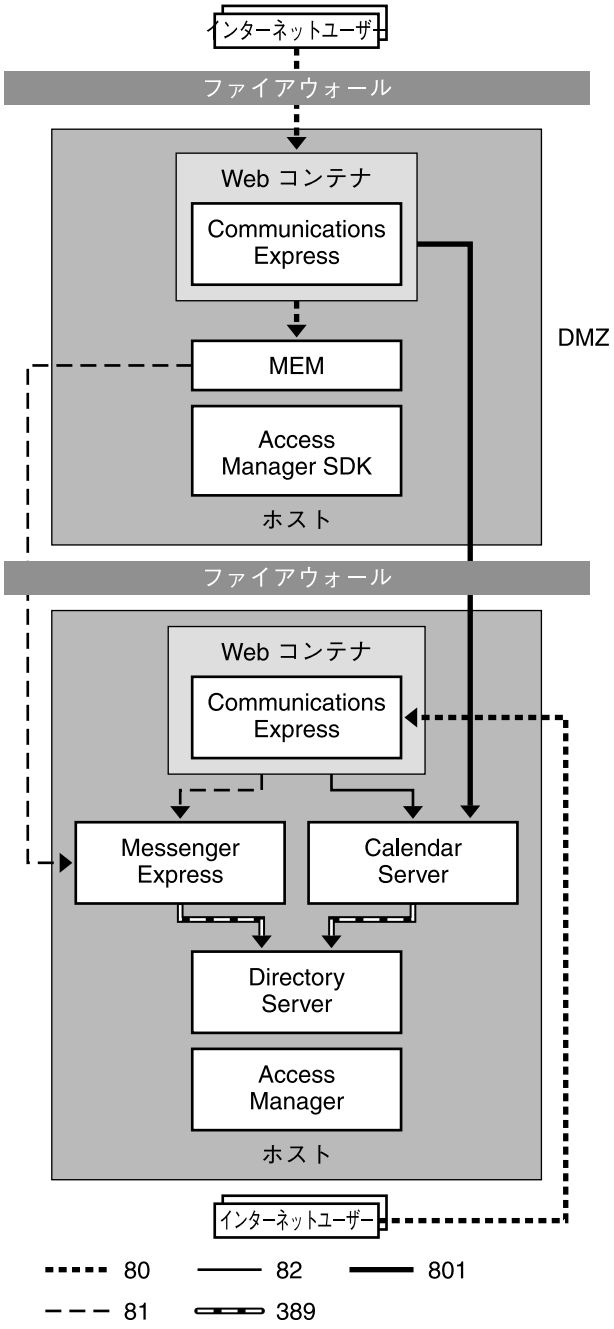

図 **26–2** リモートホストアーキテクチャーの Communications Express

次の表は、このアーキテクチャー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 **26–2** Communications Express リモートホスト配備例で使用されるプロトコルとポート

プロトコル	ポート	用途
HTTP	80	DMZ 内の Communications Express フロントエンドホストと通信するインターネットユーザー
HTTP	81	DMZ の背後にあるバックエンドホスト上の Messenger Express と通信する、DMZ 内の Communications Express フロントエンドホスト上の Messaging Express マルチプレクサ (MEM)
HTTP	82	同じくバックエンドホスト上に存在する Calendar Server と通信する、バックエンドホスト上の Communications Express
LDAP	389	LDAP ディレクトリと通信する Messaging Server と Calendar Server
HTTP	8081	バックエンドホスト上に存在する Calendar Server と通信する、フロントエンドホスト上の Communications Express

第 27 章

Communications Express のインストール前の考慮事項について

この章では、Communications Express のインストール前に考慮が必要な事項について説明します。

この章には、次の節があります。

Communications Express インストール時の考慮事項

Communications Express をインストールする前に、次の計画局面を検討してください。

- Delegated Administrator を使用する場合、Access Manager と Web コンテナ (Web Server または Application Server) を同一ホスト上にインストールする必要があります。
- Communications Express と Access Manager は、SSL モード、非 SSL モードのどちらでも配備可能です。また、両者の配備先 Web コンテナは、同じであっても異なっていてもかまいません。
- JavaScript のセキュリティ上の理由により、Communications Express と Messenger Express (多層環境の場合は Communications Express と Messaging Express マルチプレクサ) を同一ホスト上にインストールする必要があります。
- Directory Server、Messaging Server、Calendar Server、Access Manager をそれぞれ異なるホスト上にインストールするような分散配備を計画してもかまいません。
- Calendar Server のホストしているドメインを使用する場合、設定フェーズの間は Communications Express のホストしているドメインのサポートを有効にします。

- Communications Express は、SSL 用にも非 SSL 用にも設定できます。SSL を設定した場合、Communications Express クライアントが SSL を認証時にのみ使用するようにするか、セッションを通じて使用するようにするかを選択できます。

Communications Express メールで S/MIME を使用するための要件

Communications Express メールで、S/MIME (Secure/Multipurpose Internet Mail Extension) のセキュリティ機能が利用可能になりました。S/MIME を使用するように設定された Communications Express Mail ユーザーは、ほかの Communications Express Mail ユーザーや、Microsoft Outlook メールシステムなどの S/MIME をサポートするメールクライアントのユーザーと、署名または暗号化されたメッセージを交換できます。

S/MIME を使用するための一般的な要件

Communications Express メールユーザーが S/MIME の署名機能と暗号化機能を使用できるようにするための要件は、次のとおりです。

- 公開鍵と 非公開鍵のペアが標準 X.509 形式の証明書とともに発行されている。証明書は、ほかのメールユーザーに対して、その鍵の使用者が本当にその鍵の所有者であることを保証します。鍵と証明書は、組織内で発行されるか、サードパーティーのベンダから購入されます。鍵と証明書の発行方法にかかわらず、その発行元は認証局 (CA) と呼ばれます。
- 公開鍵 / 非公開鍵ペアとその証明書が、ローカルのキーストア内に適切かつ電子的に格納されているか、スマートカードと呼ばれる CAC (Common Access Card) 経由でエンドユーザーに配付されている。
- すべての公開鍵と証明書が、Directory Server 経由でアクセス可能な LDAP ディレクトリ内に格納されている。これは「公開鍵の発行」と呼ばれ、これにより、S/MIME メッセージを作成するほかのメールユーザーが公開鍵を利用できるようになります。
- 公開鍵 / 非公開鍵ペアとその証明書をスマートカードに格納する場合、カード読み取りデバイスがクライアントマシン上に正しく設置されている。
- Communications Express メールにアクセスするクライアントマシン上に、すべての必要なプラットフォームソフトウェアがインストールされている。
- すべての必要な Sun Microsystems ソフトウェアがインストールされており、S/MIME 用に設定されている。
- Communications Express メールユーザーが Sun Microsystems メールシステムを使用するように設定されている。これには、S/MIME 機能の使用権限をユーザーに与えることも含みます。

S/MIME 配備前に知っておくべき概念

S/MIME 用のメールシステムを配備する前に、次の概念を熟知しているか確認してください。

- プラットフォームの基本的な管理手順
- LDAP ディレクトリの構造と使用方法
- LDAP ディレクトリに対するエントリの追加または変更
- Sun Java System Directory Server の設定プロセス
- 次の概念とその目的
  - SSL (Secure Socket Layer) によるセキュリティ保護された通信回線
  - デジタル署名された電子メールメッセージ
  - 暗号化された電子メールメッセージ
  - ブラウザのローカルキーストア
  - スマートカードと、それを使用するためのソフトウェアとハードウェア
  - 公開鍵 / 非公開鍵ペアとその証明書
  - 認証局 (CA)
  - 鍵とその証明書の検証
  - CRL (証明書取り消しリスト)

Communications Express の詳細情報の入手先

Communications Express をインストールおよび設定するには、『Sun Java System Communications Express 6 2005Q4 管理ガイド』の手順を参照してください。

S/MIME を管理するには、『Sun Java System Messaging Server 6 2005Q4 管理ガイド』の第 20 章「Communications Express メールでの S/MIME の管理」を参照してください。

パート VI　配備例

この部には、次の章があります。

- 第 28 章

第 28 章

Communications Services 配備の例

この章では、Communications Services の配備例を紹介します。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ーをインストールする必要があります。

この章には、次の節があります。

注 – 単一ホスト配備から多層配備に至るまで、さまざまなアーキテクチャーの中から選択する場合には、常に多層にわたるサービス定義を念頭において計画することをお勧めします。したがって、単一ホスト配備であっても、論理サービス名を使ってインストールしてください。論理サービス名を使用すると、配備を拡張しやすくなります。詳細については、84 ページの「論理サービス名の使用」を参照してください。

Communications Services の単一ホスト用の単一層論理配備の例

名前からもおわかりのように、この例では、コンポーネントを単一サーバー上にインストールおよび設定します。ご購入先のクライアントサービス担当者に相談しながら、最適なサーバータイプやシステム構成を決定してください。

一般に、単一層単一ホストアーキテクチャーは、次のような企業に最適です。

- ユーザー数が 1,000 人未満の場合
- 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

単一ホスト構成に関するトレードオフを次に示します。

- インフラストラクチャーとしての可用性は低く、サービスの信頼性が低い (ただし、サーバー自身が自動システム再構成を提供する場合を除く)
- サービス拒否攻撃を防止できない

図 28–1 に、単一ホスト配備の例を示します。次の Communications Services コンポーネントが同一ホスト上にインストールされます。

- Messaging Server (MTA、メッセージストア、および Messenger Express)
- Calendar Server (管理サービス、HTTP サービス、およびバックアップサービス)
- Communications Express
- Web サーバー

この例では、ディレクトリサービスは、Communications Services とは異なるホスト上に存在しています。Directory Server と Access Manager 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。

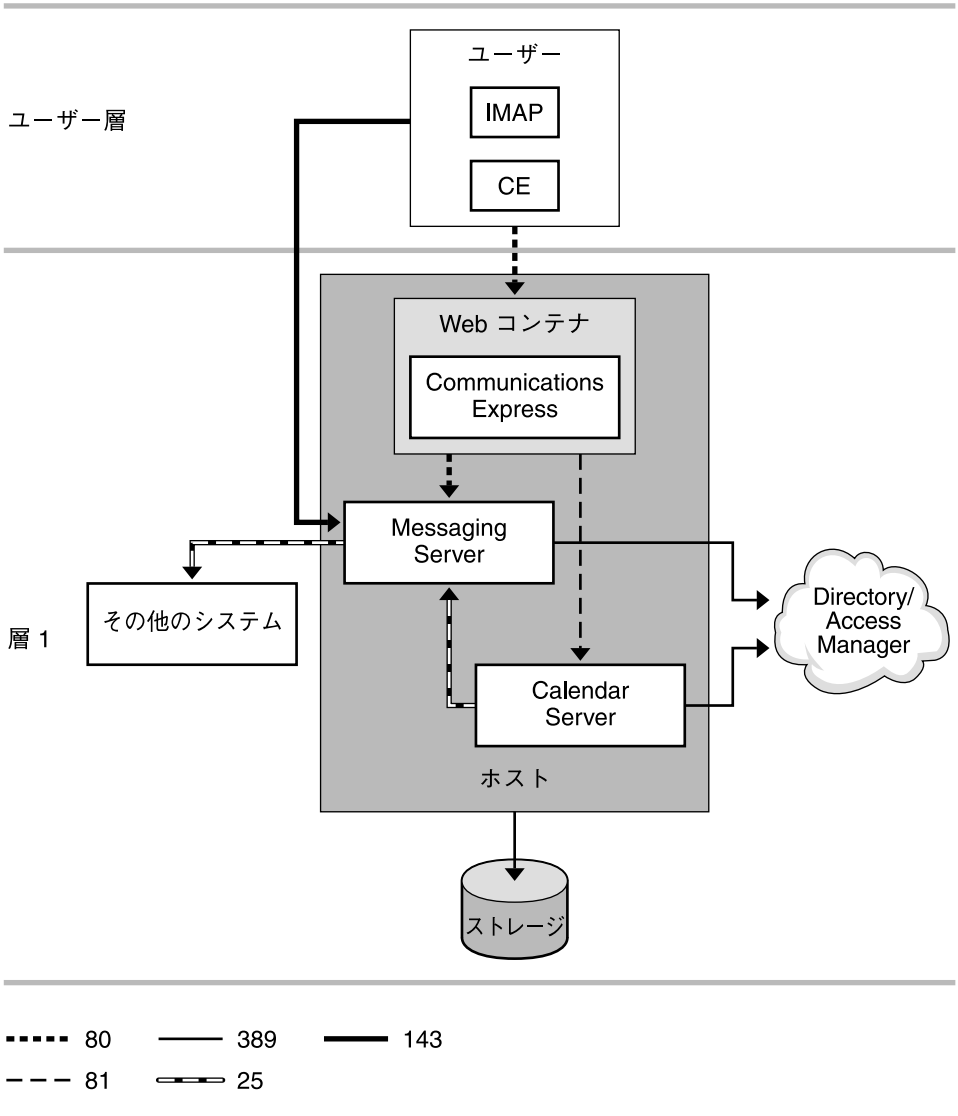

図 **28–1** Communications Services の単一ホスト用の単一層配備の例

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 28–1 単一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
SMTP	25	他のシステムと通信する Messaging Server MTA コンポーネント、および電子メール通知を送信する Calendar Server (csenpd) コンポーネント
HTTP	80	Messaging Server Web メール (httpd) コンポーネントと通信するクライアント
HTTP	81	Calendar Server (cshttpd) と通信するクライアント
IMAP	143	Messaging Server imapd コンポーネントと通信するクライアント
LDAP	389	LDAP ディレクトリと通信する Messaging Server と Calendar Server

この配備の将来的な拡張性を高めるには、論理サービス名を使ってインストールします。論理サービス名を使用すると、配備を拡張しやすくなります。詳細については、84 ページの「論理サービス名の使用」を参照してください。容量、パフォーマンス、サイトの地理的な分散、および可用性に関する問題が発生した場合は、2 層アーキテクチャーへの拡張を検討します。

Communications Services の複数ホスト用の 2 層論理配備の例

図 28–2 は、Messaging Server と Calendar Server の 2 層論理配備の例を示します。第 0 層はロードバランサで構成されます。第 1 層は 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第 2 層を形成します。

Directory Server と Access Manager 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。

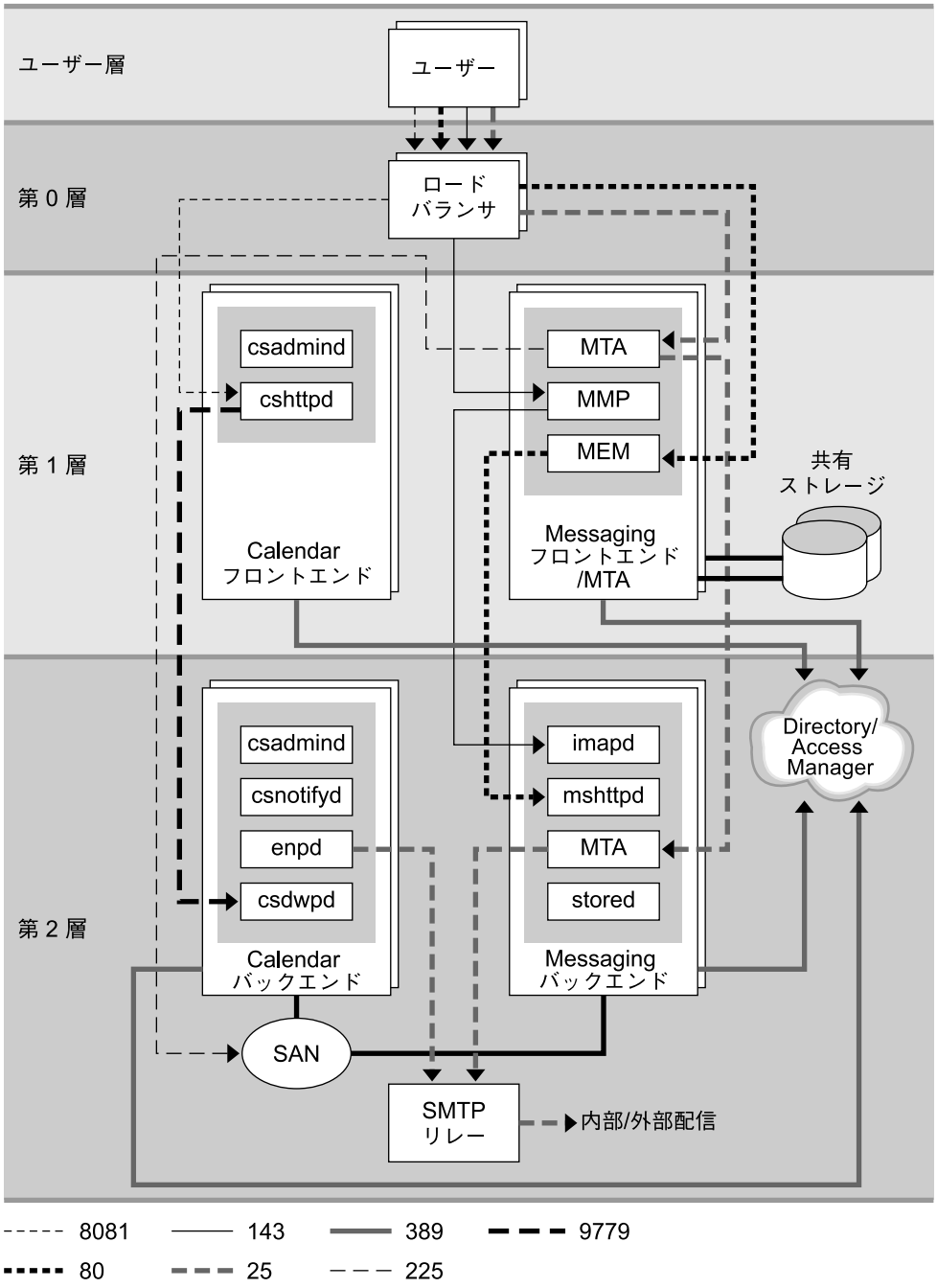

図 **28–2** Communications Services 2 層配備の例

上図の例では、ロードバランサが第 0 層を形成し、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できます。CPU の合計数がいったん決定すると、垂直および水平方向の構成を選択できます。たとえば、アーキテクチャーが合計 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。

もう一つのマシンは、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 **28–2** 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知用 Calendar Server (`csenpd`) 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	カレンダバックエンド (`csdwpd`) と通信するカレンダフロントエンド (`cshttpd`)

用語集

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、『Sun Java Enterprise System Glossary』を参照してください。

索引

D

J

L

M

N

O

P

R

S

T

U

V

W

X

き

く

け

こ

さ

と

な

に

ね

は

へ

ほ

ま

め

る

れ

ろ

わ