FUJITSU Software ServerView Suite
Basic Concepts
著作権および商標

Copyright © 2015 Fujitsu Technology Solutions GmbH.
All rights reserved

Microsoft、Windows、Windows Server、Hyper-V は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Linux は、Linus Torvalds 氏の米国およびその他の国における登録商標あるいは商標です。

Red Hat および Red Hat をベースとしたすべての商標とロゴは、米国およびその他の国における Red Hat, Inc. の商標または登録商標です。

BrightStor, ARCserve は、CA, Inc の登録商標です。

VMware、VMware ロゴ、VMware ESXi、VMware SMP および VMotion は VMware,Inc の米国およびその他の国における登録商標または商標です。

その他の各製品名は、各社の商標、または登録商標です。
その他の各製品は、各社の著作物です。
目次

1 はじめに ........................................... 5

2 基本概念 ........................................... 7

  2.1 サーバ管理コンポーネント .......................... 7
  2.1.1 監視対象サーバとエージェント ....................... 9
  2.1.2 管理アプリケーション ............................. 9
  2.1.3 管理コンソール ................................. 10
  2.1.4 ヘルパー .................................... 11
  2.2 管理の原則：エージェントと管理ステーション .......... 11
       2.2.1 管理サイクル ................................ 12
       2.2.2 監視と分析の間のインタフェース ................. 14
       2.2.3 適応と実行の間のインタフェース ................. 15
       2.2.4 サーバ上での自律管理 ......................... 16
  2.3 階層的な設定 ..................................... 17
       2.3.1 ローカル管理 ................................ 17
       2.3.2 ポイント・ツー・ポイント管理 .................... 17
       2.3.3 集中管理 .................................... 20
       2.3.4 カスケード管理 ............................... 22
       2.3.5 他の管理システムへの統合 ....................... 23
  2.4 プロトコル ...................................... 25
       2.4.1 SNMP - Simple Network Management Protocol ........ 25
       2.4.2 CIM （Common Information Model） ............... 29
       2.4.3 IPMI （Intelligent Platform Management Interface） 32
       2.4.4 PXE （Preboot Execution Environment） .......... 34
       2.4.5 Telnet ..................................... 35
       2.4.6 HTTP （Hypertext Transfer Protocol） ........... 35
       2.4.7 SOAP （Simple Object Access Protocol） .......... 36
       2.4.8 ITIL （IT Infrastructure Library） .............. 37
  2.5 ヘルパー ...................................... 38
       2.5.1 DHCP サーバ ................................ 38
       2.5.2 PXE ブートサーバ .............................. 39
       2.5.3 TFTP サーバ ................................ 39
       2.5.4 メールサーバ （SMTP） ........................ 40
2.5.5 Web サーバ

2.5.6 MS Excel、Access、または SQL データベース

2.6 管理モード

2.6.1 「インバンド」管理モード

2.6.2 「アウトオブバンド」管理モード

2.7 まとめ

3 ServerView の位置づけ

3.1 ServerView Suite

3.2 他の管理システムへの統合

3.3 他のコンポーネントの統合
1 はじめに

ServerView Suite は、ネットワーク上のサーバを集中管理するための製品を提供します。このマニュアルでは、サーバ管理アーキテクチャの基本原則の一般的な情報について説明します。基本的な内容を把握していただけるように、ServerView Suiteに関連するトピックを取りあげています。

「基本概念」の章では、最初にサーバ管理コンポーネントをカテゴリーに分類し、これらのコンポーネントを使って実現できる管理の概念を紹介します。次に、管理システム上の通信に必要となる標準プロトコルとプログラムについて説明します。また、ServerView Suite でこれらのプロトコルやプログラムがどのように使用されるかについての概要も説明します。さらに、リモートアクセス用の管理モードについて紹介します。

「ServerView の位置づけ」の章では、他のエンタープライズ管理システムとの統合の概要について説明します。

このマニュアルは、概要をすばやく把握できるとともに、各トピックの基本情報をより詳しく理解できるように構成されています。

一部のテキストは、背景がグレーに色づけられています。ここには、上級者やトピックの詳細を知りたい読者のために、特定の内容について、より詳しい情報が記載されています。ServerView Suite の標準的な使用では、これらのテキストは無視してかまいません。
2 基本概念

この項では、サーバ管理における一般的な原則と概念の概要について説明します。主として、PRIMERGY システムの計画、設定、運用を最適に行うために、理解しておく必要があるトピックをとりあげています。

2.1 サーバ管理コンポーネント

サーバ管理コンポーネントは、4 つのカテゴリに分類できます。

- 監視対象サーバ
- 管理アプリケーション
- 管理コンソールと
- ヘルパー

図1は、これらサーバ管理コンポーネントの相互関連を示す概略図です。
サーバ管理コンポーネント

これらのコンポーネントは、プロトコルを使用して相互に通信します。各コンポーネントは、それぞれ別々のコンピュータにインストールできます。異なるカテゴリの複数のコンポーネントを1台のコンピュータにインストールすることもできます。次では、4つのカテゴリについて簡単に説明します。さらに、これらのコンポーネントの組み合わせによる各種ネットワークトポロジを示し、各トポロジの利点と欠点について説明します。
サーバ管理コンポーネント

2.1.1 監視対象サーバとエージェント

サーバを管理するには、通常、そのサーバに最低1つのエージェントがインストールされている必要があります。サーバにインストールされたエージェントを使って、リモートでのサーバの監視、管理ができるようになります。このインタフェースを使って、以下を実行できます。

- 情報提供の要求を受信する
- 要求された情報を配信する
- 動作実行の要求を受信する
- 監視対象サーバ上で特殊な状況を検出した場合、ただちに自発的なメッセージ（トラップなど）を送信する

エージェントは、管理アプリケーションと通信します。エージェントは、管理アプリケーションから情報配信またはアクション実行のジョブを受信します。また、特殊な状況を検出した場合には、エージェントが自発的にメッセージ（トラップなど）を送信します。したがってエージェントは、管理アプリケーションから指示されたジョブを実行したり、管理アプリケーションに情報を提供したりすることによって、監視対象サーバ上で管理アプリケーションの「補佐」としての役割を果たします。

監視対象サーバは、SNMP、CIMベースプロトコル、IPMIなどのプロトコル（25 ページの「プロトコル」の項をご覧ください）を使用して、管理アプリケーションと通信します。

2.1.2 管理アプリケーション

管理アプリケーションは、以下のタスクを指示します。

- 複雑な設定の実行
- 複数のシステムからのデータ収集
- 一貫性のチェック
- 情報のグラフィカルな表示
- トラブル状況に対する応答
- サーバのインストール
- サーバのアップデート
- アーカイブ、レポートなどの作成
サーバ管理コンポーネント

管理アプリケーションは、次のコンポーネントと通信します。

- 管理コンソール

  コンソールは、管理者が使用するインタフェースです。管理アプリケーションの機能はすべて、コンソールから実行できます。よく使用されるコンソールは、Webブラウザです。この場合、管理アプリケーションは、HTTP/HTMLまたはHTTPS/HTML（25ページの「プロトコル」の項をご覧ください）を使用してコンソールと通信します。

- 監視対象サーバ上のエージェント

  管理アプリケーションは、エージェントと通信することによって、監視対象サーバから情報を取得し、ログファイルを読み込み、アクションや設定を実行し、特殊な状態に関する情報を入手します。

- 他の管理アプリケーション

  ServerView Suite のコンポーネントは、複数の管理アプリケーションを連携させることによって、より複雑なタスクを処理できます。この場合、管理アプリケーションは、他の管理アプリケーションから提供されたサービスを使用します。使用するプロトコルは、SOAP（25ページの「プロトコル」の項をご覧ください）または独自仕様のプロトコルです。

- ヘルパー（38ページの「ヘルパー」の項をご覧ください）

  管理アプリケーションは、ヘルパーからのサービス（OSが提供するサービスなど）を使用することもできます。この場合、ヘルパーによって提供されるインターフェースを使用して通信します。リポジトリは、管理アプリケーションのコンポーネントですが、ヘルパーとして実装することもできます。

ServerView 管理アプリケーションには、ServerView Operations Manager、ServerView Download Manager、ServerView Event Manager、ServerView Deployment Manager、ServerView Archive Managerなどがあります。

2.1.3 管理コンソール

コンソールを使うと、通信回線を介してリモートで管理アプリケーションを利用できるので、管理者は、いつでも、どこからでも、ServerView 管理アプリケーションによって提供されるサービスにアクセスできます。

ServerView Suite では、管理コンソールとして、WebブラウザやTelnetコンソールなどを使用します。
管理の原則：エージェントと管理ステーション

2.1.4 ヘルパー

ヘルパーとは、実際には ServerView Suite のコンポーネントではなくても、ServerView Suite の運用時に使用できるサービスまたはプログラムのことです。

ServerView Suite で使用するヘルパーには、ServerView Deployment Manager で使用する DHCP サービス（38 ページの「DHCP サーバ」の項）や TFTP サービス（39 ページの「TFTP サーバ」の項）、または PRIMERGY/PRIMEPOWER ServerView で作成したファイルを手動で編集するときに使用する Microsoft Excel/Access などがあります。

2.2 管理の原則：エージェントと管理ステーション

管理の原則を説明するには、前述のカテゴリについて、概要を把握しておく必要があります。ここで説明する概要は、SNMP 標準のアーキテクチャの記述に基づいています。まず、監視対象エレメント（サーバ、ネットワークコンポーネントなど）上のエージェントと、管理ステーションがあります。

- エージェント

エージェントは、監視対象サーバ上で管理アプリケーションから指示されたタスクを実行します。

- 管理ステーション

管理アプリケーション、管理コンソール、ヘルパーは相互に連携しているので、単一ユニットとしてまとめることができます。これが、管理ステーションです。管理ステーションは、エージェントに対応する存在であり、単一コンピュータ上または分散環境の複数のコンピュータ上で物理的に実行できる論理要素です。

次では、このエージェントと管理ステーションの原則に基づいて、一般的なアクションを例にあげ、管理運用の手順について説明します。
管理の原則：エージェントと管理ステーション

2.2.1 管理サイクル

サーバ管理は連続的なプロセスです。図2に、サーバ管理のサイクルを示します。

図2: 管理サイクル

このサイクルには、監視、分析、適応、実行という4つのフェーズがあります。

- 監視では、監視対象サーバとそのコンポーネントに関する現在の管理情報が提供されます。
- 分析では、管理情報から結果が引き出されます。しきい値の超過、リソースのボトルネックの予測、一連のトラップの原因の判別などです。
- 適応では、分析フェーズで検出された状況に対処するためのアクションが計画されます。例えば、再設定、サービスの再開、サーバの停止、サーバの再インストール、更新版のインストール、リソース割り当ての変更などです。
- 実行では、適応フェーズで計画されたアクションが、監視対象サーバ上で実現されます。

特に、「分析」と「適応」のフェーズでは、知識とルールが必要になります。

- 知識とは、コンポーネント間の相互関係の理解、各種設定の重要性と影響についての知識を意味します。また、それぞれのアクションが与える影響や、起こりうる副作用についても理解している必要があります。
- ルールとは、仕様書に直接記述されている指示、または、知識に基づいてサービス内容合意書から推測される指示、を意味します。
管理の原則：エージェントと管理ステーション

つまり、管理サイクルには、エージェントと管理ステーション間における２つの通信インタフェースが含まれることになります。すなわち、図２の左側に示されている監視と分析の間のインタフェースと、右側に示されている計画と実行の間のインタフェースです。この２つのインタフェースについては、次で詳しく説明します。

原則的に、このサイクルは連続的に実行されます。この場合、次の２つのポイントが「推進力」になります。

- 分析（内部的な推進力）
  システムの継続的な運用中に、適応を必要とする状況が分析フェーズで検出された場合、新しいサイクルが生成されます。

- 計画（外部的な推進力）
  管理者が、追加サーバのインストールや、サービス内容合意書の改訂に基づくリソース割り当ての変更などを決定した場合、計画のフェーズで新しいサイクルの実行が開始されます。

これにより、これらのフェーズを、「エージェントと管理ステーション」の管理原則に簡単に当てはめることができます。

- エージェント
  エージェントは主に、「監視」と「実行」のフェーズで使用されます。状況によっては、しきい値の監視、トラップの送信、ServerView エージェントの PDA（Prefailure Detection Analysis）機能など、「分析」のフェーズにかかわることもあります。
  ただし、OS の再設定、ハードウェアコンポーネントの手動交換といった計画の実行には、他のツールや動作が使用されることもあります。

- 管理ステーション
  管理者や管理者の知識は、分析と適応の段階において不可欠です。管理者は、管理ステーションを使用して、分析に必要な情報をエージェントから取得し、エージェントを使用して、適応フェーズで生成された計画をシステム上で実現します。日常的な管理タスクは、段階的に自動化されます。自動化されると、管理者が介入しなくても、分析と適応のフェーズが管理ステーション上で完全に実行されます。この場合、ServerView Suite では、ServerView Event Manager を設定しておくことにより、特定のトラップの受信時にスクリプトを自動的に実行できます。
管理の原則：エージェントと管理ステーション

2.2.2 監視と分析の間のインタフェース

このインタフェースでは、エージェントは管理ステーションと連携して、監視フェーズから分析フェーズに情報を転送します。次に説明するように、これには永続的な通信が含まれます。特に、ポーリングの場合です。

ポーリング

SNMP または CIM などの管理プロトコルは、主にポーリングモードで使用されます。このモードにおいて主導権を持つのは、エージェントに対して情報を要求する管理ステーションです。通常の環境では、エージェントは管理ステーションに対して、要求された情報を配信します。さらに、分析フェーズで、計画のフェーズを必要とする状況が存在するかどうかが決定されます。存在する場合は管理サイクルが引き起こされ、計画フェーズが開始されます。

これに関係なく、プロセスは繰り返し実行されます。つまり、管理ステーションは一定の間隔で、エージェントに対して最新情報を定期的に要求し、分析します。ポーリングの間隔が短すぎると、ネットワークの負荷が大きくならず、エージェントが消費するサーバ上のリソース、例えば CPU 利用率が増加します。とはいえ、ポーリングの間隔が長すぎると、最新の情報が得られません。情報がどの程度最新である必要があるかは、関連する管理情報、情報がどれくらいの速さで変化する可能性があるか、および応答をどれくらい急ぐ必要があるかによって異なります。例えば、3日以内にドライブが故障する可能性がある PDA 情報は、ミリ秒単位でポーリングする必要はありません。そのため、ServerView では、より妥当なポーリング間隔が自動的に設定されます。また、ポーリングによる負荷をできるだけ削減するために、他のインテリジェントな技術（キャッシュなど）も使用されます。その他、ServerView のコンソールにはアップデートボタンが提供されているので、これを使ってポーリングを手動で開始することもできます。

トラップ

温度の超過など、簡単に検出できる特定のイベントは、エージェントによって直接、検出されます。この場合、エージェントは管理ステーションに対して、トラップと呼ばれる自発的なメッセージを送信します。これは、永続的な通信ではありません。その後、分析フェーズで、トラップにより報告されたイベントについて、計画フェーズが必要かどうかが決定されます。
管理の原則 : エージェントと管理ステーション

2.2.3 適応と実行の間のインタフェース

詳細に考えると、このインタフェースは単なる情報転送にはとどまりません。アクションが引き起こされた後、そのアクションの状態、またはエラーが正常に解決されたかどうかを、計画フェーズに報告する必要があります。アクションが正常に実行されなかった場合には、再実行する前に、新しい適応フェーズでプランを修正する必要があるかもしれません。

次の例で、このインタフェースでの管理ステーションとエージェント間の連携を説明します。

例 1
サーバのLAN接続についてパフォーマンスの問題を検出した管理者が、分析を行った結果、TCP/IPスタックの設定パラメータを変更する必要があると判断しました。管理者は、自身の管理コンソールから、管理アプリケーションを使用して対応するジョブをエージェントに送信することにより、この計画を実現できます。エージェントは、アクションが正常に完了したことを管理者に通知します。

例 2
サーバが「ハング」状態、すなわち、OS上で実行しているエージェントからの応答が停止しています。管理者は、このサーバを再起動する必要があります。この計画は、管理者のServerViewコンソールから実行できます。例えば、OSの状態に関係なくサーバ上のエージェントとして機能するRemoteView Service Boardを使用して、このアクションを引き起こします。

例 3
新規のブレードサーバをインストールすることになりました。個々のサーバブレードをインストールまたはクローニングするには、ServerView SuiteのDeployment Managerを使用します。この場合、初期状態のサーバブレード上にはエージェントは存在しません。Deployment Managerが、PXE（25ページの「プロトコル」の項をご覧ください）を使用して、サーバブレードにエージェントを自動的に追加します。これらのエージェントがDeployment Managerのコンポーネントや他のヘルパーと連携し、インストールまたはクローニングの処理を実行します。
管理の原則：エージェントと管理ステーション

2.2.4 サーバ上での自律管理

ここまでは、管理サイクルのフェーズについて説明してきました。もう1つ重要な考慮事項は、これらのフェーズをどこで実行するか、ということです。原則的には、管理サイクルの設計は、できる限りローカルでなければなりません。例えば、管理者が「分析」と「適応」フェーズにかかわっているとします。管理者は、発生した問題を自力で解決することを試みます。管理者が他の管理者と連携するのは、特定の問題にとって、その必要性があると判断した場合だけです。つまり管理者は、できる限りローカルな状態で、管理サイクルを維持するように努めます。

管理者が介入することなく、管理サイクルが完全に自動的に実行されることを、自律サイクルと呼んでいます。

注意：自律サイクルという用語は、植物性神経系である「自律神経システム」という用語に由来しています。このシステムは、無意識のうちに、人間を新しい状態に持続的に適応させます。例えば、運動をすると脈拍や呼吸数が自動的に増えるのは、自律神経システムによるものです。

自律サイクルもまた、できる限りローカルで実現する必要があります。たとえば、自律サイクルが完全に監視対象サーバ上で実行される場合、自律サイクルは他のリモートコンポーネントや機能しているネットワークに依存しないという長所があります。次に2つの例を示します。

– Automatic Server Reconfiguration & Restart （ASR&R）

PRIMERGY サーバにより、BIOS レベルで完全な自律サイクルが実現されています。RAM モジュールの故障などのコンポーネントの欠陥や、マルチプロセッサシステムのプロセス障害などによって、システム障害が発生した場合、異常のあるコンポーネントはマスキングされます。さらに、サーバの再起動が試行されます。システムをシャットダウンするまでの再起動の試行数など、自律サイクルを制御するパラメータは、管理者が ServerView コンソール上で設定できます。

– ローカルの ServerView Event Manager

ServerView Event Manager は、トラップを受信して、実行する処理を決定します。ここで決定される処理の候補には、1 人または複数の管理者に E メールまたは SMS で特定のイベントを通知する、ログファイルのエントリを作成する、より高レベルの ServerView Event Manager などの他のサービスにトラップを送信する、といったアクションが含まれます。また、スクリプトを自動的に実行して、特定の状況において必要なアクションを引き起こすこともできます。
階層的な設定

ServerView Event Manager は、監視対象サーバ上にインストールし、リモートまたはローカルで運用できます。スクリプトの自動実行とトラップ受信を組み合わせることにより、監視対象サーバ上で、完全な自律サイクルを設定できます。

2.3 階層的な設定

この項では、管理コンソール、管理アプリケーション、エージェント、ヘルパーを使用して設定できるトポロジーについて説明します。

ServerView Suite で実現できる、いくつかの典型的な例を紹介します。それぞれの特性、利点、欠点について説明します。

2.3.1 ローカル管理

エージェント、管理アプリケーション、ヘルパー、管理コンソールのすべてのコンポーネントを単一サーバにインストールした場合、ローカル管理の設定になります。ServerView Suite では、これは理論的には可能ですが、単一サーバのローカル監視と管理のために ServerView Suite を使用するといった特殊な状況でなければ、この設定は意味がありません。

2.3.2 ポイント・ツー・ポイント管理

ポイント・ツー・ポイント管理では、各監視対象サーバ上に、エージェントと管理アプリケーションの両方をインストールします。コンソールは、物理的に分離します。複数のコンソールを使用することもできます。ポイント・ツー・ポイント管理の特徴は、コンソールから直接管理できるのが個々のサーバに限定されることです。したがって、サーバのリストを表示したり、サーバのグループを作成したりすることはできません。
階層的な設定

図 3: ポイントツーポイント管理

この設定の利点は、セキュリティが強化され、可用性が向上することです。

－ セキュリティの強化

エージェントと管理アプリケーション間で頻繁に使用される SNMP プロトコルは、使用状況によってはセキュリティ機構が十分であるとは言えません。監視目的であれば問題はありませんが、SNMP 経由でアクションが引き起こされるような場合には、十分なセキュリティレベルが得られません。この設定では、SNMP メッセージはネットワーク経由では送信されないので、SNMP を使用してもセキュリティ上のリスクは生じません。コンソールと管理アプリケーション間の通信には、SSL を使用した安全なプロトコルが使用されます。
階層的な設定

– 可用性の向上

ポイントツーポイント管理ではなく、中央のコンピュータ上でマネージャを実行する場合、そのコンピュータが使用可能でなければサーバ管理を実行できません。したがって、サーバ管理の複製が必要になります。

ポイントツーポイント設定の場合には、監視対象サーバ上でマネージャを実行するので、中央コンピュータに依存しません。

ServerView Suite のコンポーネントを使用したポイントツーポイント管理の典型的な設定例:

– 監視対象サーバ上でエージェントと ServerView Operations Manager を実行します。

  この場合、任意の Web ブラウザからコンピュータを管理できます。コンソールとマネージャ間の通信は、SSL (HTTPS) を使用して保護できます。

– HTTPS を使用して、integrated Remote Management Controller（iRMC）または管理ブレードにアクセスすることもできます。これらのコンポーネントでは、情報が提供されるだけでなく、Web ページとしてグラフィカルに表示されます。つまり、エージェントとマネージャの機能が同時に実行されます（42 ページ の「アウトオブバンド」管理モード」の項をご覧ください）。

– 監視対象サーバ上でエージェントと Event Manager を実行します。この場合、トラップはネットワーク経由では送信されないので、トラップが失われることはありません。ServerView Event Manager は、監視対象サーバから管理者に直接問題を通知するか、スクリプトを自動的に実行します。
階層的な設定

2.3.3 集中管理

この設定では、分離された中央コンピュータ上で管理アプリケーションを実行します。したがって、ポイントツーポイント管理で説明した利点は得られません。

この設定の利点は、総合的な概要を表示し、グループを作成し、包括的な自動ソリューションを提供できることです。

- 総合的な概要
  マネージャは複数のサーバからの情報にアクセスできるので、サーバリストなど、総合的な概要を表示できます。また、各サーバの全般的な状態も表示されます。

- グループの作成
  管理者は、総合的なサーバリストを表示し、複数のサーバを組み合わせて、サーバグループを作成することができます。グループに対して処理を適用すれば、各サーバに個別に適用する必要がありません。これにより、作業効率が著しく向上します。

- 包括的な自動ソリューション
  複数サーバの概要を必要とする包括的な自動ソリューションは、集中管理設定でのみ実現できます。
図 4: 集中管理

エンタープライズサーバのすべてのハードウェアコンポーネントを監視、設定できるのは、この方法だけです。
2.3.4 カスケード管理

同時にエージェントとしても機能する管理アプリケーションは、カスケード設定することができます。この場合、図5に示すようなトポロジになります。カスケード設定は、情報が下位レベルから上位レベルに集約されるように構成します。

ServerView Suite の場合、管理アプリケーションとエージェントの両方の機能を実行できるのは、ServerView Event Manager です。管理アプリケーション機能としては、Event Manager にトラップを送信するように設定されたすべてのエージェントから、SNMP トラップを受信します。ただし、ServerView Event Manager は、特定の状況では自身にトラップを送信するようにも設定できるので、エージェントとしての動作も実行します。例えば、受信したトラップをフィルタリングして、特定のトラップだけを転送できます。

図5: カスケード管理
階層的な設定

管理コンソール上で複数のServerView Event Managerを段階的に設定することにより、どのEvent Managerから、どのトラップを次の上位レベルに送信するかを指定できます。

多数のサーバーを管理する必要がある場合は、ServerView Event Managerをカスケード設定する方法が適しています。セキュリティを考える必要がある場合は、冗長パスを備えた階層を設定することもできます。

2.3.5 他の管理システムへの統合

ServerView Suiteは、他の管理システム（CA Unicenter、IBM Tivoli、BMC Patrolなど）に統合することもできます。統合するための特別なServerView製品があります。

図6に、サーバー管理コンポーネントを、より高レベルのエンタープライズ管理システムに統合する場合の概要図を示します。

![図6: 他の管理システムへの統合]
階層的な設定

統合ソリューションには、トラップ統合、ステータス統合、コール統合の3つの統合タイプがあります。

- トラップ統合

トラップ統合には、2つのオプションがあります。1つは、エージェントからエンタープライズ管理システムに直接トラップを送信する設定です。もう1つは、ServerView Event Managerのカスケード設定と同様です。エージェントからServerView Event Managerにトラップを送信します。Event Managerは、設定に応じて、すべてのトラップまたはフィルタリングしたトラップをエンタープライズ管理システムに転送します。

同じ種類の統合が、SCOM(System Center Operations Manager)などのエンタープライズ管理システムで構成され、オペレーティングシステムで管理されるイベントログファイルのエントリを評価します。ServerView Event Managerは、SNMPトラップをイベントとしてログファイルに入力するので、ある意味、間接的なトラップ統合と言えるかもしれません。

- ステータス統合

エンタープライズ管理システムにServerViewエージェントを統合することにより、ネットワーク上の監視対象サーバの存在を確立(検出)することができます。また、エンタープライズ管理システムに対して、サーバの全般的な状態を提供できます。この情報に基づいて、アイコンでサーバの種類が区別され、アイコンの色でサーバの全般的な状態がわかるトポロジビューなどに、監視対象サーバを表示できます。

- コール統合

前述したように、エンタープライズ管理システムでは、対応するアイコンを使用して、管理対象のPRIMERGYまたはPRIMEPOWERサーバを表示できます。これらのアイコンの1つをダブルクリックすると、管理コンソールが自動的に開き、管理者はServerView管理アプリケーション経由で、そのアイコンが示している監視対象サーバにアクセスできます。
プロトコル

プロトコルは、OS に依存しません。プロトコルを使用して通信することにより、異なる OS を使う複数のコンピュータを、問題なく連携させることができます。これは、管理タスクにとって極めて重要です。

ServerView Suite は、すべての通信バス上で、標準プロトコルを使用していきます。各通信バスには、それぞれに対応する、グローバルに受け入れられ使用されている標準規格やデファクトスタンダードが存在します。

2.4.1 SNMP - Simple Network Management Protocol

SNMP（Simple Network Management Protocol）は 1990 年代の初めに登場し、簡単で安定性に優れていることから、すぐに普及しました。しかし、後に SNMPv1 と呼ばれることになるこの SNMP には、データ転送に対する保護機構がありません。SNMPv2（1994）と SNMPv3（1998）では、適切なセキュリティ機構が統合されました。そのため、このプロトコルの実現がより複雑になり、対応するコンポーネントの使用、設定にかかることも増加したため、これらの 2 つのバージョンはさほど普及しなませんでした。
プロトコル

図 7: SNMP（Simple Network Management Protocol）

SNMP は、もともとネットワーク管理用として設計されたものです。しかししながら、プロトコルの観点からすれば、ネットワークに限らず、システム、デバイス、アプリケーションの管理も基本的に同じであることが、すぐにあきらかになりました。図 7 に、SNMP の基本概念を示します。

SNMP プロトコルと管理オブジェクトは、最初から分離されていました。SNMP プロトコルは処理を指定しますが、管理対象オブジェクトは、管理情報ベース（Management Information Base、MIB）で別途定義されています。これらのオブジェクトは、グローバルに定義されたツリー構造に基づくネーミングシステムによって一意に識別され、このツリー構造によって順序付けられています。
基本的に、SNMP プロトコルは、サーバ（より一般的にはコンポーネント）の管理について、4 つの処理を提供します。

- 管理ステーションから、情報を要求できます。
  これには、2 つの異なる操作があります。指定したオブジェクトの値を要求するには、Get 操作を使用します。オブジェクト名の順序に従って、次のオブジェクトの値を要求するには、Getnext 操作を使用します。
  SNMPv3 は、1 回の操作で指定した数量のオブジェクトを読み込む、Getbulk 操作もサポートしています。
- 管理ステーションから、値を変更できます（Set 操作）。例えば、設定を変更したり、処理を間接的に引き起こしたりできます。
- エージェントは、要求された情報を提供するか、または値の設定を確認します（Response 操作）。
- エージェントそのものをアクティブにして、特定のイベントを管理ステーションに通知できます（トラップ、通知）。例えば、しきい値の超過、サブシステムの状態の変移、PDA メッセージなどです。

SNMPv1 では、データは暗号化されずに転送されますが、コミュニティの概念によってアクセス権限を設定できます。コミュニティは、SNMP を使用して通信するエレメント（管理ステーションとエージェント）のグループです。グループは、コミュニティストリングによって個別に識別されます。同じコミュニティに属しているシステムだけが、相互に通信できます。システムは、複数のコミュニティに属すことができます。

管理ステーションとエージェント間の通信では、コミュニティストリングがパスワードの役割を果たします。エージェントは、そのエージェントのシステムに関する情報を提供する前に、管理ステーションに対してコミュニティストリングを要求します。これは、各 SNMP パッケージに適用されます。
各オブジェクトに適用されるアクセスの種類（例：読み取り専用または読み書き）は、MIB に指定されています。エージェント情報を管理ステーションのアクセス権限も、コミュニティストリングにリンクされています。アクセス権限とコミュニティストリングの関連付けにより、MIB アクセスの種類をさらに制限できます。権限の拡張は許可されません。MIB 定義によってオブジェクトが読み取り専用に指定されていれば、コミュニティストリングが読み書きアクセス権限に関連付けられていたとしても、そのオブジェクトへの読み書きアクセスは許可されません。次に、コミュニティストリングとアクセス権限の使用例を示します。

例：
SNMP エージェントが、読み取り専用アクセス権限が指定された、public という名前のコミュニティに属しています。public コミュニティに属している管理ステーションは、コミュニティストリング public を使用して対応するメッセージを送信することにより、この SNMP エージェントに情報を要求できます。この SNMP エージェントはまた、読み書きアクセス権限が指定された、net_5 という名前の 2 つのコミュニティにも属しているとします。net_5 コミュニティに属しているのは、別の管理ステーションです。この例の場合、SNMP エージェントを使用して書き込み操作を実行できるのは、net_5 コミュニティの管理者だけです。

ネットワークコンポーネントに特定のイベントが発生した場合、SNMP エージェントは、トラップを送信することによって、1 つ以上の管理ステーションにこのイベントを通知することができます。管理ステーションによる SNMP トラップの受信にも、コミュニティストリングが必要になります。SNMP エージェントは、管理ステーションに対してトラップメッセージを送信する場合、管理ステーションでそのメッセージが許可されるように、トラップ用コミュニティストリングを使用する必要があります。

前述したように、管理対象オブジェクトは、SNMP 標準には定義されていません。オブジェクトの意味、構文、値の範囲は、MIB ファイルによって別途定義されています。これらは、名前と値で構成された単純なオブジェクトです。例えば、SNMP の場合、数千個の標準オブジェクト定義があり、さらに製造元独自のオブジェクト定義もあります。ただし、管理ステーションのユーザフレンドリーなインターフェースは、管理情報が明確に表示されるように、これらのオブジェクトの意味を考慮する必要があります。そうでない場合、MIB ブラウザにテーブルは表示されますが、情報の意味はほとんど不明確になります。

ServerView で認識される MIB は、標準 MIB2 と、すべての ServerView 独自の MIB です。ServerView では、これらのオブジェクトの意味を認識して表示するか、しきい値の超過などの場合には、これらの意味に応じてオブジェクトを解釈します。
MIB と SNMP について、詳しくは次の URL をご覧ください。
http://www.simpleweb.org/ietf/

次のような技術書にも、詳細情報が掲載されています。
「The Simple Book」 Marshall T. Rose 著

2.4.2 CIM （Common Information Model）

SNMP が標準プロトコルとなってから数年後、CIM の標準化が開始されました。SNMP ではプロトコルが最初に標準化されてから MIB が徐々に定義されましたが、CIM では別の方法が採用されました。最初にオブジェクトの定義から開始されたため、これが「共通情報モデル (CIM)」という名前の由来となり、それから転送プロトコルの仕様が確定されました。最初の仕様はすでに確定されて、使用されていますが、これらの標準化プロセスは実際にはまだ完了していません。

まず、モデル化の方法について説明します。CIM と、SNM 内の簡単なオブジェクト定義との本質的な違いは、CIM にはクラス階層が含まれ、新しいオブジェクトは階層の中でのみ定義できることです。クラス A から派生したクラス B は、クラス A のすべてのプロパティを継承します。つまり、クラス B は固有のプロパティとクラス A のプロパティを持ちます。そのため、クラス階層には「継承ツリー」を定義します。
プロトコル

図 8: CIM（Common Information Model）

CIM は、DMTF によって開発され、次の主要コンポーネントで構成されています。

- メタスキーマ
  DMTF に基づくこの仕様は、オブジェクトを指定するのではなく、スキーマの定義ルールを定義します。

- コアスキーマ
  このスキーマでは、DMTF が継承階層のトップレベルのクラスを定義します。すべての CIM クラスは、これらのトップクラスから直接派生するか、またはいくつかの階層レベルを介して間接的に派生している必要があります。「依存」などの関係もまた、クラスとして定義され、派生スキーマ内で調整されます。各種の管理データ内における複雑な相互関係を表すには、関係の表示が重要になります。
プロトコル

- 共通スキーマ

共通スキーマは、コアスキーマのクラスから、特定の領域のクラスを派生させます。これらのクラスも、DMTFによって定義されます。現在、共通スキーマがすでに採用されている領域、またはDMTFワーキンググループが現在かかわっている領域には、システムとデバイス、ネットワーク、ユーザとセキュリティ、動作と状態、データベース、アプリケーション/メトリック、ポリシー、プレOS、ユーティリティコンピューティングがあります。これらのクラス定義はすべて、製造元に依存せず、DMTFによって標準化されています。

- 拡張スキーマ

拡張スキーマでは、製造元が独自のクラスを定義できます。ただし、これらのクラスは共通スキーマクラスから派生している必要があり、例外的な場合には、コアスキーマクラスからのみ派生させることができます。SNMPのMIB定義と比較すると、未知の製造元独自クラスであっても、継承ツリーに基づいて、ある程度までは解釈できるという利点があります。したがって、CIMでは、製造元独自MIBクラスの「未制御の拡張」が防止されています。

製造元独自スキーマには、MicrosoftのWin32スキーマとServerViewSuite用のFSCSVスキーマがあり、いずれも「システムとデバイス」の共通スキーマから派生されています。

- MOF（Managed Object Format）

すべてのスキーマは、必要に応じて概略を定義できます。できるだけ、統一モデルリング言語（Unified Modeling Language、UML）で定義します。自動的に解釈され、交換できるように、MOFと呼ばれる構文が定義されています。したがって、スキーマの具体的な仕様は、MOF構文を使用したテキストファイルになります。

CIMについて、詳しくはDMTFのCIMページをご覧ください。
http://www.dmtf.org/standards/cim

CIMオブジェクトに、製造元に依存せずにリモートアクセスするために、CIM-XMLというメタスキーマベースのコーディングが開発されました。「メタスキーマベース」とは、CIMオブジェクトに加え、関連するCIMクラス定義の階層が要求され、転送されることを意味しています。このプロトコルがほとんど使用されていないのは、こうした複雑さが原因であると考えられています。現在、標準化団体により、CIMアクセス用のより簡素化されたプロトコルの開発が検討されています。
2.4.3 IPMI （Intelligent Platform Management Interface）

IPMI 標準は、ハードウェアまたはファームウェアレベルで実装される管理機能を定義します。これにより、メインプロセッサ、BIOS、OS の状態に関係なく、すべての機能を使用できるようになります。これらの機能は、システムがシャットダウンされている状態でも使用できます。

図 9: IPMI （Intelligent Platform Management Interface）

IPMI は、管理ソフトウェアがスタックを設定できるインタフェースを提供します。図 9 に、IPMI インタフェースと、その上位スタックでのいくつかの標準の構成方法の例を示します。

IPMI の実装は、ベースボード管理コントローラ（Baseboard Management Controller、BMC）に基づいています。BMC は、独自のプロセッサとメモリを備えた、コンピュータに依存しないマイクロコントローラです。このコンントローラが、IPMI インタフェース経由で提供されたすべての機能を実行します。IPMI は継続的に開発されており、管理機能の範囲が拡張されています。

ServerView Suite のコンポーネントは、インバンド管理（OS と管理エージェントがシステム上で実行されている）と、アウトオブバンド管理（システム上で OS が実行されていないか、システムが故障している）の両方に、IPMI インタフェースを使用します。

IPMI について、詳しくは次の URL をご覧ください。
http://developer.intel.com/design/servers/ipmi/index.htm
IPMI V1.0（1999）:

IPMI V1.0の主要タス克はモニタリングですが、IPMIはこれを直接コマンドで実行するではなく、センサーデータレコード（Sensor Data Records、SDR）を含むモデルを介して実行します。SDRは、ポーリング可能なシステムに存在するセンサーの記録です。これにより、IPMIは各種システムに柔軟に適用することができます。また、BMCの永続メモリには、イベントメッセージが書き込まれるシステムイベントログ（SEL）があります。このシステムイベントログには、IPMIインタフェースからアクセスできます。そのため、システムがシャットダウンしていても、これらのイベントメッセージを評価できます。IPMIでは、システムコンポーネントを個別に識別でき、IPMIインタフェースから識別情報を要求できます。これにより、FRU（Field Replaceable Unit）情報の概念に基づいています。この情報の構造はIPMI仕様に定義されており、製造元独自ベース上で拡張できます。Fujitsuでは、PRIMERGYコンポーネント用の拡張情報を定義しています。FRU情報には、製造元、モデル、シリアル番号などのデータが含まれ、マザーボード、メモリボード、またはコントローラ上にあるSEEPROMに保存できます。

IPMI V1.5（2001）:

IPMI V1.5で使用できるようになった重要な機能は、「IPMI over LAN」です。V1.0では、IPMIインタフェースを（I/Oマップされた）ローカルインタフェースとしてしか使用できませんでしたが、V1.5には、IPMIメッセージをRMCP（Remote Management Control Protocol）メッセージ内に組み込んで、UDPパッケージとしてネットワーク経由で送信する方法が定義されています。SNMPトラップを使用して、発生したイベントの情報を提供することもできます。このLAN通信は、システムのLANコントローラ上で実行できます。ただし、システムがシャットダウンしても管理ポートが動作するように、管理ポートがスタンバイ状態で動作するように設定する必要があります。IPMI専用のLANコントローラを使用することもできます。

IPMI V2.0（2004）:

V2.0で導入された新機能には、VLAN（Virtual LAN）があります。VLANサポートにより、他のVLANから分離している管理VLANを作成し、管理VLANに設定されたデバイスだけが相互に通信できるように設定できます。また、暗号化、ログイン、ファイアウォールの機能を使用して、セキュリティを強化できます。
プロトコル

2.4.4 PXE（Preboot Execution Environment）

WfM（Wired for Management）戦略の一環として、Intelは、WfMの要件に適合させるためにシステムに実装する必要があるデスクトップ、モバイル、サーバ管理用の機能と標準規格を指定しました。これには、SNMPと、特定のMIBオブジェクトまたはIPMIのサポートが含まれます。また、まだ実現されていなかった一部の標準規格について、開発の必要性が指摘されました。その1つが、PXEプロトコル（プリブート実行環境）でした。

PXEの開発目的の1つは、次の問題を解決することでした。ネットワークに加入する新規システムには、特定のサーバから設定パラメータとイメージを自動的にロードできるプロトコルが提供されている必要があること、このプロトコルを実行することでOSなどをインストールできること、などです。

この目的を達成するために、PXEプロトコルは、TCP/IP、DHCP、TFTPインターネットプロトコルに基づいて定義されました。基本的には、PXEプロトコルは次のように動作します。

- 新規システムがネットワークに対してブロードキャストを送信する。
- ネットワーク上にDHCPサーバがあり、新規システムにIPアドレスが割り当てられていれば、DHCPサーバから応答として適切なブートサーバのリストが送信される。
- 新規システムは、通知されたブートサーバの1つからTFTP経由で実行ファイルをロードし、実行する。
- システムは、より詳細な管理機能を実行または開始できる状態になる。

ServerView Suiteでは、PXEはDeployment Managerによって使用されます。

PXEとWfMについて、詳しくは次のURLをご覧ください。
http://www.intel.com/technology/computing/wfm.htm
2.4.5 Telnet

Telnet は、インターネット経由でリモートコンピュータにログインするために開発されました。Telnet は、Network Virtual Terminal (NVT) の概念に基づいています。Telnet 経由で通信する 2 つのデバイスは、それぞれのローカルデバイスのプロパティを、仮想端末のプロパティにマップします。

また、Telnet プロトコルには、セッションの開始時に両デバイス間でネゴシエートできる多数のオプションが定義されています。これにより、一定の範囲までは、最低レベルではなく最小公分母レベルによって通信することができます。

IETF によって RFC 854 に Telnet プロトコルが定義されたのは、1983 年のことです。以来、主にセキュリティに関連する機能が追加されてきました。例えば、現在の Telnet 通信は、SSL によって保護できます。

ServerView Suite では、Telnet プロトコルは、アウトオブバンド管理のコンポーネントによって使用されます。

Telnet について、詳しくは次の URL をご覧ください。
http://www.ietf.org/rfc/rfc0854.txt

2.4.6 HTTP (Hypertext Transfer Protocol）

HTTP は、Worldwide Web 上で 1990 年から使用されています。HTTP は、通常は TCP/IP 接続上で実行される要求/応答プロトコルですが、他の信頼できる転送サービスによって処理することもできます。デフォルトのポートは、TCP 80 です。

HTTP は、ファイルに他のファイルへの参照を含めることができ、これらの参照の 1 つを選択することによって対応するファイルを要求できるという概念に基づいています。

HTTP クライアント（通常は Web ブラウザ）は、HTTP サーバの IP アドレスに、希望する情報のパス名を指定した要求を送信します。HTTP サービスは、HTTP サーバ上で実行され、HTTP 要求を待ちます。要求を受信すると、指定されたファイル（通常は複数のファイルで構成されている Web ページ）、またはエラーメッセージを戻します。

HTTP サービスによって配信できるのは、スタティックな情報だけではありません。HTTP 要求を受信すると、最新の情報が取得され、応答として返送されます。例えば、最新の株価の情報を得るときの状態がこれに該当します。
HTTP 経由でダイナミックなデータを要求できるので、HTTP プロトコルは、サーバ管理にも使用されています。ServerView では、ページを監視するデータは、常時、ダイナミックに取得されます。

2 つの典型的なアーキテクチャを次に示します。

- HTTP サービスを監視対象サーバ上で実行します。要求を受け取ると、HTTP サービスは要求された情報をローカルインタフェースから取得し、適切に準備して、応答として返送します。この場合の HTTP サービスは、Web エージェントとも呼ばれられます。

- HTTP サービスを、監視対象サーバとは別のサーバ上で実行します。両サーバは、SNMP などの他のプロトコルを使用して通信します。このタイプの HTTP サービスは、要求を受け取ると、SNMP などを使用して監視対象サーバから要求された情報を取得し、適切に準備して、応答として返送します。また、複数の監視対象サーバからの情報をサーバリストまたはデータベースにまとめたり、キャッシュにバッファしておいて、応答として戻したりすることもできます。

基本的に ServerView では、管理コンソールとして Web ブラウザを使用する場合、常に HTTP が使用されます。

HTTP について、詳しくは次の URL をご覧ください。

2.4.7 SOAP （Simple Object Access Protocol）

SOAP は、Web サービスにリモートアクセスするためのプロトコルです。SOAP は、主として HTTP 転送プロトコルに基づく Remote Procedure Calls (RPC) の実行時に使用されます。RPC には、SOAP が HTTP と同様に、ファイアウォールを通過できるという利点があります。ただし、原則的に SOAP は、SMTP (Simple Mail Transfer Protocol) または MIME (Multipurpose Internet Mail Extension) などの他の転送プロトコルを使用できます。

SOAP メッセージは、XML ドキュメントです。つまり、コーディングは OS や転送プロトコルに依存しません。XML と同様に、SOAP の標準化は World Wide Web コンソーシアム (W3C) の管轄です。
SOAP はまた、Microsoft のサービス指向アーキテクチャ（Service Oriented Architecture、SOA）の中心的な役割を担っています。このアーキテクチャの基盤は、実装とインタフェースが完全に分離されていることです。実装は、サービスのユーザに通知することなく、適宜、変更することができます。サービスへのインタフェースは、基本的に SOAP インタフェースとして定義されています。この傾向はアプリケーションソフトウェアに限定されたものではありませんが、管理ソフトウェアにとっては重要な意味があります。

ServerView Suite では、SOAP は、Deployment Manager のコンポーネント間など、管理カテゴリのコンポーネント間の通信に使用されます。

SOAP について、詳しくは次の URL をご覧ください。
http://www.w3.org/TR/soap または
http://www.microsoft.com/mind/0100/soap/soap.asp

2.4.8 ITIL （IT Infrastructure Library）

ITIL は、イギリスのノリッジにある英国政府機関 CCTA（現 OGC）によって開発された IT Infrastructure Library ガイドラインの略語です。ITIL は現在、サービス管理における世界的なデファクト・スタンダードであり、IT サービスの計画、提供、サポートに関する総合的な技術文書が公開されています。ITIL には、運用 IT インフラストラクチャの使用と効率を向上させるための基盤が提供されています。

ITIL は、IT インフラストラクチャの使用と運用を支援、改善するために、コンピューターセンタのスタッフとコンサルタント会社の専門家によって開発されました。ITIL は、IT サービス管理における 2 つの主要点に着目しています。1 つは IT サービスのライフサイクル、もう 1 つはカスタマビューです。

ITIL の書籍は、サービス管理に関するベストプラクティスとは「何か」を示すガイドラインであり、具体的な「方法」が示されているわけではありません。具体的な方法は、それぞれの企業の規模、社内文化、そして必要に応じて調整し、それに応じて実現する必要があります。つまり、関連する IT 環境と提供するサービスに応じて、コンピューターセンタのプロセスを具体的に定義する必要があります。基本的には、ISO9000 に準じた品質管理と似たようなものです。

ITIL では、これらのプロセスを、次のモジュールによって提示しています。インシデント管理、問題管理、変更管理、設定管理、リリース管理、サービスレベル管理、コスト管理、キャパシティ管理、可用性管理、緊急時対策です。
ヘルパー

ServerView Suite のコンポーネントは、サーバの管理をライフサイクル全体においてサポートしています。サーバ管理は、サーバの状態に関係なく実行でき、資産情報の作成、アーカイブまたはレポートなどの機能によって、これらのプロセスに関する重要な情報が提供されます。そのため、ServerView Suites は、ITIL に準拠した現実的な管理プロセスを実装する場合に最適です。

ITIL について、詳しくは http://www.itil.co.uk をご覧ください。

2.5 ヘルパー

ヘルパーは ServerView Suite のコンポーネントではありませんが、ServerView の運用中に ServerView で使用できるサービスまたはプログラムです。

2.5.1 DHCP サーバ

DHCP（Dynamic Host Configuration Protocol）は、TCP/IP ネットワーク上でホストを起動して TCP/IP ソフトウェアを設定する場合、中央ポイント (DHCP サーバ) からホストに IP アドレスを割り当てるために使用します。ホストは、DHCP 要求を使用してブロードキャストにより IP アドレスを要求し、これにより自身の MAC アドレスを使用して自己を識別できます。MAC アドレスは、ネットワーク上のすべてのデバイスを固有に識別するハードウェアアドレスです。

DHCP サーバは、このような要求を受信すると、DHCP サーバの設定に従って使用可能な IP アドレスを戻します。DHCP サーバの設定により、IP アドレスをスタティックに割り当てるか、ダイナミックに割り当てるかが決まります。スタティックの場合、DHCP サーバは、MAC アドレス用に予約された固定 IP アドレスがテーブルに割り当てられるように設定されています。要求への応答では、関連 MAC アドレスに割り当てられた IP アドレスが返信されます。ダイナミックの場合、使用可能な IP アドレスのプールから、IP アドレスがダイナミックに割り当てられます。不要となった IP アドレスは、このプールに戻されます。スタティックとダイナミックの両方の割り当てを混在させることもできます。

DHCP サーバは、ServerView Suite の重要なヘルパーです。また、ServerView は、次に説明する PXE ブートサーバとともに DHCP サーバを使用します。
2.5.2 PXE ブートサーバ

PXE は LAN コントローラのブートモードです。PXE を使用してブートされるコンピュータの BIOS は、ブート処理中に、ブートデバイスとして LAN コントローラをアクティブにする必要があります。つまり、PXE LAN Boot Extension を起動します。

LAN コントローラのブート機能の他に、PXE ブート処理では、LAN 上に PXE サーバと DHCP サーバが必要になります。

PXE ブート処理は、次のように実行されます。PXE Lan Boot Extension が、DHCP 要求を LAN にブロードキャストで送信します。DHCP サーバの応答により、要求された IP アドレスと追加パラメータが戻されます。ブートサーバにも IP アドレスを割り当てることができます。ブートサーバの IP アドレスを転送する場合には、このサーバだけに直接送信されるように設定しておく必要があります。そうでない場合、以降の要求がブロードキャストとして送信されるからです。この要求は、ブートサーバに対して、サーバに割り当てられたブートイメージの名前を問い合わせるものです。ブートサーバは、事前設定に基づいてブートイメージを決定します。PXE LAN Boot Extension は、このイメージをファイル転送（TFTP または MTFTP）によって取得し、アドレス 07c0h にコピーして、起動します。

PXE ブートサーバは、ServerView Suite で次のように使用されます。

- ServerView Installation Manager のリモート使用。この場合、Installation Manager OS ディレクトリが PXE ブートサーバに作成され、PXE 経由でインストールされるサーバにコピーされて起動されます。

- Deployment Manager でのクローニング処理。この場合、PXE ブートが通常 2 回実行されます。最初にブートサーバから設定エージェントがコピーされて起動され、RAID システムなどをセットアップします。2 回目の PXE ブートで、RAID システムに実際のイメージがコピーされます。

詳細については、『FUJITSU Software ServerView Suite, Deployment Manager』マニュアルを参照してください。

2.5.3 TFTP サーバ

TFTP (Trivial File Transfer Protocol) は、完全なファイルの読み取りまたは書き込みだけでなく使用される、極めて単純なプロトコルです。より高性能の FTP (File Transfer Protocol) と混同しないようにしてください。多数のネットワークコンポーネントが、ネットワーク経由で基本 OS をロードしたり、初期設定を実行したりする場合、TFTP サーバを使用しています。
ServerView Suite は、Deployment Manager や Update Manager の特定のフェーズ、または RARP 連携など、サーバ上に有効な通信プロトコルスタックが設定されていない場合に、TFTP サーバを使用します。

### 2.5.4 メールサーバ（SMTP）

メールサーバ ServerView Suite でメールサーバを使用して、ServerView が特定の状況を検出した場合に管理者に報告します。例えば、特定のトラップを受信したときに管理者またはサービスプロバイダにメールが送信されるように、ServerView Event Manager を設定できます。

ServerView Event Manager は、次のメールサーバと連携させることができます。

- SMTP サーバ（Simple Mail Transfer Protocol）

  このプロトコルは、Internet Engineering Task Force （IETF）によって標準化されました。SMTP サーバは、Linux または Solaris 環境で、ヘルパーとして使用されています。

詳しくは、「ServerView Event Manager」のマニュアルをご覧ください。

### 2.5.5 Web サーバ

ServerView Suite の特定のコンポーネントは、Web ブラウザを使用して、管理情報にアクセスできます。ServerView Operations Manager を含むこれらのコンポーネントでは、Web サーバとの連携により、ServerView の情報を HTTP 経由で任意のブラウザで使用できます。

### 2.5.6 MS Excel、Access、または SQL データベース

Microsoft Excel/Access プログラムと SQL データベースも、ヘルパーに含まれます。インベントリ情報が含まれている ServerView Inventory Manager の生成ファイルを、MS Excel/Access または SQL データベースにインポートして、処理したり、他のデータにリンクしたりすることができます。
2.6 管理モード

前項では、サーバ管理コンポーネントをカテゴリに分類し、それぞれの相互関係と、関連する標準規格について説明しました。ここでは、管理モードについて説明します。モードは、次の要因によって異なります。

- 監視対象サーバの OS の状態。この OS は、標準運用に使用されるターゲット OS のことです。状態は、OS-up または OS-down です。

- 監視対象サーバのリモート管理に使用される通信パス。
  「インバンド」は、運用と同じパス（ネットワークコントローラ）が使用されることを意味します。
  「アウトオブバンド」は、管理用に異なるパスが使用されることを意味します。このパスは、「セカンダリ管理チャネル」とも呼ばれています。

OS の状態と通信パスによって、次で説明する管理モードが決まります。

2.6.1 「インバンド」管理モード

このモードの特徴は、監視対象サーバ上のターゲット OS が稼働し、管理用の通信パスが運用目的のパスと同じであることです。管理は通常、ターゲット OS 上で稼働しているエージェントを使用して実行されます。ポーリングとトラップの送信は、いずれも、この通信パス上で行われます。
管理モード

この管理モードは、実行中のサーバを監視するときに使用されます。ターゲット OS 上のエージェントと通信パスは、運用上の通信にも使用されるので、監視対象サーバを管理することにより、プロセッサ性能とネットワーク性能の一部が消費されます。しかし、ServerView コンポーネントの自動制御されたキャッシュイングと適切なポーリング間隔により、性能劣化を気にかける必要はありません。

監視対象サーバの状態が異なる場合（インストールされていない、ブートしていない、ブートしたがハングしているなど）、またはハードウェアエラーによりシステムを稼動状態にできない場合、サーバ管理に「インバンド」管理モードは使用できません。そのため、ServerView Suite には、次で説明する別の管理モードが提供されています。

「インバンド」管理モードは、一般的に、OS 独自の ServerView SNMP エージェントを使用して監視対象サーバを管理する場合に使用されます。

### 2.6.2 「アウトオブバンド」管理モード

この管理モードは、監視対象サーバの運用に使用されているハードウェア、ファームウェア、およびソフトウェアコンポーネントにまったく依存しません。この管理モードでは、個別の独立したハードウェアコンポーネント、通信パス、ソフトウェアが使用されます。したがって、このマニュアルではこのモードを「セカンダリ管理チャネル」と呼びます。

独立したハードウェアを使用して、SNMP エージェント、Telnet サービス、または Web エージェントを、SSL により保護していても実行できます。「インバンド」管理モードと異なり、管理情報は、ターゲット OS の機能から提供されません。管理情報は、IPMI と互換性のある integrated Remote Management Controller（iRMC）または別のログファイルにより、センサーから直接、提供されます。

ServerView Suite では、統合された Remote Management Controller（iRMC）で、IPMI-over-LAN を使用してステータスに関係なくサーバに直接アクセスできます。IPMI-over-LAN は、ネットワークセグメント内で使用する場合のみ意味をなす Remote Management Control Protocol（RMCP）に基づいているので、ServerView Suite では次のソリューションを提供します。

ServerView Remote Management や Deployment Manager などの ServerView アプリケーションは、IPMI-over-LAN 経由で iRMC と通信し、管理者に HTTP などのユーザインターフェース経由での通信を提供します。

しかしながら、「アウトオブバンド」管理モードで使用するこれらの独立コンポーネントは、単一の監視対象サーバ専用として割り当てる必要はありません。複数の監視対象サーバのセカンダリ管理チャネルとして使用でき、ある
程度の範囲であれば、コンセントレータとしても使用できます。例えば、管理ブレードは、ブレードサーバ内のすべてのサーバブレードにセカンダリ管理チャネルを提供するコンセントレータになります。

「アウトオブバンド」管理モードを使用する典型的な状況は、次のとおりです。

- サーバに、ターゲット OS がインストールされていない場合。サーバブレード（空のメタルブレード）をリモートでインストールする場合。
- サーバの電源をオフにし、このモードを使用してリモートから電源をオンにし、プートする場合。
- サーバ上の OS が稼働しているが「ハング」していて、運用に使用する通信パスではアクセスできない場合。このようなシステムは、セカンダリ管理チャネルを使用して監視、分析し、必要に応じてシャットダウンし、リプートできます。
- サーバのハードウェアに異常がある場合。このような場合、「アウトオブバンド」管理モードを使用して追加診断を行っておけば、問題解決に役立つ情報を、サービスエンジニアが現地に行く前に提供することができます。

2.7 まとめ

この章では、ServerView Suite を使用したサーバ管理の計画と実現に関する重要な概念について説明しました。

最も重要な概念を、次のようにまとめます。

- サーバ管理コンポーネントのカテゴリ分類

管理アプリケーションと通信するエージェントは、通常、監視対象サーバ上で実行します。管理アプリケーションは管理機能を提供し、必要に応じて、他の管理アプリケーションまたはヘルパーと連携します。管理コンソールは、これらの管理アプリケーションにアクセスするために使用します。
まとめ

- 原則：エージェントと管理ステーション

ここでは、監視対象コンポーネント（エージェント）と監視コンポーネント（管理ステーション）の概要について説明しました。これらの2つのコンポーネントは、連続的に実行される管理サイクルにかかわっています。管理サイクルは、監視、分析、適用、実行の4つのフェーズで構成されています。

- 階層的な設定

サーバ管理コンポーネントを異なるレベルで使用することにより、階層的な設定を実現できます。ここでは、5つの設定例を提示しました。ローカル管理、ポイントツーポイント管理、統合管理、ServerView Event Managerのカスケード設定や、他の管理システムへの統合です。

- プロトコル

オープンであることは、ServerView Suiteの重要な基本概念です。したがって、可能な限り、標準プロトコルが使用されています。ここでは、次のプロトコルの最も重要なポイントについて説明しました。SNMP、CIM、IPMI、PXE、Telnet、HTTP、HTTPS、SSL、SOAP、ITIL。

- 管理モード

監視対象サーバ上のOSの状態（OS-up/OS-down）や、管理に使用する通信パス（インバンド／アウトオブバンド）に応じて、PRIMERGYおよびPRIMEPOWERサーバの管理に、異なる管理モードを適用できます。
3 ServerView の位置づけ

この章では、ServerView と他のエンタープライズ管理システムとの関連について説明します。

図 10 に、エンタープライズ管理領域の概要図を示します。

図 10: エンタープライズ管理

ここでは、管理領域と管理機能が区別されています。
管理領域
管理領域は、管理対象に着目した区分です。

- システム管理は、ハードウェアレベルから OS レベルまで、コンピュータシステムに属するすべてのオブジェクトを対象とします。もちろん、ハードウェアやファームウェアに独特の特性を考慮する必要があります。これまでシステム製造元でなければ最適なシステム管理ができなかったのは、この理由からです。標準規格は、可能な限り製造元に依存しないための手段ですが、特定の領域では、現在でも標準規格がサポートされていません。

- アプリケーション管理は、常に、特定のアプリケーションに依存します。この領域には、標準規格はほとんどありません。

- メモリ管理は、コンピュータシステムとは別にメモリ容量を提供するメモリシステムを対象とします。

- ネットワーク管理は、ネットワークを構成しているすべてのオブジェクトを対象とします。

それぞれの管理領域では、適切な管理システムを使って、管理対象とするオブジェクトのライフサイクルの全フェーズをサポートする必要があります。図では、「ライフサイクルの矢印」として表現されています。

ServerView Suite の管理領域は、PRIMERGY および PRIMEPOWER サーバのシステム管理です。

管理機能
管理機能は、図では縦長四角の列として表現されています。各機能は、1つ以上の管理領域に基づくことができます。例えば、総合的な資産管理では、すべてのオブジェクトの情報を取得するために、4つのすべての管理領域にアクセスできます。図に示した管理機能のリストは、完全ではありません。これらは、いくつかの重要な管理機能にすぎません。管理者の業務を簡素化するために、継続的に新しい機能が追加されています。

ServerView Suite は、次の主要な管理機能を提供します。

- 問題管理
- パフォーマンス管理
- 資産管理

ServerView Suite の統合機能を使って、対応する ServerView をエンタープライズ管理システムに統合できます。これにより、エンタープライズ管理システムで使用できる管理機能が大幅に充実します。
### 3.1 ServerView Suite

ServerView Suite は、PRIMERGY サーバのハードウェアとファームウェアに最適化されたシステム管理を提供します。これらのサーバの各モデル独自のハードウェアとソフトウェアの特性は、ServerView Suite により、それぞれのライフサイクルの全フェーズにわたって最適に使用されます。

ServerView Suite は、多数のサーバの監視や管理にも使用できます。ServerView では、前述したサーバ管理領域や管理機能において、サーバ数の制限がほとんどありません。ServerView Suite はそのオープン性によって、次の 2 つの方向に向けて統合できます。

- ServerView Suite を、ServerView Suite がサポートしていない管理機能を提供している他の管理システムに統合する。
- PRIMERGY の設定では特に重要となる各種スイッチなど、他の管理領域にある個々のオブジェクトを ServerView Suite に統合する。

### 3.2 他の管理システムへの統合

ServerView Suite を他の管理システムに統合する場合は、標準プロトコルおよび標準インタフェースが基盤となります。ServerView 統合モジュールにより、ServerView のコンポーネントと他の管理システム間のリンクが作成されます。これにより、他のシステムの管理機能を PRIMERGY サーバで使用できるようになります。同時に、この統合により、ServerView の機能をシステム管理に利用できます。
他のコンポーネントの統合

ServerView 統合モジュールは、次の管理システムに対して利用できます。

- Microsoft SCOM
- Microsoft SCCM
- Microsoft SC VMM
- Microsoft SC PRO Packs
- VMware vCenter
- Nagios
- Icinga
- HP Systems Insight Manager
- CA Spectrum

3.3 他のコンポーネントの統合

ここでは、ServerView Suite を使用している管理環境に、PRIMERGY サーバ以外のオブジェクトを統合する場合の 2 つの方法について説明します。

ServerView Event Manager への統合

ServerView Event Manager は、標準管理プロトコルの SNMP に基づいて実現しています。したがって、SNMP を使用して、ServerView Suite に他の IT コンポーネントを統合できます。

SNMP エージェントを実行している IT コンポーネントは、ServerView Event Manager に対して、自身のトラップを送信できます。Event Manager は、対応するトラップ定義を認識すると、ServerView エージェントからのトラップと同様に、そのトラップを処理します。

ServerView Event Manager には、すでに多数のトラップ定義が統合されていま。例えば、ファイバチャネルスイッチ、RAID コントローラ、ストレージシステム、電源装置などからのトラップが含まれています。

ServerView エージェントを使用した統合

別の統合方法として、監視対象システム上に ServerView エージェントをインストールする方法があります。ServerView エージェントをインストールしたシステムの情報は、PRIMERGY サーバの情報と同様に、ServerView コンソール上に表示されます。
この方法で、HP サーバを ServerView Suite に統合できるようになりました。したがって、ServerView Suite を使用して異種環境を完全に管理できます。詳細については、ホワイトペーパー『Integration of HP Servers into ServerView Operations Manager』を参照してください。