

インフラストラクチャの計画とデザイン

Windows Server® 2008 Terminal Services

Version 1.0

Published: February 2008

For the latest information, please see [microsoft.com/technet/SolutionAccelerators](http://www.microsoft.com/technet/solutionaccelerators/default.mspx)

Copyright © 2008 Microsoft Corporation. All rights reserved. Complying with the applicable copyright laws is your responsibility. By using or providing feedback on this documentation, you agree to the license agreement below.

If you are using this documentation solely for non-commercial purposes internally within YOUR company or organization, then this documentation is licensed to you under the Creative Commons Attribution-NonCommercial License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc/2.5/ or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

This documentation is provided to you for informational purposes only and is provided to you entirely “AS IS”. Your use of the documentation cannot be understood as substituting for customized service and information that might be developed by Microsoft Corporation for a particular user based upon that user’s particular environment. To the extent permitted by law, MICROSOFT MAKES NO WARRANTY OF ANY KIND, DISCLAIMS ALL EXPRESS, IMPLIED AND STATUTORY WARRANTIES, AND ASSUMES NO LIABILITY TO YOU FOR ANY DAMAGES OF ANY TYPE IN CONNECTION WITH THESE MATERIALS OR ANY INTELLECTUAL PROPERTY IN THEM.

Microsoft may have patents, patent applications, trademarks, or other intellectual property rights covering subject matter within this documentation. Except as provided in a separate agreement from Microsoft, your use of this document does not give you any license to these patents, trademarks or other intellectual property.

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places and events depicted herein are fictitious.

Microsoft, Active Directory, SoftGrid, SQL Server, Windows, Windows Server, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

You have no obligation to give Microsoft any suggestions, comments or other feedback (“Feedback”) relating to the documentation. However, if you do provide any Feedback to Microsoft then you provide to Microsoft, without charge, the right to use, share and commercialize your Feedback in any way and for any purpose. You also give to third parties, without charge, any patent rights needed for their products, technologies and services to use or interface with any specific parts of a Microsoft software or service that includes the Feedback. You will not give Feedback that is subject to a license that requires Microsoft to license its software or documentation to third parties because we include your Feedback in them.

### Contents

[Planning and Design シリーズのアプローチ 1](#_Toc229552853)

[序文 3](#_Toc229552854)

[Microsoft Infrastructure OptimizationにおけるTerminal Servicesの位置づけ 4](#_Toc229552855)

[Windows Server 2008 Terminal Servicesのデザイン･プロセス 5](#_Toc229552856)

[Step 1: プレゼンテーション層における仮想化プロジェクトのスコープを決定する 11](#_Toc229552857)

[Step 2: 配信されるアプリケーションと、それらの用法について判断する 12](#_Toc229552858)

[Step 3: Terminal Services とアプリケーションの相性を判断する 15](#_Toc229552859)

[Step 4: ユーザーの分類 20](#_Toc229552860)

[Step 5: ターミナル･サーバー･ファーム数を決定する 21](#_Toc229552861)

[Step 6: アプリケーションとユーザーを、ファームにマップする 23](#_Toc229552862)

[Step 7: ファームのデザイン 24](#_Toc229552863)

[Step 8: ユーザー･データをストアする場所を決定する 33](#_Toc229552864)

[Step 9: ファームにおける Terminal Servicesのサイズと配置を考える 38](#_Toc229552865)

[Step 10: セキュアな通信 51](#_Toc229552866)

[まとめ 54](#_Toc229552867)

[Appendix A: ユーザーとアプリケーションに関するデータを、補足資料に記録する 57](#_Toc229552868)

[Appendix B: アプリケーション分析に関する補足資料 59](#_Toc229552869)

[Appendix C: ファーム･デザインに関する補足資料 61](#_Toc229552870)

[Appendix D: サーバー･パフォーマンスの分析とスケーリング 63](#_Toc229552871)

[謝辞 80](#_Toc229552872)

# Planning and Design **シリーズのアプローチ**

このガイドは IPD シリーズ一部であり、マイクロソフトの インフラストラクチャ･テクノロジーに関するデザイン･プロセスを明確にし、その合理的な運用を目指していくものです。このシリーズにおける個々のガイドでは、それぞれのインフラストラクチャに関するテクノロジーとシナリオに取り組んでいきます。 それらのガイドには、以下のテーマが含まれます：

* プランニング･プロセス全般におよぶ、技術面からの意思決定フローの定義。
* 実施すべき決定と、そのための判断時に考慮すべき各種の共通オプションに関する説明。
* コストや複雑さなどの、ビジネスに関連する判断と選択肢。
* ビジネスの視点からの、包括的で適切な理解を促進する、付加的な問題点を判断するのための枠組み。

このシリーズにおけるガイドは、製品ドキュメントを補完し補強していくという意図を持ちます。

## **このドキュメントにおけるアプローチ**

このガイドは、Terminal Services に関するインフラストラクチャの、適切な実装を促進するために作成されました。そして、そこで必要となる最も重要な取り組みと判断のための、一貫した構造を提供するようにデザインされています。

それぞれの取り組みと判断は、以下の４つの要素に細分されます：

* 取り組みと判断における背景のことであり、前後関係の設定と全体的な検討項目をを含む。
* 取り組みにおける、汎用的な Option と Task。
* それぞれのOption と Task におけるコストや複雑さといった項目を、評価するためのリファレンス･セクション。
* 実施されるべき決定事項に対して、重要な影響をおよぼすと思われるビジネス上の疑問点。

このドキュメントは Step 1 ～ Step 10 の意思決定フローで構成され、その中にいくつかの Option と Task が含まれます。Table 1 に列挙されるのは、各セクションの Optionで解説される特質の概要です。それらの項目にしたがって、各 Step の中では特定された Option に関する特質だけが説明されます。

Table 1. Option の特質

| 特質 | 内容 |
| --- | --- |
| 複雑さ | Option 間に干渉をもたらす、複雑さに関する特質です。 |
| コスト | Option の初期コストと継続的なコストに関する特質です。 |
| フォールト･トレランス | システムの可用性に大きな影響をおよぼす、インフラストラクチャにおける弾力性をサポートするための方式を示します。 |
| パフォーマンス | インフラストラクチャの性能に影響をおよぼす、Option に関する特質です。 |
| スケーラビリティ | インフラストラクチャのスケーラビリティに影響をおよぼす Option を示します。 |
| セキュリティ | この値は、全体的なインフラストラクチャに対して、プラスあるいはマイナスにはたらく Option を反映します。 |

それぞれのデザインにおける Option は、上記の表と同じ観点を持ち、また、特質に対する相対的な重み付けを提供するために主観的に評価されます。正確に Option を比較するとき、ビジネスの要素が増えるにつれて、その評価は不明瞭なものとなります。そのときの等級は、以下の2つの形態を持ちます：

* 複雑さとコストは、High／Medium／Low で評価される。
* 他の特質は、以下のテーブルにおけるスケールで評価される。

Table 2. 付加的な特質

| Symbol | Definition |
| --- | --- |
| ↑ | 特質に対してプラスに作用します。 |
| → | 特質に影響を与えない状態、もしくは、比較の基準が無い状態を表します。 |
| ↓ | 特質に対してマイナスに作用します。 |

このセクションで説明している特質は、2カラムあるいは3カラムのテーブルとして提示されます。対象となる 特質が、すべての Option に適用可能なとき、あるいは利用可能な Option が存在しないときには、2カラムのテーブルが使用されます。例としては、タスクを実施するときなどが挙げられます。

3カラムのテーブルは、特質における、選択肢と、詳細、効果（Table 2）を記述するために、その順番で使用されます。

## **このガイドが想定する読み手**

このガイドは、Terminal Services 実装のプランニングとデザインに責任を持つ、ITインフラストラクチャの専門家のために記述されています。具体的には、Windows Server 2008 環境から、Remote Desktop Client（RDC）ソフトウェアを実行するデバイスに、デスクトップやアプリケーションをサポートしていくことが目的となります。それらを実践していくスペシャリストには、仮想化に関連するデザイン上の判断を専門とする、コンサルタントや IT スタッフなどが含まれます。

このガイドのコンテントは、Terminal Services サービス･テクノロジーに精通した読み手を前提にしています。そして、Windows Server 2008 上の Terminal Services を用いて、サーバーやファームを実装していくためのプランニングも取り込んでいます。

**Note**  ターミナル･サーバー･ファームとは、同一のアプリケーション･セットを配信する、ターミナル･サーバーのグループのことです。

# **序文**

このガイドでは、段階的に読み手を導きながら、Windows Server 2008 Terminal サービスのインフラストラクチャについて、プランニング･プロセスを説明してきます。具体的には、以下の基本的な判断と課題に取り組んでいきます：

* Terminal Services を用いて配信すべきアプリケーションを識別し、その適切な使用法について判断していきます。
* 選択されたアプリケーションを、Terminal Services を用いて提供する際に、必要とされるリソースについて決定します。
* Terminal Services インフラストラクチャにおける、コンポーネント、レイアウト、セキュリティ、接続性などをデザインしていきます。

プロジェクトにおける全体的なビジネス目的を、テクニカルなデザインを開始する前に理解することが、きわめて重要になります：

* プレゼンテーション層を仮想化することで、ビジネス面において、どのようなメリットが達成されると予測するのでしょうか? プレゼンテーション層の仮想化は集中化されたシステムを用いて、多数のユーザー･セッションをホストし、また、すべての処理をホスト･システムで実行します。そこでのユーザー･セッションは、相互に分離された状態におかれます。そして、キーボードとマウスの入力やビデオ更新といった、プレゼンテーションのための情報だけが、クライアントとホストシステムの間で転送されます。Windows ベースのフル･ワークステーションあるいは 、Windows ベースのターミナル･デバイスを、クライアントとして用いることが可能です。
* それらのメリットにつて、どのように評価したら良いのでしょうか? また、それらのメリットを Terminal Services から得るために、どれだけのコストがかかるのでしょうか?
* そのコストの正当性は、複数のビジネス･グループに受け入れられるのでしょうか? そして、このプロジェクトが成功することで、グループ間の関係は適切な方向へ導かれるのでしょうか?

プロジェクトを開始する際には、ビジネスの目的について、適切な優先順位を付けなければなりません。そうすることで、IT 部門とビジネス部門の間で、相互に目的が理解され、また、合意が形成されていきます。このようなプロセスが必要とされる理由としては、いくつかのアプリケーションと Terminal Services が直ちに適合しない可能性があること、そして、それらを変更するためにはコストがかかることが挙げられます。そのため、こうした変更に着手する前に、ビジネス部門へのフィードバックを行い、追加のコストに対する理解と、最適なビジネス上の判断を得るようにする必要があります。

# Microsoft Infrastructure Optimization**における**Terminal Services**の位置づけ**

マイクロソフトのInfrastructure Optimization (IO) Model は、組織の成熟度を連続的にカバーするかたちで、IT のプロセスとテクノロジーを統合するためのものです（詳細については Microsoft.com/io を参照してください）。このモデルは、Massachusetts Institute of Technology (MIT) Center for Information Systems Research (CISR)の産業アナリストおよび、マイクロソフト自身と顧客の経験から作成されています。マイクロソフトが Infrastructure Optimization Model を作成する主な目的は、技術的な能力とビジネスを測定するためのベンチマークとして、柔軟で容易で成熟したフレームワークを利用する、シンプルな方式を開発することにあります。

このInfrastructure Optimization Modelは、核となるインフラストラクチャの最適化、および、アプリケーション･プラットフォームの最適化、そして、ビジネスの生産性のためのインフラストラクチャの最適化といった、3種類の IT を取り囲むかたちで構造化されています。核となるインフラストラクチャの最適化モデルにしたがい、Windows Server 2008 における作業負荷の軽減に、プレゼンテーションの仮想化を実装する組織は、合理化（Rationalized）の レベルへと移行するための要件を満たすことになります。このガイドでは、Terminal Services を実装することで、アプリケーションや、デスクトップ、そしてエンドユーザーをサポートする、インフラストラクチャを対象としたプランニングとデザインを支援していきます。



図1. 核となるインフラストラクチャ･モデルにWindows Server 2008 Terminal Servicesをマッピングする

## **インフラストラクチャとビジネスのアーキテクチャについて**

マイクロソフトが作成する、意志決定のための構造化されたガイダンスは、ITインフラストラクチャとビジネス･アーキテクチャを対象としています。そして、Infrastructure Planning and Design シリーズが提供する構造化された手法と判断の形態は、IT のインフラストラクチャ･アーキテクチャに関するものとなっています。その一方で、マイクロソフトのビジネス･アーキテクチャ･テンプレートは、価格計算や、支払のプロセス、発注業務といった、ビジネスにおける詳細な項目にフォーカスしています。 IT インフラストラクチャはビジネスの可能性に貢献すべきであり、ビジネス･アーキテクチャの要件はインフラストラクチャの決定に寄与すべきです。しかし、Infrastructure Planning and Design シリーズは、特定のビジネス･アーキテクチャ･テンプレートを定義することはく、また、そこに関連付けられることもありません。それに替えて、Infrastructure Planning and Design では判断のための重要なポイントを提供し、そこで、サービス･マネージメントやビジネス･プロセスなどの要素を取り込んでいきます。

ビジネス･アーキテクチャのツールとモデルの詳細については、マイクロソフトへお問い合わせください。

# Windows Server 2008 Terminal Services**のデザイン･プロセス**

Windows Server 2008 Terminal Services に関する IPD のゴールは、Terminal Services インフラストラクチャの設計と作成において必要とされる、情報収集および、判断、設計、選択を介して、プランナーを導くことです。

その目的は、適切なサイズとコンフィグレーションにより配置されるインフラストラクチャを用いて、定義されたビジネス･メリットを提供する点にあります。そして、エンドユーザーのエクスペリエンスや、セキュリティ、マネージャビリティなどについて、また、システムの性能や、容量、フォールト･トレランスなどについても、検討を加えていきます。この ガイドで取り上げるシナリオは、Terminal Services インフラストラクチャを設計する際に、誰もが遭遇する可能性のあるものです。 Microsoft Customer Service and Support によるアーキテクチャのレビューについても、実装に取り掛かる前に検討すべきです。そうすれば、独自にデザインした環境のサポータビリティについて、適切に把握することが可能になるでしょう。

Windows Server 2008 Terminal Services の主要コンポーネントを、図 2 に示します。



図 2. Windows Server 2008 Terminal Services のアーキテクチャ

この図が示すのは、Terminal Services を構成するコンポーネント間の関係であり、それらが協調動作することでアプリケーションが配信されます。ここでは、具体的な実装モデルを説明するために、ひとまとめにコンポーネント群を表示していますが、この図とは異なる形態で、構成することも可能です。

一般的に、ターミナル･サーバーが実装されるファームは、高度なスケーラビリティとフォールト･トレランスを提供する環境となります。ターミナル･サーバーのファームとは、同じセットのアプリケーションを配信する、ターミナル･サーバーのグループのことです。つまり、ファーム内のターミナル･サーバーは、同一のエクスペリエンスを配信することになります。そして、サーバーごとのハードウェアの差異にかかわらず、同一のアプリケーションに対して、同一のコンフィグレーションが行われます。

なお、ロード･バランシングされたファーム内の、いずれかのメンバー･サーバーに対して、ユーザーを一意の名前で接続することが可能です。また、ファームに参加するサーバー数に、プログラム上の制限はありません。

## **判断および決定**

このガイドでは、Windows Server 2008 Terminal Services の準備において必要となる、以下の判断とアクティビティに取り組んでいきます。この 10 段階の Step は、Windows Server 2008 Terminal Services のデザインを適切に計画していくための、最も重要な要素を表しています：

* Step 1: プレゼンテーション層における仮想化プロジェクトのスコープを決定する
* Step 2: 配信されるアプリケーションと、それらの用法について判断する
* Step 3: Terminal Services とアプリケーションの相性を判断する
* Step 4: ユーザーの分類
* Step 5: ターミナル･サーバー･ファーム数を決定する
* Step 6: アプリケーションとユーザーを、ファームにマップする
* Step 7: ファームのデザイン
* Step 8: ユーザー･データをストアする場所を決定する
* Step 9: ファームにおける Terminal Services ロール･サービスの、サイズと配置を考える
* Step 10: セキュアな通信

いくつかの項目は、実施されなければならない決定を表します。そのようなケースにおいては、Option がリストアップされ、その中から一般的な回答が得られるでしょう。

同様に、インフラストラクチャ･デザインを完成させるために、実施しなくてはならない重要な Task も、これらの Step 内には含まれます。

## **意思決定フロー**

上記の各 Step は、組織として判断し決定していくポイントを示しており、それぞれが意思決定フローのノードに該当します。また、インフラストラクチャのデザインを完了するために、組織として選択すべき Option と、達成すべき Task が、それらの Step に含まれています。



図3. Windows Server 2008 Terminal Servicesに関する意思決定フロー

## **適用可能なシナリオ**

このガイドでは、Windows Server 2008 Terminal Services インフラストラクチャを成功に導くために必要な、コンポーネントのプランニングとデザインに取り組んでいきます。このガイドを作成する際に検討されたシナリオには、以下の各項目が含まれます：

* 集中化されたデータセンターあるいは、ローカルなデータセンターの内に、デスクトップ環境を集約しようと計画している組織にとって有効です。
* 企業の合併や買収の後に、コンピュータ･リソースとユーザーを統合していこうとする組織にとって有効です。双方の企業の各部門から、アプリケーションにアクセスできるようにするためには、既存のデスクトップと平行して、別のリッチ･クライアント･デスクトップやアプリケーションが必要になるかもしれません。このような状況において、Terminal Servicesを用いることで企業システムの統合が加速されます。その場合には、アプリケーション･アクセスにおけるセキュリティとディレクトリの問題が関心事となりますが、このドキュメントのスコープには含まれません。
* Windows Server 2003 Terminal Services と平行して、Windows Server 2008 Terminal Services を実装しようとする組織にとって有効です。
* 新しいオペレーティング・システムへの更新を待っているクライアントもしくは、更新が不可能なクライアントに対して、新規のフル･デスクトップ環境の提供を選択するかもしれない企業にとって有効です。
* 新しいアプリケーションをデスクトップへのインストールするのではなく、エンタープライズのいたるところで素早い展開を実現することで、エンドユーザーによる利用を迅速に開始したい状況に有効です。
* 互換性の問題のために、更新不能あるいは更新待ちのクライアントに対して、各種アプリケーションの最新バージョンを迅速に展開したい状況に有効です。
* パートタイムあるいはフルタイムで作業する在宅勤務の従業員に対して、そのパーソナル･ワークステーション上にフル･コーポレート･デスクトップを提供する場合に有効です。この方式により、クライアントのメンテナンスとセキュリティに関与することなく、フル機能のアプリケーションを安全に提供ができるようになります。
* ベンダーやサプライヤーといったサードパーティに対して、各種のアプリケーションをWeb ブラウザを用いて提供する場合に有効です。
* 被災時におけるビジネスの連続性を、確保する場合に有効です。新しいロケーションで新しいワークステーションを取得あるいは貸与する場合や、移動するユーザーに完全なデスクトップを迅速に共有する場合に、Terminal Services の利用が可能です。
* マルチ･フォレスト環境において、Terminal Services の基本コンポーネントはフォレスト境界線をまたぐかもしれません。この方式は、合併や買収のシナリオにおいて、きわめて適切なものとなります。
* メンテナンスが困難なアプリケーションや、使用頻度の低いアプリケーションのプロビジョニングに対応します。エンドユーザーのワークステーション上で、このようなアプリケーションを利用する場合には、膨大な間接費が必要になるかもしれません。それに代えて、Terminal Services を介して集中化されたアプリケーションを実行することで、ビジネスの効率化が促進されます。
* 大量のデータを取り扱うクライアント･アプリケーションを、低帯域幅のネットワーク上で利用する場合に対応します。このようなケースにおいて、Terminal Services を用いたアプリケーションの提供により、大量のデータをリモートでアクセスし操作する方式が、きわめて有効なソリューションとなります。その理由は、 実際のデータではなく、データのスクリーン･ビューだけが、ネットワーク上のクライアントに送信される点にあります。

## **スコープから除外される要素**

このガイドがフォーカスするのは、Terminal Services のデザインとプランニングだけであり、以下の要素を含むソリューションはスコープから排除されると考えるべきです：

* リモート･アシスタンス. この機能もRemote Desktop Protocol （RDP）を用いますが、プレゼンテーション層の仮想化というより、ユーザー･アシスタンスのために利用されます。
* アドミニストレーション･デスクトップ. この Terminal Services の機能は、アドミニストレーションを目的とした Windows デスクトップ･セッションへの接続を、最大で2人のリモート･ユーザーに許可します。この機能は、Windows Server 2003 と Windows Vista Ultimate に加えて、Windows 2000と Windows XP のプロフェッショナル･エディションにも含まれます。
* マルチテナントおよびリモートでのホスティング. リモート･アプリケーションと共有サーバーをホストする企業では、いくつかの形式によるプレゼンテーション層の仮想化が利用されます。ただし、このドキュメントでは、完全なマルチテナント･デザインにおける、すべての複雑さを取り扱うわけではありません。
* Windows 2000 Server と Windows Server 2003 の Terminal Services からの移行. このドキュメントは、従来からの Terminal Services を、Windows Server 2008 の Terminal Services へ移行させるためのガイドではありません。
* Citrix MetaFrame. この製品は、Terminal Services に対するサードパーティー･アドオンであり、拡張された機能および、各種の状況に応じたソリューションを提供しますが、ここではカバーしません。
* Microsoft Application Virtualization に関するプランニング. アプリケーション競合を解決し、アドミニストレーションを大幅に単純化するために、Microsoft Application Virtualization on Terminal Services の利用が可能です。Terminal Services インフラストラクチャを設計するときの補足資料として、Infrastructure Planning and Design Microsoft SoftGrid® のApplication Virtualization Solution Accelerator を参照してください。<http://www.microsoft.com/ipd>

## **参考資料**

* 以下のURL のTerminal Services on the Windows Server 2008 TechCenter のページを参照してください ： <http://go.microsoft.com/fwlink/?LinkID=73931>.
* IPD のMicrosoft App-V を参照してください：<http://www.microsoft.com/ipd>.

# Step 1: **プレゼンテーション層における仮想化プロジェクトのスコープを決定する**

プレゼンテーション層を仮想化するためのインフラストラクチャをデザインする前に、プロジェクトにおける設計と目的に取り込むべき、対象となる環境のパートについて、組織的に判断する必要があります。

この Step 1 では、Terminal Services で利用するアプリケーションの種類や、ユーザー･ポピュレーションなどに関する判断を進めていきます。また、サーバーのファーム数や、サイズ、配置についても、プロジェクトのスコープを定めることで促進していきます。最適なサービスを提供するためには、これらの展開に関する Option を組み合わせていくことが、きわめて頻繁に要求されます。Terminal Services が提供するアプリケーションとしては、たとえば、エンタープライズをカバーするタイムカード･システムや、会計部門用の経理パッケージ、そして、各国に点在するオフィスごとの税務パッケージなどが挙げられます。

## Task 1: **ロケーション･スコープの決定**

プランナーがソリューションを構成する際に、その境界線を認識するためのプロジェクト･スコープを、アーキテクチャを生成する前に決定しなければなりません。このプロジェクトのスコープは、エンタープライズ全体あるいは、複数のロケーション、そして、単独の部門になるかもしれません。数多くの Terminal Services を実装する場合には、それぞれのインスタンスに対して、このガイドを反復して適用することができます。

プランナーがプロジェクトのスコープを、はっきりと確かめることが重要であり、それを怠ると、プロジェクトの細部に陥ってしまうようなことが容易に起こり得ます。スコープを定めることで、ビジネスの目的に見合ったプロジェクトを維持するための、テクニカルな判断が適切に行われ、また、フォールト･トレランスや、キャパシティ、パフォーマンスに関する適切なトレードオフがもたらされます。

プロジェクトのロケーション･スコープが、明確に維持されるようにするために、それらを記録していくべきです。それた記録は、たとえば Appendix A、B、Cで示されるように、この後で作成されるジョブ･エイドのヘッダーとして利用できます。

## Task 2: **アプリケーション･スコープの決定**

この段階において、プロジェクトのロケーション･スコープが決意しているはずです。この Task 2 では、Terminal Services でホストするビジネスを、正確に定義していきます。

* デスクトップ全体と、すべてのアプリケーションを、Terminal Services でホストすることが目的なのか?
* 単一のアプリケーションを、Terminal Services を介して利用可能にすることが目的なのか?
* 数多くのアプリケーションを、Terminal Services でホスト可能にすることが目的なのか?

必要とされるサービスの提供にフォーカスする、継続したプロジェクトを保証するためには、ビジネスの目的を完全に理解することが、きわめて重要となります。

* プレゼンテーション層を仮想化することで、どのようなメリットがビジネスにもたらされると予期するのか?
* それらのメリットの価値は何なのか? そして、それらのメリットを提供するために Terminal Services を用いると際の、コストの事例はあるのか?

ビジネス目的における優先順位付けは、スタート時から正しいく行われるべきです。それにより、それらのビジネス目的は正確に理解され、また、IT とビジネスの間に合意が生じます。

プロジェクトのアプリケーション･スコープが、明確に維持されるようにするために、それらを記録していくべきです。それた記録は、たとえば Appendix A、B、Cで示されるように、この後で作成されるジョブ･エイドのヘッダーとして利用できます。

配信されるアプリケーション環境と、ビジネスにおける期待との組み合わせも、重要になります。アプリケーション環境の制御が失われることについて、また、新しい運用モデルがもたらす追加の作業について、ユーザーは必然的に関心を持つでしょう。そのことが、ユーザーが気づくと思われる変更点について、オープンに議論していく良い機会になるでしょう。それらの変更点の例として、資格証明をユーザーに追加していくという課題があります。アプリケーションに最初にアクセスするとき、一度で認証をパスするためには、シームレスなエントランスを提供しなければなりません。 ただし、そのためには、最新テクノロジーの利用と、いくつかのアップグレード、そして、追加のコストが必要になるかもしれません。

## **意思決定の概要**

プロジェクトのスコープは、組織固有の必要性に基づいて決定されなくてはなりません。Windows Server 2008 上の Terminal Services についてプロジェクト･スコープを定義することで、この後に続く各 Step において、キャパシティ要件に関する判断などが促進されていきます。そのときに用いる手段として、最適と思われるいくつかのアプローチがあるでしょう。しかし、どのような進め方が採用されるにしても、プランニングのプロセスに着手する前に、組織が選択したアプローチに対して、足並みを揃え、集中し、サポートしていくべきです。

# Step 2: **配信されるアプリケーションと、それらの用法について判断する**

Step 1 では、ロケーションとアプリケーションについて、プロジェクトのスコープを決定しました。

この Step 2 のゴールは、スコープに取り込まれるエンドユーザーに注目すること、そして、アプリケーションの種類と、それらの用法について、判断することにあります。この Step 2 が完了すると、プロジェクトのプランナーは、候補となるアプリケーションのリストを手にすることになります。

## Task 1: **ユーザーとアプリケーションの情報を収集する**

Step 1で作成されたプロジェクト･スコープを、ユーザー･ポピュレーションを取り囲む境界線として用いて、それぞれのユーザーについて、以下の情報を収集していきます。

なお、以下の各項目は、一般的なユーザーを想定して作成したものであり、Appendix Aの ”ユーザーとアプリケーションに関するデータを、補足資料に記録する” に該当するものとなっています。スプレッド･シートなどを用いて情報を整理することで、収集のための作業が効率よく進みます。

* ユーザーのロケーション. それぞれのロケーションにおけるユーザー数を把握することで、サーバーのサイズと配置に関する判断を支援します。
* クライアントのオペレーティング･システム. Terminal Services によりアプリケーションが配信されるときには、エンドユーザーのマシン上で実行される Remote Desktop Client（RDC）との間で、RDP プロトコルを用いた通信が行われます。 しれにより、スクリーン更新および、キーボード入力、マウス･クリックなどの情報が、ネットワーク上で交換されていきます。RDC と RDP の最新バージョンは、暗号化におけるレベルの向上と、ネットワークを介したデータ転送効率を、大幅に拡張しています。ただし、クライアントにインストールされているオペレーティング･システムの種類により、実行可能な RDC と RDP のバージョンが限定されるため、それらの拡張部分の利用にも制約があります。たとえば、シングル･サインオンの機能は、Windows Vista Service Pack 1（SP1） もしくは、Windows XP SP3 を要求します。
Step 5 でターミナル･サーバーのファーム数を決定する際に、こうした情報が利用されますが、その理由は、それぞれのファームにおけるクライアントに対して、最高と最低のセキュリティ設定が必要とされるからです。同様に、それらの情報は Step 9においても、TS Gateway ロール･サービスをデザインするために用いられるでしょう。
* アプリケーションに対するシングル･サインオンが提供されるのか. ユーザーがシングル･サインオンの利便性を期待する場合には、アプリケーション配信で用いる Remote Desktop Client（RDC）のレベルが、そのことの可否を決定するでしょう。配信されたアプリケーション環境で、このエクスペリエンスを提供するためには、RDC 6.1 が必要とされます。そのために使用される RDC のレベルが、Step 5 で要求されるファーム数と、Step 9 でデザインされるロール･サービスに、影響を与えるかもしれません。
* 使用されるアプリケーションと、そのバージョン、そして、特殊なローカル･カスタマイズについて. 前提となるアプリケーションに、複数のバージョンが存在する場合には、同一の Terminal Services 環境での共存について、確認していく必要性が生じるでしょう。共存が不可能な場合には、別々のターミナル･サーバー･ファームから、それぞれのバージョンやカスタマイズされたものを配信する必要が生じ、プロジェクトの複雑さとコストが増大していくでしょう。この問題については、Step 5 で判断することになります。
Microsoft App-V を用いて、それぞれのアプリケーションをターミナル･サーバー上で実行することで、この問題を解決できるかもしれません。そのような場合には、対象となるアプリケーションの配信を、Microsoft App-V でサポートする必要があり、また、 Microsoft App-V 環境のインスタンスも必要になります。
**Note** このドキュメントを記述する時点において、Microsoft App-V は 64-bit 環境をサポートしていません。
* SLA のレベル. どのような SLA を設定し、どのようなサービス･レベルを約束し、どれだけのユーザーを個々のサービス･レベルに取り込むのでしょうか? アプリケーションに対する企業やユーザーからの期待を理解することで、パフォーマンスと可用性の要件が定義されていきます。それにより、Step 7では、ターミナル･サーバー･ファームのサイジングが促進され、ユーザーの期待するパフォーマンスが提供可能となるでしょう。また、 Step 7 では、フォールト･トレランスのアプローチも決定されるでしょう。SLA が正式なものであっても、また、非公式なものであっても、ユーザ－･グループが期待するパフォーマンスと可用性のレベルを理解することが、とりわけ重要となります。そして、ユーザ－･グループが経験してきた実際のパフォーマンスや可用性よりも、このドキュメントから得られる情報を、デザインのインプットとして用いることが重要です。
* 接続のタイプ. すべてのユーザーが、LAN で接続されているのでしょうか? それとも、一部のユーザーは、WAN や 、ダイアルアップ、ISPを介して接続されているのでしょうか? ネットワークやセキュリティ･モデル、そしてサーバーの配置やサイズをデザインするために、それらの情報が利用されるでしょう。ユーザーへのインタビューでは、この種の情報を集めることは不可能かもしれません。それよりは、IT 担当者がインタビューから結論を導き出すことで、ネットワーク･デザインを検証することが必要になるでしょう。

さらに、以下の情報を集めることも有益な方式となるでしょう。アプリケーションと用法について追加の質問が生じる場合には、それらの情報がさらに有用になります。

* アプリケーション･サポート･グループ. アプリケーションを責任を持ってサポートする、担当者をリストアップします。これから発生してくる、テクニカルな問題に対する初期段階でのテストと調査において、能力を発揮する人材を把握します。
* アプリケーション担当者. アプリケーションに関するアップグレードとライセンス管理に責任を持つ、部門あるいは責任者をリストアップします。それにより、アプリケーション配信の方式について、変更を承認する責任者が明確にされ、また、発生すると思われるビジネス上の問題が把握されるでしょう。

## **意思決定の概要**

この Step 2 では、候補としてリストアップされたアプリケーションと、その用法における特徴を、補足資料であるスプレッドシートに記録します。 次に続く Step 3 では、Terminal Services で配信されるアプリケーションの適合性を評価していきますが、その前に、プロジェクト･スコープをビジネスの側面から再確認するには、最適なタイミングです。アプリケーションの用法と期待されるサービス･レベルについて、ビジネス面からの理解と確実に一致させるために、そのためのミーティングを行い、この Step 2 での調査結果をレビューすべきです。

次の Step 3 では、Terminal Services により配信されるアプリケーションの適合性を判断するために、それらの特徴について調査していきます。

## **参考資料**

* IPD の*Microsoft App-V* を参照してください：<http://www.microsoft.com/ipd>.

# Step 3: Terminal Services **とアプリケーションの相性を判断する**

この Step 3 のゴールは、Terminal Services で使用できないアプリケーションや、使用すべきではないアプリケーションを識別することです。 そのためには、 Windows Server 2008 の Terminal Services を介した利用を前提としている、それぞれのアプリケーションについて調べなければなりません。

この Step 3 を進めていく間に、プロジェクト･スコープに含まれるアプリケーションは、Terminal Services 環境での配信に対する適合性にしたがって分類されてきます。そして、それらの情報を、Appendix B の”アプリケーション分析に関する補足資料” に該当する、スプレッド･シートに記録します。いくつかのアプリケーションについては、直ちにTerminal Services で配信できない可能性があるため、この Step 3 の最後で、分類されたリストをビジネスの視点からレビューすべきです。それらのアプリケーションを、このプロジェクトのスコープに残すためには、いくつかの変更が必要になるでしょう。それらの変更はコストの増大を生じるため、作業を開始する前に、そのための情報をビジネスの視点からレビューすべきです。それにより、追加のコストとリスクが理解され、また、最適なビジネス上の判断へとたどり着きます。

この Step が終了した後でもスコープに残っているアプリケーションは、次の Step 4 で、さらなるテストと測定の対象となります。そのような段階を踏むことで、対象となるアプリケーションをホストするために、サーバーとサーバー･ファームのサイズを最適化していきます。

## Task 1: **提供するアプリケーションの機能を調べる**

Windows Server 2008 のTerminal Services 環境において、ホストできるアプリケーションと、ホストすべきアプリケーションについて検討するとき、数多くの問題から影響を受けることになるでしょう。 それらの問題は、以下のように分類できます：

* アプリケーションにおけるビジネスの実現性に影響を与える問題
* アプリケーションの提供を阻害する問題
* マルチ･ユーザー･サーバーにおけるアプリケーションの振舞の問題

それらの問題に関するデータは、アプリケーション･ベンダーや、Terminal Servicesコミュニティ･サイト、アプリケーション･サポート要員などから集められるでしょう。

さらに、 Application Compatibility Toolkit（ACT）を、アプリケーションの評価に利用できます。Terminal Services 環境での安定性を対象とした、スタンドアロン･コンピュータ上での互換性の確認が可能となります。 以下のURLから入手できます：

<http://technet.microsoft.com/en-us/desktopdeployment/bb414773.aspx>,

以下の設問を用いて、それぞれのアプリケーションをレビューし、対象アプリケーションの適合性における最終的な判断にしたがって、 Appendix B 補足資料に評価結果を記録します。それにより、このプロジェクトから除外すべきアプリケーションが、抽出さられるでしょう。

### ビジネスの実現性に影響をおよぼす問題

アプリケーションを Terminal Services 環境で配信しようとするとき、ビジネス上の問題が、数多く生じるかもしれません。プレゼンテーション層の仮想化に対して、技術面で適合する アプリケーションであっても、それらの問題により配信が阻害される可能性があります。そのような状況に対処するために、最初に、以下の問題について検討します：

* アプリケーションのライセンスは、Windows Server 2008 Terminal Services 環境での実行に対応しているのか? この情報について、アプリケーション･ベンダーと確認します。
* アプリケーションは、Windows Server 2008 のTerminal Services上でサポートされているのか? それらのアプリケーションが Terminal Services 環境でサポートされない場合には、正確に動作しない状況や、特定の機能が利用できない状況が考えられます。さもなければ、アプリケーションが動作するにしても、追加のテストが必要という状況もあり得ます。サポートされないアプリケーションに関しても、実行する際のリスクについてレビューすべきです。それにより、Terminal Services でのアプリケーション実行について、保証の有無が確認されます。
* アプリケーションの配信が、ライセンス料を高額にする原因にならないか? ソフトウェア･ライセンスには、多数のモデルがあります。ファーム内の全てのターミナル･サーバーがアプリケーションをインストールしますが、対象となるプリケーションのライセンス構造のもとで、過度の料金の支払いという問題が、発生しないことを確認する必要があります。
* アプリケーション配信を阻害する、法的な事由が存在するのか? 国家の機密や患者の個人データのような秘匿性の高いデータが、たとえば国外に配置されたサーバー上に、存在することが許容されるのでしょうか? 信頼性の低いコンピュータやオフサイトに置くことが許されないデータは、どうでしょうか? ネットワーク上の保護されない領域に、データを転送できるのでしょうか?
* **RDC クライアントの大半の方式において、暗号化のレベルが要求されるのか?** RDC クライアントのバージョンが、望まれる暗号化レベルをサポートしない場合には、それらがアップグレードされるまで、アプリケーションの展開を待つべきです。代替案として、セキュリティが遵守されるクライアントだけにアプリケーションを配信し、その他のクライアントはローカルで実行するという方式があるかもしれません。

### 運用中のアプリケーションが利用できないという問題

アプリケーションを Terminal Services 環境で実行できなくしてしまう、デザイン上の制約が存在するかもしれません：

* アプリケーションをWindows Server 2008 のTerminal Services で実行するようにできるのか? ユーザー･アプリケーションの中には、旧式のオペレーティング･システム上で実行するように、記述されたものもあるでしょう。それらのアプリケーションが 、Windows Server 2008 の Terminal Services 上で、実行できることを確認していきます。実行できない場合には、互換性モードでのアプリケーション実行を確認するために、ACTを使用します。

### マルチ･ユーザー･サーバーで、障害を引き起こすアプリケーション

このプロジェクトが完了するとき、別々のワークステーション上で実行されている複数のアプリケーションを、1つのサーバー環境で一緒に実行できるようになるでしょう。そのために必要となるのは、対象となる環境において、それらのアプリケーションが共存し、相互に影響せずに動作することです。さらに、その環境における費用効果の限界に至るまで、スケーラビリティとパフォーマンスを低下させない方式で、アプリケーションを動作させなければなりません。 以下の設問を用いて、アプリケーションが含むと思われる、共通の問題を探っていきます：

* **アプリケーションは過大なリソースを消費するのか? それとも、リソースが適切にリリースされないのか?** CPUやメモリなどの共有サーバー･リソースを、何らかのアプリケーションが独占することは許されません。たとえば、スタンドアローンのコンピュータの場合は、小規模のメモリ･リークが大きな問題を引き起こすことは無いかもしれませんが、50人のユーザーが同じアプリケーションをコンカレントに実行する場合には、きわめて短時間でサーバー･メモリが枯渇してしまう可能性があります。この問題についてテストを行なう場合には、メモリの用法をモニタリングします。欠陥が発見されるなら、ソフトウェアの作成者が、その問題を解決すべきです。また、リークを生じながらアプリケーションを実行するリスクについても、評価すべきです。
* Windows Server では利用できないと思われる、クライアント･ベースのリソースに、アプリケーションはアクセスする必要があるのか? 対象となるアプリケーションが使用する、スキャナーや、プリンター、USB デバイス、フロッピー、ハードディスクといったクライアント･サイドのデバイスについて、Terminal Services 環境での互換性をチェックする必要があります。それらのデバイスの、Terminal Services 環境での利用について、その可否を判断しなくてはなりません。
* 適切で標準的なインストールとアンインストールを、アプリケーションは理解するのか? アプリケーションはインストレーションされた後に、すべてのユーザーから利用できるようになる必要があります。そして、アンインストールを行うときには、ターミナル･サーバー上の他のアプリケーションが必要とするコンポーネントを、削除してはなりません。 ACT ツールを活用することで、セットアップに関する問題について、判断が促進されていきます。何らかの問題が見いだされた場合には、サーバーとユーザーに対する影響について検討すべきです。
* アプリケーションは、共有のシステム･ファイルやコンポーネントを、変更してしまうのか? フォントや、DLL、ドライバなどのファイルを変更してしまうアプリケーションは、他のアプリケーションの機能に影響をおよぼしてしまいます。ACT を活用することで、こうした振舞いをモニタリングします。問題が検出された場合には、該当するアプリケーションの修正が、Terminal Services 上で使用する前に必要となります。

代替案として、Microsoft App-V の利用が考えられます。ターミナル･サーバー上の App―V 仮想環境で、それぞれのアプリケーションを実行することで、この問題を克服できるかもしれません。この場合には、Microsoft App-V によるアプリケーション配信をサポートする必要があり、また、Microsoft App-V 環境のインスタンスを作る必要があります。

* **ユーザー･データと、他のデータは分離されているか?** ユーザー･ファイルとテンポラリ･ファイルを、同じロケーションにアプリケーションが配置してしまう場合に、インスタンス間でのファイルの上書きや変更が行なわれ、予想できない結果を生じることが考えられます。その有無を確認するために、テストを行うべきです。問題が生じる場合には、該当するアプリケーションの修正が、Terminal Services 上で使用する前に必要となります。
* レジストリへのアクセスは適切か? アプリケーションはレジストリに対して、他のユーザーやアプリケーションに影響をおよぼすような、変更をしてはなりません。通常において、Terminal Services のインストールでは、標準的なデスクトップの HKLM\Software に設定されたレジストリ項目を、ターミナル･サーバーの HKCU\Software にリダイレクトしますが、こうした保護のための仕組みを回避する方式で、ソフトウェアが記述されているかもしれません。不適切なレジストリ領域に施された変更を監視するために、ACT あるいはレジストリ比較ツールを利用します。そのような変更が見いだされる場合には、サーバーとユーザーに対する影響を調査すべきです。
* ユーザーによるプログラム･オプションへの変更が、他のインスタンスに影響をおよぼすか? 特定のユーザー環境での変更が、他のユーザーに影響を与えるべきではありません。たとえば、ユーザー「A」が電子メール･クライアントのデフォルト･ビューを変更するとき、他の全てのユーザーのためのデフォルト･ビューを、同じように変更すべきではありません。この問題を改善するには、権限を用いたプログラム変更の防止や、ユーザー･プロファイルの強制的な作成、ベンダーによる問題の解決などが含まれます。
* アプリケーションはビデオ依存型か? Terminal Services サーバーからのビデオ･ストリームはクライアント上にキャッシュされるため、クライアントがリフレッシュする変更点だけを、サーバーは送信する必要があります。スクリーン･アニメーションを伴うアプリケーションや、相当量のスクリーン更新を頻繁に送信するプロセスなどは、サーバーとネットワークに大きな負荷をかけ、アプリケーションのスケーラビリティを極端に低減させます。 アプリケーションから配信されるビデオが、アプリケーションが用いるグラフィック･レベルに対して、Low、Medium、High のいずれになるのか、Application Analysis 補足資料に記録します。スクリーン更新がほとんど無い Notepad のようなアプリケーションは、グラフィックへの依存が Low になると考えられるのに対して、いくつかのアニメーションやエンベッドされたビデオを取り込むアプリケーションは、グラフィック依存が High になるでしょう。グラフィック依存が High のアプリケーションは、サーバー･サイドのプロセッサとメモリを大量に消費する一方で、スクリーンを頻繁にアップデートするために、大量のネットワーク帯域幅も消費します。
* アプリケーションはオーディオ依存型か? オーディオも、ネットワーク帯域幅を大量に消費します。それらの評価において、少量のシステム･サウンドは Low となり、また、低帯域幅のオーディオを頻繁に配信する場合は Medium となり、Media Player などのオーディオ依存型のアプリケーションや、コンピュータ･ベースのトレーニングやボイス･インストラクションなどは High となります。
* 最適化されていないアプリケーションのコードを、32-bit or 64-bit アーキテクチャの、いずれで取り扱うのか? たとえば 16-bit コードを含むプログラムは、32-bit あるいは 64-bit コードで記述されたプログラムと比較して、不釣り合いな CUP とメモリのリソースを消費します。そのため、そのようなアプリケーションは、別のオペレーティング･システム上で取り扱う必要があります。対策としては、16-bit コンポーネントを検出し、新しいバージョンに移行させるために ACT を用いるケースや、ソフトウェアを修正するケースがあります。また、アプリケーションの実行で消費されるリソースについて、受け入れが可能な場合もあります。

リストアップされた上記の要因と、その適合性にしたがって、それぞれのアプリケーションを、以下の 3つのカテゴリーに分類していきます：

* **対象となるアプリケーションは、適切な候補である。** そのアプリケーションをプロジェクトに取り込み、この後に続く各 Step での検討を継続します。
* 対象となるアプリケーションには、いくつかの問題点が存在するが、Terminal Services 上で実行可能である。その問題により、実装に影響を生じる可能性がありますが、プロジェクトへの取り込みを妨げるほどの事ではありません。アプリケーションのリスクをモニターし続ける一方で、そのアプリケーションを今回のデザインに加えます。

たとえば、対象となるアプリケーションにおいては、Terminal Services に関するベンダー･サポートが無いかもしれませんが、その実施を必要とするビジネスはクリティカルです。1つの判断は、そのアプリケーションを候補として残すことです。このプロセスの後半で、アプリケーションのスケーラビリティなどの理由により、別のリスクが生じる可能性があります。そのときには、対象となるアプリケーションに対して、別のレビューが強制されることになるでしょう。

* **対象となるアプリケーションは、この時点において** Terminal Services **に適合しない。**そのアプリケーションを、このプロジェクトのスコープから除外します。そして、アプリケーションを取り扱うための代替案を、別のプロジェクトで検討するための必要性が生じるでしょう。Terminal Services に対応していないアプリケーションが、他の仮想化テクノロジーにおける、適切な候補になるかもしれません。Infrastructure Planning and Design シリーズの Selecting the Right Virtualization Technology を参照してください：<http://www.microsoft.com/ipd>

## **意思決定の概要**

この Step 3 では、Windows Server 2008 上の Terminal Services から配信する、すべてのアプリケーションの適合性が検証され、基準にしたがい識別されました。

プロジェクト･スコープの内側に入ったアプリケーションは、Terminal Services 環境の配信における適合性に従い分類されます。そこで得られる情報を 、Appendix Bの補足資料のようなスプレッド･シートに記録します。

そして、この Step が終了するときには、分類されたリストが、ビジネスの視点からレビューされるべきです。その理由は、いくつかのアプリケーションについては、いま直ぐ Terminal Services から配信するには、その基準に適合しないと考えられないからです。それらのアプリケーションを、プロジェクト･スコープ内に残すためには、いくつかの変更が必要となるかもしれず、また、そのためのコストも伴います。そのため、作業に着手する前に、ビジネスの視点からのフィードバックが必要となります。そうすることで、追加コストの意味が理解され、また、ビジネス面での最適な判断へと到達します。

ビジネスの視点で、プロジェクト･スコープへの取り込みが望まれる、いくつかのアプリケーションについて検討する必要があります。その結果として、大掛かりな変更を施さないことには配信できないという状況や、プロジェクト自身のレビューが必要という状況に至るかもしれません。

この後に続く各 Step では、Terminal Services を介してアプリケーションを配信していくための、リソース要件を決定していきます。

## Task および検討項目

Application Compatibility Toolkit (ACT). ACT とは、スタンドアローン･コンピュータ上や、ターミナル･サーバー上でアプリケーションを実行する際の情報を、収集し評価することが可能なツールのグループです。 その中の、いくつかのツールは、Terminal Services 上で実行するアプリケーションの評価に最適です：<http://technet.microsoft.com/en-us/desktopdeployment/bb414773.aspx>

* Setup Analysis Tool (SAT). 設定を監視し、インストレーション･カーネル･モード･ドライバと16-bit コンポーネントを見つけ出します。
* Inventory Collector. 組織内にインストールされたアプリケーションを識別します。
* Compatibility Reporting. 収集されたデータの蓄積と評価を支援します。
* Find compatibility information from the community. 互換性の問題と修正に関する、共通のリポジトリです。

## **参考資料**

* Application Compatibility Toolkit 5.0 サイトを参照してください：<http://technet.microsoft.com/en-us/windowsvista/aa905102.aspx>
* 以下のURLのApplication Compatibilityを参照してください：<http://technet.microsoft.com/en-us/windowsvista/aa905066.aspx>
* 以下のURLのDevice Driver INF Changes for Plug and Play Device Redirection on Terminal Serverを参照してください：<http://www.microsoft.com/whdc/driver/install/TS_redirect.mspx>
* IPD のSelecting the Right Virtualization Technology を参照してください：<http://www.microsoft.com/ipd>.
* IPD のMicrosoft App-V を参照してください：<http://www.microsoft.com/ipd>.

# Step 4: **ユーザーの分類**

この Step 4 が始まる時点において、ユーザー･リストだけではなく、Terminal Services でホストされるアプリケーションの検証も済んでいます。Terminal Services 環境のサイズを最適化するためには、ユーザーがもたらす負荷について、理解する必要があります。

Terminal Services の実装においては、エンドユーザーがアプリケーションを実行する方式に応じて、大幅な変更が生じるでしょう。その アプリケーションは、ユーザー固有のローカル･ワークステーションではなく、リモート･サーバー上の共有環境で実行されます。このような変更を実現していく計画において、必要とされるリソースを正確な計算できると期待することは、非現実的なことです。それに代えて、それらのリソースを、可能な限り詳細に見積もることから始めます。そして、想像し得る利用形態において、いかなるリソースの上昇が生じても、それを埋め合わせることが可能なように、若干の予備的なキャパシティをシステムに加えるべきです。

そのためには、対象となるアプリケーションの一般的な用法に基づき、Heavy、Normal、Light というカテゴリーのいずれかにユーザーを配置していきます。

この分類は、ターミナル･サーバー･ファームのサイズを決定するための、インプットとして Step 7 で使用されます。そうすることで、それぞれアプリケーションにおける、ユーザー負荷が決定されるでしょう。

それぞれのユーザーあるいはユーザー･グループを、いずれかのカテゴリーに割り当てます。なぜなら、カテゴリーは、すべてのアプリケーションで等しくなると思われる、振ユーザーの振る舞いを反映するものになるからです。

ユーザーを、以下のカテゴリーに割り当てます：

* Heavy ユーザー. 複数のアプリケーションでの作業に一日の大半を費やし、同時に2～3種類のアプリケーションを使うこともあります。ソフトウェアの先進的な機能を利用し、また、新しい機能を素早く見つけ出し、それらを採用する能力を持ちます。この種のユーザーは、デベロッパーや、エンジニア、グラフィック･アーティスト、リサーチ･アシスタント、プロジェクト･マネージャーに多く見られます。多くの場合に、彼らが用いるアプリケーションは、画面の頻繁な更新やアニメーションを含み、また、高解像度のグラフィックスを配信します。 さらには、アプリケーションからアプリケーションへと、大容量のマルチメディア･ファイルがコピーされるかもしれません。アプリケーションのログオンとログオフは、一日に何度も行われ、その作業は長時間におよびます。それらのユーザーは多くの場合、アプリケーションでの作業の進捗につれて、ファイルをセーブする可能性が高いと思われます。
* Normal ユーザー. コンピュータを頻繁に使用しても、それ以外の、コンピュータを伴わない仕事も同時にこなします。こうした標準的なユーザーは、先進的な機能の存在を知っていても、新しいタスクや稀なタスクを実施するとき、多くのケースで誰かのアシスタントを必要とします。 時々、異なるアプリケーション間で、データ交換も行うでしょう。彼らの職種には、秘書や、販売員、医者、プロデューサーなどが含まれるでしょう。コンピュータから離れ、他の作業を行う場合、おそらくログオフはしないでしょう。それどころか、アプリケーションは実行されたままで、放置されるでしょう。
* Light ユーザー.コンピュータの電源はオンになっていても、1時間あたり数分だけ使用します。仕事を進めるために必要なアプリケーションは1種類だけであり、おそらく、他のアプリケーションは必要とされません。ログオフしない状況において、アプリケーションは何日間も、実行されたままに置かれるでしょう。パン屋や、医療ボランティア、企業のフィットネス専門家、レジ係、電気技師などが該当します。

## **意思決定の概要**

この Step 4 が完了する時点において、ホストされるアプリケーションの使用頻度にしたがって、ユーザーはカテゴライズされます。アプリケーションの用法に基づき、それぞれのユーザーに Heavy、Normal、Light というマークが付けられます。このカテゴリーと、その中に分類されるユーザー数が、 ターミナル･サーバー･ファームのサイズを決定するために、Step 7 でのインプットとして使用されます。その結果として、それぞれのアプリケーションにおける、ユーザー負荷が決定されるでしょう。この後に続く各 Step では、アプリケーションを供給するために必要な、ターミナル･サーバーのファーム数を決定していきます。

# Step 5: **ターミナル･サーバー･ファーム数を決定する**

ターミナル･サーバー･ファームとは、同一のアプリケーション･セットを配信する、ターミナル･サーバーのグループのことです。ユーザーがファーム内のターミナル･サーバーに接続するとき、そのリクエストは TS Session Broker に受け渡されます。そして、ファーム内で最も負荷のかかっていないサーバーにセッション･リクエストをリダイレクトするために、ロード･バランシングのアルゴリズムが用いられます。

Note Windows Server 2008 Terminal Services ファーム内に、Windows Server 2003 用の Terminal Services を実行するサーバーを含むことが可能です。しかし、このようなバージンが混在するファームでは、TS Session Broker ロード･バランシングの利用が不可能となります。そのため、Windows Server 2003 を必要とするアプリケーションが存在する場合には、それらを、分離されたファームに配置することを推奨します。

この Step 5 の目的は、Terminal Services デザインにおいて最適な、ターミナル･サーバー･ファーム数を導き出すことです。この Step が完了する時点において、ターミナル･サーバー･ファーム数と、そのロケーションが補足資料に記録されます。この情報を用いることで、その他の Step におけるサイズとキャパシティの考察を、進めていくことが可能になります。

## Task 1: **ターミナル･サーバー･ファーム数を決定する**

ベスト･プラクティスとして示唆すべきは、シングル･ファームでデザインを開始し、その後に、必要に応じてファームを加えていくという方式です。 追加のターミナル･サーバー･ファームが要求されると思われる条件について、以下にリスト･アップします。必要とされるファーム数を検討するために、このリストを使用します：

* WANスピードに応じて、既存のファームからクライアントを分離する。ファームにとって受け入れが可能なリンク数よりも、ターミナル･サーバー･ファームにアクセスするユーザーのほうが多い場合には、追加のターミナル･サーバー･ファームをリモート･ロケーションに配置すべきかもかもしれません。このようなケースにおいては、リモートに配置されるデータベースのような、バックエンド･サービスへのリンクに対して、対象となるファームが充分な帯域幅を持っていることを確認すべきです。代替案として、対象となるファームとリモート･クライアントの間で、リンクの帯域幅を増やす方式もあります。他のファームでインスタンスを作る方式と、帯域幅を増やす方式の、いずれが効果的になのか、その点について検討すべきです。
* トラベリング･ユーザー. この種のユーザーを受け入れるために、ファームの追加と帯域幅のアップグレードの、いずれが要求されるのかという視点で判断すべきです。重要なグループに属するユーザーが移動する先の、別のリモート･ロケーションで期待されるサービス･レベルを充たすかたちで、アプリケーションの配信を保証するための Step が必要とされるでしょう。その場合には、そのリモート･ロケーションにおいて、追加ファームのインスタンスが作られるかもしれません。また、トラベリング･ユーザーを受け入れるために、そのロケーションに対するバンド幅が引き上げられるかもしれません。たとえば、United States へ頻繁に旅行するユーザーのために、England の London というロケーションが、ファーム･ホストの役を務めるかもしれません。
* 別バージョンのソフトウェアやDLLなどが必要な場合. ソフトウェアや DLL などに関して、複数のバージョンへのアクセスが必要とされ、さらに別バージョンの共存が不可能な場合には、それらのバージョンを、個別のターミナル･サーバー･ファーム上で実行する必要がでてくるでしょう。ただし、ターミナル･サーバー上でMicrosoft App-V の仮想環境を用いて、それぞれのアプリケーションを実行することで、この課題を克服できるかもしれません。このケースにおいては、Microsoft App-V による配信をサポートする必要があり、また、Microsoft App-V 環境のインスタンスが必要となります。
* いくつかのクライアントにおけるセキュリティ上の制約｡ クライアントにおける制約により、セキュリティ･レベルに応じて分離された、複数のアプリケーション･スイートが実装がされるかもしれません。たとえば、ターミナル･サーバーのアクセス制限により、Network Level Authentication のような最新のセキュリティ機能をサポートするクライアントだけに配信が可能な、セキュリティ･レベルの実装を要求するアプリケーションがあるかもしれません。そのようなセキュリティ機能は、RDC クライアントのレベルと、オペレーティング･システムのレベルに依存する、サーバー／クライアント接続において利用可能となります。そのため、高レベルのセキュリティを必要とするアプリケーションは、分離されたターミナル･サーバー･ファームで、インスタンスを作成する必要があるかもしれません。RDC の初期バージョンを実行するクライアントや、低レベルのオペレーティング･システムでは、それらのファームに接続することが不可能です。
それらのアプリケーションを使用する、ユーザーに与えられたセキュリティ･レベルと、必要とされるセキュリティ･レベルとを比較します。それぞれのクライアントが、セキュリティ要件を満たすことが可能なのでしょうか? もし不可能な場合には、ユーザーの RDC クライアントのアップグレード、もしくは、セキュリティ･レベルの引き下げ、さらには、アプリケーションのプロジェクトから除去や、ローカルでの実行などについて検討します。あるいは、レガシーなクライアントのために、アップグレードが完了するまで、セキュリティ･レベルの低いターミナル･サーバー･ファームを設置するという選択もあります。
* 内側と外側のユーザー･ポピュレーション. 同一のアプリケーションが、組織の内部と外部のユーザーに配信される場合、別のファームで外部ユーザーをサポートする場合があります。 たとえ、アプリケーションとデータが全く同じものであっても、ビジネス要件により分離が必要になるかもしれません。
* **特定の暗号化要件**. 特定の国や地域だけに適用される、特定レベルの暗号化要件があります。クライアントからの接続に対して、特定レベルの暗号を用いて可否を判断する、ターミナル･サーバーの高度なコンフィグレーションが可能です。いくつかのアプリケーションへのアクセスを、レガシー･クライアントが要求する場合には、この最新の設定を利用するために、ターミナル･サーバー･ファームの分離が必要とされるかもしれません。
* ビジネス･グループの分離いう要件. セキュリティ上の理由により、会計部門のサーバーに、他部門のユーザーがログオンすることは望まれないでしょう。また、研究開発部門では、大半のクライアントで高レベルのセキュリティ･ポリシーが要求され、それらのクライアントは、別のファームへの接続を許可されないでしょう。
* 法的な視点からのファーム分離の要求. 国家機密および、プライバシーの問題、法的な要件などにより、特定データの分離が要求され、また、国境を越えた特定データへのアクセスが抑制されるでしょう。それらのアクセスが必要な場合には、要求に対応するためのファームを設置します。

単一のターミナル･サーバー･ファームからプロジェクトを開始して、上記の事柄が要求されるにつれて、デザインにファームを追加していきます。

## **意思決定の概要**

この Step 5 が完了する時点で、ターミナル･サーバー･ファーム数と、そのロケーションが決定されます。ファーム数を記録し、Appendix のFarm Design のような補足資料を作成します。 その中に、それぞれのファームを追加する、必要性についても記録します。

次の Step 6 では、それらのファームに、アプリケーションを割り当てます。それにより、それぞれのファーム･サイズが計算できるようになります。

# Step 6: **アプリケーションとユーザーを、ファームにマップする**

この Step 6 の目的は、アプリケーションとユーザーを適切なターミナル･サーバー･ファームに割り当て、それを補足資料に記録することです。すでに、ターミナル･サーバー･ファームの数が決定しているため、そこにユーザーとアプリケーションを割り当てるときに、それぞれのファーム自身も補足資料に配置できるでしょう。 また、それらのファームが多数のロケーションに配置されている場合には、サンプルとして Appendix C で提供されるスプレッドシートによる補足資料に加えて、地図情報を背景としてファームを表現する方式がが有用になるかもしれません。既存のネットワーク･インフラストラクチャ図などを、背景として有効に利用できます。

## Task 1: **ユーザーとアプリケーションを、適切なテーミナル･サーバー･ファームにマップする**

ユーザーおよび、ファーム、アプリケーションなどの量的な要因は、その全てが管理上の混乱を引き起こす存在となり得ます。この Step では、それぞれのファーム上のユーザーたちのためにホストされるアプリケーションを、記録するために補足資料を用います：

1. Appendix C を補足資料のサンプルとして使い、ファームをアプリケーションをマップします。 そのときに、相互依存するアプリケーション同士が、同じファーム内に集約されるように配慮すべきです。
2. それぞれのアプリケーションにアクセスするユーザーを、それぞれのファームにマップし、そのことを補足資料に記録します。そして、ファーム間を渡り歩くユーザーにとって固有のアプリケーションも、同時に記録します。
3. 相当量のユーザーがファーム間を渡り歩く場合には、その対象となるロケーションの補足資料に、それらのユーザーを記述します。

## **意思決定の概要**

この Step 6 における記述は、それほど難しいことではありませんが、クリティカルで時間のかかる Task となります。それは、適切とされる容量と性能の要件を、次の Step 7で正確に分析するための重要な Task でもあります。この Step が完了すると、ユーザーとアプリケーションは、最も適切なファームにマップされます。そのためにマッピングも、ファーム･デザインのための補足資料に記録します。こうした情報は、それぞれのターミナル･サーバー･ファームが必要とするリソースを決定するために、次の Step 7 で利用されます。それにより、接続されたユーザーへの、適切な対応が実現されていきます。

# Step 7: **ファームのデザイン**

ファーム内のターミナル･サーバー群は、同一のアプリケーションを配信するという役割を持ち、また、同じ方式でコンフィグレーションする必要があります。そうすることで、ユーザーは、どのターミナル･サーバーにが接続しても、同一のエクスペリエンスが得ることが可能になります。

それぞれのターミナル･サーバー･ファームには、アプリケーションとユーザーに対する固有の設定が含まれます。ただし、アプリケーションを利用するユーザーの振舞いは、きわめて多様なものになり得ます。こうした理由により、それぞれのターミナル･サーバー･ファームに対して、キャパシティに関する計画とテストが必要になります。

一連のアプリケーションは、クライアント･コンピュータではなく、ターミナル･サーバー上で稼動するようになります。そして、それらを移行するプロセスには数多くの不確定要素が存在するため、ここでは、合理的なキャパシティ見積もりの達成を目標とし、モデリングや過度の精度に依存しないようにします。最初のユーザー･セットを用いて学習を進め、その結果を記録します。続いて、次のユーザー･セットを用いて、学習結果を加えていくための調整を行います。

この Step 7 では、それぞれのファーム内のサーバーに対する構成要素を決めていきます。それにより、アプリケーションを配信していくために必要な、ファーム内のサーバー数も決定されます。さらに、そこに加えていくものとして、Terminal Services Web Access や、フォールト･トレランス、ロード･バランシング、メンテナンスのための、追加サーバーが指定されるかもしれません。

それぞれのファームに対して、以下の各 Task と実施して、その結果を、Appendix C におけるサンプルのような補足資料に記録していきます。

## Task 1: **サーバーの構成要素を選択する**

この Task 1 のゴールは、ターミナル･サーバーを展開するための、最適なタイプのハードウェアを決定することです。

このガイドにおける構成要素とは、対象となるサーバーの複合的な特質のことであり、以下の項目を含みます：

* プロセッサのアーキテクチャ（32 bit と 64 bit）
* CPU のコア数と速度
* 実装されたメモリの総量
* ディスク･ストレージの容量と、ディスク･サブ･システムのデザイン
* コンフィグレーションされたネットワーク･ポートの数

これらの構成要素の選択においては、Windows Server 2008 を運用するというレベルを、最低限の要件としてスタートします。まず、以下の選択肢から、採用するアプローチを選びます：

* **既存のハードウェアを使用する**. コンフィグレーションと配置をし直せば、Terminal Services を介したアプリケーションの配信に利用できる、サーバー･ハードウェア･リソースが組織内に存在するかもしれません。既存マシンを利用するという方式の欠点は、対象となるハードウェア･コンフィグレーションが、ターミナル･サーバー･ファームのための「理想的」なコンフィグレーションと一致しないという可能性です。それらのマシンは、ハードウェアの年式や、システムの仕様などにより、きわめて多様なものになり得ます。多くの場合、既存ハードウェアの利用は、全体的な実装コストを低減しますが（コスト回避）、標準化を犠牲にするという結果をもたらすかもしれません。それぞれのファームにおけるサーバー群は、同一のアプリケーションを配信しなくてはなりません。こうした理由により、サーバーは同一の構成要素を持ち、また、一貫性のあるユーザー･エクスペリエンスを提供すべきです。さらに、強力なサーバーをTS Session Broker に割り当て、ファーム内のサーバー負荷を分散していくことも可能です。 利用していくサーバー構成要素において、かなりの多様性が存在してしまう場合には、そのための書式をデザインに加えることで、ファーム内の均一性を改善することが要求されるかもしれません。ただし、それにより、実装と管理のコストが押し上げられることになるでしょう。
* **既存の構成要素に合わせて、新しいハードウェアを購入する**. このアプローチを成功させるためには、ターミナル･サーバー･ファームのための「理想的」なコンフィグレーションと、組織の標準的な構成要素の間に、共通性が存在しなくてはなりません。大量購入されたハードウェアのコンフィグレーションを用いることで、全体的な実装コストを引き下げるかもしれません（コスト回避）。ただし、理想的なコンフィグレーションとかけ離れてしまった場合には、要求されるパフォーマンスとスループットをファームから供給するために、追加のハードウェア･ユニットの購入が必要になるかもしれません。
* **ファームの要件に最適な構成要素に合わせて、新しいハードウェアを購入する**. このケースでは、Terminal Services 実装を最適化する、ハードウェア･コンフィグレーションが得られるでしょう。ただし、以前からの標準コンフィグレーションと異なるものになる場合には、ユニットごとの調達コストと、初期サポートのコストが、押し上げられるかもしれません。

新しいハードウェアを購入するとき、実質的な適合性を考えながら、構成要素を選択していくという方式は存在しません。それがベスト･ケースに該当するなら、step 7: task 2 のテストは、いくつかの構成要素をまたいで運用されるでしょう。そして、そこでのコンフィグレーションが、大半のユーザーにとって最適なものになるでしょう。 実際に、それぞれのファームにアプリケーションが割り当てられるとき、この Step 7 と定義された戦略の、レビューが必要になるかもしれません。最適なコンフィグレーションを見いだすために、構成要素の決定について反復することは、珍しいことではありません。

以下のガイドラインは、理想的なサーバーの構成要素決定するために必要な、優先順位で並べられています：

* **64-bit と 32-bit のアーキテクチャ.** 64 bit アーキテクチャでは、 カーネル･アドレス空間の制約が排除されます。この制約は、32 bit アーキテクチャのオペレーティング･システムがサポートする、ターミナル･サーバーのセッション数に影響を与えています。そのため、64 bit アーキテクチャでは、さらに多くのユーザー･セッション数をサポートできるようになるだでしょう。64 bit アーキテクチャに移行する際の最大の欠点は、より大きなメモリ容量が必要とされる点にあります。
* **大容量メモリの実装.** 多数のユーザーが同じアプリケーションを共有する Terminal Services 環境において、大容量のメモリを実装することで、パフォーマンスを大きく改善できます。その理由は、それぞれのユーザーの作業結果が、物理ディスクにスワップ･アウトされる前に、メモリ上で高速に処理されるという点にあります。同じ理由により、追加されるメモリにより、ディスク I/O に関する要求も低減されるでしょう。もし、64 bit アーキテクチャが、32 bit に替わって展開されると、大よそ で、2倍以上のメモリを、実装することになると考えるべきです。
* **少数の大容量ディスクよりも、多数の小容量ディスク.** ディスクのフォールト･トレランスに対応するために、RAID コンフィグレーションの使用を検討すべきです。ただし、パフォーマンスを高めるために注目するのは、ディスク全体の IOs （IOs per second）となります。ディスク･サブシステムの IOPS を高めるためには、少数の大容量ディスクよりも、多数の小容量ディスクを採用すべきです。このアプローチでは、さらに多くのスピンドル上に、作業負荷を分散するため、全体的なパフォーマンスが増加します。使用されるスピンドル数は、ファイル･アクセスにおけるレスポンスに大きな影響を与えます。一般的なターミナル･サーバー･システムで生じるディスク･アクティビティは、以下の3つのエリアに影響を与えます：
* システム･ファイルとアプリケーションのバイナリ･コード
* ページ･ファイル
* ユーザーのプロファイルとデータ

これらの3つのエリアは、明確に区別できるストレージ･デバイスにより、サポートするのが理想です。ライトバック最適化を実現する、バッテリー･バックアップ･キャッシュを提供するストレージ･アダプターを私用します。ライトバック･キャッシュ･サポートを持つコントローラーは、ディスクへの書き込みを同期させるために改善されています。すべてのターミナル･サーバーのユーザーが、個別のハイブを持つため、ディスク書き込みの同期は、ターミナル･サーバー･システムにおいて、きわめて一般的なものとなります。ディスク書き込みにおいて、同期オペレーションを用いることで、レジストリ･ハイブは周期的にディスクに保存されます。

* **マルチ･コアCUP の利用**. ターミナル･サーバーにおけるマルチ･ユーザー環境では、多重化されたプロセッサとコアにより、CPU の過密状態が緩和されます。

Terminal Services構成要素の詳細なガイダンスについては、この Step の最後にある“Additional Reading” セクションの、 Performance Tuning Guidelines for Windows Server 2008 を参照してください。

どのアプローチが採用されるにしても、ビジネス要件を充たすサーバーのサイズと台数を選択することが、この Step のゴールとなります。そのための要件として、可能な限り少ないマシンで実現する、セキュリティや、サービス配信、フォールト･トレランスなどが挙げられます。

Appendix C におけるサンプルのように、ファーム･デザインのための補足資料に、選択された構成要素を書き込んでいきます。そして、次の Task である、スケーリングの評価へと進みます。

## Task 2: **ファーム内で必要とされるターミナル･サーバー数を決定する**

ターミナル･サーバー･ファームにおけるサーバーの最適な台数とは、一貫性と応答性をベースとした、ユーザー･エクスペリエンスとシステム利用のバランスを備え、容量の拡張とコストの低減に応えるものになります。それぞれのファームに最適なサーバー数を、一貫性をもって導き出すためには、以下に示すいずれかの方式にしたがい、小さなファームの実装から開始します。そして、対象となるファームで期待される、パフォーマンスの達成状況を確認するために、その実行結果を測定します。こうしたエクスペリエンスから学習を進め、その結果を、次のファームを計画する際に利用し、適切な方式へと向けて調整していきます。

ターミナル･サーバー･ファームのサイズを定めるために、2つの方法が提供されています：

* 最初の方法では、フル･コンフィグレーションされたサーバーに、大量の負荷をかけるテストを用います。そのときに、heavy／normal／light の各ユーザーが、選択された構成要素を取り扱う様子を確認していきます。それぞれのテストの後に、たとえば、さらに多くのメモリを加えるような、構成要素の調整が可能となります。続いて、それらの構成要素を最適化するためのテストを再実施します。
* 2番目の方法は、クライアント･コンピュータにおける、その時点の測定結果を用いて、サーバー負荷を推定するというものです。その結果を確認するために 、選択された構成要素を持つサーバーに対して、ロード･テスト･スイートを実行すべきです。

それぞれ組織に適合するアプローチを選択し、以下の具体的な手順を実施していきます。ロード･テストのシナリオが直ちに準備できるなら、最初の方法は効果的であり、時間を短縮できるかもしれません。また、2番目の方法では、ワークステーション環境からデータを収集するという、別の作業を最初に行わなければなりません。

### サーバー･サイジング手法1: サーバー負荷を見積もる

実運用環境への導入に先立って、サーバー容量を見積もるための現実的な方法は、選択された構成要素内のサーバーに対して、アプリケーションの完全なセットとユーザーの総数を用いた負荷テストを実施することです。

Appendix D: “サーバー･パフォーマンスの分析とスケーリング” のガイドラインを用いて、負荷テスト･プロダクトを利用しながら、もしくは、現実的なテストを実施しながら、サーバー性能をモニタリングします。 Windows Server 2003 Deployment Kit は、アプリケーション･スクリプティングのサポートをするキャパシティ･プランニング･ツールとして、Roboserver（Robosrv.exe）と Roboclient（Robocli.exe）を提供します。Windows Server 2003 Deployment Kit companion CD で提供される、それらのツールを利用して、擬似的な負荷をサーバーにかけることができます。

負荷テストを実施するときには、一貫性のあるユーザー･グループを用いてサーバーに負荷をかけ、時間をかけながらグループ数を増やしていきます。そのとき、それぞれのグループは、標準的なアプリケーション･セットを実行すべきです。プロセッサや、メモリ、ディスク、NIC などの使用が、許容範囲を越えるまでシステムをモニターします。その許容範囲を超えたときに実施されていた、テストの内容から引き算を行い、ユーザーが快適に作業を進めるための基準を取得します。その引き算を行った結果としてのユーザー数が、構成要素におけるサーバーごとのユーザー数の合計となります。サーバーにおける予備的なキャパシティを確保するためには、つまり、バッファやヘッドルームとして参照が可能になる空間を得るためには、さらなるユーザー数の引き下げを、パーセンテージを用いて行うべきです。その値が、サーバー･キャパシティと、ユーザー数を対比するものとなります。 Appendix C のサンプルのように、その値を補足資料に記録します。

以下の、4つのテストを組み合わせて実施すべきです：

1. Heavy ユーザーに対して、ユーザーごとのコストを明確にする。
2. Normal ユーザーに対して、ユーザーごとのコストを明確にする。
3. Lightユーザーに対して、ユーザーごとのコストを明確にする。
4. 上記のユーザー混在環境については、Heavy、Normal、Light ユーザーの割合から予測値を引き出す。

それぞれのテストにおいて、ユーザーを追加し、パフォーマンス面で受容可能な閾値に達するまで、システムをモニタリングします。そして、サーバーにおいて保持が可能な、それぞれのタイプのユーザー数と、シングル･ユーザーごとのユニット･コストを、補足資料（Appendix C）に記録します。混成ユーザーについてテストを完了するには、予測される Heavy、Normal、Light ユーザーの組み合わせに適合すると思われる、最終テストの結果を確かめます。

このタイミングで、ログオン時の負荷をテストを行います。ログオンが集中するピーク時を想定して、そのときにログオンする思われるユーザーのパーセンテージと、サーバーごとのユーザー数を掛け算します。その計算結果である、同時ログオン数を示す値を用いて、サーバーに負荷をかけていきます。プロセッサ利用は、100%に達するはずなので、このログオン時の値は重要な指標になります。ログオン時の結果が SLA に合致しない場合には、サーバーごとのユーザー数（*users per server*）を減らして、再テストを実施しなくてはなりません。

このようにして、サーバーごとのユーザー数（*users per server*）を確定します。続いて、ピーク時にターミナル･サーバー･ファームに接続されるユーザー数の合計（*total number of users*）を、サーバーごとのユーザー数（*users per server*）により割り算することで、ファーム内のサーバー数を導き出すことができます。

ピーク時に接続されるユーザー数 ／ サーバーごとのユーザー数＝ファーム内のサーバー数

この値を、補足資料（Appendix C）に記録します。そして、すでに明らかになっている、それぞれのターミナル･サーバー･ファームに対して、このプロセスを繰り返します。

TS Web Access ロール･サービスで要求される、サーバー数を決定するためには、Task 3 へと進みます。

### サーバー･サイジング手法 2: クライアント･ベースの計算

サイジングのための2番目のアプローチは、既存のクライアント･コンピュータにおけるリソース利用をモニタリングします。そして、その結果を推定し、サーバー･システムへの適用を試みます。

Windows Performance Monitor と Windows Task Manager を用いて、クライアント･コンピュータ上でアプリケーションが消費するリソースを測定します。その値を用いて、一連のアプリケーションが、既存マシン上で実行される状況を想定します。 サーバー･ファーム上に集約された、すべてのアプリケーションについて、このデータを収集します。その後に、Terminal Services ユーザー･エクスペリエンスを充たすために必要な、メモリや、プロセッサ、ディスク IO、ネットワークなどの、全体的なリソース量を決定するために、このデータを利用できるようになります。

理想的には、Terminal Services セッションでアプリケーションを実行する、実際の Windows Server 2008 システム上で、これらのテストを実行すべきです。しかし、情報の収集や見積の作成を目的として、スタンドアローンのユーザー･ワークステーションの使用も可能です。こうした負荷を処理するために必要な、リソースに基づいたキャパシティ計画において、最も重要なパフォーマンス･カウンタを、Appendix D: “Server Performance Analyzing and Scaling”で解説しています。アプリケーションの負荷を測定するためには、以下のカウンタを用いて収集する情報を、記録する必要があります。それにより、必要とされるキャパシティが推定できるようになります：

* Processor usage. *% Processor Time* カウンタは、その時点で使用されている、プロセッサの容量を示します。
* Memory usage. *Memory\Available Mbytes* カウンタは、利用可能なメモリ残量を示します。
* Disk.  *IO per second* (IOPS). 通常においては、ディスクの空き容量が関心事になるべきではありません。 ただし、アプリケーションの実行により、この値が大幅に変化するなら、その値は重要になります。
* Network. ***Bytes Total/sec.***RDC クライアントからアクセスするターミナル･サーバー上で、テストが行われる場合に重要な値となります。

以下のセクションにリストアップされる、それぞれの条件に基づく基準を取得するために、Appendix D のインストラクションを用います。 Step 3 でリスト･アップした、それぞれのアプリケーションについて、上記の情報を記録します。

さらに、ターミナル･サーバーに対するリモート･デスクトップ･セッションを、アプリケーションとして取り扱い、同じステップを実施していきます。そのときのスクリーン情報が、Remote Desktop Session であっても、アプリケーションであっても、基本的になにも変わりません。つまり、それぞれのスクリーン更新のために、同量のデータが送信されるという意味では、リモート･デスクトップ･セッションと、Terminal Services 上に配信されるアプリケーションは、機能的に等しいことになります。 さらに、配信されたアプリケーションの測定が可能なように、残されたリソースの測定も可能です：

* Baseline. システムにおけるプロセッサとメモリの平均値、そして、ディスク･アクセスとネットワーク･アクティビティの平均値を、比較のために集約します。
* Initial logon cost. セッションを開始するために、ユーザーがサーバーに接続するとき、リソースを大量に消費するログオン･プロセスが存在します。ターミナル･サーバー･セッションを開始するために、ユーザーがサーバーに加えていく値を測定します。さらに、ログオン時刻を記録し、Step 8 におけるピーク･ログオン･タイムに関するプランニングを可能にします。
* Startup cost. システムのメモリや、プロセッサ、ディスク･アクセス、ネットワーク･アクティビティにおいて、アプリケーションによるリソース消費を開始する、ピーク値のレベルを測定します。リソースの消費量が、スタート･レベルに留まる状況、あるいは増加したり低下する状況をを測定します。
* Operating levels. アプリケーションが起動し、利用を開始する準備が整った後に、Baseline の測定値と、アプリケーション運用時に使用されるリソース量との、差分を測定します。
* Successive user startup. 2番目のユーザーがアプリケーションを起動したときの、リソースにおける変化を記録します。 その値は、最初のユーザーの場合と大きく異なるかもしれません。ターミナル･サーバー･ファームのサイズを定めるときの、要因に入れる必要があります。
* Resource release. ユーザーがアプリケーションを終了するときに、システムに戻されるリソースを測定するために、以前に名前付けされたステップを逆方向へ進めます。

それぞれのアプリケーションに対して、上記のステップを繰り返して実施し、その結果を記録します。

Note   スコープ内に異なるバージョンの RDC クライアントが存在する状況で、にターミナル･サーバーのテストを行なう場合には、それぞれのRDC クライアント･バージョンに対して、一連のテストを実施しなくてはなりません。それぞれの新しいバージョンが、性能と効率を改善しているため、このような措置が必要となります。

選択された構成要素を持つサーバーが、サポートが可能なユーザー数を、リソース使用に関する上記の情報と、Step 6で決定されたファームとユーザーのマッピング情報を用いて計算します。

上記の決定にしたがい、以下のステップを、プロセッサとメモリの使用量に対して実施します：

* 1. 最初のユーザーが開始するアプリケーション･セットの使用パーセンテージを合算し、その値を100から差し引く。
	2. 次に続くユーザーたちの、使用パーセンテージの合計値で、上記の計算結果を割り算する。

この割り算の結果に、1（最初のユーザー）を加えたものが、平常時におけるサーバーごとのユーザー数（*users per server*）における最大値となります。プロセッサと、サーバー･メモリの容量を比較して、小さい方の値をとります。

計算の事例.

|  |  |  |
| --- | --- | --- |
| 代表的なアプリケーション | 最初のユーザー  | 2番目以降のユーザー  |
|  | プロセッサ使用率 | メモリ使用率 | プロセッサ使用率 | メモリ使用率 |
| Microsoft Office Word 2007 | 2.0 | 1.2 | 1.5 | .5 |
| Adobe Acrobat 8.0 | 3.0 | 1.6 | 1.0 | .7 |
| Total | 5.0 | 2.8 | 2.5 | 1.2 |
| プロセッサ･キャパシティ = ((100 – 最初のユーザーのプロセッサ使用率) ÷ 2番目以降のユーザーのプロセッサ使用率) + 1 = 39 ユーザー |
| サーバー･メモリ･キャパシティ = ((100 - 最初のユーザーのメモリ使用率) ÷ 2番目以降のユーザーのメモリ使用率) + 1 = 82 ユーザー |
| サーバーごとのユーザー数（users per server）として、低いほうの値である 39 を選びます。 |

サーバーごとのユーザー数（users per server）の最大値に、同時ログオン数が接近するにつれて、ログオン時間が増加することに気づくはずです。たとえば、コールセンターや会計事務所のように、毎朝の同じ時間帯に人々がログオンするケースでは、それを補うためにユーザー数（users per server）の値を 10% 程度を低減すべきです。以下のように、このときのユーザー数を、サーバーごとのユーザー数（users per server）で割り算します。その計算結果として得られるのは、対象となるファームに割り当てられ、また、ある時間帯にアクティブとなる、ファーム内で要求されるサーバー数を導くために必要とされる値です。

Active users on farm / users per server = number of servers required in the farm

ファーム内のアクティブなユーザー数 / サーバーごとのユーザー数（users per server）= ファーム内で必要とされるサーバー数

この値を、補足資料 Appendix （C） に記録します。

## Task 3: **フォールト･トレランスで必要とされる、追加のサーバー数を決定する**

前述のTask 1 と 2 で決定されたサーバーごとのユーザー数（users per server）では、まだ、フォールト･トレランスを計算が行われていません。したがって、フォールト･トレランス実装するためには、そのために追加するサーバー数を決定する必要があります。

ターミナル ･サーバーを、Microsoft Cluster Service （MSCS）内に配置することはできません。そのため、フォールト･トレランスが必要な場合には、追加のサーバーを用いたロード･バランシングのソリューションを提供しなくてはなりません。仮に、サーバーがダウンした場合には、そのサーバーを試用しているユーザーは他のサーバーを初期化し、セッションを再開する必要があります。 インカミング･セッションのリクエストは、TS Session Broker により、他のサーバーとロード･バランシングされます。このケースにおけるユーザーは、セッションの中断を経験するでしょう。

このようなダウン･タイムにおいては、他のサーバーは新たな負荷を強いられ、また、全体的なサーバー容量は低減します。したがって、不測の事態やメンテナンスにおけるダウン･タイムに対処する容量レベルを、維持するための予備サーバーについて計画すべきです。

フォールト･トレランスの要件を充たすために必要な、追加のサーバー数を決定して、その値を補足資料 Appendix（C）に記録します。

## Task 4: TS Web Access**で必要とされる、追加のサーバー数を決定する**

TS Web Access は、単なる Terminal Services ロール･サービスであり、ファームを横断して共有することはできません。その理由により、ここでは、TS Web Access をファームの一部として考えていきます。

Note    TS Web Access は、Windows Server 2003 Terminal Services 実装と共有できない。

TS Web Access ロール･サービスを利用することで、ユーザーがブラウザを介してアクセスする Web サイト上に、アプリケーションを受け渡すことが可能になります。ただし、ユーザーがWeb上のメニューを介して、アプリケーションを起動するときには、ターミナル･サーバーがダイレクトに接続されています。つまり、そのときには、TS Web Access は既に関与していないことになります。

このドキュメントを記述している段階では、 TS Web Access ロールのサーバー･サイズ要件に利用可能な、正式なプロダクト･ガイダンスが存在していません。TS Web Access はライトウェイトなアプリケーションであり、Internet Information Services（IIS）アプリケーションをサポートする、あらゆるハードウェア･コンフィグレーションにおいてホストすることが可能です。少数のサーバーで構成されるターミナル･サーバー･ファームでは、TS Web Access がホストするIIS Web サーバー･ファームが、他の処理も受け持つかもしれません。したがって、TS Web Access アプリケーションの、既存のサーバーへのインストールは、慎重に行われるべきです。そして、専用ハードウェアの必要性について、その有無を判断するための、システム用法に関するモニタリングを行うべきです。

TS Web Access ロール･サービスの、フォールト･トレランスも可能です。その際には、Network Load Balancing あるいはハードウェア･ロード･バランシングのソリューションを用いて、追加のサーバー上にロール･サービスを複写し、それらの負荷分散を行います。TS Web Access ロール･サービスが利用できない場合、そのメニューを用いるユーザーは、アプリケーションにアクセスできなくなるでしょう。ただし、それらのユーザーが、別の方法でアクセスできる場合には、ユーザーの利便性という観点から、TS Web Access ロール･サービスで高可用性を実現するためのコストと、ビジネス･メリットを比較すべきです。たとえば、顧客などの重要な外部ユーザーにとって、TS Web Access がアプリケーションにアクセスする唯一の手段である場合には、高可用性が必須になるかもしれません。

なお、内部と外部のユーザーにアプリケーション･メニューを手供するために、TS Web Access を使うケースもあります。その際には、セキュリティの観点から、TS Web Access を用いる内部と外部のユーザーを、別々のシステム上にホストする必要性について考えるべきです。ホストされる TS Web Access ロール･サービスについて、それぞれのファームごとに、理由と、場所と、数を、Appendix C に記録すべきです。

## **意思決定の概要**

この Step 7 で、ファームのデザインは完了します。ファームにおけるターミナル･サーバーの構成要素が決定され、また、サーバーサイズが定められ、また、フォールト･トレランスのアプローチも決定されます。さらに、TS Web Access ロール･サービスが使う場合には、それをファームの一部として取り込むためのデザインも行います。

次の Step 8 へ進む前に、実装していく他のファームについても、一連の Task を繰り返されなくてはなりません。次の Step では、ユーザーのプロファイルとデータに関するストレージについて、いくつかの項目を決定していきます。

## **参考資料**

* 以下のURLの Performance Tuning Guidelines for Windows Server 2008 を参照してください：<http://www.microsoft.com/whdc/system/sysperf/Perf_tun_srv.mspx>.
* 以下のURLの Windows Server 2008 Terminal Services RemoteApp Step-by-Step Guide を参照してください： <http://go.microsoft.com/fwlink/?LinkID=84895>.

# Step 8: **ユーザー･データをストアする場所を決定する**

Terminal Services ユーザーは、Windows Server 2008 の標準的なユーザー･プロファイルと、Terminal Services プロファイルを持つことになるでしょう。それにより、サーバー･オペレーティング･システムと、ターミナル･サーバーにログオンする際に、異なるセッティングを保持することが可能になります。ローミング･ユーザーについては、フォールト･トレラントなファイル･サーバー上に、それらのプロファイルがストアされるべきです。 ユーザーがログオフするときには、ターミナル･サーバーのハード･ドライブやレジストリから、プロファイル･データを削除すべきです。それにより、ターミナル･サーバーはクリーンな状態に保たれます。そして、それらのユーザーがファーム内の他のサーバーに接続する場合に、ユーザー自身のプロファイルを受け取ることが保証されます。

ユーザーが複数のファームにログオンするかもしれない場合には、それぞれのファームに適合した個々の Terminal Services プロファイルが必要になるでしょう。たとえば、Farm1 がファーム名の場合には、[\\fileserver\share\Farm1\username](file:///%5C%5Cfileserver%5Cshare%5CFarm1%5Cusername) のようなパスにより、それらのプロファイルをファイル･サーバ上にストアできます。また、Farm2 や Farm3 などのプロファイルについても、同じようなパスになるでしょう。それらパスは、ユーザー･アカウント内で [\\fileserver\share\%FarmName%\username](file:///%5C%5Cfileserver%5Cshare%5C%25FarmName%25%5Cusername) として設定され、また、それぞれのファーム上の %FarmName% に対して環境変数が作成されるでしょう。特定のターミナル･サーバー･ファームにログオンする際に、Terminal Services プロファイル･パス内で、それらの環境変数は Farm2 などの実際の名前に自動的に置き換えられ、正しいプロファイルとして利用できるようになります。

一般的に、ユーザー･プロファイルは極めてコンパクトなものになりますが、ユーザー･データについては、相当量のストレージが必要になる場合もあります。ユーザー･データ（My Documents やApplication Data）を、プロファイルから分離すべきです。そして、ユーザーが移動する場合には、それらのプロファイルを、複写が可能な別個のネットワーク･ドライブ上にストアします。

ユーザーのプロファイルとデータを、ターミナル･サーバー上にストアすべきではありません。もし、特定のターミナル･サーバーにストアした状態で、対象となるサーバーが利用できなくなった場合には、ファーム内の別サーバーで新しいセッションが確立できても、プロファイルとユーザー･データの利用が不可能になるでしょう。

この Step 8 の目的は、 ユーザーのデータとプロファイルをストアする場所を決定し、ファーム内の全てのサーバーから、それらを常に利用できるようにすることです。

## Task 1: **ユーザー･プロファイルのポリシーとストレージ･ロケーションを決定する**

ユーザーストレージについて、以下の項目を決定して、Appendix C の補足資料に記録します。

* ユーザーは強制的なプロファイルを持つのか? 個別のユーザーやユーザー･グループを、強制的なプロファイルに割り当てられることが可能です。それらは、マルチ･ユーザーのコンピュータやkioskにおいて、きわめて有用なものとなります。それぞれのユーザー･グループに、複数の強制的なプロファイルを割り当てることができます。たとえば、ひとつの強制的なプロファイルを kiosk ユーザーに割り当て、別の強制的プロファイルをコールセンター要員に割り当てられることなどが可能です。強制的なプロファイルは、ローミング･ユーザー･プロファイルよりも管理が簡単ですが、セッションの間にプロファイルに加えられた変更は、ログオフ時に失われてしまいます。
* ユーザー･プロファイルのために必要とされるスペースを決定する. 強制的なプロファイルが選択される場合には、それぞれの強制的プロファイルのサイズが測定し、それらの合計を得ます。それらのプロファイルは、おそれく1MB 以下のサイズになるでしょう。そして、定義された強制的プロファイルごとに、ひとつの物理的なユーザー･プロファイルだけがストアされます。もし、ユーザーが Terminal Services ローミング・プロファイルを持つなら、そのプロファイルが使用する最大サイズを決定します。
* ユーザー･プロファイルをストアする場所を決定する.このストレージのロケーションは、ターミナル･サーバーから容易にアクセスできる必要があります。その理由は、クライアント･コンピュータでなく、ターミナル･サーバーが、プロファイルにアクセスする点にあります。このプロファイルは、Storage Area Network（SAN）上の、あらゆるファイル･サーバーに配置できます。

ユーザー･プロファイルがストアされるロケーションと、そのサイズを、補足資料（Appendix C）に記録します。

## Task 2: **ユーザー･データ･ポリシーとストレージ･ロケーションを決定する**

ユーザー･ストレージについて以下の項目を決定し、それらを Appendix C 補足資料に記録します。

* ユーザー･データのために必要なスペースを決定する. ストレージ･サーバーの My Documents フォルダに、ユーザー･データをストアすることが許可されている場合には、そのストレージ･スペースが大きくなっても問題は生じません。それぞれのユーザーに許可される、データ･ストアのためにスペースを決定します。このストレージの要件は、ユーザー数に応じて増えていく総量となります。その値を、補足資料に記録します。
* ユーザー･データをストアする場所を決定します. このストレージのロケーションは、ターミナル･サーバーから容易にアクセスできる必要があります。その理由は、クライアント･コンピュータでなく、ターミナル･サーバーが、プロファイルにアクセスする点にあります。このプロファイルは、Storage Area Network（SAN）上の、あらゆるファイル･サーバーに配置できます。

ユーザー･データがストアされるロケーションと、そのサイズを、補足資料（Appendix C）に記録します。

## Task 3: **ユーザー･プロファイルとユーザー･データのためのストレージをデザインする**

この段階で、ユーザーのプロファイルとデータのための、ストレージ要件が決定されているはずです。続いて、この2つのストレージ･ロケーションについて、デザインしていきます。上記の説明のように、単体のファームや複数のファームに、ユーザーが接続する状況において、それぞれのファーム上に、別々のプロファイルとデータが配置される可能性があります。そのため、それらのファームごとに、以下の作業を完了しなければなりません。

ユーザー･プロファイルとユーザー･データにストアに関して、このソリューションがユーザーとビジネスの要件を満たすことを確認するために、以下の領域に取り組んでいきます：

* Capacity.それぞれのユーザー･アカウントごとに、ユーザー･プロファイルの最大サイズと、Terminal Services プロファイルの最大のサイズを加えます。たとえば、複数のファームにログオンするユーザーの場合には、そのファーム数を掛け算します。この段階で得られた値と、ユーザー･アカウント数を掛け算します。それにより、すべてのユーザー･プロファイルを保持するために必要な、スペースが確保されます。

((ユーザー･プロファイルの最大サイズ + Terminal Services プロファイルの最大サイズ) x ファーム数) x ユーザー･アカウント数 = すべてのユーザー･プロファイルに必要なスペース

それぞれのユーザー･アカウントで利用可能な、最大データ･ストレージ容量と、ユーザー･アカウント数を掛け算します。それにより、すべてのユーザー･データを保持するために必要なスペースが確保されます。

最大ユーザー･データ･ストレージ容量 x ユーザー･アカウント数 = すべてのユーザー･データ･ストレージに必要なスペース

* Performance. ユーザー･プロファイル･ファイルへのアクセス･スピードは、ユーザー･ログオンにおいてクリティカルな要件となります。そのため、それらのファイルは、ファイル･サーバーにアタッチされる、高速のストレージ･デバイス上にストアすべきです。

ユーザー･データ･ファイルへのアクセス･スピードは、ユーザーが認識する、アプリケーション応答速度として重要です。そのため、それらのデータは、ターミナル･サーバー･ファームに配置された、ファイル･サーバーにアタッチされる、高速のストレージ･デバイス上にストアすべきです。ストレージのデザインにおけるゴールは、ディスク･パフォーマンス要件を充たすIOPS 値を達成することです。ファイル･サーバー IOPS 要件を決定するためのテスト環境を準備した後に、データ･セントリックなアプリケーションを利用して、ユーザーが生成する IOPS 値を記録します。そして、全体的な IOPS 負荷を得るために、この値とユーザー･アカウント数を掛け算します。続いて、必要とされる IOPS 値を得るために、標準的なバッファを加えます。

* Fault tolerance. ユーザー･プロファイルと Terminal Services プロファイルへのアクセスが、ログオンにおいて必要とされます。そして、ユーザー･データへのアクセスが、大半のアプリケーション実行において必要とされるでしょう。

それらの理由により、プロファイルとデータのファイルは、複写された DFS 共有や フェイルオーバー･ クラスタなどの冗長性を必要とします。こうした、高可用性を備えたコンフィグレーションを用いて、ファイル･サーバーを配置すべきです。

フォールト･トレランスの供給と、ディスク･アレイのパフォーマンス改善において、Redundant Array of Independent Disks（RAID）を利用できます。 実運用環境のファイル･サーバーにおける共通のRAID オプションには、RAID 1（Disk Mirroring）、および、RAID 5（Disk Striping with Parity）、RAID 0+1（Mirror Stripe Sets）が含まれます。それぞれのオプションが、容量および、パフォーマンス、コストなどに関する、それぞれのポートフォリオを提供します。

このディスク容量の要件とRAID コンフィグレーションを組み合わせたものを、選択されたサブシステムのディスク･サイズにマップすることで、パフォーマンスを確立するために必要な、実際のドライブ数を決定することが可能になります。

RAID 1 や RAID 0 ＋ 1 のようなディスクの冗長性レベルの選択が、IOPS 計算に影響を与えます。この IOPS 要件を、サブシステムで選択されたディスク･タイプと、そのサブシステムにおけるドライブ特質にマップすることで、パフォーマンスにおける目標を達成するために必要な、実際のドライブの数を決定することが可能になります。

必要とされる実際のディスク数は、パフォーマンス達成のために必要とされる数と、容量のために必要とされる数の、大きい方となります。 ただし、通常では、パフォーマンスにより決定されるディスク数が選ばれるでしょう。

ファーム･デザインにおけるストレージ･デザインを、Appendix C のように文書化してから、次の Task であるスケーリングの査定に進みます。

## **意思決定の概要**

ユーザー･プロファイルとして許可されるタイプや、それらがストアされる場所、そして、必要とされるストレージ･スペースが必要などの、いくつかの決定が下されます。さらに、 同じ方式により、ユーザー･データのストレージ要件が決定されます。それにより、ユーザーのプロファイルとデータの、一貫性をもった利用が可能になます。

## Additional Reading

## **参考資料**

* 以下のURLのStep-by-Step Guide for Configuring a Two-Node File Server Failover Cluster in Windows Server 2008 を参照してください：<http://www.microsoft.com/downloads/details.aspx?FamilyID=518D870C-FA3E-4F6A-97F5-ACAF31DE6DCE&displaylang=en>.

#  Step 9: **ファームにおける** Terminal Services**のサイズと配置を考える**

Terminal Services Session Broker (TS Session Broker)および、Terminal Services Licensing (TS Licensing)、Terminal Services Gateway (TS Gateway) の各ロール･サービスについては、特定のファーム内で利用するだけではなく、マルチ･ファームで共有することも可能です。この Step 9 のゴールは、 それらのロール･サービスを最適に配置した後に、必要とされるサーバー数を決定し、そのアーキテクチャを考えることです。

Noteこれら全てのロール･サービスを、仮想マシン上で実行することが可能ですが、適切なパフォーマンスを保証するための慎重なテストが必要です。

このドキュメントを記述している時点では、TS Session Broker および、TS Licensing、TS Gateway における、ロール･サービスの容量と性能のために利用できるガイダンスが存在していません。そのため、小規模なファームにおけるロール･サービスの実装から始めて、パフォーマンスを測定し、また、結果の解析と確認を行った後に、適切なロール･サービスのサイズに調整していきます。

このような調整は、それぞれのロール･サービスにおける負荷の、実質的な見通しを得るための助けになるかもしれません。つまり、Terminal Services Web Access（TS Web Access）も含めて、ロールにおける有用なサイズを証明できる可能性があります。

|  |  |  |  |
| --- | --- | --- | --- |
| **ロール･サービスおよび機能** | **セッションごとのアクセス頻度** | **備考** | **セッションでの相対的な荷重** |
| TS Licensing | 0 | ターミナル･サーバーに、最初にアクセスするクライアントに対応。また、ライセンスの失効時にも対応。 | Light |
| TS Session Broker | 1 | ターミナル･サーバー上のセッション数を継続的にモニタリング。 | Light |
| Redirector | 1 |  | Light |
| TS Gateway | Many |  | Normal |
| TS Web Access | 1 |  | Light |

より以上の正確さが必要とされるなら、それぞれの Terminal Services ロール･サービスに対して、サーバーがサポートするユーザー数を決定するために、ロードテストを行うことが可能です。3つのロールサービスのための最初のテストでは、同じ構成要素を用います。最初のテストが完了したとき、それらのロール･サービスの共有サーバー上への配置が可能なのか、それとも不可能なのか、その点に付いて決定していきます。そして、不可能な場合には、それぞれのロール･サービスからの要求に応じた、相違を受け入れるように、構成要素を調整することになるかもしれません。

Note Windows Server 2008 Terminal Services には、Windows Server 2003 Terminal Services を実行する サーバーを含むことができます。しかし、このような混成ファームでは、TS Session Broker のロード･バランシングが使用できません。そのため、Windows Server 2003 を必要とするアプリケーションが存在する場合に派、それらを別のファームへ移動することが推奨されます。

この Step 9 が完了すると、 Terminal Services ロール･サービスの数と、それらの構成要素が決定されます。 そして、それぞれのTerminal Services ロール･サービスが、補足資料に記載されます。

最初に行うべきことは、Active Directory の実装に関する調査です。 その理由は、以下の 3つ Taskで説明する、すべての Terminal Services ロール･サービスに対して、AD が同じように関わっている点にあります。Terminal Services ロール･サービスはドメインの境界線をまたいで機能するため、それらを複数のドメインで共有することが可能かもしれません。しかし、組織的な理由により、ロール･サービスのスコープを特定のドメインに制約するという、Group Policy を用いて実装されるケースがあるかもしれません。このようなケースでは、それぞれのドメインにおいて、別個の Terminal Services ロール･サービスを初期化する必要性が出てくるでしょう。

Terminal Services ロールサービスは、クロス･フォレストのための信頼関係が確立されていない限り、Active Directory フォレストの境界線をまたいで機能することができません。そのため、適切な信頼関係が設定されていない場合には、それぞれのフォレストで必要とされる、それぞれの Terminal Services ロール･サービスの実装について計画することになります。

## Task 1: TS Session Brokers **のデザインと配置**

TS Session Broker は、インカミング･ユーザー･セッションのリクエストに対して、ロード･バランシング･サービスを提供します。TS Session Broker によりロード･バランシングされるターミナル･サーバー･ファームでは、ユーザーは最初にシングル･ターミナル･サーバーに接続されます。続いて、そのターミナル･サーバーは、最も適切なターミナル･サーバーにリクエストを転送するロード･バランシングのアルゴリズムを用いて、 TS Session Broker に接続リクエストをリダイレクトします。このアルゴリズムでは、最適なサーバーを決定するために、すでに接続されているユーザー･セッションが、対象となるファームのサーバー上で実行されていることを最初に判断します。 該当するユーザー･セッションが見つかった場合には、対象となるサーバー上のセッションにユーザーは再接続されます。また、ユーザーが既存のセッションを持たない場合には、最もセッション数が少ないファーム上のサーバーに接続されるでしょう。ファーム内のサーバーにおける相対的な「weight」を、TS Session Broker が評価することで、最適な接続バランスが達成されます。

複数のターミナル･サーバー･ファームをまたいで、TS Session Broker サーバーをの共有できますが、ユーザー･セッションの配信については、それらのファームをまたぐことができません。つまり、接続されるユーザーとは、TS Session Broker 対してリダイレクトされる、ファーム内のユーザーだけとなります。ファーム間で共有される TS Session Broker サーバーは、それぞれのファームのために独立して機能します。以下のプロセスを介して、 TS Session Broker サービスが必要とされる場所および、そこで用いられるフォーマット、そして、それぞれの TS Session Broker サーバーに接続されるファームを決定していきます。それらの情報を、補足資料（Appendix C）に書き込みます。

1. TS Session Broker ロール･サービスの必要性を判断する. 複数のサーバーを取り込んだターミナル･サーバー･ファームであっても、TS Session Broker を用いることで、切断されているセッションの再接続が可能になります。そのときのセッションは、ユーザー･アプリケーションを実行するサーバーに再接続されるため、迷子のような状況は回避されます。したがって、マルチ･サーバー･ファームでは、TS Session Broker ロール･サービスが必要とされるでしょう。

Note   TS Session Broker のリダイレクションは、RDC クライアントの 5.2 以降でサポートされます。

1. TS Session Broker に対する要求を決定する. このドキュメントを記述している段階では、TS Session Broker の容量とパフォーマンスを計画するために、利用可能なプロダクト･ガイダンスがありません。TS Session Broker はライトウェイトなサービスであり、ユーザーがファームに接続するときだけに使用されるため、それに準じた実装を計画できるはずです。小さなファームのロール･サービスの実装から始めて、パフォーマンスを測定し、結果の解析と調査を行い、それに応じたサイズに調整していきます。

さらなる正確性が必要とされる場合には、ユーザーの振舞いに合せたロード･テストにより要件を確立します。このロード･テストは、TS Session Broker がボトルネックになるように設計しなくてはなりません。それにより、充分な容量を持つターミナル･サーバーで構成される、ファームに対する処理が実行できます。

TS Session Broker のための構成要素を選択し、テスト用のプロダクトや実際のシステムによりロード･テストを行います。Roboserver（Robosrv.exe）と Roboclient（Robocli.exe）という、アプリケーション･スクリプト･サポートを取り込んだ、Terminal Services キャパシティ･プランニング･ツールが、 Windows Server 2003 Deployment Kit CD で提供されます。非接続の状態から再接続し、ファーム内のターミナル･サーバーをセッションを持つという、典型的なユーザーの行動を反映するサーバ上で、それらのツールを用いて模擬的な負荷を容易に生成し、管理することが可能です。

CPU および、メモリ、ディスク IO、ネットワークの負荷を記録するために、Appendix D のガイダンスを用いて、 TS Session Broker サーバーの性能をモニタリングします。 ひとまとめにされたユーザー･グループを持つサーバーに負荷をかけ、長い時間をかけながら負荷を増やしていきます。それぞれのグループでは、アプリケーションの一般的なセットを動かすべきです。プロセッサや、メモリ、ディスク、NIC の受容限度を超えるまで、システムをモニタリングします。そして限界を超えた直前のテスト結果から、ユーザーにとって快適な基準まで値を下げていきます。この低減されたユーザー数が、構成要素におけるサーバーごとの合計ユーザー数になります。サーバーにおける予備の容量を確保するために、バッファやヘッドルームとして引用されるパーセンテージを用いて、さらにユーザー数の引き下げについて検討します。その結果として得られる、サーバーごとのユーザー数に関するキャパシティを、補足資料（Appendix C）に記録します。

このテストでは、以下の4つの組み合わせを実行します：

* Heavy ユーザーに対して、ユーザーごとのコストを明確にする。
* Normal ユーザーに対して、ユーザーごとのコストを明確にする。
* Lightユーザーに対して、ユーザーごとのコストを明確にする。
* 混成ユーザーについては、Heavy、Normal、Light ユーザーから予測値を引き出す。

それぞれのテストにおいて、ユーザーを追加し、パフォーマンス面で受容可能な閾値に達するまで、システムをモニタリングします。そして、サーバーにおいて保持が可能な、それぞれのタイプのユーザー数と、シングル･ユーザーごとのユニット･コストを、補足資料（Appendix C）に記録します。混成ユーザーについてテストを完了するには、予測される Heavy、Normal、Light ユーザーの組み合わせに適合すると思われる、最終テストの結果を確かめます。続いて、受容できる稼働時間のレベルに至るまで、その時点のユーザー数を減らした後に、選択されたサーバーの構成要素における、TS Session Broker がサポートできるユーザー数とします。

1. 専用のリダイレクターの使用について判断する.大規模なターミナル･サーバー･ファームにおいて、セッション･リダイレクションの性能を高めるために、ターミナル･サーバーを専用のリダイレクターとしてコンフィグレーションできます。これらのサーバーは、インカミング･リクエストを処理しますが、ユーザー･セッションを受け付けることはありません。そのため、さらに高速のログオン･エクスペリエンスを、ユーザーに提供できます。

選択された構成要素における個々のサーバーで設定されたリダイレクターと、TS Session Broker を用いて、負荷テストを再実施します。このテスト結果を記録して、それぞれのサーバー上のリダイレクターによる分離が、サポート可能なユーザー･セッション数を増加する状況と、以前のテスト結果を比較します。もし、その結果が良ければ値を記録し、TS Session Broker からのリダイレクターの分離を判断すます。具体的には、サーバーの追加により生じる、コストの上昇分とメンテナンス･オーバーヘッドを比較します。

1. 要求されるサーバー数を決定する. TS Session Broker が必要とするサーバー数を決定するために、対象となるファームでセッションを持つユーザーの合計数を、TS Session Broker サーバーがサポートできるユーザー数で割り算します。このプロセスを個々のファームで完了し、対象となる値を補足資料（Appendix C）に記録します。

専用のリダイレクターを使用するという決定が下された場合には、Step 3 におけるテスト出力を用いて、専用リダイレクターのサイズを判断します。ファームに接続するユーザーの合計数を、専用リダイレクターがサポートできるユーザー数で割り算します。その結果が、専用リダイレクターが必要とする値となります。このプロセスを個々のファームで完了し、対象となる値を補足資料（Appendix C）に記録します。

1. TS Session Broker をホストするサーバーについて、共有化を検討する. この時点で、TS Session Broker サービスのためのリソース要件が、それぞれのファームに通知されます。また、TS Session Broker ロール･サービスの共有について、その可否を判断できるようになります。そのための処理は、以下の方式で完了します：
2. TS Session Broker サーバーの、ファーム間での共有が可能です。この場合、それぞれのファームにおいて、TS Session Broker サーバーに対して定められたユーザー数を合計して、それの値をファーム数で割り算します。それにより、共有される個々の TS Session Broker サーバーでサポートが可能な、ユーザー数が算出されます。

TS Session Broker が必要とするサーバー数を決定するために、ファームでセッションを持つユーザーの合計数を、TS Session Broker サーバーがサポートできるユーザー数で割り算します。このプロセスを、それぞれの「共有された」ファームで完了し、その値を記録します。

1. TS Session Broker サーバーにより、他の Terminal Services ロール･サービスをホストすることが可能です。サーバー上で組み合わされる個々のロール･サービスに対して、サーバーの構成要素でサポートできるユーザー数を決定するためには、この Step の全 Task を完了しなければなりません。その後に、組み合わされる個々のロール･サービスのユーザー数を合計して、その値を共有されたロール･サービス数で割り算します。それにより、個々のロール共有サーバー上でサポートできる、ユーザー数が算出されるでしょう。

それらのロールが必要とするサーバー数を決定するために、ファームに接続されるユーザーの合計数を、ロール･サーバーがサポートできるユーザー数で割り算します。このプロセスを、それぞれの共有サーバーで完了し、その結果として得られる値を記録します。

1. **TS Session Broker ロールを配置する、ネットワーク内の場所を決定する.** TS Session Broker ロールサービスは、自身が管理するターミナル･サーバー･ファーム内のセッションと、通信するための能力を持たなければなりません。高速 LAN 上のターミナル･サーバー･ファームは、TS Session Broker ロール･サービスを共有することが可能です。その反対に、 WAN 接続により分離されるファームでは、それぞれのロケーションに、TS Session Broker ロールサービスを割り当てられることが必要とされます。TS Session Broker ロール･サービスのインスタンスについて、その情報を補足資料に書き込みます。そして、TS Session Broker ロールサービスを必要とする、すべてのターミナル･サーバー･ファームが、高速で接続できることを確認します。
2. **TS Session Broker のフォールト･トレランス要件について判断する.** TS Session Broker サーバーは、対象となるファーム内のターミナル･サーバーに対して、ロード･バランシング機能を提供します。さらに、切断されたセッションを実行しているサーバーについても、再接続のリクエストを誘導します。もし、再接続が不可能な場合には、ファーム内のターミナル･サーバーに対して、新しいセッションが開始されます。

このような理由により、ロール･サービスのコピーである複数のインスタンスを、 MSCS クラスタ内で作成することで、TS Session Broker ロール･サービスを失敗から保護すべきです。そうすることで、セッションの再接続時にイベントで失敗が起きても、サービス提供は継続されるでしょう。MSCS クラスタから配信が可能なサービスの、実質的な必要性の有無を判断するために、ユーザーと共に SLA を調べるべきです。

以上の意思決定の結果を、補足資料に記録します。

## Task 2: Terminal Services License Server **のデザインと配置**

Terminal Services のクライアント･アクセス･ライセンス（TS CAL）には、2つのタイプがあります：

1. **TS デバイス CAL.** デバイス･ライセンス･モードで、クライアントのコンピュータやデバイスから初めてターミナル･サーバーに接続するとき、それらのクライアントはデフォルトとして、テンポラリー･ライセンスを支給されます。また、クライアントのコンピュータやデバイスから、二度目にターミナル･サーバーに接続するときには、ライセンス･サーバーが作動します。そして、必要とされる TS デバイス CAL が利用可能な場合には、対象となるライセンス･サーバーから、クライアントのコンピュータもしくはデバイスに対して、恒久的な TS デバイス CAL が支給されます。
2. **TS ユーザー CAL.** このライセンスは、クライアント･コンピュータやデバイスからターミナル･サーバーへの無制限のアクセス権利を、一人のユーザーに対して与えます。クライアントやデバイスがターミナル･サーバーに接続するとき、対象となるユーザーの Terminal Services クライアント･アクセス･ライセンス（TS CAL）の有無を判断するために、Active Directory へのコンタクトが取られます。ユーザーが TS CAL を持たない場合には、Active Directory 内のクライアントと組み合わされるライセンスを提供するために、TS Licensingロールサービスが呼び出されます。TS ユーザー CAL は、TS Licensing により実施されるわけではありません。そして、ライセインス･サーバー上にインストールされた TS ユーザー CAL 数にかかわらず、クライアントの接続が可能になります。

Noteなお、それぞれのユーザーに対する有効な TS ユーザー CAL を保持する、Microsoft Software License Terms からアドミニストレータが免除されるわけではありません。このユーザー･ライセンス･モードが使われているときに、それぞれのユーザーが TS ユーザー CAL を利用できない状況は、ライセンス･タームへの違反を意味します。

この、TS Licensing ロール･サービスについては、複数のターミナル･サーバー･ファームをまたいだ共有が可能です。そのような共有が行われる場合、対象となる TS CAL プールに基づくファームへ向けた、クライアントからの接続を実現するための、TS CAL が提供されます。

以下のプロセスを介して、ライセンス･サーバーのキャパシティを計画することで、新しい TS CAL への要求に取り組んでいきます。 続いて、TS Licensing サービスが必要とされる場所と、そのフォーマットを決定し、さらに、個々の TS Licensing サーバーと接続するファームを決定します。その結果として得られる情報を、補足資料（Appendix C）に書き込みます。

1. TS Licensingサーバーに対する要求を決定する.このドキュメントを記述している段階で、TS Licensing のキャパシティと性能を計画するえうで、利用できるガイダンスが存在しません。TS Licensingは、ユーザーがファームに接続するときにが使われるライトウェイトなサービスであり、それを前提とした実装を計画していけます。小規模なファームにロール･サービスを実装するところから始めて、パフォーマンスを測定し、その結果の解析と調査を行い、ファームのロール･サービスに応じたサイズに調整していきます。

さらなる正確性が必要とされる場合には、ユーザーの振舞いに合せたロード･テストにより要件を確立します。このロード･テストは、TS Licensing ロール・サービス がボトルネックになるように設計しなくてはなりません。それにより、充分な容量を持つターミナル･サーバーで構成される、ファームに対する処理が実行できます。

TS Licensing のための構成要素を選択し、テスト用のプロダクトや実際のシステムによりロード･テストを行います。Roboserver（Robosrv.exe）と Roboclient（Robocli.exe）という、アプリケーション･スクリプト･サポートを取り込んだ、Terminal Services キャパシティ･プランニング･ツールが、 Windows Server 2003 Deployment Kit CD で提供されます。非接続の状態から再接続し、ファーム内のターミナル･サーバーをセッションを持つという、典型的なユーザーの行動を反映するサーバ上で、それらのツールを用いて模擬的な負荷を容易に生成し、管理することが可能です。

CPU および、メモリ、ディスク IO、ネットワークの負荷を記録するために、Appendix D のガイダンスを用いて、TS Licensing サーバーの性能をモニタリングします。 ひとまとめにされたユーザー･グループを持つサーバーに負荷をかけ、長い時間をかけながら負荷を増やしていきます。それぞれのグループでは、アプリケーションの一般的なセットを動かすべきです。プロセッサや、メモリ、ディスク、NIC の受容限度を超えるまで、システムをモニタリングします。そして限界を超えた直前のテスト結果から、ユーザーにとって快適な基準まで値を下げていきます。この低減されたユーザー数が、構成要素におけるサーバーごとの合計ユーザー数になります。サーバーにおける予備の容量を確保するために、バッファやヘッドルームとして引用されるパーセンテージを用いて、さらにユーザー数のを引き下げについて検討します。その結果として得られる、サーバーごとのユーザー数に関するキャパシティを、補足資料（Appendix C）に記録します。

1. 要求されるサーバー数を決定する. TS Licensing が必要とするサーバー数を決定するために、対象となるファームでセッションを持つユーザーおよびデバイスの合計数を、TS Licensing サーバーがサポートできるユーザー数で割り算します。このプロセスを個々のファームで完了し、対象となる値を補足資料（Appendix C）に記録します。
2. TS Licensing をホストするサーバーについて、共有化を検討する. この時点で、ライセンス･サーバーのためのリソース要件が、それぞれのファームに知らされます。また、TS Licensing ロール･サービスの共有について、その可否を判断できるようになります。そのための処理は、以下の方式で完了します：
3. **TS Licensing サーバーの、ファーム間での共有が可能です。**この場合、それぞれのファームにおいて、TS Licensing サーバーに対して定められたユーザー数を合計して、それの値をファーム数で割り算します。それにより、共有される個々のTS Licensing サーバーでサポートが可能な、ユーザー数が算出されます。

TS Licensing が必要とするサーバー数を決定するために、ファームでセッションを持つユーザーの合計数を、TS Licensing サーバーがサポートできるユーザー数で割り算します。このプロセスを、それぞれの「共有された」ファームで完了し、その値を記録します。

1. **TS Licensing サーバーにより、他の Terminal Services ロール･サービスをホストすることが可能です.**  サーバー上で組み合わされる個々のロール･サービスに対して、サーバーの構成要素でサポートできるユーザー数を決定するためには、この Step の全 Task を完了しなければなりません。そして、個々のロール共有サーバー上でサポートできる、最も小さなユーザー数を取り出します。

それらのロールが必要とするサーバー数を決定するために、ファームに接続されるユーザーの合計数を、ロール共有サーバーがサポートできるユーザー数で割り算します。このプロセスを、それぞれの共有サーバーで完了し、その結果として得られる値を記録します。

TS Licensing サーバーは、 たとえばファイル･サーバなどの、Terminal Services ロール以外のサービスをホストできます。おそらく、それらのサービスは可変的な性質を持つため、この種のシナリオでは、TS Licensing ロール･サービスとホストされるサービスの、組み合わせに対する負荷テストが必要とされるでしょう。上記の Step 2 のように進め、続いて、必要とされるサーバーの数を決定するために、結合されたサービス（TS Licensing と他のサービス）に接続するユーザーの合計数を、共有サーバーがサポートできるユーザー数で割り算します。

1. TS Licensing **ロールを配置する、ネットワーク内の場所を決定する.** TS Licensing ロール･サービスは、自身が管理するターミナル･サーバー･ファームのセッションと、通信するための能力を持たなければなりません。高速 LAN 上のターミナル･サーバー･ファームは、TS Licensing サービスを共有することが可能です。その反対に、 WAN 接続により分離されるファームでは、それぞれのロケーションに、TS Licensing サービスを割り当てられることが必要とされます。

ターミナル･サーバー･ライセンスにおいて、独立した部門が入手して運用するの領域が存在する場合には（たとえば、研究開発とエンジニアリングを別のグループとしてサポートするケース）、それらのグループに対して、追加の TS Licensing ロールを提供することが必要になるでしょう。

1. TS Licensing **のフォールト･トレランス要件について判断する**. TS Licensing が利用できない場合には、新しいクライアントや、ライセンス期限の切れたクライアントは、ターミナル･サーバーへのアクセスを拒否されるでしょう。そのような事態に陥る可能性を減らすために、2つの TS Licensing サーバーを用意して、それぞれに半分ずつのライセンスをインストールします。続いて、双方のサーバーを、Active Directory 内に配信します。

以上の意思決定の結果を、補足資料に記録します。

## Task 3: Terminal Services Gateway Servers **のデザインと配置**

以下の方式を用いることで、外部ユーザーはセキュアではないロケーションから、ターミナル･サーバー･ファームに接続できます：

* Virtual Private Network（VPN）を介して、ユーザーに対するセキュア･ゾーンを拡張し、すべてのコーポレート･アプリケーションへのアクセスを提供します。このように、 VPN が確立されるときには、ユーザーはネットワーク上で認証されます。
* RDP プロトコルを介して、TS Gateway に接続します。そうすることで、暗号化された接続が提供されますが、ファイアウォールで RDP ポート（3389）をオープンする必要があります。
* HTTPs で暗号化された RDP を介して、TS Gateway に接続します。そうすることで、インターネット用の RDP ポート（3389）をオープンする必要なく、Network Address Translation （NAT）とファイアウォール通過が容易に提供されます。

外部クライアントが TS Gateway を介して、Terminal Services 環境に接続するとき、TS Gateway がセキュリティ･ブローカーとして振舞うことで、バックエンド･サービスを呼び出され、クライアント認証が行われる。 それにより接続が受け入れられ、また、Active Directory から呼び出される、Terminal Services 接続承認ポリシー（TS CAP）と、リソース承認ポリシー（TS RAP）を介して、クライアントが認証される。 さらに、クライアントの健康状態をテストするための、Network Access Protection も呼び出されるかもしれない。これらの接続を、Figure 4 が例証する。

承認が行われた後に、クライアントが意図するターミナル･サーバー接続が、TS Gateway サービスに伝えられます。TS Gateway サービスは、リクエストされたサーバーやターミナル･サーバー･ファームへの接続を、ファイアウォールを介して実現します。続いて、ターミナル･サーバーは、ユーザーからの Windows 認証の申請を処理します。そして、ユーザー認証が完了した場合に、 Terminal Services セッションの開始が可能となります。



図 4. TS Gateway により実現される、外部クライアントからTerminal Services 環境へのアクセス

コーポレート･ネットワーク内のブローカー認証が、ビジネス要件において必要とされる場合には、そのために TS Gateway を利用できます。

この機能を提供するためには、TS Gateway と RDC クライアントとの協調が前提となるために、RDC クライアント 6.0 以降が必要となります。

以下のプロセスを用いて、TS Gateway ロール･サービスが必要とされる場所と、そのフォーマットを決定し、さらに、個々の TS Gateway を接続するファームについて決定していきます：

1. TS Gateway ロール･サービスの必要性を判断する. ファイアウォール外のクライアントからの、ターミナル･サーバー･ファームへの接続の有無について、最初に決定します。このような 接続を行う場合には、クライアントからファームへの接続における、VPN ネットワーク利用の可能性について判断します。VPN ネットワーク接続が利用できない場合、あるいは、クライアントに内部ネットワークの全体を公開することが不適切な場合には、ターミナル･サーバー･ファームへのアクセスを提供するために、TS Gateway を配置することが可能です。ファイアウォールの外側からの、クライアントによるファーム接続が行われない場合には、TS Gateway は不要となります。
2. TS Gateway ロールの配置とセキュリティ･モデルに関する、要求を判断する. TS Gateway ロールは、セキュア･ネットワークの内外に配置が可能です。それに応じた、セキュリティ･モデルを決定していきます：
* TS Gateway を、セキュア･ネットワークの外側の、スクリーン･サブネットあるいはペリメータ･ネットワーク（DMZ）に配置します。この場合のクライアントは、TS Gateway サーバーのポート 443 を通じて、また、TLS/SSL のインターネットを介して、TS Gateway に接続します。

この方式は、TS Gateway にとって最も単純なコンフィグレーションであり、必要とされるサーバー数も最小で済みます。このコンフィグレーションの欠点は、TS Gateway が Active Directory ドメインのメンバーになることが、要求される点にあります。そのため、対象となるドメインを、ペリメータ･ネットワーク（DMZ）まで拡張しなくてはなりません。同様に、ポート 3389 と、デフォルト RDP ポートを、TS Gateway の背後にオープンする必要が生じます。

* セキュア･ネットワーク内に、TS Gateway を配置します。この場合には、 ISA サーバーあるいは、SSL 暗号を解除するための専用 SSL ターミネータが必要となります。

ポート 443を外側のファイアウォール上でオープンし、ポート 443 あるいは 80を、内部ファイアウォール上でオープンしまう。Active Directory は公開されず、また、ポート 3389 は使用されません。 さらに ISA サーバーにより、TS Gateway にバインドされるパケットをモニターすることで、もうひとつの保護ファイアウォール･レベルをレイヤ 7 として付け加え、疑わしい接続をシャットダウンすることが可能になります。

TS Gateway の配置について、ファイア・ウォールの内側あるいは外側するのか、それを決定します。

さらに、Network Policy Server（NPS）にホストされた Network Access Protection（NAP）サービスに対して、クライアントから供給される健康状態のステートメントをリクエストするように、TS Gateway を設定することが可能です。クライアントがヘルス･ポリシーを受け渡す場合には接続が完了し、また、適切なヘルス･レベルに達していない場合には更新が要求されます。

セキュリティ･ポリシーによる、NAP 実装の要求について、その有無を決定します。

クライアント･ログオン･プロセスにおける重要な部分を、TS Gateway は実行します。そして、ターミナル･サーバー･ファームや、Active Directory ドメインと、LAN スピードで通信を実現する必要があります。対象となるドメイン･サーバーやファームと、高速での接続を維持できるような状況に、TS Gateway ロール･サービスを配置します。

1. TS Gatewayに対する要求を決定する. このドキュメントを記述しているときには、TS Gateway の容量とパフォーマンスを計画するために、利用可能なガイダンスが存在していない。 小さなファームのロール･サービスの実装から始めて、パフォーマンスを測定し、結果の解析と調査を行い、それに応じたサイズに調整していきます。

さらなる正確性が必要とされる場合には、ユーザーの振舞いに合せたロード･テストにより要件を確立します。このロード･テストは、TS Gateway がボトルネックになるように設計しなくてはなりません。それにより、充分な容量を持つターミナル･サーバーで構成される、ファームに対する処理が実行できます。

TS Gateway のための構成要素を選択し、ターミナル･サーバー･ファームに接続する典型的なユーザーの、振舞いに対して影響をおよぼすような負荷テストを行います。

具体的には、テスト用のプロダクトや実際のシステムを用いた負荷テストを実施します。Roboserver（Robosrv.exe）と Roboclient（Robocli.exe）という、アプリケーション･スクリプト･サポートを取り込んだ、Terminal Services キャパシティ･プランニング･ツールが、 Windows Server 2003 Deployment Kit CD で提供されます。非接続の状態から再接続し、ファーム内のターミナル･サーバーをセッションを持つという、典型的なユーザーの行動を反映するサーバー上で、それらのツールを用いて模擬的な負荷を容易に生成し、管理することが可能です。

CPU および、メモリ、ディスク IO、ネットワークの負荷を記録するために、Appendix D のガイダンスを用いて、TS Gateway サーバーの性能をモニタリングします。 ひとまとめにされたユーザー･グループを持つサーバーに負荷をかけ、長い時間をかけながら負荷を増やしていきます。それぞれのグループでは、アプリケーションの一般的なセットを動かすべきです。プロセッサや、メモリ、ディスク、NIC の受容限度を超えるまで、システムをモニタリングします。そして限界を超えた直前のテスト結果から、ユーザーにとって快適な基準まで値を下げていきます。この低減されたユーザー数が、構成要素におけるサーバーごとの合計ユーザー数になります。サーバーにおける予備の容量を確保するために、バッファやヘッドルームとして引用されるパーセンテージを用いて、さらにユーザー数のを引き下げについて検討します。その結果として得られる、サーバーごとのユーザー数に関するキャパシティを、補足資料（Appendix C）に記録します。

このテストでは、以下の4つの組み合わせを実行します：

1. Heavy ユーザーに対して、ユーザーごとのコストを明確にする。
2. Normal ユーザーに対して、ユーザーごとのコストを明確にする。
3. Lightユーザーに対して、ユーザーごとのコストを明確にする。
4. 混成ユーザーについては、Heavy、Normal、Light ユーザーから予測値を引き出す。

それぞれのテストにおいて、ユーザーを追加し、パフォーマンス面で受容可能な閾値に達するまで、システムをモニタリングします。そして、サーバーで保持可能な、それぞれのタイプのユーザー数と、シングル･ユーザーごとのユニット･コストを、補足資料（Appendix C）に記録します。ユーザーの混在状態でのテストを完了するには、予測される Heavy、Normal、Light ユーザーの組み合わせに適合すると思われる、最終テストの結果を確かめます。続いて、受容できる稼働時間のレベルに至るまで、その時点のユーザー数を減らした後に、選択されたサーバーの構成要素における、TS Gatewayがサポートできるユーザー数とします。

1. **要求されるサーバー数を決定する.** TS Gatewayが必要とするサーバー数を決定するために、対象となるファームでセッションを持つユーザーの合計数を、TS Gateway サーバーがサポートできるユーザー数で割り算します。このプロセスを個々のファームで完了し、対象となる値を補足資料（Appendix C）に記録します。
2. TS Gateway をホストするサーバーについて、共有化を検討する.この時点で、TS Gatewayサービスのためのリソース要件が、それぞれのファームに通知されます。また、TS Gateway ロール･サービスの共有について、その可否を判断できるようになります。そのための処理は、以下の方式で完了します：
	* 1. TS Gateway サーバーの、ファーム間での共有が可能です. この場合、それぞれのファームにおいて、TS Gateway サーバーに対して定められたユーザー数を合計して、それの値をファーム数で割り算します。それにより、共有される個々のTS Gateway サーバーでサポートが可能な、ユーザー数が算出されます。

TS Gatewayが必要とするサーバー数を決定するために、ファームでセッションを持つユーザーの合計数を、TS Gateway サーバーがサポートできるユーザー数で割り算します。このプロセスを、それぞれの「共有された」ファームで完了し、その値を記録します。

* + 1. The TS Gateway サーバーにより、他の Terminal Services ロール･サービスをホスとすることが可能です. サーバー上で組み合わされる個々のロール･サービスに対して、サーバーの構成要素でサポートできるユーザー数を決定するためには、この Step の全 Task を完了しなければなりません。そして、個々のロール共有サーバー上でサポートできる、最も小さなユーザー数を取り出します。

それらのロールが必要とするサーバー数を決定するために、ファームに接続されるユーザーの合計数を、ロール共有サーバーがサポートできるユーザー数で割り算します。このプロセスを、それぞれの共有サーバーで完了し、その結果として得られる値を記録します。

* + 1. TS Gateway サーバーは、ファイル･サーバーなどの、他のサービスをホスとすることが可能です. おそらく、それらのサービスは可変的な負荷を持つため、この種のシナリオでは、TS Gateway ロール･サービスとホストされるサービスの、組み合わせに対する負荷テストが必要とされるでしょう。 上記の Step 2 のように進め、続いて、必要とされるサーバー数を決定するために、結合されたサービス（TS Gateway と他のサービス）に接続するユーザーの合計数を、共有サーバーがサポートできるユーザー数で割り算します。
1. **TS Gatewayのフォールト･トレランス要件について判断する**. TS Gateway が利用できない場合には、外部のクライアントは、ターミナル･サーバーへのアクセスを拒否されるでしょう。そのような事態に陥る可能性を減らすために、複数の TS Gateway を用いるフォールト･トレランス･ソリューションの実装を計画し、それらの TS Gateway を横断するトラフィックに対して、ロード･バランシングをとります。それを実施するためには、2つの方式があります：
	* 1. ハードウェア･ロード･バランサーを実装する. 多種多様なソリューションが利用可能であり、きわめて高いパフォーマンスが提供されます。
		2. Windows Network Load Balancing を実装する. ISA サーバーを用いて実装され、相対的に容易な設定が提供されます。

Note   この TS Gateway ロール･サービスは、 MSCS クラスタ内には実装できません。

以上の意思決定の結果を、補足資料（Appendix C）に記録します。

## **意思決定の概要**

これらのロール･サーバーは、 多数のファームにサービスを提供するように設計され、また、サイズを定められ、実装された Terminal Services に対して配置されます。すべてのターミナル･サーバー･ファームと、すべてのユーザーが必要とする、ロール･サービスへのアクセスの実現を保証するために、デザインについて再確認してください。

この Step 9 で取り上げた、TS Session Broker および、TS Licensing、TS Gateway といった一連のロール･サービスの中で、Windows Terminal Services の初期バージョンと共存できるのは TS Licensing だけです。そのため、Windows Server 2003 Terminal Services から Windows Server 2008 Terminal Services へ移行する際には、ファーム内のターミナル･サーバー機能やロール･サービスを提供するための、新しい環境の構築が必要になるでしょう。

## **参考資料**

* 以下のURLのTerminal Services on the Windows Server 2008 TechCenter を参照してください：t <http://go.microsoft.com/fwlink/?LinkID=73931>
* 以下のURLの What's New in Terminal Services for Windows Server 2008 を参照してください： <http://go.microsoft.com/fwlink/?LinkID=84635>
* 以下のURLの Windows Server 2008 TS Session Broker Load Balancing Step-by-Step Guideを参照してください： <http://go.microsoft.com/fwlink/?LinkID=92670>.
* 以下のURLの Windows Server 2008 TS Licensing Step-by-Step Setup Guide を参照してください：<http://go.microsoft.com/fwlink/?LinkID=85873>
* 以下のURLの Windows Server 2008 TS Gateway Server Step-by-Step Setup Guide を参照してください： <http://go.microsoft.com/fwlink/?LinkID=85872>.

#  Step 10: **セキュアな通信**

企業ネットワークの外側にあるクライアントと、ターミナル・サーバーが接続する場合には、その間の通信を保証するための数多く方式が存在します。それぞれのファームで選ばれるセキュリティ･レベルは、そのファームで実行されるアプリケーションや、クライアントの機能およびロケーションから生じる、セキュリティ要件に応じたものとなります。もし、すべてのクライアントが企業ネットワークの内側にあるなら、秘匿性の高いトランザクションを盗聴の危険性からの保護するための、追加のプロテクションを必要とするケースを除いて、あえてセキュアな通信を保証する必要性は生じないかもしれません。

## Task 1: **クライアントとターミナル･サーバー間の暗号化レベルを決定する**

Terminal Services セッションでは、ネイティブ 128 bit の RDP 暗号がデフォルトで用いられます。しかし、RDP の暗号強度については、ターミナル･サーバーで設定される、56 bit 暗号に置き換えることも可能です。クライアント接続において、高度な暗号化レベルのサポートが不可能な場合や、サーバーやクライアントが配置される国あるいは地域で、128 bit 暗号の使用が認められない場合には、この方式が必要になるかもしれません。

アメリカ 連邦政府の各部門と機関は、Federal Information Processing Standard（FIPS）140 暗号アルゴリズムの使用を要求しています。 Windows Server 2008 Terminal Services における通信で、そのアルゴリズムを使用することは可能ですが、その際には、対応するセキュリティ･レベルをサポートする、クライアントだけが接続可能となるでしょう。

対象となるクライアントからの接続が可能な、最も高いレベルの暗号化を決定して、そのレベルをターミナル･サーバーに実装します。

## Task 2: **コミュニケーション･シールドについて決定する**

ターミナル･サーバーにおいてアイデンティティを検証する認証は、RDP では提供されないため、介入者攻撃の危険性が生じてきます。通信の手順を開始する前に、TLS/SSL 暗号を利用することで、クライアントとサーバーを結ぶ相互認証を強制できます。この認証は、証明書の交換により成立します。

介入者攻撃に対するリスクを査定し、それがもたらす損害を算出します。その結果として得られる情報は、次の Task で証明機関を判断するときに用いられるでしょう。

## Task 3: **認証機関（**CA**）に関する判断**

クライアントとターミナル･サーバを結ぶ通信において、RDP あるいは HTTPs を使用する場合には証明書が必要とされるでしょう。ターミナル･サーバーで利用が可能で、すべてのクライアントが要求する証明書を、調達するための3つの方法があります：

* **自身で証明書を発行する**. TS Gateway がセキュア･ゾーンに配置される場合には、証明書はローカルで生成され、TS Gateway もしくは SSL ターミネータに配置されます。続いて、コンパニオン証明書が、これから接続される個々のクライアントに配信され、インストールされなくてはなりません。インフラストラクチャの設定は必要とされず、また、コストも発生しませんが、すべてのクライアント･マシン上に証明書をインストールする作業は、相当量の手間になり得ます。
* **外部のCAを利用する**. たとえば VeriSign のような、証明書サービスを有料で提供する、 数多くの信頼できる証明機関（CA）があります。それらの CA を利用する場合には、クライアントとサーバーは CA を信頼し、そこから提供される証明書を交換します。管理されていないクライアントが Terminal Services 環境に接続する場合には、クライアント･インストールが必要とされないため、この方式は極めて便利な実装となり得ます。
* **証明サーバーを独自運用する**. このケースにおいては、クライアントがアクセスするセキュア･ゾーンの外側に、証明書サーバーが配置されます。そのため、ハードウェアとソフトウェアに対する投資が必要となり、また、セットアップ作業と継続したメンテナンスも発生します。 それに加えて、すべてのクライアントが、このサーバーを信用しない可能性が残ります。

組織における選択にしたがって、それぞれのハードウェアと、ソフトウェア、作業全体のコストを判断します。続いて、全体的なコストと、クライアントに提供する利便性を比較します。そのプロセスが終了したところで、全体的に費用効果が最も高い方式を選択します。この時点で、以前の Task で算出した、介入者攻撃のリスクに対するコストとの比較を行います。コストよりも大きな利益が得られるなら、証明書を実装します。

## Task 4: HTTPs **を用いたカプセルかについて判断する**

クライアントが RDP を使って接続する場合には、ポート 3389 を外部ファイアウォール上でオープンしなくてはなりません。しかし、多くの組織は、80（http）と 443（https）にポートを絞り込み、インターネット上にオープンされるポート数を制限しようと努力しています。

外部ファイアウォールでオープンされるポートを、80と443 だけに絞り込むポリシーの有無について判断します。この制限が適用される場合には、 TS Gateway ロール･サービスを用いて、HTTPs 通信を実装します。

## **意思決定の概要**

クライアントとターミナル･サーバーを結ぶ通信において、実装するセキュリティが決定されはずです。ファーム･デザインの補足資料（Appendix C）に、この情報を記録します。

## **参考資料**

* 以下のURL のWindows Server 2008 Security Guideのページを参照してください ：<http://www.microsoft.com/technet/security/prodtech/windowsserver2008/default.mspx>

# **まとめ**

このガイドは、Windows Server 2008 をベースとした、Terminal Services インフラストラクチャにおける重要な判断および、アクティビティ、デザインを成功させるために必要な作業を要約しています。

ガイドでは、以下の項目と関連する、基本的な決定と作業に取り組んできました：

* Terminal Services により配信すべきアプリケーションと、Terminal Services で用いる正しいアプローチを決定する。
* 選択されたアプリケーションをサポートするために、 Terminal Services の利用で必要となるリソースを決定する。
* Terminal Services インフラストラクチャにおける、コンポーネントおよび、 レイアウト、セキュリティ、接続性を設計する。

これらの決定と作業は、適切なデザインを得るために提供される、10段階の意思決定フローを介して読み手を導くことで完了します。 対象となるデザインが適切な場合には、典型的な用法のシナリオを用いて、判断と作業を例証してきました。

このガイドは、意志決定プロセスの包括的なレビューを達成するために必要な、技術的な側面および、サービスの特質、そしてビジネス要件についても説明しています。

導入部分で述べたように、プロジェクトにおけるビジネス目的を充分に理解することが、Terminal Services プロジェクトのスタートにおいて、きわめて重要な事柄となります：

* 仮想化されたプレゼンテーション層を活用することで、どのようなメリットに到達することを、企業は望むのだろうか?
* それらのメリットから得られる価値は何であり、また、それらのメリットを提供するために Terminal Services を使用する際のコストとは何なのか？

プロジェクトのスタート時において、ビジネスの目的が、最優先に位置づけられるべきです。そして、それらが正しく理解されることで、IT とビジネスの間に合意が生まれます。つまり、いくついかのアプリケーションは、すぐに Terminal Services で配信されることは無く、また、プロジェクトのスコープに収めるためには、いくつかの変更が必要とされるからです。そして、それらの変更にはコストがかかるため、プロジェクトを具体的に進める前に、ビジネス担当者に対してフィードバックを行うべきです。このようなプロセスを経ることで、追加のコストが理解され、また、ビジネス上の最適な判断へと到達することになります。

アーキテクチャのドラフトが仕上がった後、主要なプロセスが開始されるまでの間に、限定された「パイロット」テストを実施すべきです。それにより、学習と実践が、デザインの中に取り入れられるようになります。

このガイドを、プロダクトのドキュメントと組み合わせて用いれば、Windows Server 2008 の Terminal Services テクノロジーについて、自信に裏付けられた実装が計画されるでしょう。

## **参考資料**

* 以下のURL のTerminal Services Team Blog を参照してください：<http://blogs.msdn.com/ts/>

# Appendix A: **ユーザーとアプリケーションに関するデータを、補足資料に記録する**

ユーザーとアプリケーションについて必要なデータを集めるために、主だったユーザーにインタビューを行い、その結果を以下のようなスプレッドシートに記録します。



主要なユーザーから得られる情報を記録します：

* ユーザーのロケーション（Location） それぞれのロケーションにおける、ユーザー数を把握することで、サーバーのサイズと配置に関する決定を進めていきます。
* クライアントのオペレーティング･システム（Client OS） Terminal Services によりアプリケーションが配信されるときには、エンドユーザーのマシン上で実行される Remote Desktop Client（RDC）との間で、RDP プロトコルを用いた通信が行われます。 しれにより、スクリーン更新および、キーボード入力、マウス･クリックなどの情報が、ネットワーク上で交換されていきます。RDC と RDP の最新バージョンは、暗号化におけるレベルの向上と、ネットワークを介したデータ転送効率を、大幅に拡張しています。ただし、クライアントにインストールされているオペレーティング･システムの種類により、実行可能な RDC と RDP のバージョンが限定されるため、それらの拡張部分の利用にも制約があります。たとえば、シングル･サインオンの機能は、Windows Vista Service Pack 1（SP1） もしくは、Windows XP SP3 を要求します。
Step 5 でターミナル･サーバーのファーム数を決定する際に、こうした情報が利用されますが、その理由は、それぞれのファームにおけるクライアントに対して、最高と最低のセキュリティ設定が必要とされるからです。同様に、それらの情報は Step 9においても、TS Gateway ロール･サービスをデザインするために用いられるでしょう。
* アプリケーションに対するシングル･サインオンが提供されるのか（Single sign on） ユーザーがシングル･サインオンの利便性を期待する場合には、アプリケーション配信で用いる Remote Desktop Client（RDC）のレベルが、そのことの可否を決定するでしょう。配信されたアプリケーション環境で、このエクスペリエンスを提供するためには、RDC 6.1 が必要とされます。そのために使用される RDC のレベルが、Step 5 で要求されるファーム数と、Step 9 でデザインされるロール･サービスに、影響を与えるかもしれません。
* 使用されるアプリケーションと、そのバージョン、そして、特殊なローカル･カスタマイズについて（Applications used）前提となるアプリケーションに、複数のバージョンが存在する場合には、同一の Terminal Services 環境での共存について、確認していく必要性が生じるでしょう。共存が不可能な場合には、別々のターミナル･サーバー･ファームから、それぞれのバージョンやカスタマイズされたものを配信する必要が生じ、プロジェクトの複雑さとコストが増大していくでしょう。この問題については、Step 5 で判断することになります。
Microsoft App-V を用いて、それぞれのアプリケーションをターミナル･サーバー上で実行することで、この問題を解決できるかもしれません。そのような場合には、対象となるアプリケーションの配信を、Microsoft App-V でサポートする必要があり、また、 Microsoft App-V 環境のインスタンスも必要になります。
* SLA のレベル（SLA） どのような SLA を設定し、どのようなサービス･レベルを約束し、どれだけのユーザーを個々のサービス･レベルに取り込むのでしょうか? アプリケーションに対する企業やユーザーからの期待を理解することで、パフォーマンスと可用性の要件が定義されていきます。それにより、Step 7では、ターミナル･サーバー･ファームのサイジングが促進され、ユーザーの期待するパフォーマンスが提供可能となるでしょう。また、 Step 7 では、フォールト･トレランスのアプローチも決定されるでしょう。SLA が正式なものであっても、また、非公式なものであっても、ユーザ－･グループが期待するパフォーマンスと可用性のレベルを理解することが、とりわけ重要となります。そして、ユーザ－･グループが経験してきた実際のパフォーマンスや可用性よりも、このドキュメントから得られる情報を、デザインのインプットとして用いることが重要です。
* 接続のタイプ（Connection type）すべてのユーザーが、LAN で接続されているのでしょうか? それとも、一部のユーザーは、WAN や 、ダイアルアップ、ISPを介して接続されているのでしょうか? ネットワークやセキュリティ･モデル、そしてサーバーの配置やサイズをデザインするために、それらの情報が利用されるでしょう。ユーザーへのインタビューでは、この種の情報を集めることは不可能かもしれません。それよりは、IT 担当者がインタビューから結論を導き出すことで、ネットワーク･デザインを検証することが必要になるでしょう。

さらに、以下の情報を集めることも支援になり得る。 もし、アプリケーションと使い方について質問が加わるならば、きわめて有用なものになる。

* アプリケーション･サポート･グループ. アプリケーションを責任を持ってサポートする、担当者をリストアップします。これから発生してくる、テクニカルな問題に対する初期段階でのテストと調査において、能力を発揮する人材を把握します。
* アプリケーション担当者. アプリケーションに関するアップグレードとライセンス管理に責任を持つ、部門あるいは責任者をリストアップします。それにより、アプリケーション配信の方式について、変更を承認する責任者が明確にされ、また、発生すると思われるビジネス上の問題が把握されるでしょう。

# Appendix B: **アプリケーション分析に関する補足資料**

Windows Server 2008 と Terminal Services を組み合わせを介して配信される、アプリケーションの適合性に関するデータを記録するために、以下のスプレッドシートを使用します。



# Appendix C: **ファーム･デザインに関する補足資料**

それぞれのターミナル･サーバー･ファームのデザインにおける、各種の決定事項を追跡するために、以下のようなスプレッドシートを使用します。



# Appendix D: **サーバー･パフォーマンスの分析とスケーリング**

以下の情報は、システムのキャパシティ･プランニングと、パフォーマンス･モニタリングのために用いる、重要なモニタリング･カウンタを確認するためのものです。

これらのカウンタは Step 7: “ファームのデザイン” で、ターミナル･サーバーと TS Web Access サーバーの、構成要素およびサイズを判断するために参照されます。さらに、Step 9: “ファームにおける Terminal Services ロール･サービスの、サイズと配置を考える” においても、それぞれの Terminal Services ロール･サービスの、構成要素およびサイズを判断するために参照されます。

ターミナル･サーバーと Terminal Services ロール･サービスは、通常のサーバーとは異なる負荷をホストするため、適切なデザインとスケールを定め、その環境を効果的に運用することが、なによりも重要となります。何故なら、もともとはクライアント上で実行するように設計されたアプリケーションの、マルチ･コピーを実行することになるからです。この負荷を構成する要素は多様であり、ターミナル･サーバーのサイズについて「すべてにフィットするサイズ」は存在しません。そのため、エンドユーザーの要求を充たすデザインに到達するためには、注意深い測定とテストを実施しなくてはなりません。

## **プロセッサ稼働率**

CPUリソースに対する過大な負荷は、同じサーバー上のすべての処理に悪影響を与え、パフォーマンス上の問題を引き起こし、多くのユーザーを深刻な状況に陥れます。CPUリソースの利用パターンは大きく変化する可能性があるため、全体的なリソース要件を数量化するための、測定基準や構成単位が存在しません。システム全体におけるプロセッサ利用とスレッド遅延の、最も高いレベルを測定できます。 以下のテーブルがリスト･アップするのは、測定インターバルを通じて全体的平均したプロセッサの使用率と、プロセッサ Ready Queue におけるスレッド待ちの量を、取得するためのパフォーマンス･カウンタです。

Table D1. プロセッサ使用率に関する、パフォーマンス･モニターのカウンタ

|  |  |  |
| --- | --- | --- |
| Object | Counter | Instance |
| Processor | % Processor Time | \_Total |
| System | Processor Queue Length | N/A |

### Processor\% Processor Time

一般的な基準として、90％ 以上の使用率を継続するプロセッサは、CPU 許容量の限界で稼働していると考えるべきです。 通常において 75-90%の範囲で実行されるプロセッサは、許容量の限界に近づいているため、しっかりとモニターすべきです。また、 20% 程度にも満たない使用率のプロセッサは統合の適切な候補となります。

レスポンスが重視されるシステムでは、その反応に影響をおよぼす負荷が生じているときに、80% 以上の使用率が持続される期間について、詳細に調査すべきでる。 スループット志向のシステムにおいては、許容量の制約を除いて、高い使用率が関心事になることは稀です。

マルチ･プロセッサ･コンフィグレーションおよび、ハイパー･スレッド論理プロセッサに固有のハードウェア要因は、このドキュメントのスコープを超えるものであり、また、難解な問題を提起するものでもあります。さらに、32-bit と 64-bit のプロセッサ比較は、ハードウェアとプロセッサのファミリーを横断した、パフォーマンス特性の比較ほど簡単ではありません。これらのトピックについて、Microsoft Windows Server 2003 Performance Guide の Chapter 6 “Advanced Performance Topics” を参照してください。

### System\Processor Queue Length

Processor Queue Length を利用することで、プロセッサ競合や高いCPU 使用率が、割り当てられた負荷を処理しきれないプロセッサ･キャパシティにより引き起こされたのか、それ以外の原因により引き起こされたのか、その点が明らかにされます。Processor Queue Length は、プロセッサ Ready Queue 内で遅延しているスレッド数と、実行を待っているスレッド数を示します。 なお、リスト･アップされる値は、直近の実測値となります。

シングル･プロセッサ･マシン上で、5 以上の待ち行列が生じる場合は、プロセッサが直ちに処理できる能力よりも高い頻度で、要求が生じている状況を示すワーニングだと判断できます。この値が 10 以上になり、また、CPU 使用率も高い場合には、プロセッサが限界に至っている可能性が、きわめて高いとみなすべきです。

マルチ･プロセッサを用いたシステムでは、待ち行列を物理的なプロセッサ数で割り算します。待ち行列に対して大きな容量を持つハード･プロセッサ･アフィニティ（特定のCPUコアへのプロセス割り当て）を用いるマルチ･プロセッサ･システムでは、アンバランスなコンフィグレーションを示す値が参照されるでしょう。

一般的に、Processor Queue Length はキャパシティ･プランニングに用いられますが、対象となる環境内で負荷を処理するシステム能力に関する判断や、将来におけるプロセッサの追加もしくは、高速プロセッサの導入について判断するためにも利用できます。

## **メモリ利用率**

サーバー上のメモリ使用率を適切に把握するためには、物理と仮想の両面からモニタリングする必要があります。物理メモリが不足すると、過度のページングや、パフォーマンスの問題が生じる可能性があります。また、仮想メモリが消耗すると、広範囲におよぶアプリケーションの失敗や、システムのクラッシュといった、致命的な障害が生じる可能性があります。

Table D2. メモリ使用率に関する、パフォーマンス･モニターのカウンタ

|  |  |  |
| --- | --- | --- |
| Object | Counter | Instance |
| Memory | Pages/sec | N/A |
| Memory  | Available Mbytes | N/A |
| Memory | Pool Paged Bytes | N/A |
| Memory  | Pool Paged Resident Bytes | N/A |
| Memory  | Transition Faults/sec | N/A |
| Memory | Committed Bytes | N/A |
| Process | Working Set | <Process Name> |

### Memory\Pages/sec

物理的な RAM が不足するにつれて仮想メモリ・マネージャは、メモリ･ページの情報をディスクのキャッシュに転送することにで、RAM を開放していきます。ディスクへの過度なページングは、利用可能なディスク帯域幅を過剰に消費し、また、同じディスク上のファイルにアクセスする、アプリケーションの処理速度を低下させるかもしれません。Pages/Sec カウンタは、ディスクに対してリード･ライトを行う、全体的なページング率を追跡します。

キャパシティ･プランニングにおいては、このカウンタの上昇傾向についてモニタリングすべきです。通常では、メモリを追加することで、過度のページングは低減します。ページング処理が、全体的なディスク I/O 帯域幅の 20-50% を消耗する場合には、メモリを追加すべきです。ディスク帯域幅は有限であるため、ページング処理のために使用される分だけ、アプリケーションによるファイル操作が妨げられます。

Total Disk I/O Bandwidth は、システム上のる全ディスクに対する、Pages/sec とPhysical Disk\Disk Transfers/sec の比率です：

Memory\Pages/sec ÷ Physical Disk (\_Total)\Disk Transfers/sec

### Memory\Available Mbytes

Available MBytes は、プロセスあるいはシステムに対して、直ちに割り当てること可能な、物理的なメモリ量をメガバイト単位で示します。 Available Megabytes のパーセントは、追加メモリに関する必要性の有無を判断するために利用できます。この値が、一貫して10%を下回る場合には、メモリを追加すべきです。 以下の計算式で、Available Megabytes のパーセントが算出されます：

(Memory\Available Mbytes ÷ System RAM in Megabytes) \* 100

この値は、物理的なメモリが十分に供給されていることを、判断するための主要な指標となります。そのため、値が下降するときには、メモリの追加について検討すべきです。カウンタは、Available Bytes と Available KBytes という、二種類の単位で利用できます。なお、メモリが不足しているときは、Pages/secがメモリ競合ににおける最適な指標となります。

### Memory\Pool Paged Bytes and Memory\Pool Paged Resident Bytes

Pool Paged Bytes は、使用されないオブジェクトをディスクに書き込むために、オペレーティング･システムが使用するシステム･メモリ領域である、プール･ページのサイズをバイト単位で示します。

Pool Paged Resident Bytes ノン･プール･ページのサイズをバイト単位で示します。ディスクへの書き込みはできなくとも、アロケートされている限り物理メモリ上に保持されるオブジェクトのために、オペレーティング･システムが使用するシステム･メモリ領域のことです。Pool Paged Resident Bytes に対する、Pool Paged Bytes の比率は、以下のように計算できます：

Memory\Pool Paged Bytes ÷ Memory\Pool Paged Resident Bytes

この比率は、メモリ競合のインデックスとして使用することが可能であり、結果としてキャパシティ･プランニングを支援することになります。 この値がゼロに近づくにつれて、Nonpaged プールと Page プールを増大するための、メモリ追加が必要となります。

この、Pool Paged Resident Bytes からリターンされるサイズを、TCP 接続を追加するための計画に利用できます。それぞれの TCP 接続のためのステータス情報は、Nonpaged プールにストアされます。そのため、メモリを追加することで、TCP 接続を追加するためのスペースが、Nonpaged プールへに追加されます。

### Memory\Transition Faults/sec

Transition Faults カウンタは、サンプリング･インターバルにおける、ソフトおよぶトランジションのフォールト数をリターンします。Standby リスト上の調整されたページが、再び参照されるときに、トランジションのフォールトが生じます。このページは、その後にワーキング･セットに戻されますが、ディスクには決して保存されないことが重要なポイントになります。

この値が上昇していくときは、メモリ不足が生じるかもしれないという指標になります。ただし、トランジション･フォールト 自身の比率が高いことが、パフォーマンスの問題を示すということではありません。通常では10%程度の最小閾値に、Available Megabytes の値が近づく場合は、利用可能なページ･プールを維持するために、オペレーティング・システムが機能すべき状態であることを示します。

### Memory\Committed Bytes

Committed Bytes は、コミットされた仮想メモリ量を測定するための値です。コミットされたメモリとは、プロセスに割り当てられたメモリを、関連するスレッドから操作するためにシステムが確保する、物理的な RAM やページング･ファイル上のスペースのことです。

Committed Bytes:RAM と呼ばれる、メモリ競合のための指標を算出し、キャパシティ･プランニングを支援し、また、パフォーマンス向上を図ることが可能です。Committed Bytes:RAM 比率が1より大きいときには、仮想メモリが RAM より大きいことを示し、若干のメモリ管理が必要になるでしょう。この比率が1.5 を超えるときには、ページング･ディスクの帯域幅の制限値まで、ディスクに対するページングは増加するでしょう。 そのときには、メモリを追加すべきです。Committed Bytes:RAM は、次のように計算されます：

 Memory\Committed Bytes ÷ System RAM in Bytes

### Process\Working Set

Working Set カウンタは、アドレス指定が可能なプロセスに対して、ページ･フォールトを発生させずに割り当てられるメモリ総量を示します。すべてのプロセス･アドレス空間をまたいで、全体的に割り当てられる RAM の総量を見るためには、Working Set の \_Total インスタンスを使用します。重要なアプリケーションについて、この値の上昇傾向をモニタリングすべきです。

IIS や、Exchange Server、SQL Server などの、いくつかのサーバー･アプリケーションは、自身のプロセス Working Set を管理します。それらの Working Set を測定するためには、アプリケーション固有のカウンタ を用いなければなりません。

## **ディスク･ストレージ要件**

ストレージ要件をプランニングするためのプロセスは、キャパシティ要件とディスク性能に分けられます。全体的なキャパシティ要件を決定しても、システムのパフォーマンス要件だけではなく、フォールト･トレランス要件がストレージ･サブシステムの実装に影響をおよぼすでしょう。たとえば、シングル･ドライブで充分なストレージ･スペースを供給できても、そのシングル･ディスクの性能が、システムの性能に対する要求を充たさない場合もあります。

そのような状況を回避するためには、キャパシティとパフォーマンスの要件について、双方を充たしていく必要があります。それにより、ストレージ･サブシステム･ドライブの、サイズや、スピード、コンフィグレーションなどの判断が変化するかもしれません。

### ディスク･スペースのキャパシティ

必要とされるストレージの総量は、オペレーティング･システムの要件に加えて、システムにストアするアプリケーション･データをベースとして、計算することが可能です。

### ディスク性能

一般的に、一定のピーク時間内で測定された、毎秒ごと I/O オペレーション（IOPS）の合計値として、ディスク･パフォーマンスは表現されます。

システムの IOPS 要件を満たすために必要な、ディスク数を決定するためには、前提となるドライブの IOPS を決定しておく必要があります。ただし、IOPS はアクセス･パターンに大きく依存するため、問題はさらに複雑になります。たとえば、一般的なディスクの特性としては、ランダム･ライトよりも、シーケンシャル･リードの方が高い IOPS を持つことになります。こうした理由により、短時間における入出力オペレーションに基づいた、最悪の IOPS を計算する方式が通常では採用されます。

ドライブの IOPS を計算するために、ドライブに関する情報を集める必要があります。ドライブの製造業者から提供される資料に基づき、必要となる情報を以下のテーブルにリストアップします。

Table D3. IOPS計算のために必要な情報

|  |  |
| --- | --- |
| 必要な情報 | 詳細説明 |
| Spindle Rotational Speed (RPM) | RPM で表現する、スピンドル回転数です。 |
| Average Read Seek Time (ms) | Read の平均シーク時間です。 |
| Average Write Seek Time (ms) | Write の平均シーク時間です。 |

IOPS 計算における最初のステップは、ドライブが処理すべき Average Seek Time を、ミリセカンドで決定することです。まず、リードとライトの比率が、50/50 という仮定を立てます。リードとライトの比率を修正する場合には、Average Seek Time の調整が必要になるだでしょう。たとえば、ドライブが、4.7ミリセカンドの Average Read と、5.3ミリセカンドの Average Write を持つ場合には、このドライブの Average Seek Time は5.0msになるでしょう：

5.0ms = (4.7ms + 5.3ms) ÷ 2

それに続いて、IO Latency の計算を行います。この値は、Average Latency と Average Seek Time を加えることで算出されます。現時点で市場に提供されている、一般的なスピンドル･スピードにおける Average Latency を、以下のテーブルにリストアップします。

Table D4. HDD の回転数をベースとした平均的な遅延時間

|  |  |
| --- | --- |
| Spindle Rotational Speed (rpm) | Average Latency (ms) |
| 4,200 | 7.2 |
| 5,400 | 5.6 |
| 7,200 | 4.2 |
| 10,000 | 3.0 |
| 15,000 | 2.0 |

10,000 rpm のスピンドル･スピードを持つドライブをサンプルとします。先ほどの 5.0ms に3.0ms を加えるので、このドライブは 8.0 msの IO Latency を持つことになります：

8.0 ms = 5.0ms + 3.0ms

ドライブが一度に行えることは、1つの IO オペレーションのみとなります。1ms の間に実行可能な IO の値を計算するために、1 を IO Latency の値でを割り算します。その結果を 1000倍にすることで、この値は毎秒の IO に置き換えられます。例題におけるドライブの IOPS は、125 と評価されます：

125 IOPS = (1 IO ÷ 8.0ms) \* 1000 ms/sec

## **ストレージ要件**

ストレージ要件を決定するためには、検討中のシステムに関する、追加情報も収集しなければなりません。いくつかの情報は、識別が容易であり、説明も不要なものになるでしょう。ただし、その他の情報は、定量化のためのデータが欠如するために、その確認が難しくなるかもしれません。以下の全項目は、共有ストレージ･システムを考慮に入れていますが、それぞれの値はサーバーごとに算出されています。そのため、そのストレージを共有するシステム数に基づいて、情報のスケールアップが可能となっています。収集する必要のある情報を、以下のテーブルに示します。

Table D5.ストレージ要件を計算するために必要な情報

| 必要な情報 | 詳細説明 | 例 |
| --- | --- | --- |
| # Users Per Server | サーバーがホストする、ユーザの総数です。 | 700 |
| % Concurrent Users | ピーク時において、サーバーに接続しているユーザーのパーセンテージです。 | 80% |
| IOPS per User Required | 不特定のユーザーが生じる、IOPS 値です。 | 0.5 |
| Storage Capacity in Gigabytes | 計画されるディスク･ストレージの容量です。 | 450 |
| % Buffer Factor (for growth) | システムが許容する、ディスク･ストレージ増領分のパーセンテージです。 | 20% |
| Read % of IOPS | Read 処理に関する、IOPS 比率です。 | 50% |
| Write % of IOPS | Write 処理に関する、IOPS 比率です。 | 50% |
| Disk Size (GB) | ストレージ･システムで検討するドライブ容量です。 | 146 |
| Calculated Drive IOPS | ストレージ･システムにおいて検討される、ドライブの IOPS 値です。 | 125 |

このテーブル内の情報は、IOPS per User Required を除いて、かなり明確なものとなっています。ただし、この測定結果では、システム上の一人のユーザーが生成する IOPS 数を用いています。アプリケーションが極端に IO を利用しない限り、それらのアプリケーションのベンダーは通常、このような情報を集めません。すべてのユーザーが生成する IOSP 数を、実際のシステムを観察することで取得できるかもしれませんが、多くの課題を伴うため、このガイドのスコープ外とします。上記の例を活用するために、このガイドでは、すべてのユーザーが 0.5 IOPS を生成するという仮説を用います。

Table D5の情報をベースにすると、計算する必要のある多数の測定値が生じてきます。以下のテーブルに、それらの値を示します。

Table D6. ストレージ要件を計算するために、生成しなければならない情報

|  |  |  |
| --- | --- | --- |
| 必要な情報 | 詳細説明 | 例 |
| # of Concurrent Users | サーバーごとのユーザー数と、同時利用するユーザー比率から計算される、コンカレント･ユーザーの人数です。 | 560 |
| IOPS per Server Required | コンカレント･ユーザー数と、それぞれのユーザーが生じる IOPS 値を掛け算した結果です。 | 280 |
| Total Storage Requirements | Storage Capacity に対して、増量分の20%を加えた値です。 | 540 |
| Number of Read IOPS | IOPS per Server Required に対する、 Reads パーセンテージの値です。 | 140 |
| Number of Write IOPS | IOPS per Server Required に対する、 Write パーセンテージの値です。 | 140 |
| Drive Size Actual (GB) | フォーマット後のディスクサイズは、公称値より 10% ほど低減します。この値により、失われる容量を補完します。 Disk Size（GB）の 90% が、この計算の結果となります。 | 32 |

RAID 0+1 と RAID 5 は、システムの冗長性を確保する上で、最も一般的なドライブ･コンフィグレーションとなります。だたし、この2つの RAID システムは、それぞれの運用形態に応じて、別個の IOPS の計算を持ちます。

### RAID 0+1 の計算

RAID 0＋1 のストレージ要件を充たすためには、まず、Total Storage Requirements を Drive Size Actual で割り算します。その計算結果を四捨五入して、2倍にします。 上記の例において、RAID 0＋1 でストレージ要件を充たすためには、10台のドライブが必要とされます：

10 = ROUNDUP(540÷132)\*2

RAID 0＋1でパフォーマンス要件を充たすためには、まず、Number of Write IOPS を2倍して、Number of Read IOPS を加えます。その合計を Calculated Drive IOPS で割り算し、結果を四捨五入します。上記の例において、RAID 0＋1 でパフォーマンス要件を充たすためには、4 台のドライブが必要とされます：

4 = ROUNDUP(((140\*2)+140)÷125)

つまり、この RAID 0＋1 の例では、4台のディスクでパフォーマンス要件を充たせますが、キャパシティ要件を充たすためには、10 台のディスクが必要とされます。

### RAID 5 の計算

RAID 5 ストレージ要件を充たすためには、Total Storage Requirements に1.2を掛け算し、パリティ･ストレージ要件を調整した上で、必要なドライブ数を算出します。この値を、Drive Actual Size で割り算し、四捨五入します。上記の例では、RAID 5 でストレージ要件を充たすために、5台のドライブが必要となります：

5 = ROUNDUP((540\*1.2)÷132)

RAID 5 でパフォーマンス要件を充たすためには、まず、Number of Write IOPS を4倍し、そこに Number of Read IOPS を加えます。この合計を、Calculated Drive IOPS で割り算し、結果を四捨五入します。上記の例では、RAID 5でパフォーマンス要件を充たすため、6 台のドライブが必要となります：

6 = ROUNDUP(((140\*4)+140)÷125)

つまり、この RAID 5 の例では、5台のディスクでパフォーマンス要件を充たせますが、キャパシティ要件を満たすためには、6 台のディスクが必要とされます。

### RAID 0+1 と RAID 5 の比較

上記の例から判るように、10000回転の 146GBのドライブを使うときには、RAID 5 が最良の選択になるでしょう。しかし、これらの計算をするとき、別タイプのドライブに注目することも重要です。たとえば、146GBのドライブの代わりに 300GB のドライブを用い、その他の特質に変化がなければ、その後に続く選択が大きく変化するでしょう。

300GBのドライブを用いた RAID 0＋1 では、4台のドライブで、キャパシティとパフォーマンスの特質を充たすことが可能でしょう。 同様に RAID 5 の場合には、3台のドライブでキャパシティ要件を充たすことが、そして、6台のドライブでパフォーマンス要件を充たすことが可能でしょう。ドライブのサイズを変更することで、最良の選択も、同様に変化していきます。

## **ストレージのモデリング**

システム･パフォーマンス要件の、特徴づけを支援するためにIOPS が用いられます。しかし、システムが稼働した後に、ディスク･サブシステムのスピードが低下する場合には、別のパフォーマンス･モニタリング･カウンタを利用できます。

Table D7. ディスク性能に関するパフォーマンス･モニターのカウンタ

|  |  |  |
| --- | --- | --- |
| Object | Counter | Instance |
| Physical Disk | % Idle Time | <All Instances> |
| Physical Disk | Disk Transfers/sec | <All Instances> |
| Physical Disk | Avg. Disk secs/Transfers | <All Instances> |
| Physical Disk | Split IO/sec | <All Instances> |

### Physical Disk\% Idle Time

％ Idle Time カウンタは、サンプリングされた時間内の、ディスク･アイドル状態をパーセントで示します。アイドル･タイムが 20% 以下の場合には、ディスクが過負荷の状態にあると疑うべきです。

Physical Disk(n)\Disk utilization の値は、100％ の数値から Physical Disk(n)\% Idle Time を引き算することで得られます。

### Physical Disk\Disk Transfers/sec

Disk Transfers/sec とは、サンプリング時間内に完了する I/O request Packets (IRP) 数のことです。ディスクが一度に処理できるのは、ひとつの I/O オペレーションだけです。そのため、コンピュータにアタッチされた個々の物理ディスクは、利用可能なディスク IO レートの上限値を引き出さなければなりません。ディスク･アレイが使われる場合には、個別のディスク I/O レートを見積もるために、Disk Transfers/sec をディスク数で割り算します。

Physical Disk(n)\Average Disk Service Time/Transfer は、Physical Disk(n)\Disk Utilization の値を Physical Disk(n)\Disk Transfers/sec で割り算することで得られます。それにより、個々の物理ディスク向けたリクエストに対する、ドライブの応答速度が示されます。この値が、ディスクに対して設定された値を上回る場合には、サブ･システムが過負荷の状態にあると疑うべきです。

### Physical Disk\Avg. Disk secs/transfers

Avg.Disk secs/transfer は、サンプリング時間内における、物理ディスク応答スピードの全体的な平均値です。 この値には、デバイスが受け付けたリクエストに対する応答時間と、キューで消費される待ち時間の、双方が含まれます。ディスクあたりの毎秒の I/Oが、15-25 に上昇する場合には、ディスク･レスポンスが不足しいると考えられるため、調査が必要となります。

The Physical Disk(n)\Average Disk Queue Time/Transfer の値は、Physical Disk(n)\Avg. Disk secs/Transfer の値から、Physical Disk(n)\Avg.Disk Service Time/Transfer を引き算することで得られます。 Average Disk Queue Time/Transfer は、サービスを提供するキューの中で、リクエストを待たせる時間の総量を示します。このキューの待ち時間が長くなることは、ディスク･サブシステムのレスポンス低下を示しますが、厳密に言えば、物理ディスクにおけるレスポンス低下となります。

### Physical Disk\Split IO/sec

Split IO/sec とは、サンプリング時間内における物理的なディスク･リクエストが、多数のディスク･リクエストに分割される比率のことです。Split IO/sec の値が上昇するときには、ディスク内のフラグメンテーションが悪化して、パフォーマンスに悪影響が生じている状況を示します。Split IOs のパーセンテージは、指定されるディスクが「n」の場合に、以下の数式で計算できます：

 (Physical Disk(n)\Split IO/sec ÷ Physical Disk(n)\Disk Transfers/sec) \* 100

このパーセンテージが、10-20 %を超える場合には、ディスク･デフラグメンテーションの必性について調べるべきです。

## **ネットワーク性能**

大半のシステムにおいて、他のアプリケーションやサービス、そしてユーザーとの通信を保証するための、実運用環境のネットワークに対するアクセスが要求されます。ネットワーク要件に含まれる要素として、たとえばスループットは、単位時間あたりのネットワーク接続における、所定のポイントを通過するトラフィックの総量を示します。

その他のネットワーク要件には、多数のネットワークとの接続という要素が含まれます。

Table D8. ネットワーク性能に関するパフォーマンス･モニターのカウンタ

|  |  |  |
| --- | --- | --- |
| Object | Counter | Instance |
| Network Interface | Bytes Total/sec | (Specific network adapters) |
| Network Interface | Current Bandwidth | (Specific network adapters) |
| Ipv4 & Ipv6 | Datagrams/sec | N/A |
| TCPv4 & TCPv6 | Connections Established | N/A |
| TCPv4 & TCPv6 | Segments Received/sec | N/A |

### Network Interface\Bytes Total/sec and Network Interface\Current Bandwidth

Bytes Total/sec とは、特定のインターフェイスを介して、毎秒ごとに送受信されるバイト数のことです。Current Bandwidth カウンタは、ネットワーク･アダプターの実質的なパフォーマンス･レベルを反映しますが、キャパシティには関係しません。セグメント内のギガビット･ネットワーク･アダプター･カードが、想定よりも低いスピードを強いられる場合には、Current Bandwidth カウンタに1 Gbps から100 Mpbs への低下が反映されるでしょう。

次の方程式の「n」に、実際に使用するネットワーク数を指定することで、それぞれのネットワーク･インターフェイスの稼働率が計算できます：

 (Network Interface(n)\Bytes Total/sec ÷ Network Interface(n)\Current Bandwidth) \*100

所定のネットワーク･アダプタの %Busy が 90% を超える場合には、ネットワーク･リソースの追加が必要になるでしょう。一般的には、スイッチ･リンク上で達成可能な最大帯域幅は、Current Bandwidth カウンタの 90-95 ％ に近づくべきです。

### Ipv4 & Ipv6\Datagrams/sec

これらのカウンタは、サンプリング時間内において、毎秒ごとに送受信される IP データグラムの合計数を示します。このカウンタからベースラインを生成することで、ネットワーク使用率の傾向を分析し、また予想することが可能になります。

### TCPv4 & TCPv6\Connections Established

Connections Established カウンタは、測定インターバルのESTABLISHED ステートにおける、最終的なTCP 接続数を示します。確立が可能な TCP 接続数は、Nonpaged プールのサイズにより制限されます。したがって、Nonpaged プールが枯渇した状況では、新しいコネクションは確立できません。

このカウンタを用いて傾向を分析し、将来の成長に備えたシステムの適切なスケーリングについて、予測することが可能になります。平均的なネットワーク･ユーザー数をベースとして、MaxHashTableSize や NumTcTablePartitions などのTCP レジストリ･エントリを用いることで、サーバーの調整が可能になります。

### TCPv4 & TCPv6\Segments Received/sec

Segments Received/sec は、確立された接続を介して受信される TCP セグメント数の、サンプリング時間内で平均化された値を示します。そして、受信されたセグメント平均数を、接続ごとに計算することが可能です。 将来におけるユーザー数の推移につれて、増大していく負荷を予想するために、この値を用いることが可能です。接続ごとに受信されるセグメント平均数を計算するために、以下の式を使用できます：

TCPvn\Segments Received/sec ÷ TCPvn\Connections Established/sec

## Windows Server **ベースのファイル･サーバー**

これまでに説明した要素の他に、Windows Server ファイル･サーバー用のパフォーマンス･カウンタを用いて、システムをモニタリングすることが可能です。以下のテーブルに、最も重要なカウンタをリストアップします。

Table D9. ファイル･サーバー性能に関するパフォーマンス･モニターのカウンタ

|  |  |  |
| --- | --- | --- |
| Object | Counter | Instance |
| Server | Work Item Shortages | N/A |
| Server Work Queues | Available Threads | <All Instances> |
| Server Work Queues | Queue Length | <All Instances> |

### Server\Work Item Shortages

Work Item Shortages カウンタは、ファイル･サーバーがクライアント･リクエストを拒絶する原因となる、ワーク･アイテムの欠落回数を示します。このエラーが生じると、通常ではセッションが終了していしまいます。そのため、File Server サービスにおけるリソース不足を示す、主要な指標となります。

SMB（Server Message Blocks ）リクエストは、ワーク･アイテム内にストアされ、利用可能なワーカー・スレッドに割り当てられます。利用できるスレッドが存在しない場合には、このワーク･アイテムは、Available Work Items キューに置かれます。このキューが枯渇すると、サーバーは SMB リクエストを処理できなくなります。

### Server Work Queues\Available Threads

Available Threads は、SMB 入力リクエストを処理するper-processor Server Work Queue をベースにして、利用可能なスレッド数をレポートします。利用可能なスレッド数が0に達するときには、SMB 入力リクエストをキューに入れなければなりません。この値は、per-processor Server Work キューに対して定義されたワーク･スレッド数が、ボトルネックになる可能性を示す主要な指標となります。

HKLM\SYSTEM\CurrentControlSet\Services\lanmanserver\parameters に配置される、MaxThreadsPerQueue registry DWORD 値により、プロセッサごとのスレッド･プールにおけるスレッド数を制御します。デフォルトのシステムでは、プロセッサごとのスレッド･プールに、10個のスレッドが作成されます。 以下の条件が True の場合に、この値を大きくすることが可能なるはずです：

* サスティーン期間のAvailable Threadsが、ゼロもしくは、セロに近い場合。
* 待ちリクエストのQueue Length が、5以上の場合。
* 対象となるプロセッサ･インスタンスの% Processor Time が、80 以下の場合。

利用可能なスレッドを増やすことで、別のワーク･アイテムを処理できるようになりますが、余分なスレッドを追加することで、システムに過大な負荷が生じないように注意すべきです。

### Server Work Queues\Queue Length

Queue Length カウンタは、ワーカー･スレッドが利用可能 になるのを待っている、キュー内の 入力 SMB リクエスト数を報告します。プロセッサ間の通信遅延を最小化するために、プロセッサごとに Server Work キューが配置されます。

Queue Length カウンタの値は、ファイル･サーバーの処理における、クライアント SMB リクエストの遅延を判断するための主要な指標となります。さらに、スレッドやプロセシング･リソースが不足したときに、プロセッサごとにWork Item キューがバックアップされることも示します。

このキューの値が 5 以上に増加する場合には、その原因となる基本的な理由を調査すべきです。ワーク･アイテムが不足すると、手動によるセッションの復活が必要になるため、ファイル･サーバーのクライアント･セッションは終了してしまいます。

# **謝辞**

The Solution Accelerators - Management and Infrastructure（SA-MI）チームは、このInfrastructure Planning and Design Guide for Windows Server 2008 Terminal Servicesを作成したメンバーを承認し、また、感謝を述べます。このガイドの記述および、作成、そしてテストにおいて、彼らは重責を担い、また、多くの貢献をもたらしました。

Contributors:

* Reed G. Porter - Studio B
* Fergus Stewart - Microsoft

Reviewers:

* Daniel H. Brown - Microsoft
* Charles Denny - Microsoft
* Michael Kaczmarek - Microsoft
* Robin Maher - Microsoft
* John Morello - Microsoft
* Tony Soper - Microsoft
* Tessa Wooley - Microsoft

Editors:

* Laurie Dunham - *Microsoft*
* Dave Field - *Studio B*
* Patricia Rytkonen - *Volt Technical Services*