

大規模インターネットサービスプラットフォームへの サーバ仮想化技術の適用

吉川晃平* 佐藤 匠*
佐藤慎太郎* 小杉英司*
山口能一* 川野啓一*

Server Virtualization: Cases in Internet Service Provider

Kohei Yoshikawa, Shintaro Sato, Yoshikazu Yamaguchi, Takumi Sato, Eiji Kosugi, Keiichi Kawano

要 旨

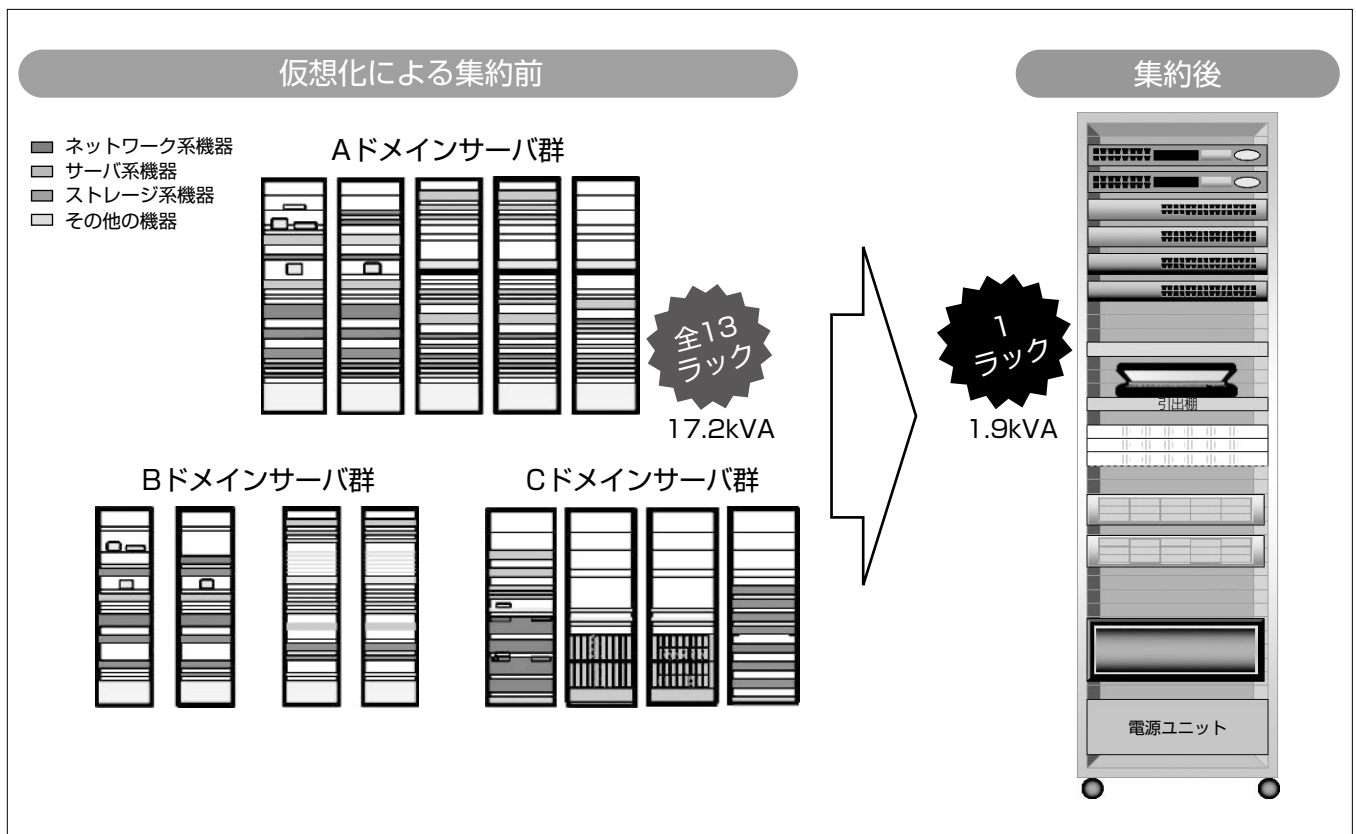
電子メールや個人ホームページのホスティングサービスを提供するインターネット事業者では、競争の激化に伴い、加入者の退出を抑えながら、更なる収益向上を図るためのコスト削減が急務となっている。このため近年では、サーバ仮想化技術を適用し、従来多数のサーバを並べていた加入者向けサービス提供システムを、数台のサーバに集約する動きも顕在化してきている。

三菱電機インフォメーションシステムズ(MDIS)では、キャリアグレードのサービス提供システムに仮想化技術を適用し、物理サーバを統合するソリューションを提供している。この事例としては、100テナント(法人加入者)分のサーバ機能を1物理サーバに集約した法人向けのマルチテナントシステムや、異なる機能の54サーバを物理サーバ2台に集約したコンシューマー向けのWWW(World Wide

Web)サービスなどがある。

これらの事例では、リソース仮想化方式を採用し、製品としてはOracle Solaris^(注1)を選定した。また、サーバマシンの台数やスペックを策定するサイジングでは、まず現システムのリソース利用状況を実測し、それを基にシステムへの要求とリソース消費量の関係をモデル化した。さらに、新システムの必要台数・スペックを算定する過程では、ハードウェア、OS(Operating System)、ミドルウェアの性能向上やアーキテクチャの違いを反映させ、仮想化によるオーバーヘッド量を見極めた上で、最大限の台数削減を図った。この結果、それぞれの事例で、サーバの台数やソフトウェアライセンス数を抑え、サービス提供にかかわるコストを大きく低減することができた。

(注1) Oracle及びSolarisは、Oracle Corp. の登録商標である。



サーバ仮想化によるコスト削減の事例

数十万加入者を収容するサービス提供設備で、サーバ54台を2台に集約した。初期構築コストだけでなく、電源容量、設置容積、さらには機器台数の削減(シンプル化)によって、運用維持コストの軽減にも貢献している。

1. ま え が き

インターネット上で提供されるサービスの加入者は、魅力的なコンテンツに対して相応の対価を支払う反面、アクセス回線やインターネット接続サービスなどの汎用的なサービスに対しては、極力、安価なサービスを求めている。このため、電子メールや個人ホームページのホスティングサービスを提供するインターネット事業者では、魅力的なサービスや価格で加入者の退出を抑えながら、収益確保に向けたコスト削減が急務となっている。

一方で、情報技術の領域では、ハードウェアの性能向上やミドルウェアの充実によって、サーバ仮想化の技術が進展・定着してきており、従来は多数のサーバ機器を並べていたセンター側のサービス提供設備を、数台のサーバ機器に集約して構築することが可能になっている。これによって、サービス提供コストの削減は図れるが、サーバ機器1台への依存度が高まるため、障害発生時の可用性確保のほか、高負荷時の影響を極力抑えるなど、サービスレベルを維持するための対策が重要となる。

本稿では、このようなサーバ集約を実現するための仮想化技術について、適用上のポイントを述べる。

2. 仮想化方式の選定

仮想化は、ハイパーバイザ型に代表される“物理仮想化”が主流であるが、これ以外の方式として、単一のOS上でメモリ、プロセス、ファイルシステム等のリソース管理情報を分割することで実現する“リソース仮想化”がある。

物理仮想化は、ゲストOSへの影響が少なくマイグレーション(新たな設備へのサービスの移行)が容易な点がメリットである。

また、リソース仮想化は、物理マシンをエミュレートする必要がないため、仮想化による性能オーバーヘッドが少なく、ゲストOSごとのライセンス費用がかからない点がメリットである。

5章で述べる事例のシステムでは、リソース仮想化を採用し、Oracle Solarisのリソース仮想化機能であるSolaris コンテナを使用したマイグレーションを実施した。この理由は次のとおりである。

- ①ホストOSのライセンスのみでゲストOSを無制限に使用でき、ライセンスにかかわるコスト低減効果が大きい。
- ②仮想化オーバーヘッドが5%以下と低い。特に、ネットワーク性能に対するオーバーヘッドがない(事前の試使用結果から)。
- ③ホストOS、ゲストOS、及び仮想化機構を一つのOSで実現することができるシンプルな構成であり、障害時の切り分けや対処が容易である。

3. サイジング

仮想化によるマイグレーションを行うに当たっては、現設備の各機器(例えばWebシステムの場合、Webサーバ、データベース(DB)サーバ、スイッチ、負荷分散装置、ストレージ等)のリソース情報を基に各機能・部位の負荷を机上で試算し、新設備に必要なリソースを導出した(図1)。

(1) 現設備のリソース利用状況測定

新設備の性能要件を明確にするため、事業者の運用部門にも協力を得て、現設備のリソース利用状況を測定した。取得対象は機器のプロセッサ、メモリ、ディスク使用量、ディスクアクセス量、ネットワークスループット等のリソース情報と、HTTP(Hyper Text Transfer Protocol)リクエスト数、DBトランザクション数、NFS(Network File System)オペレーション数等の負荷に関連した要求情報である。実際の利用状況をより的確に把握するため、測定は最繁忙時間帯に実施した。

(2) 現設備のトラフィックモデル策定

取得した稼働データを基に、要求情報とリソース消費の関係を示すトラフィックモデルを策定した。モデル策定上の考慮点は次のとおりとした。

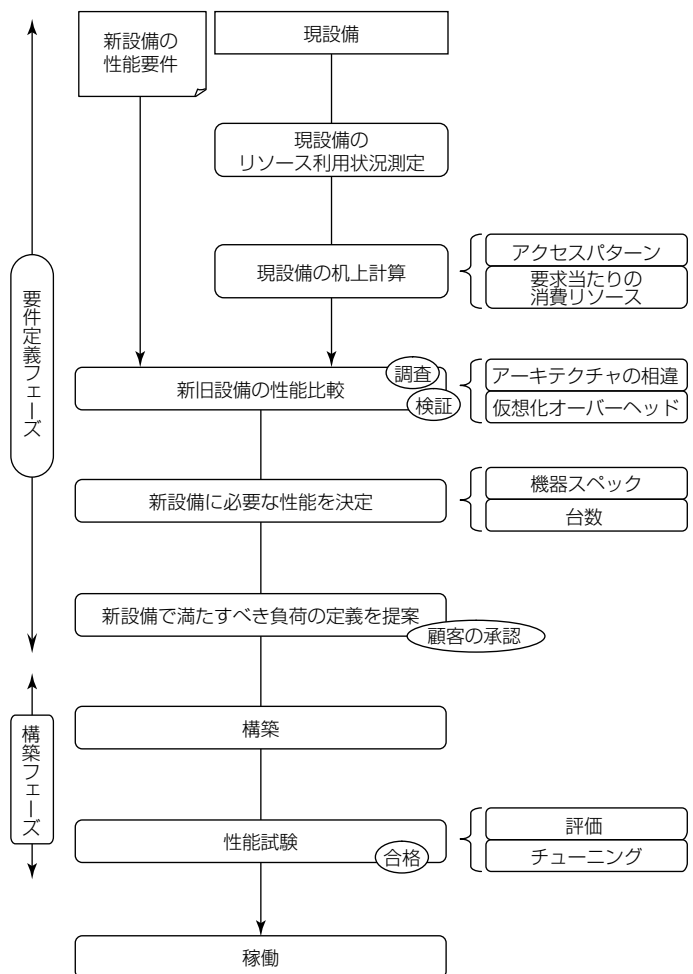


図1. 仮想化技術の導入プロセス

①外部からシステムへのアクセスを調査し、リソース消費傾向の似た要求(例えば、応答データサイズの量など)で分類した。外部からのアクセス要求ごとに、機器内に起こる処理(他機器への要求)を推定し、それによってフロントエンド機器からバックエンド機器まで連鎖するアクセス要求と消費リソースの関係をモデル化(算式化)した。

②バックエンドの機器(ファイルサーバ、ストレージ、ネットワーク装置等)については、アクセス要求ごとの消費リソースの測定が困難なため、機器間を流れるアクセス要求の平均データ量やリクエスト数、キャッシュヒット率等を推定し、実際の機器のリソース情報・負荷情報と比較し矛盾が起こらないように、繰り返し調整しマッチングさせた。

③システムには、性能向上のための様々なキャッシュ機構(仮想メモリ、ファイルシステム、プロトブロックデバイス等)が介在しており、キャッシュヒット率が変化することで負荷部位が移動してしまう。このため、各キャッシュ機構の仕組みと有効度(キャッシュヒット率)をモデル化し、係数を決定した。

(3) アーキテクチャと仮想化オーバーヘッドの考慮

仮想化によるマイグレーションでは、通常、現設備での消費リソースの積算によって新システムの必要スペック・台数を算定するが、現設備と新設備では、ハードウェア、OS、ソフトウェアのいずれでも、性能の差異やアーキテクチャの違いがある。事例のシステムでは、この違いを反映させ、かつ、仮想化によるオーバーヘッドを考慮している。具体的には、次のような工夫を行っている。

①プロセッサは、現システムの使用クロック数の積算値×余裕度が新システムの要件のベースとなるが、クロック当たりの処理性能向上も考慮した。例えば、インテルの公称ベンチマークによるとXeon^(注2) 5500世代は前世代と比較しクロック当たりの処理性能は1.7~2.8倍に増加⁽¹⁾⁽²⁾しており、これを反映させた。

②ディスク、メモリについては、現設備のリソース使用状況から見て冗長度の高いリソースを削減するとともに、キャッシュ機構の活用を図ることで、更なる縮減ができないか検討した。例えば、サーバのメモリを増強し、外部ストレージへのアクセスを減らすことで、高価なストレージのスペックを下げた。

③ハードウェアの処理能力から仮想化オーバーヘッド5%を差し引いた95%を新設備の処理能力とした。

これらの検討の過程では、機器調達コストと改修コストのトレードオフも考慮し、ランニング費用を含めたライフサイクルコストの試算も行った。

(注2) Xeonは、Intel Corp. の商標である。

4. 性能検証による妥当性評価

これらの検討によって新設備の仕様を決定し構築した。しかし検討はあくまで机上試算であり、最終的には稼働前に実施する性能検証試験で、構築した実システムに対してモデルに基づくサイジングの妥当性評価を行った。

性能検証試験では、ボトルネック部位の動作を実環境と類似な環境にするため、各機器内のキャッシュヒット率を考慮したサンプルデータ、及び負荷パターンを作成することがポイントであり、この設計に注力した。特に、連続試験の場合には、キャッシュが飽和してキャッシュヒット率が安定化する(キャッシュアウトが均衡する)状態を作る必要があり、これを実現するのに十分なサンプルデータを用意した。

検証試験の結果に基づいて、次のような評価を行い、マイグレーション後の構成でもサービスレベルの維持に問題ないことを確認した。

(1) 性能要件の評価

システムが新設備の最繁時の負荷で安定稼働することを確認し、また各機器のリソース余裕度を評価した。枯渇気味のリソースについては、プロセッサやメモリ割当て、アプリケーション等のパラメータを調整し、ボトルネック部位を分散させて余裕を確保した。

(2) 障害時の評価

最繁時の負荷をかけた状態で特定機器に擬似的に障害を発生させ、システム全体の動作を確認した。仮想化で集約したシステムでは、物理サーバの障害が多重障害として表面化する場合があり、この際のサービス停止時間、性能劣化時間の評価とともに、障害回復に必要なインシデント管理(多重障害から障害部位を特定、回復)のための運用性評価も合わせて実施した。

(3) 連続稼働の評価

高集約システムでは、メモリリークも集約(積算)されるため、長期間の安定稼働を確認する必要がある。24時間365日の自動運転を前提とするシステムでは、この確認が不可欠であり、先に述べた机上試算をベースに、平均的な負荷を長時間かけ、安定した応答・スループットを返答し続けることを確認した。

(4) 限界性能の評価

要求性能の数倍の負荷を投入し、システムの限界性能と限界時の挙動を確認した。インターネット上のサービスの場合、想定値を超えた大負荷が発生する可能性があり、そのときに仮想サーバ、アプリケーション等の設定が適切に働き、システムへの影響を最小限に抑制することが求められる。なお、確認した限界性能は、今後の増設計画(キャパシティ・プラン)の指標としても重要なデータとなる。

5. 商用構成事例

5.1 事例1：異なる機能の54サーバを2台の物理サーバに集約

対象となったWWWサービス用設備は、①コンテンツ閲覧者にHTTPやCGI(Common Gateway Interface)等のサービスを提供するフロント側機能、②コンテンツ公開者を管理するバックエンド側機能、③コンテンツを格納する大容量ストレージで構成される。また一部機能の異常なリソース消費がシステム全体に影響しないよう、機能ごとに物理サーバを割り当てていた。新設備では機能ごとにゲストOSを構築してリソース使用上限を設けることで独立性を確保し、同一物理サーバに全機能の集約を実現した(図2)。

(1) 機器構成上の考慮

コンテンツ閲覧者へサービスを提供するフロント側機能(従来はN+1冗長構成のサーバに搭載)だけでなく、システムのバックエンド側で稼働する加入者情報を管理する機能(従来は二重系構成のサーバに搭載)も合わせて集約した。フロント側機能とバックエンド側機能は、対応するネットワークインタフェースを分けることでネットワーク性能及びセキュリティを確保している。また、プロビジョニング機能が稼働する仮想サーバのみ異なる物理サーバで二重化することで、集約前と同様の可用性を確保している。

(2) 拡張性の確保

仮想サーバに搭載するゲストOSの構成は、2台の物理サーバで同一とした。これによって、これまで別々の物理

サーバに存在していた機能群を同一の物理サーバに集約し、あたかも専用のアプライアンスサーバのように動作するシンプルなシステム構成を実現した。この構成によって、複数台のサーバの構築がマスタサーバからの単純な複製(ディスクイメージの複製)で生成可能となり、工期の短縮だけでなく、試験済みマスタサーバ複製による品質向上も実現した。実際に、サービス開始後のサーバ追加工事では、収容する加入者数を2倍としたにもかかわらず、構築コストは初期構築時に比較して70%削減を実現した。

5.2 事例2：法人100社を1台の物理サーバに集約

各ユーザー企業内で個別に稼働していたDHCP(Dynamic Host Configuration Protocol: IP(Internet Protocol)アドレスなどネットワーク情報をクライアント機器に動的に割り当てるプロトコル)サーバの機能を集約し、ネットワークサービスとしてアウトソースで提供するシステムとして再構築した。センター設備への集約にあたっては、ユーザー企業の管理者が、従来どおり自社のDHCPサーバとして運用できることが不可欠要件となっていたが、仮想化技術によって、仮想サーバ単位でDHCPサーバを独立に運用することが実現できている(図3)。

(1) サイジング

限られた物理リソースを用いて100ゲストOSを集約するため、ゲストOSのリソース使用量を最小限とする必要があった。そこで、ゲストOSのDHCPサーバ機能提供のみの最小構成化、運用データ管理GUI(Graphical User Interface)でのフレームワーク利用、ファイルベースDBの採用、

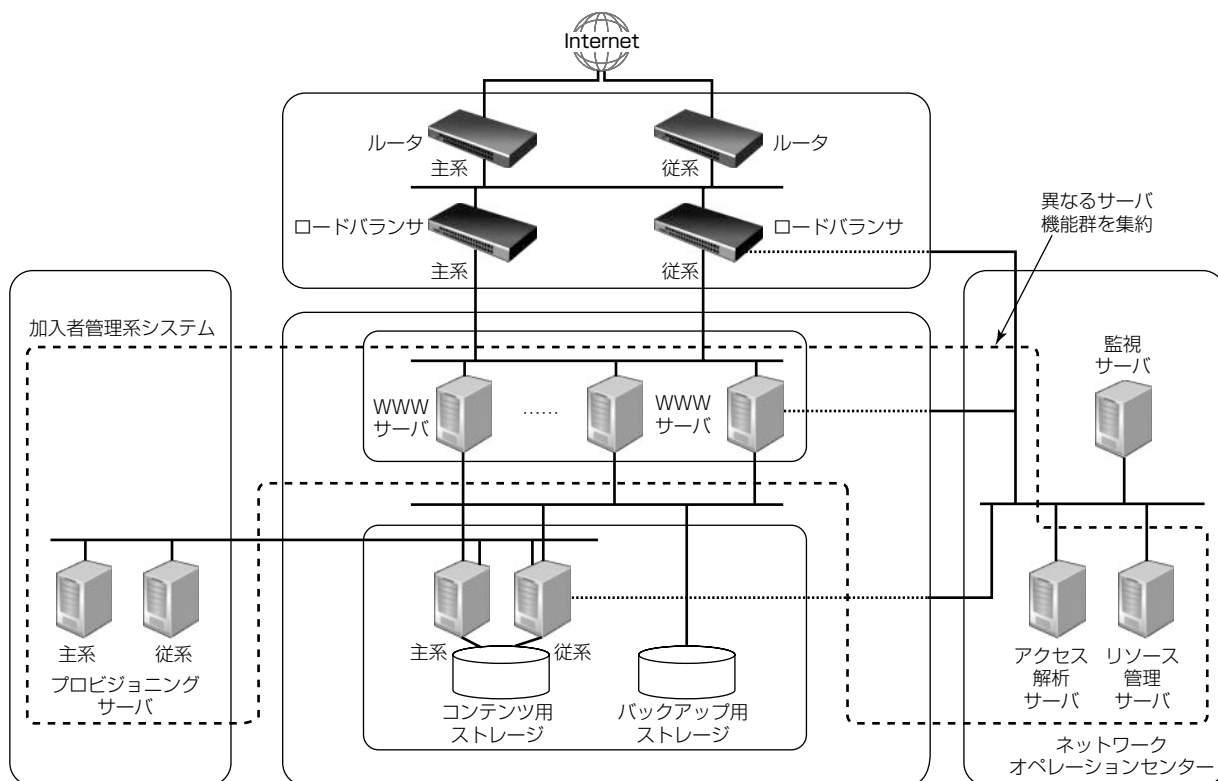


図2. 大規模WWWサーバ設備の集約

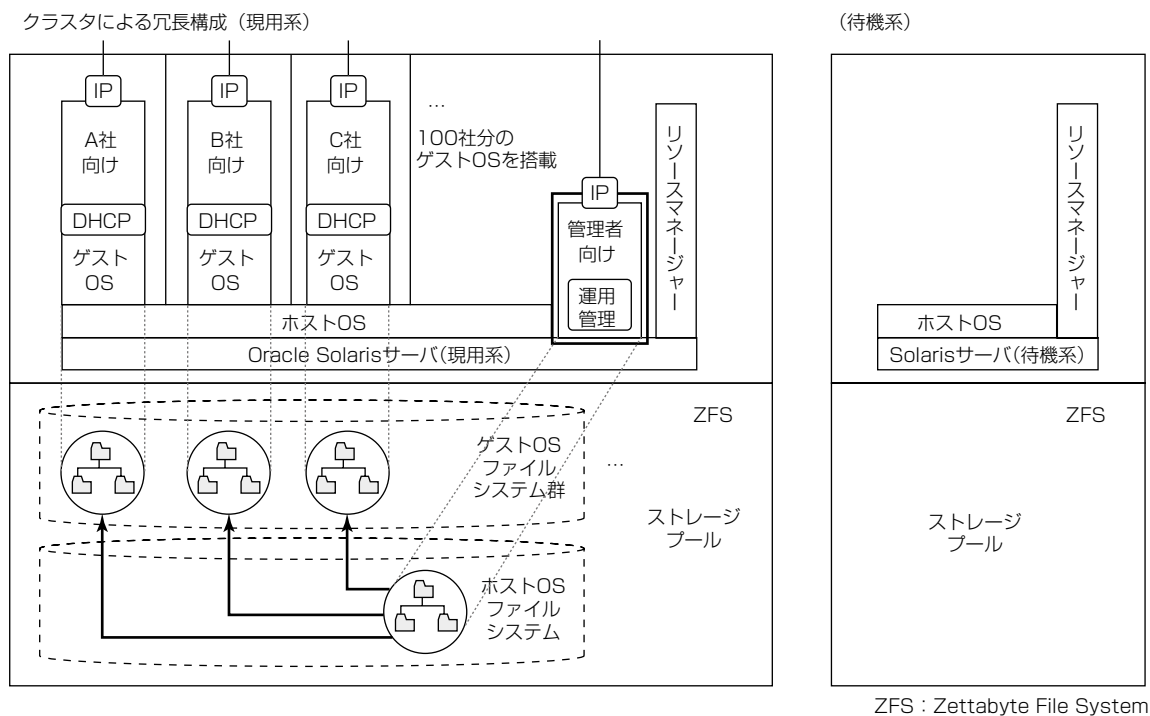


図3. 法人100社の集約

サービス正常性監視処理のホストOS集約化を行い、1ゲストOSあたりのメモリ使用量を、1物理サーバ分のリソースに集約可能な100MB以下に制限した。また、各ゲストOSにリソース使用上限を設定し、一部のゲストOSの異常なリソース消費がシステム全体に影響しないようにした。

(2) 可用性の確保

DHCPサーバに異常があると、クライアント機器はIPアドレスを取得できず、ネットワークに接続できなくなるため、ユーザーの業務への影響は極めて大きい。24時間365日の安定したサービスを提供するため、物理サーバは現用系と待機系の二重系構成とした。現用系サーバの異常発生時には、現用系サーバで稼働中のゲストOSを停止後、待機系サーバ上でゲストOSを起動してサービスを再開する。ゲストOSは、DHCPサーバ機能提供のみの最小構成としているため、起動や停止に要する時間は短縮されており、異常検知から100ゲストOSすべてが待機系サーバでサービス再開するまでの所要時間を5分以内に抑えている。

(3) ネットワークの独立性確保

各ユーザー企業の運用データの独立性、及びネットワークの独立性を確保するため、ゲストOSごとに異なるタグVLAN（Virtual LAN、標準規格はIEEE802.1Q）インタフェースを付与した。ネットワークの独立性を確保することで、例えば、同様のDHCPサーバ運用データ（192.168.100.0/

24セグメント等）で複数ゲストOSを運用する場合でも、ゲストOS間の干渉を防ぎ、セキュリティの確保を実現している。また、タグVLANの利用によって、100ゲストOS分のネットワークインタフェースを物理インタフェースカード1枚に集約している。

6. む す び

本稿で述べたようなサーバ仮想化に対しては、OSS（Open Source Software）を採用する機運も高まっている。MDISは、OSSによるサーバ仮想化にも積極的に取り組み、インターネットサービスに係るコストの更なる削減を目指す。

参 考 文 献

- (1) SAPジャパン(株)：仮想化テクノロジーを活用した基幹システムのディザスタリカバリ対策、http://download.intel.com/jp/business/japan/pdf/N035_COIL.pdf
- (2) インテル(株)：検証事例インテルXeonプロセッサ5500番台クラウド・コンピューティングの基盤検証、<http://download.intel.co.jp/jp/business/japan/pdf/322095-001JA.pdf>