

仮想化と SDx

平成 26 年 6 月 18 日

ネットコマース株式会社

目次

仮想化とは	4
仮想化の本当の意味	4
仮想化の 3 つのタイプ	4
● 分割(パーティショニング/PARTITIONING).....	4
● 集約(アグリゲーション/AGGREGATION).....	5
● 模倣(エミュレーション/EMULATION).....	5
仮想化誕生の歴史	5
仮想化の歴史	6
仮想化と SDx.....	8
<u>仮想化から SD(SOFTWARE-DEFINED)化へ.....</u>	<u>8</u>
仮想化についての一般的理解	8
SOFTWARE-DEFINED INFRASTRUCTURE とは何か.....	9
<u>仮想化の種類と仕組み</u>	<u>10</u>
仮想化の種類	10
サーバーの仮想化	11
デスクトップの仮想化	12
アプリケーションの仮想化	14
クライアントの仮想化	16
ストレージの仮想化	18
ネットワークの仮想化	21
<u>サーバーの仮想化 そのメリットと課題.....</u>	<u>23</u>
物理的資源の削減	23
ソフトウェア管理への移行	24
ライブマイグレーション	24
BCP 対策・仮想マシン・レプリケーション	25
サーバーの仮想化／課題	25

本書は、下記クリエイティブコモンズライセンスに準拠します。



表示・非営利・改変禁止

原作者のクレジット(氏名、作品タイトルなど)を表示し、かつ非営利目的であり、そして元の作品を改変しないことを主な条件に、作品を自由に再配布できる。



仮想化とは

仮想化の本当の意味

「仮想」という言葉を聞くと、「虚像の」、「実態のない」という意味を思い浮かべてしまいます。ところが、この言葉の元となった英語の「Virtual」は、どうもそういう意味ではないらしいのです。調べてみると、「(表面または名目上はそうでないが)事実上の／実質上の／実際の」という意味があるようです。また、ラテン語の語源を見ると「力のある」と記されています。辞書を引くと英語の文例には、次のような記述がありました。

It was a virtual promise.

(約束ではないが) 実際には約束も同然だった。

He was the virtual leader of the movement.

彼はその運動の事実上の指導者だった。

このように見ていくと、私たちが IT の用語として使っている「仮想化=Virtualization」は、次のような意味になるのでしょうか。

「物理的実態とは異なるが実質的機能を実現する仕組み」

仮想化は決して、「虚像で実態のないシステムを作り出す仕組み」ではないということになります。

仮想化の3つのタイプ

さて、仮想化というと「物理的実態を分割する仕組み」というとらえ方もされますが、決してそれだけではありません。「分割」、「集約」、「模倣」の3つのタイプに整理することができます。

- **分割(パーティショニング/Partitioning)**

ひとつの物理資源を複数の仮想資源に分割する仮想化。ひとつの物理資源を複数の仮想資源に分割するサーバーの仮想化や仮想 LAN(VLAN)

などがこれに相当します。

- **集約(アグリゲーション/Aggregation)**

複数の物理資源をひとつの仮想資源に分割する仮想化。複数の仮想資源をひとつの仮想資源と見せかけるストレージの仮想化がこれに相当します。

- **模倣(エミュレーション/Emulation)**

ある物理資源を異なる資源に見せかける仮想化。物理的な実態とは違うシステム資源を作り出すもので、Java 仮想マシンやデータベースの仮想化などが、これに相当します。

仮想化誕生の歴史

仮想化は、どのような経緯で生まれてきたのでしょうか。

1950年代から1960年代にかけての黎明期のコンピューターは、大変高価なものでした。そのため、このコンピューターを多くのユーザーと共用する方法がいろいろと考えられました。そのひとつが、「バッチ」です。ひとつひとつの処理単位(ジョブ)が、順次実行される処理形態です。この方法では、前の処理が終わらなければ、次の処理に移ることはできません。これでは不便だと言うことらになり、生まれたのが、「タイムシェアリング」です。

タイムシェアリングとは、1台のコンピューターのCPU処理時間を極めて短い時間単位で分割し、複数のユーザーに順次割り当て、見かけ上複数のユーザーが同時にコンピューターを利用できるようにすることでコンピューターの共用を実現しようとしたのです。1961年、IBM 709でタイムシェアリングが実装され、さらに使いやすいシステムの実現を目指し、MITでMULTICS (Multiplexed Information and Computing Service) プロジェクトがスタートとしたのです。MULTICSは必ずしも成功とは言えなかったのですが、この考えは受け継がれ、後のUNIXへとつながります。

さて、分割したCPUの処理時間は、ユーザーにとって独立した個別のCPUと見做されます。ならば、これにユーザー個別のオペレーティング・システムを動かせるように発展させることで、今でいう仮想化が生まれました。1967年、

IBM System/360 モデル 67 で実装された CP-67/CMS が商用ベースでは、初めてのものでされています。また、その時始めて、Virtual Machine (VM) という言葉が使われました。

このようにして、仮想化が誕生したのです。

仮想化の歴史

ここで改めて、仮想化の歴史を振り返ってみましょう。

- 1964 年以前

メインフレームが生まれる 1964 年以前、コンピューターは業務にあわせた専用機として存在していました。企業は、業務毎にその専用機を使わなくてはならず、分散システムの時代でした。また、ハードウェアは高価であり、複数の機器を利用しなければならず、その運用負担やコストの増大は課題となっていました。

- 1964 年

この年に、メインフレームの先駆けである IBM System/360 が誕生しました。「汎用機」とも呼ばれ、異なる業務もこれ一台で対応できる万能機としてその存在感を示しました。浮動小数点演算も可能であり、事務計算ばかりでなく、技術計算もこれ一台でこなすことができました。ただし、異なる業務での共用は、当初はバッチ処理でした。その後、タイムシェアリング使えるようになりましたが、同じ OS 上に複数のアプリケーションを稼働させる処理形態をとっていました。

- 1967 年

仮想化の技術は、この年以前からもいろいろと研究されていましたが、1967 年 IBM System/360 モデル 40 用に研究用の仮想化 OS として CP-40 が開発され、これをベースに System/360 モデル 67 用に CP-67 が開発され始めて商用製品として発売されました。1972 年 仮想記憶をサポートした System/370 シリーズ用に CP-370 が開発され本格的に普及するようになったのです。

この仕組みにより、高価であったハードウェアをユーザー個別の専用機として「分割」することで、利用の自由度を高め、見かけ上のマシン・

コストの削減を図ろうということが、この当時の仮想化の目的となっていたのです。

- 1980年代～

安価な小型のコンピューターの出現により、何でもメインフレームの時代から、ダウンサイジング、分散の時代へと移っていきました。メインフレームほどの能力や機能はிரらない、その代わり、集中システムであるメインフレームの運用上の制約を受けることなくもっと自由に使用したいとの思惑から、高価なメインフレームを仮想化で「分割」するのではなく、低価格の専用のハードウェアを購入した方が便利であるという考え方が広がり、「分散」がすすんでいったのです。その一方で、企業が抱えるコンピューターの台数は増大し、TCOの増大を招くこととなりました。

- 1999年

この前年の1998年にVMware社が設立され、翌年の1999年に初めての製品、VMware Workstationが発売されました。x86ベースの仮想化ソフトウェアで、これ以前も研究目的では存在していましたが、商用ベースでは初めてのものでした。当時は、WindowsとLinuxに対応し、主にソフトウェア開発・評価・テストを目的とするものでした。その後、機能も拡張され適用範囲が広がってゆきました。また、VMware以外にも多くの仮想化ソフトウェアが出現し、現在に至っています。

この時代の仮想化は、安価な物理マシンの増大によりTCOが増大、そのことが企業のIT予算を圧迫するに至り、仮想化により物理マシンを「集約」することでTCOを削減したいという思惑から仮想化が広く使われるようになったのです。

このように歴史を振り返れば、仮想化そのものの仕組みは、誕生した当時から、基本的には変わっていません。しかし、その目的は、大きく変わりました。当初は高価なシステム資源を「分割」し、ユーザー毎に安価なシステムを提供することでしたが、昨今はシステム資源が安く手に入るようになり、その結果、台数が増え、それに伴うTCOを削減するための「集約」へと変わってきたのです。

仮想化と SDx

さらに、仮想化は次のステージへと移り始めています。

当初、仮想化は、単一のシステム資源を仮想化し、その組合せを動的に定義できる仕組みでした。

しかし、それら仮想化させた様々なシステム資源をとりまとめ、システム資源全体の組合せを動的に定義できる仕組みへと発展しつつあります。この考え方が、SDx(Software-Defined:ソフトウェアで定義される)と言われるものです。

言葉の意味だけを考えれば、従来の仮想化も SD といえるでしょう。ただ、昨今では、従来の仮想化を包括する概念として、この言葉が使われるようになっていきます。SDS(Software-Defined Storage)、SDN(Software-Defined Networking)、SDI(Software-Defined Infrastructure)、SDDC(Software-Defined Data Center)、SDE(SDI(Software-Defined Environment)などの言葉が使われるようになりました。

インフラストラクチャーをSD化する一連のソフトウェアが、「クラウドOS」と言われることがあります。厳密に言えば、OSとは異なる仕組みですが、ハードウェアを抽象化し、ハードウェアとアプリケーションとのインターフェースを提供する仕組みとしての共通点から、このような言葉が使われることがあります。これは、クラウド・コンピューティングのIaaSに相当する機能を提供するもので、VMwareのvCloudやOSSのCloud Stack、Open Stackがこれに相当します。

仮想化から SD(Software-Defined)化へ

仮想化についての一般的理解

仮想化の技術を使うことで、次のふたつの価値をユーザーは享受することができます。

- 物理的資源の削減

物理マシンの台数を減らすことができます。これにより、システム資源の稼働率を高め、システム資産の購入を減らすことができます。また、

電量消費を削減し、電気料金や CO2 の排出量を削減できます。

- ソフトウェア管理への移行

システム資源の調達や構成の変更をソフトウェアによる設定作業として行えるようになります。これにより、物理的な作業を行うことなく、システムの追加、削減、複製、構成の変更の変更ができるようになります。

Software-Defined Infrastructure とは何か

仮想化の価値のひとつである「ソフトウェア管理への移行」を様々なシステム資源に広く適用した概念が SDI です。

システムを構成する要素は、サーバー、ストレージ、デスクトップ PC、ネットワークなどですが、これらシステムの物理資源をまとめて抽象化し、ソフトウェアによる設定作業で、その組合せや構成をおこなうことができるようになりました。対象は、特定のシステム資源だけ、例えば、サーバーだけ、あるいは、ストレージだけではなく、その組合せ全体を対象としています。

この仕組みにより、次の 3 つの価値をユーザーは享受することができます。

- スピードとアジリティ

システム・リソースの調達や構成の変更を、物理作業を伴わず柔軟・迅速にできます。

- 機器構成の変更や障害による影響の最小化

物理リソース（リソース・プール）での構成変更や障害による影響を回避、または、局所化できます。

- 運用の自動化

物理的な作業を伴わずソフトウェアの設定を変えることで、システム全体の構成変更や運用管理ができます。

SD という言葉が、使われるようになった背景には、SDN(Software Defined Networking)への注目があります。

SDN とは、ネットワークの構成、機能などの設定や変更をソフトウェアの操作だけで動的に行える仕組みあるいはコンセプトを指す言葉です。

従来、ネットワークにサーバーを追加する場合やひとつのネットワークを複数に分割する場合には、ネットワーク管理者がケーブルの接続や機器の設置などの物理的な作業や、ルーターやスイッチなどのネットワーク機器を個別に設定しなければなりませんでした。

しかし、サーバーの仮想化が普及するに伴い、ネットワーク上に多数の仮想サーバーが、生成や消滅を繰り返し、物理マシン間の移動（ライブマイグレーション）も頻繁に行われるようになりました。しかし、ネットワーク管理者は、その都度、ケーブルの接続やネットワーク機器の設定を行わなくてはならず、その運用負担が大きくなっていったのです。

この事態に対処するために、もっと柔軟に、自由に、そして、さらには、自動的に、ネットワーク構成、あるいは、性能(QoS : Quality of Service)を設定、変更できる仕組みが求められるようになりました。

そんな背景の中で、「OpenFlow」と呼ばれる新しいネットワーク標準が登場し、SDN の実現に向けた動きが加速しました。これを切っ掛けに、SD という言葉が広く知られるようになり、他のシステム資源にも SD という言葉が使われるようになったようです。

また、従来の VLAN(仮想 LAN)では、「システム資源(この場合はネットワーク資源)」のソフトウェアによる動的な構成や性能の設定や変更に制約が多く、他のシステム資源と同じようにネットワーク資源を扱うことは困難でした。それが、SDN の登場と標準化により、ネットワーク資源も含めたシステム資源全体の SD 化が可能になり、SDI が注目されるようになったのです。

仮想化の種類と仕組み

仮想化の種類

仮想化をその機能や特徴から分類すると、以下の 6 つの種類に区分できます。

- サーバーの仮想化
- デスクトップの仮想化
- クライアントの仮想化
- アプリケーションの仮想化
- ストレージの仮想化

➤ ネットワークの仮想化

さらにその実装方法の違いにより、さらに細分化できます。全てを網羅することはできませんが、主要なものについて、みてゆきましょう。

サーバーの仮想化

ふたつのタイプに分類してみました。ひとつは、「ハイパーバイザー方式」、もうひとつは、「コンテナ方式」です。

● ハイパーバイザー方式

「ハイパーバイザー」と言われる仮想化のためのソフトウェアを使い、このサーバーのハードウェアと同じ振る舞いや動作をする仮想サーバーを生成します。

このハイパーバイザーは、ソフトウェアの3階層(オペレーティング・システム(OS)、ミドルウェア、アプリケーション)とは異なる位置づけのもので、オペレーティング・システムと入れ替わり、ハードウェアの上で直接動作します。このようなソフトウェアのことを「ハイパーバイザー」と呼んでいます。

OS からみれば、仮想マシンは実態のあるサーバー(仮想マシンに対して物理マシンと言います)に見えます。ですから、仮想マシンごとに OS を動かすことができるのです。

ユーザーからみれば複数の OS が独立して動いているように見えますから、あたかも独立したサーバーがそこに存在しているように見えるのです。

仮想マシンは、それぞれ個別の独立したサーバーとして振る舞います。ですからそれぞれに異なる種類の OS (例えば、Windows Server や Red hat Enterprise Linux など) を稼働させることができます。つまり、一台の物理マシンに複数の異なるオペレーティング・システムを同時に動かしていることになります。

サーバーの仮想化を利用することで、一台あたりのサーバーの稼働率を高めることができます。これにより、物理マシンの稼働率を上げることで、サーバーの台数を減らし、設置面積の削減による設備料の削減、消費電力の削減による電気料金やCO2の排出量を減らすことができます

具体的な製品には、VMware ESXi、Xen Server、Hyper-V、KVM などがあります。

- コンテナ方式

目的や期待される効果はハイパーバイザー方式とほぼ同じですが、同じ OS しか使えないことが大きな違いです。

1つのカーネルで OS の複数のユーザー空間(インスタンス)を生成します。このユーザー空間がコンテナです。ユーザー毎に独立した各コンテナは完全に独立しているため、あるコンテナが別のコンテナのファイルにアクセスすることができません。また、全てが同じカーネル上で実行されるため、リソースの利用効率は非常に高く、カーネルにパッチを当てる際もひとつで済むことなど運用管理負担が少なくなります。

具体的な製品には、Docker、LXC(Linux Containers)、Parallels Virtuozzo Containers などがあります。

デスクトップの仮想化

デスクトップの仮想化には、「仮想 PC 方式」と「ブレード PC 方式」があります。

- 仮想 PC 方式

サーバーの仮想化と同様、ハイパーバイザーで仮想マシンを生成します。サーバーの仮想化と異なるのは、作られた仮想マシンがクライアント PC(仮想 PC)であり、その上で動くオペレーティング・システムは、Windows7 や Windows XP といったクライアント PC 用のものであることです。

ユーザーは、自分の仮想 PC にネットワークを介してこの仮想マシン上のオペレーティング・システムが作るディスプレイ画面（これをデスクトップ画面と言います）を見ることができます。また、手元にある PC のキーボードやマウスの操作はそのままサーバー上の仮想 PC に送られ、そこで処理されます。

デスクトップの仮想化とは、本来ならば自分の手元にあるクライアント PC 上で動かすはずの様々なソフトウェアをサーバー上で動かす技術

なのです。このタイプ(仮想 PC 方式)のデスクトップの仮想化は VDI (Virtual Desktop Infrastructure) と呼ばれています。

このような使い方をすれば、クライアント PC 側に OS やアプリケーションを導入する必要はありません。そこで、ハードディスクを持たず必要最小限のメモリーやプロセッサでも必要なソフトウェアを利用することができます。そこで、VDI の利用を前提として機能を最小限に絞ったクライアント PC が作られました。これをシンクライアント (Thin Client) と言います。さらに、タブレットやスマホからその画面を表示させ利用できるようになっています。

この技術により、散在するクライアント PC の運用管理を一カ所で行えるようになり管理に伴うコストを削減できるようになります。

また、プログラムやデータはすべてサーバー上にあるので、クライアント PC が盗難してもセキュリティは安心です。また、自宅で仕事をする場合、自宅の PC からネットワークを介して会社で使っている PC のデスクトップを呼び出せば同じ環境をそのまま引き継いで、仕事をするのが可能になります。

また、社員でなくなったらこれまで使っていたデスクトップの使用を管理者が停止できます。

東日本大震災では多くのクライアント PC が破損し使えなくなりました。また、交通手段が絶たれることや、節電のために自宅での作業の必要性が求められています。デスクトップの仮想化はこのような時にも仕事を継続するための手段として、注目されている技術でもあるのです。

この方式に対応した製品としては、VMware Horizon View、Citrix XenDesktop、Microsoft VDI などがあります。

- ブレード PC 方式

仮想 PC 方式と同様にデスクトップ画面を、ネットワークを介して転送するコトでは同じ出すが、その画面を作っている PC が仮想マシンではなく、それぞれのユーザーごとに割り当てられた「ブレード PC」と呼ぶ物理 PC を使っている点が大きく異なります。利用者は、自らに割り当てられたブレード PC をネットワーク経由で操作することになります。それぞれのユー

ザーが、物理的に割り当てられた専用の CPU やメモリーを独占できるので、アプリケーションの制約や、他の利用者との競合による影響をうけることはありません。ただし、ユーザーの人数だけ、ブレード PC を用意する必要があるために、コストがかかります。

最近では、CPU の性能や仮想化ソフトウェアの性能の向上により、仮想 PC 方式が主流となりつつあります。

ちなみに、ここに紹介したふたつの方式、および次に説明する「アプリケーションの仮想化（画面転送方式）」のように、端末側の処理を画面表示や入力などの最小限にとどめ、サーバー側に処理を任せる方式を総称して「シンククライアント」と呼ぶ場合もあります。そのための端末をシンククライアントという場合もあり、区別する必要があります。

アプリケーションの仮想化

アプリケーションの仮想化には「画面転送方式」と「ストリーミング方式」のふたつのタイプがあります。

- 画面転送方式

デスクトップの仮想化では、サーバー上で稼働するクライアント用のオペレーティング・システムの画面（デスクトップ）を、ネットワークを介してクライアント PC に表示します。アプリケーションの仮想化は、見かけ上このデスクトップの仮想化とよく似ているのですが、表示される画面がデスクトップ画面でなく、特定のアプリケーション・プログラムの画面だけなのです。

サーバーにインストールされたアプリケーションをネットワーク越しに遠隔で起動し、サーバーで実行されるアプリケーションの画面を転送して、ユーザー側の端末で操作する仕組みです。サーバーベース（SBC: Server Based Computing）方式とも言われています。

ユーザー毎に個別の仮想マシンを立ち上げる必要がないため、システム・リソースの負担も少なく、多くのユーザーに効率よくアプリケーションを使わせることができます。

ただし、アプリケーション・プログラムが、マルチユーザー・アクセスに対応している必要があります。Microsoft Office を含めて多くのプログラムが対応しています。但し、CPU やメモリーを共有しているため、利用者の誰かが重い処理を動かすと、同時アクセスしている他の利用者にも影響を与えてしまい、応答が遅くなることがあります。

この技術も、デスクトップの仮想化と同様に、散在するクライアント PC 上で稼働するアプリケーション・プログラムの運用管理を一カ所でできるようになり、管理に伴うコストを削減することができます。

また、プログラムやデータはすべてサーバー上に保管されますので、クライアント PC が盗難してもそれらが抜き取られる心配はなく安心です。

また、正社員と派遣社員が混在し仕事をするような場合、これまでは、使えるプログラムをひとつひとつのクライアント PC 毎に設定しなければならず、大変手間の掛かる仕事でした。しかし、この仕組みがあれば、管理者は利用の可否を一カ所で設定するだけですから、管理工数が大幅に削減されます。また、ユーザーが特定のアプリケーション・プログラムしか使用しないコールセンターや事務部門などで、安価なシンクライアント端末でこの機能を使用すれば、端末コストや管理工数が削減できます。

また、クライアント PC のオペレーティング・システムが同じ Windows 7 であっても、PC の機種により使われる部品や独自のソフトウェアの組み合わせの違いにより、必ずしもすべての PC で動かない可能性があります。この方式であれば、画面の転送とキーボードやマウスの入力だけのやりとりになりますので、機種に依存する問題は起こらないというメリットもあります。

代表的なソフトウェアとして、Microsoft Remote Desktop Service、Citrix XenApp があります。

- ストリーミング方式

「ストリーミング方式」では、まずクライアント PC に「アプリケーション実行環境 (ランタイム・モジュール)」と言われるソフトウェアを導入します。このソフトウェアは、個々の PC に最適化され導入、設定さ

れます。

ユーザーは使いたいアプリケーション・プログラムをサーバーからダウンロードします。このアプリケーション・プログラムは、実際には、アプリケーション・プログラム本体とランタイム・モジュールで動かすためのデータをセットにしたパッケージとして登録されています。

このパッケージをクライアント PC 上のランタイム・モジュールで動かします。プログラムは PC のオペレーティング・システムの上で直接動くのではなく、そこから切り離されたランタイム・モジュールの上で動きますので、機種に依存する問題は起こりません。

また、作業を終了すると、一旦ダウンロードしたプログラムは自動的に消去され、PC 上には残りません。

「画面転送方式」では、ネットワークがつながっていなければ何もできませんが、「ストリーミング方式」では、一旦プログラムをダウンロードすれば、ネットワークとの接続が切れても作業が継続できます。ネットワーク環境が不安定な場合などには、メリットのある方法です。

また、「画面転送方式」では、画面データを、ネットワークを介して送らなければなりません。3D グラフィックスなど、複雑な形状とその動きを送るとなると、その転送量は膨大になり、なめらかな表示ができなくなることがあります。このようなケースでは、手元にある PC の CPU やグラフィック・エンジンを直接使えるこの方法が有効です。また、手元の PC にデータを保管し、他の目的でも使いたいといった用途にも効果的です。

Microsoft の App-V や VMware の Thin App などがあります。

クライアントの仮想化

クライアントの仮想化は、ひとつのクライアント PC 上に複数の仮想マシンを動作させそれぞれに異なるオペレーティング・システムを同時に稼働させる技術で、「アプリケーション方式」と「ハイパーバイザー方式」に分けられます。

- アプリケーション方式

本来、ひとりのユーザーに占有されるクライアント PC に、なぜ複数

の仮想マシンを動かす異なるオペレーティング・システムを稼働させる必要があるのでしょうか。その答えは、プログラムやデータの互換性を確保するためなのです。

例えば、Window 7と言われるクライアント PC 用のオペレーティング・システム上で、アプリケーション・プログラムとして「XP モード」と呼ばれる仮想化ソフトウェアが動きます。このソフトウェアは Windows 7 の前のバージョンである Windows XP を動かすことができる仮想マシンを創り出します。

実は、Windows XP では動くが Windows 7 では動かないソフトウェアは数多くあるのですが、このソフトウェアのおかげで Windows 7 ユーザーは、Windows XP でしか動かないアプリケーションを同時に動かすことができることから、XP 用として開発、購入したソフトウェアを無駄にしないですむのです。

また、Apple の Mac OS 上で Windows を動かす仮想化ソフトウェアもあります。これを使えば、一台の Mac PC で Mac OS も Windows も動かすことができます。また、データも相互にやりとりできますので、大変便利になります。

Windows7 で動く Windows XP モード（2014 年 4 月 8 日以降サポートは終了）や Mac OS X で Windows を動かす Parallels Desktop や VMware Fusion などがあります。

- ハイパーバイザー方式

クライアント PC 上に複数の OS を稼働させる方法として、サーバーの仮想化と同様に、クライアント PC 上でハイパーバイザーを起動させ複数の仮想マシン（クライアント PC）を生成するものです。

ベアメタル方式とも言われ、OS の上でアプリケーションとして仮想化ソフトを動かすのではなく、クライアント PC のハードウェア上で直接動作するもので、OS が途中で介在しないことから、パフォーマンスやセキュリティなどの面で有利とされています。

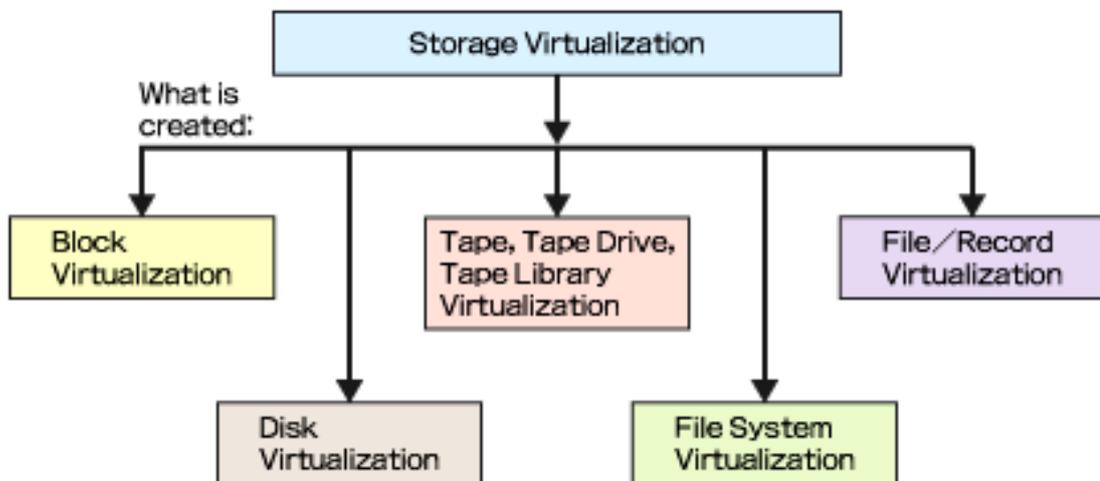
例えば、業務用と個人用の Windows7 を同時に動作させ、業務用は会社のセキュリティ・ポリシーに従い運用させるといった、個人所有の PC を業務用途で使用する BYOD（Bring Own Your Device）の用途が考え

られます。

具体的な製品としては、Citrix の Xen Client があります。

ストレージの仮想化

ストレージの業界団体である SNIA (Storage Network Industry Association) が作成した「ストレージ仮想化技術の分類」があります。ここでは、ストレージで使われる仮想化技術を次の 5 つのに分けています。



- Disk Virtualization (ディスクの仮想化)
- Block Virtualization (ブロックの仮想化)
- File System Virtualization (ファイル・システムの仮想化)
- File Virtualization (ファイルの仮想化)
- Tape Virtualization (テープの仮想化)

ストレージを構成するほぼ全てのレイヤに仮想化技術があることが、分かります。本書では、この中で、「ブロックの仮想化」と「ファイル・システムの仮想化」について解説します。

● ブロックの仮想化

サーバーにはストレージが必要です。しかし、サーバー毎に個別にストレージを持つことになると、あるストレージは 3 割、あるストレージは 2 割と、その使用量にばらつきが出てしまい無駄も多くなってしまい

ます。

そこで、これらストレージをひとまとめにして、その必要量だけを個々のサーバーに割り当てることができれば、全体として無駄を省き少ないストレージ台数でまかなうことができるはずです。これを実現するのが「ブロックの仮想化」です。

通常、ストレージにデータやプログラムを保存する場合、それらを連続したひとかたまりにまとめて保存することはありません。「ブロック」と言われる一定の容量単位に分割され、保存されます。そしてストレージのハードウェア上にばらばらに分散して保存されます。

ブロックは、「分割された内容」+「次につながるブロックの所在地を示す情報」がセットになったものですので、次の所在地をたどることで、「連続したひとかたまりの内容」を再現することができるのです。

ブロックの仮想化は、このブロックをサーバーごとに割り当てる仕組みです。

例えば、ブロックが 100 個あるストレージであれば、A サーバーに 50、B サーバーに 30、C サーバーに 20 というように割り当てることで、ストレージを使い切ることが可能になります。

実際には、ひとつのブロックの大きさは、4KB であり、128GB のストレージであれば、1,342,217,728KB ありますので、33,554,432 個のブロックに分割されることになります。

このように、ブロック単位で個々のサーバーにストレージの容量を分割して割り当てる技術が「ブロックの仮想化」です。

なお、この仮想化を実現するには様々な方法がありますが、一般的には SAN (Storage Area Network) と言われる装置が、この機能を提供しています。

ブロックの仮想化には、「容量を順次拡張」、「容易なオンラインデータ移行」、「集中化による管理負担軽減」のメリットがあります。

➤ **容量を順次拡張**

ブロックの仮想化を使い、複数の物理ストレージを統合して論理的なストレージのまとまりを作ることができます。必要となる容量が増えれば、順次物理ストレージを追加し、論理ストレージを拡張していくことができます。

システム導入当初、ストレージ容量の見通しが立たないことも多く、余裕を持って導入はしたものの、使われずに容量が余ってしまうなどのリスクを抱えることがあります。ブロックの仮想化を使えば、このような課題を解決することができます。

➤ **オンラインデータ移行**

物理ストレージ間でのデータの移行は、リスクの高い作業です。装置間でデータどうしの同期をとって装置を切り替える際に、どうしてもシステムを一旦止める必要があるからです。また、接続されているサーバー側の設定情報を全て書き換える必要もあります。そのため、綿密な計画が必要であり、管理作業の負担も大きなものでした。

しかし、ブロックの仮想化を使えば、仮想化されたストレージ装置の間で、データをオンラインで移動することができるようになります。そのため、サーバー側の設定情報を書き換える必要もなく、無停止でデータ移行を行うことができます。

➤ **集中化されたストレージ管理**

ブロックの仮想化により、管理対象を減らすことができます。個々の物理ストレージを個別に管理するのではなく、仮想化されたストレージを集中して管理できる部分があるからです。

例えば、「自律的な容量バランス機能」というものがあります。これは、管理者から与えられたポリシーにしたがって、ストレージ装置間の容量バランスを最適化するために、データを物理ストレージ間で自律的に移動するという機能で、この機能により、各物理ストレージ装の空き容量が自動調整されます。管理者が各装置の空き容量を個別に管理する必要はなくなり、ストレージのまとまり（ストレージ・プール）の空き容量だけを監視していれば良くなり、運用管理負担の軽減に貢献します。

- ファイル・システムの仮想化

複数のユーザーで同じ文書ファイルや画像ファイルを共有したいというニーズはよくある話です。これを実現するのが、「ファイル・システムの仮想化」です。

ブロックの仮想化は、特定のブロックを特定のコンピューターに割り当てます。そのため、他のコンピューターからは、そのブロックを共有することはできません。完全に占有となってしまいます。

これに対して「ファイル・システム」の仮想化は、ファイルを共有することが目的であり、方法は全く異なるものです。

この技術は、個々のストレージ装置によって個々に管理されているファイル名やディレクトリへのパス名を複数のシステムで共用できるようにするものです。論理的なファイル名やパス名を使って、複数のストレージ装置に存在するファイルが、あたかも単一のファイル・システムに存在するものとして扱えます。

このような仮想化ができると、ファイル・システム処理を複数のマシンで分散処理できるので、大きなファイル・システムを効率よく運用できます。また、ファイルを物理ストレージ間で移動する場合も、これを使用するシステム側は、仮想化された論理ファイルを使いながら、物理的な配置だけを変えるというようなことができるようになり、このファイルを使用するシステム側の作業はなくなります。

また、ファイル形式がオペレーティング・システムごとに異なってもその際を吸収し、お互いにファイルにアクセスすることが可能になります。例えば、Mac OS と Windows ではお互いに読み書きすることができません。これでは、不便なので、ファイル・レベルの仮想化では、このオペレーティング・システムごとに異なるファイル形式を変換して、相互に読み書きできるようにするための機能も備わっています。

一般的には、このような仕組みを「NAS (Network Attached Storage)」と呼び、アプライアンス製品として提供されているものもあります。

ネットワークの仮想化

ネットワークの仮想化として、「ネットワーク接続の仮想化」と「ネットワー

ク構成の仮想化」について解説します。前者は、1つのネットワーク接続（この場合は LAN）を複数の LAN に分割して、他の LAN からは分離する技術で、VLAN（Virtual LAN または仮想 LAN）と言われています。後者は、ネットワークの接続構成だけではなくネットワーク機器の構成や QoS（Quality of Service）といったネットワークの性能についても、ソフトウェアの設定で構成および管理できる仕組みです。

- ネットワーク接続の仮想化

実際の LAN は、ケーブルでつながれています。しかし、この接続先や接続する機器の組み合わせを変えようとする、その都度ケーブルをつなぎ直さなければなりません。

ネットワーク接続の仮想化は、実際にはケーブルでつながっている機器の接続をソフトウェアで管理し、設定だけでデータのやりとりできる組み合わせを変更することができる技術です。これにより接続変更に伴う作業の手間を削減することができます。

- ネットワーク構成の仮想化

物理的なネットワーク機器の構成から、ソフトウェアの設定により必要な構成を組むことができる技術です。

これを実現する技術や考え方の総称は、SDN（Software Defined Networking）と呼ばれています。

これまでは、複数の機器を接続するためのスイッチやルーターといったネットワーク機器は、技術者が機器ごとに IP アドレスの指定や通信経路などを設定しなければなりません。この課題を解決するための手段として、ソフトウェアでネットワーク機器を設定・制御できるようにし、自動化を可能にする技術として SDN が登場したのです。

この仕組みは、米スタンフォード大学の研究者たちが、2008 年以降実証実験を重ね、2011 年に「Open Networking Foundation（ONF）」という標準化団体を設立、そこに米グーグルや NTT コミュニケーションズ、NEC など、多くの技術者が係わるようになり、ソフトで制御できるネットワーク機器の標準仕様「OpenFlow」が策定されました。

2011年にはNECが世界初のOpenFlowに準拠したスイッチを発売以降、米シスコシステムズや、米ヒューレット・パッカード、米ビッグ・スイッチ・ネットワークスなど多くのネットワーク機器ベンダーも製品を提供するようになり、その普及が有望視されています。

従来のネットワークの仕組みでは、データを届ける際に、複雑に張り巡らされたネットワークの中をどういう経路やルールで送るのか、それぞれのネットワーク機器が個別に判断してバケツリレー式にデータの受け渡しを行っていました。そのため、データの混雑状況や、一部ネットワークの増強・メンテナンス状況に応じて最適なルートに変更することが困難でした。また、ネットワークの構築や変更では、大量の機器の設定が必要となり、大変な手間がかかっていました。

そこで、従来、個々のネットワーク機器が1台ずつで行ってきたネットワーク制御とデータ転送処理を分離し、制御機能を持つ機器またはサーバー（OpenFlowコントローラー）から、ソフトウェアの設定で転送処理のみを行う機器（OpenFlowスイッチ）を制御することで、通信を柔軟に効率よく、安全に行えるようにすることを実現しようとしています。また、OpenFlowコントローラーは、APIを解放しており、他のプログラムから制御することも可能となっており、人手を介することなくアプリケーション・プログラムや運用管理システムから、ネットワークを直接制御することも可能となっています。

これにより、ネットワーク機器の物理的な構成に依存することなく、ネットワーク全体の機能構成やQoSを、ソフトウェアで動的に制御、管理できるようになったのです。

サーバーの仮想化 そのメリットと課題

物理的資源の削減

サーバーの仮想化で、まず期待されるのは、物理資源の削減によるメリットです。CPUの使用率が低いサーバーをひとつのサーバーに集約することや、処理能力の低い古いCPUを搭載しているサーバーを、高性能な最新のCPUを搭載しているサーバーに集約し、物理サーバーの台数を削減することができます。

また、稼働率の低いサーバーを集約して削減することもできます。

これにより、設置スペースが削減でき、土地や建物に関わるコストを削減できます。例えば、データセンターの使用料が、月額 20 万円／ラックとした場合、10 ラックを 2 ラックに集約できれば、月額 200 万円を 40 万円、つまり、160 万円の削減ができます。また、消費電力の削減により、電気使用料、並びに CO2 の削減ができます。さらに、サーバーの稼働率を高めることや、平準化できますので、サーバーの購入代数を減らすことができます。

ソフトウェア管理への移行

仮想マシンは、プロセッサの数やメモリー容量などの構成に関わる情報が書き込まれた「設定ファイル」を、仮想化ソフトウェア（ハイパーバイザー）が読み取り生成されます。

もし、全く同じ構成の仮想マシンを生成したいのであれば、この「設定ファイル」を複製するだけです。また、プロセッサの数を 2 から 4 に増やしたければ、設定ファイルの記述を書き換えるだけで実現します。このように物理的な作業を行うことなく、サーバーを追加することや構成変更ができるのです。

これにより、サーバーの導入や構成変更に伴う工数や時間を大幅に削減できます。

また、ソフトウェア的な操作だけで導入や構成変更ができることから、運用管理や構成管理などを行う他のソフトウェアから人手を介することなく操作することができます。

ライブマイグレーション

さらに、仮想マシンで稼働している OS やアプリケーションを停止させることなく他の物理マシンに移行する「ライブマイグレーション」ができるようになります。

ライブマイグレーションを実行するには、以下の 2 つのデータの移動が必要になります。

- 動作中のプログラムが利用するメモリーの内容
- 仮想マシンが利用する仮想ディスク

メモリーの内容については、高速ネットワーク（ギガビット・ネットワーク）を介して、全てを移動先の物理サーバーに転送します。仮想ディスクについては、SAN や NAS といった共有ストレージを使い、移動前と移動後の仮想マシン

ンの両方からアクセスできるようにしておきます。

例えば、ひとつのストレージをふたつの物理サーバーで共有しているケースを想定してください。「設定ファイル」が、ここに保管されています。また、仮想化ソフトウェアは、お互いの死活を監視しています。

このような状態で、一方の物理サーバーが停止したことをもう一方の仮想化ソフトウェアが検知すると、直ちに、停止した物理サーバーで使われていた設定ファイルを自らの物理サーバーで起動させて、見かけ上の停止を回避することができます。このような仕組みを「ハイ・アベイラビリティ（HA：High Availability）機能」と呼んでいます。

また、保守や点検作業のためにあらかじめ停止することが分かっている場合は、ライブマイグレーションを使って、あらかじめ別の物理マシンに、停止する物理マシン上の仮想マシンを移動しておき、作業が終わったら元に戻すことで、見かけ上の稼働停止を回避することができます。

また、負荷の高くなった物理マシンから負荷に余裕のある物理マシンに仮想マシンを稼働させたまま移動させることで、負荷の平準化を図ることができます。

サーバーの仮想化の「ソフトウェア管理への移行」の特長を活かすことで、「障害に伴うシステム停止の回避」、「保守作業に伴うサービス停止の回避」、「負荷の平準化」、といった物理マシンではできなかったことも可能になるのです。

BCP 対策・仮想マシン・レプリケーション

また、仮想マシンの設定ファイルや仮想ストレージ、ログ・データを、他の場所に設置してある物理マシンに定期的に複製（レプリケーション）しておけば、災害時など、通常使っている物理マシンが使用できなくなったときにも、短時間のうちに同じ構成のシステムを立ち上げることができ、業務を継承させることができるようになります。

これもまた、「ソフトウェア管理への移行」によって可能になります。

サーバーの仮想化／課題

ただ、このようなメリットを享受できるサーバーの仮想化もいくつかの点で注意する必要があります。

ひとつは、「サーバー・スプロール」への対応です。

仮想マシンは、ソフトウェアによる設定だけで簡単に生成することができます。また、設定ファイルを複製するだけで簡単に仮想マシンを増やすことができます。そのため、未使用の仮想マシンが乱立することが考えられます。このような現象を「サーバー・スプロール」と呼びます。その結果、管理が複雑化しシステム資源を圧迫する可能性があります。運用ルールや管理方法を適切に設定することで、安易な仮想マシンを生成させないようにすることや、一定期間使用されない仮想マシンは運用管理者や運用管理システムにより、自動的に消去するなどの対策が必要となります。

「ストレージ設計」にも配慮が必要です。

ライブマイグレーション、ストレージ共有、ランダムアクセスの増大により I/O ボトルネックが発生しやすくなるなど、物理マシンをそのまま使っている場合とは比べものにならない大量の I/O が発生することや、異なる特性のディスク I/O が発生し、スループットの低下や安定しないなどの事態になりかねません。フラッシュ・ストレージなど I/O の高速化やボトルネックの生じにくい設計にすることが必要です。

最後は、「ポリシー管理」の問題にも対処する必要があります。

サーバーとネットワークが物理的に独立している場合は、ポリシーも個別に管理できたのですが、それが仮想化によって集約されると、その運用管理が複雑化します。追加や変更などが頻繁に起きるような場合は、ますます事態を難しくします。

ポリシーが似たもの同士をグループにまとめ、その単位で標準化してから集約することや、クラウド OS、運用や構成管理の自動化ツールをうまく使うことで対処する必要があります。

物理システムを前提としたシステムとは異なることをあらかじめ考慮し、システム設計を行うことが必要です。