



「サーバ仮想化」に求められる 「I/O 仮想化」とは？

Ariel Cohen 著

「サーバ仮想化」に求められる「I/O 仮想化」とは？



目次

はじめに	3
仮想マシン テクノロジ	5
仮想マシンの利点	7
サーバの統合と利用率の向上	8
システムの移行	9
I/O 仮想化	10
VM (仮想マシン) への I/O 仮想化の実装	14
仮想マシンにおける Xsigo I/O 仮想化の利点	16
仮想マシンにおける I/O サービス レベルの保証	17
問題点	17
解決策	17
ビジネス上の利点	17
サーバのインストール時間の短縮	17
問題点	17
解決策	17
ビジネス上の利点	18
サーバ I/O の拡張性の向上	18
問題点	18
解決策	18
ビジネス上の利点	18
ゲスト OS の迅速かつスムーズな移行	18
問題点	18

「サーバ仮想化」に求められる「I/O 仮想化」とは？



解決策.....	19
ビジネス上の利点.....	19
サーバ、ネットワーク、ストレージの統合	19
問題点.....	19
解決策.....	19
ビジネス上の利点.....	20
まとめ	20

「サーバ仮想化」に求められる「I/O 仮想化」とは？



はじめに

仮想マシンの概念が初めて実装されたのは、今から 40 年前にさかのぼります。当時、仮想マシンはメインフレーム コンピュータの一機能として生まれ、それ以来、ユーザーによるコンピュータの共有を可能にするために、メインフレームで幅広く使用されてきました。仮想マシンがつくる環境下では、アプリケーションとオペレーティング システムがまるでそれぞれ専用の物理マシン上で実行されているかのように見えます。よって、複数の仮想マシンを単一の物理マシン上でそれぞれ独立したエンティティとして実行できます。さらに、仮想マシンは相互に保護、分離され、同一物理マシンのリソースを消費する以外は互いに干渉することがないため、セキュリティ面での問題を気にすることなく共有という形態を確立することが可能なのです。また、仮想マシンに割り当てる物理マシンのリソースを制御できるため、各仮想マシンに対して必要なメモリ、CPU、およびその他リソースを確保することができます。

以前は、マイクロプロセッサベースの汎用サーバは処理能力とメモリ容量が限られていたことから、仮想マシンの実行に十分なプラットフォームとは考えられておらず、仮想マシンの使用は主にメインフレームに限定されてきました。

また、汎用サーバはメインフレームとは異なり、汎用サーバは仮想マシンの実行を想定して設計されていなかったため、仮想マシンに必要なハードウェア機能やメインフレームで提供されていたハードウェア機能を備えていませんでした。その結果、汎用サーバは一般にメインフレームとはまったく異なる方法で使用されることになりました。通常、汎用サーバが特定のアプリケーション（Web、データベース、数値演算処理など）に充てられるのに対し、メインフレームは多数のアプリケーションで共有されます。また、一般に汎用サーバが組織の特定部分専用（部門サーバ）に使用されるのに対し、メインフレームは組織全体で共有されます。さらに、メインフレームは高価なため、そのコストを償却するため最大限に活用することが求められますが、汎用サーバは比較的安価であるため、必ずしもそのような利用は必要ないと考えられてきました。

近年、仮想マシンのサポートが汎用サーバにも実装され、汎用サーバでの仮想マシンの使用は徐々に一般化しています。当初、その原動力となっていたのは、最新マシンでの旧OSの実行やソフトウェアのテストといった特殊なニーズでしたが、汎用サーバで仮想マシンを実行す

「サーバ仮想化」に求められる「I/O 仮想化」とは？



ることが、サーバをより有効に活用する上でも重要なツールになるという認識は高まりつつあります。

サーバを特定のアプリケーション専用を使用するこれまでのやり方が、サーバの低い利用率と極めて非効率なシステム構築の原因となっています。そこで、仮想マシン技術を使用すればこれらのサーバを統合できるため、効率が向上します。また、汎用サーバの処理能力が大幅に高まったことも、仮想マシンの使用を可能にしました。最近、マルチコア CPU をサポートした汎用サーバが流通し始めていますが、一部のアプリケーションはシングル スレッドであることから、マルチコア CPU で実行してもメリットはありません。しかし、仮想マシンは、異なる仮想マシンで実行される複数のシングル スレッド アプリケーションを別々のコアで並行して実行することができるため、マルチコア CPU を活用する有効な手段になります。

このほか、最近の汎用サーバが遂げた進歩として、64 ビット アーキテクチャへの移行により、非常に大容量のメモリに対処できるようになったことが挙げられます。この進歩とメモリの低コスト化により、単一の物理マシンで複数の仮想マシンを実行することが魅力的な手段となっています。

最近、汎用サーバのマイクロプロセッサ メーカーは、ハードウェアで仮想マシンをより良くサポートすることの必要性に着目しています。

Intel は Intel 仮想化テクノロジー (Vanderpool、VT-x、および VT-i としても知られる) を、また AMD は Pacifica テクノロジーを採用しました。これらのテクノロジーにより、仮想マシンの実行に伴う CPU オーバーヘッドの多くを排除できる新たな実行モードと命令を提供し、より効率的に仮想マシンを実装できるようになりました。

物理マシンを仮想化するには、CPU、メモリ、および I/O の 3 種類のリソースの仮想化が必要となり、CPU とメモリの仮想化には、既存のテクノロジーで効果的に対応できます。しかし、ネットワークとストレージを含む I/O サービスについてはそうはいきません。現在のソリューションでは、I/O 仮想化がソフトウェアで実行され、処理能力の制限や高いオーバーヘッドを伴います。

Xsigo の I/O 仮想化コントローラは、物理マシンの仮想化に必要な個々の I/O をコントロールします。コントローラの構成は、イーサネット、ファイバー チャネル、および SSL アクセラレーション用のアダプタが搭載されたシャーシで、サーバとの接続は非常に高速な (10Gbit/sec) リンクとなります。また、各物理 I/O アダプタは、QoS 保証パラメー

「サーバ仮想化」に求められる「I/O 仮想化」とは？



タ（帯域幅や優先度など）を備えた多数の仮想 I/O アダプタに分割できます。実装はハードウェアベースで行われるため、ワイヤスピードが実現され、サーバにオーバーヘッドがかかることはありません。

Xsigo ソリューションでは、効率的なハードウェアベースの仮想化に加え、豊富な I/O 仮想化機能をサポートします。たとえば、仮想マシンを物理サーバ間で移行する際に、仮想マシンに関連する I/O リソースと一緒に簡単に移動できます。物理アダプタの取付けや取外し、再配線が必要ないため、I/O のプロビジョニングも非常に簡単です。また、I/O リソースのプロビジョニングをポリシーやテンプレートによって自動的に行えるため、サーバ I/O の管理は大幅に簡略化されます。

本書の以降の部分では、仮想マシンテクノロジーとその利点、Xsigo I/O 仮想化ソリューション、および仮想マシンにおける Xsigo ソリューションの利点について詳細に考察します。

仮想マシン テクノロジー

仮想マシン テクノロジーにより、単一の物理マシン上で複数のオペレーティング システムを同時に実行できるようになります。複数のインスタンスを同じオペレーティング システム上で実行することも、異なるオペレーティング システム上で実行することも可能で、これらのインスタンスは相互に独立しています。さらに、CPUやメモリなどの物理マシンのリソース配分を制御した形でそれぞれのインスタンスに割り当てることができます。

図 1 は、仮想マシン環境を構成する各要素を示しています。物理ハードウェアは、特権モードで実行されるソフトウェア層で制御されます。このソフトウェア層はハイパーバイザまたは仮想マシンモニターと呼ばれ、物理リソースの管理を担い、それぞれのゲストOSが独立、隔離された形で実行されるように管理します。さらにこのソフトウェア層は I/O 仮想化も提供します。

図 1 の一般的な仮想モデルに準じた仮想マシン用のソフトウェアとして、VMware ESX Server、Microsoft Virtual Server、Xen などがあります。VMware と Xen のアプローチは多少異なっており、VMware が純粋な仮想化アプローチを取るのに対し、Xen はパラバーチャライゼーション（擬似仮想化）アプローチをとっています。純粋な仮想化では、ゲストOSは仮想マシン環境で実行されていることをまったく認識しません。一方、パラバーチャライゼーションでは、仮想マシンとして実

「サーバ仮想化」に求められる「I/O 仮想化」とは？



行するときに使用される特殊なコードをゲストOSに付加できるため、ゲストOSの一部は仮想マシン上で実行されていることを認識します。

パラバーチャライゼーションを使用することにより、演算処理およびI/Oのパフォーマンスを高めることができます。特に、効率的な仮想マシンの実装の障害になるような命令が x86 ISA（命令セットアーキテクチャ）に含まれる場合に有効です。ただし、この問題は、Intel 仮想化テクノロジおよびAMD Pacifica で解決されるため、ISA の制限に対処するためにパラバーチャライゼーションを実行する必要はなくなりますが、I/O 仮想化に関しては今後もパラバーチャライゼーションが有効な場合があります。

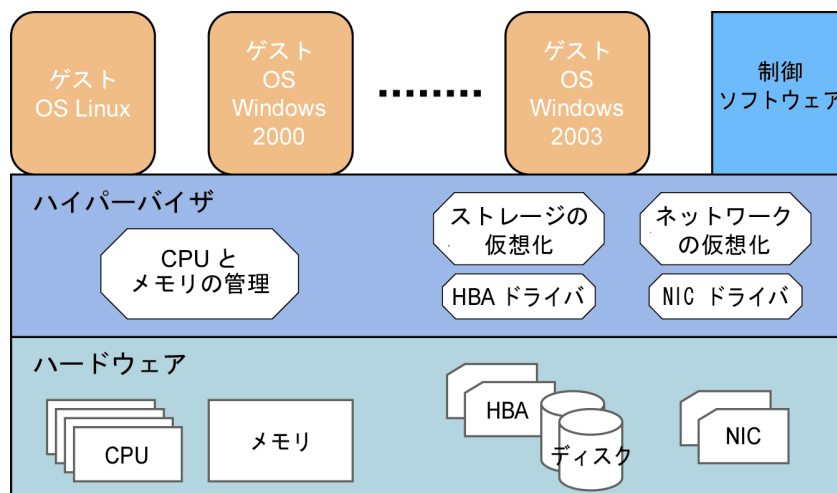


図 1: 仮想マシン

Xen と VMware では、I/O 仮想化に対するアプローチも多少異なります。VMware は汎用 I/O デバイスをゲストOSに公開しますが、Xen は、ゲストOS内に Xen 対応のフロントエンド ドライバと、ドメイン0と呼ばれる特殊なドメインで実行される特権付きゲストOS 内に対応するバックエンド ドライバを持っています。通常のゲストOSは非特権ドメインで実行され、特権付きのドメイン0は管理用およびバックエンド ドライバで使用されます。フロントエンド ドライバは、共有メモリによって実装されたデバイス チャネルを使用してバックエンド ドライバと通信します。

仮想環境内でのゲストOSが実際の物理デバイスを直接制御しないという意味では、現在の仮想マシンソフトウェアで構成される I/O デバイスは仮想デバイスです。ゲストOSが実際の物理デバイスを直接制御しない代わりに、この仮想デバイスが、ハイパーバイザまたは特殊な特

「サーバ仮想化」に求められる「I/O 仮想化」とは？



権付きゲストOSのソフトウェアによって物理デバイスに関連付けられ制御します。たとえば、ゲストOSから見た仮想ネットワークアダプタは、ソフトウェアに実装されたネットワークブリッジによって物理アダプタに関連付けられます。同様に、ゲストOSから見たストレージブロックデバイス上のブロックは、ハイパーバイザまたは特殊な特権付きゲストOSで実行されるソフトウェアによって物理デバイス上のブロックにマップされます。

サーバの仮想化は、物理マシンの CPU、メモリ、および I/O の 3 つの要素の仮想化から成ります。その中でも I/O仮想化は、現在 I/O仮想化をサポートしているハードウェアがなく、またソフトウェアでの仮想化が困難で非効率であることから、最も実現が困難な要素です。このことが、実稼動環境において、高パフォーマンスの I/O やサービス品質の保証を必要とするアプリケーションでの仮想マシンの使用に歯止めをかける要因となっています。このI/O仮想化の問題は、ネットワーク機器やストレージ、さらには同じ物理マシン上のゲストOS間の通信上でも起こりえます。

Xsigoのソリューションは、ハードウェアに実装された仮想 I/O をサポートすることにより、この問題を解決します。このソリューションでは、ソフトウェアのオーバーヘッドが排除されるほか、仮想 I/O デバイスをより厳密に制御することができます。詳細については、後述します。

仮想マシンの利点

ここ数年で、汎用サーバでの仮想マシンの活用は進展しました。当初、仮想マシン技術は主に開発、テスト、および旧OSの継続利用のために使用されていました。

ソフトウェア開発者は、複数のオペレーティング システムでソフトウェアを開発、テストするために 仮想マシンを使用してきました。単一の物理マシンで複数の仮想マシンを実行することにより、マルチプラットフォームでの開発とテストをよりコストパフォーマンスの高い環境で行えます。このほか、当初は、旧OSの継続利用の目的で、最新のハードウェア上の仮想マシンで旧式のOSを実行するといった利用方法もありました。

現在、仮想マシンは、データセンタ向けのアプリケーションとして使用されるようになってきています。これは、仮想化の利点が認識されつつあり、また近年のパフォーマンスの向上により、データセンタ内

「サーバ仮想化」に求められる「I/O 仮想化」とは？



で仮想マシンを使用することが可能になったからです。最新バージョンの仮想マシンソフトウェアは、パフォーマンスを向上させるよう設計されています。また、Intel および AMD による CPU ハードウェアでの仮想化のサポートと、Xsigo による I/O 仮想化により、わずかなオーバーヘッドで、またはオーバーヘッドをまったく伴わずに、仮想マシン上でアプリケーションを実行できるようになりました。このような技術の進歩により、仮想マシン上で実行可能なアプリケーションの幅は広がっています。

仮想マシンの普及により、データセンタには、サーバの統合とシステムの移行時に2つの大きな効果があらわれます。これらの効果について次に詳しく説明します。

サーバの統合と利用率の向上

IT 管理者は、導入、管理の必要のあるサーバ数の継続的な増加に直面しています。その一方で、平均的なサーバの利用率は非常に低く、10 ~ 20% 程度であることが多くの文書で裏付けられています。これは、サーバを特定のアプリケーションおよび IT ユーザー専用を使用する習慣が原因となっています。複数のアプリケーションで単一のサーバを共有し、複数の IT ユーザーが単一のサーバを共有できれば、必要なサーバ数は大幅に軽減され、結果的にサーバ、電力、空調、設置スペース、および管理コストを削減できます。または、システムの移行によりサーバを解放し、ほかのタスクを実行することもできます。仮想マシンテクノロジーは、このようなサーバ統合を次のように実現します。

- **複数のアプリケーションサーバを複数のアプリケーション仮想マシンに置き換えることができます。**仮想マシンは、それぞれ独立した環境でアプリケーションを実行できます。これにより、アプリケーションを専用サーバ上で実行する際のほとんどの利点は、専用仮想マシンで実行しても得られます。それぞれのアプリケーションは、異なる OSカーネル上で、異なるバージョンのライブラリを実行できます。よって、特定のアプリケーションの特異な動作がほかのアプリケーションに影響を与えることはありません。また、各アプリケーションに対して、保証された特定量のCPUおよびメモリリソースを割り当てることができます。さらに Xsigo ソリューションと組み合わせれば、割り当てる I/O リソースのサービス品質も保証できます。
- **仮想マシンでは、複数の IT ユーザーが同一サーバを共有するために必要なセキュリティとパフォーマンスを保証できます。**これ

「サーバ仮想化」に求められる「I/O 仮想化」とは？



まで専用サーバでアプリケーションを実行してきた IT ユーザーは、複数のユーザーで共有されるサーバへの移行に消極的な場合があります。このようなユーザーの不安要因として、セキュリティとパフォーマンスが挙げられます。仮想マシンでは、どちらの不安も解消されます。仮想マシンはそれぞれが独立しているため、特定の仮想マシンのユーザーが別の仮想マシンのユーザーのリソースにアクセスすることはできません。

システムの移行

仮想マシン テクノロジーにより、仮想マシン全体の状態をカプセル化し、別の物理マシンに移行することが可能になります。すなわち、実行中のオペレーティング システムを、そのアプリケーションを含めて物理マシン間で移行できます。

仮想マシンの実行中に仮想マシンの状態をあらかじめコピーすることにより、ダウンタイムを非常に短く抑えることができます。たとえば、Xen を使用した実験では、ネットワーク集中型アプリケーションの停止時間は 100ms 未満でした。

仮想マシンの移行には、次のように多くの利点があります。

- **サーバ ハードウェアのアップグレードおよびメンテナンス中のサービス停止を排除。**サーバで実行中のアプリケーションとオペレーティング システムを別のサーバに移行できるため、元のサーバでサービスを停止できます。
- **動的なロード バランシング。**ビジー状態のサーバから処理能力に余裕のあるサーバへと仮想マシンを移行することにより、サーバの負荷を均等化することができます。この移行は、負荷パターンの変化に応じて動的に行なえます。これにより、パフォーマンスとサーバ リソースの利用率を高めることができます。
- **クライアント近接性の向上。**仮想マシンを、よりクライアントに近いマシンで実行するよう移動できます。これはロード バランシングに似ていますが、この場合はクライアント近接性が基準として使用されます。たとえば、あるアプリケーションのクライアントの多くがネットワークの特定部分に集中していることが判明したときに、そのアプリケーションを実行する仮想マシンをよりクライアントに近いサーバに移行することにより、アプリケーションとクライアント間で使用できるネットワーク帯域幅を高めることが有益である場合があります。

「サーバ仮想化」に求められる「I/O 仮想化」とは？



I/O 仮想化

汎用サーバに搭載されている I/O アダプタには、I/O 仮想化機能が組み込まれていません。このようなアダプタは、単一の OS カーネルの単一ドライバによって制御されるよう設計されています。したがって、アダプタでは、その全帯域幅を含むすべてのリソースが単一のコンシューマ、すなわちマシン上で実行されている OS に提供されます。

ハードウェアベースでの I/O 仮想化がサポートされていないために、仮想マシンでの仮想 I/O デバイスのサポートが困難になっています。仮想マシンソフトウェアでは、ソフトウェアベースの I/O 仮想化が唯一の手段となっており、実際の I/O ハードウェア用のデバイス ドライバをハイパーバイザまたは特権付きゲスト OS に用意する手法が一般的なソリューションとなっています。この I/O 仮想化により用意されたデバイスドライバは、物理サーバ上の OS 内のドライバと同様に、実際の I/O ハードウェアの唯一の所有者となります。ハイパーバイザは、I/O ハードウェアの制御に加え、ゲスト OS 用の仮想 NIC および仮想ストレージデバイスの制御をソフトウェアでサポートします。

たとえば、ゲスト OS 内の仮想 NIC はアプリケーションからネットワークインタフェースとして認識されますが、そのドライバによってハードウェアが操作されることはありません。代わりに、パケットは、ゲスト OS のフロントエンド ドライバとハイパーバイザまたは特権付きゲスト OS のバックエンド ドライバ間のソフトウェアデバイスチャネルによって仮想 NIC ドライバと実際の NIC ドライバ間で転送されます。そして、ソフトウェアで実装されたブリッジを使用することにより、複数の仮想 NIC で単一の物理 NIC を共有できます。

ソフトウェアによる I/O 仮想化アプローチでは、I/O 処理に追加のオーバーヘッドが生じ、処理能力が制限されます。そのため、I/O 要件の厳しい実稼動環境に仮想マシンを導入すると、ソフトウェアによるアプローチが障害になってしまいます。より魅力的なアプローチは、帯域幅などのリソースの割当てを制御可能な、それぞれ独立した I/O アダプタのインスタンスを複数提供することにより、ソフトウェアでの I/O 仮想化の必要性を排除した、より高度な I/O アダプタを提供するものになります。この種の機能は、サーバで実行されるソフトウェアではなく、I/O を提供するハードウェアで実現されます。

Xsigo 製品により、I/O 仮想化という重荷から 仮想マシンソフトウェアを解放することができます。この製品は、I/O モジュール用のスロットを多数搭載したシャーシで構成されており、I/O モジュールとサー

「サーバ仮想化」に求められる「I/O 仮想化」とは？



バ間の接続は、リモート DMA 機能を備えたノンブロッキングの高速 InfiniBand ファブリックによって確立されます。また、仮想アダプタの種類ごとに異なる I/O モジュールが用意されています。VNIC I/O モジュールは仮想イーサネット NIC を、VHBA モジュールは仮想ファイバー チャンネル HBA を、また VSSL モジュールは仮想 SSL オフロード アダプタを提供します。

各モジュールには、Xsigo が開発した特殊な I/O 仮想化ロジックが組み込まれています。このロジックにより、I/O モジュールのリソースを多数の仮想デバイスに分割し、これらの仮想デバイスを別々のデバイスとしてサーバに公開することができます。たとえば、ギガビットイーサネット ポートが 4 基搭載された VNIC I/O モジュールでは、最大 256 の仮想 NIC を提供できます。また、I/O モジュール上にはサービス品質保証のためのメカニズムが実装され、管理者は必要に応じてそのパラメータを設定できます。たとえば、各 VNIC に対して帯域幅と優先度を設定したり、これらの設定を動的に変更することも可能です。

図 2 は、Xsigo の I/O 仮想化の基本概念を示しています。

サーバには、高速 (10Gb/s) InfiniBand リンクで Xsigo シャーシに接続するための PCI Express InfiniBand ホスト チャンネル アダプタ (HCA) が搭載されています。

複数のリンクを使用することにより、冗長性を持たせたり、帯域幅を増加させることも可能です。

イーサネットおよびファイバー チャンネル ネットワークは、Xsigo シャーシに接続された Xsigo I/O モジュール上のイーサネットおよびファイバー チャンネル ポートに接続されています。I/O モジュールは、サーバに VNIC および VHBA を提供するように構成されており、サーバ上で実行される Xsigo ドライバでは、オペレーティング システムに公開されるネットワーク インタフェースとストレージ アダプタを提供するためにこの VNIC および VHBA が使用されます。各サーバは、自由に設定変更可能な QoS パラメータを備えた高速リンクによってイーサネットおよびファイバーチャンネルネットワークに接続されています。これらのリンクは仮想的であり、動的に変更できますが、実際にはあらゆる用途において物理リンクと区別することができない同等のリンクとなります。

「サーバ仮想化」に求められる「I/O 仮想化」とは？

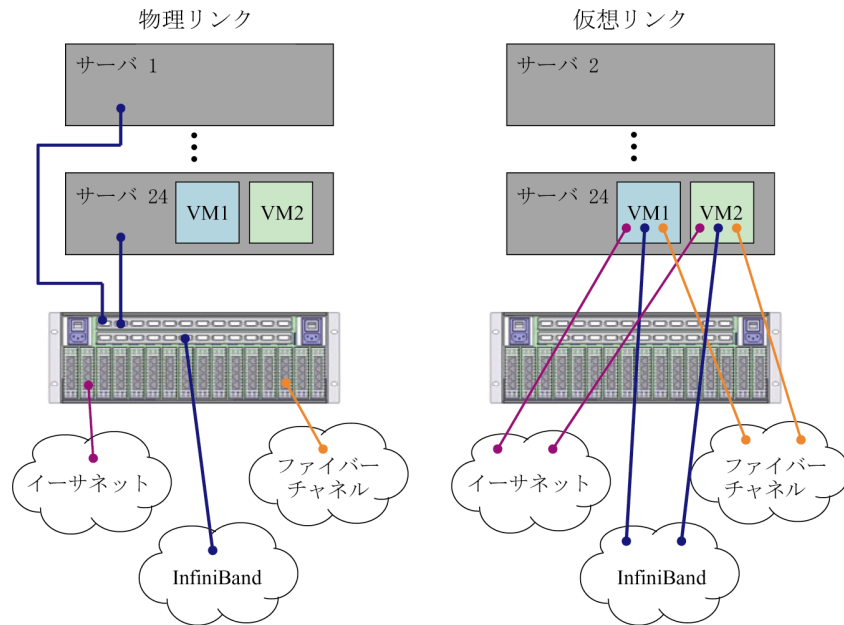


図 2: Xsigo の I/O 仮想化

図 3 に例を示します。この図には 3 つの VM (仮想マシン) が示されており、それぞれ独自の VNIC と VHBA を保持しています。また、各 VNIC および VHBA には帯域幅が設定されています。VNIC1 と VNIC2 は同じ VNIC モジュールの同じイーサネット ポートを共有していますが、VNICn は別の VNIC モジュール上にあります。VNIC と VHBA の帯域幅および優先度の設定は、仮想マシンの実行中に動的に調整できます。また、仮想リソースを動的に追加および削除することも可能です。たとえば、新しい VNIC を追加したり、既存の VNIC を削除できます。

「サーバ仮想化」に求められる「I/O 仮想化」とは？

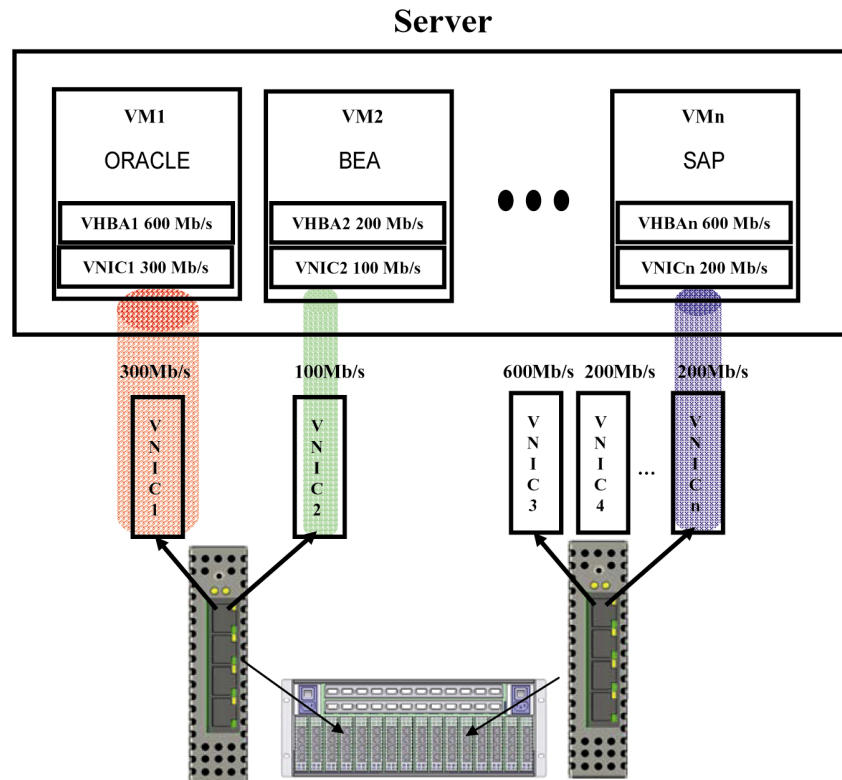


図 3: I/O 仮想化の例

ここで、I/O リソースの仮想化および帯域幅の保証という負担を、仮想マシンソフトウェアではなく Xsigo I/O モジュール内のハードウェアが負うことに注目してください。これにより、仮想リソースのパフォーマンスが向上し、サーバ CPU のオーバーヘッドは排除されます。また、物理リソースの仮想化を複数のサーバに渡って、また同じサーバ内の複数の仮想マシンに渡って行えるという利点もあります。たとえば、この例で同じ I/O モジュールとイーサネット ポートを共有している VNIC3 と VNIC4 を別々のサーバに割り当てることも可能です。

この種の I/O 仮想化の興味深い機能として、サーバ間で仮想 I/O リソースを移行できることが挙げられます。たとえば、この例の VM2 を別のサーバに移行する場合、VHBA2 および VNIC2 も一緒に移行できます。InfiniBand HCA では HCA 間での QP パスの移行がサポートされるため、データを失うことなく移行を実行できます。

Xsigo 製品では、各仮想リソースのサービス品質が保証されることに加え、同じ仮想リソース上の異なる種類のトラフィックに対してサービス品質を設定できます。

「サーバ仮想化」に求められる「I/O 仮想化」とは？



図 4 は、VMn 上で SAP、Exchange、およびバックアップの 3 つのアプリケーションが実行されている例です。これらすべてのアプリケーションは VNICn を使用していますが、VNICn で使用できる帯域幅 (200Mb/s) は 3 つのアプリケーションに分割されるため、SAP では 100Mb/s、Exchange では 70Mb/s、バックアップでは 30Mb/s というように、各アプリケーションに対して特定量の帯域幅が保証されます。これは、TCP ポートなどのパケット フィールドに基づいてトラフィックを分類できる Xsigo I/O モジュール内の分類エンジンによって実現されます。

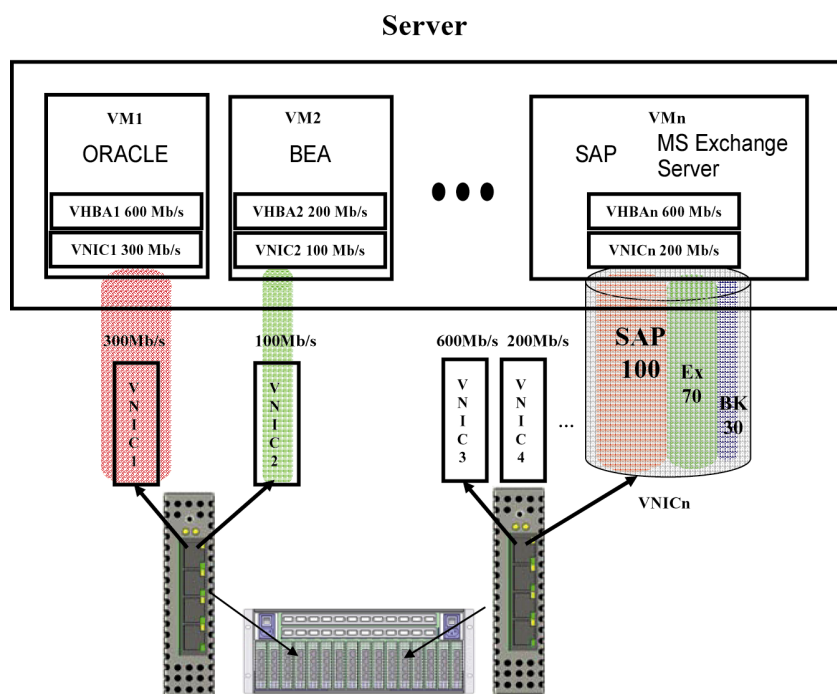


図 4: アプリケーションベースのサービス品質

VM (仮想マシン) への I/O 仮想化の実装

Xsigo システムを使用した仮想 I/O の実装アプローチは 2 通りあります。最初のアプローチは、一般に仮想マシンソフトウェアで物理 NIC およびストレージ デバイスを仮想化するために使用されているアプローチに似ています。違いは、ソフトウェアによって仮想化される対象が物理 NIC ではなく Xsigo VNIC であり、仮想化される対象が物理 HBA に接続されたストレージデバイスではなく VHBA に接続されたストレージ デバイスであるということだけです。このアプローチを図 5 に示します。

「サーバ仮想化」に求められる「I/O 仮想化」とは？

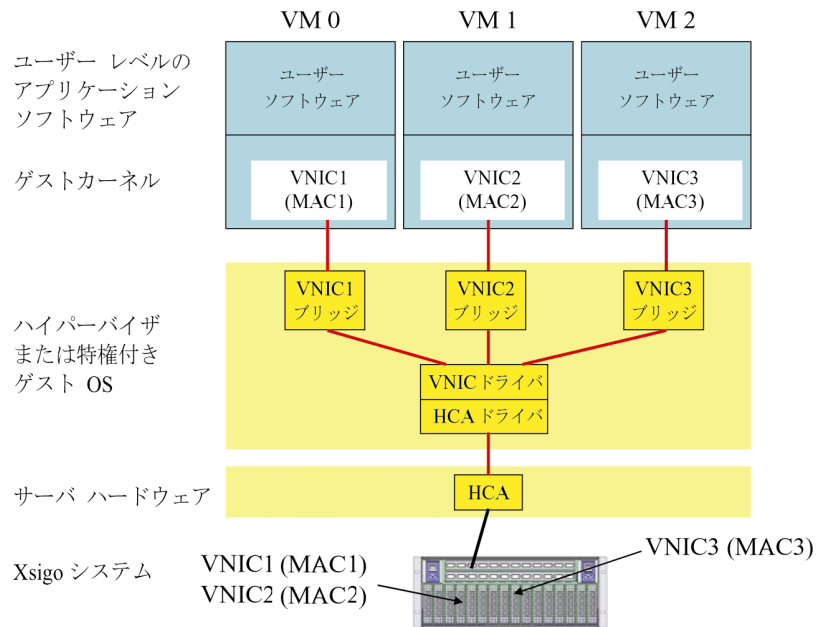


図 5: ソフトウェア ブリッジによる VNIC の実装

VNIC および HCA ドライバは、物理 NIC ドライバと同様にハイパーバイザまたは特権付き OS ゲストに存在しています。VNIC は、ソフトウェア ブリッジによってゲスト OS の VNIC にブリッジされます。この点も、物理 NIC で行われる方法と同じです。このアプローチの主な欠点は、ゲスト OS のフロントエンド VNIC ドライバとハイパーバイザまたは特権付きゲスト OS のバックエンド VNIC ドライバ間の通信によってオーバーヘッドが生じることです。利点は、シンプルであり、十分にテストされたドライバ コードが使用されることです。

上記のアプローチでは、一般的な物理 NIC と HBA を仮想化する場合に伴うオーバーヘッドの一部が生じますが、Xsigo システムを使用した場合は、大きな利点もあります。たとえば、1 つの物理イーサネット ポートを、帯域幅などの QoS パラメータがそれぞれ異なる複数の VNIC に分割することができます。この QoS はハードウェアに実装されます。

また、これらの VNIC を同じサーバ上の異なる仮想マシンに割り当てることも、異なるサーバ上の仮想マシンに割り当てることもできます。

2 つ目のアプローチでは、InfiniBand HCA の特殊な機能を利用することにより、I/O 仮想化に伴うオーバーヘッドを排除します。HCA ドライバは、ゲスト OS 内で実行できます。これは、HCA が複数のコンシューマからの独立したアクセスをサポートしているためです。その

「サーバ仮想化」に求められる「I/O 仮想化」とは？



ため、ほとんどの HCA 操作をゲストOS 内のドライバで安全に実行でき、特に、すべてのデータ パス操作でハイパーバイザおよび特権付きゲストOS をバイパスして実行することができます。

これを図 6 に示します。

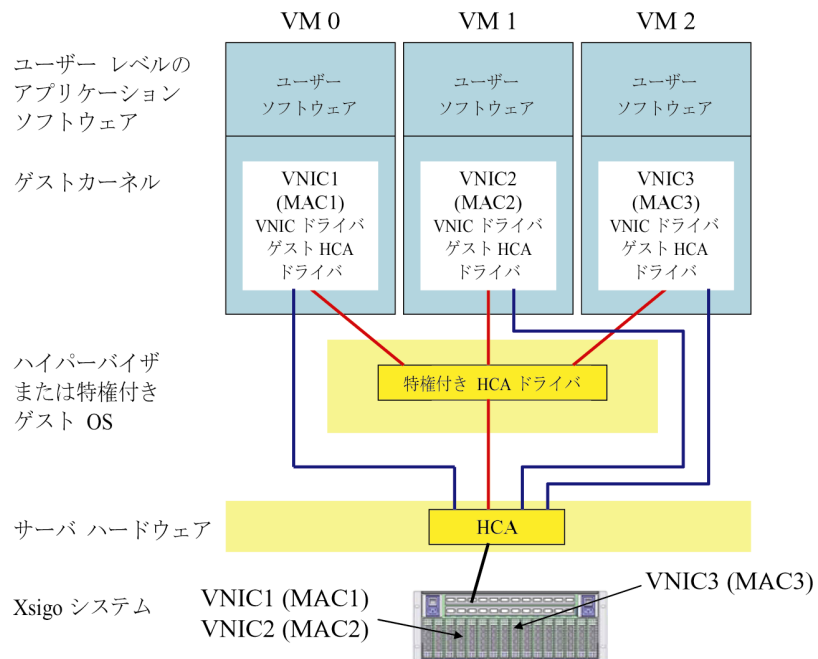


図 6: ゲスト OS 内の VNIC ドライバ

このアプローチでは、VNIC ドライバと、HCA ドライバのほとんどの機能がゲストOS 内で実行されます。ゲストHCA ドライバは、ハードウェアに直接アクセスすることにより、ゲストOS に割り当てられた HCAリソースを制御できます。ソフトウェア ネットワーク ブリッジとバックエンド VNIC ドライバをバイパスするため、最初のアプローチで生じるオーバーヘッドがなくなります。各ゲストOS は、ソフトウェア オーバーヘッドを生じさせることなく、ハードウェア内で完全に仮想化された仮想リソースを保持することになります。

仮想マシンにおける Xsigo I/O 仮想化の利点

この節では、仮想マシンと Xsigo I/O 仮想化ソリューションを組み合わせ使用した場合に得られるビジネス上の利点についてまとめます。

「サーバ仮想化」に求められる「I/O 仮想化」とは？



仮想マシンにおける I/O サービス レベルの保証

問題点

帯域幅や優先度などのサービス レベルを保証しながら、仮想マシンで使用される物理 I/O リソースを仮想 I/O リソースに分割する効率的な方法を提供することが求められています。

解決策

Xsigo システムは、NIC、HBA、および SSL アクセラレータに対してハードウェアベースの I/O 仮想化を実現します。サービス品質パラメータはユーザーが設定できます。実際には、仮想マシンは、物理リソース（100Mb/s NIC、1Gb/s HBA など）であるかのように機能する仮想 I/O デバイスを保持することになります。

ビジネス上の利点

仮想マシンで実行されるアプリケーションの動作を、よりビジネスニーズに近づけることができます。アプリケーションに対して必要な I/O リソースを確保できます。これにより、アプリケーションのパフォーマンスとレスポンスタイムが向上します。

サーバのインストール時間の短縮

問題点

サーバへの物理ネットワークおよびストレージ インターフェースのインストールは、時間のかかる作業です。

解決策

Xsigo システムを使用すれば、サーバへの I/O リソースのプロビジョニングを瞬時に行えます。サーバの設置前に、あらかじめ I/O リソースをプロビジョニングすることも可能です。サーバを設置した時点で、その I/O はすでに使用可能な状態にあります。これは、サーバ内のハードウェア I/O デバイスで仮想マシンテクノロジーを使用するだけでは不可能だったことです。Xsigo ソリューションは、サーバの I/O リ

「サーバ仮想化」に求められる「I/O 仮想化」とは？



ソースをサーバ間で共有することにより、本当の意味での I/O リソースの仮想化を実現します。

ビジネス上の利点

インストール時間が短縮されるため、サーバの利用率が向上します。サーバのダウンタイムが最小限に抑えられます。サーバに割当てた新しい I/O の設定をテストする効率が高まります。

サーバ I/O の拡張性の向上

問題点

サーバに十分なネットワークおよびストレージ接続性を提供するには、高価な I/O スロットが多数必要になります。マルチコア CPU と多数の仮想マシンを使用する新たなサーバでは、特にこの傾向が強くなります。

解決策

Xsigo システムは、複数の仮想 I/O トラフィックを、大容量の 10Gb/s InfiniBand HCA 上で多重化します。Xsigo ソリューションでは、通常なら複数のアダプタと複数の I/O スロットが必要になるトラフィックにも、1 つのアダプタと 1 つの PCI Express スロットで対応できます。

ビジネス上の利点

サーバの使用の柔軟性が向上し、サーバ コストが削減されます。

ゲスト OS の迅速かつスムーズな移行

問題点

実行中のゲスト OS をそのアプリケーションと一緒にサーバ間で移動する機能が求められています。これにより、サーバのメンテナンス時

「サーバ仮想化」に求められる「I/O 仮想化」とは？



にアプリケーションを停止する必要がなくなり、負荷をより均等に分散させることができます。

解決策

仮想マシンソフトウェアでは、VM（仮想マシン）を移行することが可能です。VM は、元のサーバ上で実行中にコピーされます。コピーが完了すると、VM は元のサーバで停止され、新しいサーバで開始されます。この移行は 100ms 未満で行えますが、コピー操作にはこれより大幅に長い時間がかかります。Xsigo ソリューションを使用すれば、10Gb/s 速度の RDMA を使用することにより、コピー時間を短縮できます。さらに重要なことは、データを失うことなく仮想 I/O デバイスを古いサーバから新しいサーバに移行できることです。

ビジネス上の利点

ダウンタイム 0 でサーバのメンテナンスを行えます。効果的に負荷が分散されるため、サーバの利用率が向上します。

サーバ、ネットワーク、ストレージの統合

問題点

データ センタは低いリソース利用率に悩まされています。これには、サーバ、ネットワーク スイッチ ポート、およびストレージ スイッチ ポートの利用率の低さが含まれます。その結果、装置、設置スペース、電力、および管理に必要以上のコストがかかっています。仮想マシンテクノロジーは、サーバ利用率を高めることによってこの問題の軽減に役立ちますが、I/O 側にはさらに改善の余地があります。

解決策

Xsigo システムを使用すれば、ネットワークおよびストレージ リソースの利用率をさらに高めることができます。NIC、HBA、SSL 暗号化デバイスなどのリソースが共有されるため、ネットワークおよびストレージ スイッチ ポートが統合されます。たとえば、単一のファイバーチャネル スイッチポート上で複数のサーバが仮想 HBA を持つこと

「サーバ仮想化」に求められる「I/O 仮想化」とは？



ができ、物理 HBA のように専用のファイバー チャネル スイッチポートを必要としません。

ビジネス上の利点

データ センタのコストと複雑さが軽減されます。

まとめ

サーバの仮想化は、よりコストパフォーマンスに優れたデータ センタの構築に向けた強力かつ有効なテクノロジーです。マルチコア CPU の普及に伴い、仮想マシンテクノロジーはこれまで以上に魅力的なものとなっています。しかし、現在のソリューションで提供される CPU とメモリの仮想化だけでは十分とは言えません。I/O への取組みも必要です。I/O 仮想化は、パフォーマンスを犠牲にすることなく、また I/O のサービス品質を保証しながらサーバを仮想化する完全なソリューションの作成に必要なテクノロジーです。Xsigo システムは、サーバの仮想化を I/O の面から解決するために必要となる豊富な I/O 仮想化機能を備えています。

完全なサーバ仮想化ソリューションの要素がすべて揃って初めて、次世代のデータセンタ アーキテクチャが実現されます。このアーキテクチャでは、リソースの利用率と柔軟性が大幅に高まります。アプリケーションとオペレーティング システムに対して、必要な CPU、メモリ、および I/O リソースを動的に割り当てることができます。また、サービス レベルを保証しながら、これらすべてのリソースを安全に共有することが可能です。これにより、専用リソースが柔軟性に乏しく十分に活用されていないという問題は解決されます。