仮想化の落とし穴と脱出法

高　橋　優　亮

要　約　サーバの仮想化の数多くのメリットは、利用者にも認知されてきており実績もあるが、いまだに実現性に疑いの目が向けられることもある。サーバ仮想化によって問題が引き起こされることがあるのは事実だが、多くの問題は解決可能である。ここでは、典型的、代表的な問題とその解決策を紹介し、起こり得る問題が解決可能であるということを示す。端的に言えば、サーバ仮想化に対応した専用ハードウェアと、専用のハードウェアの力を活かせる新しい世代の仮想化ソフトウェアを使用し、十分な知識と経験を持ったパートナーと付き合うということに尽きる。本稿を基に、読者の皆様が自信をもってサーバ仮想化に取り組んでくだされば幸いである。

1. は　じ　め　に

サーバ仮想化技術が注目されるようになって久しい。IT 系の雑誌やハイパーバイザーベンダーの Web サイトには、以下のような仮想化のメリットが紹介されている。

・サーバ統合でコスト削減が可能
・古い物理サーバ上の現役アプリケーションを延命できる
・CPU やストレージの利用効率を向上できる
・運用台数を減らすことで、管理コストを低減できる
・電力や熱や設置面積の減少で、コスト削減やグリーン化を推進できる
・冗長化やディザスタリカバリ対策が従来より簡単になる
・クライアントの仮想化で情報システム部門のデスクトップ管理が楽になる
・開発・テスト・デプロイ・運用といったサイクルの回転速度を上げられる
・運用柔軟性が高まり、保守性が向上する
・投資柔軟性が高まり、ROI 向上が見込める

あまりに「ウマイ話」が並び、眉唾に感じられるかもしれないが、適切に実装すれば、これらはどれも真実となりうる。すべてを実現できれば、それは「仮想化の楽園」と言えるだろう。

かつては、これらのすべてを実現するには技術的なハードルが高く、「仮想化の楽園」は言わば「危険な密林の奥にある秘密の桃源郷」のような存在で、幸運に恵まれた一握りのプロフェッショナルしかゴールすることはできなかった。しかし、ハードウェアやソフトウェアの技術の進歩と、設計構築側のノウハウが蓄積されてくるにつれ、方法論が確立し、今日では優秀なガイドが同行すれば比較的安全に楽園に至ることが可能になった。本稿では、そんな楽園に至る途中の密林にある代表的な落とし穴や危険と、その回避策について解説する。

2.　仮想化の本質と得失

2. 1 「仮想化」の定義

仮想化とは、「サービス提供者」と「サービス利用者」の間に「新しい中間者」を挿入する

ことと定義できる。「新しい中間者」は「サービス提供者」のサービスを利用し、何らかの付加価値を付けて「サービス利用者」に提供する。

こうした「新しい中間者」を挿入するモデルの典型的なものは、OSI や TCP/IP に代表される階層型ネットワークのアーキテクチャに見ることができる。この場合の「新しい中間者」とは「新しい階層/レイヤ」のことである。逆に階層型ネットワークアーキテクチャを、仮想化の塊であると捉えなおすこともできる。

2. 2　仮想化のメリットとデメリット

仮想化する、つまり新しいレイヤを既存レイヤ間に挿入するためには、上下のレイヤの間の関係を整理し、インターフェースとして規格化する必要がある。インターフェースが規格化されることで、リソースの統一的な取り扱いが可能になり、運用上さまざまな恩恵がもたらされる。

また、インターフェースが統一化され、新規レイヤが挟み込まれることによって、上下レイヤの結びつきが弱まることになる。これにより、上位レイヤに影響を与えることなく独立に下位レイヤを冗長化したり、逆に下位レイヤを多重化してリソースの利用効率を向上させるといった、多様な付加価値サービスを提供することができる。

リソースの統一的な取り扱いと、新規レイヤによる付加価値サービス。この二つが仮想化によってもたらされるほとんどすべてのメリットの源泉となる。

これらのメリットと引き換えにされるのが、パフォーマンスとトラブルの透明性である。新しいレイヤの追加は、すなわち追加の処理をもたらし、パフォーマンスは必ず低下する。また、上下レイヤの結びつきが弱まることは、下位レイヤで発生したトラブルが上位から見えにくくなるということでもある。また新しいレイヤが挿入されことによって、障害発生時の挙動が変化したり、障害発生時に観測すべきインターフェース境界が増加するので、多くの場合、障害の解析や原因の特定が困難になる。

仮想化のメリットをデータセンターのサーバ運用に活かそうとするのが「サーバ仮想化」である。ハードウェアと OS との間にハイパーバイザを挿入し、複数の OS に均質化したハードウェアを提供して、ハードウェアの利用効率を上げる（図 1）。一方、サーバ仮想化にも上記デメリットは発生する。それが、本稿で述べる「仮想化の落とし穴」である。

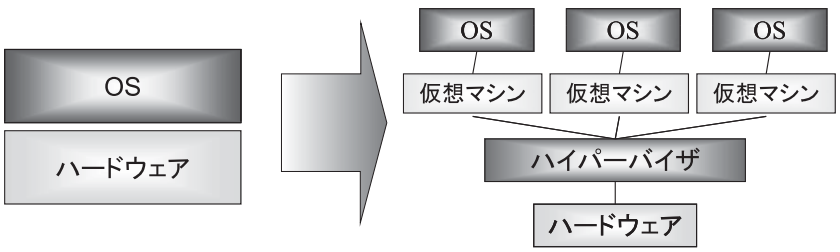

図 1　物理サーバ（左）とその仮想化（右）

3.　仮想化の落とし穴と脱出法

サーバ仮想化で引き起こされがちな問題を、原因別に分類した上で以下に列挙した。

1）　パフォーマンス低下に起因するもの

・Linux 仮想サーバの時計が合わない

・I/O 性能が驚くほど低くて統合率を上げられない

2)　ネットワークに絡むもの（実際に設計や構築をしないと分からない）

・ブレード LAN スイッチやラックの LAN スイッチのポート数が足りない

・サーバから出てくる I/O ケーブルの数が増えて、ラックがスパゲティ状態になる

・ネットワーク側でセキュリティや QoS をかけられない

3)　仮想マシンのファイルシステムに絡むもの

・ファイル単位のバックアップからのリストアが難しくなった

4)　ハイパーバイザの制限によるもの

・専用デバイスを使ったシステムが仮想化できない

5)　サーバ仮想化が導く変化を原因とするもの

・ストレージ設計の重要性と難易度が上がり、そちらでコストが上昇した

6)　大人の事情によるもの

・NT4.0 サーバの延命で障害が起きても、先に進むためのサポートがなくなっていた

本稿ではこれらのうち、パフォーマンスとネットワークに起因するトラブルと解決策について、続く 4 章と 5 章で解説する。

4.　パフォーマンス限界に起因するトラブル

4. 1　Linux 仮想マシンの時計が合わない

Linux 仮想マシンの時計が異常に進んでしまうという現象が発生する場合がある。これは、VMware ESX サーバ上で稼働する仮想マシンが Linux で、カーネルバージョンが 2.6.0 から 2.6.12 の場合に発生する。時刻異常は、ログファイルの解析を難しくするほか、トランザクション異常を引き起こす可能性もあり、決して見過ごすことのできない重大な問題である。この問題の原因を理解するためには、PC のハードウェアと現代の OS における時刻管理の知識が必要である。

現代の PC では、RTC（Real Time Clock）と呼ばれるハードウェアが現在時刻（TOD：Time of the Day）を保持する唯一のハードウェアである。このほかに、指定時間の経過を知らせるタイマーとして、PIT（Programmable Interval Timer）、ACPI（Advanced Configuration and Power Interface）、LAPIC（Local Advanced Programmable Interrupt Controller）、HPET（High Precision Event Timer）、そしてパフォーマンス測定用の経過時間計測用カウンタ TSC（Time Stamp Counter）など、沢山の時刻関連ハードウェアが搭載されている（図 2）。

PC アーキテクチャが登場した 1980 年代にはこれほど多くの時間関連ハードウェアは搭載されていなかったが、時代が進むにつれて多様化するアプリケーションの要求に応えるために、ハードウェアが追加され、旧ハードウェアも過去との互換性を維持するために残された結果、このようになっている。

現代のオペレーティングシステムでは起動時に一度だけ、RTC から現在時刻 TOD を問い合わせて、結果をメモリに書き込み、「システム時刻」というデータを作る。その後はタイマーを使う。たとえば 1/100 秒を計時単位とする OS の場合は、いずれかのタイマーに「100 分の 1 秒経過」を割込みという手法で CPU に通知させ、そのたびにメモリ上のデータであるシステム時刻を 100 分の 1 秒進める。このようにソフトウェア的に現在時刻を維持・管理するの

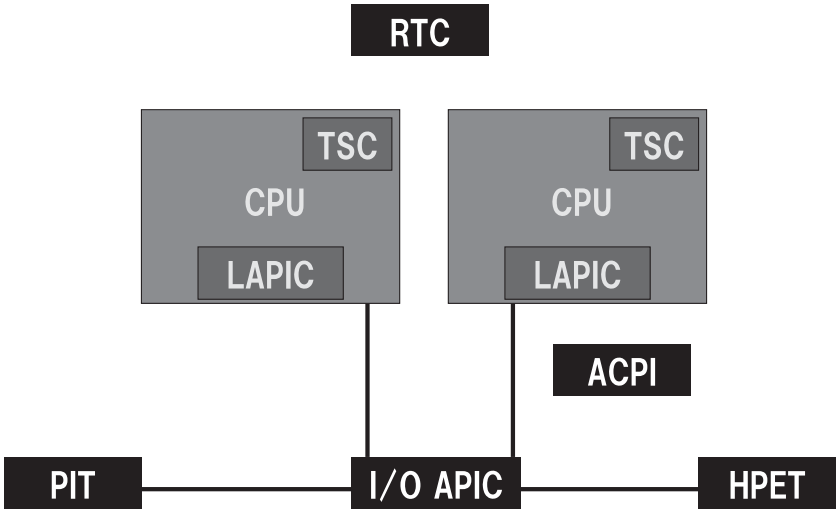

図2　x86システムのタイムソース

である。

アプリケーションからOSに対する時刻問い合わせがあった場合は、メモリ上のシステム時刻の値を回答する。現在時刻を保持するRTCがあるのに、OSが稼働中にRTCへの問い合わせをしないのは、RTCが1980年代に設計された古くて遅いハードウェアであり、問い合わせの応答には現代のCPUの速度と比べて大変長い時間がかかるためである。これに対しシステム時刻はメインメモリ上のデータであるため、高速に読み出すことができる。OS自身やサーバソフトウェアはログ記録などのために、頻繁に現在時刻を問い合わせるので、都度RTCを使うことは現実的でないのである。

Linuxカーネル2.4（古いバージョン）とWindowsは、いずれかのタイマーをひとつだけ利用してその割込みでシステム時刻を更新する。ところが、時刻更新よりも優先度の高い処理を実行している場合、このタイマー割込みは無視されてしまう。このため、システム負荷が高いとわずかに時計が遅れることがあった。

Linuxカーネル2.6.0ではこの問題を解決するために、タイマーとTSCなど複数を組み合わせて、それらを較べることによって取りこぼした可能性のある割込みの数を推測し、必要に応じて取りこぼした分時刻を進ませ、時計の遅れを取り戻せるようにした（Lost Tick Correctionアルゴリズム）。また、計測単位を1000分の1秒にして、より細かい単位の時間を正確に維持・管理するようにした。これらはLinuxカーネル2.6の改善点で、物理マシンの上では良好に動作した。だが、これが仮想マシンの上では仇となってしまう。

計時単位が1/1000秒ということは、少なくとも1秒間に1000回の割込み処理を行う必要がある。物理マシン上ではこの要件はさほど厳しいものではないが、仮想化によってパフォーマンスが低下した仮想マシンにとって、1秒間に1000回の割込みはほとんど達成不可能で、せいぜい1秒間に300回程度が限界であった。これは計時単位が1/100秒のカーネル2.4やWindowsには十分でも、カーネル2.6にとっては1秒に割込みを700回も取りこぼしたのと同じことになる。

Lost Tick Correctionは、これほど大幅な取りこぼしを想定していなかったため、失われた

割込み数の予測を大きく誤り、過剰に時計を進めてしまうのである。

システム負荷が高くなると、さらに割込み数が減ってしまい、一層 Lost Tick Correction が余計に時刻を進めてしまう。負荷が高くなればなるほど、パフォーマンス不足であればあるほど、時計が進むという、それまでの OS の常識とは逆の現象が発生したのである。またその原因を究明するためには Linux OS とプラットフォームとなる VMware に関する深い知識が必要であった。

これは 2005 年当時のトラブルであり、現在は解決されているが、「パフォーマンスが失われた」ことと「原因究明が難しくなる」という、仮想化環境で起こる象徴的な現象として採り上げた。

4.2 I/O 性能が驚くほど低くて統合率を上げられない

5 年ぶりのシステム入れ替えで Microsoft SQL Server を仮想化したお客様の事例である。従来システムはアプリケーションタイムアウト時間に対して応答時間は 80％で許容範囲に入っていた。仮想化にあたり、5 年の間でのハードウェアのパフォーマンスの進化を見込んでいたが、新システムのテストではタイムアウト時間に対して応答時間が 150％となってしまった。分析の結果、CPU は速くなっているものの、I/O が足を引っ張っていることが判明し、各所のチューニングで従来システム並みの応答時間に持ち直した。仮想マシン環境では、CPU はほぼ期待通りの性能が出るが、I/O はときおり期待はずれの性能になることがある。

いつの時代においても、CPU に比べて I/O は低速なものではあるが、長い IT の歴史の中で、I/O の低速性に CPU が足を引っ張られないように様々な工夫がなされてきた。サーバ仮想化は時に、こういった工夫を無効化してしまうことがある。

I/O の本質は、デバイスとメモリの間のデータコピーである。PC ハードウェアでは、大きなデータをやり取りするデバイスとの I/O は DMAC（Direct Memory Access Controller）という、CPU を介することなくデバイスとメモリの間でデータをコピーする専用ハードウェアを介するようになっている。DMAC が低速なデバイスとのやりとりから CPU を開放することにより、CPU は遅い I/O の実行中にも他の処理を進めることができる。

仮想マシン環境では、物理マシン上の DMAC を複数の仮想マシンで共有して利用することになるので、順番の制御や排他制御をソフトウェアで行う必要がある。また、DMAC は物理メモリアドレスしか理解できないが、仮想マシンの OS が物理メモリアドレスだと思っているものは、ハイパーバイザが仮想化したメモリアドレスなので、本物の物理メモリアドレスにするためにはもう一段変換をする必要がある。新しい世代の CPU ではサーバ仮想化における 2 段階アドレス変換を支援するハードウェア機構を備えているが、当時の CPU は備えていなかった。

これらの事情から、DMAC を使用したい状況において、様々なソフトウェアの関与が必要となり、場合によっては DMAC の動作そのものをソフトウェアでエミュレートすることも起こる。ソフトウェアの実行には当然 CPU が必要となる。

DMAC は本来 I/O と CPU を切り離して CPU を開放するための仕掛けのはずなのに、仮想化された環境で DMAC を実行しようとすると結果的に CPU が要求されてしまい、存在意義が本末転倒となってしまうのである。

これら仮想化 I/O のパフォーマンス問題を軽減するために、ハイパーバイザ側でも様々な

ソフトウェア的な工夫は継続されているが、ここ数年で台頭してきたのが仮想化 I/O 対応ハードウェアの利用である。Cisco Systems の VIC M81KR や、Intel 82599 10Gbit Ethernet Controller 等、仮想化支援機構搭載ハードウェアの価格がどんどん安くなり、性能は向上している。特別な理由がなければ最初から対応ハードウェアを選択するべきと言ってよい。

5. ネットワークに起因するトラブル

5. 1 仮想化で増えるケーブル

仮想データセンタを設計する際に必要なネットワークとして、一番最初に考えなければいけないのはストレージネットワークである。

サーバを仮想化すると、ハードウェアサーバだったものは、OS のインストールイメージとサーバの設定情報を集めたデータの塊、仮想マシンイメージデータへと変化する。ストレージに格納された仮想マシンイメージデータを、ハイパーバイザが読み出して実行するのが仮想マシン実行の本質である。したがってストレージアクセスのパフォーマンスと信頼性が仮想データセンタ全体のパフォーマンスを決めてしまうと言っても過言ではない。ストレージネットワークの設計には手を抜かず知恵を絞るべきである。

続いて Ethernet/IP ネットワークである。アプリケーション要件によって導かれるネットワーク、たとえばインターネット接続のための DMZ（DeMilitarized Zone）接続やサーバ間接続のためのいわゆる裏 LAN、物理ハードウェアの運用・監視用ネットワーク、また仮想環境独自のハイパーバイザ通信用の LAN など、これらすべてを二重化して配線すると、1 台の物理サーバから 10 本の Ethernet ケーブルが出てくるような事態も珍しくない。

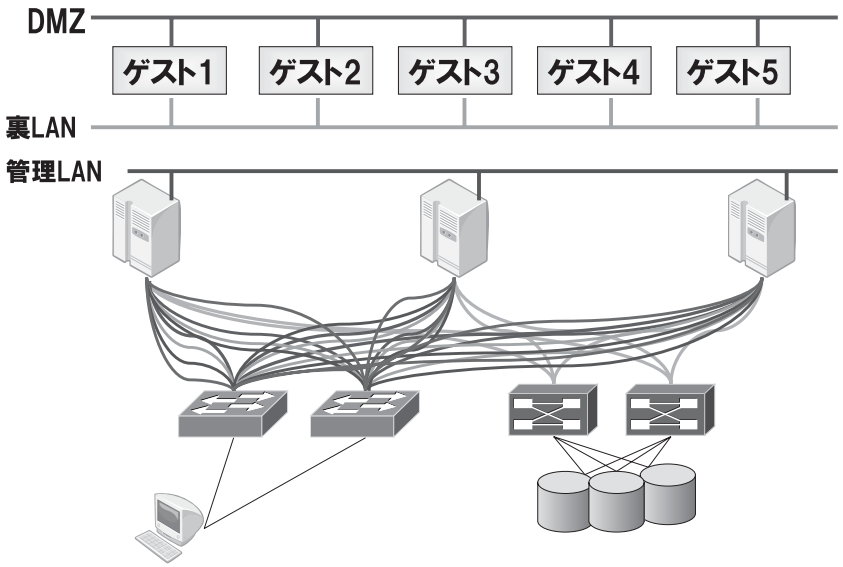

図 3　サーバ仮想化で増えるケーブル

もともと多数の物理サーバにそれぞれつながっていたネットワークケーブルが仮想化で統合されて集まってくることになる。VLAN で束ねようにも、帯域が足りなかったり、QoS やセキュリティや拡張作業などに対する要件の大幅な違いなど、ワイヤリングをどう設計するかに

よってはケーブル数の爆発を引き起こすことも珍しくない。

こうした問題については、10Gb Ethernet への移行が有効である。帯域が拡大することだけでなく、10Gb Ethernet の DCB（Data Center Bridging）機能は、FCoE（FiberChannel over Ethernet）、iSCSI、NFS などのストレージプロトコルを Ethernet に統合する時の効率を大きく改善する。これにより I/O ケーブルの数を大幅に少なくできる。

5. 2　vSwitch の問題：エンジニアの垣根

VMware の仮想スイッチ vSwitch とは、仮想マシンを接続する仮想的な LAN スイッチである。実態はサーバ上のソフトウェアで、VMware のカーネルモジュールとして実装されている。同じ物理サーバ内の仮想マシン同士をつなぐ機能を持っており、仮想マシンが物理マシンをまたがっている場合は、外付けのネットワーク機器を使って接続する。

vSwitch の設定は、サーバエンジニアとネットワークエンジニアの押し付け合いになる。LAN スイッチなのでネットワークエンジニアが設定するのが望ましいが、Cisco CLI（Command Line Interface）のようなネットワークエンジニアになじみのインターフェースがなく、またサーバ上のソフトウェアなので、大抵はサーバエンジニアにまかされることが多く、ずさんな設定になりがちである。

5. 3　vSwitch の問題：QoS やセキュリティの問題

物理サーバ 1 に仮想マシン VM1 ～ VM4 が、物理サーバ 2 に仮想マシン VM5 ～ VM8 があるとする。VM1 ～ VM4 は web サーバなのでインターネットからの HTTP を通すが、VM5 ～ VM8 は DB サーバなので HTTP を通したくない、というようなアクセスコントロールを実現する設定は、以下の 3 カ所に考えられる（図 4）。

1）　仮想サーバ自身で設定
2）　サーバに隣接する LAN スイッチ（エッジスイッチ）で設定
　　（しかし vSwitch にはそのような機能がない）
3）　外部ネットワーク機器で設定

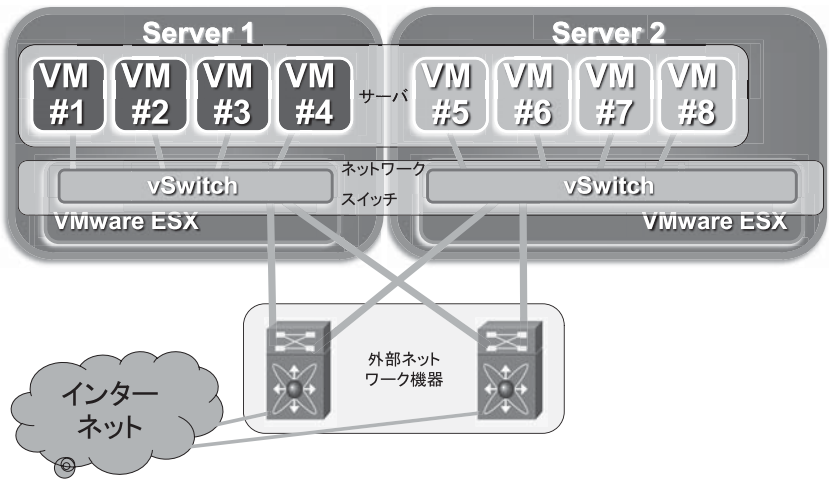

図 4　VMware のアクセスコントロール設定

通常、3)の外部ネットワーク機器でアクセスコントロールを設定するのが常套手段だが、この場合、コントロールの対象は物理サーバとなってしまう。一方 VMware には vMotion があって、仮想サーバが移動してしまうので、大きな問題に発展する。VM1 ～ 4 が物理サーバ 2 に vMotion する場合は web から見えなくなるだけなので問題は少ないと言えるが、逆に VM5 ～ 8 が物理サーバ 1 に vMotion する場合は、DB サーバが望まない形で web に晒されてしまう。解決策は以下の三つである。

1） vMotion をあきらめる
2） ネットワークでセキュリティや QoS を設定するのをあきらめる
3） サーバと外部ネットワーク機器のポート数を増やす

1)は運用自由度を手に入れるせっかくの機能が使えなくなるのでナンセンス、2)は仮想マシン自身への設定となるが、数が多いと管理の負荷が増大する。3)は 8 台の VM x 2 台の物理サーバ x 二重化で 32 本のネットワークケーブルが必要となってしまう。だが、従来はこの三つの解決策を組み合わせて適用するしかなかった。

5. 4　vSwitch の問題：解決策

VMware の vSwitch もバージョンを重ねるごとに高機能化してきており、徐々に問題は緩和されている。また、Cisco Systems 製のソフトウェアスイッチ Nexus 1000V を用いてエッジスイッチを高機能化すれば、QoS やセキュリティを設定できるようになる。設定作業には集中管理コンソールからコマンドラインインタフェースの利用が可能である。物理マシンの境界を越えて仮想スイッチを統合する vNetwork Distributed Switch の機能を利用すれば、物理サーバをまたいで vMotion してもセキュリティや QoS に矛盾を生じない。

こうしたソフトウェアスイッチを使う方法のほか、仮想サーバ I/O に対応した物理スイッチを使用することで、仮想マシンと物理スイッチを論理的に直結する方法もある。この場合、QoS やセキュリティは物理スイッチに設定するが、現在、サーバ仮想化対応ネットワークの実装方式がベンダ毎に異なるので、ベンダ混在環境でポリシーを伴った vMotion を実現することはまだ困難である。

6. お わ り に

仮想化の落とし穴と脱出法について、主なものを説明した。このコンテンツは、ユニアデックスが Web サイトで展開しているオンラインビデオセミナー「仮想化の落とし穴と脱出法 Season 1」および「仮想化の落とし穴から 1 年 今はどうなってるの？」の一部をテキスト化したものである。当該サイトでは、新しいビデオセミナーも続々と掲載されているので、最新情報をご覧になりたい方はぜひご訪問いただきたい。

http://www.uniadex.co.jp/virtualization/

執筆者紹介　高　橋　優　亮（Yusuke Takahashi）
ユニアデックス株式会社のエバンジェリスト。ネットワークの仮想化、ストレージの仮想化、サーバの仮想化など仮想化を中心に、最新の IT 製品やサービスや技術の良さを皆様に知っていただき、その世界を楽しんでいただくことを生業としている。