クラウドネイティブな基盤で進化するIaaS

クラウドというサービスがまだ懐疑的にみられ、及び腰ながらも国内で広まり始めたのが4、5年前のこと。それからわずかの間にクラウドファーストのような言葉が生み出され、急速に普及し、いまやIaaSは一定の完成の域に達しつつあると言ってよいでしょう。これからのクラウドビジネスの主戦場は、IaaSを基盤としたより上位層へと徐々にシフトしていくことになるはずです。

そんななか、2015年11月30日に新しいIaaSである「IIJ GIOインフラストラクチャーP2」をリリースしました（以後P2と呼称します）。高い信頼性と安全性を備えたIaaSをご提供すると共に、今後のサービス開発の基盤となる堅牢なインフラとなります。クラウドが新たなステップへ進むこのタイミングで、将来を見越したIaaSが必要と判断しての新サービスのリリースです。

本稿では、サービスのコンセプトを切り口に、この新しいIaaSの基盤で用いられている技術について解説していきます。

■ 選択できるクラウド

クラウドサービス、特にIaaSの導入を決める過程で多くの方が一度は考えることがあります。パブリッククラウドを利用すべきか、自社内にプライベートクラウドを構築すべきか、です。

利便性や初期投資を考えれば、今後ますますパブリッククラウドの活用が活発になっていくことは間違いありません。その一方で、プロダクトライセンスの条件やコンプライアンスの問題、更には非常に高レベルでの安定的なパフォーマンスなどを考えた場合、必ずしも仮想化されたパブリッククラウドが最善とは言えません。これはいかにクラウドが洗練されようとも、当分は変わることがないでしょう。

こうした悩みを解決すべく、P2のサービスメニューは、大規模な仮想化基盤からなるパブリックリソースと、テナントごとにデバイスを占有できるベアメタルサーバやストレージから構成されたプライベートリソースと呼ばれる2つのサービス体系から構成されています。用途によってはどちらかのリソースだけを利用するでしょうが、両者を任意に組み合わせて1つのシステムを構成することも可能です。パブリックリソースとプライベートリソースを合わせてP2という1つのクラウドサービスなのです（図-1）。

□ コストと効率を両立する複数のリソースタイプ

それからもう1つ、パブリックリソースの中では更に2つのリソースタイプを提供しています。性能保証タイプとベストエフォートタイプです（図-2）。この2種類のリソースタイプを提供するというコンセプトは、サーバとストレージ、ネットワークそれぞれについて一貫してサービスデザインに盛り込まれています。

クラウドサービスにおいて性能を保証するということは、「利用する／しない」にかかわらず、必要量のリソースを固定的に確保しておくということを意味します。例えば、テナントごと

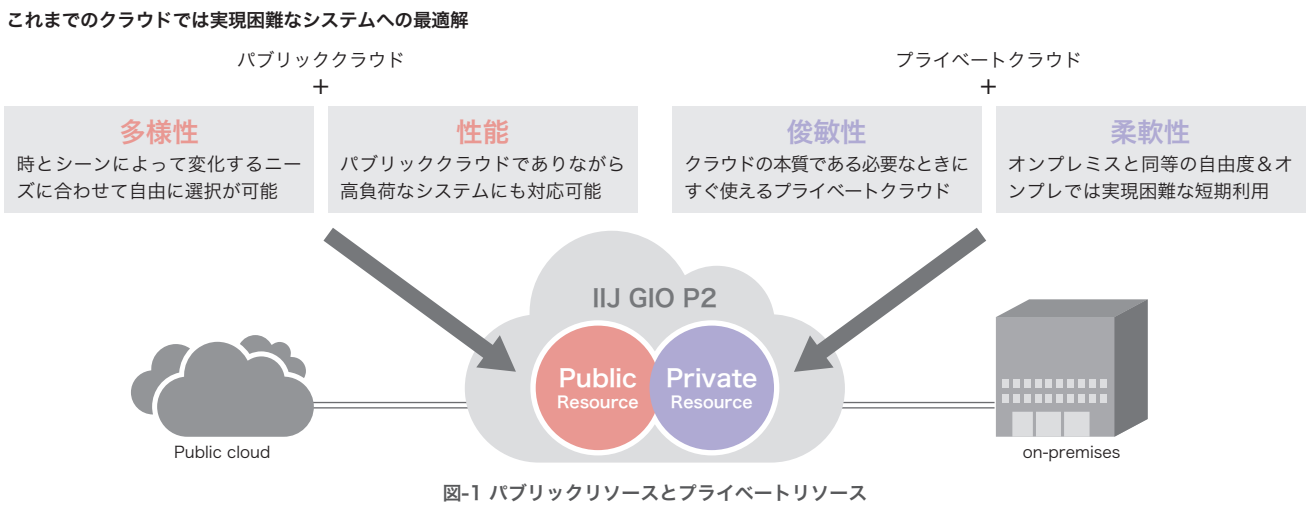


図-1 パブリックリソースとプライベートリソース

に100Mbpsの帯域を割り当て、ネットワーク性能を保証する場合、1Gbpsのネットワークならば10件のテナントを収容可能です。帯域が保証されていれば、安定的に性能が発揮されるので安心感があるものの、実際にはそのような性能は不要であることが多々あり、過剰なコストとなりがちなことが知られています。クラウドを利用しない場合は、回線にしろサーバの性能にしろ、設計段階で定めた能力が固定的に発揮される（はず）ですから、否応なく性能保証タイプと言えます。

一方のベストエフォートタイプは、これとは対極に位置するコンセプトです。先に1Gbpsのネットワークを用意しておき、そこに何件までのテナントであれば快適な性能を維持できるかを見積もって収容をコントロールします。その際、各テナントはある程度のばらつきを持ってネットワークが利用されることを期待します。そうすることで、すべてのテナントに均等にリソースを割り当てることなく、快適さが維持されるというわけです。とはいえ、仕様としては発揮しうる最大値のみが示されるため、実際に利用してみるまで発揮される性能が分からない面もあり、不安に感じることもあるでしょう。実際、ごく小規模なリソースをシェアする場合には、ばらつきが少し偏るだけで性能劣化につながりかねません。しかし、クラウドのような極めて大規模なリソースプールでは、極めて多数のテナントが同居するため、うまい具合にばらつき、効率的にリソースが配分される状態が期待しやすくなります。もっとも、そのためにはリソースを動的に再配分する仕組みが欠かせず、自然とそのようになるわけではありません。

さて、どちらがよいサービスでしょうか？どちらと言わず、性能が保証されていて、なおかつ安価で固定的な料金で利用できるクラウドがあれば、それは大変魅力的ですが、そのためにはなんらかのトリックが存在するはずです。システムリソースが有限である限り、コストか、性能か、信頼性か、トレードオフが発生します。それならば、用途に応じて選択できることが最善と言えないでしょうか。

例えば、社内システムをクラウド上に展開する場合。そのシステムは定常的に必要とされる性能と、ピーク時に必要とされる性能を比較しても、それ程極端な差が生まれにくいはずです。そのようなシステムは安定的に性能を発揮し、コストコントロールの容易な性能保証タイプが適しています。

一方でインターネットサービスのように、平常時とピーク時のシステム負荷に極端な差があるシステムをクラウドに展開する場合。ピーク時を想定したサイジングは主にコスト的な観点から難しいものになりがちです。それならば、コストに比較して上限性能が高く設定されているベストエフォートタイプを活用し、平常時のコストを抑え、ピーク時には発揮した性能に応じたコストをかけるスタイルが適しています。

□ リソースタイプの使い分けで全体を最適化するクラウド

最近ではあまり聞かれない言葉になってきましたが、クラウドの黎明期にはクラウドコンピューティングのエッセンスの1つとして、ユーティリティコンピューティングが必須と言われ

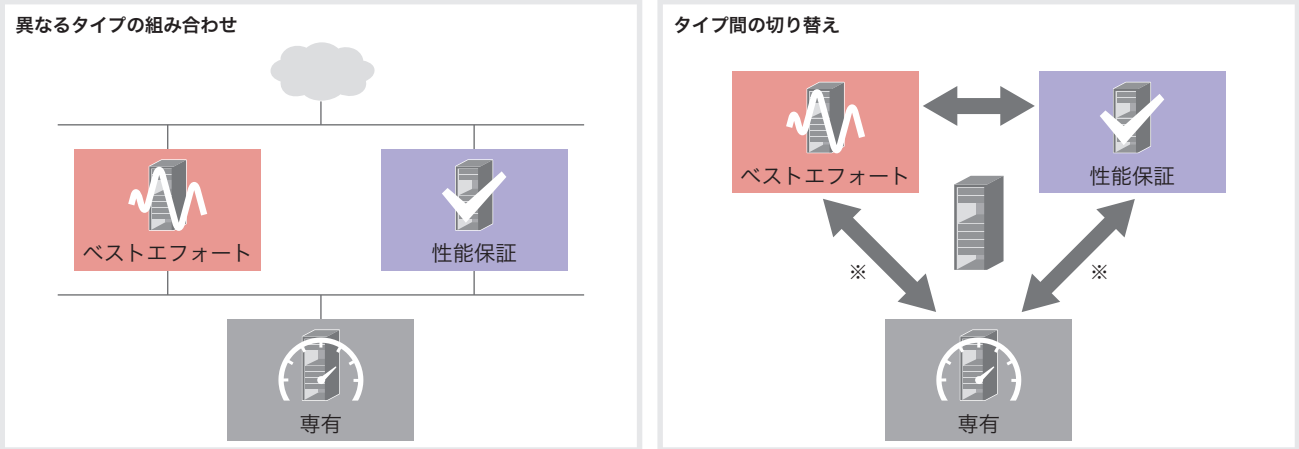


図-2 性能保証とベストエフォート

ていました。コンピューティングリソースを水道・ガス・電気といったユーティリティのごとく利用するという考え方です。

水資源に例えるならば、従来のシステム設計は庭に井戸を掘るようなものです。初期投資は大きく、利用までに時間がかかりますが、その後は汲み上げるだけで追加費用はなく、水を利用することができます。その代わり日照りが続けば水は出なくなり、水質汚染の対策も自身で実施する必要があります。一方でクラウドは水道です。天候に左右されず、安定的かつ安全に水資源を利用できるうえに、初期投資は不要で、使用量に応じたコストだけで利用可能です。大半の利用者にとって水道の方が優れた選択肢になることは言うまでもありませんが、豊富な水源を所有していたり、良質な地下水をビジネスにされていたり、一部の利用者にとってはまた違った判断があるでしょう。

クラウドの登場によって、システムをオンデマンドかつ迅速に変化させることができるようになりました。一方で、従来と同等のスタイルでのシステムも一定の条件で利があります。パブリックとプライベート、そしてクラウド上でのリソースタイプの使い分けが、システム全体の効率を決める鍵となるのです。

■ 規模から価値を生み出すクラウドネイティブな基盤

こうしたコンセプトを実現するP2の基盤とはどういったものか、お話ししたいと思います。

我々は2009年に「IIJ GIOサービス（以後GIOと呼称します）」を立ち上げ、クラウドサービスを発展させて参りましたが、その基盤がピュアなクラウド基盤であったかと言えば、当初からそうだったわけではありません。伝統的な設計がなされた物理基盤の上にクラウドサービスを組み上げたもの、それがこれまでのGIOだったと言えるでしょう。これはよく機能し、安定的で、クラウドらしさと引き換えにもたらされる不便さのない、安心して使えるクラウドサービスとしてご利用いただいてきました。

これはクラウド黎明期においてはよい構成だったと思いますが、今の市場で求められるクラウドらしい魅力を提供するサービスの基盤としては、いささか物足りないものとなりました。そこで、数万台規模のクラウドを提供してきた経験から得られた知見を活かし、クラウドネイティブな基盤上に刷新されたクラウドサービスがP2というわけです。

□ クラウド基盤で活きるSDN

さて、クラウドネイティブな基盤とはなんでしょう。1つ特徴を挙げるならば、従来ハードウェアが担っていた機能のソフトウェアによる実現です。つまり、SDN（Software Defined Networking）、SDDC（Software Defined DataCenter）といった技術の導入です。

一時期、ネットワークの仮想化やOpenFlowといったキーワードと共に世間をにぎわしていた感のあるSDNですが、最近は下火に感じられる方もいるのではないでしょうか。サーバの仮想化であればそのメリットは広く浸透し、今や利用して当然の技術となっていることと比較すると、意外に思われるかもしれませんが、それはSDNに価値がなかったからではなく、広く一般にメリットをもたらす技術とは言い難いからです。ある程度条件が整った環境でこそ真価を発揮する技術であり、その環境がクラウド基盤なのです。

P2のネットワークは、その構造に着目すると実はさほど特殊なものではありません。むしろ、できる限りシンプルで、ごく普遍的な技術のみで実現されているといっていいでしょう。ただし、規模と用途が特殊と言えます。

例えば、数百台のサーバから構成されたシステムは従来であれば十分大規模と言えましたが、クラウド基盤としてはいささか小規模です。そうした桁違いの規模のサーバやストレージをネットワークで結び、巨大なリソースプールとして扱う環境というのは、クラウド基盤以外にはなかなかありません。巨大であるということは、通常想定されている様々な上限を超える可能性があるということです。それはVLANの最大数だったり、学習可能なMACアドレスの最大数だったり、帯域だったりと様々ですが、いずれにしても単一の物理デバイスが想定している規模を容易に超える規模であり、SDNを導入し、ソフトウェアで制御しなければ実現し得ないのです。

また、仮想化されたマルチテナントシステムであることもクラウド基盤を特殊たらしめています。いかに巨大なリソースプールであっても、それが物理デバイス単位で切り出され、テナントに割り当てられていれば、それは小規模なシングルテナントシステムの寄せ集めでしかなく、クラウドらしいスケーラビリティや柔軟なシステム構成は実現できません。それに対して、クラウド基盤はどのネットワーク上のどのサーバやストレージにでも、いかなるテナントでも収容できる巨大なマルチテナントシステムです。それが故に、オンデマンドに必要なリソースを割り当て、迅速に展開できるクラウド基盤が実現しているのです。

一定の規模を超えてなおスケールメリットを追求した結果、SDNの採用は必然でした(図-3)。

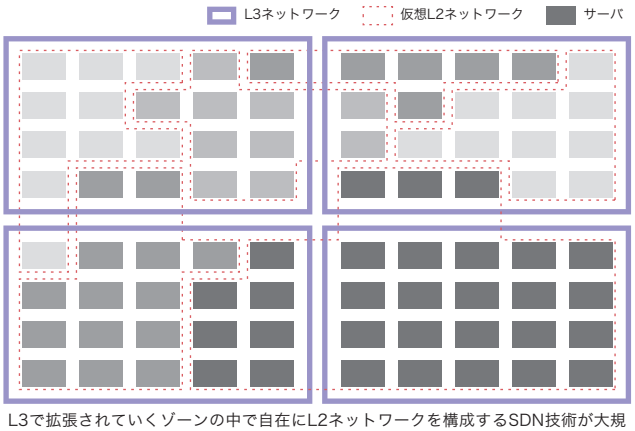
□ ライフサイクルの長期化

P2を構成するソフトウェアはミドルウェアこそ多数のOSSを活用していますが、基本的にオリジナルのソフトウェアスタックで構成されています。その一部であるSDNの制御システムもやはりオリジナルのSSPと呼ばれるシステムと、P2のオーケストレータが連携することで実現されています。

クラウドベンダーがこうした独自の基盤ソフトウェアを実装するのは、それ自体が他にはない特徴を備えていることもありますが、クラウドを構成するデバイスやソフトウェアのライフサイクルから解放され、クラウド基盤のライフサイクルを長期化できることにメリットを見出しているからでもあります。これはクラウドサービスの長期安定化に大きく貢献します。

IT業界では4、5年をサイクルとしてシステム更改を繰り返すことは、半ば暗黙の了解とされているかもしれません。しかし、昨今ではデバイスの性能向上のペースは以前に比べると控えめになりつつあります。それに引き換え複雑さは増す一方であり、リプレースの負担が増えつつあることから、システムのライフサイクルをできるだけ長期化したいと考えるケースが増えているのではないでしょうか。

サーバに関しては、仮想化技術の普及によって物理的な環境に依存しないシステム運用が可能になりつつあります。一方でネットワークの仮想化は一般的ではなく、ネットワークの制御は特定のプロダクトに依存した状態が普通です。シングルテナントシステ



L3で拡張されていくゾーンの中で自在にL2ネットワークを構成するSDN技術が大規模なクラウド基盤を実現する

図-3 SDN

ムであり、ハードウェア、OS、アプリケーションなどのライフサイクルがそろっていれば、それがデメリットになることはあまりありませんが、マルチテナントシステムであるクラウド基盤ではそうはいきません。様々なテナントが思い思いのライフサイクルでシステムを運用されています。それがクラウド基盤のライフサイクルと一致することなどありえません。今後、クラウドサービスには長期的に継続してサービス提供されることがますます望まれていくことは間違いないでしょう。

■ 止まらないクラウドを支える技術

P2のクラウド基盤は大規模であるが故にSDNを必要としていることを述べましたが、もう1つクラウド基盤の運用を支える技術としてライブマイグレーションについて触れたいと思います。SDNと同じように、ライブマイグレーションもクラウド基盤に固有の技術ではありません。むしろ、ほとんどのハイパーバイザに機能として備わっているため、日常的に活用している方もいることでしょう。もっとも、用途によっては「ライブ」ではなく、停止を伴うマイグレーションで十分なケースが多いかもしれません。

一方で、クラウドサービスにおいて、ライブマイグレーションは基盤に欠かせない機能の1つです。それは、ユーザに極力影響を及ぼすことなく様々なメンテナンスを実施し、クラウドをヘルシーに保つためです。

例えば、クラウド基盤に脆弱性が発覚した場合。マルチテナントな基盤に脆弱性があれば、深刻なものであれば広範囲に影響が出る可能性がありますから、迅速な対応が求められます。そのような場合、ライブマイグレーションを用いて仮想マシンを追い出してしまい、空になったハイパーバイザを安全にアップデートしていきます。

例えば、一部の仮想サーバにトラフィックが集中した場合。ベストエフォートタイプの仮想サーバはすべてが平等とは言えませんが、それでもこの種の極端な事例では対象となるサーバを隔離し、同一設備に収容される他テナントへの影響を一部にとどめる努力がなされています。

□ ライブマイグレーションの仕組み

ライブマイグレーションとは、仮想マシンを停止することなく、異なるハイパーバイザ上へ移し替える技術の総称です（図-4）。ただし、この「停止しない」というのが曲者で、場合によっては一定の停止時間が発生します。このロジックを簡単に説明しましょう。

まず、ライブマイグレーションが実行されると、次のような処理が行われます。

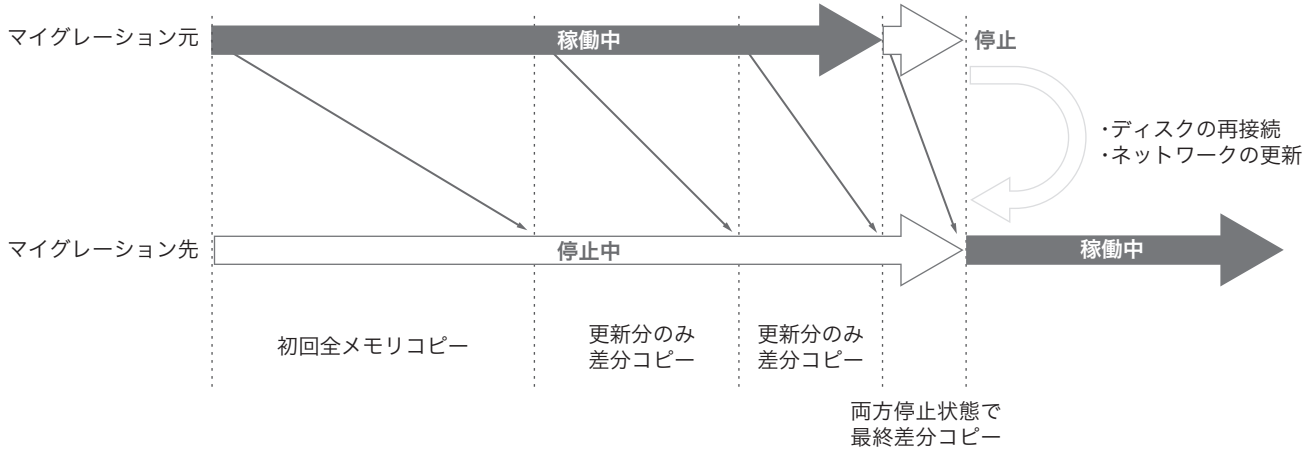


図-4 ライブマイグレーションの仕組み

1. マイグレーション先の仮想マシンを準備
2. 仮想マシンのメモリをすべて転送
3. 仮想マシンをサスペンド
4. ストレージをマイグレーション先のハイパーバイザに再接続
5. CPUのステートなど、仮想マシンの各種状態を転送
6. マイグレーション先の仮想マシンをレジューム
7. ネットワークスイッチでMACアドレスを再学習

基本的に仮想マシンが持っているすべての情報を転送する必要があるわけですが、多量のデータを持っているのはストレージとメモリです。ただし、ストレージのデータはマイグレーションするにはあまりにも多量であるため、ライブマイグレーションを利用する仮想環境ではローカルディスクを持たず、すべてiSCSIやNFSで接続されたリモートストレージに置かれるのが一般的です。逆に言えば、ローカルディスクを搭載したサーバを提供するクラウドサービスでは、ライブマイグレーションによる恩恵には預かれず、クラウドベンダーの都合で仮想サーバを停止する可能性があるということです。

したがって、ライブマイグレーションにおける停止時間を決めるのはほぼメモリです。P2の場合、48GBのメモリを搭載した仮想サーバを利用可能ですが、このメモリを、すべてネットワークを介して転送する場合、最善の状態でも1分近い時間を要します。この転送時間に仮想マシンを停止しては、ライブマイグレーションになりません。そこで、メモリの転送はまず仮想マシンを動かしたままで行われます。すると当然転送中にもメモリが更新されることになりますが、その差分は次回に転送されます。この処理を必要に応じて複数回実行し、差分が十分小さいと判断されると仮想マシンがサスペンドされ、最後の転送が行われます。この最後の転送量がライブマイグレーション中の停止時間を決めます。つまり、単純な仮想マシンの搭載メモリ量ではなく、メモリの更新量がライブマイグレーションを左右するのです。これが、ライブマイグレーションがあたかも無停止でハイパーバイザを移動できる仕組みです。

ただし、あまりにもメモリの更新頻度が高いシステムの場合、何度差分転送を繰り返しても転送量が減らず、最終的には一定時間の停止が発生する可能性もあります。経験的には、定常的に高負荷がかかるデータベースサーバは頻繁にメモリが更新されており、ライブマイグレーションとは相性が悪いシステムの1つです。そこで、P2では計画的なメンテナンスを行う前に、事前に仮想サーバの停止をお願いする場合があります。一度仮想サーバを停止した後、再度起動するとアップデート済のハイパーバイザへ収容されるため、ライブマイグレーションの対象から外れ、不意の停止を避けることができます。

クラウド基盤は極めて複雑なシステムで、限られた誌面では語り尽くすことはできませんが、ネットワークと運用にフォーカスして、ご紹介しました。もしP2の基盤に興味を持っていただけたなら、昨年実施したIIJの技術イベント「IIJ Technical WEEK 2015」の講演[*1]で少し触れていますので、そちらもご覧になってください。

執筆者:
田口 景介（たぐち けいすけ）
IIJ プラットフォーム本部 プラットフォームサービス1部 部長。
2008年、クラウド「IIJ GIOサービス」の企画立ち上げからプロジェクトに参画。
「IIJ GIOインフラストラクチャー P2」ではパブリックリソースのサービスマネージャを務める。

＊1 IIJ Technical WEEK 2015「刷新されたIIJ GIOクラウド基盤におけるSDNの実践」2015年11月11日 講演資料(http://www.iij.ad.jp/company/development/tech/techweek/pdf/151111_1.pdf)。