2

OSS/OpenStack に見るクラウド基盤技術

基専

西島　直（(株) 日立製作所）

クラウド概要

クラウドの定義

近年，IT システムの利活用に大きな変化が起きている．所有から利用へ，すなわち，自分たちでハードウェアやソフトウェアのリソースを所有せずに一時的に借りて，自分たちのやりたいことをすぐに開発できるようになってきた．この環境により，以前は充実した開発環境を持つ大手企業しか提供できなかった規模のサービスでさえ，誰でも容易に短期間で開発・提供できるようになり，多くのビジネスチャンスが生まれている．このようなコンピューティングリソースの利用形態をクラウド・コンピューティングという．

「クラウド・コンピューティング」という用語は，2006 年，当時の Google の CEO であった Eric Schmidt が Search Engine Strategies Conference で使ったのが最初とされている．インターネットはよく「雲」で描かれるが，そんな「雲」の中にリソースもデータも置いておき，Web でいつでもどこからでも利用できるのが「クラウド・コンピューティング」だと表現した．

その後，多くの企業が競ってこのモデルに沿ってこのクラウド・コンピューティング基盤を構築し，サービスを展開したが，統一した定義や規格がなかったため，2009 年アメリカ国立標準技術研究所（NIST）が産学連携でそれらのサービスを分類して，クラウド・コンピューティングの定義を公表した[1]．NIST は，クラウド・コンピューティングを，ネットワーク／サーバ／ストレージ／アプリケーション／サービスなどのコンピューティングリソースをオンデマンドに利用できるモデルと定義した．このモデルは以下の 5 つの特性（仕組み）を持つ．

① オンデマンド・セルフサービス
② 幅広いネットワークアクセス
③ リソースの共用
④ スピーディな拡張性
⑤ サービスの計測可能性

また，その特性を使って利用者に届けるサービスとして，IaaS（Infrastructure as a Service），PaaS（Platform as a Service），SaaS（Software as a Service）の 3 種類のモデルを定義し，これらのサービスを提供する基盤をどこにどう置くかの配置モデルも，プライベートクラウド，コミュニティクラウド，パブリッククラウド，ハイブリッドクラウドの 4 種類を定義した．各定義内容を**表 -1** に示す．

本稿では，この後，このクラウド・コンピューティングをクラウド，クラウドのサービスを提供する基盤をクラウド基盤，IaaS/PaaS のクラウド基盤をそれぞれ IaaS 基盤，PaaS 基盤と記す．

OSS クラウド基盤の誕生

現在のクラウド基盤の形を最初に作ったのは Amazon といっても過言ではない．2006 年，Amazon Web Services（AWS）においてパブリッククラウドの IaaS である Amazon Elastic Compute Cloud（Amazon EC2）/ Amazon Simple Storage Service（Amazon S3）の提供を開始し，IT 業界に大きな影響を与えた．Amazon のクラウド基盤ソフトウェアは独自開発であるが，仮想化を実現するハイパーバイザには OSS（オープンソースソフトウェア）の Xen が使われている．Amazon はそのサー

特　性	オンデマンド・セルフサービス	リソースが必要なときに利用者自身で調達可能
	幅広いネットワークアクセス	インターネットなどのネットワーク経由でアクセス可能
	リソースの共用	リソースの所有が不要
	スピーディな拡張性	欲しいときにすぐに調達可能
	サービスの計測可能性	リソース利用量等が計測でき，従量課金などを可能にする仕組みが存在
サービスモデル	IaaS（Infrastructure as a Service）	利用者は，提供されているサーバ，ストレージ，ネットワークから利用分を確保し，任意のソフトウェアをデプロイして利用
	PaaS（Platform as a Service）	利用者は，提供されているプログラミング言語やツールを用いて，アプリケーションプログラムを仮想マシン上にデプロイして利用 ネットワーク，サーバ，ストレージ，OS などの管理・制御は不可
	SaaS（Software as a Service）	利用者は，Web ブラウザやクライアントデバイスを通して，仮想マシン上で稼働するアプリケーションプログラムにアクセスして利用 ネットワーク，サーバ，ストレージ，OS などの管理・制御は不可
配置モデル	パブリッククラウド	サービスプロバイダと契約して利用するタイプで，個人から企業まで，一般に広く利用が可能
	コミュニティクラウド	パブリッククラウドの一種で，グループ会社など，限られた範囲内で利用
	プライベートクラウド	自社サーバにクラウドシステムをインストールして利用するタイプで，自社内でのみ利用．ネットワークの接続範囲限定
	ハイブリッドクラウド	パブリックとプライベートの両方を組み合わせたタイプ

表 -1　NIST の定義

ビスの豊富さと価格の安さでシェアを急速に伸ばし，パブリッククラウド市場で常にトップを走り続けている．PaaS では 2008 年に Microsoft の Windows Azure，Google の Google App Engine が相次いで発表された．SaaS では特定のアプリケーションに特化したサービスプロバイダが多いが，Salesforce は多様なアプリケーションに応用できる柔軟な基盤の上に豊富なアプリケーションのメニューを提供している．

一方，プライベートクラウドでは，仮想化ソフトウェア市場でトップシェアであった VMware が，2009 年にその仮想化技術を活かし IaaS ソリューションの提供を開始した．自社の既存サーバを有効活用してクラウドの恩恵が得られること，自社データを社外に持ち出す必要がないことなどから，大手企業を中心に採用が進んでいる．

このようなクラウドサービスのプロバイダの動きに対抗するように登場したのが，OSS のクラウド基盤ソフトウェアである．独自にクラウドのビジネスを展開したいベンダと，ベンダロックイン回避や自分の利用に適した環境を自由に構築し，最新技術の恩恵も迅速に受けたい利用者の思惑が重なり，複数のプロジェクトが立ち上がり，活発に開発が行われた．

OSS において開発が先行したのは IaaS 基盤である．最初に普及したのは，カルフォルニア大学で開発され 2008 年に公開された Eucalyptus であった．Amazon EC2/S3 互換のプライベートクラウド環境を構築できることが特徴で，Web サービスを行う企業などで採用された．2010 年には Cloud.com が CloudStack をリリースした．複数種類のハイパーバイザのサポートやプライベートネットワークを実現する VPN（Virtual Private Network）の提供など汎用性があり完成度も高く，本番環境での稼働実績数も多い．Citrix が買収し開発を継続していたが，2012 年に Apache Software Foundation に寄贈された．同じく 2010 年に，米航空宇宙局（NASA）と Rackspace がリリースしたのが OpenStack である．OpenStack はさまざまな用途に利用できる汎用的なクラウド基盤を構築することを目的としたソフトウェアである．開発参加者の急増や大手企業の協賛を受け，2012 年には OpenStack Foundation が設立され，プロジェクトの運営が移管された．Rackspace が OpenStack をクラウド基盤に使って IaaS を開始し，IBM が OpenStack を自社クラウドビジネスの基盤に採用，さらには HP や VMware が OpenStack ディストリビューション計画を発表したことなどが追い風となり，他を凌駕する勢いで成長している．現在，約 2 万人が開発に参加する巨大

プロジェクトであり，急速に普及している[2]．

最近では PaaS 基盤の開発も進んでいる．OpenStack では Hadoop 等のデータ処理基盤サービスの Sahara やデータベースサーバの構築・管理サービスの Trove が追加された．そのほかには，Pivotal Software が開発を主導する Cloud Foundry，Red Hat が開発を主導している OpenShift がある．また，Cloud Foundry をベースとする IBM の BlueMix，HP の Helion Development Platform なども次々と発表された．

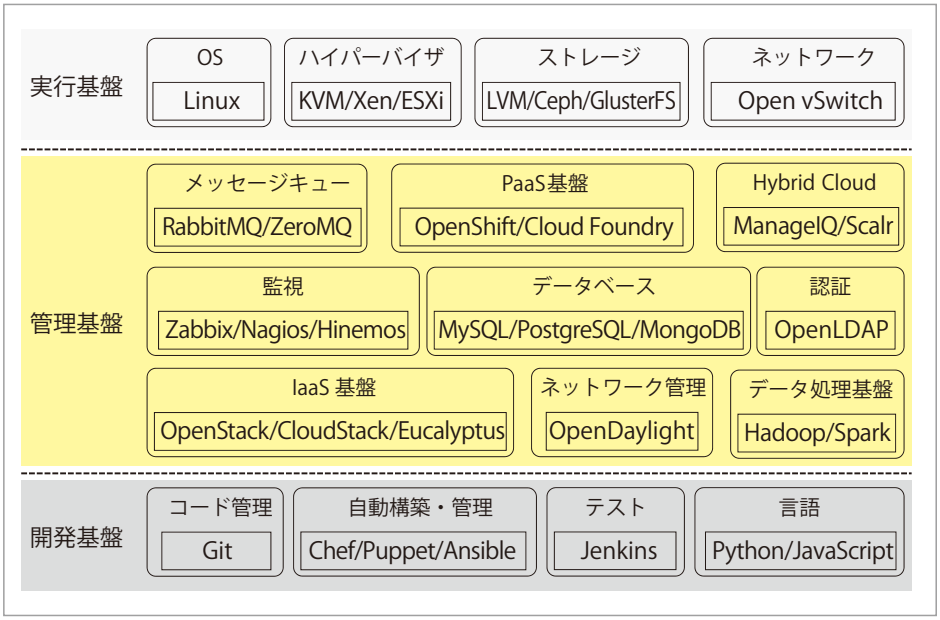


図 -1　クラウド基盤（IaaS/PaaS）で利用される OSS

この状況を見ても分かるように，クラウド基盤では IaaS/PaaS の開発が活発に進んでおり，OSS も多く利用されている．本稿では IaaS/PaaS 基盤に焦点を絞って解説する．

IaaS/PaaS 基盤における OSS

IaaS/PaaS 基盤の構成と OSS

まず，クラウド基盤について整理する．クラウド基盤は以下の 3 種類に大別できる．

1）実行基盤：仮想マシンを実行

2）管理基盤：実行基盤を管理しクラウドを運用

3）開発基盤：実行基盤，管理基盤の開発環境

これらの基盤は複数のコンポーネント（ソフトウェア）で構成されている．たとえば実行基盤は，OS（オペレーティングシステム）やハイパーバイザ，ネットワーク，ストレージがそれにあたる．管理基盤では，IaaS 基盤，PaaS 基盤，データ管理，ネットワーク管理，認証，監視などがある．開発基盤には，開発言語，コード管理，テスト実行基盤，自動構築・管理などがある．

これらには数多くの OSS が使われている．**図 -1** にその例を示す．ここでは各々の OSS の詳細は説明しないが，この後の解説の中で適宜説明を行う．IaaS/PaaS 管理はクラウドサービスそのものを構築・運用するソフトウェアである．図 -1 を見ても分かるように，IaaS/PaaS の両方に関係している OSS として OpenStack がある．AWS と互換性を持ち，IaaS/PaaS の要件に対する開発が急速に進んでおり，クラウド基盤の状況が分かりやすい．本稿では，OpenStack を中心にクラウド基盤の技術を解説する．次節ではこの OpenStack について説明する．

OpenStack 概要

OpenStack は汎用的なクラウド基盤を構築するための OSS であり，2010 年に最初のバージョン Austin がリリースされた．OpenStack は半年ごとにリリースされ，その名称はアルファベット順につけられ，OpenStack Summit ☆1 が開催された地域に関連する名前を採用している．OpenStack は複数のサービスコンポーネントを持つモジュラー

☆1 OpenStack Summit とは，年に 2 回開催される数千人規模が参加する国際カンファレンスであり，事例や技術開発の発表や企業展示が行われている．また，次のバージョンの仕様や課題を議論して方針を決める Design Summit も同時に開催されている．

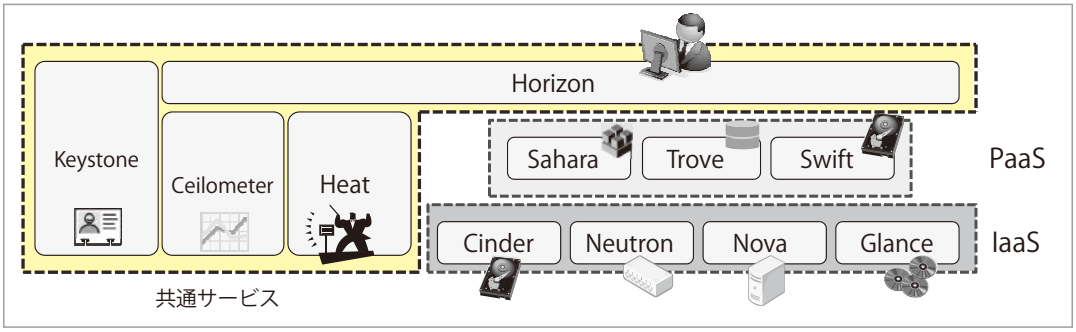


図 -2　OpenStack Juno の構成

コンポーネント	提供するサービス	概　要	リリース名と内容								
			Austin	Bexar	Cactus	Diablo	Essex	Folsom	Grizzly	Havana	Icehouse	Juno
Nova	仮想マシン	仮想マシンの起動・停止などの操作を行う	●	●	●	●	●	●	●	●	●	●
Swift	オブジェクトストレージ	オブジェクトストレージを管理する	●	●	●	●	●	●	●	●	●	●
Glance	仮想マシンイメージ	仮想マシンの起動イメージを登録・削除を行う		●	●	●	●	●	●	●	●	●
Horizon	ポータル	各プロジェクトは REST API を持っておりコマンドラインから操作できるが，Horizon はそれらの操作を Web から行える				●	●	●	●	●	●	●
Keystone	認証	Horizon のログインや，各プロジェクトの操作の認証のために利用される				●	●	●	●	●	●	●
Neutron（旧 Quantum）	ネットワーク	仮想マシンが通信するためにネットワークだけでなく，ロードバランサや FireWall 等の機能も提供する						●	●	●	●	●
Cinder	ブロックストレージ	仮想マシンへの永続ブロックストレージを提供する．バックエンドは LVM（Logical Volume Manager）や Ceph などの OSS から日立や EMC 等のストレージを利用することができる						●	●	●	●	●
Heat	オーケストレーション	テンプレートを書くことで，複数の仮想マシンを作成することや，仮想マシンにミドル・アプリケーションのインストールなどを自動化することができる								●	●	●
Ceilometer	計測	各仮想マシンのリソースを計測する								●	●	●
Trove	データベース	MySQL や MongoDB 等のデータベースを提供する										●
Sahara	データ処理基盤	Hadoop や Spark のデータ処理基盤を提供する										●

表 -2　OpenStack のコンポーネント

アーキテクチャであり，必要なモジュールだけを組み合わせることができる．コンポーネント間を HTTP ベースの REST（Representational State Transfer）API や AMQP（Advanced Message Queuing Protocol）を使った非同期通信で接続し，分散型の協調動作を行っている．その API は Amazon EC2 と Amazon S3 と互換性があり，AWS のアプリケーションを最小限の移植作業で OpenStack にて使用することができる．

OpenStack は当初，IaaS 基盤として開発された．Austin では，仮想マシン環境の構築・管理を行う Nova とオブジェクトストレージ管理の Swift の 2 つのコンポーネントの構成だった．NIST 定義の特性のうち，サービスの計測可能性以外の 4 つを備え，基本的な IaaS 基盤を実現していた．

開発が進むにつれ Nova の機能が肥大化したため，機能ごとのコンポーネントに分割し，平衡開発することで開発の効率向上が図られた．Grizzly までは IaaS 基盤の開発に注力していたが，Havana からは運用管理の効率向上のため，オーケストレーションサービスの Heat と計測サービスの Ceilometer が追加された．この Ceilometer の追加により，NIST 定義の 5 つの特性をすべて満たす実装となった．最新の Juno の構成を**図 -2** に，機能の概要を**表 -2** に示す．この後の解説でも OpenStack のコンポーネント名が出てくるので，この図を参照してほしい．

OpenStack の特徴の 1 つとして，機能がコンポーネント化され API で互いに呼び出せることと，プラグインやドライバ登録の仕組みにより機能を自由に差し替えたりできることが挙げられる．このため，ベンダは自社製品と組み合わせて OpenStack を使うことも多く，OpenStack の技術動向を見ることで，次のクラウドの方向性も見えてくると考える．

IaaS/PaaS 基盤の技術進化

クラウド基盤は，その普及が広がるにつれ利用者や運用者からの要件も増えている．そしてその要件に応える技術開発が行われており，それを支えているのも OSS である．本章では，IaaS/PaaS 基盤における，仮想環境の構築と管理，性能保証の 2 つの技術について解説する．

仮想環境の構築と管理の容易化

クラウドを利用，もしくはクラウド基盤を運用する場合，最初に直面する問題が仮想環境の構築とそのリソース管理である．これらにかかわる技術で最近注目されているのが自動化である．

これまでのシステム構築では，構築手順を作成してインストールやセットアップを行っていたが，大規模データ処理や Web 上の大規模分散システムをはじめとして，何百台ものシステム環境を作るケースが増えており，手作業での構築には限界があった．そこで，手順書の代わりにすべてをソースコードに書いて自動実行することが行われるようになった．さらに，一度作った手順書を他のサーバ構築やシステム構築でも簡単に再利用できるような工夫を含み，複数台のサーバの OS やミドルウェアの構築・管理を自動化する「オーケストレーションツール」が生まれた．OSS の著名なオーケストレーションツールには Chef/Puppet/Ansible などがある．たとえば Chef では，パッケージ管理ソフトウェアのインストール手順を記述するのではなく，「Web サーバが 1 つ存在する」という“システム構成のあるべき状態”をソースコードの形で「レシピ」と呼ばれるファイルに記述する．実際のシステムの構築手順は「クックブック」と呼ばれる複数のレシピをまとめたものに記述されていて，Chef はクックブックの手順どおりにレシピの内容を構築する．また，ソースコードを Git などのバージョン管理ソフトウェアで履歴管理しておけば，システムの更新に失敗しても，以前のコードからすばやく更新前のシステムを復旧できる．このようにインフラ，OS，ミドルウェアの構築・設定をソースコードに書いて，誰が行っても同じ基盤を構築できるようにすることを「Infrastructure-as-Code」という．この技術は，Web サービスなど，頻繁にアプリケーションを拡張・更新しながら運用する「DevOps」型アプリケーション開発を中心に，活用が広まっている．

OpenStack でもオーケストレーションツールとして Heat が開発された．Heat とは，仮想マシン環境の構築に必要なシステム構成情報をまとめてテンプレートファイルに記述し，自動で一括構築を行うサービスである．テンプレートファイルにはシステム構成のみを記述すればよく，詳細な API 呼び出しを知らなくともサーバ，ストレージ，ネットワークを構築できるようになった．Chef や Puppet を呼び出せるので，一元的な構成管理もしやすい．このテンプレートの 1 つとして OASIS（Organization for the Advancement of Structured Information Standards）標準の TOSCA（Topology and Orchestration Specification for Cloud Applications）がある．OpenStack では IBM が中心となり，Heat で TOSCA テンプレートを利用可能とするための開発が進められている．

PaaS 基盤の自動構築技術はテンプレートからさらに進化し，「Web サーバ」「データベースサーバ」という機能やいくつかの非機能要件を指定すれば，それらを実現するシステムを自動的に構築することができる．前述した OpenShift や Cloud Foundry などはその機能を提供する OSS である．

これらの技術から，利用者が目的に合う環境を 1 つずつ構築するのではなく，利用者が“最終的にやりたいこと”を伝えれば，クラウド基盤が自動構

築する時代へと進んでいることが窺える．構築や管理の容易性はクラウドの利用の普及に大きく影響する要素である．今後も“簡単に使えて管理が楽”なサービスの提供に期待したい．

性能保証の向上

クラウド基盤の運用者には，クラウド利用者に提供するサービスに対する保証が求められる．特に性能保証は利用者のビジネスを左右しかねないため求められることが多い．

一方で，クラウド基盤の運用者の立場からすると，1 台のサーバに多くの仮想マシンを作成し，最小限の物理リソースで最大限に利益をあげたいという思いがある．しかし，1 台のサーバに複数の仮想マシンを作成するとリソースの競合（ノイジー・ネイバー問題）が起こり，性能保証ができなくなる可能性がある．

この課題に対し，OpenStack では，Linux の cgroups というリソース制限機能を用いて仮想マシンが利用するリソースを限定する機能を提供している．たとえば，ある対象仮想マシンの CPU 負荷が高くなっても設定した量以上の CPU は割り当てられることはなく，同じサーバ上の他の仮想マシンのリソースに影響を与えることはない．本件は本特集 5.「基幹システムを実現する Linux 技術」記事でも説明しているので参照してほしい．また，ブロックストレージのクォータ（運用上の上限）設定，ネットワークの帯域制限などの開発も行われている．

また，リソース競合が起こらないように物理サーバを割り当てるサービスがある．IBM の SoftLayer などでは，物理サーバも仮想マシンと一元管理して，確実な保証を求める場合は物理サーバそのものを契約できるようにしている．

OpenStack でも仮想マシンと物理サーバの一元管理機能の開発が進んでいる．2015 年春リリースの Kilo で，物理マシンを管理するコンポーネントの Ironic が採用される予定である．これが実現できれば，高速で安定した性能保証を仮想マシンと同じ管理で提供できるため，利用企業の期待は大きい．なお，物理マシン管理機能を最初に開発し，コミュニティに寄贈したのは NTT ドコモである [3]．

別の解決方法として，仮想マシンのオートスケーリングで解決を図る方法もある．負荷を監視し，上昇してきたら仮想マシンの台数を増やし分散処理を強化する．ただし，追加仮想マシンのコスト，負荷の検知時間，仮想マシンの調達方法などの課題もあり，限定的な提供にとどまっている．

今後の動向と技術

クラウド基盤の技術は数が多く，しかも進化が早い．今回はすべてを紹介することはできなかったが，少しでも役に立つ情報があれば幸いである．

最後に，今後開発が活発化していくであろう，新たな潮流について，ここで簡単に紹介する．

ハイブリッドクラウド

ハイブリッドクラウドは，重要なデータはプライベートクラウドで運用し，一時的に必要となるシステムのリソースはパブリッククラウドで運用するなど，システムの特性に応じで異なるクラウドを使い分けることでコストを削減することができる．パブリッククラウドとプライベートクラウドをシームレスに使える IBM PureApplication Software V2.0 や HP Cloud Service Automation，パブリッククラウド同士をシームレスに使える日立のフェデレーテッドクラウドなど，発表が相次いでいる．OSS では，ManageIQ という会社が開発した ManageIQ，SCSK が開発した PrimeCloud Controller，Scalr という会社が開発した Scalr などがある．たとえば Scalr では，Amazon EC2，Rackspace Cloud，Google Compute Engine などのパブリッククラウドと，OpenStack などのプライベートクラウドをつなぐ仕組みを提供している．ハイブリッドクラウドは，パートナーの幅を広げることで今後も用途は拡大していくだろう．すでに，OpenStack では，欧州原子核研究機構（CERN）と Rackspace がプラウベートクラウドでリソースが足りなくなった場合，パブリッククラ

ウドのリソースを借りることができる技術を開発している．新しいビジネスが生まれる可能性もあり，大いに期待したい．

コンテナ技術 Docker

ハイブリッドクラウドの普及に伴い，アプリケーション実行環境の移行が重要な課題となってくる．そこで，注目されているのがコンテナ技術である．コンテナとは，OS のリソースを論理的に分割して，それぞれ別の空間を作り出す技術である．コンテナの説明も本特集 5.「基幹システムを実現する Linux 技術」記事を参考にしてほしい．コンテナの概念自体は 10 年以上前からあったが，最近注目を浴びた理由の 1 つは，コンテナ技術を利用してアプリケーション実行環境のポータビリティを高めた Docker の登場である．Docker は Docker という会社が開発した OSS であり，LXC（LinuX Containers）と呼ばれる Linux コンテナ技術と Aufs（Advanced multi layered unification filesystem）という特殊なファイルシステムを利用してコンテナ型の仮想化を実現している．

従来のコンテナ技術はリソースを分割するものであり，その上で動作するアプリケーションの実行環境には関与しない．そのためコンテナ内でアプリケーションの実行環境を構築するためには，実行に必要なツールや設定を利用者側ですべて用意する必要があった．Docker はその実行環境をイメージ化して管理できるようにし，構築の手間を省いている．Docker で構築した環境は，物理マシンでも仮想マシンでも動作できるため，クラウドにおいても，アプリケーション実行環境のポータビリティの向上手段として注目されている．すでに，Amazon，Google，VMware，Microsoft などが自社クラウドでの Docker サポートを表明している．また，OpenStack でも，Heat から Docker を利用することが可能である．

Immutable Infrastructure への対応

クラウド上で構築されたシステムの更新・管理に関して，新しい考え方が出てきているので紹介する．Immutable Infrastructure（不変なサーバ基盤）という考え方である．Immutable Infrastructure は，サーバ環境を一度構築したらそのサーバのソフトウェア構成を変更せずに運用するというコンセプトである．

通常，サーバ環境は構築後にソフトウェア構成の変更がしばしば行われ，場合によってはそれがアプリケーションの安定稼働に大きな影響を与えることがある．これに対し，Immutable Infrastructure では，ソフトウェア構成の変更が必要となった場合には既存環境とは別に新しい環境を構築し，新しい環境で動作確認が取れた場合に限り，アプリケーション実行環境を移して既存環境を廃棄する．これにより，新しい環境への更新が失敗しても既存環境で業務を継続できる．

この考え方は，クラウドのようにリソースを十分に使える環境が整ってきたからこそ，そしてその環境を OSS で身近に構築できるようになったからこそ，生まれたといえる．今後の選択肢の 1 つとなるのか注目したい．

クラウド・コンピューティングの用途は，すでに情報系から基幹系へ，さらに IoT（Internet of Things）の普及に伴うビックデータ解析の場へと広がってきている．OSS によって，その開発の利点でもある各種開発ベンダや利用者との共創を活かし，操作性，信頼性，連携性，さらには未知の利用の可能性までも追及し続けていくことを期待している．

参考文献

1）Mell, P. and Grance, T.：The NIST Definition of Cloud Computing：NIST Special Publication 800-145 (2011).
2）OpenStack Web サイト，http://www.openstack.org/
3）NTT ドコモ：クラウドサービスの性能を向上させる新技術を開発，報道発表資料（2013/4/8）．

（2014 年 11 月 14 日受付）

西島　直　nao.nishijima.xt@hitachi.com

2007 年東京理科大学理学研究科数理情報科学専攻修了．同年，日立製作所に入社．横浜研究所所属．Linux カーネルの研究・コミュニティ活動，OpenStack およびクラウドシステムにおける高信頼システムの研究に従事．