Technology Reports

保守・運用業務の効率化　OpenStack　TCO削減

クラウド環境と仮想化技術を用いた大規模OSS移行の実現

ドコモ・テクノロジ株式会社　ソリューションサービス事業部　岸川 雄紀（きしかわ ゆうき）　石居 健太郎（いしい けんたろう）

ドコモは，サービス監視や装置監視を担うOSSについて，利用するHWのサポート終了に伴い設備更改の検討を行い，保守・運用業務の効率化の観点から，OpenStackプライベートクラウドと仮想化技術の導入を行った．本稿では，従来のOSSの課題と更改要件，更改方式の検討結果，更改方式における課題と対策，業務効率化のための機能開発，導入効果と今後の課題について解説する．

1. まえがき

安定したモバイルサービスを継続的に提供するため，ドコモではサービス監視や装置監視を担うオペレーションサポートシステム（OSS：Operation Support System）*1の開発および商用導入を行っている．

今回，OSSで利用するハードウェア（HW）のサポート終了（EOL：End Of Life*2）に伴い，将来的な更改も見据えた設備更改の検討を進める事となった．その際，多くのHWで構成されるシステムの構築とOSやHW変更に伴うソフトウェア（SW）開発の効率化が重要なテーマとなっていた．本稿では，この課題解決のためOpenStack*3プライベートクラウド*4環境を構築し，OSS SWを仮想化環境で構築／運用する上での課題や対策案の検討，結果について解説する．

2. 従来のOSSの構成と課題，および更改要件

(1)HWの課題

従来のOSSは，千台オーダーの汎用のブレード型IAサーバ*5，ストレージ，ネットワーク機器から構成されている．また，大規模の災害を想定した効率的なディザスタリカバリ*6の仕組み［1］～［3］を導入し，OSSの継続性を確保している．従来のOSSの構成イメージを**図1**に示す．

©2017 NTT DOCOMO, INC.
本誌掲載記事の無断転載を禁じます．

＊1　OSS：事業者の運用支援システム．通信事業者の場合，提供しているサービスを運用するために，ネットワークやシステムの「障害管理」「構成管理」「課金管理」「性能管理」「セキュリティ管理」のすべて，もしくは一部を行う．

＊2　EOL：製品の生産／販売終了，SW製品のサポートサービス終了，バグや機能改善の修正・更新プログラムの提供終了を指す．

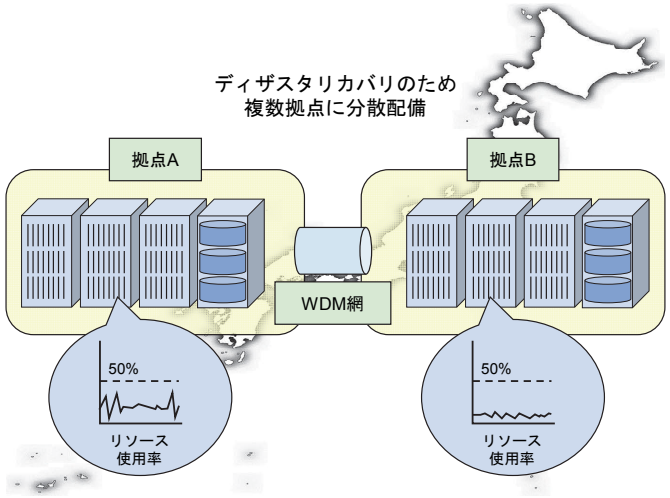

図1　従来のOSSの構成イメージ

災害やネットワークの輻輳*7のような緊急時においても，OSSを継続的に運用できるようHWリソースに余裕をもたせることは重要だが，一方でそのような構成では，平常時はリソースの使用率が低くなり，設備の利用効率が下がる．さらに，初期導入時の設備投資（CAPEX：CAPital EXpenditure*8）や，HW設置スペース借用代・電気代・保守費などの運用費用（OPEX：OPerational EXpense*9）が余計に必要となるなど，従来のOSSには設備コスト面の課題がある．

⑵ソフトウェアの課題

SWの観点では，OSSが利用するOSのバージョンアップに伴う動作検証が課題となっている．例えば従来のOSSで採用しているバージョンのCentOS*10は最新HW製品ではサポートされておらず，設備更改によってHWを新調すると，SWが利用するOSも更新する必要があり，これに対する動作検証も実施しなければならない．ドコモではこれまでも定期的に設備更改（OSバージョンアップに伴う動作検証を含む）を行ってきた．これに対して，SWのOS依存度を適切に管理し，影響を局所化することでOSのバージョンアップに伴う開発投資の最適化も重要である．

以上の課題解決を目的とした今回の設備更改の具体的な要件を**表1**に示す．これら要件を満たすための更改方式の検討を行った．

3. 更改方式の検討結果

設備更改の要件を満たすためには，設備の利用効率の改善が必要である．そこで1つのハイパーバイザ（HV：HyperVisor）*11に複数のアプリケーションを重畳させることで利用効率の改善を図る．

⑴従来の分散アーキテクチャ

従来のOSSアプリケーションは分散アーキテク

＊3　OpenStack：サーバ仮想化技術を用いて，一台の物理サーバを仮想的に複数のサーバのように動作させ，仮想サーバをユーザが利用するクラウドサービスごとに割り当てるクラウド基盤のSW．オープンソースSWとして提供されている．

＊4　プライベートクラウド：企業・組織が社内でクラウド環境を構築し，社内の各部署やグループ会社に提供するクラウド形態を指す．一方利用・提供する人の範囲が限定されず，オープンに提供されるクラウドサービスはパブリッククラウドと呼ばれる．

＊5　IAサーバ：Intel社製のマイクロプロセッサ，およびIntel互換プロセッサを搭載したサーバ．内部構造はパソコンとほぼ同様であり，他社製プロセッサベースのサーバと比べて安価であることが特長．

＊6　ディザスタリカバリ：自然災害などで被災したシステムを復旧および修復させること．また，被害を極小化させるための予防措置のこと．

チャを採用しており，D3A（Distributed Data Driven Architecture）*12と呼ばれる基盤SW［4］と，その上で動作する複数のエレメント（EL：ELement）*13と呼ばれる分散アプリケーション群から構成されている．D3A基盤は分散配備されたEL間のデータ通信を受け持ち，EL機能のシンプル化を実現している．従来のOSS SWの分散構成のイメージを**図2**に示す．

⑵HV型仮想化技術の採用

従来の分散アーキテクチャに対して，さらにHV型の仮想化技術を用いることで，1つのHW上に複数の仮想マシン（VM：Virtual Machine）*14を構築・動作させることができ，効率的に設備を利用することが可能となる．

HV型仮想化技術を採用したOSS SWのシステム構成イメージを**図3**に示す．従来のD3A基盤活用方式ではD3A基盤の仕様上，同一HWに同一種別のELを2つ以上実装する事ができなかったが，本技術はVMごとに独立した動作空間となるため，同一HW上に同一種別のELをインストールしたVMの重畳が可能となる．またHWの更改によりOSやミドルウェア（MW：MiddleWare）*15に変更が生じた場合も，HVがサポートするOSであればその変更は隠蔽されVMには影響が及ばない．結果，OSバージョンアップの影響を基本的に回避する事が可能となり，長期的なSW開発投資の最適化が実現できる．HV型仮想化技術は表1の設備更改の要件を最大限に満たすことができることから，今回は本方式を採用した．

なお，ドコモはOSS構築にあたり無償のオープンソースSWを活用する方針のため，今回の設備更改では，HVはLinux®*16標準の無償仮想化機能である

表1　設備更改の要件

通　番	要求条件
要件1	HWリソース最適利用による効率化
要件2	SW開発投資の最適化
要件3	HWの投資効率化

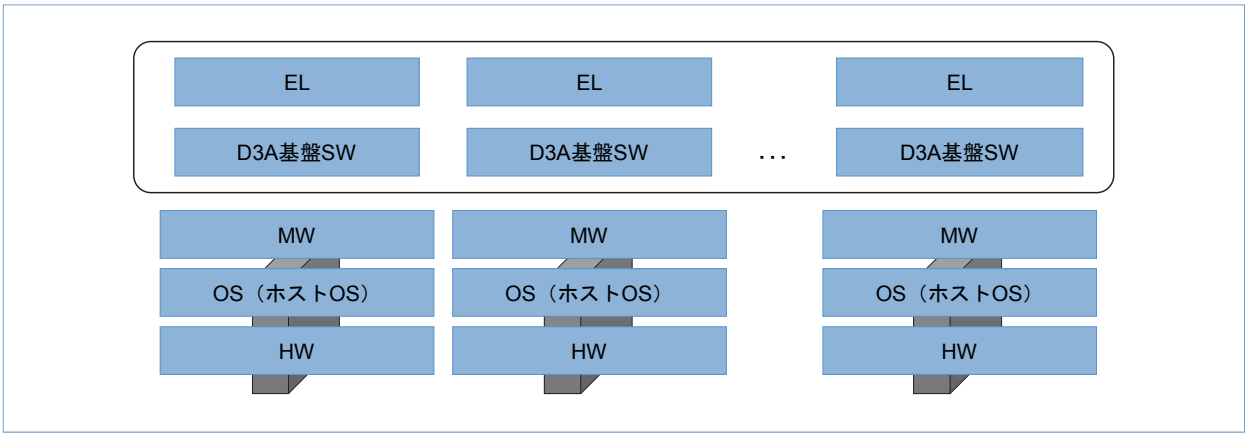

図2　OSSアプリケーションの分散構成イメージ

＊7　輻輳：通信の要求が短期間に集中して通信制御サーバの処理能力を超え，通信サービスの提供に支障が発生した状態．
＊8　CAPEX：設備投資のために支出する金額．
＊9　OPEX：設備を維持し運用するために発生する費用．
＊10　CentOS：Red Hat Enterprise Linux（＊16参照）に含まれているSWのソースコードを基に，レッドハットの商標や商用パッケージなどを除去したものをリビルドし，完全互換をめざしたフリーのLinuxディストリビューション．
＊11　ハイパーバイザ（HV）：サーバ仮想化技術で，物理リソース上で複数の仮想マシンを実行するため，仮想マシンに実装されたSWに対する物理リソース資源の割付け・管理を行う．
＊12　D3A：ドコモにおいて開発した，複数のIAサーバを束ねて高い処理能力を得ることが可能なアーキテクチャ．

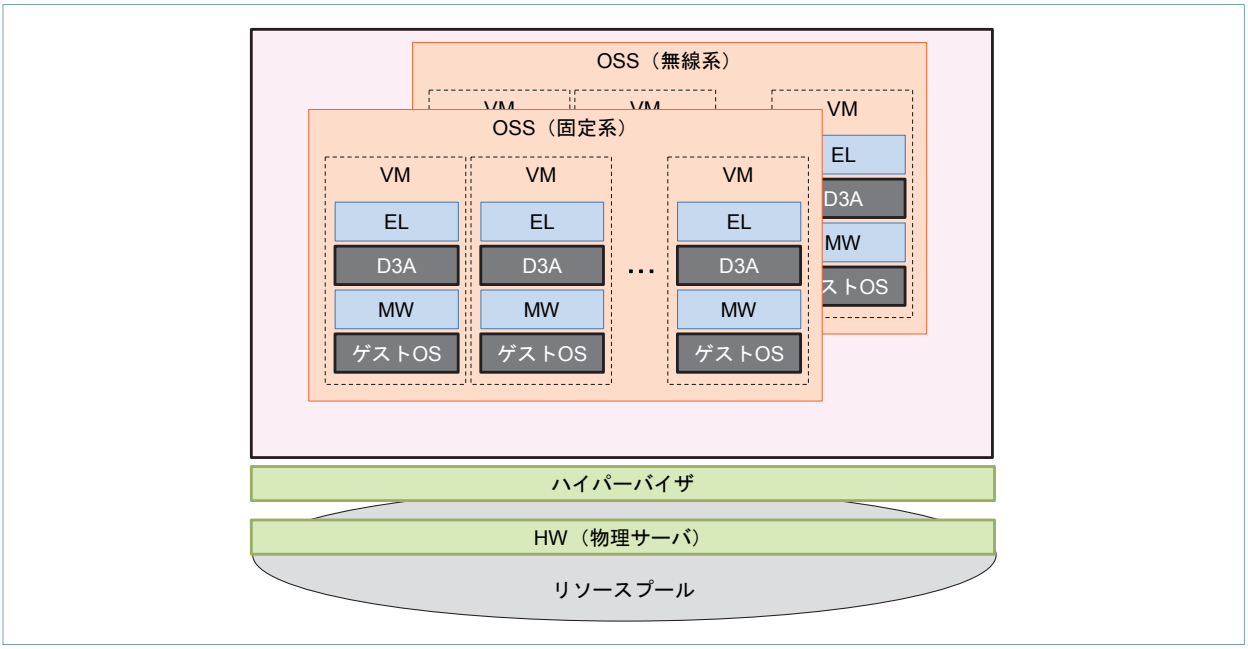

図3　HV型仮想化導入後の構成イメージ

KVM（Kernel-based Virtual Machine）[*17]を採用する．

⑶プライベートクラウド方式の採用

インフラ面では，従来のOSSは独自に環境構築を行ってきたが，本更改で仮想化技術を導入するにあたり，OpenStackを用いたプライベートクラウドを利用する方式を採用した．特に余剰リソースを有効活用する観点においても，OSS独自の環境構築や，社外のクラウドサービスを利用するより，会社全体で効率的な運用が図られる．

4. クラウド利用に伴う課題

⑴クラウドの利用による課題

従来のOSS SWはもともとオンプレミス[*18]の物理サーバ運用を前提に構築されているが，更改により仮想化・クラウド環境となることに伴い，新たな課題が見つかった．例えば，冗長構成がACT/SBY[*19]となるELにおいて，オンプレミス環境では搭載される物理サーバが分かれているため，HW故障による両系同時停止は極めてまれであり，OSSがサービス停止に至る可能性は極めて低かった．しかし仮想化・クラウド環境では，同一のHW上に冗長構成がACT/SBYとなるELが搭載される可能性があり，1つのHW故障で両系停止に至る恐れがある．また，共有ディスク構成であったELはACT/SBYの切替え時に共有ディスクのマウントを切り替えてデータの一貫性を保っていたが，現状のOpenStackはマウントが適切に外れない場合があり，共有ディスクのマウント切替えが失敗し，復旧に時間を要して業務への影響時間が延伸する恐れがある．そのほかに，複数のACT構成のELや性能確保のため同一HWに搭載しているELなど，従来のOSS物理構成を意識しているELは，仮想マシンの配置を適切にコントロールする必要がある．具体的な課題事項とイメージを**表2**と**図4**⑷に示す．

＊13 エレメント（EL）：D3Aの基本構成で，SWを機能単位に分割し，複数サーバで分散稼働させる事を可能とする仕組み．ELはJava® VM上で動作するため特定のHWベンダに依存しない．OracleとJavaはOracle Corporationおよびその子会社，関連会社の米国およびその他の国における登録商標．

＊14 仮想マシン（VM）：SWによって仮想的に構築されたコンピュータ（仮想マシン）．

＊15 ミドルウェア（MW）：オペレーションシステムとSWの間に位置し，OS上で動作するすべてのSWに共通する基本機能や汎用的な処理を集約実装し，SWに提供する．

＊16 Linux®：GPL（GNU Public License）に従って自由に再配布可能なUnix系のオープンソースOS．Linux®は，Linus Torvaldsの米国およびその他の国における登録商標あるいは商標．

＊17 KVM：仮想化を実現するSWの1つ．オープンソースとして提供されている．

表2　クラウド利用に伴い解決すべき課題

通　番	検討事項
1	ACT/SBY構成のEL：冗長性を確保するためACT側/SBY側EL（VM）を同一HW上に搭載しない
2	複数ACT構成のEL：冗長性を確保するためすべてのACT EL（VM）を同一HW上に搭載しない
3	動作上の関連が大きい一部EL：処理性能を確保するため関連EL（VM）を同一HW上に搭載する
4	共有ディスク構成のEL：VM停止時にマウントが適切に外れないOpenStackの制約のため，共有ディスク方式を個別ディスク方式に変更する

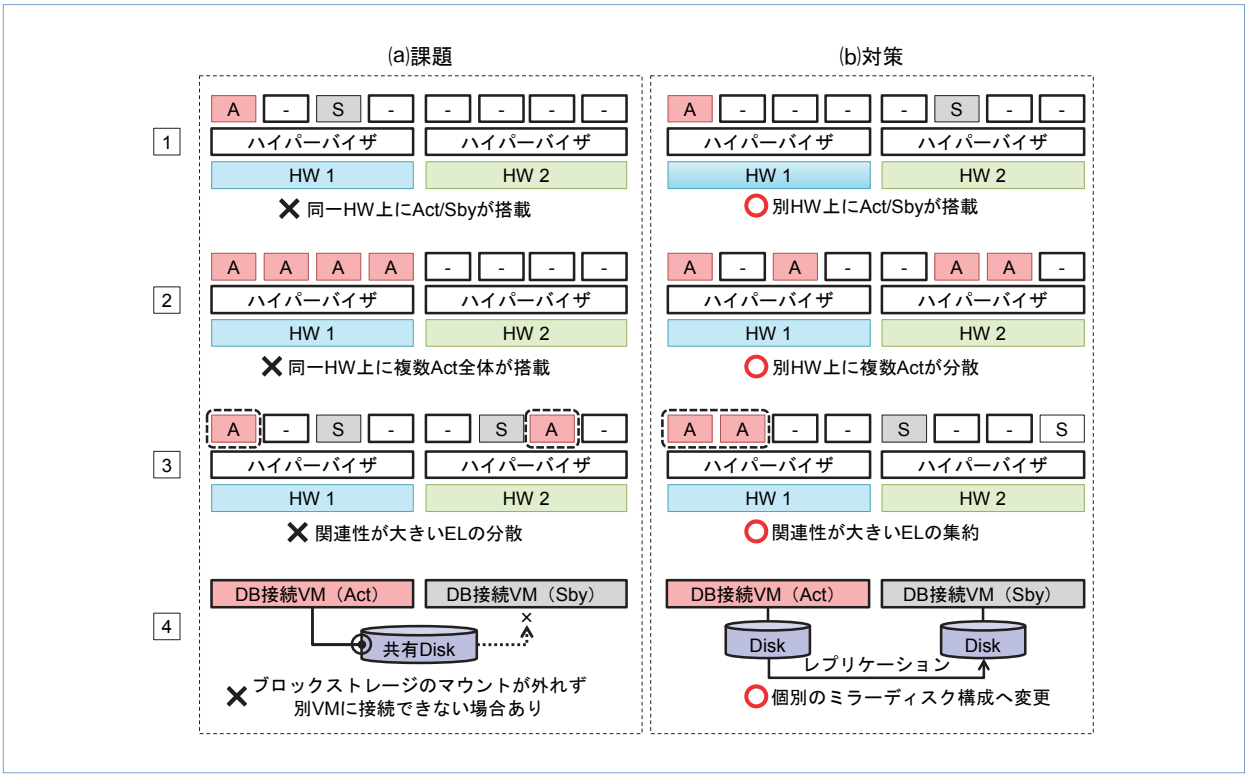

図4　課題と対策のイメージ

(2)対　策

クラウド環境を利用し，HWの選定やクラウド環境自体の設定変更が難しい場合，SWの，仮想化への仕様変更による課題解決が必要となる．対策イメージを図4(b)に示す．表2通番1～3の具体的な対応方法としては，OpenStackに具備された特定のHWへのVM配置を指示できるアベイラビリティゾーン＊20指定機能やフィルタ機能（Different Host Filter，Same Host Filterなど）[5]を用いる方式を採った．前者は物理的な隔離や冗長性確保のため，特定のラックやフロア，データセンタなど，仮想マシンの物理的配置先をユーザ側で指定する機能である．後者も同様の機能で，仮想マシンの起動ルールをより細かく指定することができる．通常これらは手動で

＊18 オンプレミス：企業がシステムを構成するHWを自社で保有し，自社で保守運用すること．

＊19 ACT/SBY：同一機能を搭載したサーバを2台設置し，1台を運用し（ACT系），残りの1台を待機状態（SBY系）にすること．ACT系サーバ故障時に即座にSBY系に処理を引き継ぐことでサービス停止を防止するシステム構成．なお，SBY系サーバは切替えに備えて，常時ACT系サーバと同じ状態を保持している．

＊20 アベイラビリティゾーン：OpenStackでは，リージョンと呼ばれる地理的に離れたデータセンタなどを指す領域があり，アベイラビリティゾーンはさらにそのリージョン内で複数のラックや電源系統ごとに独立したロケーションを指す．

指定するものだが，ドコモにおけるOSSのような大規模なシステムの場合，1システムでVMが1,000オーダーもあるため，人手による指定は稼働が膨大となり，人的ミスの要因となる恐れがある．そのため，今回はこれらの制御を行う仮想化コントローラを導入した．

表2通番4においては，ストレージの構成変更とACT/SBY間のデータレプリケーション＊21実装のSW修正により対応した．これにより従来のOSS SWの冗長性や性能・機能を維持したまま，仮想化・クラウド環境上への移行を可能とした．

5. HV型仮想化技術による課題と対策

HV型仮想化技術はSWによりHWをエミュレートし，かつ複数の仮想マシンが1つのHWリソースを共有する方式であるため，仮想層（SW処理）による性能劣化が発生するデメリットがある．このため，HV型仮想化技術における特有の性能劣化について，特にストレージ性能とネットワーク性能については十分な検討・対策が必要である．

今回の設備更改では，利用するOpenStackプライベートクラウド環境における外部ストレージのIO性能＊22が従来の方式に比べ低下する事が確認された．そのため，クラウド環境におけるHWとして外部ストレージの利用を避け，より高性能のサーバ内蔵ディスクを利用する事とした．これにより性能面の対策は完了したが，データの永続性に課題が残る．

OpenStack仮想化環境では，外部ストレージに保存されたデータは基本的に永続であり，仮にVMが停止した場合でもディスク内のデータは残存する．そのため再度VMを起動し当該ストレージをVMに接続すれば保存データが再利用できる．しかしOpenStackのストレージ種別の1つであるエフェメラルストレージ＊23に保存されたデータは非永続であり，VMを削除すると保存データも削除されてしまう． OSSには装置の監視制御履歴やトラフィックデータなど一定の期間保持が必要なデータがあるため，SWのデータレプリケーション機能を修正し，対象のデータの永続性を確保した．具体的には，ACT/SBY関係のELの場合はSBY側ELの内蔵ディスクにACT側ELのデータをレプリケーションする対策を実施した．具体的なイメージを**図5**に示す．

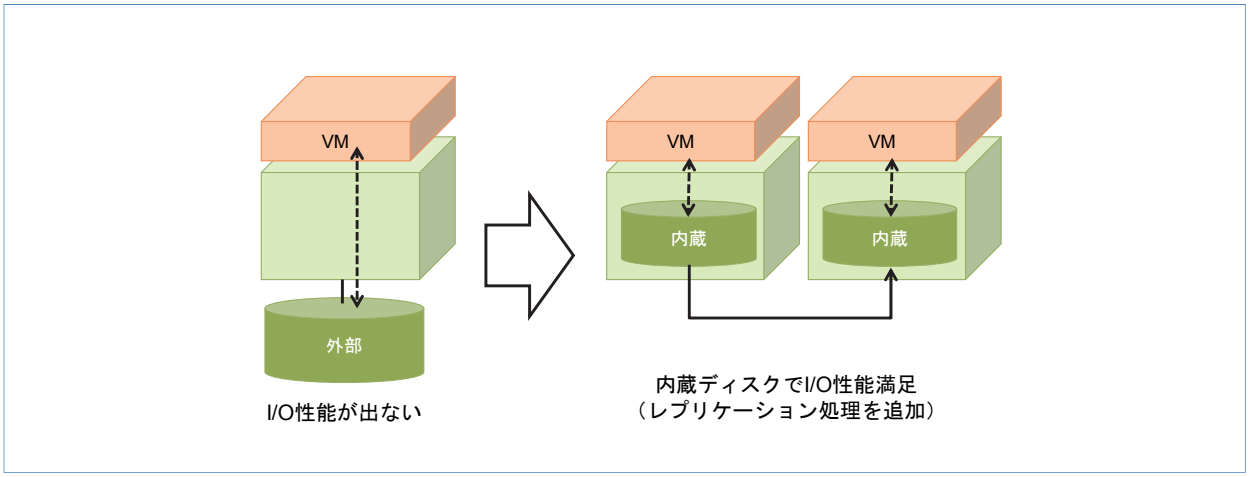

図5　性能改善に伴う永続データの取扱対策

＊21 レプリケーション：レプリケーションは1対1のファイルシステム間で行われる．レプリケーションを行うファイルシステム間はペアでレプリケーションセットを定義し，転送先のシステムに格納されていないデータをコピーする．コピー対象と同じデータが格納されている場合は差分をコピーする．

＊22 IO性能：CPUやメモリが，ストレージやディスクに対してデータや信号をやり取りする際のinput/output性能を指す．

＊23 エフェメラルストレージ：OpenStackの揮発性ストレージ．インスタンスを削除するとストレージ内のデータも削除される．

6. 保守運用業務のさらなる効率化

物理環境による運用方式では，設備を増設する際，規模にもよるがHWの調達・構築・SWインストールに2～3カ月程度要していた．一方で仮想化技術を用いるとシステム構築に必要なHWは論理化され，仮想的なサーバ，ネットワーク，ストレージはSW制御により，迅速な構築が可能となる．またシステムの検証に環境の変更を必要とする場合，仮想環境の停止も同様に迅速化できるため，構築・停止を組み合わせることで**表3**に示した機能が実現できる．

⑴3つの機能開発

仮想化技術の導入は，HW利用効率を向上させ，さらに余剰リソースを有効活用する事で業務効率化が実現できる点に加え，保守業務の効率化や運用業務の自動化によるさらなる業務効率化が可能となる．今回の設備更改ではこれを見据え，3つの機能開発を実施した．

・インスタンシエーション

インスタンシエーションはVMを生成する事を意味する．物理マシンへの仮想マシンのインストール，仮想ネットワーク設定，仮想マシンの起動，その他各種設定の投入を経てSWが起動され，利用できる状態にする一連の動作を指す．今回はこれをワンタッチで実現する．

・ヒーリング

ヒーリングはVMおよびSWの冗長構成の復旧を意味する．ACT/SBYの冗長構成とする仮想マシンの片系が失われた場合，アプリケーション層でACT/SBYの切替えを行い運用が続けられるが，当該SWは片系運転の状態となる．ここでさらに故障が発生し仮想マシンが停止するとシステム停止に至るため，冗長構成の復旧が急務となる．今回の開発により，冗長構成の復旧までの一連の手順をワンタッチで実現する．

・スケールアウト

スケールアウトはVMの増設を意味する．稼働するVMの性能が逼迫した際に同一種別のELをインストールしたVMを増設する事で性能向上を図ることが可能だが，これまではHWの増設工事から計画を行う必要があり，突発的な性能懸念に対するHW増設の対応は困難であった．今回の開発により，設備増設の一連の手順をワンタッチで実現する．

⑵仮想化コントローラの導入

さらなる効率化を実現するOSSのアーキテクチャを**図6**に示す．表3に示す各機能の実行・管理は仮想化コントローラが担い，OpenStack/D3A基盤と連携して各機能を実現する．インスタンシエーションやヒーリング，スケールアウトはOpenStackとD3A機能を活用すれば手作業でも実現は可能である．しかし仮想化・プライベートクラウド導入による保守作業の迅速化，簡易化の効果を最大化するため，実行・管理機能を自動化するシステムを導入した．

本仮想化コントローラは基本的にSWや仮想マシンの起動・停止を制御するものである．従来のOSSは

表3　さらなる効率化に向けた開発機能

通　番	検討事項
機能1	VM新設のワンタッチ化（インスタンシエーション）
機能2	冗長構成のワンタッチ復旧（ヒーリング）
機能3	設備のワンタッチ増設（スケールアウト）

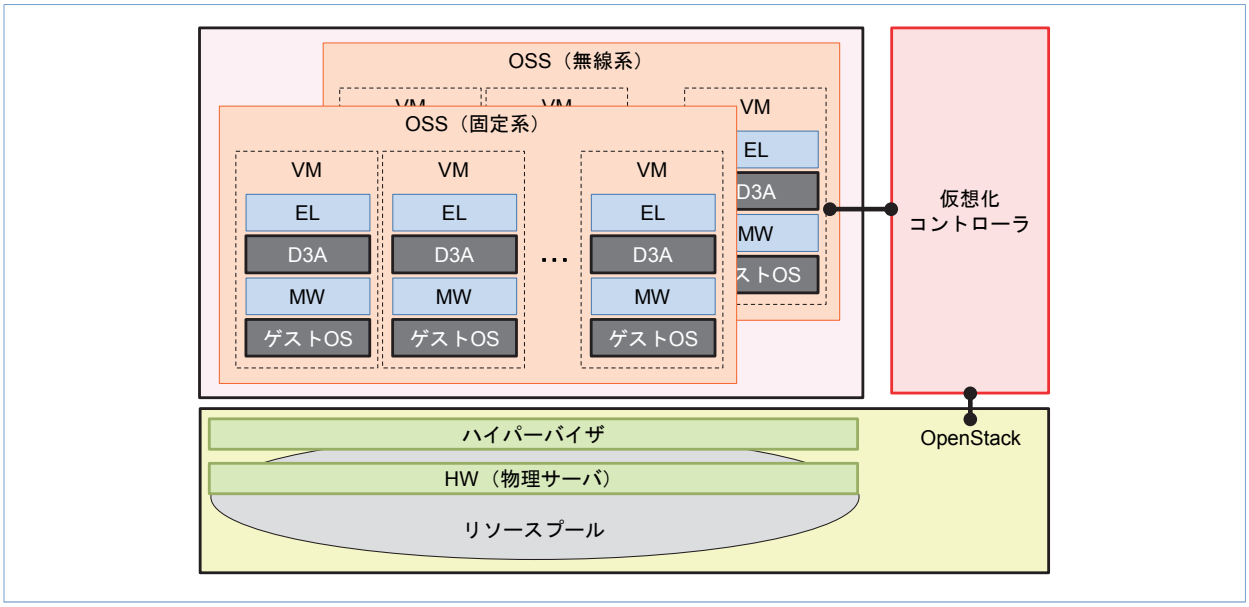

図6　さらなる効率化を実現するアーキテクチャ

D3A基盤SWとSW（EL）の組合せで構成されており，SWの起動や停止，系状態の管理はD3A基盤が担っている．そのため仮想化コントローラは各SWと連携する必要は無く，D3A基盤と連携するだけでSW・VMのライフサイクル管理が実現できる．結果，既存SWへの影響が局所化されるため，本アーキテクチャは開発コストの削減に寄与するものとなった．

加えて，各々の機能は人手を介さず完全に自動化する事も可能である．運用業務の自動化により人為ミスの削減に繋がるが，一方で自動化に伴う追加の開発コストも必要となる．完全自動化の実現コストとそれによるOPEXの削減効果を見極め，今回はすべてワンタッチで実現する方式とした．なおこれらは基本的にETSI（European Telecommunications Standards Institute）[*24] NFV（Network Functions Virtualisation）[*25]標準［6］に則る形で構成しており，仮想化コントローラはNFV標準におけるMANO（Management and Orchestration）[*26]に含まれるVNF Manager（Virtual Network Function Manager）[*27]に相当する．

7. クラウド導入効果と今後の課題

OpenStackプライベートクラウドと仮想化技術導入に伴うTCO（Total Cost of Ownership）[*28]削減の算出結果を**図7**に示す．本図は現状のままの物理環境で運用した場合（図7(a)）とOpenStackプライベートクラウドと仮想化技術導入の場合（図7(b)）のTCOを7年間の運用費用を含め比較した結果である．

CAPEX面では設備投資の削減が大きい．更改後HWの性能向上と仮想化による効率的なリソース利用によるHW設備投資の最適化がTCO削減の大部分を占めている．

OPEX面の効果は，先のHW設備投資の最適化に伴う保守費用や電力費用の削減が挙げられる．また仮想化技術と表3に示す機能を組み合わせることで

＊24 ETSI：欧州電気通信標準化機構．ヨーロッパの標準化団体．電気通信技術に関する標準化を行っている．本部はフランスのSophia Antipolisにある．
＊25 NFV：仮想化技術を用いて通信機能処理を実現するSW処理を汎用製品上で実現する技術．
＊26 MANO：仮想化環境におけるHWリソース，SWリソースなどのVNF管理機能とオーケストレーション機能を提供する仕組み．
＊27 VNF Manager：VNFのライフサイクル制御として起動や停止などVNFの制御を担うシステム．
＊28 TCO：システム運用にあたって必要な初期導入費用と運用管理費用の総額．

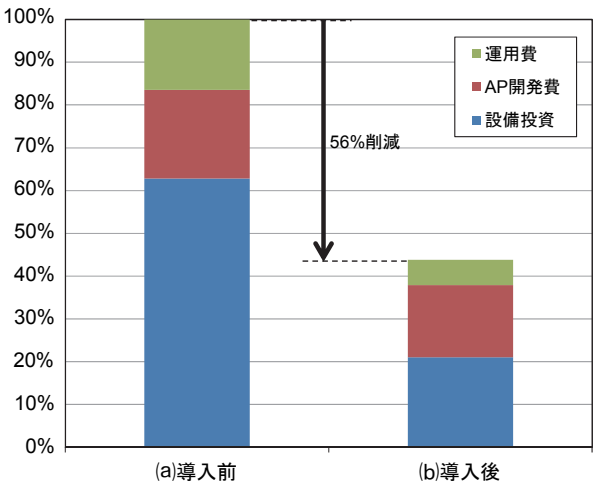

図7　クラウドと仮想化技術導入に伴うTCO削減効果

保守対応の計画的な実施が可能となるが，さらなる効率化をめざして，効果を最大化する運用方式の検討が今後の課題である．

8. あとがき

本稿では，従来のOSSの設備更改に伴いOpenStackプライベートクラウドと仮想化技術の導入について検討し，TCO削減や余剰リソースの有効活用が可能であることを示した．大規模OSSをクラウド環境へ移行する事はドコモにとって初の試みであった．クラウドネイティブ*29でないOSS SWをクラウド環境に移行し，所望の性能や冗長性を発揮させるためのさまざまな工夫をした結果，商用導入を実現した．仮想化技術の導入により，新たな運用プロセスが必要となるが，効率的かつ俊敏・柔軟な運用が実現できるメリットは非常に大きい．クラウドと仮想化技術の導入効果を最大限に引き出し，さらなる業務効率化に向けた検討を今後も継続する．

文　献

[1] 高橋 和秀：“ディザスタリカバリを経済的に実現するOSS構成技術，”電子情報通信学会技術研究報告，NS，ネットワークシステム　112（392），pp.61-66，Jan. 2013.
[2] 高橋 和秀，大貫 雅史：“大規模分散オペレーションサポートシステムの実用化，”電子情報通信学会論文誌．B，通信　J92-B（7），pp.970-980，Jul. 2009.
[3] 北濱，ほか：“大規模災害時におけるIP系OSSの信頼性向上，”本誌，Vol.24，No.2，pp.69-73，Jul. 2016.
[4] 加藤 泰文，田向 正一：“LXCで学ぶコンテナ入門－軽量仮想化環境を実現する技術，”Gihyo.jp ADMINISTRATOR STAGE　連載，技術評論社，May-Jun. 2014.
http://gihyo.jp/admin/serial/01/linux_containers
[5] OpenStack Administrator Guide: “Filter Scheduler.”
http://docs.openstack.org/developer/nova/filter_scheduler.html
[6] ETSI: “ETSI Network Functions Virtualization completes first phase of work,” Jan. 2015.
http://www.etsi.org/news-events/news/864-2015-01-press-etsi-network-functions-virtualisation-completes-first-phase-of-work

*29 クラウドネイティブ：オンプレミスではなく，クラウド上での構築運用を前提として設計されたシステムやサービスを指す．