

# 大規模高信頼データセンター向け VM作成自動化方式の提案と事例 による評価

金子 聡<sup>†1</sup> 中島 淳<sup>†1</sup> 坂下 幸徳<sup>†1</sup> 敷田 幹文<sup>†2</sup>

<sup>†1</sup>(株)日立製作所 <sup>†2</sup>北陸先端科学技術大学院大学

データセンター（以下、DCと略す）内のサーバやストレージといったリソースの管理体系は、DCの規模により異なる。北米の銀行や証券に代表されるエンタプライズ系の企業が有する大規模DCでは、大量のデータを高信頼に扱うことが求められるため、管理者にも専門知識が必要となり、リソース種別ごとに管理を水平分業することで、管理を効率化していた。そこで、ストレージ管理者は、専門知識に基づいて、エンドユーザの組織間での性能干渉抑止、および、ストレージ内部のリソースの稼働率平準化、といった運用を行い、高信頼なストレージリソースを提供していた。このような水平分業環境においても、ビジネスに求められるスピードが速まり、仮想マシン（以下、VMと略す）の迅速な提供がエンドユーザから求められ、サーバからストレージまでを垂直管理するニーズが高まっている。しかし、従来の垂直管理では、前述の水平分業によるストレージ運用の質を維持することができず、結果として、サーバとストレージの管理者間で複雑な調整が必要となっていた。そこで本稿では、ストレージの複雑な知識を持たないサーバ管理者が、ストレージ管理者との調整なしに、従来通りの運用を実現しつつもVM作成を自動化する方式を提案する。提案方式は、サーバ管理者のリソース利用ポリシーに基づいて利用リソースの制約を定義し、その定義に基づいてストレージ設定パラメータを自動決定することにより、従来通りのストレージ設定を自動化する。提案方式を大規模DCの運用方針の実事例に照らし合わせて評価した結果、今後の大規模DC運用において有用であることを確認した。

## 1. はじめに

急速なデータ量の増加に伴い、データセンター（DC）の大規模化が進んでおり、250台以上のサーバラックを有する大規模DCの数は、北米、欧州において2016年には6,500サイト以上にもなると予測される[1]。また、近年DC内では、サーバ仮想化技術の利用が急増している[2]。これにより、ITインフラの管理者にとっては、従来に比べて管理レイヤとして仮想マシン（VM）が増え、VMからストレージといったEnd to Endの構成管理は複雑さを増していた。このように大規模化、複雑化したDCでは、ITインフラの運用管理が困難化しており、管理コストの増大が問題となっていた。特に、北米の銀行や証券などに代表される比較的規模の大きな企業の自社向け大規模DCでは、サーバやストレージ、ネットワークといった管理レイヤごとに管理者を分け、水平分業することで、各管理者に特定の専門知識の習得に専念させ、管理の効率化を図っていた。ストレージに関しては、高いI/O性能を安定して提供するために、高性能かつ高信頼なストレージの利用率が高く、さらに、ストレージ管理者は、ストレージ内の物理ディスク等の内部リソース

をサーバ間で共有させ、利用効率を高めつつ、異なる組織のサーバ間では性能干渉が起きないように、利用リソースを分離する、という高信頼なストレージ運用を行っていた。以降では、前述の、規模の大きな企業をエンタプライズ系の企業、高性能かつ高信頼なストレージをエンタプライズストレージ、と呼ぶ。

一方、ビジネスに求められるスピードの向上に伴い、IaaS（Infrastructure as a Service）のようなクラウドサービスが登場しており、VMの迅速な提供が、大規模DCにおいても求められている。IaaS環境では、同じ管理者がサーバ、ネットワーク、ストレージを垂直管理することでVMの迅速な提供が可能となる。しかし、同じ管理者が複数の管理レイヤのスキルを習得することは困難なため、IaaS管理者がほかのサーバや内部リソースを考慮せずにストレージを利用することで、性能干渉が生じる可能性があった。そのため、前述の水平分業管理によるストレージ運用を維持するために、サーバとストレージの管理者間で複雑な調整に基づいてVM作成を行う必要があった。以上に示すように、大規模DCでは水平分業によるストレージ運用を維持しつつVM作成を自動化することが課題であった。

そこで、本稿ではエンタプライズ系の企業が有する大規模DCにおいて従来通りの運用を維持しつつVM作成時間を短縮するべく、ユーザ側のリソース利用ポリシーに基づいて利用リソースについての制約を定義し、定義された範囲内で従来ストレージ管理者が手動で選出していたリソースと同等の条件のリソースを選出するVM作成自動化方式を提案する。さらに方式の有用性を実際の大規模DCの運用事例により評価する。

以下、第2章では従来の運用管理技術と課題について述べ、第3章では提案方式について述べる。第4章では提案方式の評価を行い、第5章では提案方式の有効性について述べる。

## 2. 従来技術と課題

本章では研究対象とするDCの管理体系およびVM作成に関する従来研究と課題を示す。

### 2.1 従来の運用管理者の体系

サーバ数十台といった小規模なDCでは図1の左側に示す通り、ITインフラを構成するサーバ、ネットワーク、ストレージといった異種機器の管理レイヤを同じ管理者が集中管理していた。しかし、データ量の増加に伴う機器数の増大、また、機器の高機能化に伴う管理の複雑化によって、集中管理が難しくなっている。この状況に対し、管理レイヤごとに管理者を分け、水平分業することで、管理者の学習コストを下げ、管理の効率化を図っていた(図1中央)。この傾向はDCの規模が大きいほど顕著であり、大量のデータを高信頼に扱うことが求められるエンタプライズ系の企業の自社向け大規模DCは水平分業の典型である。このようなDCでは、サーバ数千台、

エンタプライズストレージ[3]数十台といった規模のリソースを管理する必要があり、管理レイヤごとの専任管理者がそれぞれ数十名の体制で運用している事例も登場している。

エンタプライズ系の企業では、VMを管理するサーバ管理者は、VMで業務を運用する組織ごとが存在する。この組織とは企業内で目的を同じくするユーザの集合であり、DCによっては、個々の部署や企業の場合がある。よって、組織間で性能干渉が発生してはいけない。DCが大規模なほど、業務の数も多くなり、組織横断でストレージを共有する傾向が強くなる。このため、エンタプライズストレージは、高いスケラビリティを特徴としており、大量のサーバで共有することで、リソースの利用効率向上が可能である。このように、エンタプライズ系の企業はサーバ間でストレージのリソースを共有し、利用率を高めたいものの、組織間の性能干渉は避けたいため、従来ストレージ管理者はサーバ管理者と利用リソースの対応関係を把握し、組織ごとに性能干渉が起きない構成を構築してきた。

### 2.2 サーバ管理とストレージ管理の連携の必要性

ビジネスに求められるスピードの向上に伴い、水平分業管理下においても、共有リソースであるサーバ、ネットワーク、ストレージの一部を垂直管理(図1右側)する管理形態のニーズが高まっている。本稿では、この管理形態をハイブリッド型の管理、と呼ぶ。図1では、垂直管理の代表例であるIaaSの管理者を垂直管理者として示しており、IaaS管理者は管理範囲ごとに分業を行う。しかし、垂直管理において管理者が複数の管理レイヤのスキルを習得することは困難であった。そこで、垂直管理の支援を目的として、VM作成を自動化する技術が登場している。

OpenStack<sup>☆1</sup>はCinder[4]と呼ばれるストレージAPIによって、VMにストレージのボリュームと呼ばれる記憶領域を直接割り当てる機能を実現しているが、APIの引数はボリュームの名前と容量だけであり、ストレージの内部リソースの選択方法は規定されていない。VMware<sup>☆2</sup>のVMware vRealize Orchestrator[5]は複

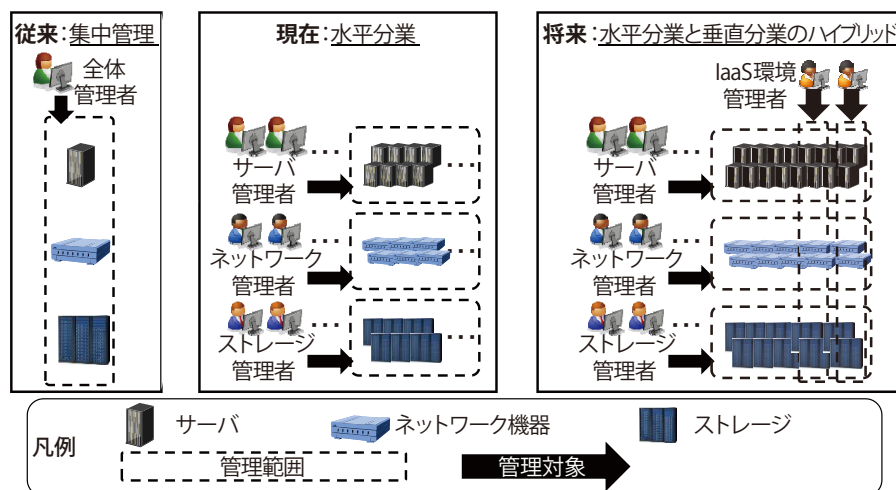


図1 DCにおける管理体系の遷移

☆1 OpenStackはOpenStack Foundationの登録商標である。

☆2 VMwareはVMware, Inc.の商標または登録商標である。

数の API を組み合わせて、処理を事前定義し、VM 作成時の管理者間の作業依頼を不要化している。しかし、ストレージの利用状況に合わせて設定したい場合は、処理の定義が毎回必要となる。Akhani らは IaaS 環境における計算機リソースの割当サービスにおいて、エンドユーザとプロバイダ間のやり取りの数がサービス提供におけるオーバヘッドになることを課題としている [6]。このプロバイダ業務は垂直管理の一例であり、Akhani らは提供するリソースの予約状態に着目し、リクエストのスケジューリングによる課題解決について議論している。しかし、ストレージの性能は考慮していない。また、Gopisetty らは、ストレージ内部のプロセッサなどの内部リソースが高負荷になり、性能ボトルネックとなることを避けるために、負荷を平準化させる方法を提案している [7]。これによれば、リソースの負荷平準化が可能だが、システム全体に適用すると、やはり負荷の状態によっては組織間で性能干渉が発生してしまう。

以上に示すように、従来方式では、組織間の性能干渉が起こり得たため、垂直管理により VM 作成の自動化はできるものの、水平分業管理下で行っていた従来のストレージ運用を維持できず、ストレージ管理者は手でストレージの割当を行っていた。ストレージ管理者はサーバ管理者、または、ボリューム上で動作するアプリケーションの管理者であるユーザからストレージ上のサービスの稼働状況をヒヤリングし、さらに、組織ごとのストレージに対する予算を調査し、それらの額に応じて、組織ごとに優先度付けを行い、リソースを調整していた。

その結果として、水平分業管理下で行っていた従来のストレージ運用を維持するために、サーバ管理者とストレージ管理者が、互いに作業依頼を実施することで、各レイヤの管理をそれぞれ実施せざるを得なかった [8]。

図 2 に従来の煩雑な VM 作成手順を示す。なお、本稿では、サーバ管理とストレージ管理における運用管理コスト削減に着目するため、ネットワーク管理者の VLAN 等の設定は実施済みとし、VM 作成には含まない。

図 2 に示すように、従来の VM 作成では、ユーザ側であるサーバ管理部門とストレージ管理部門との間の作業依頼に伴い複数の承認作業が必要であった。承認作業に要する時間は、責任者や担当者の状況に応じて変わり、VM 作成に時間を

要する主要因となっていた。

以上より、サーバ管理者向けにストレージの専門知識が不要、かつ水平分業管理による高信頼なストレージ運用を維持可能なボリューム作成自動化方法が必要となる。

### 2.3 ストレージ管理者による従来のボリューム割当作業

エンタプライズストレージを複数台、および、100 台以上のサーバを有するような大規模 DC において、ストレージ管理者がボリューム割当時に実施していた作業の洗い出しを行った結果、作業は以下の 4 つであった。

作業 a: 要求元のサーバ管理者のストレージの利用に関する予算や、ストレージ上で稼働しているサービスの重要度に基づいて、要求元サーバ管理者が利用可能なリソースを決定

作業 b: 共有により性能干渉を発生し得るリソースを選出する際に、要求元と異なる組織が利用中のリソースを、選出対象から除外

作業 c: サーバ管理者の要求を満たすリソースを特定。要求の一例として、ボリュームの容量がある

作業 d: ストレージ内部のリソースの高負荷によりボリュームの性能が劣化することを避けるべく、負荷状態に基づいて、負荷を平準化するリソースを特定

これらのリソース選出条件はすべて同時に満たす必要がある。つまり、作業 d について、単にストレージ内の全リソースで平準化するのではなく、作業 a, b の観点も踏まえ、ユーザ組織間で性能干渉が出ないように平準化する必要があった。以上に示すストレージ管理者による運用を維持することが求められていた。

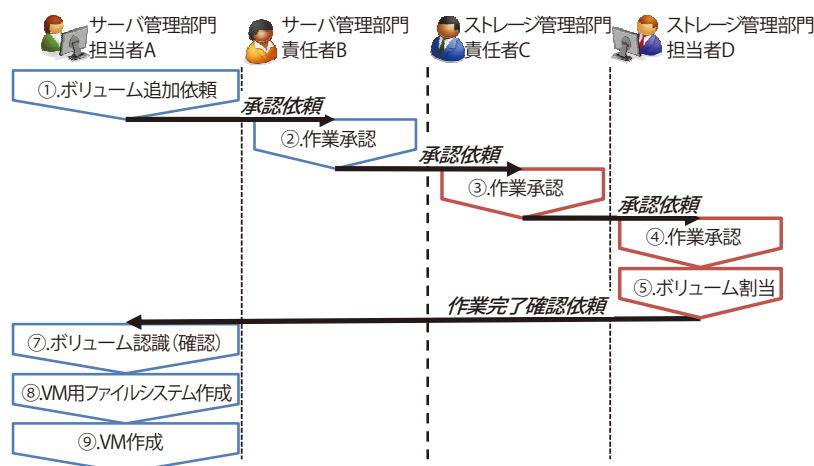


図 2 従来の煩雑な VM 作成手順



### 3. リソース利用ポリシーに基づくVM 作成自動化方式

本章では前章で述べた課題の解決に向け、ユーザ側のリソース利用ポリシーに基づく利用リソースの制約定義と、ストレージ設定のパラメータ自動決定により、従来通りの運用を維持可能なVM作成自動化方式を提案する。

#### 3.1 課題解決方針

提案方式は、予算配分の更新やユーザの利用組織の変更の頻度がVM作成頻度に比べてきわめて低いことに着目し、これらのユーザ側のリソース利用ポリシーに基づいて利用リソースの制約を事前に定義し、当該制約を満たしつつ、ストレージの利用状況に基づいて、リソースを自動選出する。

ユーザ側のリソース利用ポリシーとは、2.3節の作業a, bのストレージの利用に関するユーザ組織の予算や、ストレージ上のサービスの重要度、ユーザの組織、についての情報を示し、当該情報に基づいて、利用可能であると定義されたリソースを利用することで、作業a, bの条件を達成する。

また、作業dのリソース選出条件はストレージの利用状況を示す構成情報に基づいており、作業cについても、ユーザの入力はデータ格納場所に対する要件になるため、リソース選出条件は基本的にストレージの構成情報に基づくと考えられる。よって、ストレージの構成情報を用いることで、作業c, dの条件を満たすリソースを自動選出する。

#### 3.2 処理方式

提案方式の概要を図3に示す。提案方式におけるVM作成は2つの操作で構成される。1つ目はストレージ管理者がユーザに対し利用リソースの制約を定義する操作である。これは、VM作成の前に、事前設定として行われる。ストレージ管理者が、ユーザ側のリソース利用ポリシーとすべてのストレージの構成情報に基づいて、各ユーザの組織ごとに利用可能なリソースを、利用リソース制約定義モジュールを介して、リソース制約情報として定義する。

2つ目は、VM作成に際し、サーバ管理者がボリュームに対する要件を入力し、ボリュームを自動割当する操作であ

る。図3のパラメータ自動決定モジュールが、前述のリソース制約情報と、ストレージの現在の利用状況を示す構成情報に基づいて、リソース群の中から、入力された要件を満たすリソースを選出し、そのリソースを用いてボリューム割当処理を実施する。この処理は、2.3節の作業b, c, dと同等のリソース選出を行う。作業b, c, dのリソース選出方法は、それぞれ、ストレージの仕様とユーザの運用によって詳細が異なる。たとえば、作業bについては、任意のサーバに対して占有割当が可能なリソースは、ストレージの仕様によって異なる。しかし、ストレージの仕様に基づいて作業b, c, dのリソース選出処理を実装することで、提案方式を適用可能である。提案方式は図3の白抜きの部分であり、サーバからストレージのボリュームまでの接続情報などを含む構成情報を収集する処理は、既存のストレージ管理ソフトウェアの機能[9], [10]を利用可能である。

次節以降では、利用リソース制約定義方式とストレージ設定パラメータ自動決定の詳細と設計例について述べる。

#### 3.3 利用リソース制約定義方式

利用リソース制約定義では、ユーザ側のリソース利用ポリシーに基づいて、ユーザが利用可能なリソースについての制約を定義する。利用リソースの制約の定義を示すデータモデルを図4に示す。リソースの集合（リソースグループ）と、管理者（ユーザ）の集合（ユーザグループ）とを定義し、両者を管理権限（ロール）で対応付け、利用可能なリソースを定義する。

大規模DCでは、複数人のサーバ管理者が任意の物理サーバ群を共同で管理する環境が多いため、ユーザをグループで管理することで、この環境を効率的に管理できる。なお、ユーザグループとリソースグループは多対多の関係であり、任意の組織間でリソースを共有する運用

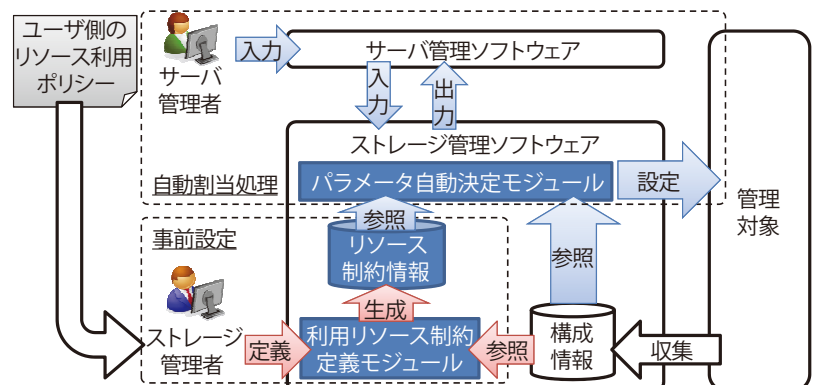


図3 提案方式の概要







表 1 運用事例の調査結果

リソース選出条件	事例の運用方針	適用可能性
a. 要求元サーバ管理者が利用可能なリソースを決定	性能要件が厳しいため、利用するリソースを分ける	◎
	性能要件が低いいため、利用するリソースを分けない	×
b. リソース共有による組織間の性能干渉抑止	性能要件が厳しいため、性能干渉抑止を行う	◎
	性能要件が低いいため、性能干渉抑止を行わない	×
c. サーバ管理者の要求を満たすリソースを特定	一度のリクエストで処理する	◎
	リクエストを繰り返し、現実的に可能な構成を決定	×
d. リソースの稼働率を平準化	ストレージポートにマッピングするサーバ数を平準化（半数以上）	◎
	ストレージ内部の CPU リソースの処理を平準化（少数）	○
	平準化しない（半数程度）	○
(該当条件なし)	ストレージポートに複数サーバをマッピング（少数）	○

とに利用可能なリソースを分ける運用が多く見られた。近年サーバ仮想化の登場により、エンドユーザあたりの VM 数は増加しており、前述の運用が多い理由となっている。性能要件が低いワークロードでは性能干渉を意識しないためリソースを分けませんが、エンタプライズ系の企業ではミッションクリティカルなワークロードが多いため、大半の事例において提案方式が適用可能であった。

作業 b に関しては、上述の通り、ミッションクリティカルなワークロードが多いエンタプライズ系の企業では、性能干渉抑止を行う傾向が強いため、大半の事例において提案方式が適用可能であった。

続いて、作業 c に関しては、ほとんどの事例にて、ボリューム割当のリクエストは提案方式と同様に、サーバ管理者が分かる用語で構成された定型フォームが利用されていた。しかし、リクエスト後のフローに違いがあり、リクエストを満たすリソースが存在しない場合は、リクエスト元がリクエスト内容を見直して再リクエストを繰り返し、コミュニケーションを行うことで現実的に可能な構成を実現していた。提案方式は、リクエスト内容を満足するリソースが存在しない場合はエラーとなるため、このケースに対応できなかった。

作業 d に関しては、半数以上の事例でストレージポートにマッピングするホストの数を平準化しており、提案方式を適用可能であった。また、少数の事例では、性能要件が厳しい顧客がストレージ内の CPU の負荷を平準化するケースが確認され、今回試作したシステムは CPU 負荷情報を収集する機能を有していないため、試作システムを適用できなかった。逆に半数程度の事例で

は、性能要件が低いいため、性能問題が発生した際に利用するリソースの割当を変更する、といった事後対応をとっており、割当時の平準化は行わず、試作システムの実装とは異なる運用が行われていた。

一方で、仮想化サーバの特徴機能である VM マイグレーション機能を利用するために、ストレージのポートに複数のサーバをマッピングさせる運用が確認され、このケースでは試作システムを適用できなかった。

## 4.2 VM 作成時間削減効果の評価

前節で述べた試作システムが適用可能であったケースについて、VM 作成時間削減効果を評価した結果を述べる。

### 4.2.1 システム構成

提案方式（図 3）の評価には 3.5 節で述べた試作システムを使用した。試作システムでは、サーバ仮想化製品に VMware ESX Server[14] を選定し、VM 用ファイルシステム作成を行う GUI を VMware vCenter Server 内にプラグインとして実装した。ストレージ管理ソフトウェア部は新規に開発した。開発環境には GUI に ActionScript 3.0、GUI 以外に Java 5<sup>☆4</sup> を採用した。また、従来システムの評価では、ストレージ管理ソフトウェア部分は新規に開発し、それ以外は試作システムと同様とした。

### 4.2.2 VM 作成業務の時間削減効果試算

提案方式では、ストレージ管理者が利用リソース制約を事前に定義する操作と、サーバ管理者が VM を作成する操作で構成される。前者は、リソースの利用に関する予算が新規に決定／変更される場合と、サーバ管理者が新規に追加された場合に実施され、VM 作成ごとに実施の必要はない。後者の VM 作成操作手順を図 6 に示す。

VM 作成操作は 2 ステップである。VM 用ファイルシステム作成では、従来作業依頼を交えて実施していたボリューム作成／割当／認識、VM 用ファイルシステム作成を一括で実行する。VM 作成は、従来と同様のステップである。

ここで VM 作成所要時間を定式化する。作業依頼で発生する社内ワークフロー等の所要時間を  $A_t$ 、VM 作成回数を  $n$ 、サーバおよびストレージの各種設定の処理時間を  $C_t$  とした場合、従来システム（図 2）の所要時間  $T_{conv}$  は次式に表すことができる。

$$T_{conv} = n(A_t + C_t) \quad (1)$$

続いて、利用可能リソース定義処理時間を  $D_t$ 、利用

☆4 Java は Oracle Corporation の商標または登録商標である。

リソース制約の定義の回数を  $m$  とした場合、試作システム (図6) の所要時間  $T_{prot}$  は次式に表すことができる。

$$T_{prot} = nCt + mDt \quad (2)$$

数式が示す通り、従来システムではVM作成の所要時間が、 $A_t$ に比例するのに対し、提案システムでは $A_t$ に依存しない。続いて、従来システム、および、試作システムを用いて、サーバ・ストレージの設定処理時間 $C_t$ 、利用可能リソース定義処理時間 $D_t$ を測定した結果を表2に示す。

表2は、作成するボリュームおよびファイルシステムのサイズは100GBとして3回測定した平均値を示す。ボリューム作成/割当/認識、VM用ファイルシステム作成の時間の総和が数式(1)の $C_t$ に相当し、VM用ストレージ利用ファイルシステム作成の時間が数式(2)の $C_t$ に相当する。また、 $A_t$ は担当者間の調整に要する時間であり、担当者の状況や企業のITシステム部門の運用ポリシーによって大きく変わるため、測定対象としなかった。 $A_t$ は数分から一日以上を要するケースも考えられ、一例として、利用リソース制約の定義は一度だけ( $m=1$ )、担当者間の調整( $A_t$ )に60分かかった場合では、1VMを3回作成する場合( $n=3$ )に従来システムでは合計で190分もかかっていたのに対し、試作システムでは、管理者間の調整が不要なため、7分で済む。

なお、実際には、従来システムでは、 $A_t$ と $C_t$ 以外にも、各リソースの利用率を確認し、負荷を平準化させるパラメータを決定するために試行錯誤する時間も含まれるが、サーバ間の性能干渉を考慮しなければ、従来技術であっても、前記試行錯誤の時間は不要化されるため、評価の対象とはしなかった。



図6 試作システムのVM作成操作手順

表2 各ステップの処理時間

処理名	変数名	時間 (秒)
ボリューム作成	Ct	57
ボリューム割当		17
ボリューム認識		61
VM用ファイルシステム作成		61
利用可能リソース定義	Dt	30
VM用ストレージ利用ファイルシステム作成	Ct	138

## 5. 考察

本章では、大規模DCの運用方針への適用可能範囲と、管理者間の連携の迅速性について述べる。

### 5.1 大規模DCの運用方針への適用可能範囲

4.1節で提案方式および試作システムが適用できなかったケースの重要性について考察する。

4.1節で述べたように、提案方式および試作システムが対応できない事例が存在した。まず、2.3節の作業cに関して、十分なリソースが存在しない場合、リクエスト元はリクエストを繰り返し、コミュニケーションを行うことで現実的に可能な構成を柔軟に実現していたが、リクエストの繰り返しは、4.2節で示した、従来システムではVMを3回作成する時にかかる190分、という作業時間を $n$ 倍化させてしまい、VM作成時間を長大化させてしまう。提案方式は、一度に利用するリソースを決定する方式のため、このケースに対応できないが、コミュニケーションの繰り返しを省くことで、大幅に作業時間を削減できる。また、前述したOpenStack[4]だけでなく、サーバ仮想化環境においてOpenStackと同様に利用率の高いVMwareもサーバ管理ソフトから直接ストレージを管理する機能を提供しており[15]、サーバ管理者によるストレージに対するリクエストは、コミュニケーションの効率化のために、今後ソフトウェアにて実現されていくと予想される。その環境では、リソースの利用状況に基づいて、要求されたリソースが利用可能な場合だけリクエストを受け付けることが可能であり、リクエストは失敗せず、繰り返しが不要となるため、提案方式を適用可能である。よって、今後は、提案方式が適用可能なケースが増加すると予想される。

ボリューム上で動作する業務の性能要件がミッションクリティカルほど厳しくない環境では、リソースの利用率の平準化を行っておらず、試作システムを適用できなかった。ストレージの性能を安定させることの重要度は、ストレージ管理者が属するITシステム部門の運用ポリシーや容量提供サービスの定義などに依存して大きく変わる。重要度が高い企業は、ボリューム作成時に、性能問題が発生しにくい構成を設計していたが、重要度が低い企業は、性能問題が発生してから、または、ユーザから性能問題の対策要求が来てはじめて、性能問題への対処を行うため、エンドユーザに対するサービスレベルを低下させていた。この問題は、データ量の増大に伴い、今後ますますDCが大規模化することで、さらに顕在化



してくると考えられる。このような環境に提案方式を適用することで、性能問題が発生しにくい構成を自動構築することが可能となり、ストレージ管理者はサーバ管理者から依頼される性能問題への対処といった作業から解放され、かつ、サービスレベルを向上させることが可能となる。

また、ストレージ内のCPUの利用率を平準化させる運用が確認された。試作システムでは、過去にCPUがボトルネックになる事例が少なかったため、この運用に対応していなかったが、ポートと同様に、CPUの稼働率に関する情報を利用することで、適用可能である。

続いて、ストレージのポートに複数のサーバをマッピングする運用(表1の最下行)は、仮想化サーバのVMマイグレーション機能を頻繁に利用するケースであり、同じポートにマッピングされているサーバ間でVMのマイグレーションが可能となる。このケースでは、サーバの負荷によってVMが移動し、移動に伴いポートへの負荷も変動するため、ポートの利用率に余裕を持って運用していると考えられる。この場合、性能劣化するほどポートの負荷が上がらないため、性能干渉が発生しにくい。よって、本ケースに対応する必要性は低いと考えられる。

以上より、対応できていなかった運用事例について、将来の大規模DCにおいて対応できないケースが減少していき、さらに、対応する必要性が低いと考えられるため、提案方式はエンタプライズ系の企業の大規模DC向けに有用であると考えられる。3.5節で述べた提案方式の試作システムは製品化済みであり、北米を中心とする大規模DCで利用されていることから、実環境での有用性を確認できる。

## 5.2 管理者間の連携の迅速性

本稿では、提案方式の効果として、4.2節で例示した、VMを3回作成する時にかかる190分、という管理者間の調整に伴う時間( $A_t$ )を評価したが、提案方式は水平分業による高信頼なストレージ運用を維持しつつ、従来技術[8]によるリソース負荷平準化を可能とするため、実際の運用では、4.2節の最後に述べた、ストレージのリソースの利用率を確認し、負荷を平準化させるパラメータを決定するために試行錯誤する時間、も削減される。この時間は、リソースの量や、平準化の方法に依存して変わるため、一概には定義できないが、DCの規模が大きいくほど、確認対象のリソースの数が増え、削減可能な時間が増大する。

また、従来大規模DCでストレージ管理のノウハウを持たないIaaS管理者がほかのサーバや内部リソースを考

慮せずにストレージを利用してしまふことで、性能干渉が生じ、IaaS管理者とストレージ管理者の間でコミュニケーションを行い、性能問題の対処にあたっていた。この対処には、まず、従来のVM作成と同様に、4.2節で述べた調整に伴う時間( $A_t$ )と、前段落で述べたリソース負荷平準化のための試行錯誤時間が必要となる。さらには、5.1節で述べた、負荷平準化を行っていないケースと同様に性能問題の原因を特定するための時間が削減される。

一方、VM作成時に利用可能リソースが枯渇している場合、ストレージ管理者によって、利用可能なリソースを増量してもらう必要が生じる。容量が足りなくなるケースは運用開始時に策定した計画以上に容量を消費してしまったケースであり、大規模な容量提供サービスを行っている大規模DC環境ではほとんど発生しない。

以上より、ストレージ管理者はサーバ管理者との膨大なやり取りから解放され、コスト削減のための構成の見直し等の高度な管理作業へのシフトが可能になると考えられる。また、管理作業の大幅な削減に伴って、ストレージ管理者の件数を2.2節に示したIaaS管理者用へ活用可能となり、エンタプライズ系企業の大規模DCの運用管理体系をハイブリッド型へ移行させる際にも有用であると考えられる。

## 6. おわりに

本稿では大規模DCにおいて従来の水平分業管理におけるストレージ運用の質を維持しつつVM作成時間を短縮するべく、ユーザ側のリソース利用ポリシーに基づいて利用リソースについての制約を定義し、定義された範囲内で従来ストレージ管理者が手動で選出していたリソースと同等の条件のリソースを選出するVM作成自動化方式を提案した。これにより、サーバ管理者とストレージ管理者がコミュニケーションによる調整作業を行うことなく、ストレージ管理者が従来行っていた運用方針でストレージリソースを自動決定することが可能である。

DCの大規模化に伴い、サーバやストレージ等の複数の管理レイヤの集中管理が難しくなり、大規模DCでは管理レイヤごとに数十名の管理者を配備し、水平分業することにより、管理を効率化していた。特に、大量のデータを高信頼に扱うことが求められる、北米の銀行や証券に代表されるエンタプライズ系の企業の大規模DCのストレージ管理者は、専門知識に基づいて、エンドユーザの組織間での性能干渉抑止、および、ストレージ内部のリソースの稼働率平準化、といった運用を行い、高信頼なストレージリソースを提供していた。このような水

平分業環境においても、ビジネスに求められるスピードが速まり、VMの迅速な提供がエンドユーザに求められ、サーバからストレージまでを垂直管理するニーズが高まっている。しかし、従来の垂直管理では、前述の水平分業によるストレージ運用の質を維持することができず、結果として、サーバとストレージの管理者間で複雑な調整が必要となっていた。この課題を解決するべく、サーバ管理者のリソース利用ポリシーに基づく利用リソース制約定義方式と、定義された範囲内で従来と同等の条件のリソースを選出するVM作成自動化方式を設計した。

提案方式について、実際の運用事例に基づいて大規模DCへの適用可能範囲を評価し、エンタプライズ系の企業の大規模DCの大半の運用および環境に適用可能であることを確認した。

従来水平分業の大規模DCにおいてサーバからストレージまでの垂直管理を行う際に、DC内のリソースを利用する個々の部署や企業といった各組織間で、リソース共有による性能干渉が発生することが懸案であったが、提案方式により、組織間の性能干渉を避けることができる。また、提案方式は商用の運用管理ソフトウェアとして製品化済みであり、これを用いることで、エンタプライズ系の企業においてもVM作成自動化が実際に行われていることを確認している。

以上より、本方式は、エンタプライズ系企業が所有する大規模DCにおいて、ストレージ管理者が従来手動で行っていたストレージ運用の質を維持しつつ、VM作成を自動化することを可能とし、エンタプライズ系の企業が有する大規模DCの運用管理体系を、水平分業型から、水平分業管理下にある共有リソースの一部を垂直管理するハイブリッド型へ移行するために有用な技術であるといえる。

本稿では、運用管理の垂直管理支援におけるサーバ管理とストレージ管理との連携に注目した。今後はDCの大規模化に伴いネットワークの構成も頻繁に変更されるようになるため、ネットワーク管理者とのコミュニケーションも考慮した管理方式が課題である。

参考文献

- 1) Gartner : Forecast : Data Centers, Worldwide, 2010-2016, 4Q12 Update, Gartner, Inc. (2013).
- 2) 宮下夏苗, 上埜元嗣, 宇多 仁, 敷田幹文: 大学におけるプライベートクラウド環境の構築と利用, 第3回インターネットと運用技術シンポジウム, pp.17-24 (2010).
- 3) (株)日立製作所: Hitachi Virtual Storage Platform, <http://www.hitachi.co.jp/products/it/storage-solutions/products/vsp/> (2015年3月31日現在).
- 4) OpenStack : OpenStack Cinder, <https://wiki.openstack.org/wiki/Cinder> (2015年3月31日現在).

- 5) VMware Inc. : VMware vCenter Orchestrator, <http://www.vmware.com/jp/products/vcenter-orchestrator> (2015年3月31日現在).
- 6) Akhiani, J., Chuadhary, S., and Somani, G. : Negotiation for Resource Allocation in IaaS Cloud, COMPUTE'11 Proc. the Fourth Annual ACM Bangalore Confer-ence, pp.1-15, ACM (2011).
- 7) Gopisetty, S., Butler, E., Jaquet, S., Korupolu, M., Nayak, T. K., Routray, R., Seaman, M., Singh, A., Tan, C. H., Uttamchandani, S. and Verma, A. : Automated Planners for Storage Provisioning and Disaster Recovery, IBM J. Res. Dev.,v.52 n.4, p.353-365 (2008).
- 8) (株)日立製作所: 統合プラットフォーム, <http://www.hitachi.co.jp/products/it/unified/products/model/iaas.html#features> (2015年3月31日現在)
- 9) 坂下幸徳, 河野泰隆, 柴山 司, 中島 淳, 敷田幹文: 大規模データセンタにおけるシステム構成情報の高速収集方式の提案, 情報処理学会論文誌, Vol.53, No.3, pp.969-977 (2012).
- 10) 金子 聡, 坂下幸徳: 異種仮想サーバ混在環境向け構成情報の統一管理方式の提案, 情報処理学会研究報告, Vol.2012-IOT-16, No.44, pp.1-6 (2012).
- 11) VMware Inc. : VMware vCenter Server, <http://www.vmware.com/products/vcenter-server/> (2015年3月31日現在).
- 12) Microsoft : System Center, <http://www.microsoft.com/ja-jp/server-cloud/products/system-center-2012-r2/> (2015年3月31日現在).
- 13) (株)日立製作所: ストレージ仮想化運用のすすめ, [http://www.hitachi.co.jp/products/it/storage-solutions/products/catalog/pdf/ca-874-hitachi\\_command\\_suite.pdf](http://www.hitachi.co.jp/products/it/storage-solutions/products/catalog/pdf/ca-874-hitachi_command_suite.pdf) (2015年3月31日現在).
- 14) VMware Inc. : VMware ESXi and ESX, <http://www.vmware.com/products/vsphere/esxi-and-esx/> (2015年3月31日現在).
- 15) VMware Inc. : VMware Virtual Volumes, <http://www.vmware.com/jp/products/vsphere/features/virtual-volumes> (2015年6月15日現在).

**金子 聡** (正会員) [satoshi.kaneko.bc@hitachi.com](mailto:satoshi.kaneko.bc@hitachi.com)  
 2008年電気通信大学大学院電気通信学研究科修士課程修了。同年(株)日立製作所入社。現在同社研究開発グループ情報通信イノベーションセンタ研究員。ストレージシステムおよびシステム運用管理の研究に従事。

**中島 淳** (非会員) [jun.nakajima.vu@hitachi.com](mailto:jun.nakajima.vu@hitachi.com)  
 2005年慶應義塾大学大学院理工学研究科修士課程修了。同年(株)日立製作所入社。現在同社研究開発グループ情報通信イノベーションセンタ研究員。ストレージシステムおよびシステム運用管理の研究に従事。

**坂下 幸徳** (正会員) [Yukinori.Sakashita@hal.hitachi.com](mailto:Yukinori.Sakashita@hal.hitachi.com)  
 2003年北陸先端科学技術大学院大学情報科学研究科修了。2015年同大学院博士(情報科学)。2003年(株)日立製作所入社。現在同社研究開発グループ情報通信イノベーションセンタ主任研究員, Hitachi America, Ltd. R&D Lab Manager. ITシステム運用管理に関する研究に従事。SNIA Technical Council, SNIA-J技術委員会副委員長。

**敷田 幹文** (正会員) [shikida.mikifumi@kochi-tech.ac.jp](mailto:shikida.mikifumi@kochi-tech.ac.jp)  
 1995年東京工業大学大学院理工学研究科情報工学専攻博士後期課程修了。博士(工学)。同年北陸先端科学技術大学院大学情報科学センター助手。2012年同大学情報社会基盤研究センター教授。2016年高知工科大学情報学群教授。大規模情報システム, ネットワークサービス, グループウェアに関する研究に従事。ACM, 電子情報通信学会, 日本ソフトウェア科学会各会員。

投稿受付: 2015年4月1日  
 採録決定: 2015年9月24日  
 編集担当: 牧野 司 (東京海上日動火災保険(株))