Vol.2012-IOT-16 No.44
2012/3/16

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

金子 聡[†]　河野 泰隆[†]　坂下 幸徳[†]

近年，IT システムの TCO 削減を目的として，サーバやストレージなどのデータセンタへの集約化が進んでいる．また，サーバ仮想化技術の進展により，利用用途に合わせた様々な仮想サーバの導入が増加している．このためデータセンタ内には，異種仮想サーバ混在環境が登場しており，管理コスト削減のために，これら環境を統一した方法で管理出来る運用管理ソフトウェアが必要となっている．しかし，これまでは，各仮想サーバのデータモデルの違いから各仮想サーバを別々の方法で管理しなければならなかった．そこで本論文では，異種仮想サーバのデータモデルと構成情報収集処理を共通化することで，統一した管理と低開発コスト化を実現する方式を提案する．本方式により，統一した管理を実現することに加え，保持する構成情報のデータ量が従来方式の 2~3%まで削減されていることと，異種の仮想サーバへの対応に要する開発コストが従来方式の 40%以下まで低減されていることを検証した．

A Unified Management Method of Resource Information for Heterogeneous Virtualization Server Environment

SATOSHI KANEKO[†] YASUTAKA KONO[†]
YUKINORI SAKASHITA[†]

In these years, servers and storages are consolidated to the datacenter for reducing TCO of IT systems. Additionally, a variety of virtualization servers is introduced along with the development of server virtualization technology. Consequently, a heterogeneous virtualization server environment appears on the datacenter, and the unified management software that can manage the environment for reducing management cost is needed. However, the each virtualization server was managed separately by a difference of their data models. This paper proposes the method which can achieve unified management and low development cost by communalizing their data models and configuration data gathering processes. This method achieved unified management, in addition we confirmed that the data which the management software had to hold was reduced to 2~3% of the conventional method, and development cost of the configuration data gathering function for another virtualization server was reduced to less than 40% of that.

1. はじめに

近年，データセンタの大規模化，複雑化により，運用管理が困難になっており，運用管理コストの増大が問題となっている．このような中，サーバ仮想化技術はサーバ統合によるコスト削減手段として急速に利用が拡大しており，2012 年における仮想サーバの出荷台数は 2008 年に比べて 10 倍(約 5800 万台)になる見通しとなっている[1]．また，ユーザが利用する仮想サーバの種類は多種に及んでおり，仮想サーバ市場では VMware vSphere[2]や，Microsoft Hyper-V[3]，Citrix XenServer[4]などが登場している．

さらに，運用管理コスト削減に向けた取り組みとしてサーバやストレージの集約が進んでおり，データセンタが大規模化している．そのため，データセンタ内に利用用途が異なる業務システムが混在している．業務システムで利用する仮想サーバ種別の選定基準は，その利用用途に寄るところが大きく，例えば重要な業務を稼働させる場合には，高価でも信頼性が高い仮想サーバが利用される．よって，データセンタ内に前述した多種の仮想サーバが混在するようなヘテロジニアス環境が増加すると予想される．また，このような大規模なデータセンタ環境では，サーバ間で大容量のストレージを共有する構成が多い．

以上より，異種仮想サーバがストレージを共有する環境を一元管理し，管理コストを削減することが求められている．しかし，仮想サーバ環境を管理するために必要な，構成情報を取得する方法と，構成情報のデータモデルが各仮想サーバで異なっていたため，異種仮想サーバを統一的に管理することが困難であった．

そこで本論文では，運用管理ソフトウェア向けに，仮想サーバ間の管理方法の差異を吸収し，ストレージを利用する異種仮想サーバを統一的に管理可能な構成情報管理方式を提案する．さらに，試作プログラムを用いて提案方式の評価を行う．

2. 従来の運用管理技術

本章では従来の運用管理ソフトウェアの構成情報管理方式と仮想サーバ管理の標準仕様について説明する．

2.1 従来の構成情報の管理方式

サーバやストレージを一元管理すべく管理対象機器と運用管理ソフトウェアとの間の管理インターフェースに関する研究[5,6]が行われている．管理インターフェースとは，サーバやストレージといった機器を管理するためのインターフェースであり，管理対象機器によって提供される．また，サーバやストレージを一元管理する商用の運用管理ソフトウェア[7,8,9]も登場している．これらの運用管理ソフトウェアは管理対象の構成情報を管理することで，任意の物理サーバに対してストレージの記憶領域

[†](株)日立製作所 横浜研究所
Hitachi Ltd. Yokohama Research Laboratory

を割り当てる，といった運用を実現していた．構成情報とは具体的に，サーバ仮想化機能を有する物理サーバに対してどのストレージのボリュームが割り当てられ，そのボリュームを物理サーバ上のどの VM (Virtual Machine) が利用しているか，といった，リソース自身の属性情報と，リソースの接続関係の情報である．この構成情報を取得する機能(以下構成情報取得機能と呼ぶ)の処理は大きく次の２つに分類することができる．

- 通信処理部　：運用管理ソフトウェアが管理対象から構成情報を取得するために利用する通信路を確立する処理部
- データ処理部：通信処理部が確立した通信路を介して，管理対象から構成情報を取得し，DB(データベース)に格納する処理部

上記のデータ処理部は管理インターフェースにより，構成情報を取得し，保持する処理である．なお，運用管理ソフトウェアが構成情報を保持する方法は，メモリ中に保持する方法もあるが，構成情報が膨大な場合，運用管理ソフトウェアのメモリを大量に消費してしまうため，管理対象のスケーラビリティを考慮し，DB に保持する方式がよく用いられる．

2.2 仮想サーバ管理のための標準仕様

近年，異種仮想サーバ間を跨って運用するユースケースが登場してきている．その一例としては，異種仮想サーバ間で仮想マシンを相互運用するケースがあり，このユースケースの実現を目的として，仮想マシンフォーマットの標準仕様である OVF（Open Virtualization Format）が登場している．OVF とは仮想マシンの実態であるファイルそのものの仕様であり，この仕様に従った仮想マシンはどのベンダのハイパーバイザ上でも動作することができる．つまり，VMware ESX Server(以降 VMware と呼ぶ)の上で動作していた仮想マシンを Microsoft Hyper-V(以降 Hyper-V と呼ぶ)の上で動作させることが可能となり，異種の仮想サーバ環境間の移行を容易にしている．

しかし，従来の仮想サーバの運用管理は，仮想サーバの機能やアーキテクチャの違いから，仮想サーバの種類ごとに別々の運用管理ソフトウェアにて管理されている．そのため，異種の仮想サーバを一元管理できず，OVF を利用して異種の仮想サーバ間を移行するためには，人手で各仮想サーバの物理構成を把握した上で，複数の運用管理ソフトウェアを使い分け，運用している．

このように仮想サーバの機能としては，異種仮想サーバ間の連携技術が登場しているが，従来の運用管理ソフトウェアでは，これを統一した管理が困難であった．

3. 異種仮想サーバの構成情報収集における問題点

本章では，3.1 節で従来の異種仮想サーバの構成情報収集における問題点について述べ，その問題点の原因として，3.2 節で異種仮想サーバ間のデータモデルの相違点について，3.3 節で管理インターフェースの相違点について述べる．

3.1 従来方式の問題点

従来の運用管理ソフトウェアは，仮想サーバの構成情報をそれぞれのモデルに基づいてユーザに提供している．そのため，ユーザは仮想サーバ毎に異なる情報に基づいて管理業務を行わなければならず，操作目的は同じだが操作内容が異なってしまう．たとえば，ストレージのボリュームからハイパーバイザ上の VM までの関連を辿る場合においては，VMware では，VMFS (Virtual Machine File System) 上にある VMDK (Virtual Machine Disk) を特定し，その VMDK を有する VM を特定する，となるのに対し，Hyper-V では，NTFS (NT File System) 上にある VHD (Virtual Hard Disk) を特定し，その VHD を有する VM を特定する，というように，ユーザ操作が仮想サーバ毎に異なるため，統一的な管理ができない．このため，ユーザに統一的な管理手段を提供する必要がある．これの原因は２つある．VMware と Hyper-V を例とした原因について図 1 に示す．

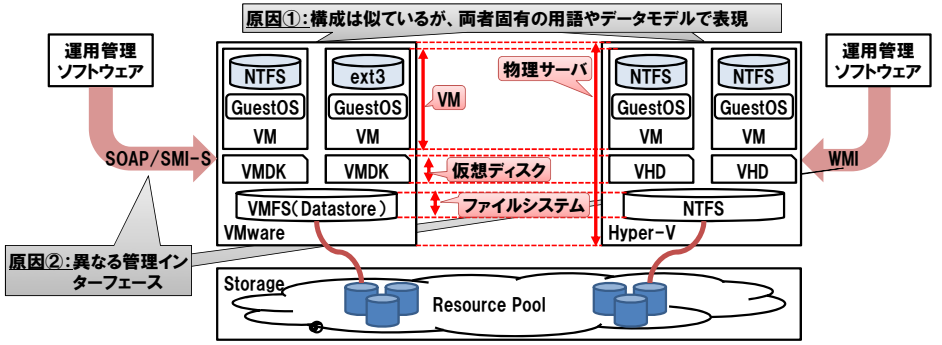

図 1 異種仮想サーバ混在構成例

まず１つ目は，共通のデータモデルがないためである．各仮想サーバは，VM の移動など各種特徴機能に違いはあるが，Hypervisor 型の構成情報としては，図 1 に例示するように各仮想サーバともに類似している．しかし，各仮想サーバのアーキテクチャの違いから用語や，データモデルに違いが出ている．たとえば，VM が利用する記憶領域である仮想ディスクを，VMware は VMDK 形式で管理しており，Hyper-V は VHD 形式で管理している．

２つ目は，仮想サーバと運用管理ソフトウェアの通信で利用可能な管理インターフェースが異なるためである．この違いにより仮想サーバ毎に別々の構成情報取得処理が必要となり，多種仮想サーバへの対応に，莫大な開発コストが必要となってしまう．

次節から，各原因について詳細に説明する．

3.2 データモデルの相違点

VMware や Hyper-V といった仮想サーバは構成要素の表現形式に共に CIM（Common Information Model）[10]を採用している．CIM はオブジェクト指向の表現形式であり，構成要素の情報種別をクラスとして表現し，その実体（構成要素の情報そのもの）をオブジェクトで表す. また, 各クラスの保持する属性情報をプロパティとして規定し，さらに，クラス間の関係はクラスとして規定される(関係クラス)．そして，仮想サーバの全構成要素間の相互関係の定義をプロファイルと呼ぶ.

仮想サーバの管理に利用されている主要な CIM のプロファイルは 2 つある．1 つ目は，ストレージ管理の標準化団体 SNIA (Storage Networking Industry Association)が策定した SMI-S (Storage Management Initiative-Specification) In-Band Virtualizer Model (以降 SMI-S モデルと呼ぶ)である．SMI-S[11]はストレージ管理の国際標準仕様であり，SMI-S モデルは元々，複数のストレージを仮想的に 1 つのストレージに見せる技術である仮想化ストレージを表現するために策定されたモデルである．SMI-S モデルを採用している主要な仮想サーバは VMware であり，VMware はこの SMI-S モデルを自らの仮想サーバ機構を表現するために拡張したモデル(以降 VMware モデルと呼ぶ)を利用している．たとえば CIM_ComputerSystem クラスは，SMI-S モデルではストレージ自体を表現しているのに対し，VMware モデルではサーバ仮想化機能を有する物理サーバ自体を表現している.

2 つ目の主要な CIM プロファイルは，システム管理の標準化団体である DMTF (Distributed Management Task Force)が策定した CIM System Virtualization Model(以降 SV モデルと呼ぶ)である．SV モデルは，仮想サーバを表現するために策定されたモデルであり，Hyper-V や Xen 等に採用されており，今後さらに普及する見込みである.

VMware モデルと SV モデルの主な相違点を表 1 に示す.

表 1 VMware モデルと Hyper-V モデルの相違点の概要

構成要素種別	VMware モデル	SV モデル
VM	CIM_ComputerSystem	CIM_ComputerSystem
仮想ディスク	CIM_StorageVolume	CIM_LogicalDisk
（物理サーバの）ファイルシステム	CIM_StoragePool	CIM_LogicalDisk
物理サーバ	CIM_ComputerSystem	CIM_UnitaryComputerSystem

*1：CIM_UnitaryComputerSystem の親クラスが CIM_ComputerSystem

表に示すように，VM については，対応する CIM のクラスが共通であるため，両モデルで表現される構成情報が類似しているが，VM が利用する仮想ディスクを示すクラスと，物理サーバのファイルシステムを示すクラスについては，対応する CIM のクラスが異なるため，同じ構成要素を表現するための情報が異なっていた．このため，ユーザは仮想サーバ毎に異なる情報に基づいて管理業務を行わなければならず，操作目的は同じだが操作内容が異なり，統一的な管理ができない.

3.3 管理インターフェースの相違点

表 2 に主要な仮想サーバの管理インターフェースを示す.

表 2 管理インターフェース一覧

仮想サーバ種別	管理インターフェース
VMware ESX Server 4.0	SOAP
	SMI-S
Microsoft Hyper-V 2.0	WMI
Citrix XenServer 6	XML-RPC
	libvirt-cim

表 2 に示すように各仮想サーバの管理インターフェースは多種多様であり,VMware は自社独自の管理インターフェースである SOAP と標準仕様の SMI-S を採用しており，Hyper-V は Windows が OS の標準機能として搭載している WMI (Windows Management Instrumentation)を採用している．このように，仮想サーバ毎に管理インターフェースが異なっており，これは各仮想サーバがそれぞれの仮想化技術の特徴を生かすべく，各仮想化技術に適した仕様を採用しているためだと考えられる.

これらの管理インターフェースの違いにより，運用管理ソフトウェアは構成情報を取得する際, それぞれ異なる管理インターフェースを呼び出す必要がある. そのため，異種仮想サーバ環境においては，サポートする仮想サーバの種類毎に開発を行う必要があり，開発コストの増大につながっていた.

4. 提案方式

3 章で述べた問題を解決するべく， 異種仮想サーバ間でデータモデルの共通化を図り，さらに異種仮想サーバに対する構成情報収集アルゴリズムを共通化することで，異種仮想サーバを統一的に管理可能とする方法を提案する.

4.1 データモデルの共通化

異種仮想サーバ間で統一的な管理を実現するために，共通データモデルを作成すべく，データモデルの要件を次のように定義した.

- 仮想マシン(VM)と物理サーバのファイルシステムとの対応関係がわかる
- 物理サーバのファイルシステムと，ESX Server/Hyper-V との対応関係がわかる
- 物理サーバのファイルシステムとストレージボリュームとの対応関係がわかる
- ストレージボリューム，物理サーバのファイルシステム，仮想ディスクの容量がわかる

これらは，仮想マシンのデータを保持するストレージから，仮想マシン上にインス

トールされたゲスト OS までの一連の関係を把握するための要件であり，運用管理ソフトウェアは，これらの要件を満たす構成情報を利用することで，ストレージを利用する仮想サーバ環境における多くの運用をサポート可能である．運用の一例としては，ストレージにて発生した障害の影響を受ける VM を特定するといったものがある．

上記要件を満たす異種仮想サーバ間共通データモデルを検討した．提案データモデルを図 2 に示す．

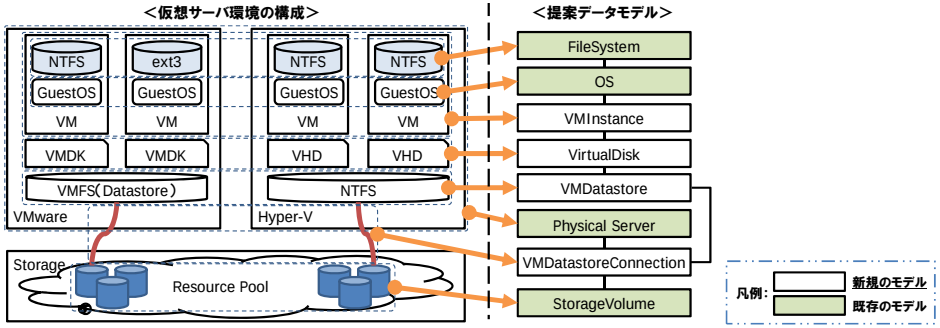

図 2 共通データモデル

図 2 ではさらに，提案データモデルと，VMware/Hyper-V の構成との対応関係を図示している．たとえば，提案データモデルの VirtualDisk は VMware では VMDK を示しており，Hyper-V では VHD を示している．

次に，提案データモデルの各クラスの意味と CIM クラスとの対応を表 3 に示す．

表 3 共通データモデルの各クラスの説明

#	クラス名	説明	CIM クラスとの対応	
			VMware モデル	SV モデル
1	VMInstance	VM	CIM_ComputerSystem	CIM_ComputerSystem
2	VirtualDisk	仮想ディスク	CIM_StorageVolume CIM_ComputerSystem CIM_StoragePool	CIM_LogicalDisk
3	VMDatastore	(物理サーバの)ファイルシステム	CIM_StoragePool	CIM_LogicalDisk
4	PhysicalServer	物理サーバ	CIM_ComputerSystem	CIM_ComputerSystem
5	VMDatastore Connection	ファイルシステム(#3)と物理サーバ(#5), ストレージボリューム(#6)との関係	CIM_StoragePool CIM_StorageExtent	-

表 3 は，提案データモデルの各クラスは，図 2 に示す仮想サーバ環境の各構成要素に対応する CIM クラスの情報をベースにしているが，VMware モデルと SV モデルとで構成要素に対応する CIM クラスが異なっていた．そのため，提案データモデルは CIM モデルと 1 対 1 で対応しておらず，複数の CIM クラスの情報を集約したモデルになっている．例えば，表 3 の#2 に示すように，VirtualDisk クラスは VMware モデルの場合, CIM_StorageVolume, CIM_ComputerSystem, CIM_StoragePool から構成される．

運用管理ソフトウェアが提案データモデルに基づき仮想サーバの構成情報をユーザに提供すれば, ユーザは異種仮想サーバが混在する環境を統一的に管理可能となる．

4.2 構成情報取得アルゴリズムの共通化

従来の構成情報取得機能について図 3 に示す．以下，図 3 に沿って説明する．

図 3 従来方式

従来方式はまず，通信部により仮想サーバが提供する各管理インターフェースに従って仮想サーバとの通信路を確立する．次に，情報取得部が仮想サーバから構成情報を取得するが，この取得処理を各データモデルに従って，全構成要素分繰り返す．そして DB 格納部が取得した情報を DB に格納し, 通信を切断する．この従来方式では，データモデル毎に処理が異なるために，開発コストが増加してしまう．

そこで，処理の共通化を目指して，共通部位の抽出を行った．その結果，図 3 に示す VMware/SV モデル情報取得処理のうち，枠で囲われている部分はモデル固有の部分であるものの，それら以外の部分については共通していることが分かった．これら共通部分は，CIM に関するデータ処理の標準仕様[12]に類似したものであった．そこで，上記の共通点を利用した方式を考案した．提案方式の概要を図 4 に示す．

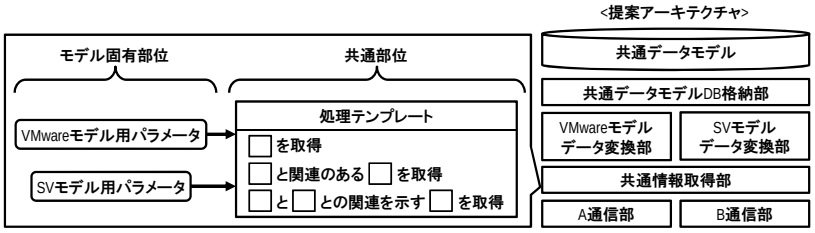

図 4 提案方式

提案方式では，仮想サーバ固有のデータモデルに従って実行する情報取得処理をテンプレート化し，各データモデルに固有の部位をパラメータ化した．本方式により管理インターフェースの違いにより，仮想サーバ毎に異なっていた構成情報取得処理に対し，共通化をはかり，開発コストの削減に成功した．

5. 評価

本章では，提案方式の試作プログラムを開発し，共通データモデルと共通アルゴリズムについて，保持するデータ量と処理ステップ数と処理性能の評価結果を述べる．

5.1 共通データモデルの評価

提案方式の処理のうち，管理インターフェースから取得した CIM 形式の構成情報を 4.2 節で示した共通データモデルへ変換する処理（変換処理）は，元の形式の構成情報の中から 4.1 節で述べた要件を満たすために必要な情報だけを抽出する．このため，運用管理ソフトウェアが保持するデータ量を削減することが可能である．そこで試作プログラムを用いて，構成情報のデータ量を測定した．測定では，DB に格納された構成情報を CSV(Comma Separated Values)ファイルへエクスポートし，それぞれファイルサイズの総計を比較した．評価結果を図 5 に示す．

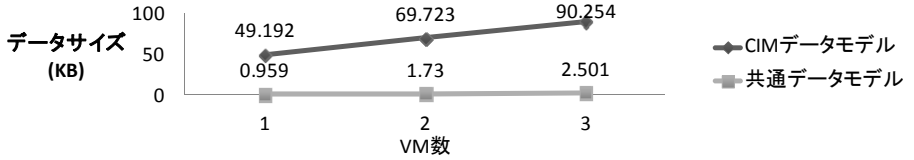

図 5 提案方式のデータサイズ評価結果

本測定では，対象の仮想サーバを VMware とした．測定環境としては，構成要素としてストレージボリュームと物理サーバのファイルシステム，仮想ディスク，VM が各 1 つ存在し，それらが互いに関連を持つ構成を 1VM という単位として，1~3VM の構成について測定を行った．測定の結果，図 5 に示すように，提案データモデルのサイズは元の VMware モデルのせいぜい 2~3%であり，高い圧縮率を示している．これは，データモデルの要件達成に不要な情報が削除されていることを示している．

5.2 構成情報取得アルゴリズムの評価

5.2.1 構成情報取得処理のステップ数

開発コスト削減に向けて，4 章で述べた提案方式の処理ステップ数を評価する．評価に際し，図 3 と図 4 に示す通信部と，情報取得部，データ変換部，DB 格納部について，通信部の通信路確立/切断処理が各 1 ステップ，情報取得部のテンプレート化した 3 種類の情報取得処理が各 10 ステップ，データ変換部の元のデータモデルのクラスから共通データモデルのクラスを作成する処理が元の 1 クラス毎に各 1 ステップ，DB 格納部の 1 クラスの情報を DB へ格納する処理が各 1 ステップ，とした．評価結果を表 4 に示す．

表 4 処理ステップ数の比較

方式	対象種別	通信部処理	情報取得部処理*1	データ変換部処理*1	DB 格納部処理*1	合計処理ステップ
従来方式	VMware	2	30	0	40	72
	VMware, Hyper-V	4	60	0	69	133
提案方式	VMware	2	30	40	5	77
	VMware, Hyper-V	4	30	69	5	108

*1：取得対象のクラス数を VMware= 40, Hyper-V = 29, 変換対象のクラス数 = 5 として算出

表に示すように，従来方式では運用管理ソフトウェアが対応する仮想サーバを追加する場合に，1 種類目の仮想サーバを対応した際に要した開発コストとほぼ同等の開発コストを要していた．提案方式では，仮想サーバを追加する場合においても，処理の共通化により，1 種類目の仮想サーバを対応した際に要した開発コストのおよそ 40%程度の開発コストで対応が可能である．

5.2.2 データモデル変換処理性能

次に，提案方式で追加したデータモデル変換処理の性能への影響を知るために，同様の試作プログラムにて，データモデル変換の処理時間を計測し，構成情報取得機能の処理における負荷を評価した．

前項の測定と同様の構成で，1VM から 3VM まで構成を増加させて処理時間を測定した結果を図 7 に示す．

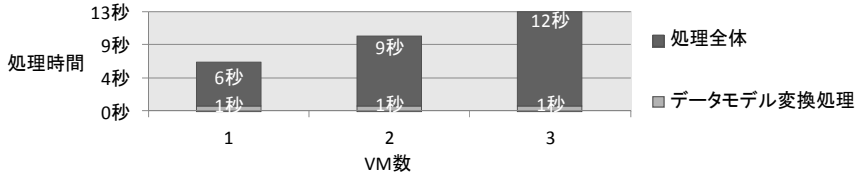

図 6 提案方式の処理性能評価

測定の結果，全体の処理時間は構成の大きさに比例して増加しているものの，データモデル変換処理時間はほとんど増加しておらず，1VM 構成でも処理全体の中でデータモデル変換部の処理が占める割合は高々10%であり，構成が大きくなるに従って割合は減少している．

6. 考察

本章では，前章の評価結果をもとに，提案方式について考察を述べる．

6.1 共通データモデルの有用性

3.2 節に述べたように，従来方式では仮想サーバごとにデータモデルが異なっていた．そのため，ユーザは仮想サーバの種別毎に管理業務を行わなければならず，目的は同じだが操作内容が異なってしまい，管理が煩雑になっていた．

そこで，これを解決すべく提案のデータモデルでは，ユーザが行う管理操作を損なわず，複数種類の仮想サーバの構成を抽象化したデータモデルを考案すると共に，仮想サーバの種類毎に異なるデータモデルから抽象化されたデータモデルへの変換を実現した．

これにより，各種仮想サーバが提供するデータモデルが異なっていても，運用管理ソフトウェアが提供するレポートや管理操作の統一，更に OVF を使ったデータ移行のように異なる種類の仮想サーバ間を連携する運用が可能である．また，運用管理ソフトウェアにて情報を収集する際, 仮想サーバの構成を抽象化したデータモデルにて, 構成情報を DB へ格納する. そのため, 運用管理ソフトウェアが保持するデータ量を, 従来まで仮想サーバの種類毎に別々に格納していた場合に比べ，2~3％まで圧縮できる．これは，大規模なデータセンタの運用管理において，管理対象の構成は大規模であっても，運用管理ソフトウェア自身の規模を小さく実現することが出来る．

6.2 共通アルゴリズムの有用性

3.3 節で述べたように，従来方式では，運用管理ソフトウェアは構成情報を取得する際，仮想サーバ毎に異なる管理インターフェースを呼び出す必要があるため，サポートする仮想サーバ毎に開発を要し，開発コストの増大につながっていた．

提案した共通アルゴリズムでは，運用管理ソフトウェアが仮想サーバを追加でサポートする場合においても，異種仮想サーバ間で類似している処理をテンプレート化したことで，処理を共通化でき，開発コストを従来の 40%程度まで削減可能である．これにより，仮想サーバへの迅速なサポートが可能となる．

また, 本論文では共通化した処理は CIM に関するデータ処理の標準仕様に従った処理としたが，提案アルゴリズムは，このデータ処理に限らず，構成情報を取得する処理をテンプレート化するものである．このため，本方式は仮想サーバが提供する独自の管理インターフェースにも拡張可能である．

7. おわりに

本論文では，異種仮想サーバを管理する運用管理ソフトウェア向けに，異種仮想サーバに対してデータモデルとアルゴリズムを共通化した構成情報取得方式を提案した．

従来方式では，各仮想サーバが採用しているデータモデルに従って構成情報を保持していたために，異種の仮想サーバ間でデータモデルの意味が異なり，ユーザに統一的な管理を提供できなかった．また，各仮想サーバの管理インターフェースが異なるため，運用管理ソフトウェアは各管理インターフェースが提供している構成情報取得手段をそれぞれ別々に呼び出す処理が必要であり, 開発コスト増大につながっていた．

提案方式では，それぞれ個別のモデルであった異種仮想サーバのデータモデルを統一可能な共通データモデルを提案し，異種仮想サーバであっても統一化された操作で管理できる基盤を確立した．また，この共通データモデルの実現により，別々のモデルで構成情報を保持するよりもデータ量を 2~3%まで削減することに成功した．

さらに，提案方式では，異種仮想サーバ間で類似している処理をテンプレート化することで, 異種仮想サーバに対する処理を共通化した. この共通アルゴリズムにより, 構成情報取得機能の処理ステップを従来方式から 40％まで削減することに成功した．

そして，提案方式が他の仮想サーバに対しても拡張可能であることを明らかにした．

本論文では，運用管理ソフトウェアを用いた，異種仮想サーバの構成情報管理機能の実現方法に着目した．今後の課題は，この構成情報をベースとした仮想サーバ環境の運用管理機能に着目し，ユーザ環境に合わせて運用管理を容易化する設計を行う．

参考文献

1) Gartner, “Server Virtualization: Emerging to Mainstream at Lightspeed”, Gartner Symposium/ITxpo 2009, (2009/11)
2) VMware, Inc.: VMware vSphere
3) Citrix Systems, Inc.: Citrix XenServer
4) Microsoft Corporation: Microsoft Hyper-V
5) 上原 敬太郎，水野 和彦，田中 剛，垂井 俊明：仮想環境統合モニタリング技術の開発と評価，情報処理学会 インターネットと運用技術シンポジウム 2010 論文集 Vol.2010, No.14, pp97-104.(2010)
6) 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ー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, 掲載予定(2012).
7) (株)日立製作所: Hitachi Command Suite.
http://www.hitachi.co.jp/products/it/storage-solutions/products/software/hsms/index.html
8) EMC: PROSPHERE
http://japan.emc.com/it-management/prosphere/prosphere.htm
9) IBM: IBM Tivoli Storage Productivity Center.
http://www-06.ibm.com/systems/jp/storage/software/productivity/
10) DMTF: CIM. http://www.dmtf.org/standards/cim/
11) SNIA: SMI-S.http://www.snia.org/forums/smi/tech_programs/smis_home/
12) DMTF: WBEM. http://www.dmtf.org/standards/wbem/