

仮想環境統合モニタリング技術の開発と評価

上原 敬太郎[†] 水野 和彦[†] 田中 剛[†] 垂井 俊明[†]

仮想化技術の発展は IT 機器の統合運用を可能にする一方で、仮想的な機器の監視手段・情報が増加するため、システム管理者にとっては仮想環境全体の動作状況監視が困難な状況を招いている。報告者らは各物理および仮想機器の統計情報の取得と可視化を一括して行う統合モニタリングツール Virtage Monitor を提案する。Virtage Monitor は複数の物理/仮想機器の稼動情報を時系列および構成情報に基づき集計し表示することで、仮想環境の統合的な稼動状況を把握を容易にする。プロトタイプを用いて CLI や個別のモニタとの比較実験を行った結果、リソースボトルネック検出までの時間が最大 70~75% 短縮し、ウインドウ操作の回数も 80% 減少した。これにより提案方法が仮想環境全体の動作状況把握作業改善に有効であることが確認された。

Development and Evaluation of an Integrated Performance Monitoring Tool for Virtualized Servers

KEITARO UEHARA[†], KAZUHIKO MIZUNO[†], TSUYOSHI TANAKA[†], TOSHIKI TARUI[†]

An integrated monitoring tool, called “Virtage Monitor,” that allows the acquisition and visualization of statistical information from individual IT components, has been prototyped. Virtage Monitor displays the statistical information concerning the operation of the multiple components in time-series form in accordance with the composition information concerning a server and storage. Moreover, it helps administrators to comprehend integrated virtualized-system information. It was experimentally shown that the prototype Virtage Monitor can reduce the time taken to identify a performance bottleneck in resources by about 75% in the best case, and that Virtage Monitor can also reduce the number of window operations by about 80% on an average than in the case of using a nonintegrated monitor. This result indicates that Virtage Monitor will be an essential tool for administrators aiming to save time when identifying bottlenecks in datacenters.

1. 背景と目的

IT システムの肥大化・複雑化を受けて運用管理コストが増加し問題となっている[1]。本問題を受けて、運用管理コスト削減に向けた仮想化技術によるサーバ統合が着目されている。日立でも、統合サービスプラットフォーム BladeSymphony[2]にサーバ仮想化機構 Virtage を搭載したモデルを 2007 年から投入している[3,4,5]。

このような仮想化技術の発展によりサーバからストレージまで統合的に運用することが可能となったが、仮想環境はより多様な機器から構成されることになった。そのためシステム管理者は、仮想/物理リソースをそれぞれの機器に準備された管理ツールや独自で開発したコマンドを利用し、各機器の統計情報取得を行っており、仮想環境全体の動作状況監視が困難となっている。

このように仮想化技術により監視対象となるコンポ

ーメントが増加する一方、仮想化ソフトウェア自体は無償化・低価格化が進行しており、今後は増加した仮想化コンポーネントを管理するツールの市場の伸びが期待されている[6]。仮想化環境における適切なサイジング・プランニングや、素早い問題検出・切り分けのために、構成情報および動作状況を可視化するツールの重要性が増している。

そこで本研究では、仮想化環境におけるシステムの動作状況の把握を容易にするために、サーバとストレージ、および物理リソースと仮想リソースといった各コンポーネントの統計情報の取得と、構成情報に基づく可視化を一括して行える、システム管理者向けの統合モニタリングツール Virtage Monitor (仮称) を提案する。本稿の目的は、Virtage Monitor のプロトタイプを開発し、実験によりシステム管理者の動作状況把握までにかかる時間が短縮できることを明らかにすることである。

[†](株)日立製作所 中央研究所
Hitachi Ltd., Central Research Laboratory

2. 仮想環境モニタリングの概要と課題

2.1 仮想環境におけるモニタリングの必要性

稼動状況や性能をモニタリングする機能は、仮想化とは無関係にこれまでも IT システムで必要とされてきた機能である。しかし仮想化が導入されることで、物理環境と比較して次の新たな要求が発生する：

(1) 物理環境を仮想環境に移行（統合）した場合の性能推定

仮想環境に移行したいというニーズは、複数の物理環境を統合して TCO 削減したい、リソースを有効活用したい、という要求から生じている。このため、実際の仮想環境への移行前に、移行後の性能やリソース使用状況を把握できることは、仮想化環境移行に対する重要な動機付けになる。

(2) 各サーバの仮想リソース—物理リソースの使用状況のマッピング

通常の運用では、各仮想サーバの管理者と物理サーバの管理者とは異なるケースが多い。各仮想サーバの管理者は、自分が使用する仮想サーバに割り当てられたリソース（仮想リソース）が足りていれば問題ないと考え、物理サーバの管理者は物理サーバ全体のリソース（物理リソース）が不足してきた場合、仮想サーバを他の物理サーバ上へ移動させるなどの処置が必要になるケースがある。このように、管理者のレイヤーによって必要とされる情報が異なるため、必要に応じて稼動状況をフィルタリングあるいは集計・統合化して、ユーザや管理者が着目している情報のみを効率よく取得できる機能が求められる。

2.2 仮想環境モニタリングの問題点

一般に物理サーバと仮想サーバが混在した仮想化環境において動作状況を把握しようとした場合、次の問題点がある。

- (1)対象コンポーネント（物理/仮想サーバ・ストレージ・ネットワーク）ごとにモニタリング方法がまちまちで、統一的に可視化する手段が存在しない。
- (2)物理リソースのボトルネックと仮想リソースのボトルネックが必ずしも一致しない。
- (3)複数の仮想化ソフトウェア（Hyper-V や VMware 等）が混在する環境では、仮想化ソフトウェアの種別ごとに異なるツールが必要で、かつ指標も統一されていない。

表 2-1：対象コンポーネントと取得できる情報

#	対象コンポーネント	取得できる情報
1	仮想サーバ	CPU 利用率, I/O 利用率, メモリ消費量
2	物理サーバ（ハイパバイザ; HVM ¹ ）	論理 CPU 利用率（LPAR 毎）, 物理 CPU 利用率
3	ストレージ	RAID コントローラ利用率, 物理ドライブ利用率

1 HVM=Hitachi Virtualization Manager

以下、順に説明する。

(1)の対象コンポーネントごとに取得できる情報の例を表 2-1 に示す。これらのサーバやストレージは元々大規模システムを前提としており、コンポーネントごとに異なる SE が対応する前提でツールが開発されてきた。また、仮想サーバの統計情報は標準的な OS のコマンド(sar 等)を使って取得できるが、物理サーバやストレージの統計情報はハイパバイザやストレージの機種毎に取得方法が異なることが多い。このため対象コンポーネントごとに異なるツールと機器を特定する固有の ID 等を用い、コマンドラインからそれぞれのコマンドを入力し、結果を取得する必要がある。

(2)の仮想コンポーネントと物理コンポーネントの関係が簡単に把握できないことによって、例えばある仮想サーバの CPU 利用率が高い場合、仮想サーバのパフォーマンス値だけを見ても、物理 CPU のリソースが足りないのか、それとも仮想サーバへの割当率が低いだけなのかを区別できない。

(3)の複数ベンダの仮想化ソフトウェア対応については、仮想化自体がまだ新しい概念であるため標準化が進んでおらず、各社独自のツール群を用意しているため、互いのインタフェースに互換性が無い点が問題となる。各社の仮想化ソフトウェアが出力する性能指標が統一されていないため、ユーザは異なる仮想マシン間の適切な比較ができない。

2.3 仮想環境統合モニタリング実現に向けた課題

表 2-2：問題点と課題

#	問題点	課題
1	複数コンポーネント（物理/仮想サーバ・ストレージ・ネットワーク）のモニタリング情報が点在し、指標も不統一。	複数のモニタリング情報を構成情報に従って統一的にまとめあげる手段が必要。
2	物理リソースと仮想リソースのモニタリング情報の相関が不明。	割当状況を考慮したうえで物理と仮想のリソース使用情報をまとめ、ユーザが着目した情報に素早くアクセスできる手段が必要。
3	仮想化ソフトウェア間のモニタリング指標・手段が不統一。	複数仮想化ソフトウェア間の指標の相互変換する手段、および後から対応仮想化ソフトウェアを追加できるような仕掛けが必要。

表 2-2 に前節の 3 つの問題点に対応する課題を示す。これら 3 つの課題を解決するツールが必要となる。

第一の課題は、物理/仮想サーバ・ストレージ・ネットワークといった複数のコンポーネントそれぞれのモニタリング情報を取得する手段はあるが、それらを統一的に見る手段が無い点である。このため、たとえばストレージのビジー率上昇が、各仮想サーバにどのような影響を与えているかの相関関係を把握するためには、各仮想サーバが使用するストレージの情報（構成情報）との対応付けが必要となる。

第二の課題は、物理リソースと仮想リソースのモニタリング手段・指標が異なるために、直接比較や相関関係を把握できない点である。たとえばハイパバイザ本体の負荷により、各仮想サーバの負荷の総和が物理サーバの負荷と一致しない場合や、逆に物理サーバに余裕があってもキャッピングにより仮想サーバは過剰負荷となっているケースなどが考えられる。稼働状況や着目している点に応じて物理と仮想リソースの使用状況を素早く切り替えて比較できる機能が必要となる。

第三の課題は、複数仮想化ソフトウェア間で、パフォーマンス情報として取得できる項目や指標が統一されていない点である。これを解決するためには、仮想化ソフトウェア間の指標を正規化する手段と、新しい仮想化ソフトウェアに素早く対応できるようなプラグイン構成を持つことが必要となる。

3. 仮想化環境統合モニタリングツールの提案

3.1 仮想化環境統合モニタリングツールの概要

前節の課題に対して、以下のように統合的に動作状況を把握できるモニタリングツール Virtage Monitor を開発することで解決を試みる：

- (1) 複数コンポーネント（物理/仮想サーバ・ストレージ・ネットワーク）の統合モニタ
 - (2) 物理リソースと仮想リソースの統合モニタ
 - (3) 複数仮想化ソフトウェアの統合モニタ
- 各項目の実現に向けて、以下のアプローチを採る。

(1) 複数コンポーネント（サーバ・ストレージ・ネットワーク）の統合モニタ

各々の機器で取得されたモニタリング値のタイムスタンプを元にして同一グラフ上に統合的に可視化することで、同時刻における分散した機器の振る舞いを一覧できるようにする。

必要があれば取得された統計情報のうち、着目する機器（例：ある仮想サーバが使用しているディスク）の情報のみを抽出して表示できるようにする。

(2) 物理リソースと仮想リソースの統合モニタ

各ゲスト OS 上の CPU 利用率/IO 利用率、および HVM 上の仮想サーバ・ハイパバイザ稼働率、共有 CPU 割当比率等を一覧できるようにする。

割当ポリシー（占有割当/共有割当/キャッピング有無）を考慮した表示を行うことで、仮想-物理間のマッピングを明確にし、ボトルネックを見つけやすくする

着目したリソースについての時系列変化を見られるようにする。たとえば、共有 CPU の場合はゲスト間の利用率を積層グラフにして表示することで、どの仮想サーバがリソースを使用しているかを一覧できるようにする。

(3) 複数仮想化ソフトウェアの統合モニタ

仮想化ソフトウェア間で共通となる部分（ベース）と、仮想化ソフトウェアごとにカスタマイズが必要なプラグイン部分に分けて設計し、プラグイン部分を後から追加していくことで、対応する仮想化ソフトウェアを増やせるようにする。

また、異なる仮想化ソフトウェア間の性能指標を比較できるように、性能指標の正規化を行う。

今回の Virtage Monitor プロトタイプ開発に当たっては、(1)のサーバとストレージの統合および(2)、そして(3)についてはまず日立仮想化機構 Virtage への対応を実装することにした。Virtage は論理分割方式(Logical Partitioning; LPAR)を採用したサーバ組み込みの仮想化機構である。

3.2 仮想化環境統合モニタリングツールの実装方法

File	LPAR	System	Blade2/LPAR Information						
Server		Ded Proc#	0						
		Shr Proc# (Srv%)	8 (100%)						
		Processor Capping	No						
		Memory (MB)	1792						
		VNIC#	0	1	2	3	4	5	6
Network		IP Addr	192.168.0.40	-	-	-	-	-	-
		MAC Addr	00:12:34:56:78:9A	00:12:34:56:78:9B	-	-	-	-	-
LPAR1		VHBA#	0						
		WWPN	9876543210ABCDEF						
FC HBA		HostGroup(Name)#	Blade2_LPAR01(124)						
		HostLUN(LUN)	0(70)	1(44)	2(9)				
		CTL/Port(RG)	1A(6)	1A(4)	1A(5)				
		Capacity	10.24GB		10.24GB				
		Unit:[HDUs]	0: [12,13,14]	0: [8,9]	0: [10,11]				
Server		Ded Proc#	0						
		Shr Proc# (Srv%)	8 (100%)						
		Processor Capping	No						
		Memory (MB)	1792						
		VNIC#	0	1	2	3	4	5	6
Network		IP Addr	192.168.0.41	-	-	-	-	-	-
		MAC Addr	00:12:34:56:78:CD	00:12:34:56:78:CE	-	-	-	-	-
LPAR2		VHBA#	0						
		WWPN	1						

図 3-1: Virtage Monitor Blade/LPAR 情報画面

図 3-1 は Virtage Monitor の Blade（この場合は Blade2）の各 LPAR の情報を示す画面である。縦に LPAR ごとの情報が表示され、それぞれの LPAR について表 3-1 に示す構成情報が一覧表示される（表 2-2 の課題 1：複数コンポーネントの構成情報を一元化に対応）。LPAR タブからリンクを辿ることで各 LPAR のパフォーマンス情報が表示される。

表 3-1: 各構成情報の意味

#	カテゴリ	項目	意味
1	Server	Ded Proc#	占有割当 CPU コア数
2		Shr Proc#	共有割当 CPU コア数 (割当率)
3		Processor Capping	割当率のキャッピングを行うかどうか
4	Network	Memory	メモリ割当量(MB)
5		VNIC#	仮想 NIC#
6		IP Addr	IP アドレス
7		MAC Addr	(仮想)MAC アドレス
8	FC HBA	VHBA#	仮想 HBA#
9		WWPN	World Wide ポート名
10		HostGroupName(#)	ホストグループ名(#)
11	RAID	HostLUN (LUN ⁴)	ホスト論理ユニット# (論理ユニット#)
12		CTL/Port (RG)	コントローラ/ポート (RAID グループ#)
13		Capacity	ボリューム容量(GB)
14		Unit: [HDUs]	ユニット# [ハードディスクユニット#]

2 NIC=Network Interface Card

3 HBA=Host Bus Adapter

4 LUN=Logical Unit Number

図 3-2 に LPAR (この場合は Blade2/LPAR3) のパフォーマンス情報表示画面を示す。上の段から大きく Server, HVM, Storage と分類されている。それぞれの項目の意味を表 3-2 に示す。また、HVM に関してはボタンを押すごとに(a)LPAR 単独のビジー率と、(b)全 LPAR 積算のビジー率の表示を切り替えられるようにした(表 2-2 の課題 2: 物理リソースと仮想リソースのマッピングと素早い切り替えに対応)。

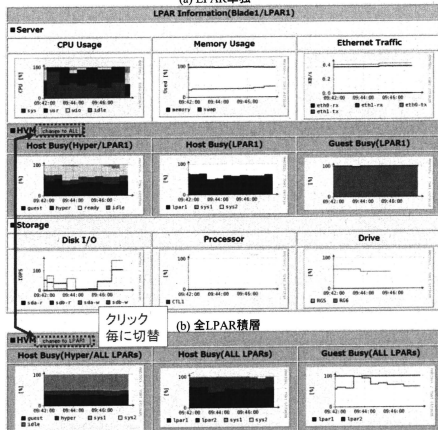


図 3-2 : Virtage Monitor のパフォーマンス情報画面

表 3-2 : 各パフォーマンス情報の意味

#	カテゴリ	項目	意味
1	Server	CPU Usage	仮想サーバ (LPAR) の CPU ビジー率
2		Memory Usage	仮想サーバ (LPAR) のメモリ利用率
3		Ethernet Traffic	仮想サーバ (LPAR) のイーサネット流量
4	HVM (LPAR 単独)	Host Busy (Hyper/LPAR)	Hyper/Guest 別の物理 CPU ビジー率
5		Host Busy (LPAR)	LPAR 分 + System 分の物理 CPU ビジー率
6		Guest Busy (LPAR)	LPAR の論理 CPU ビジー率
7	HVM (全 LPAR 積層)	Host Busy (Hyper/All LPARs)	Hyper/Guest 別の物理 CPU ビジー率 (全 LPAR 分合計)
8		Host Busy (All LPARs)	LPAR 別の物理 CPU ビジー率 (全 LPAR 分積層)
9		Guest Busy (All LPARs)	全 LPAR の論理 CPU ビジー率一覧
10	Storage	Disk I/O	仮想サーバの I/O トラフィック (ディスク/RW 毎)
11		Processor	コントローラ・プロセッサのビジー率 (コントローラ毎)
12		Drive	物理ドライブのビジー率 (RAID Group 毎)

4. 仮想化環境統合モニタリングツールの評価

本章では、Virtage Monitor の効果を評価するため、3通りの手段を用いてリソースボトルネックを検出する作業を行ってもらい、かかった時間および操作回数を測定する実験を行う。

4.1 評価環境

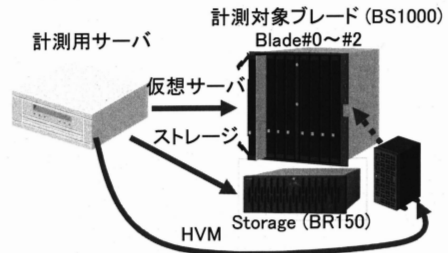


図 4-1 : 実験環境のシステム構成

仮想化環境統合モニタリングツールの評価を行うために、図 4-1 に示す実験環境を用意した。計測対象のサーバとしては統合サービスプラットフォーム BladeSymphony BS1000 のブレード上に Virtage を動作させ、合計 5 つの仮想サーバ (LPAR) を用意し、それぞれの LPAR 上では Linux を動作させる。ストレージは日立製ストレージ BR150 を使用する。ブレード (物理サーバ) およびストレージのスペックを表 4-1 に示す。

表 4-1 : 実験環境のマシンスペック

#	筐体	項目	設定値
1	Blade#0	CPU	Dual Core Xeon 5160 (Woodcrest) 3.0GHz×1
2		Memory	4GB
3		LPAR 数	3
4	Blade#1	CPU	Dual Core Xeon 5110 (Woodcrest) 1.60GHz×2
5		Memory	8GB
6		LPAR 数	2
7	Blade#2	CPU	Quad Core Xeon E5430 (Harpertown) 2.66GHz×2
8		Memory	16GB
9		LPAR 数	3
10	Storage	ユニット数	72GB×15 ドライブ数
11		RAID クラス	RAID0 (Striping)
12		Group 数	7 (2D or 3D)
13		コントローラ数	2

4.2 評価方法

2台のブレード上に合計5台の仮想サーバ（LPAR）が動作している状態で、それぞれの仮想サーバ上に異なる負荷を生じるプログラムを稼働させる。負荷をかけるために使用したプログラムを表4-2に示す。

表 4-2：負荷プログラムの種類と目的

#	負荷プログラム	目的
1	SPECjbb2005[7]	CPU 負荷をかける
2	iozone[8] (write)	I/O 負荷（主に write）をかける
3	iozone (read)	I/O 負荷(主に read) をかける

被験者には、次の5つのパラメータについて、性能ボトルネックが生じている仮想サーバ（LPAR）を探してもらい、完了するまでの時間を計測する。ここで、性能ボトルネックとはビジー率80%以上と定義する：

- (1) 仮想サーバ CPU ビジー率
- (2-a) HVM からみたゲスト CPU ビジー率
- (2-b) HVM からみたホスト CPU ビジー率
- (3-a) ストレージのコントローラビジー率
- (3-b) ストレージの物理ドライブのビジー率

本実験では、表 2-2 の課題 1・課題 2 に対する Virtage Monitor の効果を測るため、複数コンポーネント（サーバ・ストレージ）および仮想リソース（仮想サーバ）と物理リソース（HVM）とを対応付ける必要がある状況を作り出している。

各被験者は CLI(コマンドラインインタフェース)、非統合版モニタ使用、統合版モニタ(Virtage Monitor)使用のケースでそれぞれ以下の条件で実験を行う：

- CLI の場合、(1)~(3)の性能値を測定するためのコマンドと、出力の見方のマニュアルを予め作成し、十分に説明した上で、実験を開始する。図 4-1 では、(1)はゲストに対する ssh 経由のコマンド実行、(2)は HVM 管理用端末を経由してのコマンド実行、(3)はストレージに対するコマンドの実行、とそれぞれ異なる経路によるコマンド発行が必要となる。
- 非統合版モニタ使用および統合版モニタ使用の場合、各モニタツールのトップページの URL と、性能グラフの見方を予め説明した上で、実験を開始する。

実験は CLI と統合版モニタ使用の場合の比較、および非統合版モニタ使用と統合版モニタ使用の場合の比較、の 2 通り行い、それぞれにおけるボトルネック検出までの時間やウインドウ操作回数を比較する。

表 4-3：被験者のプロフィール

業種	初心者	ベテラン
システムエンジニア	1 人	1 人
サーバ研究者	0 人	2 人
ネットワーク研究者	1 人	2 人

今回 2 種類の実験を延べ 7 人の被験者で行った。本実験は、運用管理 SE が仮想環境においてリソースボトルネックを検出する作業を想定しているため、被験

者には SE 作業に慣れた者と慣れていない者を取り混ぜて行った。各被験者の簡単なプロフィールを表 4-3 に示す。

4.3 CLI と統合版モニタの比較

最初の実験は CLI と統合版モニタの比較である。順序としては、まず CLI の場合を測定し、その後休憩を挟んで統合版モニタを使った場合を測定する。なお、仮想サーバへの負荷のパターンとしては 2 種類を用意しておき、2 つの実験の間に切り替えることで、ボトルネック箇所を覚えておくといった慣れによる時間短縮効果を打ち消し、純粋にツールの有無による時間短縮効果を測定することを狙う。

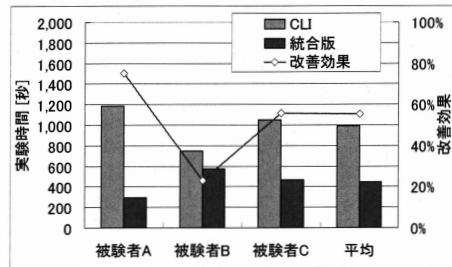


図 4-2：CLI と統合版モニタの実験時間比較

図 4-2 に 3 人の被験者に対して CLI と統合版モニタによる実験を行い、正味の実験時間（コマンド待ち等の時間を除いた時間）で比較したグラフを示す。コマンドラインによる性能取得作業に不慣れだった被験者 A が最も効果が大きく、75%の時間短縮効果が得られている。被験者 B は、予め 3 つのターミナルコンソールを立ち上げて並行にコマンドを打ち込むという方法を採用しており、CLI の場合も短い時間で完了しているために効果が小さく見えているが、それでも 25%程度の時間短縮効果が得られている。3 人の平均では約 55%の時間短縮効果が得られた。

4.4 非統合版モニタと統合版モニタの比較

前節の実験では、CLI と GUI というインタフェースの異なる手段の比較となっており、必ずしも統合化した効果を純粋に測定したことはなっていない。そこで、次の実験では同じ GUI ベースのモニタリングツールを用意し、統合化されたモニタリングツール(Virtage Monitor) と、仮想サーバ・HVM・ストレージそれぞれに個別にモニタリングツールがある場合とを比較し、統合化による効果抽出して評価することにした。なお、本実験に当たっては、最終的に表示されるグラフは同じであるため、2 回実験を行った場合後に行ったほうが慣れの効果がある分有利と考えられる。そこで、被験者ごとに統合版と非統合版のどちらを先に行うかを変えることで、慣れによる効果を除いて比較を行った。

図 4-3 に非統合版モニタのトップ画面を示す。非統合版モニタでは、仮想サーバ、ハイバイザ(HVM)、ストレージの各モニタ画面へのリンクが用意されてお

り、特定の LPAR に関するパフォーマンス情報を一覧して見ることはできない。

Monitoring Top

- Virtual Servers
- Hypervisor (HVM)
- Storage

図 4-3：非統合版モニタのトップ画面

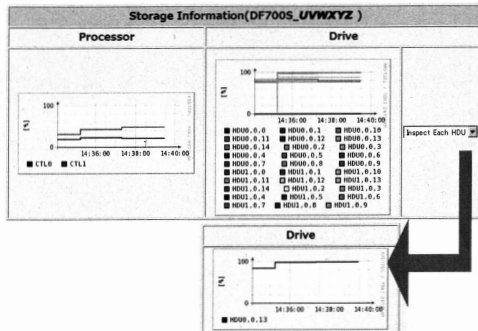


図 4-4：非統合版モニタでのストレージモニタ画面

図 4-4 に非統合版モニタにおけるストレージモニタ画面を示す。特定の LPAR との関連付けがないため、ストレージ上に存在するコントローラおよび物理ドライブ全てのパフォーマンス情報が一度に表示される。右側のタブを選択することで、ドライブ 1 つずつのグラフも表示することができるようになっている。被験者は、LPAR と物理ドライブとの関連付けを手動で行いながら、各 LPAR がストレージネックとなっているかどうかを検証する必要がある。

本評価に当たっては、Windows システムのイベントのモニタリングを行うエージェント BMI[9]をインストールした PC 上で実験を行い、実験にかかった時間だけでなく、マウスクリック回数やウィンドウの切替回数といった、GUI 操作に関連するイベントについても定量的な比較を試みた。

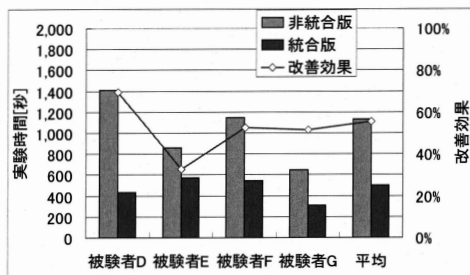


図 4-5：非統合版と統合版モニタの実験時間比較

4 人の被験者に対して非統合版モニタと統合版モニタによる実験を行い、正味の実験時間を比較したグラフを図 4-5 に示す。最大で 70%、最小でも 30%程度の

改善効果が見られる。実験の順序を入れ替えても常に統合版の方が短い実験時間で終了していることがわかる。平均では約 55%の改善効果が見られており、これは CLI 版と比較した場合の改善効果とほぼ同じである。このことから、リソースボトルネック検出の時間短縮にはユーザインタフェースの違い(CLI か GUI か)よりは、統合化されているかどうかの効果の方が大きいと言える。

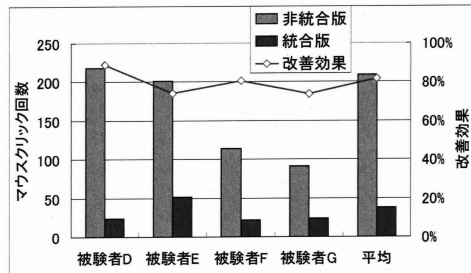


図 4-6：マウスクリック回数による比較

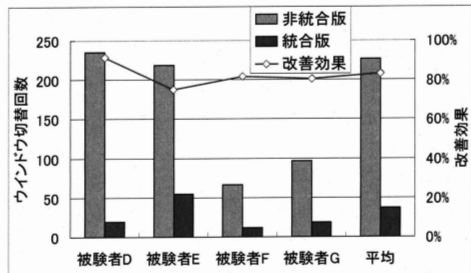


図 4-7：ウィンドウ切替回数による比較

図 4-6 と図 4-7 に、BMI から得たマウスクリック回数の合計およびウィンドウ切替回数の合計を比較したグラフを示す。なお、被験者の中にはマウスだけでなくキーボードショートカットを利用した者もいたため、キーボードの押下回数もマウスクリック回数として加算している。目的のグラフを一瞥しただけですぐに閉じるタイプの被験者や、ウィンドウを多数並べて比較しながら解析を進めるタイプの被験者など、被験者ごとに解析のスタイルが異なるため、回数自体は被験者によりバラつきがある。しかし全体として非統合版に比べて統合版は 80%前後操作回数が減っている。このことは、統合化によって、関連する情報を一覧できるようになったことで、複数ウィンドウ間の切替操作が減ったことにより、改善効果が得られたためと考えられる。

4.5 考察

図 4-8 に、仮想サーバ・HVM・ストレージの各項目別にかかった時間の内訳を比較したグラフを示す。CLI は 3 人、非統合版は 4 人、統合版は 7 人の平均である。ただし、統合版モニタでは 3 種類の情報が Web 画面で一度に表示され、個々の項目にかかった時間を厳密に分離することは困難であるため、ここでは単純

に 1/3 ずつかかったと仮定している。この結果より、CLI・非統合版のいずれと比較しても、統合化による効果大きいのは、ストレージであることがわかる。これは、統合化されていないモニタの場合、ストレージのパフォーマンス情報として得られたコントローラ・プロセッサのビジネ率および物理ドライブのビジネ率を、各仮想サーバ(LPAR)とストレージの対応関係を元に手動で対応させる作業が発生するためである。ストレージが出力するパフォーマンス情報は仮想サーバ(LPAR)基点にはなっていないため、ある LPAR に着目した時にその LPAR が使用するストレージがビジネ率であるかどうかを瞬時に判断するのは難しい。統合版モニタでは図 4-1 に示すように、LPAR が使用するコントローラ・物理ドライブのビジネ率のみを抽出して表示することで、LPAR ごとの動作状況把握がより容易になるようにしたが、その効果が本実験により確認できたと言える。

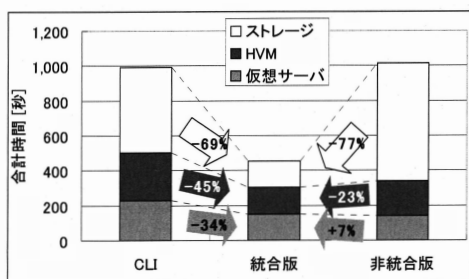


図 4-8: モニタ項目別内訳時間の比較

仮想サーバに関しては、非統合版と比べて統合版の方が+7%とわずかに時間が延びている。この程度の差は誤差の範囲とも言えるが、非統合版モニタの方が一度に表示されるグラフの数が少ないため、目的の情報を探しやすい、といった理由も考えられる。このように目的が比較的単純で見べき情報が限定されている場合は、必ずしも複数の情報を統合化して表示する必要は無い。しかし一方で、実際のボトルネック解析においては、メモリの使用状況を見てスワップが発生しだすと、ストレージの利用率が上がるといった複数コンポーネント間の相関関係を把握する必要がある場面もある。従って、実際の運用管理の現場に適用するに当たっては、作業する SE の作業内容に応じて、必要とする情報を抽出して表示できるようなカスタマイズができることが望ましい。

5. 関連研究

(1)複数コンポーネント間の統合モニタリングに関して、仮想化サーバおよびストレージの統合監視・性能監視を行うツールとして、Akkori 社の BalancePoint[10]がある。特にストレージとサーバの連携に力を入れており、Oracle や MS SQL といった DB のインスタンスやスキーマ構成の可視化や DB パフォーマンス最適化、ストレージ仮想化を通じた I/O データパスを追尾する BalancePoint for Storage Virtualization を提供している。一般にストレージの性能ボトルネッ

クをサーバ側の振る舞いだけから見つけることは困難であり、Akkori 社のようにサーバとストレージの紐付けを行った上で性能値を出すというアプローチは有効であると考えられる。一方、ネットワークとサーバの統合運用として、仮想サーバや VLAN を含めた仮想と物理のネットワーク構成を可視化・設計を支援するツールの研究として、EasyLayering[11,12,13,14]がある。しかしサーバ・ストレージ・ネットワークの全てを統合的・横断的に管理可能なツールはまだ無く、今後の課題である。

(2)仮想リソースと物理リソースの統合管理に関連して、VMware vCenter Server では、仮想マシンおよび物理ホストの統計ログをレポートする機能を備えている。また vSphere 4[15]ではデータセンタ・クラスレベルの仮想化により、仮想化されたリソースをリソースプールとして管理できる機能が追加された。一方、日立製作所は同社のサーバ仮想化機構 Virtage 向けの運用管理ツール Virtage Navigator[16]の発売を開始し、LPAR のモニタリングを可能としている。また、NTT データが提供するオープンソースソフトウェアの Hinemos[17]は、複数のサーバ群を統合運用管理するためのツールである。Hinemos は有償の VM 管理オプションを提供しており、仮想化システムに特化した監視・管理・操作を可能にするとしている。しかし本研究のように物理リソースと仮想リソースの対応関係を素早く切り替えられるようにするアプローチを採用している製品はまだ少ない。

(3)複数種類の仮想化ソフトウェア間の統合管理に関連して、日立製作所は JP1/IT リソース管理[18]の発売を予定している。今後は、Cisco/VMware 社の Unified Computing System[19]のようにサーバからネットワーク、ストレージまで垂直統合で管理するアプローチと、複数ベンダの機器を統合的に管理できるようにするマルチベンダ対応のアプローチと、両方のアプローチがそれぞれ発展していくと予想される。本研究で提案した統合モニタリングツールは、マルチベンダ・マルチコンポーネントの統合運用管理を実現するツール群の一つとして位置づけられる。

その他、運用管理に関連した研究としては、クラウドコンピューティング実現に向けた要件やコストの定量化を行った[20]や、複数の稼働情報やログを Signature という形で可視化することでサービスレベルや障害の判定を容易化する[21]などがある。運用管理ツールの開発や改善効果の見積りに当たっては、運用管理コストの定量化が鍵となるため、本研究のような SE 作業工数を客観的指標で定量化する試みは今後重要性が増すと考えられる。

6. まとめと今後の課題

6.1 まとめ

仮想化環境におけるシステム稼働状況把握の迅速化・導入運用容易化に向け、構成情報と連動した物理/仮想およびサーバ/ストレージ統合可視化を実現する統合モニタリングツール Virtage Monitor を提案し、プロトタイプを開発して実験・評価を行った。実験の結果、Virtage Monitor を使用した場合、仮想サーバのリ

ソースポトルネック検出までの時間がCLIや統合化されてないモニタと比較して最大で70~75%短縮された。また、ウインドウ操作の回数も平均で80%削減された。本実験を通して、物理リソースにおける仮想サーバのビジー率(物理-仮想統合化)や、仮想サーバが使用するストレージのビジー率(サーバストレージ統合化)といった、構成情報と連携した一元可視化が、仮想化環境における稼動状況把握には特に有効であることが確かめられた。

6.2 今後の課題

今回のプロトタイプは統合モニタリングに向けた第一のステップである。今後は(1)モニタリング結果を使つての、仮想化サイジング提案への貢献、(2)モニタリング結果を、構成変更へと自動的にフィードバックする機能、(3)ネットワーク情報取得(特にVLANや仮想NICなど)およびネットワーク仮想化への対応、などの開発が必要と考えられる。

謝辞 本稿執筆に当たり、実験にご協力頂いた皆様に、謹んで感謝の意を表する。

参考文献

- 1) “戦略的 IT イニシアチブ: SQL Server の TCO における優位性”, NerveWire, Inc., May2001.
- 2) 上原 他, “スケールアウトとスケールアップを両立するブレードサーバ向けアーキテクチャの提案と評価”, 情報処理学会 第17回システム評価研究会(SIGEVA17) 研究報告, pp.35-40, 2006年6月.
- 3) 上野 他, “情報システムの運用効率を向上する「BladeSymphony」のサーバ仮想化機構「Virtage」”, 日立評論 Vol.89 No.07, pp.52-57, 2007年7月.
- 4) “仮想化技術による BladeSymphony の進化”, 日立ホワイトペーパー, 2008年.
- 5) Tanaka, T. et al., “Investigating Suitability for Server Virtualization using Business Application Benchmarks”, In Proceedings of the 3rd international workshop on Virtualization technologies in distributed computing, VTDC 2009, ACM, pp.43-50.
- 6) “国内バーチャライゼーションソフトウェア市場 2008年の分析と2009~2013年の予測”, IDC Report #J9360102, 2009年3月.
- 7) Standard Performance Evaluation Corporation, <http://www.spec.org/>
- 8) IOzone Filesystem Benchmark, <http://www.iozone.org/>
- 9) “PC業務効率分析システム BMI”, 日立システムホームページ, <http://www.hitachi-system.co.jp/bmi/>
- 10) Akorri BalancePoint home page, Akorri, <http://www.akorri.com/>
- 11) 吉澤 他, “サーバ仮想化環境における管理コストを低減するネットワーク管理システムの実装および評価”, 電子情報通信学会ネットワークシステム研究会 2009年11月研究会研究報告, 2009年11月.
- 12) 沖田 他, “データセンター向け仮想ネットワーク構成情報収集機能の開発と評価”, 電子情報通信学会 ICM 研究会 2010年7月研究報告, 2010年7月.
- 13) Okita, H. et al., “Proposal of Virtual Network Configuration Acquisition Function for Data Center Operations and Management System”, In Proceedings of the 5th Workshop on Virtualization in High-Performance Cloud Computing (VHPC '10), ICAR.
- 14) Yoshizawa, M. et al., “Implementation and Evaluation of Network Management System to Reduce Management Cost Caused by Server Virtualization”, In Proceedings of the 2nd Workshop on Data Center - Converged and Virtual Ethernet Switching (DC CAVES 2010).
- 15) “VMware vSphere の概要”, VMware Inc., 2009年, <http://www.vmware.com/>
- 16) “Virtage Navigator”, 日立ホームページ, 2010年, http://www.hitachi.co.jp/products/bladesymphony/virtual/virtage_navi.html
- 17) Hinemos home page, NTT データ, <http://www.hinemos.info/>
- 18) “統合システム運用管理 JP1:IT リソース管理”, 日立ホームページ, 2010年, <http://www.hitachi.co.jp/Prod/comp/soft1/jp1/product/merits/itrm/index.html>
- 19) “A Platform Built for Server Virtualization: Cisco Unified Computing System”, Cisco/VMware White Paper, 2009.
- 20) Armbrust, M. et al., “Above the clouds: A Berkeley view of Cloud computing”, Technical report UCB/ECS-2009-28, Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, USA, February 2009.
- 21) Cohen, I. et al., “Capturing, Indexing, Clustering and Retrieving System History”, Proceedings of the 20th ACM symposium on Operating systems principles, pp. 105-118.