情報処理学会論文誌 Vol.54 No.8 2071–2078 (Aug. 2013)

推薦論文

大規模データセンタ向け運用管理ソフトウェアにおける情報取得方式の決定手法

森 宣仁[1,a)] 原 純一[1] 坂下 幸徳[1] 牧 晋広[1]

受付日 2012年12月15日, 採録日 2013年5月18日

概要：近年，コストの低減を目的としたデータセンタ集約の進展にともなってデータセンタ内のストレージ，サーバ，アプリケーションといったリソースの数が増大，データセンタが大規模化している．一方で，管理ソフトウェアがこれらの各種リソースを関連付けて管理し，性能・コストを最適化する技術が登場している．大規模データセンタにおいてこのようにリソースを関連付けて管理する場合，リソースからの情報取得時間が増大してしまい実運用に耐えないため，情報取得時間を短縮する各種方式が研究されてきた．しかし，これらの方式は時間を短縮できる反面，リソースに負荷を与える，管理ソフトウェアに負荷を与えるといった側面があり，どの方式を用いればよいかを一概に決定することはできなかった．そこで本論文では，方式選択にあたって考慮すべき要件と各方式を特徴付けるパラメータの関係性を明確化することで，管理対象に応じて，適切な方式およびそのパラメータを決定する手法を提案する．この手法により，大規模データセンタ向け管理ソフトウェアにおける情報取得の設計・設定を体系的に実施することが可能となった．

キーワード：データセンタ，運用，管理，大規模，高速化

A Method for Selecting Information Acquisition Methods of Management Software for Huge Data Centers

Nobuhito Mori[1,a)] Junichi Hara[1] Yukinori Sakashita[1] Nobuhiro Maki[1]

Received: December 15, 2012, Accepted: May 18, 2013

Abstract: In recent years, the number of resources such as storages, servers and applications in a data center is increasing since consolidation of datacenters is under way in order to reduce cost. On the other hand, a new technology is developed that management software integrally manages and associates these resources in order to optimize performance and cost. Since acquiring information of resources takes too long in huge data centers, several methods to shorten time for acquiring information were researched. However, each information acquisition method has demerits that loads of resources or management software itself increase. Therefore, it was difficult to indiscriminately determine which information acquisition method to utilize. In this paper, we propose a determination method to determine one of information acquisition methods according to managed environments by clarifying relationship between parameters which characterize information acquisition methods and requirements to consider in determination. By this method, it is possible to systematically design or configure management software for huge data centers.

Keywords: datacenter, operation, management, large-scale, speed-up

1. はじめに

近年，業務のIT化や各種コンテンツのデジタル化にともない，データセンタの規模が拡大している．また，多数のデータセンタを所有・運用してきた企業が，管理コスト

[1] 株式会社日立製作所
Hitachi Ltd., Yokohama, Kanagawa 244–0817, Japan
[a)] nobuhito.mori.dn@hitachi.com

本論文の内容は2012年7月のマルチメディア，分散，協調とモバイル（DICOMO2012）シンポジウム2012にて報告され，インターネットと運用技術研究会主査により情報処理学会論文誌ジャーナルへの掲載が推薦された論文である．

© 2013 Information Processing Society of Japan

等のコスト低減を目的としてデータセンタを集約する動きが広がっている [1], [2]．それにともない，1 つのデータセンタ内におけるストレージや，ストレージに接続された物理/仮想サーバ，およびそれらの上で稼働するアプリケーションといったリソースの数が増大している．

一方で，ストレージ，サーバ，アプリケーションといった各種リソースを関連付けて管理し，性能・コストを最適化する技術が登場している [3], [4], [5]．このような管理においては，管理ソフトウェアが各種リソースから情報を取得し，それらの情報を突き合わせて評価し，その結果を各種リソースの管理にフィードバックすることで最適化を図る（**図 1**）．この一例として，データベース（DB）とストレージの情報を用いることで，DB の表・索引等のオブジェクト単位で，ストレージ内の性能・コストの異なる各種記憶メディアへのデータ配置を制御し，性能・コストを最適化する DB オブジェクト階層制御機能があげられる [6]．

しかし，特に大規模データセンタでは，このようにストレージやアプリケーション等の各種リソースを関連付けて管理する場合，稼働するリソース数が多いため情報取得において取得時間が増大してしまう．このため，情報取得が完了して評価する時点で，初期に取得した情報が陳腐化してしまう等実運用に耐えない．そこで，情報取得時間を短縮するために各種の方式が研究されてきた（後述）[7], [8]．しかし，各方式は時間を短縮できる反面，リソースに負荷を与える，管理ソフトウェアに負荷を与えるといった側面もあるため，どの方式を用いればよいかを一概に決定することはできなかった．

そこで本論文では，方式選択にあたって考慮すべき要件と各方式を特徴付けるパラメータの関係性を明確化することで，管理対象に応じて，適切な方式およびパラメータを体系的に決定する手法を提案する．また，提案手法を実際に DB オブジェクト階層制御機能における DB からの情報取得に適用して評価を行う．

2. 本研究の位置付け

2.1 従来研究

管理ソフトウェアの情報取得時間の短縮のための 3 つの方式を以下に述べる [7], [8]．

(1) 多重コネクション方式

図 2 に多重コネクション方式の概要図を示す．本方式においては，管理ソフトウェアが 1 つのリソースに複数のコネクションを張ることにより，リソースからの情報取得を分割し，並列に取得する．これにより単一コネクションで情報取得する場合に比べて取得時間を短縮することが可能である．しかし，取得時間の短縮効果がある一方，リソースに対して複数のコネクションを張るため，1 本だけコネクションを張る場合に比べてリソースの負荷（CPU やメモリの消費）が増大するというデメリットがある．

(2) 多段並列方式

図 3 に多段並列方式の概要図を示す．本方式においては，管理ソフトウェアから情報取得モジュールを別モジュールとして分離し，多数のリソースから同時に情報取得を行う．これにより，逐次的にリソースから情報取得を実行した場合に比べて取得時間を短縮することが可能である．しかし，取得時間の短縮効果がある一方，多数のリソースからの情報を同時に処理しなくてはならないため，管理ソフトウェア自身の負荷が増大するデメリットがある．

なお，多数のリソースから同時に情報取得を行うために情報取得モジュールを分離することは必須ではなく，管理ソフトウェア上で多数のスレッドを起動しても同様の効果を得られる．ただしその場合，管理ソフトウェアの処理能力は 1 台の管理用マシン（PC）の備えるメモリ・CPU 等の性能に制限されるため，スケーラビリティに限界がある．一方，情報取得モジュールを分離した場合，リソース数が増加して 1 台の管理用マシンで対応しきれなくなったとしても，新たな管理用マシンおよび情報取得モジュールを追

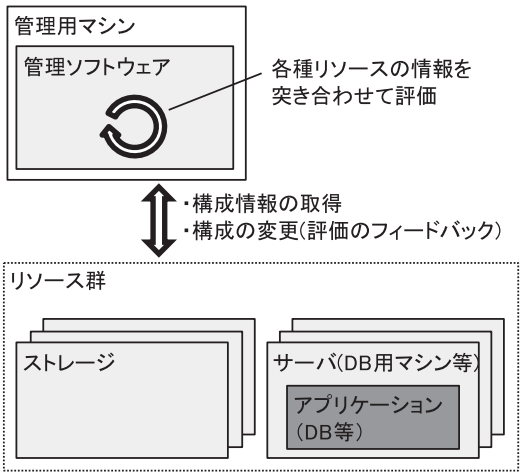

図 1 管理ソフトウェアの概要図
Fig. 1 Schematic diagram of management software.

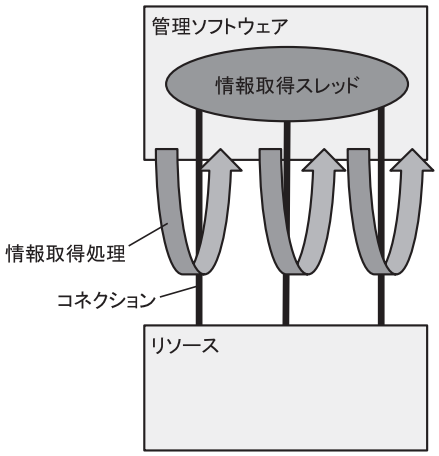

図 2 多重コネクション方式
Fig. 2 Multi-connection method.

© 2013 Information Processing Society of Japan

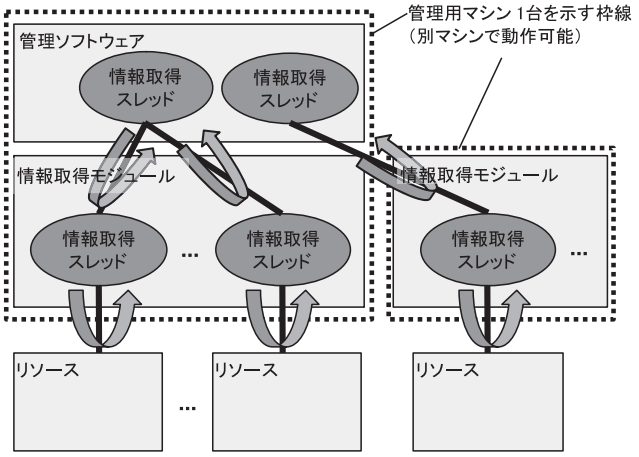

図 3 多段並列方式
Fig. 3 Multi-step parallel method.

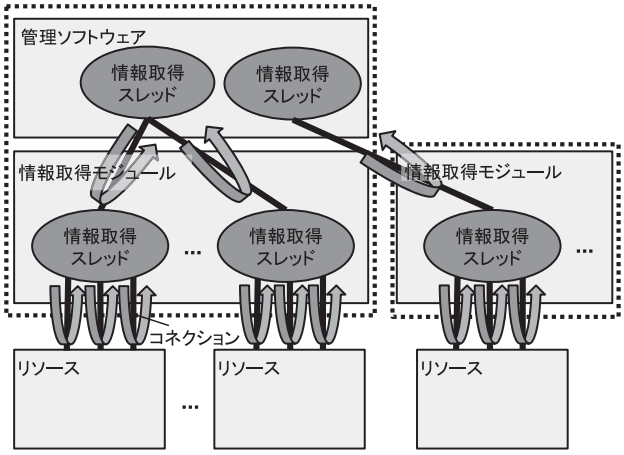

図 4 ハイブリッド方式
Fig. 4 Hybrid method.

加することでスケールアウトが可能であり，大量のリソースを管理する管理ソフトウェアの設計として適している．

(3) ハイブリッド方式

図 4 にハイブリッド方式の概要図を示す．本方式は，多重コネクション方式と多段並列方式を組み合わせた方式である．すなわち，多数のリソースから同時に情報取得を行うのに加え，1 つのリソースに対してコネクションを複数張ることで情報取得を分割・並列実行する．これにより，多重コネクション方式と多段並列方式を単体で用いた場合より，さらに取得時間を短縮することが可能である．しかし，取得時間の短縮効果の反面，リソースの負荷，管理ソフトウェア自身の負荷が増大するというデメリットがある．

2.2 要件

以下に，大規模データセンタでリソースを関連付けて管理する際の情報取得に関する 4 つの要件をまとめる．

要件 (1) 個々のリソースに与える負荷
ストレージ，ホスト，アプリケーションといったリソースは，サービスに直接的に関係するため，負荷がかかって性能劣化が発生するとサービスのレスポンス・タイムの悪化等に直結する．このため，情報取得において個々のリソースに与える負荷を一定以下に抑える必要がある．

要件 (2) 管理ソフトウェア自身の負荷
情報取得において管理ソフトウェアに負荷を与えると，ストレージのボリューム作成やホストへのボリューム割当てといった他の管理操作のレスポンス・タイム悪化につながる．このため，管理ソフトウェア自身の負荷を一定以下に抑える必要がある．

要件 (3) まとめて情報取得すべきリソースの範囲
リソースを関連付けて管理する場合，全体最適化を図るためには関連する全リソースの情報を考慮する必要がある．

要件 (4) 全リソースの情報取得に必要な時間
関連するリソースの情報取得時刻にずれがあると，各リソースの情報の間で整合性が保てなくなり，全体最適化を図ることができなくなる．このため，取得時間を一定以下に抑える必要がある．

2.3 課題

2.1 節で述べたように，多重コネクション方式，多段並列方式，ハイブリッド方式の 3 つの方式は，取得時間を短縮する効果がある反面，デメリットもある．このため，管理ソフトウェアにおいて使用する方式を一概に決定することはできず，決定にあたっては大規模データセンタでリソースを関連付けて管理する際の 4 つの要件（2.2 節）を考慮して決定する必要がある．

しかし従来は，この決定にあたっての体系的な手法が確立されていなかった．このため，各々の方式においてパラメータを様々に変化させ，要件を満たす方式およびそのパラメータを決定しなくてはならなかった．

3. 情報取得方式およびパラメータ決定手法

本章では，2.3 節で述べた課題に対し，4 つの要件と各方式のパラメータであるコネクション数と並列数の関係性を明確化し，管理対象に応じて適切な情報取得方式およびパラメータを決定する手法を述べる．

3.1 全要件を満たすパラメータ空間の決定方法

本節では，無数に存在するコネクション数と並列数の組合せ（パラメータ空間）の中から，全要件を満たすものを決定する方法について述べる．

まず，図 5 に各方式のパラメータの定義を示す．図 5 (a) に示すように，多重コネクション方式においては，管理ソフトウェアからリソースに対して張るコネクションの数（コネクション数）N_{conn} がパラメータとなる．また図 5 (b) に示すように，多段並列方式においては，同時に情報取得

© 2013 Information Processing Society of Japan

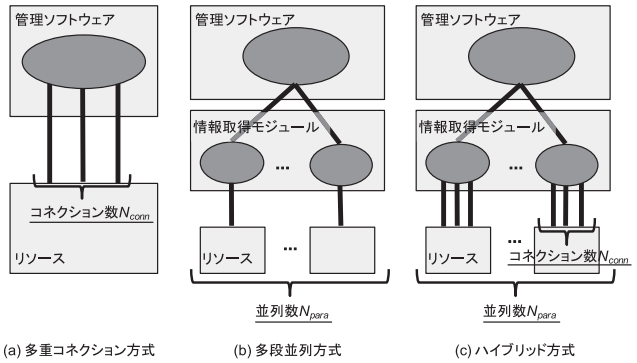

図 5 各方式のパラメータ

Fig. 5 Parameters of each method.

を行うリソースの数（並列数）N_{para} がパラメータとなる．また，ハイブリッド方式においては図 5 (c) に示すようにコネクション数 N_{conn} および並列数 N_{para} の両方がパラメータとなる．

次に，各方式のパラメータ N_{conn} および N_{para} と，各要件の関係を数式で表現する．まず要件 (1) に関して，一般に個々のリソースの負荷 α はコネクション数 N_{conn} を増やすほど大きくなると考えられるため，ある単調増加関数 $f(x)$ を使って，以下のように表すことができる．

$$\alpha = f(N_{conn}) \tag{1}$$

ここで，要件として許容されるリソースの負荷の最大値 α_{max} を考慮すると，要件 (1) は，$f(x)$ の逆関数 $f^{-1}(x)$ によって計算される最大コネクション数 $N_{max\text{-}conn}$ を使って以下のように表すことができる．

$$N_{conn} \leq N_{max\text{-}conn} = f^{-1}(\alpha_{max}) \tag{2}$$

同様に，要件 (2) に関して，一般に管理ソフトウェアの負荷 β は並列数 N_{para} を増やすほど大きくなると考えられる．これより，要件 (2) はある単調増加関数 $g(x)$，要件として許容される管理ソフトウェアの負荷の最大値 β_{max} を使って，以下のように表すことができる．

$$\beta = g(N_{para}) \tag{3}$$

$$N_{para} \leq N_{max\text{-}para} = g^{-1}(\beta_{max}) \tag{4}$$

また，要件 (3) および要件 (4) は「まとめて情報取得すべき N_{res} 個のリソースからの情報取得を目標時間 T_{obj} 以内に完了する」ことにほかならない．一般に，N_{res} 個のリソースからの情報取得時間 T は，コネクション数 N_{conn} と並列数 N_{para} を増やすほど短縮されると考えられる．これは，N_{conn} が大きいほど，1 リソースからの取得時間 t が短縮され，N_{para} が大きいほど全体の取得時間が短縮されるためである．以上のことから，要件 (3) および要件 (4) は，ある単調減少関数 $h(x)$ を用いて以下のように表すことができる．

$$T = t \cdot \left\lceil \frac{N_{res}}{N_{para}} \right\rceil = h(N_{conn}) \cdot \left\lceil \frac{N_{res}}{N_{para}} \right\rceil \leq T_{obj} \tag{5}$$

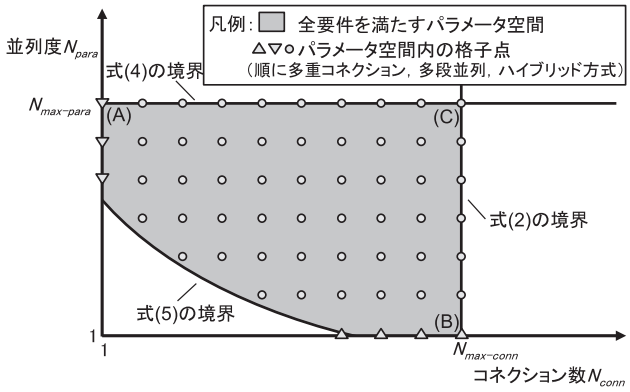

図 6 要件を満たすパラメータ空間の概要図

Fig. 6 Schematic diagram of a parameter-space which satisfies all requirements.

ただし，ここで $t = h(N_{conn})$ である．また，$\lceil x \rceil$ は天井関数であり，x 以上の最小の整数を表す．

図 6 に式 (2)，(4)，(5) をコネクション数 N_{conn}-並列数 N_{para} 平面上に図示したものを示す．なお，図 6 はあくまで概要図であり，実際のデータを反映したものではない．また式 (5) の表す曲線は，本来は天井関数の影響により階段状になるが，図 6 ではその効果は省略して滑らかな曲線として表現している．実際には式 (2)，(4)，(5) における未知の関数 $f(N_{conn})$，$g(N_{para})$，$h(N_{conn})$ を測定により決定することでパラメータ空間を決定することができる．なお，式 (2)，(4) に関しては，4 章で述べるが，測定結果から $N_{max\text{-}conn}$ および $N_{max\text{-}para}$ を容易に決定できるため，$f(N_{conn})$，$g(N_{para})$ を明に求める必要はない．

ここで重要な点は，$f(N_{conn})$，$g(N_{para})$，$h(N_{conn})$ はそれぞれ N_{conn} または N_{para} の一方のみに依存していることから，両パラメータの組合せを変化させて測定を実施する必要はなく，個別に測定を実施すればよいということである．また，図 6 に示すように，並列数 $N_{para} = 1$ の空間が多重コネクション方式，コネクション数 $N_{conn} = 1$ の空間が多段並列方式，$N_{conn} > 1$ かつ $N_{para} > 1$ の空間がハイブリッド方式に対応する．

3.2 情報取得方式およびパラメータの決定ガイドライン

本節では，3.1 節で決定したパラメータ空間に基づいて，情報取得方式およびパラメータを決定する方法について述べる．

情報取得方式およびパラメータを決定するということは，パラメータ空間内の格子点から 1 点を選択することと同義である．要件間に優先度がなく，要件を満たせば十分なケースでは，パラメータ空間から適当な 1 点を選択すればよい．しかし，実際には各要件に優先度がある場合が多く，このような場合，要件の優先度を考慮して格子点を決定する必要がある．以下に例を述べる．

- 業務要件が厳しい等，リソースに負荷を与えないこと

© 2013 Information Processing Society of Japan

（要件 (1)）が最優先の場合，コネクション数 N_{conn} が最小となる図 6 の格子点 (A) を選択する．
- 管理用マシンの性能が限られている等，管理ソフトウェアに負荷を与えないこと（要件 (2)）が最優先の場合，並列数 N_{para} が最小となる図 6 の格子点 (B) を選択する．
- 将来的なリソース数の増大に備える等，取得時間の短さ（要件 (3)，要件 (4)）が最優先の場合，コネクション数 N_{conn} と並列数 N_{para} がともに最大となる図 6 の格子点 (C) を選択する．

以上の例を一般的なガイドラインとしてまとめると，要件間の優先度に応じて，要件 (1) を優先する場合はコネクション数 N_{conn} を小さく，要件 (2) を優先する場合は並列数 N_{para} を小さく，要件 (3) および要件 (4) を優先する場合はコネクション数 N_{conn} と並列数 N_{para} を大きくとればよい．

4. 実験および考察

本章では，3 章で述べた手法を用いて，大規模データセンタにおける DB オブジェクト階層制御機能で DB からの情報取得時間を短縮する方式およびそのパラメータを決定し，その評価を行う．

4.1 DB オブジェクト階層制御機能の概要

DB オブジェクト階層制御機能は，DB と，DB の表や索引といったオブジェクトを格納するストレージを対象とした機能である．

近年，ストレージには，ストレージ内のデータを細粒度（数十 MB 程度）に分割したうえで，アクセス頻度に応じて，高速な記憶メディアと低速な記憶メディアに自動的に配置する自動階層制御機能が搭載されている．しかし，自動階層制御機能においては，アクセス頻度のみによりデータの配置が決定されてしまうため，業務上は高応答性能が必要なデータが低速な記憶メディアに配置されるケースがある．この課題に対し，DB オブジェクト階層制御機能は，業務上の重要度に応じてデータの配置を行うことを目的としたものである．その具体的な処理としては，管理ソフトウェアが DB からオブジェクトの格納位置情報を取得し，その情報に基づいてストレージに適切な配置先の記憶メディアを指示する．ここで，一般的なシステム構成では複数の DB が 1 台のストレージを共用しているため，取得する情報の整合性を保つためには，これら複数の DB の情報を所定時間内に取得する必要がある．所定時間とは，具体的にはストレージの自動階層制御機能によるデータの再配置の周期である．しかし，大規模な環境では DB が多数となり，情報取得に要する時間が増大してしまう．このため，本実験においては多数の DB からの情報取得に注目し，手法の適用を行う．

4.2 DB オブジェクト階層制御機能における要件

本節では，DB オブジェクト階層制御機能の要件について述べる．以下の要件は 2.2 節で述べた 4 つの要件に順に対応している．

要件 (i) 個々の DB の性能低下率 α を 5%以内とする（$\alpha_{max} = 5\%$）．

要件 (ii) 管理ソフトウェア（情報取得モジュールを含む）が DB オブジェクト階層制御機能向けに使用するメモリ消費量 β を 512 MB 以内とする（$\beta_{max} = 512\,\mathrm{MB}$）．

要件 (iii) 100 台の DB の情報をまとめて取得する（$N_{res} = 100$）．

要件 (iv) 30 分以内に情報取得を完了する（$T_{obj} = 30$ 分）．

以上の要件で述べた具体的な数値は，あくまで本実験を行うために定めた一例であり，環境に応じて変更してよい．なお，要件 (ii) について，本実験では管理ソフトウェアと情報取得モジュールを同一の管理用マシン上で稼働させたため，メモリ消費量 β は管理ソフトウェアと情報取得モジュールによるメモリ消費量を合算している．このため，本章では以降，管理ソフトウェアと情報取得モジュールを区別せず，両者を合わせて単に管理ソフトウェアと記載する．また，DB オブジェクト階層制御機能も含め，管理ソフトウェアの処理はリソースからの情報取得，情報に基づいた計算，リソースへの設定指示からなるが，本論文および本実験ではこのうち情報取得を，情報に不整合が発生しない時間内に完了することを目的とする．このため，要件 (iv) は単純に，取得する情報の整合性が保証できる時間であり，計算および設定指示に要する時間は含まない．しかし，情報取得・計算・設定指示という機能全体として管理ソフトウェアの処理を動作させる際には，その総所要時間に制限がある場合もある．この場合は総所要時間の制限も考慮して別途要件を設定する必要がある．

4.3 測定環境

測定環境は**表 1** に示すとおりである．なお，本測定環境においては，DB が図 1〜図 4 等に示す「リソース」に相当し，これは DB 用マシン上で稼働している．なお DB 用マシンそのものも，通常は管理ソフトウェアが管理するリソースだが，本実験においては情報取得を行う対象ではない．また，管理用マシン上で管理ソフトウェアが稼動しており，先述のとおり，ここには情報取得モジュールを含む．

4.4 DB 性能低下率とコネクション数の関係

図 7 に多重コネクション方式を用いて，コネクション数 N_{conn} を 1 から 6 まで変化させた場合の DB 性能低下率 α の測定グラフを示す．この測定結果から，要件 (i) の閾値 $\alpha_{max} = 5\%$を超えない最大コネクション数は $N_{max\text{-}conn} = 5$ と判明した．

© 2013 Information Processing Society of Japan

表 1 測定環境
Table 1 Measurement environment.

大項目	小項目	詳細
DB 用マシン	OS	Solaris 10 (08/07)
	CPU	UltraSPARC-T2 (1.2Ghz/4core, 32 threads)
	RAM	4GB
DB	データベース容量	100GB
	オブジェクト数	400
管理用マシン	OS	Windows XP Professional SP3
	CPU	Intel Core2 Duo 2.93GHz
	RAM	2GB

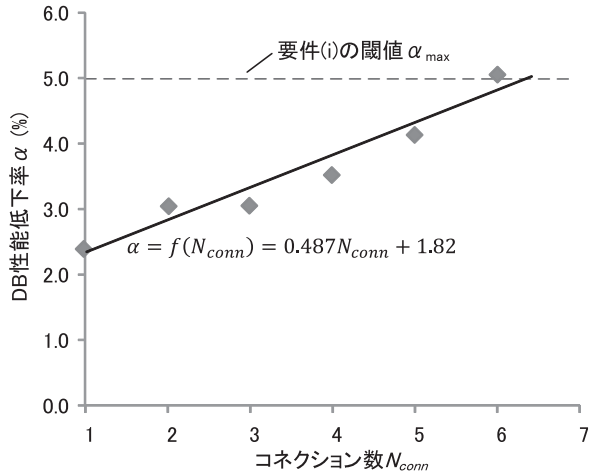

図 7 DB 性能低下率 α とコネクション数 N_{conn} の関係
Fig. 7 Decreasing rate of DB's performance α vs. Number of connections N_{conn}.

4.5 管理ソフトウェアのメモリ消費量と並列数の関係

図 8 に多段並列方式を用いて，並列数 N_{para} を 5 から 40 まで変化させた場合の管理ソフトウェアのメモリ消費量 β の測定グラフを示す．この測定結果から，要件 (ii) の閾値 $\beta_{max} = 512$ MB を超えない最大並列数は $N_{max\text{-}para} = 20$ と判明した．

4.6 情報取得時間とコネクション数の関係

図 9 に多重コネクション方式を用いて，コネクション数 N_{conn} を 1 から 6 まで変化させた場合の DB 1 台の取得時間 t の測定グラフを示す．この測定結果から，以下の近似式を得た．

$$t = h(N_{conn}) = 7.91{N_{conn}}^{-0.856} \tag{6}$$

4.7 パラメータ空間の決定と実測および考察

図 10 に測定結果から決定したパラメータ空間を示す．この結果から，本実験の測定環境においては，ハイブリッド方式のみが要件を満たす方式であることが分かった．

また，パラメータ空間内の 1 点を選択し，所要時間 T の実測を行った．ここでは DB への負荷（要件 (i)）を抑える観点から，コネクション数 $N_{conn} = 2$，並列数 $N_{para} = 20$ の点を用いた．この結果，$T = 27$ 分を要した．個別の測定結果から予測される値が 22 分であることから約 23%の誤差が生じている．この誤差要因を確認したところ，「管理

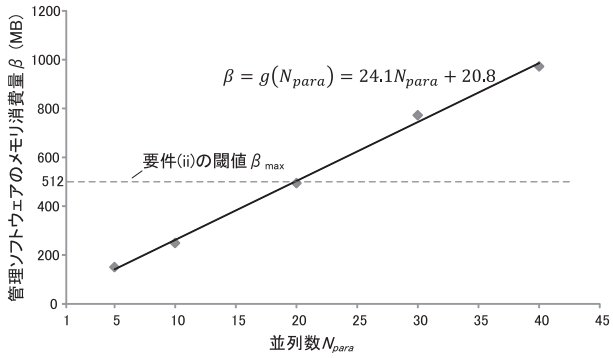

図 8 管理ソフトウェアのメモリ消費量 β と並列数 N_{para} の関係
Fig. 8 Memory consumption β vs. Number of parallelism N_{para}.

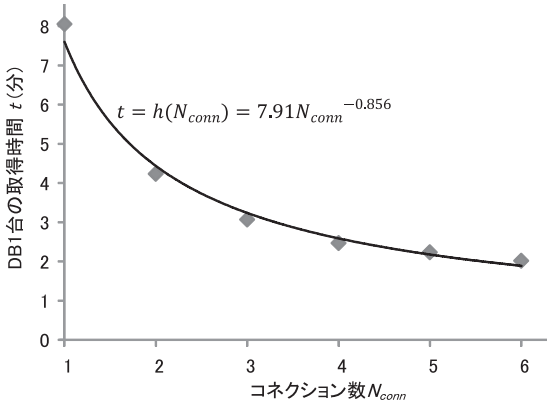

図 9 DB 1 台の取得時間 t とコネクション数 N_{conn} の関係
Fig. 9 Time for acquiring information t vs. Number of connections N_{conn}.

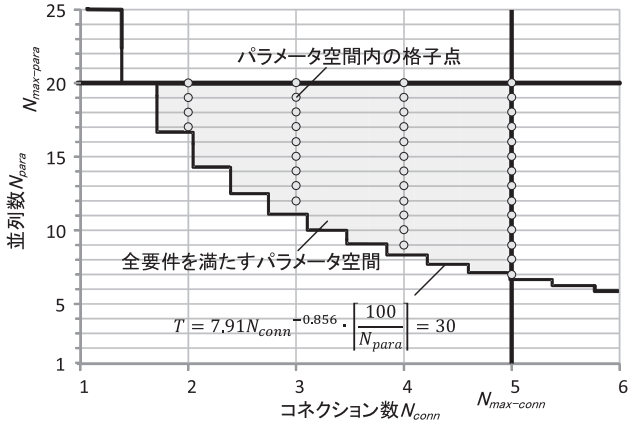

図 10 測定結果から決定したパラメータ空間
Fig. 10 Parameter-space determined by measurement.

© 2013 Information Processing Society of Japan

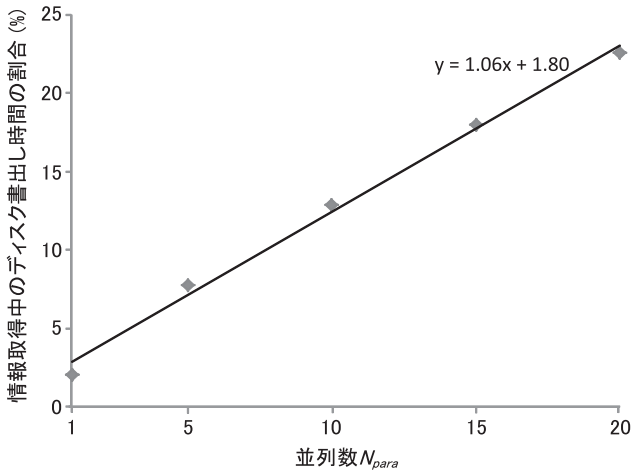

図 11 情報取得中のディスク書き出し時間の割合
Fig. 11 Percentage of time for writing a disk in the process of acquiring information.

ソフトウェアが，取得した情報を管理用マシンのディスクへ書き出す時間」が主要因であることが判明した．**図 11** に示すように，並列数 N_{para} が小さい場合，ディスクへの書き出し時間は十分無視できる大きさだが，並列数 N_{para} が増大するにつれて，情報取得時間に占めるディスクへの書き出し時間の割合も増大する．このため，より正確なパラメータ空間決定のためには，ディスクの書き出し時間を考慮する必要がある．

5. おわりに

本論文では，大規模データセンタにおいてストレージやアプリケーション等のリソースを関連付けて管理する際の情報取得に焦点を当て，従来研究されてきた 3 つの情報取得時間の短縮方式から，環境に応じて適した方式とそのパラメータを決定する手法およびガイドラインを提案した．この手法により，大規模データセンタ向け管理ソフトウェアの情報取得方式における設計・設定を体系的に実施することが可能となった．

ただし，実際に DB オブジェクト階層制御機能を対象として，提案した手法で決定したパラメータで実測した結果，情報取得に要する時間は，理論値に対して約 23%の誤差が生じた．この主要因は，管理ソフトウェアおよび情報取得モジュールが取得した情報を管理用マシンのディスクへ書き出す時間であることが判明した．このため，ディスクの書き出し時間を考慮した決定手法を確立することで，より正確なパラメータ空間決定を行うことができると考えられる．しかし一方で，近年，SSD（Solid State Drive）等の高性能の記憶メディアが一般的な PC にも使用されつつあり，従来のディスクに代えて，このような記憶メディアを用いることで誤差が小さくなることも想定される．このため今後は，近い将来，一般的に使用されるであろう管理用マシンのスペックにおいて，本論文の手法による誤差がどの程度縮小するかを評価したうえで，より正確な決定手法の確立を行いたい．

参考文献

[1] Gartner, Inc.: Best Practices in Data Center and Server Consolidation, Gartner Inc. (online), available from ⟨http://www.gartner.com/it/content/1548100/1548114/march_15_best_practices_in_data_center_mchuba_jphelps.pdf⟩ (accessed 2012-04-23).
[2] IDC Japan：国内データセンターサイト数の予測を発表，IDC Japan（オンライン），入手先 ⟨http://www.idcjapan.co.jp/Press/Current/20100113Apr.html⟩（参照 2011-11-20）．
[3] （株）日立製作所：ストレージ管理ソフトウェア Hitachi Command Suite 7 ストレージ仮想化運用のすすめ，（株）日立製作所（オンライン），入手先 ⟨http://www.hitachi.co.jp/Prod/comp/soft1/download/catalog/ca/ca874.pdf⟩（参照 2012-04-12）．
[4] Hewlett-Packard Development Company: HP P4000 Application-Aware Snapshot Manager Deployment Guide, Hewlett-Packard Development Company (online), available from ⟨http://h10032.www1.hp.com/ctg/Manual/c03037557.pdf⟩ (accessed 2012-04-12).
[5] VMware, Inc.: VMware vCenter Operations Management Suite, VMware, Inc. (online), available from ⟨http://www.vmware.com/jp/products/datacenter-virtualization/vcenter-infrastructure-navigator/overview.html⟩ (accessed 2012-04-13).
[6] 松沢敬一，林 真一，大谷俊雄：Application-aware なデータ階層管理によるアプリケーション処理高速化方式，研究報告システムソフトウェアとオペレーティング・システム，Vol.2012-OS-120, No.8, pp.1–7 (2012).
[7] 坂下幸徳，河野泰隆，柴山 司，中島 淳，敷田幹文：ストレージ管理標準仕様を用いた大規模環境向け構成情報収集方式の提案，情報処理学会第 3 回インターネットと運用技術シンポジウム（IOTS2010），pp.67–74 (2010).
[8] 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).

推薦文

本論文は，大規模なデータセンタの管理運営において，ストレージ，サーバ，アプリケーション等のリソースを管理するとき，その情報取得時間が増大していくという問題に対して体系的に対処する手法を示したものである．理論的な考察を行い，実システムを用いた実験で理論値のパラメータの推定を行い，実験値と理論値の誤差が計測されている．これらは近年大変重要になっているデータセンタの管理運営を行うための重要な知見になる可能性があり，多くの読者に参考になる．

（インターネットと運用技術研究会主査 山之上卓）

© 2013 Information Processing Society of Japan

森 宣仁

2010 年東京大学大学院総合文化研究科広域科学専攻修士課程修了．同年株式会社日立製作所システム開発研究所（現，横浜研究所）入所．現在に至る．IT システム運用管理およびストレージシステムの研究開発に従事．

原 純一

2001 年東北大学大学院情報科学研究科修士課程修了．同年株式会社日立製作所入社．現在，IT プラットフォーム事業本部開発統括本部所属．IT システム運用管理ソフトウェアの開発に従事．

坂下 幸徳 （正会員）

2003 年北陸先端科学技術大学院大学情報科学研究科情報システム学専攻修士課程修了．同年株式会社日立製作所システム開発研究所（現，横浜研究所）入所．現在に至る．IT システム運用管理の研究開発に従事．SNIA 日本支部技術委員会委員長．

牧 晋広 （正会員）

1998 年電気通信大学大学院博士後期課程単位取得退学．同年株式会社日立製作所入社．横浜研究所にてストレージ管理ソフトウェアの研究開発に従事．現在同研究所主任研究員．博士（工学）．

© 2013 Information Processing Society of Japan