情報処理学会論文誌　コンピューティングシステム　Vol.5　No.4　1–12 (Aug. 2012)

データ転送とメタデータアクセスの競合を考慮したストレージシステムの性能モデリング

下薗 紀夫[1,a)]　山本 彰[2]

受付日 2012年1月17日, 採録日 2012年5月4日

概要：ストレージシステムのキャパシティプランニングツールを開発するために，設計の容易さと実用的な精度を両立させる性能モデルが求められている．エンタープライズストレージシステムについて，十分な精度を実現するためにデータ転送とメタデータアクセスの競合をモデル化する必要が生じている．しかし，競合を解析的にモデル化するには Read/Write 要求処理の詳細な分析が必要であり，性能モデルの設計が難しい．そこで本稿では，待ち行列理論の公式を用いてカーブフィッティングを行う，待ち行列近似モデルを提案する．本モデルは，競合の基本的な特性を表現しており，高い精度が期待できる．実際のストレージシステムの測定データを用いて待ち行列近似モデルを評価し，従来の性能モデルや他の方法で競合をモデル化した場合と比べて優れていることを検証した．

キーワード：ストレージシステム，性能モデリング，メタデータアクセス競合

Throughput Modeling of Storage System Considering Data Transfer and Meta Data Access Contention

Norio Shimozono[1,a)]　Akira Yamamoto[2]

Received: January 17, 2012, Accepted: May 4, 2012

Abstract: To develop a capacity plannning tool for storage systems, a performance model that is easy to design and sufficiently accurate is required. And it is nessesary to consider the contention of data transfers and metadata accesses in recent enterprise storage systems to achive sufficient accuracy. But the analytical modeling of the contention requires fine analysis of Read/Write request processing, and it makes to design the performance model difficult. In this paper, we propose the Queue Approximate Model that uses curve fitting with the formula of queuing theory. This model can represent the basic characteristics of contention, so sufficient accuracy is expected. We evaluate this models using performance data of our prototype system. And we verified the Queue Approximate Model is better than an usual model and some other models these are modeled the contention in the other ways.

Keywords: storage system, performance modeling, metadata access contention

1. はじめに

大規模な計算機環境では，記憶容量の高効率利用やホスト間でのデータ共有といったニーズが存在する．そのため，複数のホストがネットワークを介して記憶装置へアクセスする，ストレージエリアネットワーク（SAN）が構築されていることが多い．そのような SAN 環境で使われる記憶装置は，ストレージシステムと呼ばれている．

データ容量や性能の要件は様々であり，それに対応して様々な規模や特徴を持ったストレージシステムが使われている．特に，ミッションクリティカルかつ高性能が必要とされる金融機関，公共機関，航空会社などでは，エンタープライズストレージシステムと呼ばれるシステムが使われ

[1] 株式会社日立製作所横浜研究所
Yokohama Research Laboratory, Hitachi, Ltd., Yokohama, Kanagawa 244–0817, Japan
[2] 株式会社日立製作所研究開発本部
Research & Development Group, Hitachi, Ltd., Kokubunji, Tokyo 185–8601, Japan
[a)] norio.shimozono.zf@hitachi.com

ている．エンタープライズストレージシステムには，データやアクセス負荷の増加に合わせてシステムを段階的に拡張したい，無停止でシステムの故障部位だけを交換したいといった，スケーラビリティやフレキシビリティに対する高い要求がある．そのため，エンタープライズストレージシステムは，大半のコンポーネントやディスクドライブを増減設可能な構造を有するものが多い [1], [2], [3].

エンタープライズストレージシステムは構成の自由度が高いため，ユーザの要求性能を満たすストレージシステムの構成を見積もること（キャパシティプランニング）が重要である．キャパシティプランニングは，システム設計に関わるシステムエンジニアなどが簡易に行えることが望まれるが，性能見積りにはストレージシステム内部に関する知識や，複雑な計算が必要とされるため容易ではない．

そのため，ストレージシステムの性能見積りを容易に行うために，キャパシティプランニングツールが研究されている [4]．このようなキャパシティプランニングツールは，内部にストレージシステムの性能モデル式を持っており，R/W 要求や構成に関する入力情報から，ストレージシステムの最大スループットを見積もることができる．

従来のキャパシティプランニングツールでは，ストレージシステムのハードウェアやソフトウェアの挙動をモデル式で表現する解析モデルが主に使われている．具体的には，ハードウェアの転送帯域や処理能力といった特性パラメータと，入力情報からストレージシステムの各コンポーネント性能を個別に算出し，それらの最小値をストレージシステムの最大スループットとして見積もる方法が使われている [5]．本稿ではこの性能モデルを個別算出モデルと呼ぶことにする．

個別算出モデルでは，特性パラメータを決定するための性能測定回数が少なくてすむ．なぜなら，各コンポーネントに影響する入力パラメータだけを振って測定すればよく，コンポーネント数の組合せは考えなくてよいからである．大規模構成のエンタープライズストレージは，1,000 個以上のハードディスクや数 100 GB のキャッシュメモリを搭載するため，装置を長時間占有することが難しい．また大容量キャッシュを有するため，スループットが安定するまでに長い時間（数分～数十分）を必要とする．そのため，必要な性能測定回数が少ないことは大きなメリットである．

ただし，実際にはコンポーネント間ではデータ転送やメタデータアクセスが行われており，コンポーネント性能はまったくの独立ではない．しかし，エンタープライズストレージでは，データとメタデータを独立したメモリに格納する [6]，キャッシュメモリへのアクセスを直結にする [1] といった，コンポーネント間の競合を抑える工夫が行われている．そのようなストレージシステムでは，個別算出モデルは十分に実用的なモデルである．

しかし，近年のエンタープライズストレージではスケーラビリティやコストパフォーマンスの向上を目的として，データ転送とメタデータアクセスに共通のネットワークやメモリを使うようになってきている [2], [7]．このようなアーキテクチャに単純に個別算出モデルを適用すると，コンポーネント間の競合の影響によって性能モデルの誤差が大きくなってしまうことがあり，これを考慮したモデルが必要である．

競合による応答時間の増加や性能低下は，一般的には待ち行列理論を用いてモデル化できる．しかし，各コンポーネントのサービス時間の平均や分散などのパラメータを決定する必要があるが，決定に必要な情報を取得するのは容易ではない．

そこで本稿では，競合による性能低下をカーブフィッティングを用いて近似的にモデル化することで，既存の個別算出モデルの精度を改善することを試みる．特に，カーブフィッティングに使用する近似式に待ち行列理論の公式を利用することで，少ない未定係数で競合の特性を表現可能な，待ち行列近似モデルを提案する．未定係数はシステムの測定データから同定する．未定係数の数が少ないため，同定に必要な測定データ数も少なくすることができる．

また，待ち行列近似モデルの有効性を検証するため，ストレージシステムの性能データを用いてモデル精度の評価を行う．その際，従来の個別算出モデル，単純にカーブフィッティングを適用した線形近似モデル，性能モデル手法として近年研究されているブラックボックスモデルの一種である Classification And Regression Trees（CART）モデルとの比較を行う．

2. 関連研究

ストレージシステムの性能モデルは，キャパシティプランニングだけでなく，性能の自動管理など他の応用も多い．そのため，ストレージシステム性能モデルについての研究は多く行われている．以下，それらの研究をアプローチ別に紹介する．

まず，ストレージシステムのハードウェアやソフトウェアの挙動をモデル式で表現する方法が，古典的な解析的な性能モデルである．キャパシティプランニングについては，R/W 要求の特性，ハードウェア構成や動作方式からコントローラとディスクの最大スループットをモデル化した研究 [8] がある．また，待ち行列理論を用いてストレージのレスポンスタイムをモデル化し，スケジューラの性能評価を行った研究 [9] や，負荷分散の自動制御方式の性能評価を行った研究 [10] がある．解析的な性能モデルは，ストレージシステムのアーキテクチャに関する情報を使うことができるため，簡単なモデル式と少ない特性パラメータで，実用的な精度が得られることが多い．しかし，ストレージシステムの挙動が複雑な場合には，モデル式の設計が難しくなるという問題がある．

次に，システムをシミュレーションする方法がある [11]．この方法は，解析モデルでは見落としている細かい挙動まで再現できるので，高い精度を実現できる点が優れており，モデル式の設計も不要である．しかし，ハードウェアとソフトウェア両方のシミュレータが必要であり，シミュレータの開発負荷が大きく，実行に膨大な計算時間を必要とするため，キャパシティプランニング向け性能モデルとしては使いにくい．

これらの問題を解決する方法として，近年ブラックボックスモデルを用いた研究が行われている．これは機械学習アルゴリズムを用いて性能をモデル化している．たとえば，Classification And Regression Trees（CART）を用いて，ストレージシステムの性能特性をモデル化する研究 [12], [13], [14] が行われている．ブラックボックスモデルは，ストレージシステムのアーキテクチャに関する情報を使わないため，性能モデルの設計が容易である．しかし，十分な学習を行わせるためには多数の性能データが必要であり，その点がキャパシティプランニング向け性能モデルとしては問題と考えられる．

また，解析的な性能モデルにブラックボックスモデルを組み合わせるようなことも可能であり，そのような研究も行われている．たとえば，IRONModel [15] では，ロバストな性能モデルを実現するために，解析的な性能モデルで定式化されていない影響を，CART モデルを用いて補正する方式が提案されている．本稿も，解析的なモデルである個別算出モデルにおいて，定式化できていない競合の影響を補正することを目的としている．そのため，本稿と IRONModel のモチベーションは近いが，本稿では CART モデルではなく近似的な解析手法で補正を行っており，少ない測定データで実用的なモデル精度を実現している．

3. ストレージシステム概要

本章では，本稿で扱うストレージシステムの概要について述べる．

3.1 ハードウェア構成

ストレージシステムのハードウェア構成を**図 1** に示す．

一般的なストレージシステムは，ホストからの Read/Write 要求（R/W 要求）を処理するためのコントローラと，大量のディスクドライブを搭載するディスクユニットで構成されている．

本稿で扱うエンタープライズストレージでは，コントローラは R/W 要求を処理するために必要な，ホスト/ディスクインタフェース（I/F），プロセッサ，キャッシュメモリなどのコンポーネントが相互結合網に接続され，独立に増減設可能な構造になっている．

ホスト I/F は，Fibre Channel といった SAN への接続プロトコルに対応し，ホストからの R/W 要求などの制御

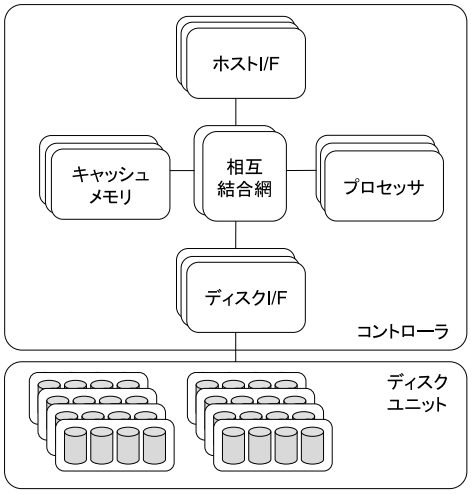

図 1　ストレージシステムのハードウェア構成
Fig. 1　Hardware configuration of storage system.

メッセージをプロセッサとやりとりする，キャッシュメモリとのデータ転送を行うといった機能を持つ．

ディスク I/F も，Serial Attached SCSI といったディスクへの接続プロトコルに対応し，ホスト I/F と同様の機能を持つ．

キャッシュメモリは，データおよびメタデータを格納するためのメモリである．ストレージシステムの信頼性において最も重要なことは，データやメタデータを失わないことである．そのためエンタープライズストレージにおいては，バッテリーバックアップ，2 重化された高信頼なキャッシュメモリを用意し，データおよびメタデータを格納する．

プロセッサは，R/W 要求の処理を行うためのものである．エンタープライズストレージシステムでは，バックアップやスナップショットなどのソフトウェアが拡張機能として提供される．それらを実行するために，R/W 要求の処理は基本的にソフトウェアで実装される．

各コンポーネントの機能を**表 1** に整理して示す．

3.2 R/W 要求処理

ストレージシステムにおいて，R/W 要求がどのように処理されるか説明する．

はじめに，ホストは SCSI コマンドなどのプロトコルを用いてストレージシステムに R/W 要求を送信し，ホスト I/F は受領した R/W 要求をプロセッサへ振り分ける．

プロセッサは，受領した R/W 要求に対応した処理を行う．

はじめに，R/W 要求を解釈する処理が行われる．たとえば，ホストから受けた R/W 要求に含まれる Logical Unit Number（LUN）や Logical Block Address（LBA）を，ストレージシステム内部のアドレス空間に変換する処理が行われる．これは，エンタープライズストレージシステムでは，ディスクドライブに冗長にデータ格納する RAID

表 1　コンポーネントの機能
Table 1　Function of each components.

コンポーネント名称	機能
ホスト I/F	ホストから R/W 要求を受信し，プロセッサへ転送する
	ホストおよびキャッシュメモリとの間でデータ転送を行う
ディスク I/F	ディスクドライブへ R/W 要求を発行する
	ディスクドライブおよびキャッシュメモリとの間でデータ転送を行う
キャッシュメモリ	ユーザデータの一部を一時的に格納する
	R/W 要求処理に必要なメタデータを格納する
プロセッサ	R/W 要求を処理する
相互結合網	データや制御情報を中継する

や，データ書き込みに応じて動的に容量を割り当てる Thin Provisioning が採用されているためである．

また，ストレージシステムはキャッシュメモリにユーザデータをキャッシュすることで高速アクセスを実現する．そのため，キャッシュメモリに R/W 要求が指定するデータが存在するか検索する処理，新規にキャッシュメモリの場所を割り当てる処理，アクセス頻度が低いデータをキャッシュメモリから解放する処理などの，キャッシュ制御が行われる．

アドレス変換処理やキャッシュ制御処理によってデータ転送元と転送先が確定すると，ディスク I/F やホスト I/F とキャッシュメモリ間でデータ転送が行われる．そのため，ホスト I/F やディスク I/F へデータ転送を指示するデータ転送制御が行われる．

3.3　データ転送とメタデータアクセスの競合

R/W 要求処理を行う途中で，プロセッサはキャッシュメモリ上のメタデータへアクセスする必要がある．たとえば，アドレス変換やキャッシュ制御において，アドレス変換テーブルやキャッシュディレクトリへのアクセスが必要となる．一方で，ホスト I/F やディスク I/F とキャッシュメモリの間ではデータ転送が行われる．

そのため，キャッシュメモリでデータ転送とメタデータアクセスが衝突する．すると，データ転送やメタデータアクセスのレイテンシが増加し，ホスト I/F，ディスク I/F，プロセッサの動作効率が低下する．

この競合の影響を受けるのは，主にプロセッサ性能である．なぜなら，ホスト I/F やディスク I/F は 1 度に大量のデータをバースト転送するが，その最大スループットは主に相互結合網との接続パスの帯域によって制限されており，競合の影響は表面化しにくいためである．一方で，プロセッサからのメタデータアクセスはデータ転送と比べてデータ量は少なく，接続パスの帯域ネックはほとんど影響を与えない．そして，R/W 要求の処理を進めたり完了させたりするためにはメタデータアクセスの完了を待つ必要がたびたびあるため，レイテンシ増加の影響を大きく受ける．

このような外部アクセスの待ち時間を隠蔽する方法としては，コンテキストスイッチによって R/W 要求処理を一時中断する方法が一般的である．しかし，ストレージシステムの R/W 要求処理ではメタデータアクセスが主要な処理であり，そのたびにコンテキストスイッチをすると処理効率の低下が大きい．また，メタデータは一般的な仮想記憶のスワップアウト先であるディスクドライブではなく，低レイテンシ（数 μs〜数 10 μs オーダ）でアクセスできるキャッシュメモリに格納されている．そのため，多くの場合においては，コンテキストスイッチをするよりもビジーループでメタデータアクセス完了を待つほうが処理効率が高い．その代わり，競合が激しい状況におけるプロセッサ性能への影響は大きくなる．

エンタープライズストレージでは，キャッシュメモリを増設したときにデータ転送帯域のスケーラビリティを保証するため，相互結合網の帯域はキャッシュメモリの帯域に対して余裕を持たせる設計をしている．そのため，競合の影響はキャッシュメモリにおける衝突が支配的である．

4.　個別算出モデル

本章では，従来の解析的な個別算出モデルにおける性能見積りについて説明する．個別算出モデルにおける性能見積りフローを**図 2** に示し，各種パラメータを**表 2** に示す．

はじめに，R/W 要求の特性と各コンポーネント搭載数が見積り条件として入力される．R/W 要求の特性としては Read と Write の比率 R_r，R_w と，1 回の R/W 要求で転送するデータ量である IO サイズ L を用いる．実際にはこれらに加えてキャッシュヒット率や，ランダムアクセスとシーケンシャルアクセスの比率も用いられることが多いが，本稿では単純化のため，それらを省略しキャッシュミスケースに限定したモデルとする．また，ストレージシステムの構成は，プロセッサおよびキャッシュの搭載数 N_P，N_C が入力される．実際にはホスト I/F，ディスク I/F，ディスクドライブの搭載数も入力されるが，本稿ではキャッシュメモリとプロセッサの性能モデルを扱うため，それらについては説明を省略する．

次に，入力された各コンポーネント搭載数，各コンポーネントの特性パラメータ（データ転送帯域など），データ転

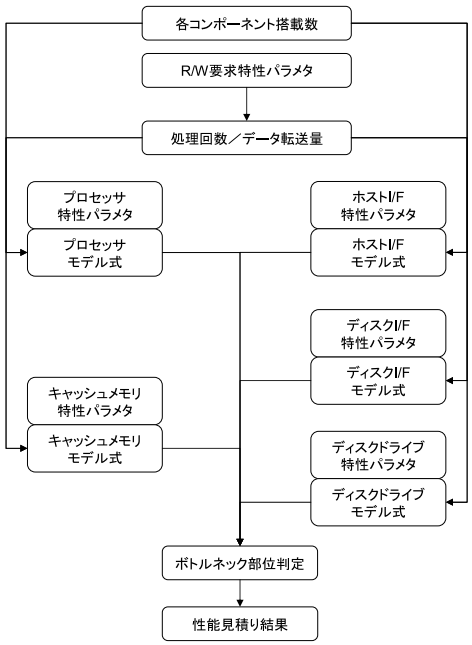

図 2　性能見積りフロー
Fig. 2　Flowchart of performance estimation.

表 2　個別算出モデルパラメータ
Table 2　Parameters of individual calculation model.

記号	意味
R_r	Read 比率
R_w	Write 比率
L	IO サイズ
N_P	プロセッサ搭載数
N_C	キャッシュメモリ搭載数
T_{Pr}	Read 要求プロセッサ処理時間
T_{Pw}	Write 要求プロセッサ処理時間
D_r	Read 要求ごとのキャッシュメモリデータ転送回数
D_w	Write 要求ごとのキャッシュメモリデータ転送回数
B	1 キャッシュメモリのデータ転送帯域
T_C	キャッシュメモリ占有時間
T_P	プロセッサ占有時間
E_{Ci}	個別算出モデルでのキャッシュメモリ限界性能
E_{Pi}	個別算出モデルでのプロセッサ限界性能

送量，処理回数を基に，各コンポーネントの限界性能（最大スループット）が個別に算出される．

プロセッサの特性パラメータには，1 回の R/W 要求の処理に要するプロセッサ時間 T_{Pr}，T_{Pw} を用い，キャッシュメモリの特性パラメータには，1 キャッシュメモリのデータ転送帯域 B と，1 回の R/W 要求あたりのキャッシュメモリへのデータ転送回数 D_r，D_w が用いられる．これらは定数であり，内部トレースなどの測定データや，ハードウェアおよびソフトウェアの設計情報から決定される．

入力された R/W 要求の特性とコンポーネントの特性パラメータから，1 回の R/W 要求あたりのプロセッサとキャッシュメモリの占有時間 T_C，T_P が個別に算出される．T_C は，R_r，R_w と D_r，D_w と L から R/W 要求あたりのデータ転送量を求め，B で割ることで求められる．一方，T_P は，R_r，R_w と T_{Pr}，T_{Pw} を用いて，単純に荷重平均で求められる．

$$T_C = \frac{(R_r D_r + R_w D_w)L}{B} \tag{1}$$

$$T_P = R_r T_{Pr} + R_w T_{Pw} \tag{2}$$

こうして求められた，T_C，T_P の逆数と N_P，N_C の積で，プロセッサおよびキャッシュメモリの限界性能（最大スループット）E_{Ci}，E_{Pi} が算出される．

$$E_{Ci} = \frac{N_C}{T_C} \tag{3}$$

$$E_{Pi} = \frac{N_P}{T_P} \tag{4}$$

最後に各コンポーネントの限界性能を比較し，それらの最小値がシステム限界性能の見積り値として出力される．

5.　競合による性能低下のモデル化

本章では，個別算出モデルに競合によるプロセッサ性能の低下を組み込む方式について述べる．

5.1　プロセッサ性能モデルの競合補正

個別算出モデルにおけるプロセッサ限界性能モデル式 (4) に，競合の影響を組み込む方法について述べる．

まず，3.3 節で述べたように，プロセッサはメタデータアクセスをビジーウェイトする．そのため，データ転送およびメタデータアクセスの競合によってキャッシュメモリからの応答が遅くなると，プロセッサ処理時間 T_P が長くなる．

このプロセッサ処理時間増加量を ΔT_P とすると，ΔT_P が T_P に加算されるため，競合を考慮したプロセッサ限界性能 E_{P_cont} は次のようなモデル式で表現できる．

$$E_{P\_cont} = \frac{N_P}{T_P + \Delta T_P} \tag{5}$$

よって，プロセッサ性能モデル式 (4) を式 (5) に置き換えることで，個別算出モデルに競合の影響を組み込むことができる．

一方，プロセッサ処理時間の増加量 ΔT_P 自体は，ストレージシステムにかかる負荷（スループット）λ のキャッシュメモリ限界性能に対する比率，すなわちキャッシュメモリ利用率 ρ_C に応じて増加する．具体的には，負荷が少ないとき（キャッシュメモリ利用率が低いとき）は競合の発生が少なく ΔT_P はほぼ 0 になり，負荷がキャッシュメモリの限界性能に近づく（キャッシュメモリ利用率が 1 に近づく）につれて，ΔT_P は非常に大きくなる．これは，データ転送やメタデータアクセスが頻繁に衝突し，それらが何重にも待たされるようになるためである．

また，Read 要求と Write 要求ではメタデータアクセス回数やパターンが異なるため，Read 比率 R_r や Write 比率

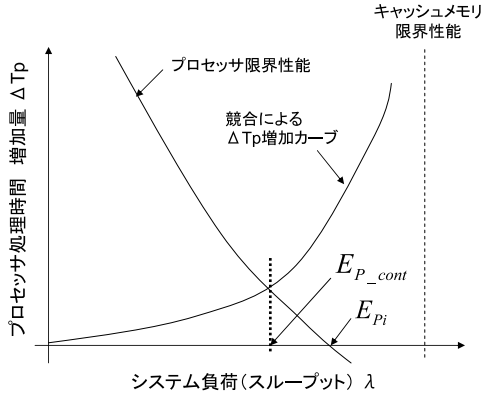

図 3　競合とプロセッサ性能の関係
Fig. 3　Relation of contention and processor throughput.

R_w も ΔT_P に影響を与える．

よって，ΔT_P にはキャッシュメモリ利用率 ρ_C，Read 比率 R_r，Write 比率 R_w が影響する．しかし，それらの影響の仕方は複雑であるため，この時点ではモデル化は行わない．

ここまでの議論を整理するために，競合とプロセッサ性能の関係を，**図 3** に示す．グラフの横軸はストレージシステムにかかる負荷 λ であり，縦軸はプロセッサ処理時間の増加量 ΔT_P である．

まず，プロセッサ限界性能は，式 (5) から右肩下がりの曲線となる．これは ΔT_P が大きいほど競合の影響が大きく，プロセッサ限界性能 E_{P_cont} が低下することを示している．そして，この曲線よりも右側では負荷がプロセッサの処理能力を超えているため，そのような負荷はストレージシステムが受け入れられないことを示す．

一方，ΔT_P は，キャッシュメモリの限界性能の近くで急増する，右肩上がりの曲線になる．

ここで，ストレージシステムにかかる負荷 λ を徐々に増やしていくと，ΔT_P がそれに応じて増大する．そして，プロセッサ限界性能の曲線にぶつかる点で，プロセッサが処理可能な限界に達する．つまり 2 つの曲線の交点における λ の値が，競合を考慮したプロセッサ限界性能 E_{P_cont} となる．

よって，ΔT_P をモデル化できれば，この交点を反復解法などを用いて算出することができる．そこで，次節以降で ΔT_P のモデル化について述べる．

5.2　待ち行列近似モデル

本稿で提案する，待ち行列理論を利用した近似モデル式について述べる．

データ転送およびメタデータアクセスの流れを待ち行列で表現したものを，**図 4** に示す．

まず，データ転送系ではホスト I/F やディスク I/F のデータ転送を制御するコントローラが，キャッシュメモリへメモリアクセスを発行する．メタデータアクセス系でも，プロセッサがキャッシュメモリへメモリアクセスコマンドを発行する．キャッシュメモリはこれらのメモリアクセスを FIFO スケジュールで処理する．

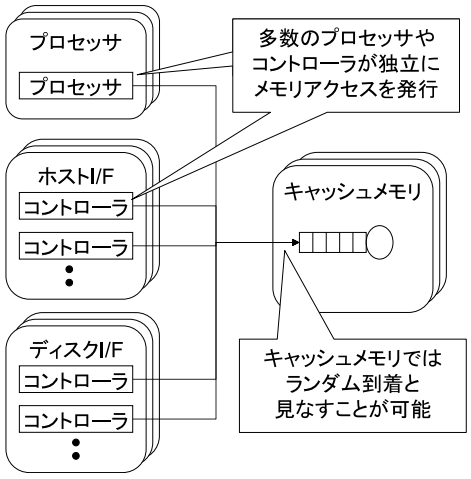

図 4　キャッシュメモリ待ち行列モデル
Fig. 4　Queue model of cache memory.

また，キャッシュメモリへのメモリアクセスは，独立に動作している多数のコントローラとプロセッサから発行される．そのため，キャッシュメモリへの到着間隔は，ランダムに近いものとして考えられる．

よって，サービス時間が一般分布で，ランダム到着を仮定する M/G/1 型の待ち行列の公式である Pollaczek-Khintchine 公式の利用が考えられる [16]．

$$W_q = \frac{\rho}{1-\rho}\frac{1+C_b^2}{2\mu} \tag{6}$$

この公式で，W_q は待ち行列で待たされる時間，ρ は利用率，μ は平均サービス時間，C_b^2 はサービス時間の変動係数を表す．

この式における ρ に依存する部分は，ρ が 0（負荷 λ が 0）のときには 0 になり，ρ が 1 に近づく（λ がキャッシュメモリ限界性能に近づく）と無限大に発散する．

つまり，5.1 節で述べた ΔT_P の右肩上がりの特性は，待ち行列理論では基本的に ρ に依存する部分によって表現されている．

一方，μ や C_b^2 に依存する部分は，サービス時間の特性に依存しており負荷には関係せず，これらの値を決定することは容易ではない．また，W_q は 1 回のメタデータアクセスに関する待ち時間であり，ΔT_P を得るには，W_q にメタデータアクセス回数を掛ける必要もある．

そのため，M/G/1 型の公式の ρ に依存する部分を用いて ΔT_P の大局的な特性を近似し，μ や C_b^2 に依存する部分は Read 比率 R_r と Write 比率 R_w に依存する関数で近似することで，ΔT_P をモデル化することにする．後者の関数については，適当な近似式形状を決めることが難しいため，単純な線形近似式を用いることにする．

以上を整理すると，未定係数 α_r，α_w を導入して，次の

ようなモデル式となる．

$$\Delta T_P = \frac{\rho_C}{1-\rho_C}(\alpha_r R_r + \alpha_w R_w) \tag{7}$$

ここで，キャッシュメモリ利用率 ρ_C は負荷 λ とキャッシュメモリ限界性能の比率で表せる．よって，個別算出モデルにおけるキャッシュメモリ限界性能 E_{Ci} を用いた場合，次式のようになる．

$$\rho_C = \frac{\lambda}{E_{Ci}} \tag{8}$$

この個別算出モデルのキャッシュメモリ限界性能 E_{Ci} は，データ転送回数だけで算出された近似値である．しかし，モデル式 (7) は，それが無限大に発散するキャッシュメモリ限界性能に対して非常に敏感であるため，E_{Ci} をそのまま用いるとうまくカーブフィッティングできない可能性がある．そこで，キャッシュメモリ限界性能およびキャッシュメモリ利用率に，メタデータアクセスを考慮した修正を行う．

メタデータアクセスによるキャッシュ占有時間は，ソフトウェアとハードウェア両方が影響するため正確なモデル化は容易ではないが，Read ミス，Write ミスなど，同一の処理ルートであれば大きな差異は発生しない．そこで，R/W 要求ごとのメタデータアクセスによるキャッシュメモリ占有時間を未定係数 M_r，M_w とする．そうすると，修正後のキャッシュメモリ限界性能 $E_{C\_correct}$ は次式で表せる．

$$E_{C\_correct} = \frac{N_C}{T_C + R_r M_r + R_w M_w} \tag{9}$$

これを用いて，修正後のキャッシュ利用率 $\rho_{Ccorrect}$ は次式のように表せる．

$$\rho_{Ccorrect} = \frac{\lambda}{E_{C\_correct}} \tag{10}$$

これを式 (7) に適用して，ΔT_P を算出するものとする．

5.3　線形近似モデル

待ち行列近似モデルとの比較として，待ち行列理論を利用しない一般的なカーブフィッティングを検討する．5.1 節で述べたように，キャッシュ利用率 ρ_C，Read 比率 R_r，Write 比率 R_w を用いて，プロセッサ処理時間の増加量 ΔT_P をモデル化している．

このようにパラメータ種別だけが既知で，モデル式の形状が未知な場合には，多項式近似が一般的な近似手法である．本稿では，必要な測定データ数を抑えたいため，必要な未定係数が最小の線形近似を用いる．具体的には，未定係数 β_ρ，β_r，β_w を用いて以下のような近似式を用いる．

$$\Delta T_P = \beta_\rho \rho_C + \beta_r R_r + \beta_w R_w \tag{11}$$

ここでは，$\rho_{Ccorrect}$ ではなく修正前のキャッシュ利用率 ρ_C を使用する．なぜなら，キャッシュ利用率の修正を行っても未定係数を増やすだけで，性能モデルの精度向上には寄与しないと考えられるためである．具体的に説明すると，この近似式においてキャッシュ占有時間の補正を行った場合，ΔT_P に切片を加えることができる．しかし，負荷 λ が低いときは ΔT_P は 0 に近いはずであり，切片は小さい値しかとりえないためである．

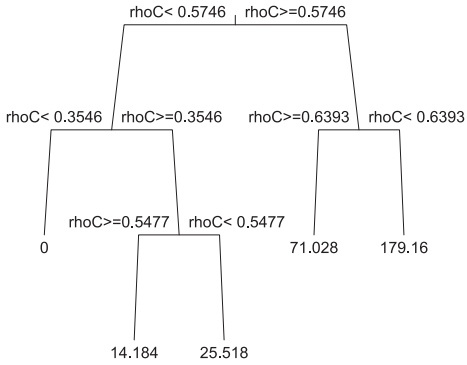

図 5　CART の例
Fig. 5　Example of CART.

5.4　CART モデル

5.1 節で述べたように，ΔT_P はキャッシュメモリ限界性能に近いところで急増する特性を有する．そのため，線形近似では精度が高くなりにくい．

そこで，近年研究が行われている CART を用いたモデル化を検討する．CART のようなブラックボックスモデルを用いる場合，入力と出力のパラメータに何を使うかを決定する必要がある．

このケースでは，入力は ρ_C，R_r，R_w で，出力は ΔT_P である．

入出力パラメータの実測サンプルをモデルに与えて学習させると，**図 5** のようなツリーが生成される．

5.5　未定係数の同定

待ち行列近似モデルでは α_r，α_w，M_r，M_w，線形近似モデルでは β_ρ，β_r，β_w の未定係数が用いられている．

これらの未定係数を，様々な条件下で測定された実測データを基に同定する．まず，プロセッサおよびキャッシュメモリの搭載数や R/W 要求の特性パラメータが異なる測定条件で，プロセッサネックの性能を測定する．競合が十分に少ない条件でプロセッサ性能を測定しておき，それらを比較することで ΔT_P の実測データを得ることができる．

また，待ち行列近似モデルではキャッシュ利用率の補正を行っている．そのため，M_r，M_w の同定にはデータ転送とメタデータアクセスの比率が異なるデータが必要である．そのため，IO サイズが異なるデータを追加する必要がある．

同定の方法には多くの方法があるが，本稿では最も一般的な同定手法である最小二乗法を用いる．

6. 評価

本章では，実際のストレージシステムにおける性能測定データを用いて，個別算出モデルに待ち行列近似モデル，線形近似モデル，CART モデルを組み込んだ各モデルが，競合の影響によるプロセッサ性能低下をどれくらい正確に補正できるか評価する．

6.1 ストレージシステムの構成

まず，評価に用いるストレージシステムの構成について述べる．

プロセッサ数 N_P を 1–8 まで，キャッシュメモリ数を 1–16 まで搭載可能な，図 1 に示す構造を持ったストレージシステムを試作した．主要緒元を**表 3** に示す．

6.2 評価方法

ストレージシステムの性能測定データを用いて，従来の個別算出モデルと，それを待ち行列近似モデル，線形近似モデル，CART モデルで補正した性能モデルについて，プロセッサ性能の精度を比較評価する．

はじめに，ストレージシステムの性能測定データを収集する．本稿ではキャッシュメモリにおける競合の影響を受けたプロセッサ性能に着目しているため，基本的にプロセッサネック（プロセッサ利用率が 100%）の性能測定データを収集する．

まず，アクセスパターンについては，ストレージシステムの性能評価で一般的に用いられている，ディスク容量全体へのランダムアクセスを用いる．この場合，ほぼ 100%キャッシュミスとなるが，キャッシュミスはメタデータを大量に参照更新する必要があり，競合の影響を強く受けるため本評価に適当であると考えられる．

一方で，ローカリティのあるアクセスパターンも性能評価では用いられており，その場合キャッシュヒット率が高い性能測定となるが，キャッシュヒットはメタデータアクセス回数が非常に少ないため，ほとんど競合の影響を受けない．よって，キャッシュミスのケースについて評価を行えば十分であると考えられる．

IO サイズや Read 比率，Write 比率については，データ転送量やメタデータアクセスパターンの違いによる競合の影響の違いを評価するため，異なる数パターンを使用する．

また，ストレージシステムの装置構成については，ホスト IF およびディスク IF については本稿の性能評価には関係しないため，これらがボトルネックにならないよう最大数搭載する．そして，キャッシュメモリ限界性能を変えた場合の競合の影響の変化をみるため，キャッシュメモリの搭載数を増減させた数パターンを使用する．

これらを整理し，**表 4**，**表 5** に示す構成で性能測定データを収集した．

次に，性能測定データのサブセットを用いて，待ち行列近似モデルと線形近似モデルについてパラメータ同定を，CART モデルについては学習を行う．各モデルを公平に比較するため，パラメータ同定や学習には同一のサブセットを使用する．

まず，1 つ目のサブセットとしては，キャッシュメモリの搭載数が最大および最小のときのデータと，IO サイズが異なるデータを用いる（**表 6**）．これは極端なケースを用いて同定や学習を行うことで，中間的な構成が補完されることを期待するためである．

しかし，大きな構成での測定は非常に多くの物理資源を

表 3　評価ストレージシステム緒元
Table 3　Specification of evaluated storage.

プロセッサスペック	Quad-Core 2.33 GHz
プロセッサ搭載数（N_P）	1–8
キャッシュメモリ搭載数（N_C）	1–16
ホスト IF 搭載数	1–16
ディスク IF 搭載数	1–8

表 4　性能評価構成 (1)
Table 4　Configuration of evaluation (1).

測定名	N_P	N_C	R_r	L [kB]
Rc1	8	2	1.0	8
Rc2	8	4	1.0	8
Rc3	8	8	1.0	8
Rc4	8	16	1.0	8
Wc1	8	2	0.0	8
Wc2	8	4	0.0	8
Wc3	8	8	0.0	8
Wc4	8	16	0.0	8
Mc1	8	2	0.6	8
Mc2	8	4	0.6	8
Mc3	8	8	0.6	8
Mc4	8	16	0.6	8

表 5　性能評価構成 (2)
Table 5　Configuration of evaluation (2).

測定名	N_P	N_C	R_r	L [kB]
Rl1	8	16	1.0	8
Rl2	8	16	1.0	32
Rl3	8	16	1.0	48
Wl1	8	16	0.0	8
Wl2	8	16	0.0	32
Wl3	8	16	0.0	48
Rs1	2	4	1.0	8
Rs2	2	4	1.0	16
Ws1	2	4	0.0	8
Ws2	2	4	0.0	16

表 6　パラメータ同定構成 (1)
Table 6　Configuration of identification (1).

測定名	N_P	N_C	R_r	L [kB]
Rc1	8	2	1.0	8
Rc4	8	16	1.0	8
Rl2	8	16	1.0	32
Wc1	8	2	1.0	8
Wc4	8	16	1.0	8
Wl2	8	16	0.0	32

表 7　パラメータ同定構成 (2)
Table 7　Configuration of identification (2).

測定名	N_P	N_C	R_r	L [kB]
Rs1	2	4	1.0	8
Rs2	2	4	1.0	16
Ws1	2	4	0.0	8
Ws2	2	4	0.0	16
Rc1	8	2	1.0	8
Wc1	8	2	1.0	8

要求するため，小構成から大構成性能を予測せざるをえないことがある．そのため，2 つ目のサブセットとしては，小構成におけるデータのみを用いる（**表 7**）．

競合近似モデルと線形近似モデルのパラメータ同定には最小二乗法を用い，CART モデルの学習には統計分析ツール R の mvpart パッケージを使用する [17], [18].

最後に，収集した全データに対する各モデルの誤差を評価する．キャパシティプランニングツールの出力をどの程度信頼できるかという観点から，誤差の平均値だけでなく最大値も重要である．そこで，全データに対する誤差絶対値の平均値と最大値を評価指標とする．

6.3　個別評価

個別の評価ケースについてグラフを用いて説明する．

まず，プロセッサ搭載数を固定しキャッシュメモリを増減させた場合について説明する．実測性能と各モデルの予測性能，実測性能におけるキャッシュメモリ帯域の利用率（個別算出モデルを用いた推定値）を，**図 6**，**図 7**，**図 8** に示す．グラフ中の各モデルの (1)，(2) は，パラメータ同定構成 (1)，(2) に対応している．

各グラフから，キャッシュメモリの搭載数が少ないとき，実測性能が低下することが確認できる．これは，キャッシュメモリの搭載数が少ないと，プロセッサ限界性能に対してキャッシュメモリ限界性能の余裕がなくなり，キャッシュメモリ帯域利用率が上昇することで，アクセス競合が激しくなるためである．そのため，競合を考慮しない個別算出モデルでは，誤差が非常に大きくなる．

まず，線形近似モデルは個別算出モデルと比較すると大幅に改善されていることが分かる．しかし非線形な競合の特性を線形近似しているために，多少誤差が残ってしまっている．

次に，CART モデルも個別算出モデルと比較すると大幅に改善されている．しかし，小規模構成のデータを学習に用いたケース (2) ではところどころで誤差が大きくなってしまっている．これは CART モデルに与えるサンプルと，評価している構成の差異が大きいためと考えられる．

一方，待ち行列近似モデルは全域で誤差を小さく抑えることができている．また，小規模構成のデータをパラメータ同定に用いたケース (2) でも同程度の精度を実現できている．

ただし，待ち行列近似モデルも Read/Write 混在ケース（図 8）の $N_C = 16$ では誤差が比較的大きい．だが，Read 100%，Write 100%ともに $N_C = 16$ では競合の影響が十分に小さいことが実測データから分かるため，混在ケースのみ競合の影響が増えるとは考えにくい．そのため，実測データの個別算出モデルに対する低下要因は競合によるものではなく，処理が混在することでプロセッサの命令/データキャッシュヒット率が低下するなどして T_P 自体が増加したためと考えられる．

次に，R/W 要求の IO サイズを増減させた場合を評価した．**図 9**，**図 10** に評価結果を示す．

キャッシュメモリを減らしたケースと同様，IO サイズを大きくするとデータ転送量が増加し，データ転送量がキャッシュメモリの帯域に近くなることで，競合によって実測性能が低下し，個別算出モデルの誤差が大きくなる．各モデルの誤差の傾向についても，キャッシュメモリを減らしたケースと同様である．

6.4　評価まとめ

全データに対する各モデル誤差の一覧を，**表 8** に示す．

キャパシティプランニングの観点から評価指標とした，誤差絶対値の平均値と最大値両方においてパラメータ同定構成 (1)，(2) ともに待ち行列近似モデルが最も優れていることが分かる．小規模な構成からパラメータ同定を行った場合でも，ほとんど誤差が変化していない点も優れている．

線形近似モデルについては，個別算出モデルからは大幅に改善されているが，非線形な競合特性を線形近似していることによる限界があり，待ち行列近似モデルほどの改善は得られない．

CART モデルについても，個別算出モデルからは大幅に改善されている．しかし，改善度合いが学習に用いるデータに大きく依存しており，今回用いた程度のデータ数では信頼できるモデルとはいえない．しかし，データ数を増やすことは容易ではなく，エンタープライズストレージシステムのキャパシティプランニング向け性能モデルとしては使いにくい．

7.　まとめ

本稿では，ストレージシステムのキャパシティプランニ

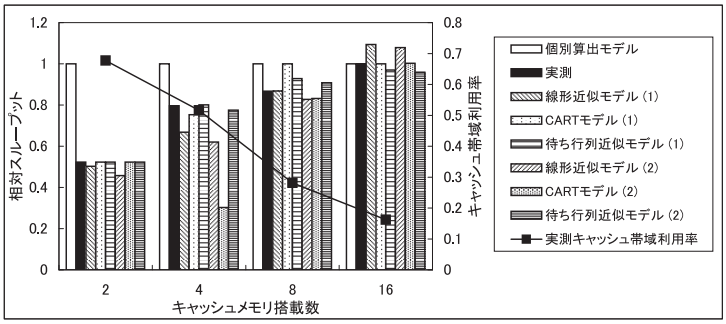

図 6　$N_P = 8$，$N_C = 2, 4, 8, 16$，Raed 100%
Fig. 6　$N_P = 8$, $N_C = 2, 4, 8, 16$, Read 100%.

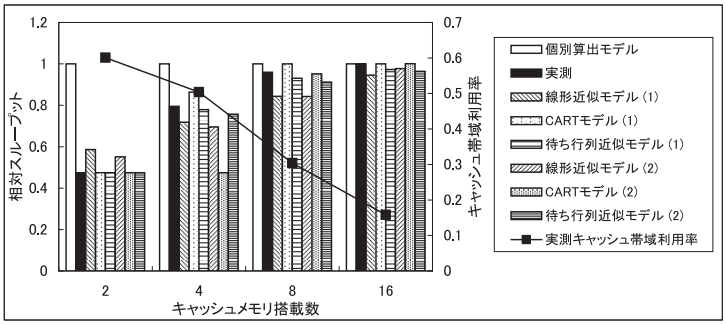

図 7　$N_P = 8$，$N_C = 2, 4, 8, 16$，Write 100%
Fig. 7　$N_P = 8$, $N_C = 2, 4, 8, 16$, Write 100%.

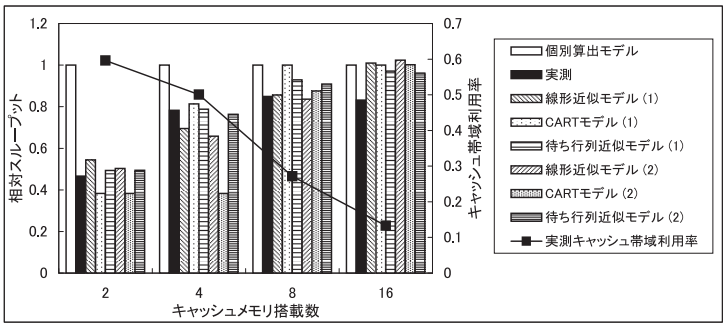

図 8　$N_P = 8$，$N_C = 2, 4, 8, 16$，Read : Write = 6 : 4
Fig. 8　$N_P = 8$, $N_C = 2, 4, 8, 16$, Read : Write = 6 : 4.

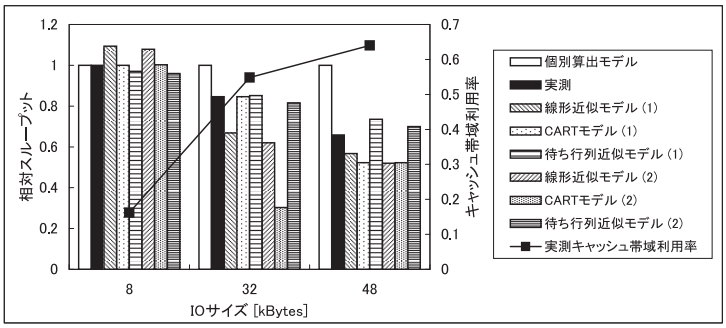

図 9　$N_P = 8$，$N_C = 16$，$IOSIZE = 8, 32, 48$ kB，Read 100%
Fig. 9　$N_P = 8$, $N_C = 16$, $IOSIZE = 8, 32, 48$ kB, Read 100%.

ングに向けた性能モデルとして，各コンポーネント性能を独立に算出する従来の個別算出モデルをベースに，データ転送およびメタデータアクセスの競合の影響を組み込んだ性能モデルの改善方式を提案，評価した．

改善方式としては，競合の影響をカーブフィッティングで近似的にモデル化する方式を提案した．特に，単純な線形近似よりも高い精度が得られるため，カーブフィッティングに用いる式に待ち行列理論の公式を利用した．ま

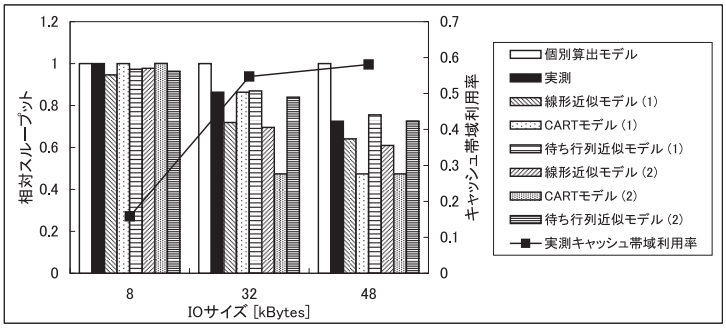

図 10　$N_P = 8$, $N_C = 16$, $IOSIZE = 8, 32, 48$ kB, Write 100%
Fig. 10　$N_P = 8$, $N_C = 16$, $IOSIZE = 8, 32, 48$ kB, Write 100%.

表 8　モデル誤差まとめ
Table 8　Summary of model errors.

測定名	個別算出モデル	線形近似モデル		CART モデル		待ち行列近似モデル	
		同定構成 (1)	同定構成 (2)	同定構成 (1)	同定構成 (2)	同定構成 (1)	同定構成 (2)
MAX ABS	114.5%	23.6%	26.7%	34.6%	64.2%	16.7%	15.5%
AVE ABS	30.0%	11.1%	12.5%	8.5%	18.3%	4.1%	4.0%

た，比較対象としてブラックボックスモデルの一種である CART モデルを用いたモデル化についても検討した．

次に，実際にストレージシステムを用いて検証を行い，各モデルの評価を行った．従来の個別算出モデルに対して，各モデルが誤差を大幅に改善できることを確認した．特に，待ち行列近似モデルの他の線形近似モデル，CART モデルに対する優位性を検証した．

これにより，待ち行列近似モデルを利用することでキャパシティプランニングツールに加えるべきマージンを大幅に減らすことができ，その結果ストレージシステムのコストパフォーマンスを改善することができることを検証した．

今後の研究としては，本稿で提案したモデルの適用範囲の拡大が考えられる．

まず，本稿と同一のアーキテクチャでさらに大規模でコンポーネントが多い場合への対応が考えられる．本稿では，相互結合網がボトルネックにならないシステムを前提としており，相互結合網をモデル化していない．しかし，実用的なコストでストレージシステムを大規模化するためには，相互結合網を疎結合な構成にする必要がある．その結果，キャッシュメモリ帯域に対して相互結合網の帯域が不足しやすくなり，相互結合網での競合が無視できなくなる．そこで，相互結合網における競合についても，競合近似モデルを適用した性能モデルの検討，評価を試みたい．

さらに，本稿で扱ったデータ転送とメタデータアクセスの競合による性能低下はエンタープライズストレージシステムだけでなく，データとメタデータを同一のメモリに格納する，多くのストレージシステムにおいて重要な問題である．近年はマルチコアプロセッサが一般化しており，またそのコア数も増加している．そのため，単純にコア数が多いプロセッサへの換装でプロセッサ性能を増強したくなるが，メモリ帯域とのバランスをとらずに過多なコア数を搭載すると本稿と同種の問題によって性能が向上しなくなると考えられ，評価を試みたい．

参考文献

[1] EMC Corporation: EMC Symmetrix DMX-4, available from ⟨http://www.emc.com/storage/symmetrix/dmx-4.htm⟩ (2007).
[2] Hitachi, Ltd.: Hitachi Virtual Storage Platform, available from ⟨http://www.hitachi.co.jp/products/it/storage-solutions/products/vsp/index.html⟩ (2010).
[3] IBM Corporation: IBM System Storage DS8000, available from ⟨http://www-03.ibm.com/systems/storage/disk/ds8000/index.html⟩ (2010).
[4] Anderson, E., Spence, S., Swaminathan, R., Kallahalla, M. and Wang, Q.: Quickly finding near-optimal storage designs, *ACM Trans. Comput. Syst.*, Vol.23, pp.337–374 (2005).
[5] 田中　徹，中嶋法子，田口雄一，田村賢司：O-003 ストレージシステムにおける負荷傾向予測方式の提案（情報システム，一般論文），情報科学技術フォーラム講演論文集，Vol.8, No.4, pp.535–536 (2009).
[6] 高橋直也，黒須康雄：キャッシュメモリと共有メモリをもつディスクアレーの高速化手法（計算機システム），電子情報通信学会論文誌 D-I，情報・システム，I-情報処理，Vol.86, No.6, pp.375–388 (2003).
[7] EMC Corporation: EMC Symmetrix VMAX, available from ⟨http://www.emc.com/storage/symmetrix/vmax.htm⟩ (2009).
[8] Uysal, M., Alvarez, G.A. and Merchant, A.: A Modular, Analytical Throughput Model for Modern Disk Arrays, *International Symposium on Modeling, Analysis, and Simulation of Computer Systems*, Vol.0, p.0183 (2001).
[9] Chen, S. and Towsley, D.: A Performance Evaluation of RAID Architectures, *IEEE Trans. Comput.*, Vol.45, pp.1116–1130 (1996).
[10] 下薗紀夫，山本　彰，芹沢　一，杉本定広：移送処理を考慮した負荷分散の最適化（コンピュータシステム），電子情報通信学会論文誌 D，情報・システム，Vol.91, No.4,

pp.897–906 (2008).
[11] Bucy, J.S. and Ganger, G.R.: The DiskSim Simulation Environment Version 3.0 Reference Manual, Technical Report (2003).
[12] Wang, M., Au, K., Ailamaki, A., Brockwell, A., Faloutsos, C. and Ganger, G.R.: Storage Device Performance Prediction with CART Models, *International Symposium on Modeling, Analysis, and Simulation of Computer Systems*, Vol.0, pp.588–595 (2004).
[13] Yin, L., Uttamchandani, S. and Katz, R.: An Empirical Exploration of Black-Box Performance Models for Storage Systems, *International Symposium on Modeling, Analysis, and Simulation of Computer Systems*, Vol.0, pp.433–440 (2006).
[14] Li, S. and Huang, H.H.: Black-Box Performance Modeling for Solid-State Drives, *International Symposium on Modeling, Analysis, and Simulation of Computer Systems*, Vol.0, pp.391–393 (2010).
[15] Thereska, E. and Ganger, G.R.: Ironmodel: Robust performance models in the wild, *Proc. 2008 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, *SIGMETRICS '08*, New York, NY, USA, ACM, pp.253–264 (2008).
[16] Kleinrock, L.: *Queuing systems: Theory*, Vol.1, Wiley-Interscience (1975).
[17] The R Project for Statistical Computing, available from ⟨http://www.r-project.org/⟩.
[18] Package mvpart Reference Manual, available from ⟨http://cran.r-project.org/web/packages/mvpart/mvpart.pdf⟩.

下薗 紀夫

平成 12 年東京大学工学部航空宇宙工学科卒業．平成 14 年同大学大学院修士課程修了．同年日立製作所入社．以来，システム開発研究所（現：横浜研究所）にてストレージシステム性能に関する研究に従事．

山本 彰　（正会員）

昭和 52 年京都大学工学部情報工学科卒業．昭和 54 年同大学大学院修士課程修了．同年日立製作所に入社し，システム開発研究所（現：横浜研究所）に入所．性能評価手法，ストレージシステムの研究に従事．平成 19 年より同社研究開発本部所属．