マルチコアプロセッサによる負荷共有方式の高性能化

吉原 朋宏† 越智 隆‡ 下薗 紀夫† 工藤 晋太郎† 山本 政行†

†株式会社日立製作所 システム開発研究所 〒224-0817 神奈川県横浜市戸塚区吉田町 292 番地
‡株式会社日立製作所 RAID システム事業部 〒250-0872 神奈川県小田原市中里 322-2

E-mail: †{tomohiro.yoshihara.rj,norio.shimozono.zf, shintaro.kudo.gs, masayuki.yamamoto.jw}@hitachi.com,
‡takashi.ochi.xn @hitachi.com

あらまし 近年，高性能なマルチコアプロセッサの対称型マルチプロセッシングの普及により，プロセッサの主記憶上で外部からの入力を共有し負荷分散する方式が有効となってきている．本稿では，ストレージシステムの I/O 要求をプロセッサの主記憶で共有したときの評価を行う．複数コアで連携処理する共有分散方式と同等の 1 ポート性能，特定コアが専有で処理する専有分散方式と同等システム性能を目標とする．I/O 量が一定以下のポートは専有分散方式で処理し，一定以上のポートは共有分散方式で処理し，各コアの I/O 処理負荷を平滑化する，専有ベース切換え方式を提案する．プロトタイプシステムでの実機検証から，提案方式が 1 ポート性能，システム性能ともに目標を達成することを示す．また，ポートの負荷が偏っている環境でも有効であることを示す．
キーワード マルチコアプロセッサ，負荷分散，性能評価

Performance Improvement of the Method Sharing Loads by Multi-Core Processors

Tomohiro YOSHIHARA†, Takashi OCHI‡, Norio Shimozono†, Shintaro KUDO†,
and Masayuki YAMAMOTO‡

† Systems Development Laboratory, Hitachi, Ltd. 292 Yoshida-cho, Totsuka-ku, Yokohama-shi, Kanagawa 244-0817 Japan
‡ Disk Array Systems Division, Hitachi, Ltd. 322-2 Nakazato, Odawara-shi, Kanagawa 250-0872 Japan

E-mail: †{tomohiro.yoshihara.rj,norio.shimozono.zf, shintaro.kudo.gs, masayuki.yamamoto.jw}@hitachi.com,
‡takashi.ochi.xn @hitachi.com

Abstract Recently, the symmetric multiprocessing on high-performance multi-core processors is efficient to blace loads by sharing external inputs on main memory for processors. In this paper, we evaluate the method sharing requets for I/Os to storage systems by multi-core processors. We target the equivant one port performance of the method shared by all cores and the equivant system performance of all exclusive methods. We propose the hybrid method based for all exclusive method. In the proposed method, one core processes subthreshold requests from a host port and multi-cores process requests over the threshold. The experimental results on a prototype system indicate that we achieve targets by the proposed method, and the proposed method is effective for configuration having requets skews among ports.
Keyword Multi-Core Processor, Load Blancing, Performance Evaluation

1. はじめに

近年，マルチコアプロセッサのロック制御等の高性能化により，外部から到着するジョブをプロセッサの主記憶を用いて共有する対称型マルチプロセッシング（SMP; Symmetric MultiProcessing）方式が高い処理効率を求められるシステムで有効となってきている．実際のシステムにおいて SMP を用いて，処理ノード数に見合った性能を引き出すためには，ジョブを負荷分散させる必要がある．

SMP システムで負荷分散する方式として，すべての資源を共有とする方式，入力のみを共有する方式，その中間の一部資源のみ共有する方式がある．共有する資源が多いほど，負荷の平準化が容易であるが，処理効率が低下する．各方式には一長一短があるが，本稿では，以下のような特徴を持ち，かつ非常に高い処理効率を求められるシステムを前提とし，負荷分散をするために最低限必要である入力のみを共有する方式を対象とする．

特徴 **1**：ネックとなり得る専用の振り分けノードを用いないため，各物理的な入力 I/F に対応する入力キューは 1 つである．

特徴 **2**：入力キューの排他には，サービス時間を消費し，そのサービス時間はジョブの処理に必要なサービス時間に対して，無視できるほど小さくない．

特徴 **3**：入力キュー以外の資源を共有していないため，あるコアで開始したジョブは，そのコア以外で検出した場合，ジョブを移送する必要がる．

本稿で扱うような非常に高い処理効率を求められるシステムとして,ストレージシステムが挙げられる．そこで，本稿ではストレージシステムの I/O 要求をプロセッサの主記憶で共有したときの評価を行う．複数コアで連携処理する共有分散方式と同等の 1 ポート性能，特定コアが専有で処理する専有分散方式と同等システム性能を目標とする．I/O 量が一定以下のポートは専有分散方式で処理し，一定以上のポートは共有分散方式で処理し, 各コアの I/O 処理負荷を平滑化する，専有ベース切換え方式を提案する．

プロトタイプシステムでの実機検証から，提案方式が 1 ポート性能，システム性能ともに目標を達成することを示す．また，ポートの負荷が偏っている環境でも有効であることを示す．

以下に本稿の構成を述べる．まず，2 節で負荷分散方式の課題と目標について述べる．3 節で課題解決のアプローチと提案する負荷分散方式の詳細について述べる．4 節では，提案方式の実機検証による評価について述べる．最後に 5 節でまとめと今後の課題を述べる．

1.1. ストレージシステム

ストレージシステムとは，多数の HDD と大容量キャッシュメモリを搭載し RAID 制御・キャッシュ制御やコピー制御などを行うことで，高性能・高信頼・高可用なボリュームと多様なコピー機能をホストに提供し，主にファイバチャネルと呼ばれる I/F で構成されるストレージエリアネットワーク(SAN)経由でアクセスできる装置である．

コネクティビティ・性能・可用性を高めるため，ストレージシステムは複数のファイバチャネルポート，プロセッサ，キャッシュメモリなどを搭載する．小～中規模システムでは，2 個のコントローラを高速バスなどで接続し HDD を共有するデュアルコントローラ構成をとることが多い（図 1）[1]. 一方 10 個以上のプロセッサやポートを搭載する，大規模なシステムも存在し[2], 規模や可用性などシステム要件に応じて使い分けがなされている．

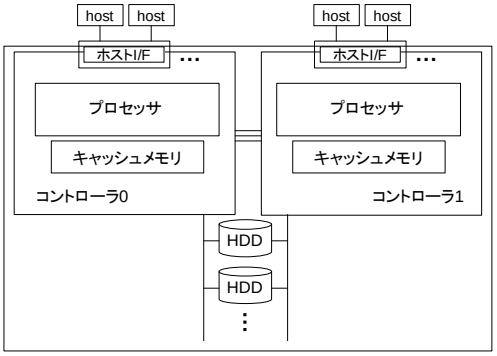

図 **1**　ストレージシステム

プロセッサの主記憶上の共有による SMP が高性能化する以前のストレージシステムでは，各ホストポートの担当プロセッサを備えた以下のようなのプロセッサの負荷分散方式が主流であった．

マルチパスドライバ方式：ホストとストレージシステムを複数のアクセス経路で接続し，ホストデバイスドライバが負荷分散を行う方式である．この方式はホストとストレージ間パスの冗長化と帯域拡張に有効であるが，担当プロセッサの負荷分散を目的としてパス数を増やすことは，冗長なポートやケーブルが必要であり，高コストとなる欠点がある．

専用振り分けプロセッサ方式：ポート担当のプロセッサは, I/O 処理プロセッサへの振り分けのみを行う，または，振り分けとプロトコル制御のみを行う方式である．この方式は振り分けプロセッサがボトルネックになりやすい点，専用の振り分けプロセッサを設けることでコストが上昇するなどの欠点がある．

プロセッサ間通信方式：各ポート担当のプロセッサは，負荷の状況に応じて他のポート担当のプロセッサに通信し処理を移送する．または，アクセスブロックに応じて決まったプロセッサに移送する．この方式は負荷分散処理が複雑になる，またはプロセッサ間通信処理にプロセッサのリソースを消費するなどの欠点がある．

これらの方式の課題を解決できる方式として，マルチコアプロセッサの主記憶上での I/O 入力共有方式を検討した．

2. 負荷分散方式の課題と目標

2.1. 入力共有型 I/O 処理

入力共有型のホスト I/O 処理について述べる(図 2)．ホストがポートに I/O 要求を送信する．ホスト I/F は主記憶上の I/O 要求キューにエンキューする．マルチプロセッサの各コアが I/O 要求キューをポーリングし，要求の有無をチェックする．要求があれば，他コアと排他するため，要求キューをロックし，要求を取り出

す．要求に従って，I/O 処理を行う．

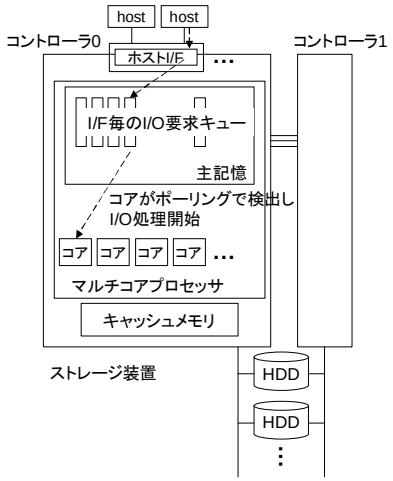

図 **2**　入力共有型 **I/O** 処理

2.2. 負荷分散方式比較

複数のポートの負荷を複数のプロセッサで分散する方法として，以下の 2 方式が考えられる（図 3）．

（**A**）共有分散方式：各ポートからの I/O 要求をすべてのプロセッサで受けて，負荷を分散する．

（**B**）専有分散方式：各ポートからの I/O 要求を受ける専有プロセッサを一定時間毎に切替えて，負荷を分散する．

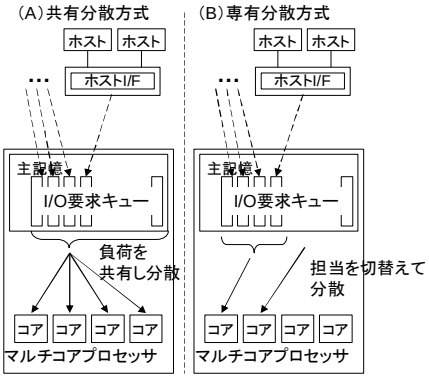

図 **3**　負荷分散方式

まず，方式（A）（B）の課題を挙げ，表 1 のように評価した．方式（A）は，他のプロセッサが一度処理した I/O を途中で受けた場合に転送する通信が課題であった.キューチェック時の要求コピーだけロックし，また 1 ロックで複数要求まとめてコピーすることで，ロック OVH とスキャン OVH の課題は解決できた．方式(B)の課題は，1 ポートや少数ポートの負荷を分散できないことであった．

表 **1**　負荷共有方式比較表

課題名	（A）共有分散方式	（B）専有分散方式
ロック OVH	○（※1）	○
通信 OVH	×（※2）	○
スキャン OVH	○（※3）	○
負荷分散	○	×（※4）

※1：キューチェック時のみ，かつ複数要求まとめて実行
※2：他コアの処理中要求のみ通信
※3：複数要求まとめて実行
※4：1 ポートや少数ポートの過負荷を分散できない

方式（A）（B）を比較する．方式（A）は，処理効率の観点では課題があり，方式（B）は，1 ポートや少数ポートの負荷を分散できず，負荷分散の観点で課題がある．これらの課題が，ホストへの性能に与える影響を具体的な状況の性能で例示する．

1 ポート性能

1 ポートにのみ I/O が到着しているときの性能（1 ポート性能）は，図 4 のようになると考える．横軸はスループット，縦軸はレスポンスを示す．横軸に大きいほどスループット性能が優れており，縦軸が小さいほどレスポンス性能が優れていることを示す．図からわかるように，方式（A）が方式（B）より優れている．このような状況では，方式（B）は負荷が集中した特定のプロセッサがネックとなる．

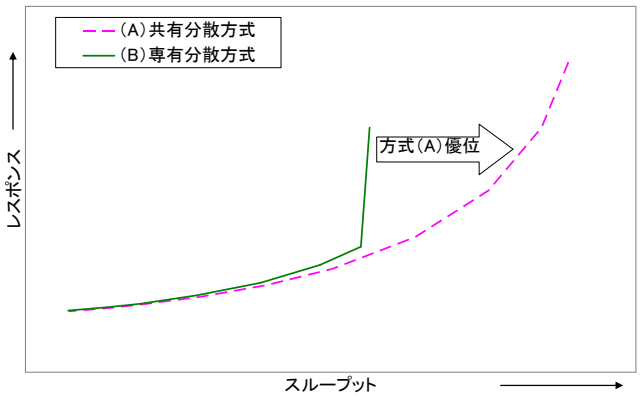

図 **4**　**1** ポート性能比較

システム性能

プロセッサに均等 I/O を割り振れるように I/O が到着しているときの性能（システム性能）は，図 5 のようになると考える．軸の示す値は図 4 と同様である．図からわかるように，方式（B）が方式（A）より優れている．このような状況では，方式（A）は通信 OVH で処理効率が低下する．

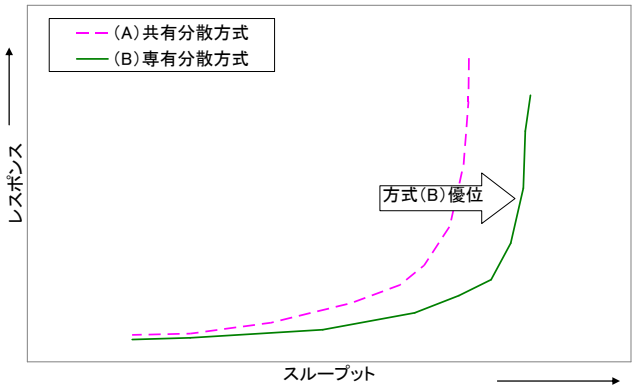

図 **5**　システム性能比較

2.3. 目標

上述の通り，方式（A）（B）単独では，1 ポート性能，システム性能いずれかの効率が悪い．そこで，本稿における負荷分散方式の目標を以下の両方を満たすこととした．

1 ポート性能：（A）共有分散方式同等

システム性能：（B）専有分散方式同等

3. 提案方式

方式（A）（B）の長所と短所は相反するため，2 方式を組み合わせて補うアプローチを採った．

3.1. 組み合わせ検討

組み合わせたときのベース方式として，方式(A)(B)を比較した結果，方式（B）をベースとして採用した．なぜならば，方式（B）が低負荷レスポンスに関して優位であるためである．これは，方式（A）は通信によって，スループットの低下だけではなくレスポンス遅延も発生するためである（図 6）．

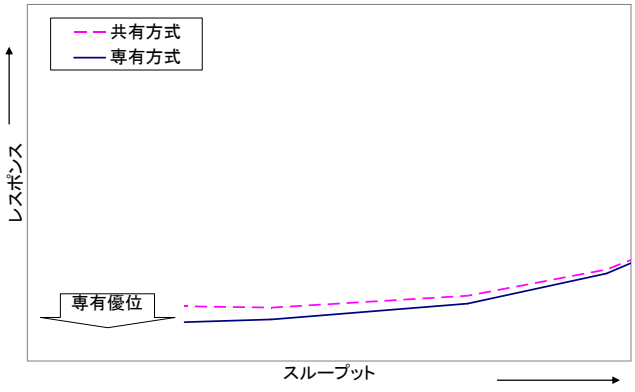

図 **6**　低負荷レスポンスの比較

専有をベースとして，切換え段階の有無で，以下の 2 つの組み合わせ方式を考えた（図 7）．

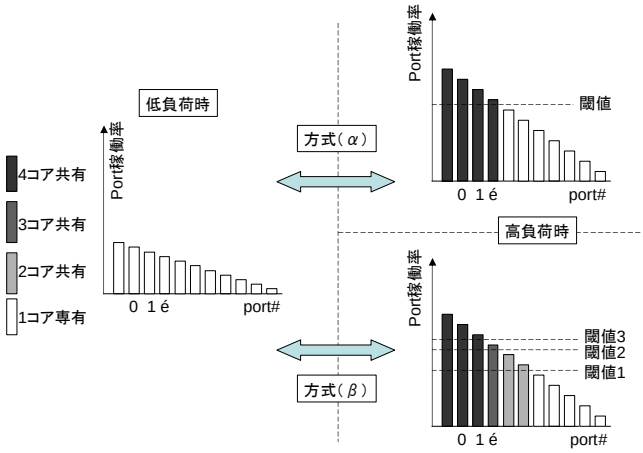

図 **7**　組み合わせ方式

（α）専有ベース一段切換方式：専有分散方式をベースとする．負荷が一定以上のポートを全コアでの共有分散方式にする．

（β）専有ベース多段切換方式：専有分散方式をベースとする．ポートの負荷に応じたコア数での共有分散方式にする．最大全コアでの共有分散方式にする．

方式（α）（β）をレスポンス変動の観点で比較した．方式（α）のように一段切替えで 1 コア専有と n コア共有で切り替える場合，レスポンスが急激に変動する．これは切換え前後で稼働率が大幅に変化するためである．これを性能特異点と呼ぶ．このような特性を持つストレージシステムでは，ホストの負荷が下がると，レスポンスが急激に悪化するという状態が発生し，逆に，ホスト負荷を上げないとレスポンスが良くならないといったことになる．それに対し，方式（β）のように多段で 1 コア専有→2 コア共有→…n コア共有で切り替える場合，大きくレスポンス変動がなくなる．多段切替え方式が優位である．

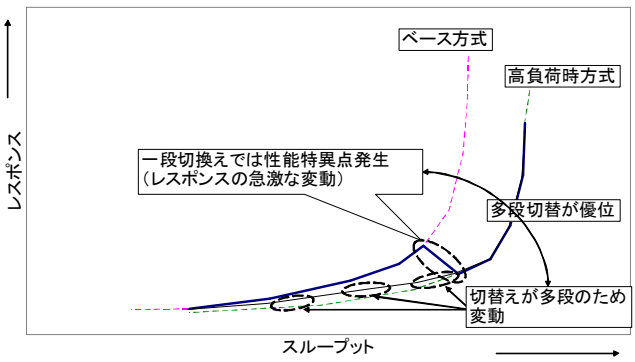

図 **8**　レスポンス変動の比較

この結果から，方式（β）を採用することとした．

4. 性能評価

4.1. 検証方法

下記 3 項目について，提案方式である専有ベース多段切換方式を検証した．

(a)　1 ポート性能

(b)　システム性能

(c)　ポート間負荷偏り時性能

（a）(b) のベンチマークで提案方式が目標を達成することを確認し，（c）で実環境においても有効であることを示す．

（A）共有分散方式，（B）専有分散方式を比較対象とした．各方式を実装したストレージシステムのプロトタイプを作成し，実機測定を行い，その結果をプロットしたレスポンスカーブを比較した．性能検証として，ポート負荷分散方式の影響が最も大きいライトコマンドのキャッシュヒット（ライトヒット）パターンを用いた．

4.2. 1 ポート性能

1 ポート性能を比較する実機測定構成は，表 2 の通りである．提案方式が共有分散方式と同等の結果になり，1 ポート性能の目標を達成することを確認する．

表 **2**　**1** ポート性能測定構成

システム内コア数	4
入力パターン	ライトヒット
入力ホストポート数	1
I/O 割合	100%

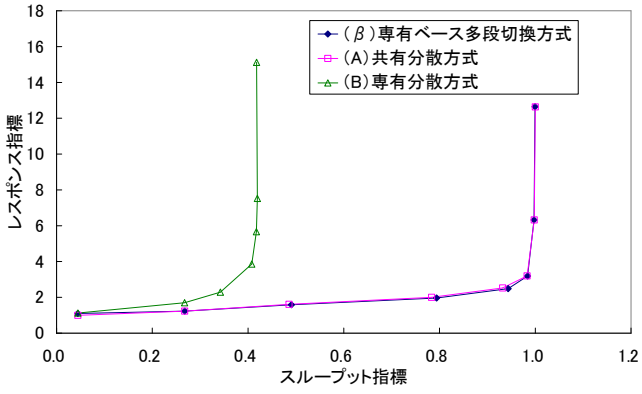

図 **9**　**1** ポート性能実機測定結果

図 9 は，1 ポート性能実機測定結果を示す．横軸は共有分散方式の最大スループットを 1 とした相対スループットであり，縦軸は共有分散方式の最小レスポンスを 1 とした相対レスポンスである．図からわかるように，提案方式は共有分散方式同等の性能であることを確認できる．また，性能特異点がないことも確認できる．

4.3. システム性能

システム性能を比較する実機測定構成は，表 3 の通りである．提案方式が専有分散方式と同等の結果になり，システム性能の目標を達成することを確認する．

表 **3**　システム性能測定構成

システム内コア数	4
入力パターン	ライトヒット
ホストポート数	16
I/O 割合	100%

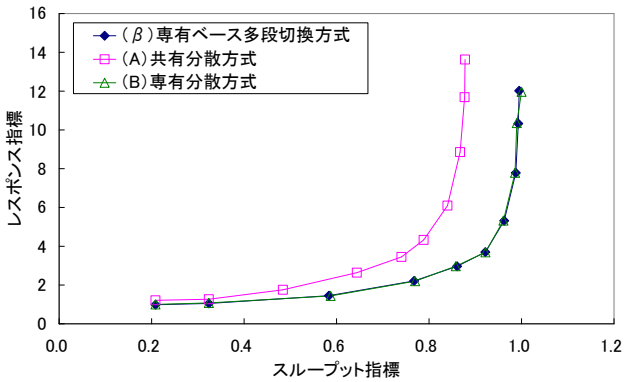

図 **10**　システム性能実機測定結果

図 10 は，システム性能の実機測定結果を示す．横軸は専有分散方式の最大スループットを 1 とした相対スループットであり，縦軸は専有分散方式の最小レスポンスを 1 とした相対レスポンスである．図からわかるように，提案方式は専有分散方式と同等の性能であることを確認できる．

4.4. ポート負荷偏り環境性能

ポート間負荷偏りがある環境を比較する実機測定構成は，表 4 の通りである．ポート間負荷偏りがある環境として，1 ポートに高負荷 IO，8 ポートに低負荷 IO を掛けた．高負荷ポートと低負荷ポートの負荷の比率は，高負荷ポートが 7 に対して，低負荷 8 ポート全体が 1 の割合とした．提案方式が共有分散方式および専有分散方式より優れていることを確認する．

表 **4**　ポート負荷偏り測定環境

システム内コア数	4	
入力パターン	ライトヒット	
入力ポート数	1	8
I/O 割合	87.5%	12.5%

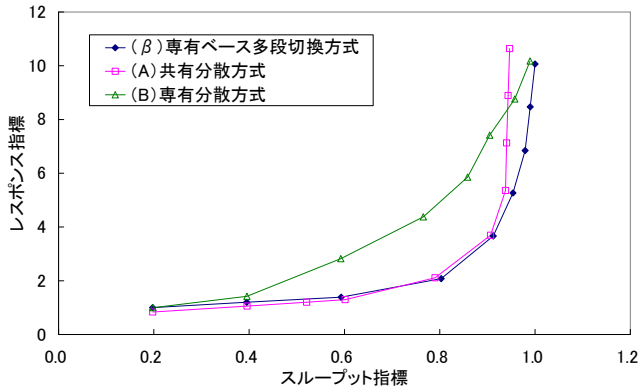

図 **11**　ポート負荷が偏ったときの比較

図 11 は，表 4 の構成での測定結果を示す．専有ベース多段切換方式の最大スループットを 1 とした相対スループットであり，縦軸は専有ベース多段切換方式の最小レスポンスを 1 とした相対レスポンスである．高低負荷ポートの負荷の比率を変えずに，レスポンスタイムが低い点から徐々に負荷を増やしている．図からわかるように，提案方式は，共有分散方式および専有分散方式より優れていることがわかる．専有分散方式は，高負荷ポート担当のプロセッサがネックとなるため，全体として中負荷時からレスポンスが悪化している．共有分散方式は，プロセッサネック時の処理効率が悪いため，プロセッサネック性能が低下している．また，提案方式に性能特異点が発生していないことを確認できる．

5. まとめ

近年，高性能なマルチコアプロセッサの SMP の普及により，プロセッサの主記憶上で外部からの入力を共有し負荷分散する方式が有効となってきている．本稿では，ストレージシステムの I/O 要求をプロセッサの主記憶で共有したときの評価を行った．ストレージシステムでは，高性能マルチコア普及前のポート負荷の分散はシステム外，専用ハード，プロセッサ間通信方式で行ってきた．しかしながら，これらの方式はコストや効率の面で課題があった．そのため，マルチコアプロセッサと主記憶の SMP 機能を検討した．

複数のポートの負荷を複数のプロセッサコアで分散する方式として，（A）共有分散方式，（B）専有分散方式を比較検討した．その結果，方式（A）が 1 ポート性能で，方式（B）がシステム性能で，優れていることがわかった．そのため，1 ポート性能は方式（A）同等，システム性能は方式（B）同等を目標とし，両方を満たす方式を検討した．

方式（A）（B）の長所を活かすため，両方式を組み合わせることを検討した．低負荷レスポンスとレスポンス変動の観点から，方式（B）をベースとし，共有するプロセッサ数を段階的に増やす専有ベース多段切替方式を提案した．

提案方式を検証するため，提案方式，方式（A），方式（B）を，1 ポート性能，システム性能，1 ポート高負荷他ポート低負荷混在性能の 3 条件でプロトタイプシステムによる実機検証を行った．実機検証の結果，提案方式の 1 ポート性能は，目標とした方式（A）と同等であり，システム性能は，目標とした方式（B）と同等であり，目標を達成できたことを確認した．また，高負荷低負荷混在時の性能は，方式（A）（B）より優位であることを確認した．

提案方式では，過去一定時間のポートの負荷を基にポートの担当を決めているため，プロセッサ性能の限界を超えた I/O を考慮できない．このような状況に対して有効な方式を検討する必要がある．

参　考　文　献

[1] 喜連川優．ストレージネットワーキング技術，オーム社，東京，2005.

[2] 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．信学論（D-I），vol.J86-D-I，no.6，pp.375-388，June 2003.