HPC 向けストレージの省電力化を図るアクセス予知階層ストレージの予知成功確率改善手法と効果の検証

赤池　洋俊† 藤本　和久‡ 岡田　尚也‡ 三浦　健司‡ 村岡　裕明‡

（株）日立製作所 システム開発研究所† 東北大学電気通信研究所‡

1. はじめに

近年、IT 機器の消費電力は無視できないほど増加しており[1]、大きな問題となっている。ストレージシステムはその中でも多くの電力を消費するシステムの一つである。特にスーパーコンピュータと接続するストレージシステムには大量のデータを高速に入出力することを目的として高い性能が要求される。そのため、性能を維持しながら消費電力を削減するストレージアーキテクチャと、その管理方式が求められている。

2. アクセス予知階層ストレージ

この背景の下で、図 1 に示す様に高性能なオンラインストレージ（以下、OL）と大容量のニアラインストレージ（以下、NL）の階層構成においてアクセス予知(図 1 中②)に基づくデータ配置(図 1 中④)と電源 ON/OFF 制御(図 1 中③①)を行う低消費電力化方式を提案した[2]。さらに、提案方式を試作機に実装し、実際に消費電力を測定することで省電力効果を検証した。その結果、階層ストレージにおいて使用頻度に基づくデータ管理とディスクのスピンダウン制御を行う従来方式と比較して、提案方式はシステム容量 1024TB の場合の試算で性能を維持しながら消費電力を 50%以上削減する見込みを得た[2]。この試作機をアクセス予知階層ストレージと呼ぶ。

アクセス予知には、キューアクセス予知方式を用いている。これは、計算機管理サーバのスケジューラ情報やジョブ情報をヒントにして、ジョブのアクセス先データの特定とジョブ実行開始までの時間的余裕の予測を行う。図 2 に示す通り、ジョブがジョブスケジューラに投入された時、アクセス予知に基づきジョブのアクセス先の NL ディスクを電源 ON し、ファイルを NL から OL へコピーすることでデータ配置する。この時間の合計を T_{Copy} と呼ぶことにする。ジョブがキュー内で待機する時間 T_{Wait} の間にコピーが完了すれば($T_{Wait} \geqq T_{Copy}$)、CPU でジョブが実行した時に高速な OL 上のデータにアクセスできる。これをアクセス予知成功と呼ぶ。一方で $T_{Wait} < T_{Copy}$ の時は、低速な NL 上のデータにアクセスするため、データ転送性能が低下してしまう。これをアクセス予知失敗と呼ぶ。

これまでは、省電力効果の確認のため、予知が 100%成功する理想的なジョブ投入条件で実験していた。ランダムなジョブ投入条件下では、予知失敗による性能低下が発生するため、キューアクセス予知方式の予知成功確率改善が課題となっていた。

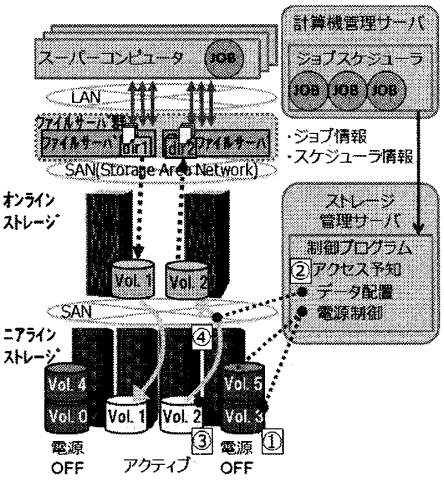

図 1. 提案手法の概要

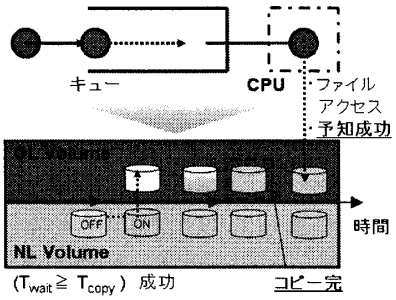

図 2. キューアクセス予知方式

3. 予知成功確率改善手法

アクセス予知失敗が発生するのはキューに並ぶジョブが少ない時である。特にキューが空だった時、投入ジョブはすぐに実行に移るので OL へのファイルのコピーが間に合わない。そこで、予知成功確率改善手法として、ジョブ実行遅延操作を提案した[3]。ジョブ実行遅延操作は、T_{Copy} 分だけジョブの実行を遅延するようにジョブスケジューラに指示を出し、その間にコピーを完了する。たとえジョブがキューの先頭で実行可能な状態であっても、T_{Copy} 分はキュー内で待機することになる。

しかし、その一方で投入と同時にジョブを実行遅延すると、待ちジョブのデータを全て OL にコピーすることになるため、待ちジョブ数が増えると OL に必要な容量が増大する問題が生じる。OL は高速な反面、容量あたりの消費電力が多く、システム全体の省電力効率が低下してしまう。そこで、防止策としてキューおよび CPU 内のジョブが少ない時はジョブ実行遅延操作とデータ配置を同時に実施し（図 3 上図）、ジョブが多い時にはジョブ実行遅延操作をせず、ジョブが k_{th} 番目になった時にデータ配置する（図 3 下図）という制御を行う。k はキューと CPU

The Verification of The Method to Improve Success Probability of Access Prediction for an Energy-efficient High Speed Tiered-Storage System (eHiTS) with Proactive Migration for HPC Systems.

† Hirotoshi Akaike, Systems Development Laboratory, Hitachi, Ltd.

‡ Kazuhisa Fujimoto, Naoya Okada, Kenji Miura, Hiroaki Muraoka, RIEC, Tohoku University.

内の合計ジョブ数、k_{th} は設定可能な閾値である。

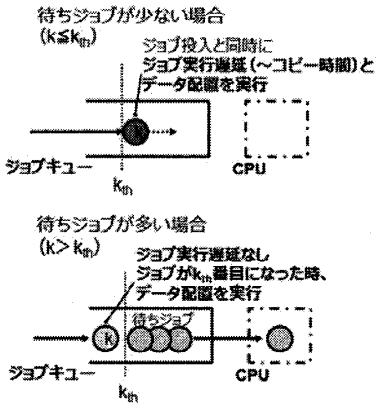

図 3. ジョブ実行遅延操作

4. ジョブ実行遅延操作の実装と効果の検証

予知成功確率改善手法の実証を目的として、試作機の制御プログラムにジョブ実行遅延操作を新規追加した(図 4)。新しいジョブ投入コマンド newsubmit を利用して、ユーザの送信したジョブを計算機管理サーバのジョブスケジューラにバイパスする実装とした。スケジューラへの指示にはジョブ投入コマンド submit に ジョブ待機オプションを用いることで、ジョブ投入時に遅延時間を設定する。ユーザ側の操作の変更点は submit の代わりに newsubmit コマンドを実行するだけである。

ジョブ実行遅延操作の動作確認として、表 1 の条件で実験を行った。今回は簡単のため、k_{th}=1 で固定とした。さらに、比較のためジョブ実行遅延操作が無い場合で、同じ条件の実験を行った。J. Jann らにより、スーパーコンピュータで実行されるジョブの投入間隔と実行時間は超アーラン分布に従うことが示されている[4]。今回は、簡単のため投入間隔と実行時間にアーラン分布に従うランダムな値を指定した。実験の結果、投入した 33 ジョブ中、ジョブ実行遅延操作なしの場合は 6 ジョブがアクセス予知に失敗し、アクセス予知成功確率は 82%(=1-6/33)であった。ジョブ実行遅延操作あり（k_{th}=1）の場合は全ジョブでアクセス予知に成功した。最悪の場合、次の 34 番目のジョブで失敗したとして、アクセス予知成功確率は 97%(=1-1/34)となり、15%(=97%-82%)改善した。一方で、ジョブ実行遅延による CPU 利用率低下は高々4%程度であり、提案方式の有効性を確認した。図 5 に実験で測定したアクセス予知失敗確率とシミュレーションの結果を示す。k_{th} を大きく設定すれば、アクセス予知失敗確率は十分小さくなる見込みである。実験結果とシミュレーションに差があるが、これは実験で用いたジョブ数が 33 と少ないことによる誤差と考えられる。

5. まとめ

アクセス予知階層ストレージの予知成功確率改善手法であるジョブ実行遅延操作を実装し、動作確認を行った。実験の結果、ジョブ実行遅延操作は正常動作し、アクセス予知成功確率の向上を確認した。今後は、スーパーコンピュータのワークロードを調査し、ジョブ投入間隔と実行時間をさらに現実に近い設定にするとともに、長時間の実験を行うことでアクセス予知成功確率を詳しく評価していく。

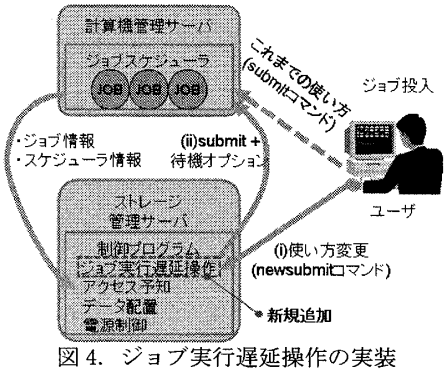

図 4. ジョブ実行遅延操作の実装

表 1. 実験条件

実験条件	設定
実験時間	36 時間
初期状態	最初に実行時間 1 時間のダミージョブを 3 つキューに投入しておく（初期状態でジョブ実行予知が成功する設定）
コピー時間 T_{Copy}	27[min](コピーはボリューム単位で行うものとし、コピー時間は全ジョブで同じとする。)
ジョブ投入間隔	平均投入間隔=1job/60[min] (アーラン分布からランダムにサンプリング)
ジョブ実行時間	平均実行時間=55[min]/1job (アーラン分布からランダムにサンプリング)

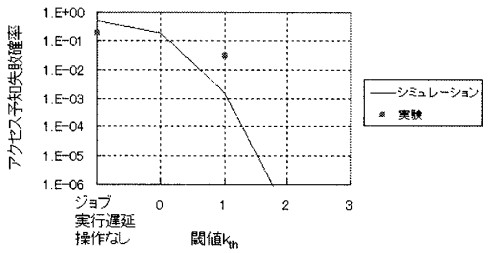

図 5. 実験結果

謝辞 本研究は、文部科学省の委託研究「高機能・超低消費電力スピンデバイス・ストレージ基盤技術の開発」の成果の一部である。

参考文献

[1] “Report to Congress on Server and Data Center Energy Efficiency Public Law 109-431”, U.S. Environmental Protection Agency, ENERGY STAR Program, Aug. 2007.

[2] 赤池洋俊, 藤本和久, 岡田尚也, 三浦健司, 村岡裕明, “HPC 分野向け高速・大容量ストレージシステムの省電力化を図るアクセス予知階層ストレージの試作と省電力効果の検証”, 第 8 回情報科学技術フォーラム, 2009 年 9 月

[3] 岡田尚也, 藤本和久, 赤池洋俊, 三浦健司, 村岡裕明, “アクセス予測を利用した HPC 向け高速大容量階層ストレージの階層管理方式の予測精度向上手法に関する検討”, 第 8 回情報科学技術フォーラム, 2009 年 9 月

[4] J. Jann, P. Pattnaik, H. Franke, et al, “Modeling of Workload in MPPs.”, LNCS, Vol 1291/1997, pp. 95-116, 2006.