C-014

HPC 向け高速大容量・低消費電力ストレージのための 負荷分散手法に関する検討

Study on load-balancing in high-speed and mass storage systems for HPC systems with access prediction

黒川 大樹† 藤本 和久† 赤池 洋俊‡ 三浦 健司† 村岡 裕明†
Hiroki Kurokawa† Kazuhisa Fujimoto† Kenji Miura† Hiroaki Muraoka†

1. はじめに

近年の情報爆発により、クラウドコンピューティングといった新たなストレージの形態が出現している。このような背景により、企業などで設置しているデータセンタでは日夜、大量のデータを処理するようになった。そのため、大容量のハードディスクドライブ(以下 HDD)を大量に利用することとなり、データセンタのストレージにおける消費電力の増大が問題となっている。

例えば、米国におけるデータセンタの総年間消費電力量は 610 億 kWh(2006 年)で、電力コストは 45 億ドルにもなる[1]。経済的な面や環境負荷への配慮から、消費電力の削減が早急に求められている。

現在に至るまでストレージの消費電力の削減については、Massive Arrays of Idle Discs[2](以下 MAID)や階層ストレージなど様々なものが研究されてきた。しかし、これらの手法の問題点として、省電力と応答性能はトレードオフの関係となっていることが挙げられる。このことから、これらの手法の使用はバックアップやアーカイブといった用途に限られてきた。例えば、MAID について単位時間あたりのアクセス回数が多くない環境では省電力面で有効であるが、スピンダウンされているドライブに保存されているデータにアクセスが発生した場合に、読み出しを行う前にドライブをスピンアップする必要があり、スピンアップ時間の分だけ応答性能を損なう。

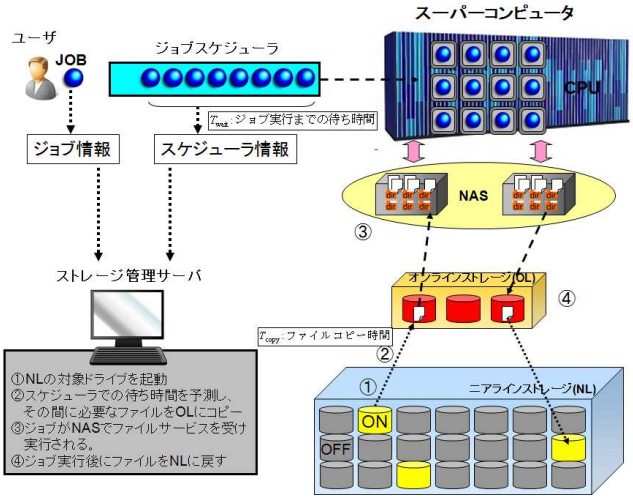

Fig. 1 提案省電力手法の概要図

† RIEC, Tohoku University.
‡ Yokohama Research Laboratory, Hitachi

この課題を解決する手法として、我々は待ち行列理論によるアクセス予測を利用したストレージシステムを提案している[3]。Fig. 1 に我々が提案している高性能と省電力を両立させる手法の概要を HPC(High Performance Computing)システムへの適用を例に挙げて示す。この手法では高速･高消費電力ドライブを用いたオンラインストレージ(以下 OL)と低速･低消費電力ドライブを用いたニアラインストレージ(以下 NL)の階層構造を利用する。従来の階層ストレージでは頻繁に利用されるファイルや作成されてから間もないファイルを OL に、作成後時間が経過し頻繁にアクセスされることのないファイルを NL に配置するのに対し、提案手法では全てのファイルを NL 側のドライブに保存しておき、アクセスのない NL のドライブは電源を落としている。

計算を行う際、ユーザは計算処理の内容を記述したジョブスクリプトをスーパーコンピュータシステムに投入する。ジョブを管理する機構をジョブスケジューラと呼ぶ。提案手法では、ジョブスクリプトからジョブの実行に必要なファイルが格納されているドライブの情報を読み取り、その情報をストレージ管理サーバへ送る。ストレージ管理サーバでは、ジョブ投入から実行開始までの待ち時間(T_{wait})を予測し、T_{wait} の間に NL の中からジョブ実行に必要なファイルが格納されているドライブだけを起動し、OL 側ドライブへのコピーを行う。NL から OL へのコピーに必要な時間を T_{copy} とおいたとき、$T_{wait} \geq T_{copy}$ となるようコピーを行えば、実行開始までに OL にファイルの配置が完了しており、ジョブ実行の際には高速なファイルの読み出しが可能となる。また、ジョブ実行終了後、ファイルは NL に戻し、ファイルを格納したドライブの電源を落とす。この手法を用いることによりシステム全体の消費電力を従来のシステムに比べ、50%以上削減することが可能であることを示唆している[4]。

この手法に関して、ファイルの読み出しや、書き込みのサービスを提供する専用サーバとして Network Attached Storage(NAS)のデータ転送性能について検討した結果、スーパーコンピュータからのアクセス負荷に関する課題が存在することが明らかになった。提案手法のシステムの試作機では NAS を複数台用意している。ジョブとファイルサービスは 1 対 1 に対応しており、ファイルサービスはユーザが利用するデータボリュームの数だけ用意している。NAS にはユーザが利用する全ファイルサービスを均等に配置しているが、各ユーザはジョブをランダムに投入するため、スーパーコンピュータで実行されるジョブがアクセスするファイルサービスもランダムとなる。したがって、Fig. 2 で示すようにファイルサービス(Fig. 2 内の dir)を初期状態で均等に配置しても各 NAS でアクセスされるファイルサービス数(=負荷)は時間

と共に変化し、青の矢印で示すように、2 つの NAS の負荷が均等になる時間帯もあれば、赤の矢印で示すように特定の NAS に偏る時間帯もあり、各 NAS の負荷は時間的に変化する。1 つの NAS のデータ転送性能には限界があるため、青矢印で示される状態で最大の転送性能が得られるとしたとき、赤矢印のような負荷が偏った場合では、片方の NAS のアクセスが減少した分だけ転送性能が低下してしまう。

そのため、ジョブスケジューラと連携した新たな負荷分散手法を提案し、提案した負荷分散アルゴリズムの評価をシミュレーションによって行った。

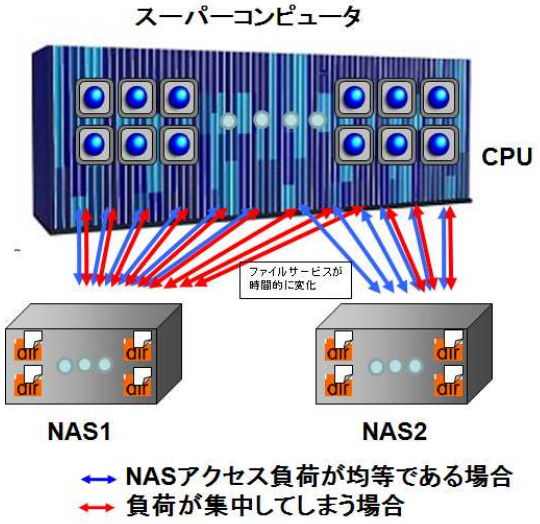

Fig. 2　NAS ノードに負荷が集中する場合

2. 提案負荷分散手法

2.1 スケジューラ連携負荷分散

従来の負荷分散の手法は、ロードバランサと呼ばれる専用の装置を用い、負荷が変化した後に負荷分散を行うため、ジョブの実行開始後に負荷分散を行うことになる。スーパーコンピュータでは、計算されるジョブの入れ替わりとともに NAS へのアクセス負荷が変化し、ジョブ投入直後にファイルの読み込みが行われるため、従来手法ではデータ転送性能の低下を招いていた。そこで、ジョブスケジューラと連携する事でジョブ実行開始前に負荷分散を行う手法(スケジューラ連携負荷分散)を提案した。スケジューラ連携負荷分散方式の模式図を Fig. 3 に示す。このスケジューラ連携負荷分散では、ジョブスクリプトからジョブ実行前に予め入出力ファイルへのアクセスを提供するファイルサービスを割り出し、そのファイルサービスがどの NAS にあるかを特定することで負荷を計算し、当該ジョブ実行時の NAS の負荷状態の予測を行う。予測の結果、Fig. 3 の赤矢印で示されるように NAS 間で負荷の偏りがある場合には、負荷の高い NAS から低い NAS へファイルサービスを移動させることで、青矢印で示されるように負荷の均衡化を図る。

スケジューラ連携負荷分散手法の特徴はジョブの実行前に予め負荷分散を行うことである。このため、NAS に対する負荷を計算し、負荷分散を行うタイミングが重要である。負荷分散を行うタイミングが遅く、ジョブ実行開始までの間に NAS 間のファイルサービス移動を行う時間的余裕が十分でない場合、ジョブの実行を待たせることになり、スーパーコンピュータの CPU 率の低下やユーザの待ち時間の増加を引き起こす可能性がある。

この負荷分散のタイミングと負荷分散の正確性の観点から、3 つのアルゴリズムを検討した。

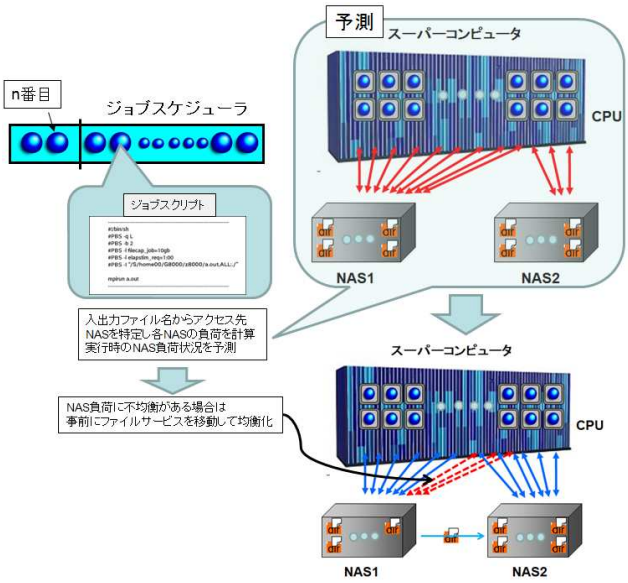

Fig. 3　スケジューラ連携負荷分散

2.2 負荷分散アルゴリズム

負荷分散のタイミングについて、基本的にキュー内でジョブの順番が前から n 番目になったときに負荷分散を行うとした。この時、以下の 3 つのアルゴリズムによって負荷分散を行うことを考えた。

1 つ目は、n=1 と設定し、実行中のジョブが終了したとき、他の実行中のジョブのアクセス先 NAS から、NAS 毎のファイルサービス数を計算し、当該ジョブのアクセス先 NAS が負荷の最も低い NAS でなかった場合に、そのアクセス先を負荷の最も低い NAS へ変更するというアルゴリズムである(以下、アルゴリズム 1 とする)。このアルゴリズムではジョブ実行直前に負荷分散を行うことにより高精度な負荷分散が可能になると考えられるが、NAS 間でファイルサービスを移動させるのに時間を要し、ジョブの実行を待たせるため、CPU 利用率やユーザの待ち時間の増加を引き起こすことが懸念される。

2 つ目は、当該ジョブがキュー内で前から n 番目になった場合に、前に並んでいる n-1 個のジョブがアクセスする NAS から各 NAS の負荷を算出し、当該ジョブのアクセス先 NAS のアクセス負荷が予め設定された閾値を超えている場合、他の NAS ノードの中で最も負荷の低い NAS ノードに当該ジョブのアクセス先 NAS を変更する(以下、アルゴリズム 2 とする)。

3 つ目は、当該ジョブがキュー内で前から n 番目になった場合に、当該ジョブがアクセスする NAS の負荷が閾値

を超えてない場合でも常に小さい NAS ノードを選択するというアルゴリズムである(以下、アルゴリズム 3 とする)。

アルゴリズム 2,3 について、n を大きくとると、よりキュー内の早い段階から負荷分散を行うことになり、ファイルサービス移動の時間的余裕を大きく取れることになる。また、ジョブ実行開始までジョブ投入時にジョブスケジューラ内のジョブが m 個(m<n)の場合は投入と同時に、1～m-1 番目の負荷を計算し、同様の操作を行うとした。

この 3 つのアルゴリズムについて、負荷分散の正確性の尺度としてデータ転送速度とユーザの待ち時間の増加の観点から比較を行い、検討した。

3. 検証方法

本検証ではイベント駆動型シミュレータとして Hyperformix 社製の Workbench5.3 を用いた。科学技術計算では、ジョブは投入間隔と実行時間が超アーラン分布に従うことが知られている[5]。このことから、投入間隔と実行時間については超アーラン分布を仮定して、パラメータを与え、10 万個のジョブを投入し、逐次実行した。ここで与える投入間隔と実行時間のパラメータは東北大学内で利用されているスーパーコンピュータのログから得られたものを用いた。

また、NAS の台数は 2 台で負荷=ファイルサービス数とした。また、実機検証のための試作機で使用している 2 台の NAS の性能が 1 台あたりファイルサービス数 6 で最大の性能となることから、CPU 数は 12 と設定している。この条件のもとでジョブの投入と実行をシミュレートし、2 台の NAS ノードの負荷と転送速度の時間変化を算出した。

4. 検証結果

まず、NAS ノードのアクセス負荷の偏りについて分かりやすく考えるために、負荷の偏りの大きさを NAS ノードのアクセス負荷が(NAS1:NAS2)=(6:6)の場合を基準にした場合に、各(NAS の負荷-6)の絶対値を足しあわせたものとして定義する。例えば(NAS1:NAS2)=(7:5)の時を 2、(8:4)を 4 と段階的に表し、(12:0)で最大の 12 をとる。NAS2 のアクセス負荷が大きい場合も同様である。全ての CPU リソースが利用されている時間での、負荷の偏りの大きさの累積度数分布を Fig. 4 に示す。Fig. 4 では線が左にシフトするに従い、負荷の均衡がとれた状態に近くなる。負荷分散をしない場合に比べ、アルゴリズム 1,3,2 の順に左に近付いている。特にアルゴリズム 1 では全ての時間帯において、大きさ 2 以内の偏りに抑えられており、平均の転送速度は 97%の性能であった。対して、スケジューラ内で負荷の計算を行うアルゴリズム 2,3 では、平均で 90%程度の転送性能となった。これは、負荷分散を早い段階から行うため、実行中のジョブの中で実行時間が特に長いものが存在した場合に、負荷の予測に利用したジョブが先に実行終了することになり、負荷の計算に利用しなかったジョブが CPU にとどまり続けることによって、負荷の偏りが起きやすいためであると考えられる。

2 つの NAS 間でファイルサービスを均等に配置し、ランダムにアクセスをするという場合でも NAS アクセス負荷が偏ってしまうが、実際にはスーパーコンピュータを扱うユーザの傾向によって、一時的にさらにアクセス負荷が偏ることが想定される。例えば、一部のヘビーユーザが特定の NAS にアクセスするファイルを集中して利用する場合などである。そのようなヘビーユーザを想定して、ファイルサービスの初期配置を変化させ、片方の NAS にアクセスを行う確率を変化させた場合の負荷分散シミュレーションを行った。結果を Fig. 5 に示す。横軸は片方の NAS にアクセスが起こる確率を示しており、もう片方の NAS にアクセスが起こる確率は(100-横軸)の値を取る。例えば、横軸が 70 の時点では、各 NAS にアクセスが起こる確率は(NAS1:NAS2)=(70:30)となる。横軸が大きくなるに連れて、負荷分散を行わない場合は平均転送速度も低下し、最終的には最大性能の半分程度まで転送性能が低下するという結果が得られた。それに対して、各アルゴリズムを実装した場合には、ファイルサービスの初期配置を変化させていっても転送速度は横ばいとなり、ヘビーユーザの利用といった極端な場合を想定しても、各負荷分散アルゴリズムには影響を与えず、90%以上の平均転送速度を維持し、安定した転送性能を得られることが確認できた。

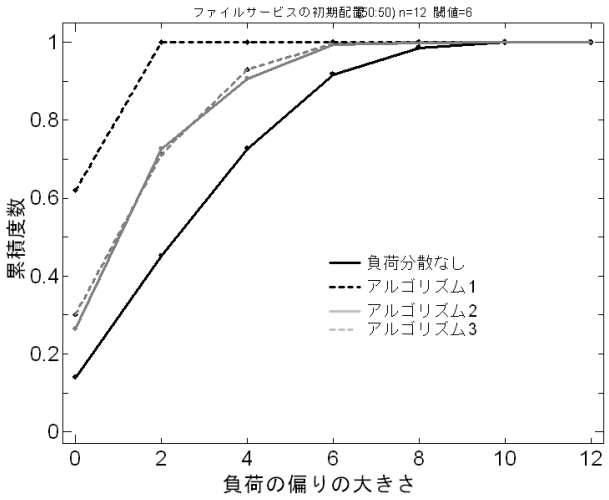

Fig. 4　各アルゴリズムの累積度数分布による比較

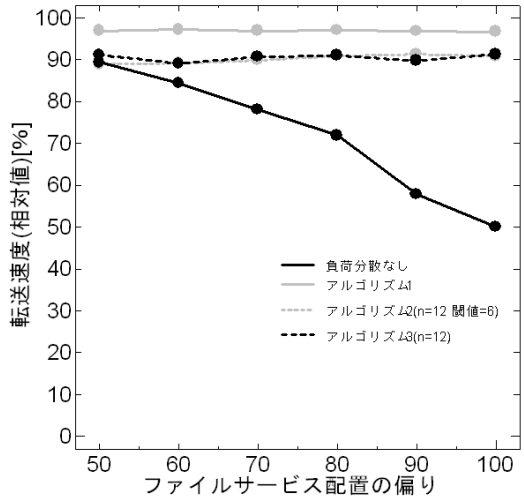

Fig. 5　ファイルサービスの初期配置条件と転送速度の関係

次に、負荷分散のタイミングが転送性能に与える影響についてのシミュレーションを行った。負荷分散のタイミングを司る n を変化させた場合について、Fig. 6,7 に示す。n を大きくし負荷分散を早くから行うことによって負荷分散の精度への影響が懸念されたが、実際にはアルゴリズム 2,3 とも n を大きくすることによる影響はごく僅かであることが確認できる。

最後に、アルゴリズム 2,3 を適用することによる平均待ち時間の増加について Fig. 8 に示す。この時のファイルサービスの初期配置は均等配置であり、アルゴリズム 2,3 ともに n は 12、アルゴリズム 2 での閾値は 6 に設定している。この条件でのアルゴリズム 2,3 の待ち時間の増加は、アルゴリズム 1 に比較してそれぞれ、1/4,1/2 程度で抑えられている。アルゴリズム 2 の方が 3 に比べ大幅に待ち時間の増加が小さいのは、ファイルサービスの移動の回数を少なく抑えることができるためであると考えられる。

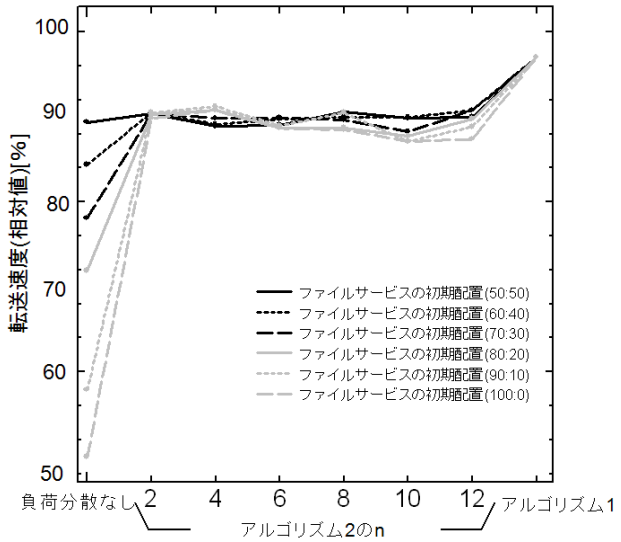

Fig. 6　負荷分散のタイミングが転送速度に与える影響 (アルゴリズム 2)

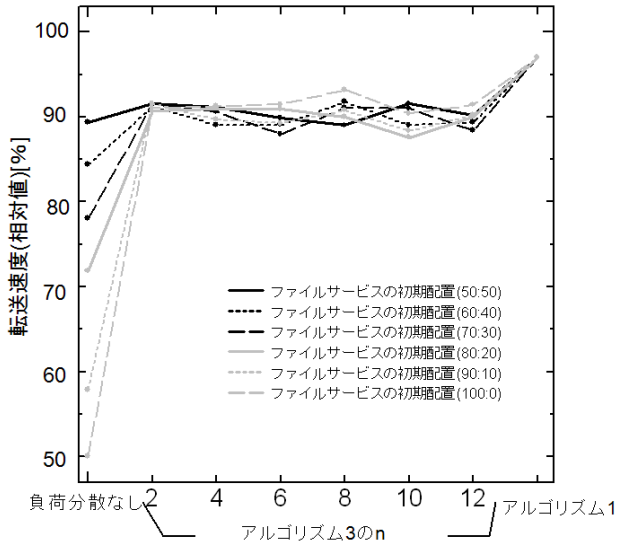

Fig. 7　負荷分散のタイミングが転送速度に与える影響 (アルゴリズム 3)

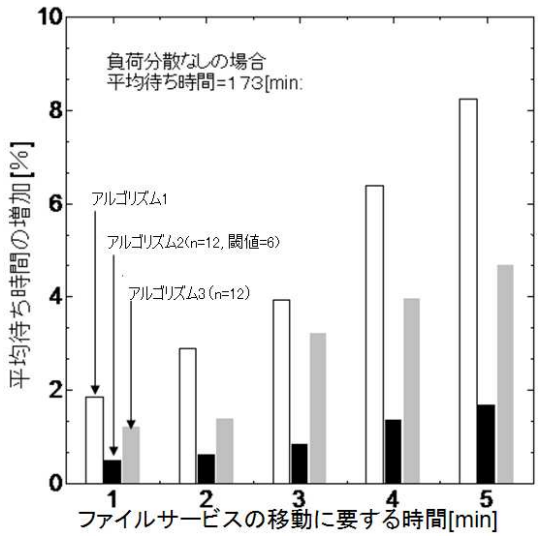

Fig. 8　各アルゴリズムによるユーザ待ち時間の増加率の比較

5. まとめ

本検証では、提案ストレージシステムのファイルサービスに関する課題の改善の為の負荷分散アルゴリズムに関する検証を行った。アルゴリズム 1 では平均 97%の転送速度が得られるが、ファイルサービスの移動に応じたユーザの待ち時間の増加(2~10%程度)が起こる。アルゴリズム 2,3 では、負荷分散から実行開始まで時間的余裕を持つため、ユーザの待ち時間増加をアルゴリズム 1 と比較して、25～50%程度抑えられる。しかし、転送性能は 7%程度劣る。

したがって、ユーザの待ち時間のロスが少なく、高い転送性能を得られる負荷分散手法として、さらにアルゴリズム(例えばアルゴリズム 1 と 2 を組み合わせたものなど)を検討した上で、CPU リソース数、NAS ノード数を拡張してのシミュレーションを行っていく。

謝辞

本研究の一部は、文部科学省による次世代 IT 基盤構築のための研究開発「高機能・低消費電力スピンデバイス基盤技術の開発」の援助を得て行いました。

参考文献

[1]U.S. Environmental Protection Agency, “report to Congress on Server and Data Center Energy Efficiency Public Law 109-431”, ENERGY STAR Program (2007)
[2]Dennis C, Dirk G and Michael N, “The Case for Massive arrays of IdleDisks(MAID)”, Boulder (2002)
[3]Kazuhisa Fujimoto, Hirotoshi Akaike, Naoya Okada, Kenji Miura, Hiroaki Muraoka, “ Power-aware Proactive Storage-tiering Management for High-speed Tiered-storage Systems”, Sustain IT '10
[4]赤池 洋俊, 他 “HPC 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)
[5]J. Jann, P. Pattnaik, H. Franke, et al, “Modeling of Workload in MPPs”, LNCS, Vol1291/1997, pp.95-116, 2006.