1H-05

HDFS シーケンシャルファイルアクセス性能の向上に関する考察

近丈一郎†1　　中島健司†2　藤島永太†2　山口実靖†12

工学院大学 工学部 情報通信工学科†1　　工学院大学大学院 工学研究科 電気・電子工学専攻†2

1 はじめに

インターネットの普及に伴い，世界で扱われるデータが増加している．特に，医療・ライフサイエンスに代表されるセンシティブなデータを安心して扱うことのできる，セキュアなコンテンツ共有・流通基盤の構築が必要不可欠である．このコンテンツ共有・流通を行うためには，暗号技術の併用が欠かせない．しかし，大規模なデータを全て暗号化すると，データ量が大幅に増加し，実用的な処理時間を実現するためには大規模なデータ処理の I/O 速度の向上が重要となる．

膨大なデータを処理する技術の一つとして，並列分散処理を行う Hadoop がある．Hadoop の I/O 性能を向上させる手法として，ファイル格納位置の動的制御[1] がある．この手法は，HDD のシーケンシャル I/O 速度の速いディスクの外周部のゾーンへファイルを積極的に格納することにより，シーケンシャル I/O 速度を向上させる．実装は，Ext2 または Ext3 ファイルシステムを用いて行われ，シーケンシャル I/O 速度の遅いディスク内周部のディスク使用状況を使用中に書き換えることで，外周部への新規ファイル格納を実現している．しかし，ディスク外周部に格納されているファイルが削除された際に作られる空き領域に対して，それを有効活用するための考慮はされていない．

本稿では，Hadoop のジョブ実行中に作成された中間データや一時ファイルが削除された際に作られる空き領域を考慮した動的制御手法を提案し，性能評価を行い，その有効性を示す．

2 ファイル格納位置の動的制御手法

ファイル格納位置の動的制御手法[1]では定記録密度方式の HDD のシーケンシャル I/O 速度が内周側より外周側のゾーンが速いという特性を利用[2]し，ファイル格納位置の動的制御により優先的にファイルを外周側のゾーンへ格納することでシーケンシャル I/O 速度の向上を図っている．

実装方法はオープンソースファイルシステムである ext2 および ext3 を用いて行っている．このファイルシステムでは，ディスクは 4KB のブロックを単位に管理され，複数のブロックを集めてブロックグループを構成する．そして，ブロックグループ毎に各データブロックが使用中であるか未使用であるかを管理するブロックビットマップがあり，これらのブロックビットマップの内，シーケンシャル I/O 速度の速いディスクの外周側以外のブロックビットマップを使用中ビットに書き換え，各時点で必要とされる容量のみを使用可能領域とし，必要量の増加に応じて使用可能領域を動的に拡張することで，ディスク外周部のシーケンシャル I/O 速度が高速な領域を効率的に利用している．

動的制御手法では，定期監視間隔ごとにディスク内の空き領域サイズを監視し，空き領域サイズが動的拡張の閾値を下回ることが検出されると，この閾値を超えるまで使用禁止領域を使用可能領域に変更していく．

3 提案手法

前章の動的制御手法によりディスクの外周側のゾーンに優先的にファイルが格納される．しかし，当該手法はディスク空き容量不足時に空き容量を拡張するのみであり，過剰な空き領域が存在する状況にてそれらの一部を使用不可として外周側を積極的に使用させることはしていない．よって，ジョブ実行中に作成された中間データや一時ファイルが削除され，再びジョブが実行される場合，過剰な空き領域が存在しその中における低速領域(内周部)が使用されてしまう可能性がある．

そこで，ファイルが削除され空き領域が増えた場合に空き領域を内周側から順に使用禁止とすることにより当該手法を更に拡張する手法を提案する．本稿では試作実装として，一度すべてのブロックビットマップを使用中ビットに書き換え，その後にディスクの外周側のブロックグループのから順に空き容量が閾値以上になるまで拡張していく実装を用いた．これにより，ファイルは外周側のゾーンに空き領域があればシーケンシャル I/O 速度の速い外周側へ書き込まれる．試作実装においては，極めて短期間であるがディスクの全領域が使用不可となる時間帯が存在し，その瞬間にファ

A Study on Performance Improvement of HDFS Sequential Access

†1. Joichiro Kon

†2. Nakashima Kenji, Fjishima Eita, Yamaguchi Saneyasu

†1. Department of Information and Communications Engineering, Kogakuin University

†2. Electrical Engineering and Electronics, Kogakuin University Graduate School

イル作成要求が発行された場合のみ正常にファイル格納領域を確保できない課題があり，その課題以外は提案手法で想定している通りの実装となっている．

4 性能評価

既存手法(動的制御手法)，提案手法にて TeraSort を実行し，それぞれの処理時間を比較した．

測定環境は，物理計算機 1 台で Hadoop は疑似分散環境で動作させ，入力データは 8 GB とした．測定に使用した物理計算機の仕様は表 1 の通りである．どちらの手法も定期監視間隔は 5 秒とし，動的拡張の閾値は 5 GB とした．また，提案手法では空き領域が 8 GB より大きくなったとき再動的制御を行うことにした．

測定結果を図 1 に示す．図より，初回の測定においては既存手法と提案手法の性能が同等であることが分かる．これは，提案手法がファイル削除後に過剰な空き領域が生じた場合にのみ動作するため，既存手法と提案手法の動作がほぼ同一となったためであると考えられる．２回目の測定結果に着目すると，提案手法の方が短い時間で TeraSort の処理を終えていることが分かる．このことから，既存手法は 1 回目の TeraSort の実行の後の一時ファイルの削除などにより過剰な空き領域を有してしまい，その領域の中で低速領域を使用してしまったが，提案手法では低速領域の使用を回避できたためであると考えられる．

両手法の 1 回目と 2 回目の性能を比較すると，既存手法では，1 回目より 2 回目の方が性能が悪化していることが分かる．このことから，既存手法は過剰な空き領域を有していない状況においては適切に動作できるが，ファイル削除などにより過剰な空き領域を有してしまった場合は性能が悪化してしまうことが分かる．一方，提案手法においては 2 回目においても 1 回目と同等の性能を得ることができており，1 回目の実行の後のファイル削除の等により生じた過剰な空き領域を適切に処理できていることが分かる．

表 1.測定用物理計算機の仕様

CPU	Intel Celeron CPU G530 2.40 GHz
OS	CentOS 6.5 x86_64 minimal
Kernel	Linux 2.6.32-431.el6.x86_64
HDD	500 GB
Main Memory	4 GB
Hadoop ver	2.7.2

表 2.測定用 HDD の仕様

型番	DT01ACA050
インターフェース	SATA 3.0
容量	500 GB
バッファ量	32 MB
回転数	7200 RPM

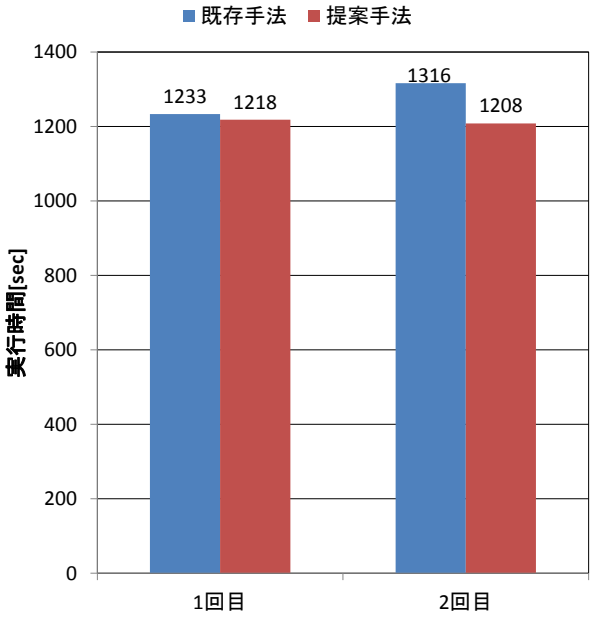

図 1TeraSort 実行時間

5 おわりに

本稿では，既存のファイル格納位置の動的制御手法を改変することで，ファイルの削除により生じたディスクの空き領域を再び動的制御し，常にシーケンシャル I/O 速度の速い外周側へファイルの書き込みをする手法を示し，性能評価により提案手法が有効であることを示した．

今後は，簡易実装の改善による一時的なディスク使用不可期間の排除などを行っていく予定である．

謝辞

本研究は JSPS 科研費 25280022，26730040，15H02696 の助成を受けたものである．

本研究は，JST，CREST，の支援を受けたものである．

参考文献

[1] Eita Fujishima, Saneyasu Yamaguchi, "Dynamic File Placing Control for Improving the I/O Performance in the Reduce Phase of Hadoop", the Tenth International Conference on Ubiquitous Information Management and Communication (IMCOM2016), 8-2

[2] Eita Fujishima, Saneyasu Yamaguchi, "Improving the I/O Performance in the Reduce Phase of Hadoop",CANDAR’15, 2015/12