D-017

ファイル格納位置制の動的制御による データ処理アプリケーション性能の向上
Performance Improvement of Data Intensive Applications with Dynamic File Storing Location Control

藤島 永太† 山口 実靖†
Eita Fujishima Saneyasu Yamaguchi

1. はじめに

分散処理システムの I/O 処理高速化手法として，ファイル格納位置制御手法[1]がある．当該手法では，ファイルシステムにおけるブロック使用情報を制御し，ストレージデバイスの高速領域を積極的に使用するとともに，データ間のシーク距離を削減しアプリケーション性能の向上を図っている．

本稿では，当手法を大規模データ処理アプリケーションに適用し，その性能向上手法について考察をする．

2. ファイル格納位置の動的制御手法

本章では，過去に提案したファイル格納位置の動的制御手法[1]について紹介する．

2.1. 手法の動作

本手法では，定記録密度方式の HDD のシーケンシャル I/O 速度が，ディスクの最外周である最低アドレス部付近と，最内周である最高アドレス部付近とで大きな差があることを積極的に利用する．一般的なファイルシステムがファイルを新規作成する際，作成するファイル容量に対して空き領域が大きい場合，空き領域内の比較的高いアドレスにファイルを作成してしまうことがある．この現象が大規模データ処理中に行われると，大きな性能低下に繋がる．

本手法では，空き領域内の低アドレス部の一定の領域(数 GB～数十 GB)以外を“使用禁止空き領域”として制限する．そして，制限されていない領域である“使用可能空き領域”にファイルが作成され，使用できる容量が減少すると，減少した容量だけ，使用禁止空き領域の外周側を使用可能へと開放していく．

また，本手法の動作は，使用可能空き領域の制限機能，使用可能空き領域の容量監視機能，使用可能空き領域の動的拡張機能の 3 つから構成される．

2.2. 手法の実装

本手法の実装は，著名なオープンソースファイルシステムである ext2 および ext3 を用いて行った．当該ファイルシステムは，4 KB のブロックを単位に管理され，連続したブロック群から複数のブロックグループを構成する．そのグループ毎に，ブロックビットマップ，inode ビットマップ，inode テーブルが用意されている．

ブロックビットマップは，グループ内の各データブロックの使用状態を管理するビットマップである．例えば，あるビットが“1”である場合，それに対応するデータブロックは使用中であると判断されるため，新たなファイル作成を行う際の作成する対象から除外される．本手法の使用可能空き領域の制限機能はこの基本動作を利用し，全空き領域内のシーケンシャル I/O 性能の高い低アドレス部分以外のデータブロックに対応するブロックビットマップ情報を，保管した上で“1”に書き換える．これにより，新たなファイル作成を行う際に性能の高い領域のみを使用する様に制御する．

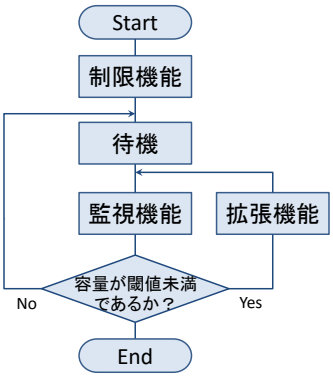

図 1 動的制御手法のフローチャート

使用可能空き領域の容量監視機能は，一定時間毎にファイルシステムの空きブロック数を監視する．この監視には，ext2 および ext3 の `s_free_blocks_count` を使用している．これにより得られた値と本手法により使用禁止としたブロック数を用いて使用可能容量を算出する．また，手法適用前に監視間隔の秒数と使用可能空き領域の閾値を決定しておく必要がある．使用可能容量の監視を行う際に，この閾値未満でないか判定を行う．もし，閾値未満であった場合，次に説明する使用可能空き領域の動的拡張機能が動作する．

使用可能空き領域の動的拡張機能は，使用可能空き領域の容量監視機能にて使用可能容量が閾値未満であると判定された場合に動作し，制限機能により保管されていた書き換え前のビットマップ情報を用いて，使用禁止空き領域を使用可能へと開放していく．このとき，使用禁止空き領域の先頭ブロックグループから順に開放していき，使用可能容量が閾値以上になるまで繰り返す．

現在の実装では，制限機能を最初の 1 回だけ行い，後は監視機能および動的拡張機能だけが動作する．監視機能および動的拡張機能は独立したスレッドを持ち，DWH などのアプリケーションと並列で動作する．

3. 性能評価

本章にて，前章で述べたファイル格納位置の動的制御手法を DWH に対して用いた場合の性能を評価する．

† 工学院大学大学院 工学研究科 電気・電子工学専攻,
Electrical Engineering and Electronics, Kogakuin University Graduate School

3.1. 測定方法

物理計算機 1 台の MySQL 上で TPC-H の q4 を 10 回実行し，通常手法と動的制御手法とで平均実行時間を比較した．各測定に必要なデータベース作成前に，MySQL データ格納用 HDD のファイルシステムの初期化を行った．動的制御手法の測定では，初期化完了直後に動的制御手法の制限機能を動作させ，使用可能空き領域を設定した．このとき，使用可能空き領域の閾値は 6 GB，監視間隔は 3 秒とした．

また，MySQL はデータベース内のファイルの分け方を設定することができ，本測定ではテーブル毎にファイルを分けるための `file_per_table` を設定している．

3.2. 測定環境

使用した物理計算機の仕様は表 1 の通りである．MySQL のバージョンは 5.1.73 である．また，MySQL で使用されるデータは表 1 の Storage 行の 2 つ目にある容量が 500 GB の HDD であり，当該 HDD の仕様は表 2 の通りである．

表 1　使用した物理計算機の仕様

CPU	Athlon II X2 220 2.7 GHz
OS	CentOS 6.5 x86_64 minimal
Kernel	Linux-2.6.32.57
Storage	ST250DM000 (250 GB), DT01ACA050 (500 GB)
Main Memory	4 GB

表 2　MySQL データ格納用 HDD の仕様

Model Number	DT01ACA050
Interface	SATA 3.0
Interface Speed	6.0 Gbps
Device Size	500 GB
Buffer Size	32 MB
Rotation Rate	7,200 rpm
Average Rotational Latency Time	4.17 ms

3.3. 測定結果

測定結果を図 2 に示す．平均実行時間は通常手法が 1030.0 秒，動的制御手法が 882.7 秒となり，約 14.3%の性能向上が確認できた．

両手法における実行時間の最大値，最小値，標準偏差は表 3 の通りである．実行時間の最大値を比較すると，通常手法に比べて動的制御手法は 33.98%の性能向上が確認できる．一方で，最小値を比較すると，通常手法に比べて動的制御手法は 0.69%の性能低下が確認できる．これは，動的制御手法を使用すると，手法の監視機能および動的拡張機能が MySQL のスレッドと並列で動作するため，通常手法が偶然にディスクの最外周部分にデータ格納を行った場合よりも CPU 使用および I/O 使用にオーバーヘッドが生じるためである．しかし，標準偏差を確認すると，通常手法では測定毎にばらつきが生じるが，動的制御手法では必ずディスクの最外周部分にデータ格納を行うため，通常手法の最小実行時間と同等の実行時間が必ず得られることが確認できる．

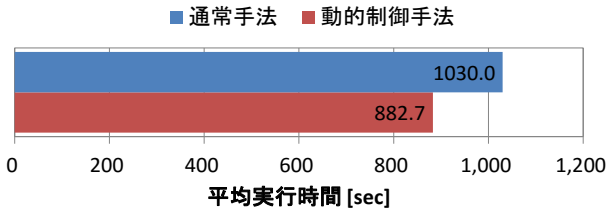

図 2　平均実行時間の測定結果

表 3　実行時間の最大値・最小値・標準偏差

	最大値	最小値	標準偏差
通常手法	1,339 sec	875 sec	166.06 sec
動的制御手法	884 sec	881 sec	1.00 sec

3.4. ディスク I/O に関する考察

通常手法および動的制御手法で TPC-H の q4 を実行したときの SCSI 層のディスクアクセスログを，図 3 に示す．縦軸のディスクアドレスの最大値は表 2 から 500 GB であるが，被アクセス部分のみを拡大して表示している．

TPC-H の q4 の I/O 要求は，lineitem 表と orders 表を読み込むことが主であるため，2 つの表を作成する時に Read 要求対象のアドレスが確定する．通常手法では図 3 よりも高アドレス部分に表を作成する場合もあるため，q4 の実行時間にばらつきが生じる．一方で，動的制御手法では必ずディスクアドレスの先頭から表の作成を行えるため，ほぼ最小の実行時間で q4 の処理が完了することが確認できる．

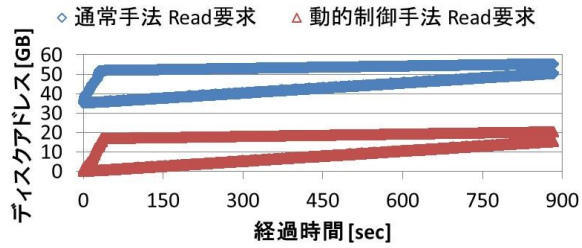

図 3　TPC-H q4 実行時の SCSI ログ

4. おわりに

本稿では，分散処理システムの I/O 処理高速化手法であるファイル格納位置の動的制御手法[1]を，DWH ベンチマーク TPC-H の q4 に適用し，性能を評価した．これにより，本手法の分散処理システム環境外における有効性を示した．

今後は，本手法に適した大規模データ処理特化型ファイルシステムを作成し，性能評価を行う予定である．

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”, 10th International Conference on Ubiquitous Information Management and Communication (ACM IMCOM’16), No.8-2,
DOI: http://dx.doi.org/10.1145/2857546.2857595 (2016).