

ディスク空き領域を用いるレプリケーション手法における 複製数の増加による性能向上

窪田 将希[†] 山口 実靖[†] 浅谷 耕一[†]

[†] 工学院大学工学部 〒163-8677 東京都新宿区西新宿 1-24-2

E-mail: [†] c203031@ns.kogakuin.ac.jp, sane@cc.kogakuin.ac.jp, asatanik@cc.kogakuin.ac.jp

あらまし 近年、情報処理システムが処理する情報量は飛躍的に増加している。この増大した情報を高速に処理するために、ストレージシステムには非常に高い性能が要求されるようになってきている。しかし、主たる情報蓄積装置である HDD はその機械的構造からランダムアクセス時における高速なデータ処理の実現が容易ではない。これに対して HDD の容量は著しい速度で増加をしており、多くの情報処理システムが使用されていない記憶領域を大量に有している。この状況を考慮し、同一の HDD の空き領域内にデータの複製を配置しランダムアクセス性能を向上させる研究が行われている。本研究では複製の数を増加させることにより、更なる性能向上を実現する手法を提案する。そして、性能評価結果を紹介し提案手法の有効性を示す。

キーワード ストレージ, 複製技術

Performance Improvement of Free Disk Space Replication by Increasing Number of Replications

Masaki KUBOTA[†] Saneyasu YAMAGUCHI[†] and Koichi ASATANI[†]

[†] Kogakuin University 1-24-2 Nishi-Shinjyuku, Shinjyuku-ku, Tokyo, 163-8677 Japan

E-mail: [†] c203031@ns.kogakuin.ac.jp, sane@cc.kogakuin.ac.jp, asatanik@cc.kogakuin.ac.jp

Abstract Recent computer systems process huge amount of data. Storage systems are required to provide very high performance. However, performance of storage device is not enough to process these huge data, especially in the case of random access. On the other hand, storage capacity is rapidly increasing. Thus computer systems have many unused space. Free Space Filesystem creates file replicas in unused storage space in order to improve storage performance. In this paper, we discuss relation between number of replicas and performance, and propose performance improving method by increasing number of replicas.

Keyword Strage, Data Replication,

1. はじめに

近年、情報処理システムで処理する情報量が飛躍的に増加している。この増大した情報を高速に処理するために、ストレージシステムには非常に高い性能が要求されるようになった。しかし、主たる情報蓄積装置である HDD はその機械的構造から情報へのランダムなアクセスの性能が高くなく、高速なデータ処理の実現は容易ではない。デバイス性能は数年で2倍程度の低い成長だが、これに対してストレージの容量は年率50%増加する急速な成長をしており、ストレージの大容量化はデバイス性能向上を上回る速度で達成されている。そのため、企業の計算機のディスク使用率は50%程度であることが多く[1]、多くの情報処理システムが

使用されていない記憶領域を大量に有している。この状況を考慮し、同一の HDD の空き領域内にデータの複製を配置しランダムアクセス性能を向上させる研究が行われている[1]。本研究では複製の数を増加させることによりこれを発展させ、更なる性能向上を実現すること目的とする

2. ディスクアクセス時間とファイルシステム

ディスクアクセス時間は、ヘッドがディスクの目的トラックへの移動する時間(シーク時間)、ヘッドが移動完了後にディスクが目的のセクタへ回転するまでの時間(回転待ち時間)、データ転送時間の3つにより構成されている。ランダムアクセスの場合ディスクアクセ

ス時間の大部分をシーク時間、回転待ち時間が占めている。ディスクの異なるアドレスにアクセスした場合、移動時間(シーク時間+回転待ち時間)が発生する。移動距離と移動時間の関係の調査結果を図1, 2に示す。図2は図1を拡大したものである。図1より移動距離が増加すると移動時間が増加していることが分かる。これは、移動距離が増加したことによりシーク時間が増加したことが原因である。また図2より移動距離がある一定まで増加すると移動時間が減少していることが分かる。この周期はディスクの回転待ち時間の周期である。

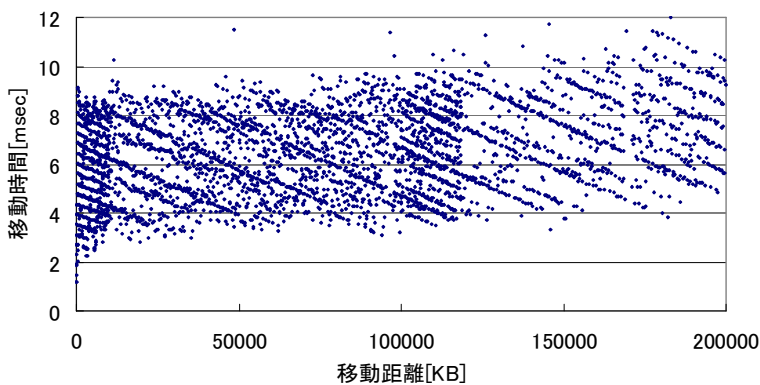


図1 シーク距離とアクセス速度の関係

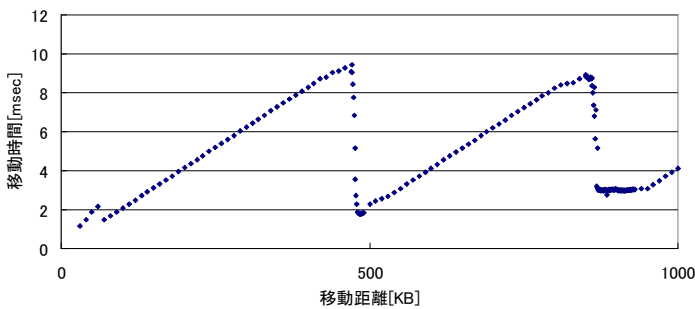


図2 シーク距離とアクセス速度の関係

ファイルを作成、配置する際にファイルシステムはアクセスパターンを想定せずに配置を行う。そのため、連続してアクセスされるファイルが離れて配置されることがあり、その場合にはシーク時間が増加し、ディスクアクセス性能が低下する。この問題は、連続してアクセスされやすいファイルを近隣に配置することによって改善することが可能である。

次章にて、ヘッドのシーク時間の短縮によるファイルアクセス性能向上を目的としたレプリケーション手法である Free Space File System を紹介する。

3. FreeSpaceFileSystem(FS2)

FS2は、ファイルシステムのExt2を改変したものであり、ディスクの大容量化により生じた空きブロックの増加に着目し、空きブロックにファイルのレプリカを配置し、ランダムアクセス性能を向上させるものである[1]。具体的にはアクセスパターンの観察に基づき連続アクセスするファイルのレプリカを近隣に配置し、HDDのヘッドの移動量を減少させることによるアクセス性能を向上させる。図3に動作例を示す。

この例ではファイル0,1,2の順にアクセスが行われる。ファイル1が他のファイルと離れた場所に配置されているために、シーク距離が増加している。そこで、ファイル1のレプリカをファイル0,2の近隣に配置することによりシーク時間の短縮が可能となる。それによりアクセス性能を向上させている。本手法では、レプリカの数は最大で1個(オリジナルとレプリカの合計で2個)までとしている。

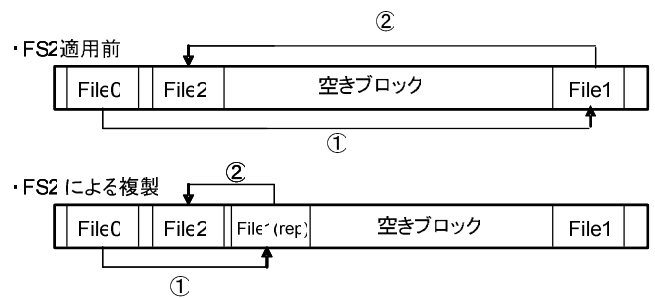


図3 FS2における動作例

4. 複数の空き領域複製

FS2において、空き領域へのファイルのレプリカの作成および配置は1個のみで行われている。そこで、本研究では空き領域へのファイルレプリカ数を増加させることにより、更なるディスクアクセス性能向上を実現させる手法について考察する。

4.1 HDDアクセスパターンの調査

アプリケーションのHDDアクセスパターンの調査として、アプリケーションを起動する際に発生する読み込みHDDアドレスとファイル名の取得を行った。アプリケーションにはFirefoxとOpenOffice Calcを用いた。実験環境を表1に示す。また、各アプリケーションのアクセスパターンの測定結果を図4, 図5に示す。図4より、Firefox起動時には主にディスクアドレス50GB近辺、80GB付近にアクセスされ、様々なアク

セスが見られた。OpenOffice Calcは主に 50GB 付近、100GB 付近に集中してアクセスされた。50GB 付近には /usr/lib/openoffice.org2.0/ 以下のファイルがあり、100GB 付近には /root/.openoffice.org2.0/ 以下のファイルがある。

表 1 実験環境

| | |
|------------------|--------------------------------|
| OS | FedoraCore6 |
| CPU | Intel(R) Pentium 4 CPU 2.80GHz |
| File System | ext3 |
| Kernel | Linux 2.6.18.8 |
| HDD | WDC WD1200JB-75CRA0 |
| Rotational Speed | 7200rpm |
| Capacity | 120GB |

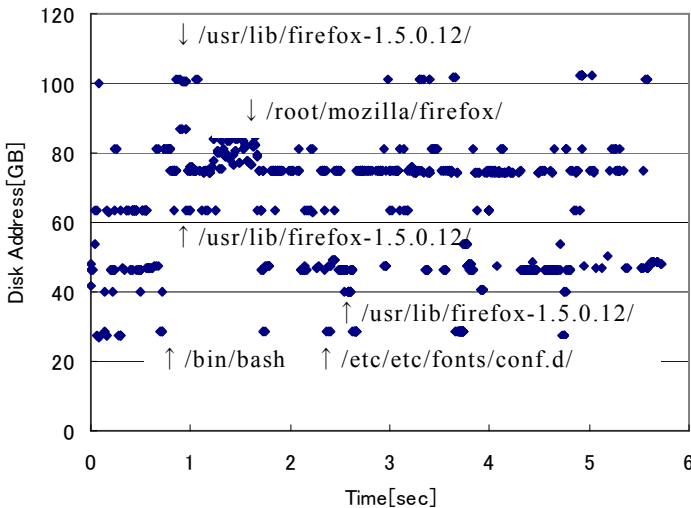


図 4 Firefox におけるアクセスパターン

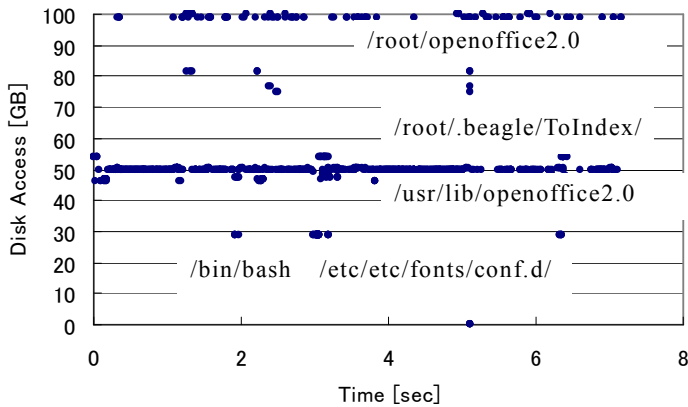


図 5 OpenOffice Calc におけるアクセスパターン

4.2 性能測定

本節では複数の複製が作成された場合のストレージアクセスをシミュレートし、複数複製手法における複製の数と性能の関係を調査する。

まず、Firefox 起動時のストレージアクセス時間について調査を行う。図 4 のアクセスパターンを元に、複製を用いない従来手法、単一の複製を用いる手法、複数の複製を用いる手法、全ブロックを連続して空き領域に複製する手法で得られる性能を測定した。

単一の複製を用いる手法では、100GB 付近のブロックのレプリカを 60GB 付近の空き領域に配置をした場合を想定し、ストレージアクセス性能の測定を行った。100GB 付近のブロックに保存されているファイルは /usr/lib/firefox-1.5.0.12/ で Firefox のファイルである。単一複製配置したアクセスパターンを図 6 に示す。本例ではアクセスされるアドレスの中心からもっとも遠いブロック群のみを複製の対象とした。

複数の複製を用いる手法では、上記単一複製に加え 30GB 付近のファイルの複製も配置した場合を想定し性能を測定した。30GB 付近のブロックに保存されているファイルは、/etc/fonts/conf.d/75-blacklist-fedora.conf や /bin/bash などの多数のアプリケーションにより使用されるファイルであり、複製数が 1 個に限定されている状況においてはこれの複製を Firefox の近隣に配置することはできない。本例でも同様に、アクセスされる中心アドレスから最も遠いブロック群のみを複製の対象とした。

全ブロック複製手法は、アクセスされる全ブロックを 100GB 付近の連続した空き領域に複製した理想的な状況における性能である。複数複製配置、全ブロック複製を行った際のアクセスパターンを図 7 に示す。

アクセス性能は、上記手法によりアクセスされるアドレスに対して I/O 要求を発行し、全アクセスに要した時間を測定した。本測定はアプリケーションが行う CPU 処理の時間を除いた HDD アクセス時間のみの測定である。測定結果を表 2 に示す。

次に、OpenOffice Calc 起動時のストレージアクセスに関する調査を行う。同様に図 5 におけるアクセスパターンを元に複製を行う。単一複製手法では 100GB 付近のブロックを 50GB 付近に複製した場合を想定し測定した。100GB 付近のブロックに保存されたファイルは /root/openoffice2.0 などの Openoffice のファイルである。複製複数手法では 30GB 付近のブロックも 50GB 付近に複製した場合を想定し、全ブロック複製手法ではアクセス対象の全ブロックの複製を 100GB 付近の連続空き領域に連続して配置した場合を想定している。測定結果を表 3 に示す。

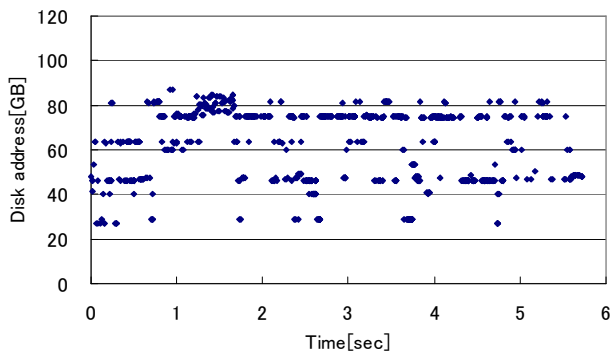


図 6 Firefox の単一複製手法におけるアクセスパターン

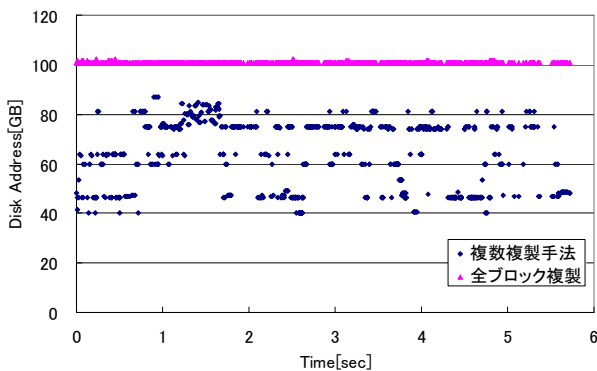


図 7 複数複製手法及び全ブロック複製手法におけるアクセスパターン

表 2,3 より、複製を用いることによりストレージアクセス性能を向上させることが可能であることがわかり、複製数を増加させることにより更なる改善が可能であることが確認された。アプリケーションがアクセスする全ブロックを連続領域に複製する手法は、大量の空き領域と多くの複製時間が必要となると考えられるが、大幅な性能改善が期待でき使用頻度の高いアプリケーションに関しては有効な手法になると考えられる。

表 2 Firefox におけるディスクアクセス性能

| | アクセス時間[msec] |
|---------|--------------|
| 複製なし | 4042.2 |
| 単一複製 | 3997.7 |
| 複数複製 | 3678.8 |
| 全ブロック複製 | 495.5 |

表 3 OpenOffice Calc におけるディスクアクセス性能

| | アクセス時間[msec] |
|---------|--------------|
| 複製なし | 7074.9 |
| 単一複製 | 6170.46 |
| 複数複製 | 6007.2 |
| 全ブロック複製 | 1386.4 |

7. おわりに

本稿では、ストレージアクセス性能向上を目的とした空き領域へのファイル複製手法を、複製数の増加によりさらに改善する手法を提案した。そして、アプリケーション起動時のストレージアクセスパターンを取得し、複製数が増加された場合に行われる I/O 処理をシミュレートし本手法のとして有効性の検証を行った。測定結果より、性能が向上されることを確認した。

今後は、提案手法のファイルシステムへの実装やそれを用いた性能の測定、別のアプリケーションを用いた検証、レプリカの管理法、さらなる性能向上手法についての考察を行う予定である。

文 献

- [1] Hai Huang, Wanda Hung, Kang G. Shin, "FS2: Dynamic Data Replication in Free Disk Space for Improving Disk Performance and Energy Consumption," SOSP 2005 pp. 263-276
- [2] John Douceur and William Bolosky, A Large-scale Study of File-System Contents, ACM SIGMETRICS Performance Review, 59-70, 1999.
- [3] C. Lumb and J. Schindler and G. Ganger and D. Nagle and E. Riedel, Towards Higher Disk Head Utilization: Extracting Free Bandwidth from Busy Disk Drives, Proceedings of the Symposium on Operating Systems Design and Implementation, 2000
- [4] Linux Filesystem Performance Comparison for OLTP, <http://oracle.com/technology/tech/linux/pdf/Linux-FSPerformance->
- [5] M. K. McKusick *et al.*, A Fast File System for UNIX, *ACM Transactions on Computing Systems (TOCS)*, 2(3), 1984.
- [6] M. K. McKusick *et al.*, A Fast File System for UNIX, *ACM Transactions on Computing Systems (TOCS)*, 2(3), 1984.
- [7] Douglas Orr and Jay Lepreau and J. Bonn and R. Mecklenburg, Fast and Flexible Shared Libraries, *USENIX Summer*, 237-252, 1993
- [8] Sedat Akyurek and Kenneth Salem, Adaptive Block Rearrangement, *Computer Systems*, 13(2): 89-121, 1995.
- [9] P. J. Denning, Effects of Scheduling on File Memory Operations, AFIPS Spring Joint Computer Conference, 9-21, 1967.