B-045

二重ファイルシステム環境を想定した仮想 HDD イメージファイルの再配置に関する考察
A Study on VM Image File Layout Optimization in Virtualized Environment

山田将也† 山口実靖†
Masaya Yamada Saneyasu Yamaguchi

1. はじめに

I/O 性能の向上を実現する手法の一つに HDD 上のデータの配置を変更する手法があり，これまで研究されてきている[1][2][3]．しかし，これまでに提案されてきた再配置手法は仮想計算機(VM)環境を想定しておらず，これらの手法を工夫なく VM 環境に適用しても得られる性能向上は限定的であると予想される．また VM は Web サーバや DB サーバなどに用いられることが多く，このような使用方法の場合は頻繁にアクセスされる領域(ホットスポット)が既知であることが多い．よって，その領域の移動による I/O 性能の向上が効果的に行えると期待できる．

本稿では，1 台の物理ストレージ上に複数の仮想 HDD イメージファイルが存在する環境を想定し，VM 環境に適したデータ再配置手法の提案と評価を行う．

2. 既存研究

初期のディスクレイアウトの理論に関する研究として文献[1]があり，シミュレーションによる研究として文献[4][5][6][7]がある．文献[1]では，最高頻度アクセスデータをストレージの中央に配置する organ pipe 手法がランダムアクセスに適していることが示されている．organ pipe 手法を現実のワークロードに適用した研究として cylinder shuffling がある[7]．本手法ではシリンダー単位で並び替えを行う．並び替えの単位をブロックとすることによりさらなる高速化を実現した研究として[6]がある．また，実システムにおける block shuffling について最初に述べた研究として文献[8]がある．

FFS[9]やその後続の研究[10][11]にて，関連するデータブロックと inode をディスク上の近隣に配置することにより I/O 速度を向上させる方法が提案されている．また文献[12]にて，参照の局所性ではなく，微小ファイルの距離を近づけることに着目して性能を上げる方法が提案されている．

ログ構造化ファイルシステム[13]では，大幅な書き込みの性能の向上が実現されている．また，アクセス頻度の高いファイルをディスクの外周に配置することによりログ構造化ファイルシステムをさらに高速化する研究がなされている[14].

HFS[2]の Hot File Clustering や Smart File System[15]では，ファイルシステムが実行時アクセスパターンを観察し，高頻度アクセスデータを予約領域に移動を行っている．FS2[3]では，ファイルの複製を用いて連続アクセスされるファイルを近隣に配置することにより I/O の高速化を実現している．

また，既存の再配置手法を変更せずに VM 環境に適用し，その効果を評価した研究として我々の文献[16]がある．

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

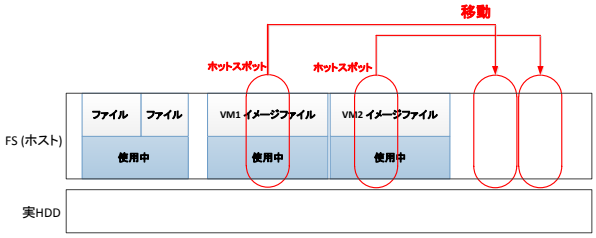


図 1. ゲスト FS を考慮しない場合

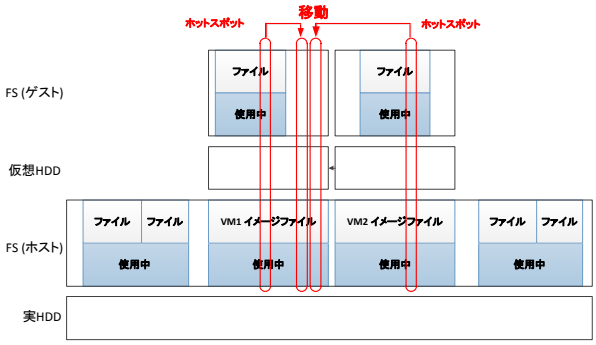


図 2. 提案手法

3. 提案手法

文献[16]では，図 1 の様にゲスト OS のファイルシステム(ゲスト FS)を考慮せずにホットスポットをホスト OS のファイルシステム(ホスト FS)における空き領域に移動した．本章では，図 2 の様にゲスト FS を考慮した再配置手法を提案する．

VM における仮想 HDD イメージファイルは，VM 作成時に必要な領域を実 HDD 上に確保して作成される．そして，そのイメージファイルの中にゲスト FS が構築される．よって，ホスト FS において使用中と見なされるが，ゲスト FS においては未使用領域とされる領域が存在し，その領域へのアクスは極めて少ない．この領域を物理ストレージの中央以外の場所に配置し，ホットスポット領域をこの場所に配置し，アクセス距離をさらに低減させる手法を提案する．

4. 性能評価

提案手法の性能評価を行うために，性能評価実験を行った．1 台の物理計算機上に 2〜3 台の VM を稼働させ，それらの VM 上で同時にベンチマークソフト FFSB を実行し，その際の実 HDD へのアクセスを観察しホットスポットを取得する．そのホットスポットをホスト FS 上の空き領域に移動する手法(通常再配置)の性能と VM 上の空き領域に移動する手法(提案手法)の性能を測定した．再配置はいずれもホスト FS 上で実行した．また，比較のために，コピーオンライト方式(CoW)で作成したイメージファイルを持

つ VM の性能の測定も行った．CoW は VM 作成時にイメージファイルの全容量をストレージ上に確保するのではなく必要な容量のみ確保し，その後必要な領域が増加するごとに必要分を追加的に確保していく手法である．VM 作成時はゲスト FS における未使用領域がストレージ上に存在しないため，各ゲスト FS の使用領域同士が実ストレージ上にて近隣に存在することが期待されるが，ゲスト FS の使用領域の増加に伴い実ストレージ上に不連続的に使用領域を確保するために性能が劣化していくと予想される．CoW に関しては，VM 内にベンチマークファイルのみが作成されイメージファイルが成長していない状態 CoW(C)と，VM 上にベンチマークファイル以外のファイルが多数存在している状態 CoW(D)の 2 通りで実験を行った．実 HDD の容量は 1[TB]で，各 VM の仮想 HDD の容量は 100[GB]である．

測定結果を図 3 に示す．図 3 より，提案手法と通常再配置手法の性能が高いこと，2VM において提案手法と通常再配置手法の性能はほぼ同等であるが 3VM において提案手法の方が性能が優れること，CoW 手法は初期状態(CoW(C))では通常再配置と近い性能を示すがディスク領域の増加に伴い性能が大きく劣化することが確認された．

5. 考察

稼働 VM 数 2 台の時の通常再配置，CoW(C)，CoW(D)，提案手法のシーク距離とシーク時間の関係を図 4(a)～(d)に示す．通常再配置と提案手法ではホットスポットの再配置を行うことによってホットスポット間のシーク距離を削減し，図 4(a)，(d) のように短い距離のシークが大部分を占めていることがわかる．CoW(C)では VM 内でベンチマークファイルが作成されるとイメージファイルに追記していくため複数 VM のベンチマークファイルが実 HDD 上で近隣に配置されている．そのため，図 4(b)では(a)，(d) と同様に短距離シークが大部分を占めている．これに対して，CoW(D)は，1VM の仮想 HDD イメージが実 HDD 上に散らばって配置されている．そのため図 4(c)のように大小様々なシークが発生している．そのため，通常再配置，CoW(C)，提案手法に比べて性能が大幅に低くなったと考えられる．稼働 VM 数 3 台の場合も同様である．

6. 終わりに

本稿では VM 環境を考慮したディスク上データの再配置手法を提案し，その性能の評価を行った．評価の結果，提案手法の性能が既存の手法よりも優れることが確認された．

今後は動的な再配置の実現方法について考察していく予定である．

謝辞

本研究は科研費（22700039）の助成を受けたものである．

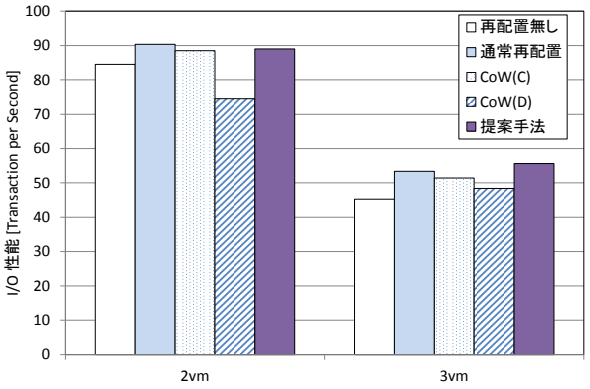
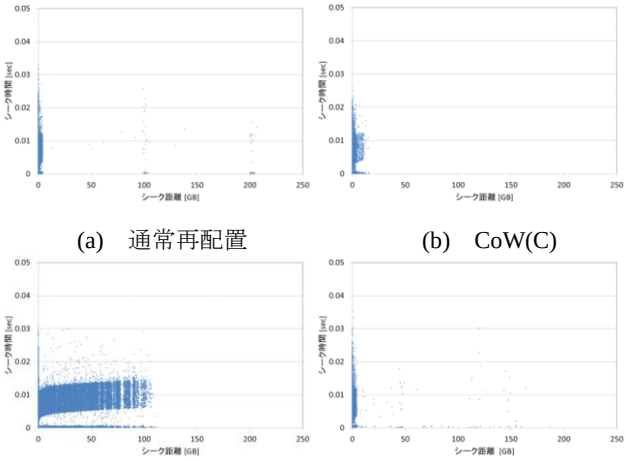


図 3. 各手法における VM の I/O 性能



(a) 通常再配置　(b) CoW(C)

(c) CoW(D)　(d) 提案手法

図 4. 各手法におけるシーク距離とシーク時間の関係

参考文献

[1] C. K. Wong, Algorithmic Studies in Mass Storage Systems Computer Sciences Press, 1983.
[2] HFS Plus Volume Format, http://developer.apple.com/technotes/tn/tn1150.html
[3] Hai Huang, Wanda Hung, Kang G. Shin "FS2: Dynamic Data Replication in Free Disk Space for Improving Disk Performance and Energy Consumption", SOSP'05, pp.263-276, October. 2005.
[4] Robert English and Stephnov Alexander, Loge: A Self-Organizing Disk Controller, Proceedings of the Winter 1992 USENIX Conference, 1992.
[5] David Musser, Block Shuffling in Loge, HP Technical Report CSP-91-18, 1991.
[6] C. Ruemmler and J. Wilkes, Disk Shuffling, HP Technical Report, HPL-CSP-91-30, 1991.
[7] P. Vongsathorn and S. D. Carson, A System for Adaptive Disk Rearrangement, Software Practice Experience, 20(3): 225–242, 1990.
[8] M. K. McKusick et al., A Fast File System for UNIX, ACM Transactions on Computing Systems (TOCS), 2(3), 1984.
[9] Sedat Akyurek and Kenneth Salem, Adaptive Block Rearrangement, Computer Systems, 13(2): 89–121, 1995.
[10] Stephen Tweedie, Journaling the Linux ext2fs Filesystem, LinuxExpo, 1998.
[11] R. Card and T. Ts'o and S. Tweedle, Design and Implementation of the Second Extended Filesystem, First Dutch International Symposium on Linux, 1994.
[12] Greg Ganger and Frans Kaashoek, Embedded Inodes and Explicit Groups: Exploiting Disk Bandwidth for Small Files, USENIX Annual Technical Conference, 1.17. 1997.
[13] M. Rosenblum and J. Ousterhout, The Design and Implementation of a Log-Structured File System, ACM Transactions on Computer Systems, 26–52, 1992.
[14] J.Wang and Y.Hu, PROFS–Performance-Oriented Data Reorganization for Log-structured File System on Multi-Zone Disks, The 9th International Symposium on Modeling, Analysis and Simulation on Computer and Telecommunication Systems, 285–293, 2001.
[15] C. Staelin and H. Garcia-Molina, Smart Filesystems, USENIX Winter, 45–52, 1991.
[16] 山田将也,山口実靖,"仮想計算機環境における VM イメージファイルの配置に関する一考察",DICOMO2011,pp1234