遠隔メモリページングにおける スワップイン履歴を用いた ページ置換アルゴリズムの初期評価

斉藤 和広† 緑川 博子† 甲斐 宗徳†

筆者らはローカル物理メモリサイズに制限されず，クラスタの各ノードの遠隔メモリを集めて仮想的に大容量メモリとする分散大容量メモリシステム DLM を構築，評価してきた．従来までの DLM では低コストのページ置換アルゴリズムとしてクロックアルゴリズムを適用していた．遠隔メモリを利用する上で，遠隔ページスワップは最もクリティカルな処理である．今回，これの高速化のために，筆者らはクロックアルゴリズムをベースにスワップインの履歴に注目した新しいページ置換アルゴリズム CASIN(Clock Algorithm with Swap-IN history)を提案する．LRU 等のメモリアクセス履歴を利用したアルゴリズムはユーザレベルソフトウェアではその記録処理のコストが大きいが，今回提案する CASIN のスワップイン履歴の記録はユーザレベルソフトウェアにおいても非常にコストが少ない．CASIN を DLM に実装した結果，従来の方式よりも遠隔スワップの回数を最大で 32%まで減らし，実行時間を最大 2.7 倍高速化することに成功した．本稿では，この新しいページ置換アルゴリズムの概要について述べ，様々なアプリケーションで性能評価を行ったので報告する．

Page Replacement Algorithm using Swap-in History for Remote Memory Paging

Kazuhiro SAITO† Hiroko MIDORIKAWA† and Munenori KAI†

The authors already proposed the Distributed Large Memory System: DLM which provides a larger size of memory beyond that of local physical memory by using remote memory distributed over cluster nodes. The original DLM adopted the Clock algorithm as a low cost page replacement algorithm. In the DLM, the remote page swapping is the most critical in performance. For more efficient swap-out page selection, we propose the new page replacement algorithm which pays attention to swap-in history: CASIN (Clock Algorithm with Swap-IN history). The LRU and other algorithms which use the memory access history generate more overhead for user-level software to record memory accesses. On the other hand, using swap-in history generates little costs. According to our performance evaluation, this new algorithm has reduced the number of the remote swapping in the maximum by 32% and gains 2.7 times higher performance in real application, Cluster3.0. In this paper, we describe the mechanism of the new page replacement algorithm and show a performance evaluation for three applications.

1. はじめに

筆者らはクラスタにおける遠隔マシン上のメモリを仮想記憶の一部として利用し，ローカル物理メモリサイズに制限されずに大容量メモリを提供する分散大容量メモリシステム DLM(Distributed Large Memory)を構築，評価してきた 1)2)．このシステムでは，ローカルマシンと遠隔マシンでネットワーク越しにページ単位のデータ通信を行うことで，DLM が提供する大容量のメモリを管理している．これを遠隔メモリページングと呼び，ここでノード間のページの交換（遠隔スワップ）が行われる．この性能はネットワーク速度に大きく依存し，現状のネットワーク速度で性能を高めるための最も有効な手段は，通信を行う遠隔スワップの回数を減らすことにある．

ディスクやキャッシュなどにおいて，スワップの回数を減らすためのページ置換アルゴリズムに最近メモリアクセスしたページを優先する LRU やアクセス頻度の多いページを優先する LFU 等のメモリアクセス履歴を利用するものが存在する．しかしこれらの実現のために,メモリアクセス時刻等を全て記録するのはそのコストの大きさから非現実的で，実際にはこれらを改良・簡易化したアルゴリズムが利用される．LRU をコスト重視に改良した LRU-K,や LIRS，LRU と LFU を両方考慮にいれた Aging や LRFU, メモリアクセス履歴の付加情報としてスワップ履歴を利用した ARC や EELRU, CAR，SEQ 等の様々な研究がある 3)-9)．また Linux kernel ではメモリアクセスの時間による二段階のレベルの LRU リスト（アクティブリストと非アクティブリスト）を用いて仮想記憶のページ置換を行っている 15)．

DLM は kernel とは独立のユーザレベルソフトウェアとして設計・評価してきたことで，OS kernel のスワップ方式に比べ高速かつ高安定であることを示してきた．そのためメモリアクセス履歴を記録するアルゴリズムを利用することは，その記録処理のコストが無視できない点から利用できなかった．このことから，従来の DLM では低コストのページ置換アルゴリズムとして，アドレス順にスワップアウトページを選択するクロックアルゴリズム 14)を用いていた．今回，従来の方式から遠隔スワップの回数を減らし効率性を高めるために，ユーザレベルで大きなコストを発生させない各ページ固有のスワップインの履歴をクロックアルゴリズムに反映した新しいページ置換アルゴリズム CASIN(Clock Algorithm with Swap-IN history)を提案する．

遠隔メモリを利用するシステムの他研究のほとんどは，OS kernel のスワップ方式を

†成蹊大学工学研究科情報処理専攻
Graduate School of Engineering, Seikei University

利用するためにスワップデバイスを遠隔メモリアクセス用に構築したネットワークブロックデバイスに置き換える手法をとっている．そのため，ページ置換アルゴリズムもその kernel のスワップ機構を改良するかそのまま利用しているため，Linux のポリシーが基本方針となっている 10)-13).

また，LRU をベースにスワップの履歴に注目しているページ置換アルゴリズムも存在する．例えば ARC7)-9)ではスワップアウトの履歴を LRU の付加情報として用い，SEQ4)ではアドレス空間上で連続したスワップアウトの履歴のみを利用する．このような研究のほとんどはスワップ情報を LRU, LFU の簡易化・汎用化のための付加情報として利用しているため，LRU によるメモリアクセス履歴の記録のコストが大きく，CASIN とはアプローチが異なる．

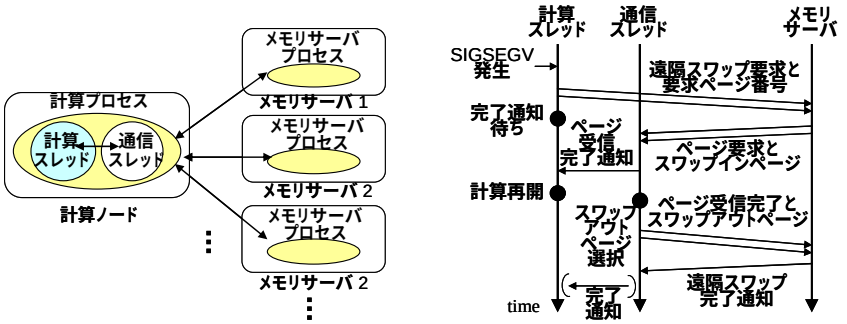
本稿では，提案する新しいページ置換アルゴリズム CASIN を述べ，これを実装した DLM での性能評価を三つのアプリケーションで行ったので報告する．

2. DLM における遠隔スワップ

2.1 DLM システム

DLM はクラスタ環境における複数のコンピュータを, あたかも一台のマシン上のメモリ資源であるかのように利用するシステムである． 図 1 (a)のようにローカルにあるマシンを計算ノード，リモートにあるマシンをメモリサーバと呼ぶ．計算ノードには，ユーザプログラムを処理する計算スレッドと主に通信を行う通信スレッドから構成される．メモリサーバは，計算ノードからのリクエストを受けメモリを提供する．ここで，ユーザプログラムがアクセスしたデータが計算ノードにない場合に，計算ノードはネットワーク越しにメモリサーバからそのデータを受信する．ここで計算ノードのメモリに余裕がない場合に，代わりのデータをそのメモリサーバに送信する．これを遠隔スワップと呼び，図 2 のように必要なデータの受信をスワップイン，計算ノードからメモリサーバへの代わりのデータの送信をスワップアウトと呼ぶ．このときのデータの通信はページ単位で行う．DLM ではこれを DLM ページと呼び，この遠隔メモリページングにより仮想的に大容量のメモリを提供している．なお，通信のプロトコルは TCP/IP を利用している．

遠隔メモリ管理は DLM ページ表を用いて DLM ページ単位で管理している．DLM ページのサイズは OS のページサイズの倍数であればシステム構築時に自由に設定することができ，通信速度にあわせた最適な値を設定することができる．更に，各ノードの利用メモリサイズと，どのノードを利用するかの優先順位は実行時に設定ファイルで指定できる．また，ユーザは DLM で利用するデータ（DLM データ）をプログラム中で指定することができ, DLM データは優先順位通りのノードに割り当てられていく．そしてあるノードの利用メモリサイズを超えた場合に次のノード（メモリサーバ）に割り当てられる．



(a) DLM のマシン構成　　(b) DLM の遠隔スワップの手順

図 1 DLM システムの構成

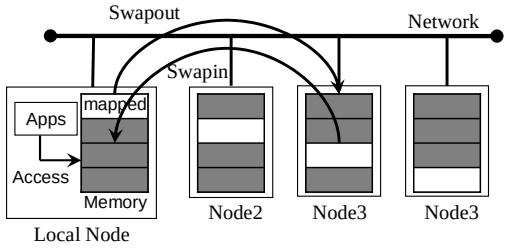


図 2 遠隔スワップの概要

ユーザプログラム（計算スレッド）が計算ノードにない DLM データにアクセスすると，図 1 (b)のような手順で遠隔スワップが発生する．計算スレッドは SIGSEGV をキャッチすると，必要な DLM データがある DLM ページ（スワップインページ）の番号をメモリサーバに送信する．通信スレッドはメモリサーバからのスワップインページを受信すると，計算スレッドの計算を再開させ，代わりとなる DLM ページ（スワップアウトページ）の選択を行う．そしてそのスワップアウトページをメモリサーバに送信する．なお，もし遠隔スワップが完了する前に再び計算スレッドで SIGSEGV が発生した場合，通信スレッドからの遠隔スワップ完了通知が来るまで新しいページ要求を待たされる．

2.2 DLM のページ置換アルゴリズム

従来の DLM では遠隔スワップで計算ノードからメモリサーバに送信するスワップアウトの DLM ページをクロックアルゴリズムで決定していた．これは探索時に，ページ単位で管理されている DLM のアドレス空間の最初の DLM ページから探していき，

最初に発見した計算ノードにマップされた DLM ページをスワップアウトページとして選択し，次回以降は前回選択した DLM ページからスタートして同様に探索するアルゴリズムである．

3. スワップイン履歴を用いたページ置換アルゴリズム CASIN

従来の DLM のページ置換アルゴリズムであるクロックアルゴリズムは，コストが非常に少なく，連続的なアクセスにおいては効率のよいページ選択を行う．しかしメモリアクセスのタイミングによっては次にアクセスする可能性の高い DLM ページをスワップアウトしてしまう可能性がある．図 3 は，Cluster3.0 というアプリケーション(詳細は 4.1)におけるその一例で，スワップイン・スワップアウトしたページの番号を時系列順にグラフにしたものである．これを見ると，スワップアウトしたページがすぐ必要となってしまっている．

このような非効率的な選択を回避し，かつユーザレベルでもコストを少なく実現するために，クロックアルゴリズムをベースにスワップインの履歴を利用したページ置換アルゴリズム CASIN を考案した．これは，スワップを発生させたページの前回のスワップイン後にスワップインしたページは，今回のスワップインでもすぐに必要となる可能性が高いという仮定を用いている．そのために CASIN では，前回のスワップインの直後にスワップインしているページをスワップアウトページとして選択しない方式をとっている．

3.1 CASIN のデータ構造

必要なデータ構造は大きくわけて図 4 の二つあり，一つがスワップイン履歴用の環状配列である．これは，遠隔スワップ発生時にスワップインされる DLM ページ番号を順に書き込むためのもので，最後まで書き込まれると先頭に戻る．この配列のサイズは設定された最大 DLM ページ数分で，メモリ全体への連続アクセスに対応できる．

もう一つが，DLM ページ表の各エントリにあるスワップイン履歴配列へのポインタである．これは，スワップインの履歴を取るために，各 DLM ページが前回のスワップインの時にスワップイン履歴配列上のどの位置に記録されたかを覚えておくためのものである．

3.2 CASIN の DLM への実装

CASIN は図 2 (b)の DLM のスワップアウトページ選択部で行われる．スワップインした DLM ページのスワップイン履歴配列へのポインタから，スワップイン履歴配列にある前回のスワップインの場所を見つける．そこからスワップイン履歴配列の次以降にある DLM ページ（次以降スワップインする可能性のある DLM ページ）を数ページ（**履歴反映数**）分選択し，スワップアウトページの対象外ページとする．そしてクロックアルゴリズムを用いて計算ノードにある DLM ページを見つけ，対象外ページ

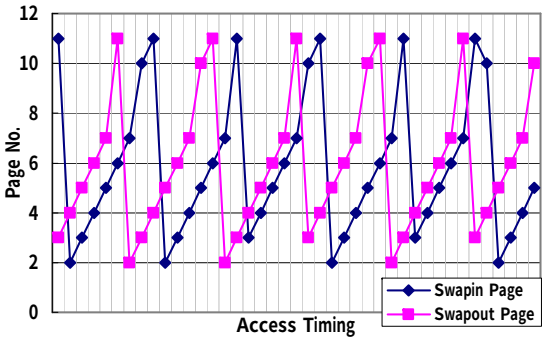


図 3 スワップしたページの時系列グラフ

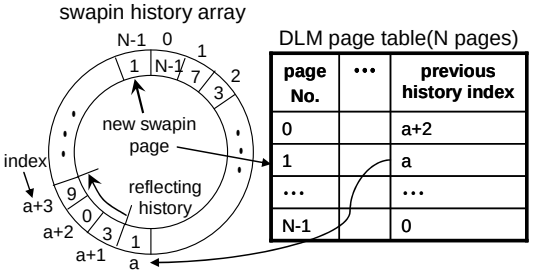


図 4 スワップイン履歴と DLM ページ表
(履歴反映数 3 でページ 0,3,9 はスワップアウトしない)

ではないものをスワップアウトする．

スワップイン履歴配列への登録は，スワップアウトページ選択が行われた後に，スワップインした DLM ページ番号をスワップイン履歴に登録し，DLM ページ表のエントリにその履歴のインデックスを登録する．

4. 性能評価

CASIN の性能評価を様々なアプリケーションで行った．実行環境は表 1 の 1GbitEthernet クラスタの 2 台(1GbpsCL)と，表 2 の 10Gbps ネットワーク(IP/Myrinet)の東京大学のスーパーコンピュータ T2K 20) (10GbpsCL)を用いた．まず従来の DLM のページ置換アルゴリズムを用い，これを**ローカルメモリ率**（プログラム全体で利用する全データのメモリサイズに対する計算ノードの利用メモリサイズ）別で実行する．

表 1　1GEthernet クラスタ（1GbpsCL）の実行環境

	Calculation Node	Memory Server
machine	HP ML150 G2	HP ML150 G3
CPU	Xeon 2.8GHz x 2CPU HyperThread	Xeon E5310 1.6GHz QuadCore x 2CPU
memory	1GB	8GB
Cache	L2 : 1MB/CPU	4MB/CPU
OS	Linux kernel2.6.20-1.2320.fc5 x86_64	Linux kernel2.6.23.17-88.fc7 x86_64
Compiler	gcc version 4.1.1 20070105	gcc version 4.1.1 20070105
NIC	Broadcom 5721 PCI-Express Gigabit NIC	NC7781 OnBoard Gigabit NIC
Network	1GbitEthernet	

表 2　東大 T2K（10GbpsCL）の実行環境

	T2K HA8000
machine	HITACHI HA8000-tc/RS425
CPU	AMD Opteron 8356 2.3GHz QuadCore x 4CPU
memory	32GB
Cache	L2 : 2MB/CPU(512KB/Core) L3 : 2MB/CPU
OS	Linux kernel2.6.18-53.1.19.el5 x86_64
Compiler	gcc version 4.1.2 20070626 mpicc for 1.2.7
Network	IP on Myrinet-10Gbps

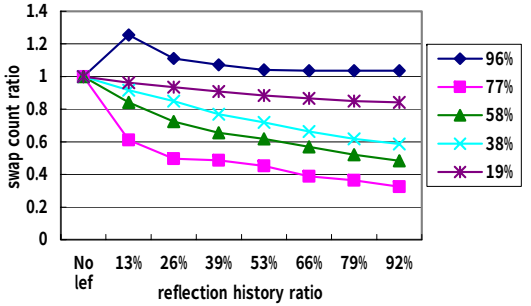
次に CASIN を実装した DLM を**履歴反映率**（スワップアウト候補となる計算ノードの DLM ページ数に対する履歴反映数）別・ローカルメモリ率別に実行する．実行結果の図 5〜8 では，(a)が従来の DLM に対して CASIN を適用した新しい DLM の遠隔スワップの回数の比で，(b)が同様に従来の DLM に対する新しい DLM の実行時間の比である．また 1GbpsCL では DLM ページサイズを 128KB，10GbpsCL では 1MB にして実行した．

4.1 Cluster3.0

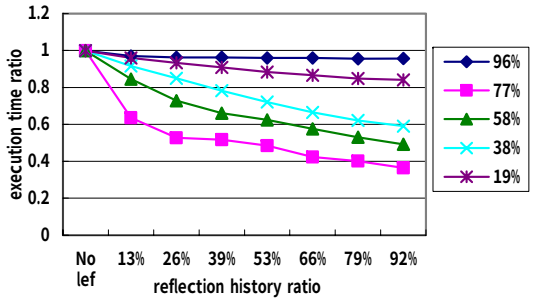
遺伝子分析のための様々なクラスタリング処理を行うアプリケーションである 19)．これは，データが大きく実行時間が非常にかかり，多くの数値計算とは違い，メモリの動的割り当てと解放を繰り返す．Cluster3.0 では数種のクラスタリングアルゴリズムを提供しているが，今回はこのうち階層クラスタリングで遺伝子分析を行い，その実行時間を計測した．また遺伝子のデータファイルは実験時間短縮のため，小さいサイズの，サンプルとしてある demo.txt（利用メモリサイズ 26MB）を用いた．

表 3　1GbpsCL における Cluster3.0 の各履歴反映率の実際の履歴反映数

ローカルメモリ率	計算ノードのDLMページ数	履歴反映率 13%	26%	39%	53%	66%	79%	92%
96%	198	26	52	78	104	130	156	182
77%	158	21	42	62	83	104	125	146
58%	118	16	31	47	62	78	93	109
38%	78	10	21	31	41	51	62	72
19%	38	5	10	15	20	25	30	35



(a) スワップ回数の比



(b) 実行時間の比

図 5　1GbpsCL での Cluster3.0 のローカルメモリ率別実行結果

4.1.1 1Gbps クラスタでの実行結果

DLM ページ数は 207 ページで，表 3 は，計算ノードの DLM ページ数，各履歴反映率における実際の履歴反映数を表に，図 5 は実行結果をグラフにしたものである．

図 5 (a)に示すように，履歴反映率を増やすほど遠隔スワップの回数が減っているのが分かる．特にローカルメモリ率が 77%で履歴反映率が 92%のとき遠隔スワップが 30%にまで減っている．全体でみると，履歴反映率は大きいほど良くなるようである．

しかし，ローカルメモリ率が高過ぎても低過ぎても履歴通りのスワップを行うことが少なく，遠隔スワップの回数の減少率は小さくなるようである．次に図 5 (b)では，実行時間が最大で 37%減少(実行速度 2.7 倍の高速化)したが，遠隔スワップの回数の減少率に比べ，実行時間の高速化は少なかった．これは，本来の処理の時間に対して遠隔スワップの処理時間がそれほど多くないために，遠隔スワップの減少に対してそれほど実行時間が減らなかったと考えられる．

4.1.2 10Gbps クラスタでの実行結果

この 10GbpsCL では, DLM ページサイズが大きいため DLM ページ数は 1GbpsCL に比べ少なく 26 ページとなる．実行結果を図 6 に示す．

図 6 から遠隔スワップ回数, 実行時間ともに 1GbpsCL のときと同様に減っていることがわかる．ただし減少率は最大でローカルメモリ率 58%のときに，遠隔スワップ回数は 50%，実行時間は 65%になり，遠隔スワップ回数の減少に比べ実行時間の減少が小さい．この原因は，10GbpsCL では DLM ページサイズは大きいために遠隔スワップの回数が少なく，ネットワークも速いため，総合的に遠隔スワップ自体のオーバヘッドが少ない．そのため 1GbpsCL のときよりも速度の高速化が見られにくい．

4.2 NAS Parallel Benchmark (NPB)

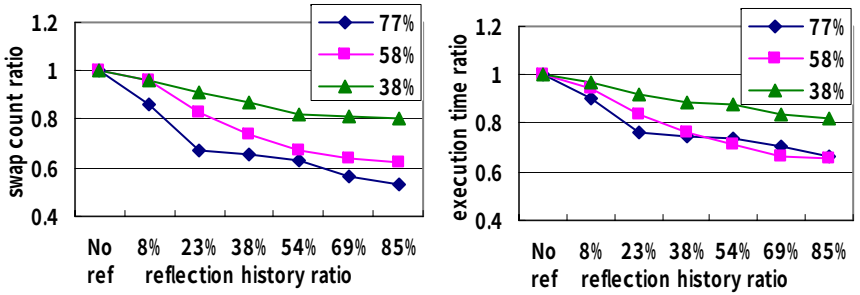
NPB2.3-omni-C17)17)の中から，IS のクラス C を用いて実行時間を計測した．これは 1.5GB のメモリを使用する. このアプリケーションはスワップ回数, 頻度が少ない．

これを 10GbpsCL で実行し，その結果が図 7 である．このアプリケーションは遠隔スワップが非常に少なく処理時間が長いため，全体的に遠隔スワップの処理時間が占める割合は少ない. 遠隔スワップが大きく減少しても実行時間の低下率は少なかった．また，ローカルメモリ率 78%に関しては遠隔スワップ回数が 883 回と全 DLM ページ数 1537 ページよりも少なく，一つの DLM ページで 2 回以上遠隔スワップが起こらずスワップイン履歴が活用されず，まったく遠隔スワップが減らなかった．しかしそのような状況においても実行時間が遅くなることはなかった．

NPB の各アプリケーション（FT, CG, MG, SP）で実行した結果では，IS 同様に性能が向上したが，アプリケーションによってその割合は異なった．また，一部のアプリケーションのあるローカルメモリ率(FT の 46%, SP の 72%)では履歴参照数が大きすぎると遠隔スワップ回数が従来の方式よりも増加する傾向にあることがわかった．これは各 DLM ページの遠隔スワップが発生した処理が，そのときにスワップイン履歴に書かれている前回の遠隔スワップ発生時の処理とは違うために，履歴が意味をなさず本来選択するべきでない DLM ページを選択してしまうためであると考えられる．しかし，そのような場合でも実行時間は悪くとも 1.1 倍に遅くなる程度で済むことがわかった．

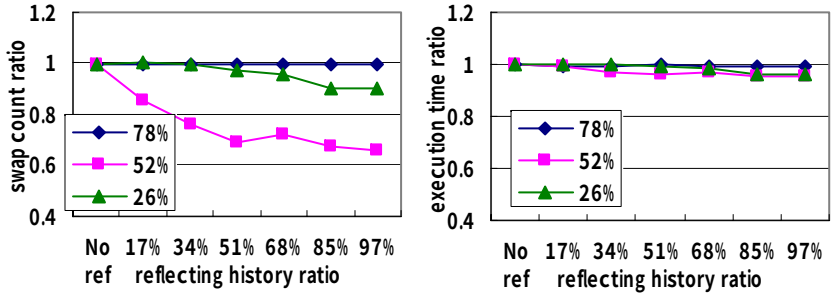
4.3 姫野ベンチマーク

非圧縮流体解析処理の性能評価のためにポアッソン方程式解法をヤコビの反復法



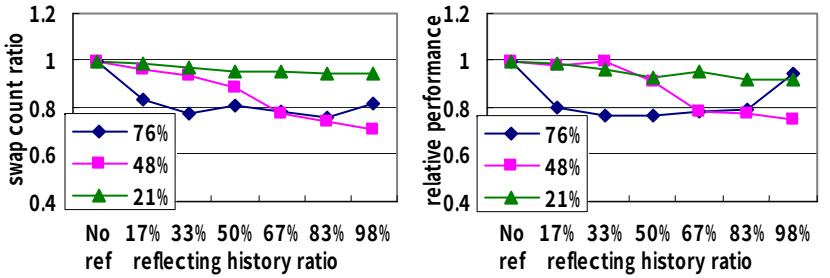
(a) 遠隔スワップ回数の比　　(b) 実行時間の比

図 6 10GbpsCL での Cluster3.0 のローカルメモリ率別実行結果



(a) 遠隔スワップ回数の比　　(b) 実行時間の比

図 7 10GbpsCL での NPB IS.C のローカルメモリ率別実行結果



(a) 遠隔スワップ回数の比　　(b) 実行時間の比

図 8 10GbpsCL での姫野ベンチマークのローカルメモリ率別実行結果

で解き，主要ループの処理速度を計るベンチマーク．メモリ負荷が高く，多重ループ処理で配列全体をスキャンする 18)．ここでは C プログラム版の ELARGE サイズ（513x513x1025 サイズ，15GB）を用いた．また性能は MFLOPS で出力されるが，図 8 (b)は相対時間で表示している．これを 10GbpsCL で実行した．DLM ページ数は 14406 ページで，図 8 が結果である．スワップ回数はローカルメモリ率 48%で，最大 70%，性能は最大でローカルメモリ率 48%の 1.3 倍まで上がった．また，ローカルメモリ率 76%では履歴反映数が 67%以上で性能が下がる傾向が見られた．このことから，履歴反映数が多くなると選択できる計算ノードの DLM ページの数が少なくなり，効率の悪い選択をする可能性が高くなることがわかる．

5. おわりに

今回，スワップイン履歴を利用した新しいページ置換アルゴリズムを提案し，DLM に実装し性能評価を行った．その結果，1Gbit/s のネットワークでは 10Gbit/s よりも効果が大きく，アプリケーションの種類によって効果は様々ではあるが，最大で遠隔スワップ回数を 30%に減らし，実行速度を 2.7 倍にまで高速化することに成功した．しかし,アプリケーション毎の最適となる履歴反映数はアプリケーション，ローカルメモリ率によって異なった．

ここで，アプリケーション毎にどのような履歴反映数を決めればいいかを考える．図 9 は様々なアプリケーションの，従来の方式に対する遠隔スワップ回数の割合を，そのアプリケーションで利用した全 DLM ページ数に対する履歴反映数別でグラフ化したものである．これを見ると，20%前後の履歴を反映することで比較的良好な性能を得られることがわかる．このことから，履歴反映数は，そのアプリケーションで利用する全 DLM ページの 20%程度に設定すると，遠隔スワップ回数を比較的減らせると考えられる．

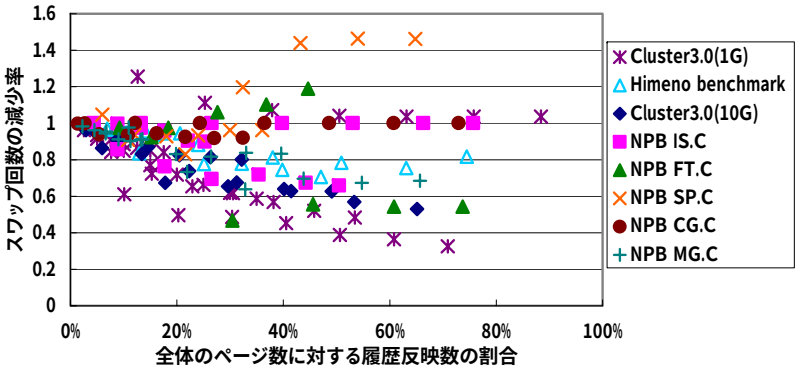


図 9　様々なアプリケーション毎の遠隔スワップ回数の比

参考文献

1)　緑川，黒川，姫野，“遠隔メモリを利用する分散大容量メモリシステム DLM の設計と 10GbEthernet における初期性能評価”，情報処理学会論文誌 Vol. 49 No. 4, Apr. 2008
2)　H. Midorikawa, K. Motoyoshi, R. Himeno and M. Sato, “DLM: A Distributed Large Memory System using Remote Memory Swapping over Cluster Nodes”, IEEE International Cluster Computing, Sept. 2008, pp. 268-273
3)　E. J. O’Neil et al., “The LRU-K Page Replacement Algorithm For Database Disk Buffering”, ACM SIGMOD, 1993, pp. 297-306
4)　G. Glass and P. Cao, “Adaptive Page Replacement Based on Memory Reference Behavior”, ACM SIGMETRICS, Feb. 1997
5) Y. Smaragdakis et al., “EELRU: Simple and Effective Adaptive Page Replacement”, ACM SIGMETRICS, 1999
6)　S. Jiang and X. Zhang, “LIRS: An Efficient Low Inter-reference Recency set Replacement Policy to Improve Buffer Cache Performance”, ACM SIGMETRICS, 2002
7)　N. Megiddo and D. S. Modha, “ARC: a self-tuning, low overhead replacement cache”, 2nd USENIX FAST, 2003, pp. 115-130
8)　N. Megiddo and D. S. Modha, “Outperforming LRU with an Adaptive Replacement Cache Algorithm”, IEEE Computer Society, 2004
9)　S. Bansal et al., “CAR: Clock with Adaptive Replacement”, 3rd USENIX FAST, Mar. 31, 2004
10)　S. Liang, R. Noronha, and D. K. Panda, “Swapping to Remote Memory over InfiniBand: An Approach using a High Performance Network Block Device”, IEEE Cluster Computing, Sept. 2005
11)　Tia Newhall et al., “ Nswap: A Network swapping Module for Linux Clusters ”, EuroPar03, 2003
12)　後藤，佐藤，中島，久門，”10GbitEthernet 上での RDMA を用いた遠隔スワップメモリの実装”, CPSY 研究会報告，信学技報 Vol.106 No.287, pp.7-12, Oct. 2006
13)　今井，松葉，石川，”大規模メモリ空間の利用を支援する遠隔スワップメモリシステム”，情報処理学会研究報告，2007-HPC-111，pp.121-126，Aug. 2007
14)　William Stallings, “Operating Systems Internal and Design Principles Fifth Edition”, Person Education Inc., 2005
15)　Daniel P. Bovet and Marco Cesati, “Understanding the LINUX KERNEL”,　O’Reilly Media Inc., 2006
16)　(2008) NPB (NAS Parallel Benchmarks) web site [Online]. http://www.nas.nasa.gov/Resources/Software/npb.html
17)　(2008) NPB2.3-omni-C web site [Online]. http://phase.hpcc.jp/Omni/benchmarks/NPB/index.html
18)　(2009) Himeno Benchmark web site [Online]. http://accc.riken.jp/HPC/HimenoBMT/index.html
19)　(2009) Cluster3.0 web site [Online]. http://bonsai.ims.u-tokyo.ac.jp/~mdehoon/software/cluster/
20)　(2009) HA8000 クラスタシステム [Online]. http://www.cc.u-tokyo.ac.jp/ha8000/