2J-04

アクセス履歴に基づいた仮想マシンライブマイグレーションのメモリ転送フェーズの最適化

岩佐 光† 梅澤 猛‡ 大澤 範高‡

千葉大学大学院融合理工学府† 千葉大学大学院工学研究院‡

1. はじめに

実行中の仮想マシンを稼働させたまま、別の物理サーバへ移送することができるライブマイグレーションでは、転送されるページのアクセスパターンによって、再転送の多発やページフォールトの発生が、仮想マシンの一時停止時間や総マイグレーション時間を増加させてしまう問題がある。筆者らは、これらの時間を削減するために、メモリへのアクセス履歴に基づく予測を用いて最適なメモリ転送フェーズを選択するライブマイグレーション手法を提案している。

本稿では、この手法の有効性を検証するため、ページテーブルエントリから物理ページの使用状況を取得するカーネルモジュールを用いてメモリアクセス履歴を収集し、これを基にメモリアクセス予測を行ってその予測成功率の評価を行った。

2. 仮想マシンライブマイグレーション

仮想マシンライブマイグレーションの工程は以下の 3 つのフェーズに分けることができる。

push フェーズ

移送元の仮想マシンを稼働させたままメモリページを転送する

stop-and-copy フェーズ

仮想マシンを停止させ、コンテキスト情報とページを転送する

pull フェーズ

移送先で仮想マシンを再開させ、移送元からメモリページを転送する

メモリページを転送するフェーズの違いで、主に pre-copy 型と post-copy 型の 2 種類のライブマイグレーションが提案されている。

pre-copy 型は、稼働中の仮想マシンのすべてのメモリページを push フェーズで転送するよう試みる。移送の実行中、移送元の仮想マシンは稼働しているため、転送後に書き換えられるメモリページ（dirty ページ）が発生する。dirty ページは push フェーズで繰り返し再送が行われるため、転送量が大きくなる。また push フェーズで転送できなかったページは stop-and-copy フェーズで転送されるため、システムの停止時間（ダウンタイム）が長くなる欠点がある。この手法では pull フェーズでの転送は行わず、システムは必要なページが揃った状態で再開する。

post-copy 型では、push フェーズでのページ転送は行わず、stop-and-copy フェーズでコンテキスト情報などの仮想マシンの再開に必要な最低限の情報を転送する。pull フェーズでシステムが移送先で再開すると、移送元からすべてのメモリページを転送する。この手法は、pre-copy 型と比べデータの転送量が小さいが、ページ転送が行われる前にシステムが再開するため、移送先で転送されていないページへのアクセスが生じると、ページフォールトが発生し、アプリケーション実行の遅延によりダウンタイムが長くなる欠点がある。

3. 関連研究

中居らは、pre-copy 型ライブマイグレーションにおいて、移送プロセスの開始の際に、拡張ページテーブルを利用した過去のメモリアクセスのプロファイリングを行い、更新される可能性の高いメモリページほど移送の後半で転送させ、ページの再送や仮想マシンの停止時間を削減する手法を提案している [1]。

また都築らは、直前のメモリアクセス履歴からアクセス予測を行い、適切なフェーズでメモリページを転送する、pre-copy 型と post-copy 型を統合したライブマイクレーション手法を提案している [2]。

4. メモリ転送フェーズの最適化

メモリページは一定の時間窓内で、読み込みアクセスのみが行われるページ（read ページ）、情報の書き込みを含むアクセスが行われるページ（write ページ）、アクセスがされないページ（clean ページ）の 3 種類に分類できる。read ペー

History-based optimization of memory transfer phases in live migration of virtual machines.

† Hikaru Iwasa, Division of Mathematics and Informatics Graduate School of Science and Informatics, Chiba University

‡ Takeshi Umezawa, Noritaka Osawa, Graduate School of Engineering, Chiba University

ジは、push フェーズで転送後にアクセスがあってもページ情報が書き変わらず、再送を行う必要がない。一方で write ページはアクセスの前後で情報が書き換わるため、push フェーズで転送をする場合はアクセスされる度に再送する必要がある。その再送の回数を削減するためには、pull フェーズや stop-and-copy フェーズでの転送が最適だと考えられる。

そこで、筆者らはメモリページのアクセスパターンを予測し、それに応じて最適なフェーズでページ転送を行うライブマイグレーション手法を提案している。この手法では push フェーズで read ページと clean ページを転送し、ページ再送回数を削減する。stop-and-copy フェーズでは、CPU ステートを含むコンテキスト情報などを転送し、移送先で仮想マシンを再開する。再開した後の pull フェーズでは、write ページを転送する。この手法ではメモリアクセスが正しく予測された場合に有効であるため、高精度で低オーバーヘッドの予測手法の実現が重要である。

5. 実験

動作する仮想マシンのメモリアクセス履歴を基に、アクセス予測のシミュレーションを行い、予測成功率の評価実験を行なった。アクセス履歴は Garg ら[3]が作成したカーネルモジュールを組み込んだ Linux カーネルを作成し収集した。

5.1. 実験手法

実験にアクセス履歴は、連立線形方程式を解く際の演算性能を計測する LINPACK ベンチマークの実行時のものを使用した。その際のパラメータとして、方程式の数を示す problem size を 18,000、行列の次元数を示す leading size を 18,008 に設定した。またアクセス履歴の収集は 16 ページを 1 つのブロックとし、500 ミリ秒間隔で行った。計測対象のアクセス履歴に対し、ある長さの時間窓を考え、「窓の先頭のアクセスが窓サイズ内でも続く」という予測方法とした。

5.2. 実験結果

窓サイズを変化させた時の予測成功率の変化を示すグラフを図 1 に示す。図 1 中の$A(w|c)$は clean アクセスと予測した場合の write アクセス率、$A(w|r)$は read アクセスと予測した場合の write アクセス率を示す。また$A(c|w)$は write アクセス予測した場合の clean アクセス率、$A(r|w)$は write アクセス予測した場合の read アクセス率を示す。窓サイズが大きくなると、read アクセスと clean アクセスと予測したもののうち、write アクセスであった率を示す$A(w|c)$、$A(w|r)$が増加していることがわかる。一方で write アクセス予測の失敗率に相当する$A(c|w)$、$A(r|w)$は低く、横ばいであった。

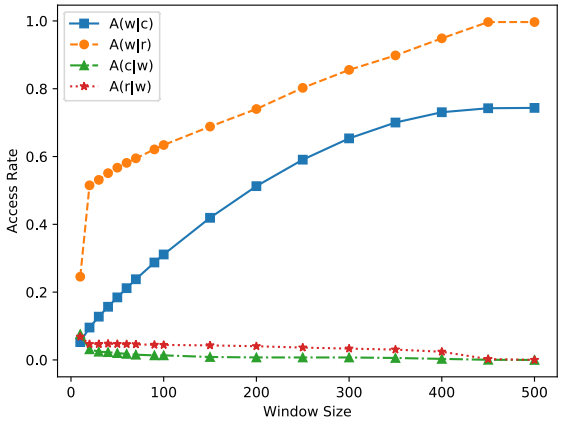

図 1 窓サイズによるアクセス率の変化

5.3. 考察

push フェーズでの転送量増加の原因となる write アクセス予測の失敗率は低い率で横ばいとなっている。これは本実験でアクセス履歴の取得に使用した LINPACK ベンチマークが、局所性の高いメモリアクセスを行うプログラムであるからだと考えられる。

6. まとめ

実験の結果からメモリアクセスに局所性があるアプリケーションの場合、write アクセス予測が高い確率で成功することがわかった。またこの結果から、push フェーズでの write ページの再送回数が削減でき、ライブマイグレーションの性能向上が見込まれる。

今後は異なるアクセスパターンのアプリケーションでの予測成功率と性能評価を行い、既存手法との比較を行う。また実環境への実装を行う予定である。

参考文献

[1] 中居新太郎 , 川島龍太, “更新履歴に基づいたメモリページ転送順序スケジューリングによる仮想マシンライブマイグレーションの高速化,” *情報処理学会論文誌*, vol 56, No.2, pp. 516-524, 2015.

[2] 都築 俊徳, 梅澤 猛 , 大澤 範高, “仮想機械ライブマイグレーションの統合方式,” *情報科学技術フォーラム講演論文集*, vol 10, No 1, pp. 371-372, 2011.

[3] Ankita Garg, Balbir Singh, Vaidyanathan Srinivasan, “Looking Inside Memory,” 2010