

## メモリの再利用により移動後の性能低下を抑えたライブマイグレーション

穂山 空道<sup>†</sup> 広瀬 崇宏<sup>††</sup>  
高野 了成<sup>††</sup> 本位田 真一<sup>†,†††</sup>

### 1. はじめに

仮想化されたデータセンターにおいて仮想マシンを頻りに移動することにより様々な効率化が可能である。例えば高性能、高消費電力の CPU と低性能、低消費電力の CPU 間を仮想マシンが行き来することで電力効率を上げる研究<sup>1)</sup> や似たメモリ内容を持つ仮想マシンを集約してメモリ利用率を上げる研究<sup>2)</sup> が行われている。このような効率化には仮想マシンのライブマイグレーション<sup>3)</sup> が不可欠である。しかし従来のライブマイグレーションには仮想マシンの移動に長い時間がかかる、移動後のページフォールトにより仮想マシンの性能が下がるという問題がある<sup>5)</sup>。この移動時間の長期化と性能低下は一回あたりでは僅かだが、仮想マシンが頻りに移動する場合には何度も繰り返すためアプリケーションの性能に大きな影響を及ぼす。そこで本稿ではメモリの一部をコピーせずに再利用することにより、仮想マシンが頻りにライブマイグレーションする場合に高速に移動できかつ移動後の性能低下も低減する手法を提案する。

### 2. 従来のライブマイグレーション

ライブマイグレーションの従来研究には pre-copy 型<sup>3)</sup> と post-copy 型<sup>4)5)</sup> がある。

pre-copy 型は移動元のメモリをすべて移動先にコピーしてから移動先で CPU の動作が再開する。CPU の停止時間を短くするため、メモリをコピーしている間も移動元で CPU を動作させる。そのため全てのメモリのコピーが完了する前に転送済みのメモリが更新されると、当該メモリは再び転送する必要がある。従ってメモリの更新が速いワークロードの場合には移

動に長い時間がかかる。

post-copy 型は先に移動先で CPU の動作を再開してから必要なメモリを on-demand でコピーする。そのため pre-copy 型に見られるメモリをコピーし直さねばならない問題はない。しかしメモリをネットワーク越しに転送する間 CPU が待たされるため、移動後に仮想マシンの性能が下がるという問題点がある。

従来のライブマイグレーションの問題点は全メモリをネットワーク越しに転送することが原因である。ネットワーク越しの転送はメモリと CPU の間のコピーに比べ遅いため、本章で述べた問題が発生する。

### 3. メモリの再利用により移動後の性能低下を抑えたライブマイグレーション

本稿ではライブマイグレーション時にメモリを再利用することで従来手法の問題点を解決する手法を提案する。一般にメモリには頻りに書き変わる部分とほとんど書き変わらない部分がある。前者には活発なプロセスのヒープ、スタック等が、後者には活発でないプロセスのヒープ、スタックやオペレーティングシステム、共有ライブラリのコード等が該当する。仮想マシンがライブマイグレーションで以前実行されたことのあるホストに移動する場合を考える。このとき前回の実行時のメモリ内容がホストに保存してあれば、前回の実行から書き変わっていない部分は再利用することができる。本研究では以下の手法によってネットワーク越しに転送されるメモリ量を減らし、高速でかつ仮想マシンの性能低下も少ないライブマイグレーションを目指す。

- (1) 基本となるライブマイグレーションには post-copy 型を用いる。
- (2) 仮想マシン  $V$  は各メモリページごとに以下のように動作するカウンタを持つ。
  - (a)  $V$  の起動時に 0 で初期化される。
  - (b) 対応するメモリページが変更されるたびに値が 1 増加する。
- (3)  $V$  がホスト  $A$  から別のホストへ移動したとき、

<sup>†</sup> 東京大学大学院情報理工学系研究科

The University of Tokyo

<sup>††</sup> 独立行政法人産業技術総合研究所

National Institute of Advanced Industrial Science and Technology

<sup>†††</sup> 国立情報学研究所

National Institute of Informatics

- $V$  のメモリ内容とカウンタを  $A$  のメモリ空間内に保持する
- (4)  $V$  が再び  $A$  に移動したとき、ページフォールトの際のメモリコピーを以下のように行う。
- (a) 当該メモリページについて  $A$  に保持されているカウンタの値と  $V$  の持つカウンタの値が一致すれば  $A$  のメモリから  $V$  のメモリへ直接コピーし再利用する。
- (b) そうでなければ通常の post-copy 型と同様にネットワーク越しにコピーする。

以上を図で表すと図 1 のようになる。図の  $A, B$  はホストを、 $V$  はホスト上で実行される仮想マシンを表す。また各四角形はそのマシンのメモリを表す。二段目において移動直後の仮想マシンはメモリが空になっており、更新されていない部分（図の青）は再利用を行い、更新された部分（図の赤）はネットワーク越しに転送する。なお本手法は  $A$  からの移動先とその後の  $A$  への移動元が別のホストでも適用可能である。

本手法は post-copy 型ライブマイグレーションを基本にしているため仮想マシンの実行ホストの切り替えはすぐに完了する。さらに従来の post-copy 型ライブマイグレーションで問題であった移動後の仮想マシンの性能低下もメモリの再利用によって軽減すると考えられる。この性能低下はライブマイグレーションを行うたびに起こるため、第 1 章で述べたように仮想マシンを頻繁に移動させる場合に本手法は特に有効である。

#### 4. 設計と実装

本稿の提案を実装するためには以下が必要である。

- 仮想マシンのメモリ内容の保持
  - 仮想マシンのメモリページごとのカウンタの保持  
前者については既存のライブマイグレーションの実装を移動元のメモリを解放しないように変更する。後者については仮想マシンモニタのメモリ管理機構を変更し、各メモリページについてカウンタの値を記録する。
- 仮想化環境ではメモリページは仮想アドレス、物理アドレス、マシンアドレスの三通りで表わせる。マシンアドレスは同じメモリページであっても仮想マシンが移動すると変わるため本手法には適さない。また仮想アドレスを用いると再利用する際に仮想アドレスから物理アドレスの変換が必要になる。従って本手法ではメモリページを表すのに仮想マシンの物理アドレスを用いる。

#### 5. おわりに

本稿では、仮想マシンが以前実行された際のメモリ

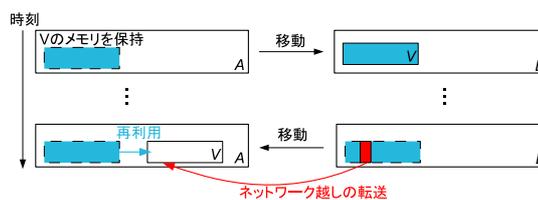


図 1 本手法の模式図

を再利用することによって、ライブマイグレーションが頻繁に起こる場合に問題となる移動時間の長期化と移動後の性能低下を抑制する手法を提案した。今後の課題は以下である。

- (1) 実装と評価
- (2) 本稿で提案しなかった手法との組み合わせによるさらなるライブマイグレーションの性能改善  
組み合わせる手法には以下のようなものが考えられる。
  - メモリ間のコピーではなく、仮想マシンの物理アドレスからホストのマシンアドレスのマッピングを変更することにより再利用を実現する。仮想マシンモニタへの変更が大きくなる代わりにオーバーヘッドが減ると考えられる。
  - 活発なプロセスのワーキングセットを pre-copy することで移動後のページフォールトを減らす。
  - メモリを圧縮して転送量を減らす。特にページ境界にとらわれない圧縮方法の研究が必要である。
  - プログラムを独立実行可能な部分にわけ、実行中の部分の次の部分を利用することで移動後に即座に必要なメモリ領域を予測する。

#### 参考文献

- 1) Vahid Kazempour *et al.*, "AASH: an asymmetry-aware scheduler for hypervisors", *VEE*, pp. 85–96, Pittsburgh, USA, March 2010
- 2) Timothy Wood *et al.*, "Memory buddies: exploiting page sharing for smart colocation in virtualized data centers", *VEE*, pp. 31–40, Washington, DC, USA, March 2009
- 3) Christopher Clark *et al.*, "Live Migration of Virtual Machines", *NSDI*, pp. 273–286, Boston, USA, May 2005
- 4) H. Andrés Lagar-Cavilla *et al.*, "SnowFlock: Rapid Virtual Machine Cloning for Cloud Computing", *EuroSys*, pp. 1–12, Nuremberg, Germany, April 2009
- 5) Takahiro Hirofuchi *et al.*, "Enabling Instantaneous Relocation of Virtual Machines with a Lightweight VMM Extension", *CCGrid*, pp. 73–83, Melbourne, Australia, May 2010