B-052

仮想機械ライブマイグレーションの統合方式
An Integrated Method of VM Live Migration

都築 俊徳† Toshinori Tsuzuki
梅澤 猛† Takeshi Umezawa
大澤 範高† Noritaka Osawa

1. はじめに

サーバ上の仮想機械（VM）を無停止で別の物理サーバへ移送させるライブマイグレーションは，負荷分散あるいは集約による省電力化により高効率な運用を実現する技術として注目されている．しかし，対象となる VM に合わない手順でマイグレーションを行うと，性能の著しい低下や無用なトラフィックの発生を招く恐れがあるため，これを回避する手法が求められている．

そこで本研究では，VM のメモリで一定時間内にアクセスされるページの集合（WS: Working Set）の中で書き換えがなされるページの集合（WWS: Writable WS）[1]，また読み込みのみがなされるページの集合（RWS: Read-only WS）に着目し，各 VM の特性に合わせたライブマイグレーション手法を提案する． WS に含まれ，WWS に含まれないページの集合が RWS である．

2. 既存ライブマイグレーション手法と問題点

ライブマイグレーションは，稼働している VM の主記憶上の使用中ページと CPU やデバイスの状態などの VM コンテキストを別のサーバへと転送することで実現される．ライブマイグレーションは以下の 3 つのフェーズに分けることができる[1].

- Push フェーズ
  →VM の稼働を続けながらページを別サーバに転送する
- Stop-and-copy フェーズ
  →VM を停止し，ページ，VM コンテキストを転送する．
- Pull フェーズ
  →別サーバで VM の稼働を再開し，未転送のページを元のサーバから取得する

既存のライブマイグレーションの多くは，Push フェーズにおいて可能な限りメモリページを転送する方式[1]を採用している（pre-copy 方式）．図 1 に pre-copy 方式によるライブマイグレーションの概要を示す．この方式では，Push フェーズで大量のページ転送が発生する．まず，稼働中の VM の全メモリページの転送を試み，続いて転送中に書き換えられたページの再送を行う．ページ転送はページ再送が収束するか，再送回数があらかじめ設定された上限に達するまで続く．Stop-and-copy フェーズでは VM を停止し，残りのページと VM コンテキストを転送し，移送先で VM の稼働を再開する．Pull フェーズでのページ転送は行わない．このため，転送するページの書き換えが多い場合，ページ再送が頻発し，Stop-and-copy フェーズでの転送に含めるページが増加するという欠点がある．その結果，マイグレーションにかかる総時間，VM の停止時間，ネットワークトラフィックが増加してしまう．

†千葉大学大学院融合科学研究科, Graduate School of Advanced Integration Science, Chiba University

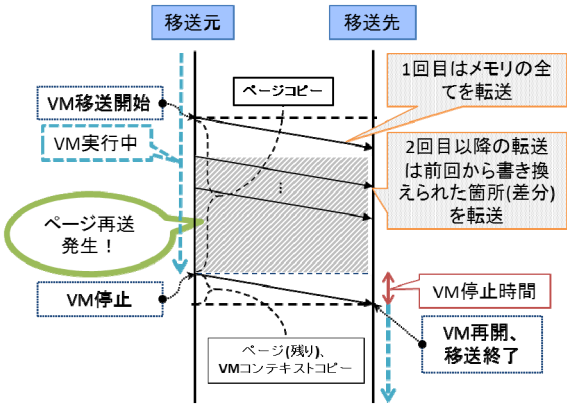

図 1　pre-copy の動作

他に Pull フェーズでメモリページの大部分を転送する方式[2]が提案されている（post-copy 方式）．図 2 に post-copy 方式によるライブマイグレーションの概要を示す．この方式では Push フェーズでページ転送は行われず，まず Stop-and-copy フェーズで VM を停止し，ページング不可メモリとプロセッサ状態等の VM コンテキストといった最小限の情報を転送し，移送先で VM の稼働を再開する．その後，Pull フェーズに入り，デマンドページングやプリページングによって他のページを転送していく．この方式では，ページ転送は VM が移送先で動作している Pull フェーズで行われるため同一ページが繰り返し転送されない．このため，転送される総ページ数は一般に pre-copy 方式と比べて少なくなる．しかし，移送した VM の稼働を再開した際にページフォルトが発生するとネットワーク越しにページを転送する必要があるため，プログラム実行の遅延が大きくなることが問題となる．

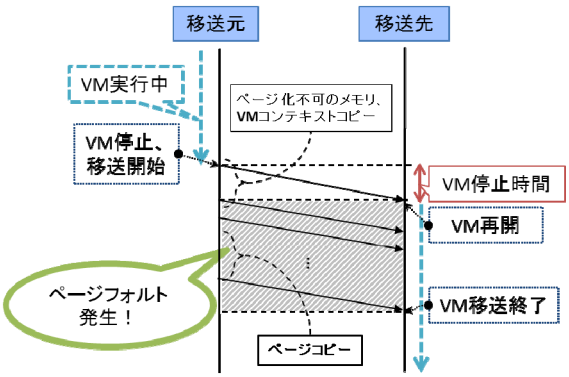

図 2　post-copy の動作

3. 統合方式

本研究では移送する VM のメモリがアクセスされるパターンをもとに既存のライブマイグレーション手法を統合する．無用なページ再送を避けるため，Push フェーズでは書き換えが発生しないページに限定して転送を行う方針である．一方で，Pull フェーズではアクセスされただけでページフォルトが発生するので，ネットワーク越しのページ取得を避けるために，WS で読み込みのみがなされるページの集合である RWS を極力 Push フェーズで転送しておくこととする．

移送する VM の停止時間の許容最大値を x (s)とし，ネットワークの実効転送レートが y (bit/s)とすると，停止時間中に送信する WWS として許容できる大きさ z (bit)は式(1)で表せる．

$$z = x \cdot y \quad (1)$$

WWS のサイズが z を超える場合，収まらなかった部分は VM の稼働を移送先で再開してからの Pull フェーズで優先的に転送する．提案する統合方式を図 3 に示す．

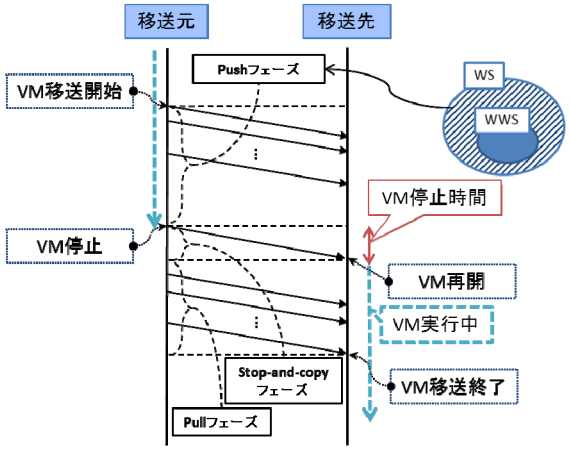

図 3　統合方式の動作

提案する統合方式によるライブマイグレーション手順は次の通りである．

1. Push フェーズ：アクセスパターンの予測から RWS を移送先サーバに転送する
2. Stop-and-copy フェーズ：VM を停止し VM コンテキストと許容する停止時間内に送ることができる WWS のページを転送する
3. Pull フェーズ：移送先で VM の稼働を再開し，未転送のページを転送していく（WWS のページが残っていれば優先的に転送）．未転送のページのページフォルトが発生した場合は対応するページを転送する

メモリへのアクセスパターンが既知であると仮定したとき，次に挙げる指標により，提案方式と既存手法を比較する．

- 総トラフィック量
- 総時間（VM 移送の開始から終了までの時間）
- ページフォルト数
- VM 停止時間

転送するメモリの総量を M (bit)，VM コンテキストの転送や VM の停止，再開にかかる時間を h (s) とおく．すると，マイグレーションにかかる最小の時間，つまり各ページを一回のみ送信してマイグレーションを行うのにかかる時間 T_{min} は，$T_{min}=M/y+h$ となる．この間のメモリアクセスをもとにページを読み込みのみされるページ，書き込みされるページ，アクセスされないページの 3 種類に分ける．それらの種類のメモリ量(bit)をそれぞれ r, w, n とする．

Pre-copy はまず全てのページを転送するため，w だけ再送する必要がある．このため，総時間は w/y だけ増加する．これらのページを VM 停止中に転送するため，VM の停止時間は$(w/y+h)$となる．Post-copy はすぐに VM の実行サーバを切り替えるため，ページに何らかのアクセスがなされるとページフォルトとしてネットワーク越しにページを取得する必要がある．プリページングで読み込み，書き込みされるページが転送できていない場合，最大で$(r+w)$だけページフォルトが発生する．各ページフォルトにかかるオーバヘッド時間を L(s)とすると，$(r+w)L$ だけマイグレーションの総時間が増える．提案手法では，停止時間を制限しないとするとページフォルトは発生しないが，VM の停止時間は w/y となる．しかし，総トラフィック量は M であり，総時間も T_{min} と最小限となる．表 1 に比較結果をまとめる．表 1 から提案手法が VM 停止時間以外の 3 つの指標で最小値をとり，メモリアクセスを正しく予測できれば有効であることが分かる．VM 停止時間は post-copy より長いが，これはページフォルト数とのトレードオフとなるので VM が提供するサービスによって VM 停止中に転送するページの適切な量を判断する必要がある．

表 1　メモリアクセスが既知の場合の
ライブマイグレーション手法の比較

	提案手法	Pre-copy	Post-copy
総トラフィック量	M	$M+w$	M
総時間	T_{min}	$T_{min}+w/y$	$T_{min}+(r+w)\,L$
ページフォルト数	0	0	$r+w$
VM 停止時間	$w/y+h$	$w/y+h$	h

4. おわりに

本研究では，VM のメモリがアクセスされるパターンに着目し， 既存のライブマイグレーション手法である pre-copy と post-copy を VM の性質に応じて適切に組み合わせる統合方式を提案した．アクセスパターンが既知である場合には，提案手法が VM のマイグレーションにおける総トラフィック，総時間，ページフォルト数において既存手法を上回る性能をもつことを示した．今後は，本手法の前提となるページ分類のためのメモリアクセス予測を高精度・低オーバヘッドで実現する手法について検討すると共に，実環境における評価を行いたい．

参考文献

[1]Christopher Clark, Keir Fraser, Steven Hand, Jacob Gorm Hansen, Eric Jul, Christian Limpach, Ian Pratt, Andrew Warfield, “Live migration of virtual machines,” Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation, p.273-286, May 02-04, (2005).

[2] Michael R. Hines, Umesh Deshpande, Kartik Gopalan, “Post-copy live migration of virtual machines,” ACM SIGOPS Operating Systems Review, v.43 n.3, (2009).