Vol.2012-ARC-200 No.11
Vol.2012-OS-121 No.11
2012/5/8

# 仮想マシンの超広域ライブマイグレーションにむけたベストエフォート型状態同期機構の試作

広渕 崇宏[1]　マウリシオ ツガワ[2]　中田 秀基[1]　伊藤 智[1]　関口 智嗣[1]

**概要**：仮想マシンの遠隔ライブマイグレーションを用いれば、災害発生時にサーバを安全な遠隔拠点に待避できる。しかし、ライブマイグレーションは大量のデータ転送を伴うため、ネットワーク帯域が限られる WAN 環境において、災害発生直後の限られた猶予時間内に再配置を完了することが難しい。本稿では、広域ライブマイグレーション時間を短縮するため、仮想マシンの実行状態をベストエフォートで事前に同期する手法を提案する。平時から仮想ディスクの状態を待避先拠点とできる限り同期しておき、ライブマイグレーションを実行する際にはその差分のみを転送する。マイグレーションに伴うデータ転送量およびマイグレーション時間を大幅に削減できる。同期データに対して世代管理を行うことで、可用帯域が許す範囲での部分的な同期を可能にし、一時的な通信の断絶に対しても再び途中から同期を再開可能にしている。提案機構のプロトタイプを実装し、日米間のネットワークを用いて基礎的な評価実験を行った。仮想マシンの実行状態全体を約 30 秒で再配置できた。提案機構を用いない場合（約 1 時間）よりも大幅に待避時間を短縮できた。

# Lessons Learnt from a Preliminary Prototype of a Best-Effort Pre-synchronization Mechanism for Wide-Area Live Migration of Virtual Machines (Work-in-Progress Report)

Takahiro Hirofuchi[1]　Mauricio Tsugawa[2]　Hidemoto Nakada[1]　Satoshi Itoh[1]
Satoshi Sekiguchi[1]

***Abstract:*** Wide-area VM live migration is a promising technology that can be used to evacuate virtualized servers to safe locations upon a critical disaster. However, existing live migration mechanisms require the transfer of a large amount of data, which makes it difficult to relocate a VM under restricting conditions of a disaster site - e.g., limited network performance and time window with electrical power. In this work-in-progress report, we propose a mechanism to partially synchronize the execution state of a VM between two sites. The synchronization is performed in a best effort basis in order to not disturb the performance of systems during normal operation. By transferring the necessary data in advance, the proposed mechanism can dramatically reduce the amount of data transferred and the time required for a live VM migration. By applying versioning control on the synchronized data, the proposed mechanism tolerates unstable network connections and works correctly even if communication is temporarily lost. We have developed a preliminary prototype and evaluated the mechanism using servers in Japan and the United States. It was possible to relocate a VM from Japan to the United States in approximately 30 seconds - a dramatic reduction of migration time considering that it would take approximately 1 hour if the proposed mechanism is not used.

## 1. はじめに

2011 年 3 月 11 日に発生した東北関東大震災においては、東日本の広範囲にわたって大規模な停電が発生し、多くの官公庁や民間事業所においてサーバ機器が停止する事態が発生した。平時において我々が享受できる IT サービスが失われたことは、事業を継続する上でも非常時対応を円滑に進める上でも大きな支障となった。大規模な災害に耐えうる IT インフラ技術の必要性が改めて認識されるところとなった。

我々は仮想マシンのライブマイグレーションを用いたサーバの遠隔待避手法に着目している。今回の地震による計算機センタの被害状況を調査した結果、地震発生直後もサーバおよびネットワークは一定時間動作していたことがわかった。この動作猶予時間内にサーバを遠隔の安全な拠点に移動することで、IT サービスを引き続き提供できると考える。しかし、従来のライブマイグレーション技術は高速な LAN 環境を前提として設計されており、遠隔環境において用いることは現実的ではない。仮想マシン実行状態の移動には大量のデータ転送を伴うため、限られた動作猶予時間内にマイグレーションを完了することが難しい。

そこで、広域ライブマイグレーション時間を短縮するため、仮想マシンの実行状態をベストエフォートで事前に同期する手法を提案する。平時から仮想マシンの実行状態（メモリページおよび仮想ディスク、ただし本稿で焦点を当てるのは仮想ディスクのみ）を待避先拠点にできる限り転送しておき、ライブマイグレーションを実行する際にはそれらの差分のみを転送することで、マイグレーションに伴うデータ転送量およびマイグレーション時間を削減する。同期データに対して世代管理を行うことで、可用帯域が許す範囲での部分的な同期を可能にし、一時的な通信の断絶に対しても再び途中から同期を再開可能にしている。高可用性サーバ向けの仮想マシン同期技術とは異なり、状態同期処理が仮想マシンの実行速度に影響を与えない。

本稿では仮想ディスクを対象にプロトタイプを実装し、基礎的な動作確認を行った。2 節において、広域ライブマイグレーションによるサーバ待避技術の可能性について説明する。3 節で提案機構について説明する。4 節でプロトタイプの動作確認について述べる。5 節で関連研究にふれ、6 節で本稿をまとめる。

## 2. 広域マイグレーションによるサーバ待避の可能性

我々は、今回の震災において大きな揺れを経験した公的研究機関に協力して頂いて、各計算機センタにおけるサーバ機器について被害状況の聞き取り調査を行った [1]。いずれの研究機関においてもサーバ機器そのものに対する物理的な被害は極めて少なかったことがわかった。適切に耐震施工された建物内に、適切にサーバ機器が設置されている限りにおいては、物理的な衝撃によってサーバ機器が損壊することは極めて希であった。またいずれの研究機関においても、電力会社からの給電が途絶えたものの、UPS や自家発電装置等のバックアップ電源設備によって一定時間（数十分から数時間程度）電源を確保することが可能であった。ネットワーク到達性に関しては、研究機関を結ぶ基幹ネットワークで通信回線の損傷が一部あったものの、通信経路が冗長化されていたため到達性が維持されていた。基幹ネットワークと計算機センタを結ぶネットワーク機器に対して電力供給が維持されていた場合においては、その間外部とのネットワーク到達性が維持されていた。

つまり、大規模な地震が発生したとしても、適切に設置されているサーバの物理的な被害はほぼ存在せず、その直後しばらくの間はネットワークおよびサーバ機器が引き続き動作すると仮定できる。我々は、この猶予時間を積極的に有効活用する手法として、仮想マシンの広域マイグレーションによるサーバ遠隔待避の可能性に着目している。地震等により大規模な停電を検知すると、バックアップ電源の動作期間内に順次サーバを遠隔に待避する。

突然のハードウェア故障や停電に耐えうるサーバ運用手法として、高可用性 (High Availability) サーバ技術が存在する。運用系サーバおよび待機系サーバという 2 種類のサーバを地理的に離れた 2 拠点に設置し、ネットワークを介して両サーバ間で実行状態を同期する。万が一運用系サーバにおいて障害が発生したとしても、待機系サーバがすぐさま自動的に実行を引き続ぐことで、途切れることなくサービスを運用できる。しかし、運用系の状態変化を逐次的に待機系に同期するため、高速かつ大容量の通信回線を必要とする。高額な導入コストおよび維持コストに見合う、限られた環境においてのみ導入が進んでおり、広く一般に普及しているとは言い難い。仮想マシンの広域マイグレーションによるサーバ待避は、高可用性サーバ技術よりは信頼性に劣るものの、それを補完するものとして、より安価で広く一般に導入しやすい災害対策技術となる可能性があると考えている。

## 3. 提案機構

本稿では仮想ディスクに対する提案機構についてのみ述べる。地震等によって停電が発生した後、バックアップ電源がサーバやルータに電力供給できる時間内に仮想マシンを遠隔拠点に待避しなければならない。マイグレーション開始後、仮想マシンの全てのステートを短時間で移動元拠点から取り除く必要がある。仮想マシンを構成する状態に

[1] 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology (AIST)
[2] フロリダ大学
University of Florida

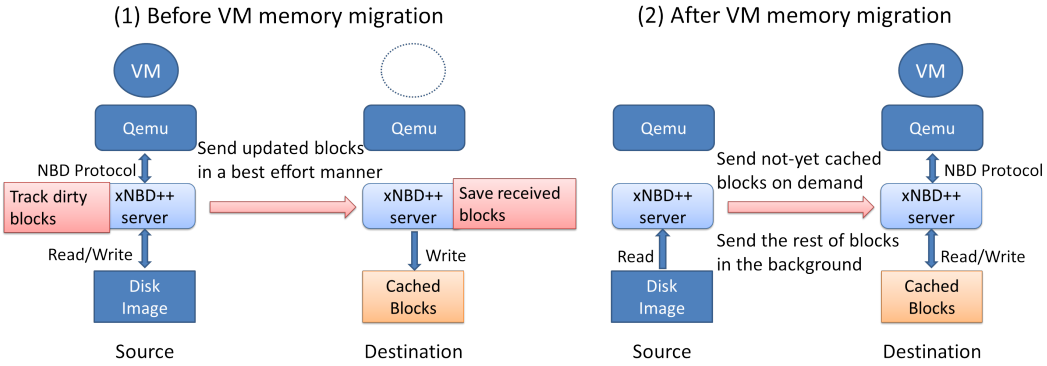


**図 1** 提案機構の概要

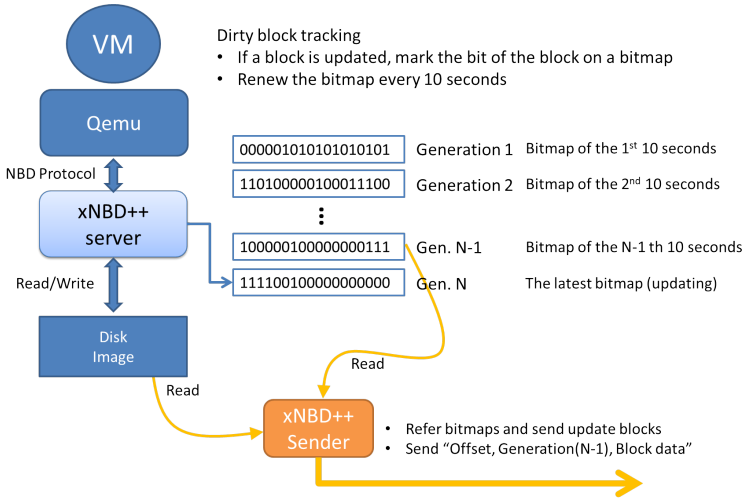


**図 2** 提案機構の送信サーバ（10 秒ごとにビットマップを更新する場合）

おいて最も大きな要素である仮想ディスクは、一般的に数 GB から数十 GB 程度もありその全てを一度に猶予期間中に転送することは困難である。

### 3.1 提案機構の概要

そこで、仮想ディスクのデータをあらかじめ普段から待避先拠点とベストエフォートで同期しておき、マイグレーション開始後のデータ転送量を削減する機構を提案する。提案機構の概要を**図 1** に示す。移動元および移動先拠点それぞれにおいて提案機構は、仮想マシンに対して NBD（Network Block Device）プロトコル [2] のストレージサーバとして動作し、仮想ディスクを仮想マシンに対して提供する。

マイグレーション開始前（定常状態）においては、提案機構が仮想マシンによって更新されたディスクブロックを検知し、更新されたブロックのデータを移動先拠点にベストエフォートで転送する。更新ブロックの転送処理は、後述するように仮想マシンのディスク I/O 処理とは独立して実行され、仮想マシンのディスク I/O 処理の速度は WAN の遅延や可用帯域の影響を受けない。この普段の状態において、移動先拠点には、移動元拠点の仮想ディスクの内容がある程度キャッシュされている。

次に、停電等が発生すると、仮想マシン本体（メモリ）のマイグレーションを行い実行ホストを待避先に切り替える。仮想ディスクのある程度のデータは既に移動先拠点にキャッシュされているものの、残りのデータは未だに移動元拠点に残っている可能性がある。もし仮想マシンが未だにキャッシュされていないデータにアクセスした際には、オンデマンドに移動元拠点からデータを取得する。また並行して残りのデータを移動先拠点に全て転送する。最終的に全てのデータが移動先拠点に転送されると、仮想マシンの状態全体のマイグレーションが完了し、移動元拠点に対する依存性が解消できる。

### 3.2 WAN 環境に対するベストエフォートでのディスク同期

定常状態においては、ディスクの更新内容を移動先拠点へ順次反映させる。しかし、WAN 環境は常に安定した通信が可能な LAN 環境と比較して、時としてネットワーク遅延や帯域が変動する。また、待避先拠点のメンテナンス

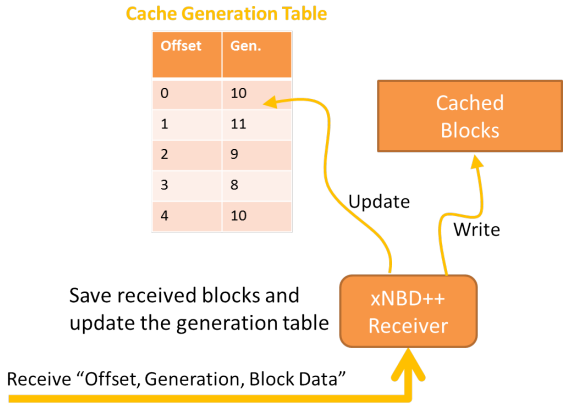


**図 3** 提案機構の受信サーバ

等により、ディスク同期の通信が途切れる可能性もある。そこで、提案機構においては、同期データに対して世代管理を行うことで、可用帯域が許す範囲での部分的な同期を可能にし、一時的な通信の断絶に対しても再び途中から同期を再開可能にしている。

ディスク同期の動作を**図 2** および**図 3** を用いて説明する。移動元拠点において、提案機構が提供するストレージサーバは更新ブロックをビットマップファイルで記録する。あるブロックに対する書き込みリクエストを受信すると、ビットマップにおいてそのブロックに相当するビットを立てる。定期的に新たなビットマップファイルを作成し、以降の更新ブロックは新たなビットマップファイルに記録する。便宜上、ビットマップファイルが作られた順に、1 世代目のビットマップファイル、2 世代目のビットマップファイルなどと呼ぶ。最新のビットマップファイルを N 世代目とする。

提案機構が提供する送信サーバ（Sender）は、新たなビットマップファイルが作成されるたび、更新が完了したばかりの、一世代前（N-1 世代）のビットマップファイルを走査する。更新ブロックを見つけると、仮想ディスクイメージの該当ブロックを読み取り、待避先へ「ブロックのオフセット、世代番号（N-1)、ブロックの内容」として転送する。ストレージサーバと送信サーバはそれぞれ非同期にディスクイメージにアクセスする。送信サーバがディスクイメージの該当ブロックを読み取った時点で、すでにブロックの内容がストレージサーバによって N-1 世代目以降に更新されている可能性がある。送信データは「あるブロックの内容は、世代番号 N-1 のある時点で読み取ったものである」ということを意味する。実行ホスト切り替え時にキャッシュ済みブロックの判別を適切に行うので問題はない。待避先の受信サーバは（Receiver）は、移動元から受信したブロックの内容をキャッシュイメージの指定されたオフセットに保存する。またキャッシュ世代テーブル（Cache Generation Table、CGT）に、キャッシュイメージの該当ブロックに何世代目以降のデータが保存されているかを記録する。

以上の動作は、不安定な WAN 環境においても、可用帯域が許す範囲で仮想ディスクを同期することを可能にする。DRDB 等の厳密なディスク同期を提供するシステムにおいては、仮想ディスクの I/O 処理速度が結果的に遅くなることで、ディスクの同期状態を維持する。しかし、停電後の猶予時間の存在を前提とする提案機構では、厳密なディスク同期は不要であるため、仮想ディスクの I/O 処理とは独立して現状の可用帯域が転送できる範囲内でディスクの同期を行う。仮想ディスクの I/O 処理は可用帯域の影響を受けない。送信サーバは、あるビットマップファイルに関するデータ転送中に次のビットマップファイルが生成されたことを検知すると、更新ブロックの転送速度が更新ブロックの生成速度よりも遅いと判断する。現在処理中のビットマップファイルのデータ転送を途中で取りやめ、次のビットマップファイルの走査に取りかかる。

ネットワーク到達性が一時的に失われた場合でも再び途中から同期を再開できる。例えば待避先拠点のメンテナンスなどでディスク同期が突然中断しても、世代番号で同期状態を管理するため一貫性を保ったまま途中から再開できる。また、同期に用いる通信プロトコルとして、再送制御を行わない UDP や DCCP 等も原理的には使用可能である。

### 3.3 実行ホスト切り替え時の動作

**図 4** において、実行ホスト切り替え時の動作について説明する。仮想マシンの遠隔待避を開始すると、最初に仮想マシン本体（メモリ）のライブマイグレーションを実行する。このとき、移動元で仮想マシンの実行を停止し待避先で実行を再開する直前に、提案機構は次の動作を行って待避先にキャッシュできたブロックを把握する。

(1) 移動元拠点で生成された全てのビットマップファイルを読み取って、最終世代テーブル（Final Generation Table、FGT）を生成する。FGT は各ブロック番号に対する世代番号が記録されており、そのブロックが最後に更新された世代番号を記録している。
(2) FGT を移動元拠点から待避先拠点に転送する。
(3) FGT と CGT を比較し、各ブロックのキャッシュが有効なものかどうか判別する。各ブロック番号に対して、FGT と「少なくとも何世代以降のデータが保存されているのか」を示す CGT を比較し、両者が同一の世代番号であればキャッシュが有効である。有効なキャッシュブロックを表すビットマップファイル（キャッシュビットマップファイル）を生成する。
(4) 提案機構のストレージサーバをプロキシモードで起動する。その際、キャッシュビットマップファイルおよびキャッシュイメージを指定する。

その後、待避先で仮想マシンの実行を再開する。

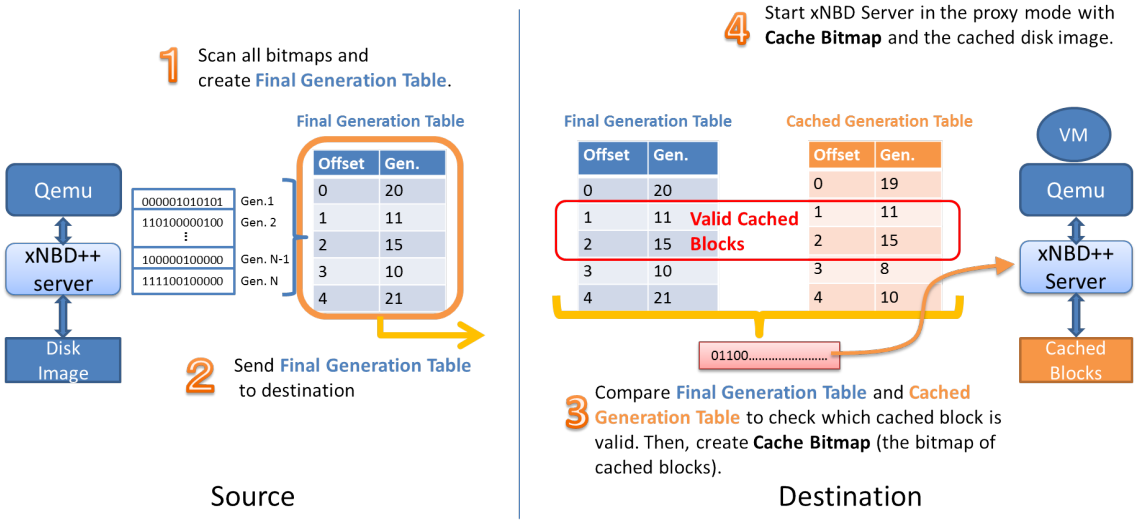


**図 4** 提案機構における実行ホスト切り替え時の動作概要

以降の動作は、我々が過去に提案したポストコピー型のストレージマイグレーション [3] に従う。移動先拠点の提案機構が仮想マシンによってアクセスされたブロックを検知し、移動先拠点に存在しないデータ（未キャッシュのデータ）に対する読み取りであれば、移動元拠点からデータをオンデマンドに転送する。データの書き込みおよび移動先拠点に存在するデータに対する読み取りに対しては、キャッシュ済みの如何に関わらず移動先拠点に保存する。またオンデマンドの転送と並行して、バックグラウンドで残りのブロックを移動先に転送する。すべてのブロックを転送すると仮想ディスクのライブマイグレーションが完了する。

## 4. 実装

提案機構のプロトタイプを実装した。我々が開発しているポストコピー型のストレージマイグレーション機構 xNBD[4] を拡張した。TCP および DCCP による同期データの転送に対応している。DCCP は輻輳制御は行うものの再送制御は行わないメッセージ指向の転送プロトコルである。仮想マシンの実行にともなって転送中のデータがすでに古くなる可能性があるため、再送制御を行わない DCCP の方が TCP よりも効率的な状態同期が行える可能性がある。

## 5. 評価実験

プロトタイプの動作確認のため簡単な実験を行った。実験環境を**図 5** に示す。移動元拠点として日本のつくば市に位置する産総研内の計算機センタを、また移動先拠点としてアメリカ合衆国フロリダ州のフロリダ大学に位置する計算機センタを用いた。それぞれの拠点において 2 台の物理計算機を用意し、提案機構を動かすストレージサーバノード（xNBD A/B）、仮想マシンを動かすホストノード（Host A/B）として設定した。仮想ディスクに対する I/O トラフィックは各拠点のプライベートネットワークを通す。ライブマイグレーションおよびディスクの同期トラフィックはインターネットを経由する。評価実験時、両拠点間の通信遅延（RTT）は `ping` の計測により 180ms 程度であった。両拠点間のネットワーク帯域は変動量が大きいもの、評価実験時は `iperf` の計測により 5Mbps から 20Mbps 程度であった。

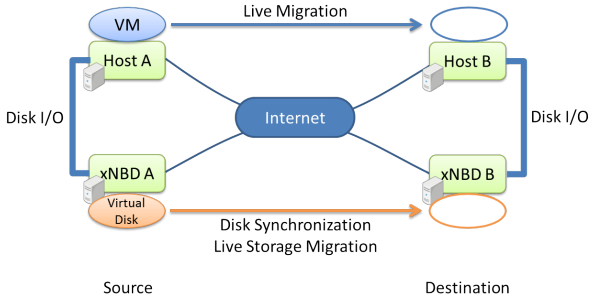


**図 5** 実験環境（移動元:つくば市産業技術総合研究所、移動先:アメリカ合衆国フロリダ大学）

仮想マシンモニタ Qemu/KVM（`qemu-1.0.1`）を用いて仮想マシンを起動する。仮想マシンに対して、512MB のメモリおよび 2GB の仮想ディスクを割り当てた。仮想ディスクのイメージファイルはストレージサーバ上に存在する。あらかじめ移動元および移動先拠点それぞれのストレージサーバに対してイメージファイルを転送しておき、提案機構はイメージファイルからの更新差分のみを同期した。提案機構は 10 秒ごとに更新ブロックを記録するビットマップファイルを更新する設定とした。10 秒ごとにビットマップファイルを走査し、更新されたブロックを移動先に転送する。

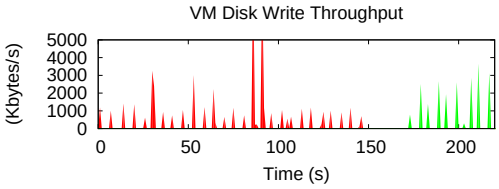


**図 6** 仮想マシンのディスク I/O 通信量（読み込み）

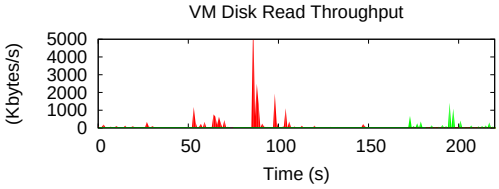


**図 7** 仮想マシンのディスク I/O 通信量（書き出し）

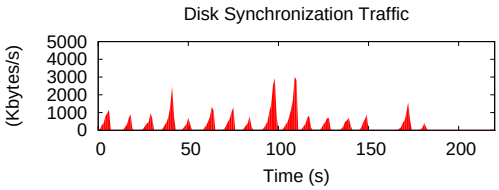


**図 8** 仮想ディスク同期の通信量

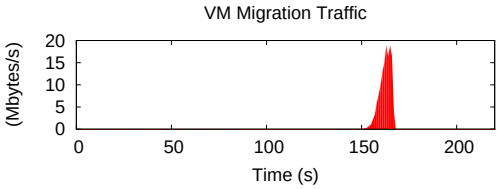


**図 9** 仮想マシン本体（メモリ）のライブマイグレーション通信量

ゲスト OS 上で Linux のカーネルコンパイルを開始し、その実行中に仮想マシンおよび仮想ディスクのライブマイグレーションを行った。仮想マシン本体のライブマイグレーションとして Qemu/KVM のプレコピー型マイグレーションを、仮想ディスクのライブマイグレーションとして提案機構を用いた。ディスク同期には TCP コネクションを用いた。

本実験において、仮想ディスクを含む仮想マシン全体のライブマイグレーションを約 30 秒で完了できた。もし提案機構を用いることなく仮想ディスクを一度に再配置していたならば、2GB 以上のデータを転送する必要があるため、評価実験時のネットワーク状態では 1 時間程度を要したと想定される。提案機構によってあらかじめ仮想ディスクの内容を同期しておくことにより、仮想マシンの状態全体の再配置時間を大幅に短縮することができた。

マイグレーション前後における仮想マシンのディスク I/O 通信量を**図 6** および**図 7** に示す。また提案機構による仮想ディスク同期の通信量を**図 8** に示す。仮想マシン本体のライブマイグレーションは各図中 150 秒付近から開始

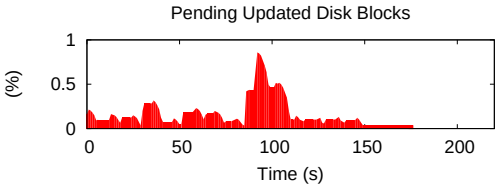


**図 10** 仮想ディスクの同期待ちブロック数

している。ライブマイグレーションの前後、移動元および移動先拠点それぞれにおいて、仮想マシンはカーネルコンパイルのためにソースファイルが存在するディスクブロックを読み込みながら、生成されたオブジェクトファイルをディスクに書き出している。ディスク同期の通信量より、提案機構は書き出されたディスクブロックを 10 秒ごとに移動先に転送していることがわかる。

**図 9** に仮想マシン本体のライブマイグレーションによる通信量を示す。150 秒から 170 秒付近の約 20 秒間において、仮想マシン本体のマイグレーションにともなうデータ転送が行われている。マイグレーションの最終段階において仮想マシンの動作を停止するまで仮想マシンは移動元で動作し続けている。しかし、Qemu/KVM におけるマイグレーション処理のオーバーヘッドにより、その間、実際には仮想マシンのディスク I/O がかなり抑制されていた。仮想ディスクの同期通信量を示す図 8 において、170 秒から 175 秒付近で発生したデータ転送は FGT の転送によるものである。FGT のデータサイズは 2GB の仮想ディスクの場合 4MB 程度である。提案機構は FGT の転送のために新たな TCP コネクションを作成する。FGT のデータサイズは大きくないものの、遅延の大きい実験環境においては輻輳制御ウィンドウが即在に拡大しないため、転送に時間がかかっている。FGT の転送中は仮想マシンを停止しているため、ダウンタイムを削減するためには FGT のデータサイズの抑制が必要になる。仮想マシンの実行ホストが 175 秒付近で移動先に切り替わった。180 秒付近のデータ転送は、提案機構が移動元拠点に存在する残りのディスクブロック（約 200 ブロック）をすべて移動先に転送したものである。

**図 10** は、移動先に対して同期できていないディスクブロックの数について、仮想ディスクの全ブロック数に対する割合を示している。カーネルコンパイルによる仮想ディスクの更新量はネットワーク帯域に比べ十分小さかったため、同期できていないディスクブロックの割合は、常に仮想ディスク全体の 1%（約 20MB）未満に抑制できていた。マイグレーションをいつ開始したとしても、ほぼ同様の所要時間で全ての状態を遠隔拠点に移動できたはずである。

## 6. 今後の課題

プロトタイプの実装および簡単な評価実験を通して、提

案機構の基本的な有効性を確認できた。本節では今後の課題について述べる。

本稿の評価実験では、ディスクの同期速度がディスクの更新速度よりも常に大きかった。今後は、ディスクの同期速度が更新速度よりも小さく、定常状態において十分なディスク内容の同期ができない場合についても評価する。現状の実装では、同期速度が更新速度よりも遅いと判断した場合はできる限り更新ブロックを同期し、間に合わなかった更新ブロックは放棄してしまう。しかし、今後、間に合わなかった更新ブロックを残しておき、後から余裕ができた時（同期速度が更新速度を上回った時）に転送するよう改良する予定である。

インターネットを経由する実験環境において、利用できるネットワーク帯域は常に不安定であった。仮想マシン本体のライブマイグレーションの完了時間は、プレコピー型ライブマイグレーションにおいてはメモリの更新速度および可用帯域によって左右される。メモリの更新速度がネットワーク越しの転送速度を上回れば、マイグレーションがいつまでも完了しない可能性もある。今後、ポストコピー型ライブマイグレーションを検討する余地があると考える。

仮想マシンのメモリに対しても提案機構を実装することで、仮想マシン全体の再配置時間をさらに抑制できると考えている。仮想マシンの利用形態にも依存するものの、必ずしも全てのメモリ領域が高い頻度で更新されるわけではない [5] ため、提案機構による状態同期がマイグレーション時間の短縮に貢献する可能性がある。

## 7. 関連研究

CloudNet[6] では、仮想マシンを遠隔拠点に仮想ディスクも含めてライブマイグレーションするために、分散ミラーリングシステムである DRDB を用いている。マイグレーションを開始した直後は、DRDB を非同期モード（Protocol A）に設定し、仮想マシンに対してディスク I/O を提供するのと並行して、移動元（DRDB におけるプライマリ）から移動先（DRDB におけるセカンダリ）にディスク内容がコピーされるのを待つ。その後、移動元と移動先双方で厳密なディスク同期が提供される同期モード（Protocol C）に切り替え、最後に仮想マシンのメモリのマイグレーションを開始する。DRDB の非同期モードでは、仮想マシンが書き込みリクエストを発行すると、移動元のローカルストレージにデータを書き込み、データ同期に用いられる TCP コネクションのソケットバッファにもデータを書き込んだ時点で、仮想マシンに対してリクエストの完了を通知する。一方、厳密な同期モード（Protocol C）では、移動先のローカルストレージにディスクを書き込んだことが確認できたあとに、仮想マシンに対してリクエストの完了を通知する。CloudNet では最初に非同期モードを用いることで、ストレージマイグレーション実行時の性能低下を緩和している。しかし、DRDB の非同期モードにおいても、ディスク同期に用いられるネットワーク帯域が小さければ、仮想マシンの書き込み速度が抑制されてしまう。提案機構においては、常に普段からディスクの同期を行う前提であることから、仮想マシンの性能が定常状態において低下しないことを念頭に置いている。時として十分なディスク同期用の帯域が確保できないことがあっても、またディスク同期用のネットワーク接続が途切れたとしても、仮想マシンの性能は影響を受けない。

## 8. まとめ

本稿では、広域ライブマイグレーション時間を短縮するため、仮想マシンの実行状態をベストエフォートで事前に同期する手法を提案した。平時から仮想ディスクの状態を待避先拠点とできる限り同期しておき、ライブマイグレーションを実行する際にはその差分のみを転送する。マイグレーションに伴うデータ転送量およびマイグレーション時間を大幅に削減できる。同期データに対して世代管理を行うことで、可用帯域が許す範囲での部分的な同期を可能にし、一時的な通信の断絶に対しても再び途中から同期を再開可能にしている。提案機構のプロトタイプを実装し、日米間のネットワークを用いて基礎的な評価実験を行った。仮想マシンの実行状態全体を約 30 秒で再配置できた。提案機構を用いない場合（約 1 時間）よりも大幅に待避時間を短縮できた。今後は実装を進め、様々な環境で評価実験を行う。

**謝辞** 本研究は、科研費（23700048）および CREST（情報システムの超低消費電力化を目指した技術革新と統合化技術）の支援を受けた。また NSF/JST RAPID プログラム（IT Virtualization for Disaster Mitigation and Recovery / 大規模災害における IT インフラ復旧技術に関する調査・研究）の支援を受けた。RAPID プログラムにおける米国側 Co-PI である Jose Fortes 教授および Renato Figueiredo 教授および日本側研究者高野了成氏に感謝する。

### 参考文献

[1] Mauricio Tsugawa, Renato Figueiredo, Jose Fortes, Takahiro Hirofuchi, Hidemoto Nakada, and Ryousei Takano. On the use of virtualization technologies to support uninterrupted IT services. In *ICC2012 Workshop on Re-think ICT infrastructure designs and operations (Accepted)*, pp. 1–5. IEEE Computer Society, Jun 2012.

[2] P. T. Breuer, A. Marin Lopez, and Arturo Garcia Ares. The network block device, 1999.

[3] 広渕崇宏, 小川宏高, 中田秀基, 伊藤智, 関口智嗣. 仮想計算機遠隔ライブマイグレーションのための透過的なストレージ再配置機構. 情報処理学会論文誌:コンピューティングシステム, Vol. ACS26, pp. 152–165, Jul 2009.

[4] Takahiro Hirofuchi. xNBD. `http://bitbucket.org/hirofuchi/xnbd/`.

[5] 穐山空道, 広渕崇宏, 高野了成, 本位田真一. 都鳥：メモリ再利用による連続するライブマイグレーションの最適

化. 情報処理学会論文誌:コンピューティングシステム, Vol. ACS37, pp. 74–85, Mar 2012.

[6] Timothy Wood, Prashant Shenoy, K. K.Ramakrishnan, and Jacobus van der Merwe. CloudNet: dynamic pooling of cloud resources by live wan migration of virtual machines. In *Proceedings of the 7th ACM SIGPLAN/SIGOPS international conference on Virtual execution environments*, pp. 121–132. ACM Press, Mar 2011.