社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2009－ARC－182 (13)
2009－HPC－119 (13)
2009／2／27

仮想計算機遠隔マイグレーションに対応するストレージ提供手法の比較検討

広渕崇宏† 中田秀基† 小川宏高†
伊藤智† 関口智嗣†

我々は，計算機センタやデータセンタにおける資源運用効率を飛躍的に向上させるため，計算資源利用を拠点横断的に最適化することを目指している．各拠点の物理資源が常に最も効率的な稼動状態となるように，仮想計算機のライブマイグレーション機能を利用して，稼動サービスを動的に拠点間で再配置する．しかし，仮想計算機の拠点間再配置においてはストレージの取り扱いが大きな課題であり，さまざまなストレージ提供技術を適切に組み合わせる必要がある．そこで本稿では，仮想計算機の動的再配置に対応したストレージ技術を比較し，それぞれの特徴を概観する．仮想計算機移動に対応するストレージ提供手法には，大別するとストレージ共有，ミラーリングおよびストレージ再配置が存在する．予備実験において，ミラーリングおよび先行型のストレージ再配置手法が仮想計算機のライブマイグレーションと連係動作することを確認した．またその基本的なI/O性能を計測した．ミラーリングおよび先行型のストレージ再配置とも仮想マシンモニタと正しく連係動作することが確認できた．書き込み処理については，ミラーリングにおいては性能低下が見られるものの，先行型のストレージ再配置においては限定的な低下にとどまった．読み込み処理については，両者とも通常のディスクアクセスと同等のI/O性能であった．

Comparison of Virtual Machine Storage Systems Supporting Wide-Area Service Migration

Takahiro Hirofuchi ,† Hidemoto Nakada ,†
Hirotaka Ogawa ,† Satoshi Itoh † and Satoshi Sekiguchi †

Wide-area live migration of virtual machines is a key technology for advanced datacenter virtualization with dynamic multi-site load-balancing; where running virtual machines are transparently migrated among datacenters, for optimizing energy efficiency and running costs. In this paper, we discuss the pros and cons of storage systems supporting wide-area live migration of virtual machines (i.e., remote storage sharing, mirroring, and pre/post-relocation). In pilot experiments, we confirmed that remote storage mirroring and pre-relocation mechanisms correctly worked with a virtual machine monitor for live migration over WAN. In addition, micro benchmarks were conducted to clarify I/O performance of these mechanisms. For write operations, the storage mirroring involved performance degradation due to remote access. For read operations, both I/O throughputs were close to that of direct disk access.

1. はじめに

計算機センタやデータセンタにおいては，資源の運用効率を向上し消費電力を低減することが強く求められている．そこで我々はこれまで，仮想化技術に基づいて複数拠点のデータセンタが連携しながら資源運用できる技術基盤の確立を目指して研究を進めてきた[12),13)]．仮想計算機 (VM) のライブマイグレーション技術によって，オペレーティングシステムやアプリケーションを起動したまま，遠隔拠点間でサービスを透過的に再配置可能にする．そしてデータセンタ間で負荷をバランスすることで，全体として最適な物理資源の稼動状態を維持することを目指している．

我々は先行研究において VM ストレージの透過的な再配置手法を提案した[10),11)]．VM の動的な再配置と連動して，VM が利用している仮想ディスクも遠隔拠点に対して透過的に移動する．ストレージも含めた VM の実行環境全体を迅速に再配置できる．ブロックレベルのストレージ I/O プロトコルのプロキシとして実装され，データセンタにおける SAN 機構への親和性が高い．また単一の実装でさまざまな仮想マシンモニタに対応できる．

しかし，この先行研究における提案手法は非常に強

† 産業技術総合研究所 / National Institute of Advanced Industrial Science and Technology (AIST)

力であるものの，実際にはその特徴をふまえて適切に用いられるべきである．VM の遠隔再配置に対応した仮想化資源運用システムでは，提案手法以外の仕組みも含めたさまざまなストレージ提供手法の利点をふまえたうえで，それらを適切に組み合わせる必要がある．

そこで本稿では，遠隔ライブマイグレーションに対応したストレージ提供手法を比較しその特徴をまとめ，今後我々の仮想クラスタ管理システムに組み込む際の指針とする．拠点間でのストレージ共有やミラーリング，および VM 移動と連動したストレージ再配置手法をそれぞれ概観する．そして予備実験として，ミラーリングと先行型のストレージ再配置手法が仮想マシンモニタ (VMM) と連動することを確認し，その基本的な I/O 性能を計測した．

2 節で各ストレージ提供手法を比較し，3 節で予備実験を行う．4 節で補足的な議論を述べ，5 節でまとめる．

2. ストレージ提供手法

VM を遠隔拠点間でライブマイグレーションさせる際には，移動元・移動先双方の拠点において VM がディスク I/O を継続できることが必要になる．その実現手法を大別すると，以下の 3 種類に分類される．

2.1 ストレージ共有

移動元・移動先双方の拠点が VM 用のストレージをネットワーク越しに共有する．例えば，NFS[9] 上に仮想ディスクイメージを作成したり，iSCSI[8] で移動元・移動先双方のマルチイニシエータ構成とする．LAN における仮想ディスクの運用手法をそのまま WAN においても適用できる．

しかし，WAN のネットワーク遅延や限られた可用帯域によって，遠隔アクセス時の I/O 性能低下が予想される．特にランダムアクセスにおいて性能低下が顕著になることが多い．また，遠隔アクセス中は常にサーバへのネットワーク到達性が維持されなければならない．稼働中の VM の信頼性は，その VM が存在する拠点のみにかかわらず，サーバが存在する拠点や途中の WAN の信頼性にも依存してしまう．

VM の再配置に対してストレージ共有が必要になる場合としては，VM 上のワークロードが極めて大きなデータセットにアクセスする時があげられる．WAN を経由してデータセットの移動を短時間で行うことは困難であるゆえに，後述するストレージ再配置は不向きである．また，データセットを複数の VM からアクセスする場合もストレージ共有が必要になる．

ストレージ共有を用いる際には，性能低下を抑制するために適切なキャッシュ機構が必要となる．一般的に，高遅延ネットワークを介したアクセスであっても，シーケンシャル I/O については適切な先読みやバッファリングによって性能低下が緩和できる．また，拠点間で VM をスケジューリングする際には，遠隔アクセスが極力発生しないような負荷バランスアルゴリズムが望ましい．

2.2 ミラーリング

遠隔拠点間であらかじめ仮想ディスクを提供するストレージをミラーリングしておく．移動元における VM の書き込みが即座に遠隔拠点に反映されるようにし，移動元・移動先双方のストレージを常に同期しておく．VM が遠隔拠点に移動する際には，移動先のローカルストレージにアクセスすればよい．VM が移動先で起動した直後から，すべてのデータは移動先拠点内に存在しており，データの読み込み速度は一切低下しない．

書き込みリクエストを処理する際に，どのタイミングで遠隔ストレージに書き込みデータを反映させるのかという点については複数の戦略が存在する．最も厳密な同期に対しては，拠点内のストレージおよび遠隔ストレージ双方にデータの書き込みが確認できた時点で，リクエストの完了を通知する．しかし，すべての書き込みリクエストの処理に遠隔ストレージへのアクセスがともなうため，書き込み性能は大きく低下する．そこで一般的には同期戦略を緩和して，遠隔ストレージへのデータ書き込みは非同期に行うことが多い．同期戦略と I/O 性能については予備実験において確認する．

VM 遠隔ライブマイグレーションのためだけにミラーリングを選択することは，コストや性能低下の観点から妥当ではない可能性がある．しかし，あらかじめ高度な可用性を想定してサービスが構築されている場合はこの限りではない．ディザスタリカバリを想定したディスクの遠隔ミラーリング機構を，拠点横断的な負荷バランスのための VM 動的再配置にも使うことができる．

2.3 ストレージ再配置

VM の移動にともなって，遠隔拠点間で仮想ディスクも透過的に再配置する手法である．最終的には移動元拠点のディスクブロックをすべて移動先に再配置し，移動先拠点の資源のみで運用可能になる．仮想ディスクを含めた VM 実行環境全体の再配置完了後には，移動元拠点での動作時と全く同じ動作形態になり性能低下は存在しない．

ディスク再配置中にも VM の I/O を継続するために，VM の I/O リクエストを処理しながら，同時にバックグラウンドで拠点間のブロック再配置も行う．ディスクの再配置を VM 起動ホストの切り替え前に完了する場合，および切り替え後から開始する場合が存在する．本稿では前者を先行再配置型，後者を遅延再配置型と呼ぶ．

2.3.1 先行再配置型

ライブマイグレーションの際に，VM を移動先拠点で起動する前に仮想ディスクの再配置を完了する[4),7)]．VM が移動元拠点で動作している時点から，仮想ディスクの遠隔コピーを開始する．動作中の VM によって更新されたブロックも継続的に移動先に対してコピー

する．また同時に VM のメモリページのコピーも継続的に行う．ほぼすべてのメモリページおよびディスクブロックが移動先に同期できた段階で，移動元での VM 実行を停止し残りのメモリページおよびディスクブロックをコピーする．そして今度は移動先で VM の実行を再開する．

この手法の利点としては，VM が移動先拠点で開始される時点ですべてのディスクブロックが移動先に存在しており，ローカルアクセスとなる．この時点以降，ディスク I/O 性能の低下が存在せず，移動元拠点へのネットワーク到達性も必要ない．また，VM 実行環境の再配置作業中にネットワーク到達性が失われたとしても，再配置作業が中断するのみである．VM は引き続き移動元拠点で動作し続けることができる．

しかし，その一方で，動作中の VM が読み書きする仮想ディスクを再配置対象とするために，移動元で VM を停止するまでは更新ブロックを継続的に移動先へコピーし続けれければならない．VM が大量のデータをディスクに書き込み続ける場合には，繰り返しコピーしなければならないブロック数も増大してしまう．このとき有限時間内でディスク再配置を完了するために，意図的にディスクの I/O 速度を低下させる必要がある．ディスク再配置にともなう WAN 経由でのデータ転送量は，仮想ディスクのサイズに加えて，移動中に更新されたブロックを繰り返しコピーする分だけ大きくなる．また，仮想ディスクを VM よりも先に再配置するので，仮想ディスクのサイズが大きければ，すぐに VM 起動ホストを切り替えることができない．

文献 7) における実装は残念ながら入手できないものの，ネットワーク越しのディスクミラーリングプログラムを利用して近い動作を再現できる．予備実験においてその動作を確認している．

2.3.2 遅延再配置型

我々が先行研究[10),11)] において提案した手法であり，VM が移動先拠点で起動した後から仮想ディスクの再配置を開始する．VM の I/O にともなうオンデマンドなキャッシュ機構により徐々にブロックを再配置するとともに，同時並行的に動作するバックグラウンドでのコピーによって残りの部分を再配置する．

この手法の利点としては，時間がかかるディスクの再配置を後回しにすることで，VM 起動ホストの切り替えを比較的迅速にできる点である．また先行更新型のように，更新ページを繰り返しコピーする必要もないため，WAN を経由する通信量を少なくできる．

一方で，VM 起動ホストの切り替え直後には，再配置済みのブロックがほとんどないため，読み込み処理が遅くなる可能性がある．リクエストの対象となるブロックを移動元拠点から取得する必要がある．ただし，書き込みリクエストやすでに再配置済みのブロックの読み込みについては，移動先拠点のディスクアクセスのみで閉じており性能低下はない．

ディスクの再配置中に移動元・移動先拠点間のネットワーク到達性が万が一失われると，VM のディスクアクセスが停止し失敗する恐れがある．その際にはディスク I/O を再開できるまで VM を一時停止したり，VM 実行環境の移動を取りやめてあらかじめスナップショットを取っておいた時点までロールバックする等の対策が考えられる．また実際には，統合的に運用されるデータセンタ間においては，その拠点間ネットワークがあらかじめ冗長化されており高い信頼性を有する場合もある．SLA において可用性が明確にされているならば，遅延再配置型に対して過度の障害対策は不要という考えもできる．

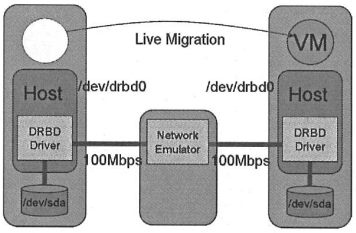

図 1 実験環境

なお，先行再配置型と遅延再配置型を組み合わせることも可能である．遅延再配置型において，VM 起動ホストが切り替わる前から投機的にバックグラウンドコピーを開始することも可能である．

2.4 補　　足

以上に加えて，雛形となる仮想ディスクイメージをあらかじめ準備しておき，VM によって更新された領域のみをストレージ再配置の対象とできる[7)]．また VM 内のシステムにおいてパーティションごとにストレージの提供方法を変更できる．システムやアプリケーションをインストールしたパーティションは雛形の仮想ディスクイメージを用い，スクラッチ領域のみを再配置の対象とできる．

3. 予 備 実 験

上述のストレージ提供手法が VMM にライブマイグレーションと連動して WAN 環境で動作することを確認した．遅延型のストレージ再配置は我々が先行研究で実証済みであるので今回は取り上げない．また一般的なストレージ共有による VM 再配置も省略する．ミラーリングと先行再配置型のストレージについて焦点を当てる．

実験環境を**図 1** に示す☆．日本国内の離れた拠点に見立てた 2 つの計算機を用意し，両者を 100Mbps の Ethernet でつなぐ．ネットワークエミュレータ (Linux カーネルの netemu 機能) で両者間のネットワーク遅延を RTT20ms と設定した．両計算機の間で VM および仮想ディスクのライブマイグレーションを実行する．

ディスクミラーリングおよび先行型のストレージ再

☆ VM ホスト計算機: Intel Xeon 2.8GHz, メモリ 2GB

配置のために，今回はDRBD(Distributed Replicated Block Device)[5),6)]を用いる．DRBDは本来LAN環境においてネットワークを介してRAID-1型のディスクミラーリングを行うプログラムである．異なるホストにつながったディスク間で更新内容を同期し，一方のホストやディスクに問題が生じた際でも他方のホストとディスクを用いてサービスを継続できる．基本的には，どちらか一方のディスクをプライマリとして設定して，そのディスクのみホストからの更新が許される☆．

VMMとしてXen-3.03[1)]を用いてparavirtualizationモードでVM(メモリ512MB, CentOS5)を起動する．各ホストOSに，SCSIディスクをバックエンドとしてDRBDデバイス(4GB)を作成する．2つの計算機間でDRBDデバイスをミラーリングするとともに，VMのバックエンドブロックデバイスとしてDRBDデバイスファイル`/dev/drbd0`を利用する．

3.1 DRBDとVMMの連係動作

VMMとDRDBがどのように連携して動作するのかを以下に説明する☆☆．まず，仮想マシンをライブマイグレーションする際VMMは次のように動作する[3)]．

(1) 移動元ホストのVMMは，移動先ホストに対してVMの実行に必要なメモリ領域を予約する．
(2) 予約したメモリ領域に対して移動元ホストで実行しているVMメモリイメージのコピーを開始する．
(3) メモリイメージのコピー中にもVM内のOSは動作しており，この間更新されたメモリ領域もさらに移動先ホストへコピーすることを繰り返す．
(4) 移動元ホストでVMを停止し，CPU状態や残りのメモリイメージを移動先ホストへ転送する．
(5) すべての実行状態が転送できたら，移動先ホストでVMの実行を再開する．

このとき第3段階までは，移動元ホストを介して周辺機器I/Oが行われ，第5段階以降は移動先ホストを介して周辺機器I/Oが行われる．

3.1.1 ミラーリング

DRBDによるミラーリングが有効であるときには，プライマリと設定されたディスクの更新は即座に他方のディスク(セカンダリ)へ反映される．データの同期が取れていれば，両者のディスクは同一内容である．

そこで，VM動的移動をDRBDによるミラーリングとともに実現するには，VM起動ホストの切り替えタイミングに合わせて，ディスク間のデータの完全な同期を確認し，プライマリ/セカンダリの設定を切り替える．VMが移動元ホストで動作しているときには，移動元ホストのディスクをプライマリ，移動先ホストのディスクをセカンダリとして設定する．そして，ライブマイグレーションの際に以下のように振舞う．

(1) VMMによるメモリイメージのコピーを開始し，メモリイメージのコピーが完了するのを待つ．
(2) 移動元ホストでVMが停止する．移動元ディスクをセカンダリにする．
(3) 直前まで動いていたVMによって移動元ディスクに書き込まれたデータが，移動先ディスクに反映されるのを待つ．
(4) ディスク内容の同期が取れたら，移動先ディスクをプライマリに設定する．
(5) 移動先ホストでVMを再起動する．

3.1.2 先行再配置

DRBDにおいては，プライマリとセカンダリのディスク間でTCPコネクションが失われても，プライマリのディスクの読み書きは可能である．その間にプライマリディスクに書き込まれたデータは，再接続後にセカンダリディスクにコピーされる．

そこで，VM動的移動をDRBDによる先行型ストレージ再配置とともに実現するには，あらかじめプライマリのディスクのみでDRBDを動かし，ストレージ再配置を開始する段階でセカンダリへの接続を確立する．VMMによるメモリイメージの再配置よりも，先にストレージ再配置を開始する．その動作は以下のようになる．

(1) セカンダリである移動先ホストのディスクへDRBDの接続を確立する．移動元ディスクにたまっていた書き込みデータが，移動先ディスクへコピーされはじめる．
(2) 移動元のディスクにたまっていた更新内容が，移動先に反映されるのを待つ．
(3) VMMによるメモリイメージのコピーを開始し，メモリイメージのコピーが完了するのを待つ．
(4) 移動元ホストでVMが停止する．移動元ディスクをセカンダリにする．
(5) 直前まで動いていたVMによって移動元ディスクに書き込まれたデータが，移動先ディスクに反映されるのを待つ．
(6) ディスク内容の同期が取れたら，移動先ディスクをプライマリに設定する．プライマリとセカンダリ間のDRBD接続も終了する．
(7) 移動先ホストでVMを再起動する．

たまっていたデータをセカンダリにコピーしている間(1および2段階目)であっても，動作中のVMはDRBDデバイスを通してI/Oを継続できる．VMによるオンデマンドなディスクI/Oと並行して，バックグラウンドでたまっていたデータの反映も行われる．

☆ 両方のディスクをプライマリとして設定し，2つのホストから同時に更新できる動作モードも存在する．しかし，プライマリを単一ディスクに限る場合に比べて性能は低下する．

☆☆ DRBDの配布物には，DRBDデバイスをXenのバックエンドブロックデバイスとて動作させる際の設定を補助するスクリプト`block-drbd`が含まれる．ディスクミラーリング時のライブマイグレーションが可能である．我々は先行型のストレージ再配置にも対応するようにそのスクリプトを改変し，WAN環境に適用している．

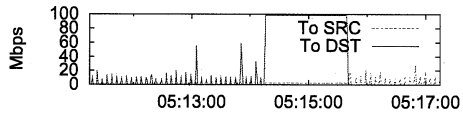

図 2 ライブマイグレーション前後の WAN 通信量 (ミラーリング)

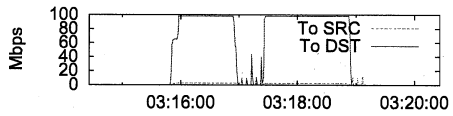

図 3 ライブマイグレーション前後の WAN 通信量 (先行型ストレージ再配置)

3.2 マイグレーション動作の確認

実験環境において VM 上で Linux カーネルコンパイルを実行しながらライブマイグレーションを行った. ミラーリングおよび先行型ストレージ再配置それぞれの場合について, ライブマイグレーション前後の WAN 通信量を**図 2** および**図 3** に示す. 各グラフ描画開始後約 3 分の時刻において, VM メモリイメージの再配置を開始している. ミラーリングにおいては, 書き込み内容をディスク間で同期するため, 常に数秒ごとに WAN 経由のデータ転送が発生している. 先行型再配置においては, VM メモリイメージの再配置よりも前に, 移動元ディスクにたまっていた書き込みデータが転送されている. メモリおよびディスクを再配置する時間以外には, WAN 経由の通信は発生していない.

3.3 DRBD によるディスク I/O 性能

基本的な I/O 性能を確認するため, DBBD を用いて遠隔拠点間でストレージをミラーリングし I/O ベンチマークを実行した. ミラーリングおよび先行型のストレージ再配置における I/O 性能の目安とする. `/dev/drbd0` を移動元ホスト計算機で ext3 でマウントし, ベンチマークプログラム Bonnie++1.0.3c[2] を用いて計測する. この際, ディスク同期を有効にしている場合 (DRBD A, B, C) に加えて, 一時的に停止している場合 (DRBD S) も計測した. DRBD S の結果は, 3.1.2 節の先行型再配置において, ディスク再配置開始前および完了後に相当する. また, DRBD を利用しないで, そのバックエンドとして利用する SCSI ディスクの I/O 性能を Direct として示す.

書き込みリクエストを処理する際に, プライマリのディスクから他方へのディスクへのデータ同期には複数の手法が提供されている. 最も厳密な同期モードでは, 書き込みリクエストについて, 両方のディスクにデータが書き込まれた後に OS に対して完了通知を返す方法である (DRBD 用語でプロトコル **C**). また, プライマリのディスクに書き込まれた時点で (他方のディスクへのデータ同期処理段階に関係なく) 完了通知を返す方法がある (プロトコル **A**). さらに, プロトコル C と A の間の動作として, プライマリのディスクへの書き込みが完了し, かつ他方のホストから書き込みデータを受け取ったという通知があった時点で (他方のディスクへのデータ書き込みの完了に関係なく)OS に対して完了通知を返す方法がある (プロトコル **B**).

表 1 シーケンシャル I/O 性能 (KBytes/s)

	Read	Write
Direct	53468 (3%)	46143 (18%)
DRBD S	46533 (1%)	37897 (16%)
DRBD A	46763 (1%)	11285 (5%)
DRBD B	46278 (1%)	11099 (8%)
DRBD C	46412 (1%)	11213 (8%)

括弧内はゲスト OS における CPU 使用率 (%)

表 2 ランダムシーク性能 (ops/s)

	Random
Direct	184.8 (0%)
DRBD S	178.4 (0%)
DRBD A	178.5 (0%)
DRBD B	178.0 (0%)
DRBD C	174.8 (0%)

括弧内はゲスト OS における CPU 使用率 (%)

実験環境においてミラーリング時の I/O 性能を計測した. 計測したシーケンシャル I/O 性能を**表 1** に示す☆. 書き込み性能は, 書き込みデータを遠隔ディスクに反映させる処理のために, WAN 帯域 12MB/s とほぼ同等の値となっている. ディスク同期を停止している場合には, 遠隔ディスクへの書き込みが発生しないため, 大きな値をとる. 読み込み性能は, WAN を経由しないでローカルのディスクから読み込むだけであることから, WAN の帯域を大きく上回る値となる. いずれの同期モードでも読み書きとも性能に大きな差はない.

表 2 には, ランダムシーク性能を示す. Bonnie++ におけるランダムシーク性能の計測は, 対象となるファイルをランダムに `lseek()` したのち 90%の割合で `read()` および 10%の割合で `write()` を発行する. 対象となるファイルは OS のメモリサイズに対して 2 倍の大きさ (1GB) で作成されるために, カーネルキャッシュのみならずディスクへのアクセスも頻発する. しかし, 各計測結果には大きな差はみられなかった. ディスクのシーク処理が計測結果に大きく影響しており, RTT20ms 程度あればネットワーク遅延の影響は現れてこない.

表 3 に, メールサーバプログラムを想定したファイル操作性能を示す. Bonnie++のファイル操作ベンチマーク機能を用いている. ひとつあたり 10KB の 102400 個のファイルを 100 個のディレクトリの下に均等になるように作成 (`creat()`, `write()`, ファイルとディレクトリの `fsync()`, `close()`) する☆☆. すべ

☆ `write()` ごとに `fsync()` する設定である.

☆☆ 作成されるファイルの総量 (10KB*102400 個) はゲスト OS に割り当てたメモリサイズ (512MB) の約 2 倍となるよう設定している. そのため, 約半分のファイルはカーネルメモリ内に

表 3 ファイル操作性能 (ops/s)

	Create	Access	Delete
Direct	761 (2%)	42654 (99%)	272 (1%)
DRBD S	694 (3%)	42622 (100%)	332 (2%)
DRBD A	274 (4%)	40330 (98%)	272 (4%)
DRBD B	11 (0%)	42715 (99%)	21 (0%)
DRBD C	11 (0%)	42855 (100%)	20 (0%)

括弧内はゲスト OS における CPU 使用率 (%)

てのファイルをランダムな順番にアクセス (`stat()`, `open()`, `read()`, `close()`) する. その後すべてのファイルをランダムな順番で削除 (`unlink()`, ディレクトリの `fsync()`) する. 最も緩やかな同期モードである DRBD A に対して, B および C は `fsync()` をともなうファイル作成および削除操作において性能が低下している.

以上の結果から, ディスクの読み込みのみであれば, WAN 経由でのミラーリングであっても通常のローカルディスク利用と同等の性能が出る. しかし, ディスクの書き込みをともなう場合は, WAN の帯域や遅延により大きく性能低下してしまう. VM のライブマイグレーションのためミラーリングを用いる場合には, 更新操作が少ない場合は大きく I/O 性能が低下することはないと予想される. また, ディスク同期を無効にした場合は, DRBD ドライバを経由することによる若干の性能低下が見られるものの, ほぼ通常のローカルディスク利用と同等の読み書き性能となる. DRBD を先行型ストレージ再配置において用いても, 再配置前後において性能面で大きな問題はない. DRBD の同期モードは, シーケンシャル I/O においては影響しないものの, 小さなファイルのランダムなファイル操作においては大きく影響する. 緩やかな同期モード (A) を用いれば若干ファイル操作性能を改善できる.

4. 議　　論

DRBD のミラーリングを VM 起動ホストの切り替え時点から開始すれば, 遅延型のストレージ再配置手法と近い動作になる. DRBD は移動元から移動先へディスクの同期を開始するとともに, 未同期ブロックに対する読み込みリクエストは移動元から取得する. しかし, 我々の提案手法と比較すると, DRBD は本来ミラーリングの機構であるがゆえに, 起動ホストの切り替え後も書き込みデータが移動元ホストのディスクへ反映され続ける点が異なる. 提案手法では, 移動後のデータ書き込みは移動先拠点のディスクのみを対象とし, 常に迅速に完了する. また, 仮想ディスク上のファイルシステムを解析して使用中のブロックのみ再配置したり, アクセスパターンに応じて重要領域から先に移動するという最適化を行っている.

キャッシュされず, 実際にディスクにアクセスして読み込まれる.

5. ま と め

本稿では, 遠隔ライブマイグレーションに対応したストレージ提供手法を比較検討した. 大別すると, ストレージ共有, ミラーリング, 先行および遅延型ストレージ再配置が存在する. 予備実験を通して, ミラーリングおよび先行型ストレージ再配置が VMM と連係動作することを確認し, その基本的な I/O 性能を確認した. 今後は, 本稿で得られた知見をもとに, 我々が開発中の拠点間 VM 負荷バランス機構に各種ストレージ再配置機構を導入する.

謝辞 本研究は科研費 (20700038) 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) の助成を受けたものである.

参 考 文 献

1) Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I. and Warfield, A.: Xen and the art of virtualization, *Proceedings of the nineteenth ACM symposium on Operating systems principles*, ACM Press (2003).
2) Bonnie++: `http://sourceforge.net/projects/bonnie/`.
3) Clark, C., Fraser, K., Hand, S., Hansen, J. G., Jul, E., Limpach, C., Pratt, I. and Warfield, A.: Live migration of virtual machines, *Proceedings of the 2nd conference on Symposium on Networked Systems Design and Implementation*, USENIX Association, pp. 273–286 (2005).
4) Luo, Y., Zhang, B., Wang, X., Wang, Z. and Sun, Y.: Live and incremental whole-system migration of virtual machines using block-bitmap, *Proceedings of Cluster 2008: IEEE International Conference on Cluster Computing*, IEEE Computer Society (2008).
5) Reisner, P.: DRBD - Distributed Replicated Block Device, *Proceedings of the Ninth International Linux System Technology Conference* (2002).
6) Reisner, P.: DRBD v8 - Replicated Storage with Shared Disk Semantics, *Proceedings of the Twelfth International Linux System Technology Conference* (2007).
7) Sapuntzakis, C. P., Chandra, R., Pfaff, B., Chow, J., Lam, M. S. and Rosenblum, M.: Optimizing the migration of virtual computers, *ACM SIGOPS Operating System Review*, Vol. 36, No. SI, pp. 377–390 (2002).
8) Satran, J., Meth, K., Sapuntzakis, C., Chadalapaka, M. and Zeidner, E.: Internet Small Computer Systems Interface (iSCSI), RFC 3720 (2004).
9) Sun Microsystems: NFS: Network file system protocol specification, RFC 1094 (1989).
10) 広渕崇宏, 小川宏高, 中田秀基, 伊藤智, 関口智嗣: 仮想クラスタ遠隔ライブマイグレーションにおけるストレージアクセス最適化機構, 情報処理学会研究報告 (2008-HPC-116), pp. 19–24 (2008).
11) 広渕崇宏, 小川宏高, 中田秀基, 伊藤智, 関口智嗣: 仮想クラスタ遠隔ライブマイグレーションにむけた仮想計算機ストレージの透過的再配置機構の評価, 情報処理学会研究報告 (2008-HPC-117), pp. 7–12 (2008).
12) 中田秀基, 横井威, 江原忠士, 谷村勇輔, 小川宏高, 関口智嗣: 仮想クラスタ管理システムの設計と実装, 情報処理学会論文誌コンピューティングシステム, Vol. ACS19, pp. 13–24 (2007).
13) 広渕崇宏, 中田秀基, 横井威, 江原忠士, 谷村勇輔, 小川宏高, 関口智嗣: 複数サイトにまたがる仮想クラスタの構築手法, 第 6 回先進的計算基盤システムシンポジウム SACSIS 2008, pp. 333–340 (2008).