

# 仮想化ホスティングサービスのための遠隔 VM Migration 手法の検討

杉田 毅博<sup>†</sup> 宇夫 陽次朗<sup>††</sup> 島 慶一<sup>††</sup> 江崎 浩<sup>†</sup>

<sup>†</sup> 東京大学 情報理工学系研究科 <sup>††</sup> 株式会社インターネットイニシアティブ 技術研究所

## Abstract

近年、仮想化技術を用いたホスティングサービスが増加している。本研究ではそのようなホスティングサービスを支える上で有効な Whole-VM Migration として、次の二点を提案する。差分ディスクイメージを利用することで複数台の VM インスタンスの Migration 時の転送量を減らすこと、及び同一インスタンスの別時点での状態のデータを利用して転送量を減らすことの二点である。

## 1 はじめに

近年、仮想化技術を用いたホスティングサービスは広がりを見せており、VM のマイグレーション技術が必要とされてきている。

ある HV から別の HV へと複数の VM を移すことで、HV のシャットダウンが可能になり、電力消費の削減が期待できる。また、予定された停電の際に別の拠点へと VM を移行させることで、停電に関わらずサービスの運用を続けることが可能になる。

だが、複数の VM を完全に移動させるためには大量の通信が必要である。加えて、拠点間の通信は帯域や RTT などの通信コストが高い。このため、拠点間で複数 VM を migrate するには、通信に配慮した migration を行う必要がある。

本稿の目的は、仮想化技術を用いたホスティングサービスにおいて、複数の VM を WAN を通して別の拠点にある HV へと移行させることに適した VM の Whole Migration 方式を提案することである。

仮想化技術を用いたホスティングサービスでは一般に、仮想マシン (VM) の管理者とその VM を制御する Hypervisor (HV) の管理者が異なる。また、サービス利用者は利用開始時にはほぼ同一のディスクイメージを与えられる。このディスクイメージは、各々の利用方法に応じて徐々に変化していくが、ベースとなる部分は共通である。

VM の Storage Migration には様々な評価軸が存在する。

- Downtime: VM の Migration のため、サービスが完全に停止する時間。
- Total Migration Time: Migration が開始してから、VM が移行先で動きだし、移行元への依存性がなくな

る時点までの時間を指す。

本稿では Total Migration Time を短くすることを重視する。ただし、将来的には Whole-VM Live Migration を行うことを目指し、live migration を行った際には Downtime を短縮することも目的とする。

## 2 VM Migration 技術

仮想化技術には、Xen[1] や VMWare[2], KVM[3] など様々な実装が活用されている。Xen, VMWare, KVM では動作している VM インスタンスを他のマシンに移動させる VM Migration 技術 [6], または Live Migration 技術が確立しており、既に商業的に使用できる段階にきている。

VM Migration 技術はサイト内での VM インスタンスの移動を基本としており、さらに共有ストレージを持つことが前提である。近年は遠隔地への VM Migration も研究段階ではあるが実現されつつある [7]。

一方でストレージの Migration は未だ成熟していない。VMWare ESX では Storage VMotion [5] により VM イメージをひとつのストレージから他のストレージに移動させることが可能だが、この技術はサイト間の移動には着目していない。

## 3 提案手法

Whole-VM Migration において、最も大きな容量を持つのはストレージである。このため、Whole-VM Migration の時間を減らす上では、ストレージの転送量を削減すると効率が良い。

VM のストレージは、HV から見てファイルシステムである場合とブロックデバイスである場合とが考えられる。仮想化技術を用いたホスティングでは HV をサービス提供者がコントロールし VM は顧客それぞれが所有することになるが、顧客の機密保持の観点から、HV にはファイルシステムは見えない方が好ましい。本研究ではこの理由からブロックデバイスを採用した。

またホスティング環境では複数の顧客に複数の VM を提供するが、それらのディスクイメージには共通のブロックが多く存在する。この共通部分を利用することにより、複

数 VM を転送する際の転送量を大幅に削減することが可能である。

さらに、単一のディスクイメージを一時的に別 HV へと避難させた場合、元の HV に戻ってくる際に必要な転送量は、本質的には避難中の変化分だけである。この変化分を発見・分離することで、往復 migrate 時の転送量を削減することが期待される。

これらの観点から、本研究では以下の二点を提案する。

1. 差分ディスクを用いた、複数 VM の共通部分の転送量削減
2. ダイジェストを用いた、同一 VM の変化分転送

### 3.1 差分ディスク

複数の VM イメージを提供する際、共通するディスクのイメージを一つだけ持ち、ユーザごとのデータはそれぞれ差分イメージとして保持することで、一つひとつのイメージの占有するディスク容量を減らすことが可能になる。

具体的な手法としては、Copy-on-Write 技術を用い、ある固定されたディスクイメージに対する差分を別のディスクイメージに保存する。複数のディスクイメージが共通ディスクイメージを土台にし、個々の差分のみを保存していくことによって、全体として転送の必要なデータ量を削減できる。Copy-on-Write は、KVM では QCoW2 形式 [4] のディスクイメージで利用出来る。

### 3.2 digest

ある拠点から別の HV へ VM を一時的に避難させ、しばらく運用したのちにまた元の HV に戻るとき、移行元には元々のディスクイメージを残すことが出来る。我々はこの性質を利用し、通信量の削減を実現する。

migrate した VM のデータはキャッシュとして各 HV が保持し、VM が乗っていた HV に戻るときにはその変化分のみを交換する。変化分の見分は、VM インスタンスのイメージのハッシュをブロックごとにとった、ディスクイメージのダイジェストを比較することで行う。この様子を図 1 に示す。ハッシュは sha1(160bit)、ブロックは 4KB を想定しており、ダイジェストの容量はディスクイメージのサイズのおよそ 200 分の 1 となる。手順は以下のとおりである。

1. HV1 から HV2 へ VM が migrate する。HV1 では時刻  $t$  での VM 全体の状態  $S(t)$  がキャッシュされる
2. HV2 上で時間  $dt$  が経過し、VM 全体の状態が  $S(t+dt)$  になる

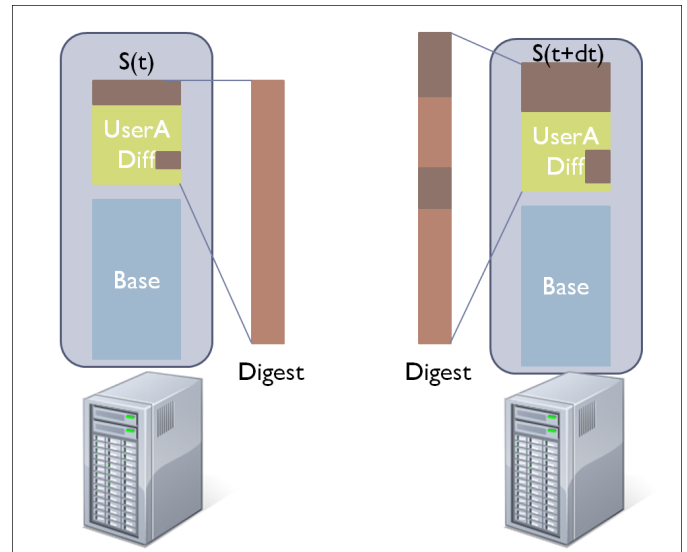


図 1: ディスクイメージの digest

3. HV2 から HV1 へ migrate しようとする。このとき HV2 は  $S(t+dt)$  のディスクイメージの digest を送る
4. HV1 は  $S(t+dt)$  の digest と  $S(t)$  の digest を比較する。HV1 は二つの digest が異なった block を HV2 に要求し、時刻  $S(t+dt)$  の VM 状態を復元する

## 4 今後の展望

以上二つの手法により、仮想マシンホスティング環境下での Whole-VM Migration の通信量削減を目指す。現在は、停止した VM を移動させるための手法を実装している段階である。今後はこの手法の評価を行ったうえで、Whole-VM Live Migration への拡張を行う予定である。

## 参考文献

- [1] Xen, <http://www.cl.cam.ac.uk/research/srg/netos/xen/>
- [2] VMWare, <http://www.vmware.com/>
- [3] KVM, <http://www.linux-kvm.org/>
- [4] qcow2, <http://people.gnome.org/markmc/qcow-image-format.html>
- [5] VMWare, Inc. "VMWare Storage Vmotion: Non-disruptive, live migration of virtual machine storage.", Product Datasheet
- [6] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, A. Warfield, "Live Migration of Virtual Machines", NSDI, 2005
- [7] R. Bradford, E. Kotsovinos, A. Feldmann, Harald Schiöberg, "Live wide-area migration of virtual machines including local persistent state", Proceedings of the 3rd international conference on Virtual execution environments, 2007