

多種の仮想計算機環境における OS イメージの運用管理負荷を軽減するシステムの提案

山崎 雅彦[†], 松永 俊介[‡], 榎田 秀夫[◇].

m-ympz07@dsm.cis.kit.ac.jp, s-mtng07@dsm.cis.kit.ac.jp, h-masuda@kit.ac.jp

[†] 京都工芸繊維大学大学院 工学科学研究科 情報工学専攻

[‡] 京都工芸繊維大学 工学学部 電子情報工学科

[◇] 京都工芸繊維大学 情報科学センター

概要

仮想計算機技術を用いることで、複数の計算機を統合できハードウェア資源を効率よく利用できる。しかし、一般的には仮想化ソフトウェアによって使用できる OS イメージは異なり、かつ、OS インスタンスの数だけ独立した OS イメージを用意し管理する必要がある。そこで、本稿ではこれらの OS イメージを共通化することで運用管理負荷を軽減し、利用環境に応じて容易に仮想化ソフトウェアを変更できる仮想計算機システムを提案する。提案する OS イメージによる仮想計算機システムを用いることで、使用する仮想計算機技術に関わらない仮想計算機の運用管理が可能となり、仮想計算機の台数が増加した場合の OS イメージの運用管理負荷も軽減されることが期待される。

キーワード 仮想計算機, union ファイルシステム

Instant OS image management system for various virtual machine environments

Masahiko Yamazaki¹, Shunsuke Matsunaga², Hideo Masuda³.

{m-ympz07,s-mtng07}@dsm.cis.kit.ac.jp, h-masuda@kit.ac.jp

¹ Graduate School of Information Science, Kyoto Institute of Technology

² Department of Electronics and Information Science, Kyoto Institute of Technology

³ Center for Information Science, Kyoto Institute of Technology

Abstract Using the VM technology, we can reduce hardware and operating cost with unified few computers. However, it is not enough reducing the management cost that the number of independent OS images is still as many as a number of OS instance, VM technology can not help us at the point of this view. In this paper, we present that consideration on the sharing OS image for a number of OS instance and switching virtual machine environments, and propose implementation of sharable OS image using union filesystem for various virtual machine environments. The virtual machine system to use proposal OS image enables management virtual machines regardless of using virtual machine technique, and we expect to reduce the management cost for OS image when the number of VM is increase.

keywords Virtual Machine, union filesystem

1 はじめに

近年、仮想計算機技術を用いて複数の計算機を統合しハードウェア資源を効率よく利用することがよく行われている。仮想計算機技術を用いることにより、物理的に必要な計算機の台数を減らすことができ、運用管理コストを削減することができる。さらに、単一の OS インスタンスで複数のサービスを稼働させる場合に対して、それぞれの環境が分離されるためセキュリティを高めることができる。また、現在、それぞれに性能や機能に特徴を持つ様々な仮想計算機技術が提案され使用可能となってきており、サーバファームにおけるサーバの統合だけではなく、テスト環境の構築や共通デスクトップの利用など様々な場面で仮想計算機技術が使用されている。

しかし、仮想計算機技術を用いて物理的な計算機の台数を削減したとしても、論理的な計算機の台数は削減されず仮想計算機自身の運用管理も必要であるため、運用管理上の負荷が増大しがちである。例えば、一般的には仮想計算機環境上においても、OS インスタンスの数だけ独立した OS イメージを用意する必要がある。また、仮想計算機技術によって使用できる OS イメージは異なるため、使用する仮想計算機技術を変更したい場合、使用する仮想計算機技術にあわせた OS イメージを毎回構築する必要がある。よって、それぞれの OS インスタンスには共通部分が多いにもかかわらず OS イメージの管理は複雑になってしまう。

そこで、本研究では複数の OS インスタンスにおける OS イメージの共通化、および、使用する仮想計算機技術の切り替えに対する OS イメージの共通化について検討を行い、様々な仮想計算機技術で共通使用可能な OS イメージの構成方法について提案する。提案する OS イメージによる仮想計算機システムを用いることで、柔軟なサービスの運用管理が可能となる。また、構築した仮想計算機システムにおいて提案する OS イメージの使用に伴うオーバーヘッドについて評価を行う。

2 システムの要件

様々な仮想計算機環境で共通使用可能な OS イメージを構築するにあたり、現状の仮想計算機環境を用いたシステムにおけるサーバ管理の問題点について述べる。

2.1 OS イメージの管理の容易化

全ての仮想計算機で同じ仮想計算機技術を用いたとしても、一般的には OS インスタンスごとに OS イメージを作成し、運用管理していく必要がある。そのため、セキュリティパッチの適用などは、どの OS イメージにおいてもほぼ同一の作業であるにもかかわらず OS イメージ毎に行う必要があり、運用管理上の負荷が高い。さらに OS インスタンスの数だけ必要となるディスク容量が増加してしまう。

2.2 様々な仮想計算機技術での動作

グリッドコンピューティングやクラウドコンピューティングと呼ばれる環境においては、各計算機上で動作させることのできる仮想計算機技術は様々である。仮想計算機技術において、仮想化の方法の違いから、必要なドライバや設定ファイルが異なるため、OS イメージは仮想計算機技術ごとに異なる。このため、仮想計算機技術を変更すると、OS イメージも構築し直す必要がある。

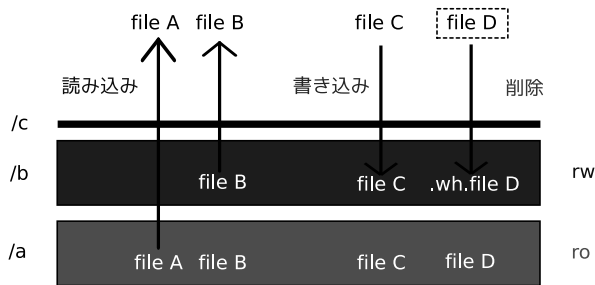
2.3 マイグレーションの実現

サーバファームのような環境においても、計算機の導入時期の違いなどから物理的な各計算機の環境は均一ではなく、それに伴って異なる仮想計算機技術が用いられている場合がある。この場合、各仮想計算機上で稼働している OS インスタンスの柔軟な配置、移動に制限が生じる。

3 検討

3.1 ディスクイメージの単一化

OS イメージの管理の容易化のため、全ての OS インスタンスの基本となる OS イメージを 1 個だけ用意し、各 OS インスタンスはこの基本イメージを元に、差分だけを個別に用意して構成する。この基本イメージを本システムにおける計算機の共通 OS イメージとし、各計算機ごとに用意された個別 OS イメージを union ファイルシステム [1, 2] (図 1) を用いて重ね合わせるによりディスクイメージの単一化 (図 2) を実現する [3]。



mount -n -t unionfs -o dirs=/b=rw:/a=ro none /c

図 1: union ファイルシステムの動作概要

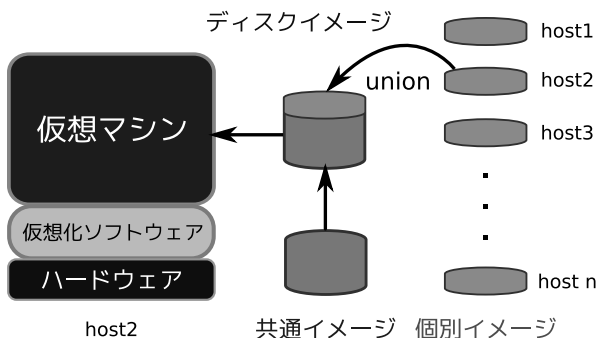


図 2: ディスクイメージの単一化

3.2 union ファイルシステム使用時の問題点の解決

各仮想計算機では、共通ディスクイメージと個別ディスクイメージが union されたものがディスクイメージとして利用される。このとき、共通ディスクイメージをアップデート作業などで変更しても、個別ディスクイメージにコピーアップされた同じ名前のファイルが存在すると、そのコピーアップされたファイルしかアクセスできないためディスク全体としてみると不整合が発生し正常に動作しなくなる可能性がある。

そこで、バージョン管理システムを利用して、共通ディスクイメージの変更と個別ディスクイメージの変更をそれぞれ管理し、共通ディスクイメージに変更があった場合にその変更点を個別ディスクイメージに反映させることでファイルの不整合の解決を図る。

バージョン管理システムには集中型と分散型が存在するが、本稿では分散型のバージョン管理システムを用いる。分散型のバージョン管理システムは従来の集中型のバージョン管理システムとは異なり、新たにリポジトリ用のディレクトリを用意する必要はなく、共通の管理対象を持つ複数のリポジトリを簡単に保持することができる。具体

的には、共通 OS イメージと個別 OS イメージで別々のリポジトリを管理し、さらにリポジトリ間の同期をとることで、共通 OS イメージおよび個別 OS イメージの変更による OS イメージの不整合の発生を防ぐ。

3.3 共通 OS イメージの共有手法

提案する OS イメージの構成では、セキュリティパッチの適用などの共通 OS イメージの変更が各仮想計算機の管理外で起こりうる。この時、カーネルによるディスクキャッシュ機能により、各仮想計算機上で OS イメージの変更が反映されない可能性があるため、これを回避する必要がある。

3.4 複数の仮想計算機技術で共有できるディスクイメージの構成

一般的には、仮想計算機技術により、仮想化の度合いやカーネルが異なるため、必要となるモジュールやデバイスファイルが異なり、そのままでは、ディスクイメージを共有することができない。一方、カーネル以外のアプリケーションの差はほとんどない。そこで、モジュールは必要なモジュールを全て、共有イメージにあらかじめ用意しておき、デバイスファイルは udev[4] を用いて起動時に自動的に適切なデバイスファイル生成するように設定することで、ディスクイメージの共有を可能とする。

3.5 マイグレーション機能の実現

3.1 項で説明した単一 OS イメージを用いることで、複数の仮想計算機でディスクイメージを共有することは可能であるが、異なる仮想計算機間でのライブマイグレーションは、メモリ内の情報の違いなどから難しい。

そこで、アプリケーションレベルのマイグレーションの実現を考える。仮想計算機を立ち上げる際に MAC アドレスを指定することが可能であると想定し、OS インスタンスごとに MAC アドレスを指定し、どの仮想計算機で起動しても同一となるように設定する。これにより、仮想計算機技術を切り替える際に発生するサービスの切断を回避することが可能となり、マイグレーション機能が実現できる。

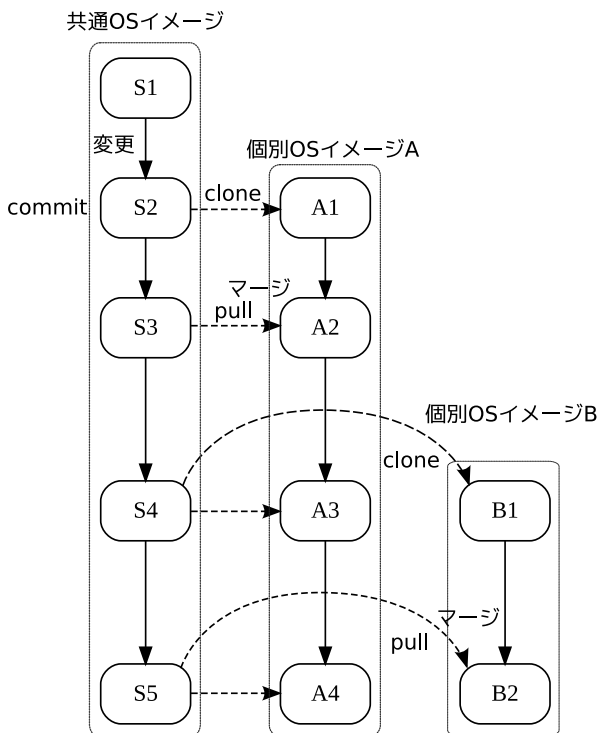


図 3: ファイル不整合の解決

4 仮想計算機システムの構築

本稿では、OS として Debian lenny[5] を用い、Xen[6]、KVM[7]、VMware Server[8]、VirtualBox[9]、UML (User Mode Linux) [10] の各実装において動作確認を行った。

4.1 バージョン管理システムの設定

分散型のバージョン管理システムとして現在広く利用されている Mercurial[11] を用いて実装を行った。まず、共通 OS イメージ側でリポジトリの初期化作業を行い、個別 OS イメージ側でその管理情報のコピーアップ (clone) を作成する。個別 OS イメージにおけるファイルの変更は各仮想計算機から把握するためリポジトリ間の同期は共通 OS イメージにおけるファイルの変更時のみでよい。

共通 OS イメージが更新されたときのリポジトリ間の同期手順を図 3 に示す。

5 ディスクイメージの管理

Linux においては、通常デバイスは “/dev/sda” のようなブロックスペシャルデバイスのファイル名で指定されている。しかし、仮想化ソフトウェ

アによって提供されるブロックスペシャルデバイスは異なるため、使用する仮想計算機技術を変更するとブロックスペシャルデバイスのファイル名を変更する必要がある。そこで、本構成ではあらかじめ共通ディスクイメージには “BASE”，個別ディスクイメージには “U” とラベル付けを行うこととし、ブロックスペシャルデバイスの指定にはラベルを用いることで、仮想化ソフトウェアを変更したときの設定ファイルの変更を回避した。

5.1 ネットワークの設定

各仮想計算機では仮想化ソフトウェアで設定した MAC アドレスを元に udev を用いてネットワークの設定を行っている。何も設定を行わないと MAC アドレスは仮想計算機ソフトウェアごとに異なるため、仮想計算機上からは異なる NIC (Network Interface Card) が接続されたと認識され、udev が新たにデバイスファイルを作成してしまう。

そこで、提案する仮想計算機システムでは各仮想計算機ごとに異なるローカルアドレスを生成し仮想計算機ソフトウェアによらず、同一の MAC アドレスを割り当てた。これにより、どの仮想化ソフトウェアを用いた場合でも、OS イメージの設定変更なしで運用することが可能である。

6 評価

6.1 ディスク容量

提案する構成では使用するすべての仮想化ソフトウェアに必要なファイルを共通 OS イメージに配置しておく必要があるため、共通ディスクイメージは通常のディスクイメージに比べて多くのディスク容量が必要となる。

実装環境 (表 1) において Debian lenny を標準インストールした時のディスク使用容量は約 580MB であった。提案する OS イメージの構成では、仮想化計算機環境を提供するために、Xen で約 62.9MB、UML で約 14.1MB 必要となる。さらに Mercurial で約 5.4MB、aufs の各カーネル用モジュールで約 0.9MB 必要となり、合計で約 83.3MB 増加する。

さらに、共通 OS イメージと個別 OS イメージでのファイル管理のためのバージョン管理システム用のディスク容量が必要となる。約 2.4MB の/etc

表 1: 実装環境

機種	HP ProLiant ML115 G1
CPU	Athlon 3500+
Memory	PC2-6400 4GB
Chipset	nVidia MCP55S Pro
Network	Gigabit Ethernet
HDD	SATA 7200rpm 320GB
OS	Debian GNU/Linux
Linux kernel	2.6.26-1-amd64
Xen	3.2-1-amd64
KVM	72+dfsg
VMware Server	2.2.0 build-122956
VirtualBox	2.1.0

ディレクトリと約 134MB の /lib ディレクトリを管理対象とした場合、初期状態で共通 OS イメージ側で約 64MB、個別 OS イメージ側で約 1.2MB 必要となる。よって、提案する OS イメージの構成では通常のディスクイメージ容量に比べて初期状態で 83.3MB+64MB+1.2MB=148.5MB の増加となる。

6.2 仮想計算機の切り替え時間

現時点の実装では、仮想計算機技術を切り替えるときには一度シャットダウンして再度新しい環境で再起動する必要がある。

実装環境において、各仮想計算機において起動およびシャットダウンにかかる時間は表 2 の通りである。1 台の計算機上で仮想計算機技術の切り替えを行うとすると、例えば Xen から KVM へ切り替える場合、Xen DomainU のシャットダウン、物理マシンのシャットダウンと通常カーネルでの起動、KVM での仮想計算機の起動という手順が必要であり、最低でも 5+14+42+19=80 (秒) かかる。

7 考察

構築する仮想計算機システムについて考慮すべき点を述べる。

7.1 ディスクイメージ単一化の影響

提案する構成では、union ファイルシステムを用いることによるファイルの不整合の問題解決のため

分散型のバージョン管理システムである Mercurial を用いている。共通ディスクイメージを更新した場合には個別ディスクイメージ側との差分管理を Mercurial 上で行う必要がある。

これらの解決策として、union ファイルシステムにバージョン管理機構を組み込むことが考えられる。union ファイルシステムにおいて、仮想計算機環境からのファイル参照時に共通 OS イメージ上で変更があるかどうかについて、自動的に判別する機能を組み込むことで差分管理が自動化できる。ただし、変更があった場合には個別 OS イメージ側のファイルとマージする必要があるが、ファイルによっては自動的にマージすることは難しく、どのように通知を行うかが今後の課題である。

7.2 共通 OS イメージの提供方法

現時点で共有 OS イメージのファイルシステムとして、ext3 および NFS での動作を確認している。ext3 形式を利用した場合、e2label を用い、パーティションごとにラベル付けを行うことで、仮想計算機技術間でのディスクデバイスの認識の違いを隠蔽している。ただし、ext3 形式を共有 OS イメージに用いた場合、3.3 項に述べたようなファイルの不整合が発生することを確認している。NFS ではそのような問題は発生しないが、ネットワーク接続が必須となる問題がある。また提案する構成では共通 OS イメージを複数の計算機から同時に参照できるだけでよく更新できる必要は無いため、NFS のような複数の計算機からの更新制御のための機構は必要ない。

7.3 仮想化ソフトウェアに必要な機能

提案するシステムにおいては仮想計算機技術の実装に Raw ディスクイメージを扱う機能が含まれていることを想定している。仮想計算機技術の中には利用するディスクイメージについて性能や安定性の面から、その仮想計算機技術専用のフォーマットを使用する場合も多い。この場合、変換作業無しに単一のディスクイメージを共有することは困難となる。Raw ディスクイメージであれば、"dd" コマンドで簡単に生成でき、仮想計算機技術を用いない場合でも簡単に扱うことが出来るため、仮想計算機技術の実装においては、Raw ディスクイメージを扱えるようにするべきである。

表 2: 起動およびシャットダウン時間

	Xen	KVM	VMware Server	VirtualBox	UML
native (boot:42)	×	19	18	19	21
(shutdown:12)	×	5	5	5	6
Xen (boot:52)	17	×	×	×	31
(shutdown:14)	5	×	×	×	7

7.4 マイグレーション機能の実現

サービスレベルのマイグレーションにおいては、稼働中の仮想計算機をシャットダウンし、移動先の仮想計算機環境で再起動させる必要がある。構築した仮想計算機システムでは、全ての仮想化ソフトウェアにおいて、使用する OS イメージは共通化されているので、特別な作業をせずに、仮想計算機技術を切り替えて使用することが可能である。しかし、仮想計算機技術により、切り替えにかかる時間が大きく異なる。特に、物理的に再起動が必要な切り替えでは、オーバヘッドがかなり大きいので、頻繁に切り替えることはできない問題がある。

Web サービスなど、ある程度サービスが中断することが許容でき、状態をファイルに保存できるサービスであれば、現時点でもマイグレーションが可能であると考えられる。

8 まとめ

本稿では、様々な仮想計算機環境で共通使用可能な OS イメージの構成方法について検討を行った。現在、union ファイルシステムを用いて統合した OS イメージを作成し、様々な仮想計算機技術を利用することができるシステムを構築している。

今後の課題としては、共通 OS イメージの効率的なアップデート手法、サービスマイグレーション機能の適用範囲の検討とその実装、および構築したシステムの性能評価などが挙げられる。

参考文献

- [1] A Stackable Unification File System, <http://www.am-utils.org/project-unionfs.html>.
- [2] Aufs – Another Unionfs, <http://aufs.sourceforge.net/>.
- [3] 梶田 秀夫, 齊藤 明紀: 「Xen と unionfs を用いたサーバファーム運用の可能性の検討」, 情報処理学会 DSM 研究会, 2006-DSM-41, pp.85-89, May 11-12, 2006.
- [4] udev, <http://www.kernel.org/pub/linux/utils/kernel/hotplug/udev.html>.
- [5] Debian GNU/Linux, <http://www.debian.org/>.
- [6] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt and A. Warfield. “Xen and the Art of Virtualization.” Proceedings of the nineteenth ACM symposium on Operating systems principles, pp.164-177, Bolton Landing, NY, USA, 2003
- [7] KVM, <http://kvm.qumranet.com/>.
- [8] VMware, <http://www.vmware.com/>.
- [9] VirtualBox, <http://www.virtualbox.org/>.
- [10] User-mode Linux Kernel, <http://user-mode-linux.sourceforge.net/>.
- [11] Mercurial, <http://www.selenic.com/mercurial/>.