

## 技術基礎講座

## 【オープンソース仮想化技術】

## 第4回 仮想化技術比較

- 第1回 サーバ仮想化とは
- 第2回 仮想ハードウェア方式
- 第3回 仮想OS方式
- 第4回 仮想化技術比較
- 第5回 OSS仮想化技術の展望

前回、前々回とサーバ仮想化の代表的なオープンソースプロダクトの仕組みに関して述べました。今回はその中で実用レベルもしくはそれに近い製品としてXen, KVM (Kernel-based Virtual Machine), OpenVZに関して基本的な機能を比較するとともに、性能面での特徴を知るうえで参考となる基礎性能データを示します。

## 仮想化技術比較の観点

仮想化を用いる目的や実現したいサービスレベルによって、選定すべきプロダクトは異なってきます。このプロダクト選定の際に重要となる機能面と性能面に着目して、最新(2009年7月16日時点)ディストリビューションを比較します。

## 機能比較

仮想化技術を適用するにあたって最低限考慮すべき基本的な機能を比較します。

## (1) 対応アーキテクチャ

- ① Xen-x86, x86\_64, ia64, ppc\*<sup>1</sup>
- ② KVM-x86, x86\_64
- ③ OpenVZ-x86, x86\_64, ia64

どのプロダクトもPCサーバのアーキテクチャとしてもっとも一般的なx86, x86\_64に対応しています。

また、XenとKVMの完全仮想化を使用する際には、仮想化支援機能(Intel VTまたはAMD SVM)を使用可能なCPUが必須です。

## (2) 動作ゲスト

- ① Xen
  - ・準仮想化 - 準仮想化対応Linux\*<sup>2</sup>
  - ・完全仮想化 - Linux全般, WindowsXP, 2003以降
- ② KVM - Linux全般, Windows2000以降
- ③ OpenVZ - Linux全般

Xenの準仮想化方式はそれに対応したOSのみ動作可能です。Xenの完全仮想化方式やKVMは、WindowsやリガシーなLinuxの動作が可能です。OpenVZは仮想OS方式であるため、同一OSのみゲストとして動作させること

が可能です。

## (3) 故障影響レベル

- ① Xen-OS故障時, 他ゲストへの影響は限定的
- ② KVM-OS故障時, 他ゲストへの影響は限定的
- ③ OpenVZ-ゲスト故障時, 他ゲストへ影響を及ぼす  
ゲスト故障のうちOSに深く関係する故障の場合, OpenVZではOS自体を共有しているため, 他ゲストへの影響は避けられません。

## (4) ライブマイグレーション機能

ライブマイグレーション機能は、ゲストを無停止で他のハードウェア上へ移動させる機能です。近年、仮想化特有の付加機能の一つとして注目されています。サービスを無停止でハードウェアメンテナンスを行う手段や、多数のハードウェア上で分散して稼働しているゲストを少数サーバへの移動・集約により通電するハードウェアを減らして省電力化を図る手段として使用されます。

- ① Xen - 使用可能\*<sup>3</sup>
- ② KVM - 使用可能
- ③ OpenVZ - 使用不可

OpenVZに関しては、カタログ上ではライブマイグレーション機能をサポートとありますが、実際に使用可能な機能はゲストOS側のディスク領域の同期機能(rsyncとほぼ同等)であり、ゲストの実行を継続したまま移動することはできない点がXenやKVMとは異なります。

\*1 RHEL (Red Hat Enterprise Linux) 同梱版Xenはx86, x86\_64, ia64のみ対応。

\*2 RHEL5ではRHEL3.9, RHEL4.5以降, RHEL5.0以降が対応ゲストOS。

\*3 RHEL5.3以前の同梱版Xenでは, x86\_64ホスト上でのx86ゲストOSマイグレーションは正常に動作しません。

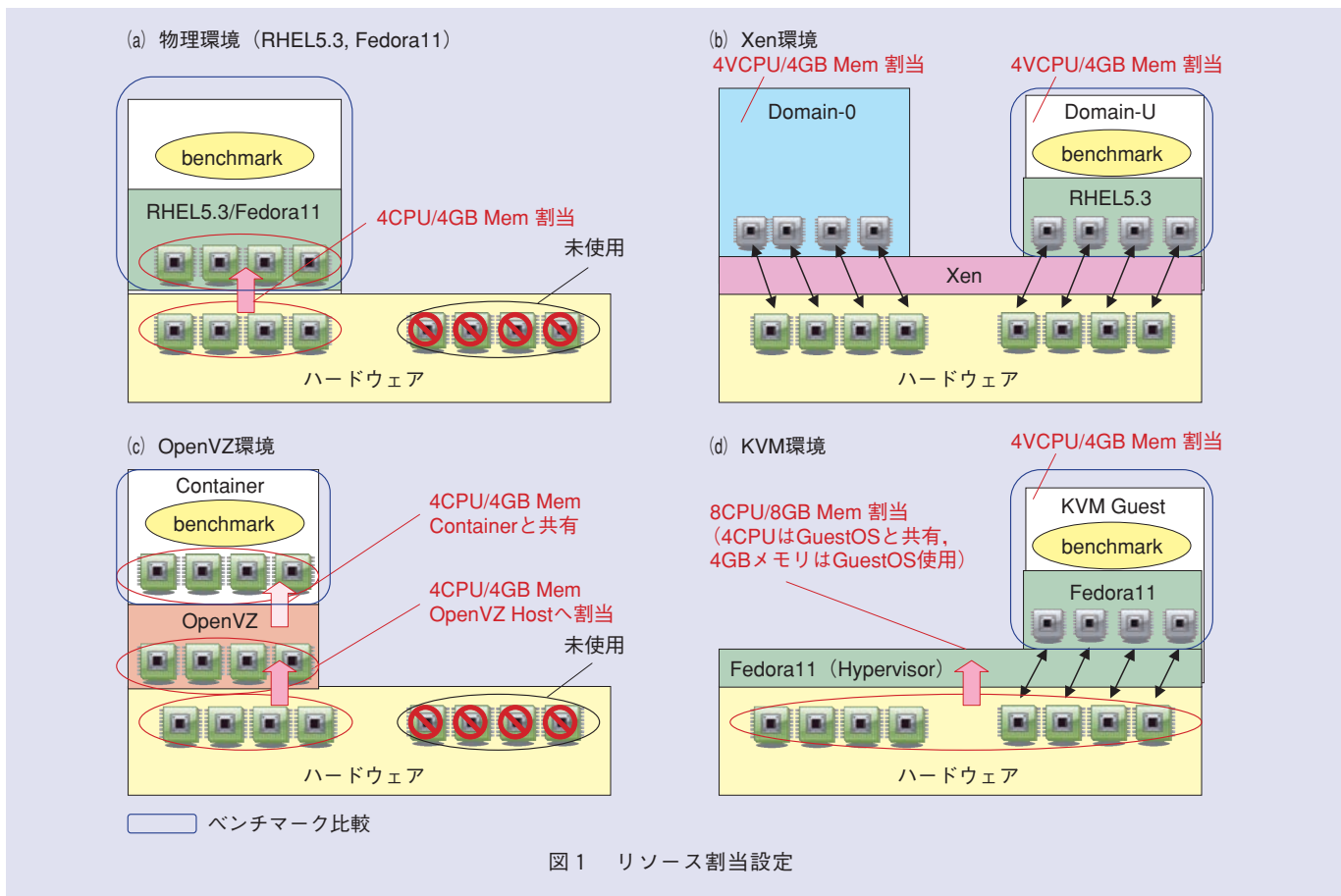


図1 リソース割当設定

## 性能比較

### ■仮想化製品の性能評価時の落とし穴

一般的に公表されている、仮想化製品に対する性能評価結果はさまざまです。同一の仮想化製品の性能評価結果として「物理環境と比較して99%の性能で動作します」のような報告例がある一方で、「仮想化のオーバーヘッドは50%です」といった異なる評価結果も存在します。では、どちらの評価が正しいのでしょうか。

以降では、仮想化製品の性能を評価する際に、これらの結果をどうとらえるべきかを中心に説明していきます。基礎的な性能を把握するために、Imbench<sup>(1)</sup>、\*4を用いて性能を測定します。測定した項目のうち、注目したものは以下です。

- ① CPU性能（整数ビット演算時間）
- ② メモリ性能（ページフォルト処理時間）
- ③ Disk I/O性能（mmap処理時間）\*5
- ④ ネットワーク性能（TCP通信）

### ■検証環境

以下の環境での性能比較を行います。

- ① RHEL5.3物理環境（kernel-2.6.18-128.el5）
- ② Xen（RHEL5.3同梱版+準仮想化方式kernel-2.6.18-128.el5xen）
- ③ OpenVZ（RHEL5.3 + kernel-2.6.18-128.028 stab062.3）
- ④ Fedora11物理環境（kernel-2.6.29.4-167）
- ⑤ KVM（Fedora11同梱版，kernel-2.6.29.4-167）

RHEL5.3カーネルとFedora11カーネルのベースバージョンが大きく異なるため、本来はXen、KVM、OpenVZの性能に関して直接比較はできません。性能比較に関してはRHEL5.3物理環境・Fedora11物理環境との比較を中心に行っていきます。そして、一般的に仮想環境と物理環境間の性能差が発生しやすい項目や、性能差が発生しにくい項目はどのようなものなのかに着目して性能検証結果を見ていきます。

使用したハードウェア「IBM x3550」のスペックは以下です。

\*4 Imbench：OS動作に関連する細かい性能を測定可能なベンチマークソフト。  
\*5 ファイルやデバイスを連続したメモリ空間にマッピングする操作。

- ・ CPU: Xeon X5470 Quad Core × 2
- ・ Mem: DDR2 SDRAM 16GB
- ・ HDD: SAS 146GB × 2 (RAID 0)

また、可能な限りゲストごとの環境をそろえるために、実質的なリソース割当を4 CPU、4 GBメモリに近づけるように図1のようにリソースを割り当てます。

■検証結果

(1) 整数ビット演算時間

CPUの整数ビット演算にかかる時間の測定結果を図2に示します。

物理環境と各仮想化方式では差はわずかns単位です。これは、比較をしているすべての仮想化方式では、CPUの単純な演算であれば、特に仮想化レイヤの介入なしで直接実行可能なためです。そのため、各仮想環境と物理環境で結果に差がでません。

(2) ページフォルト処理時間

メモリアクセス速度を示す1つの要因となる、ページフォルト処理時間の比較を行います。ページフォルトと呼ばれる例外が発生して物理メモリを仮想メモリ上にマップする処理にかかる時間の測定をし、結果を図3に示します。

各仮想化方式に関して、ページフォルト処理時間が物理環境と比較して大きくなっています。Xen, KVMに関しては、Hypervisorが物理メモリをゲストOSに対して割り当てる際に、ゲストOSに対して仮想的な物理メモリ空間を構成してマッピングするため、メモリアクセス時の性能に影響を及ぼします。OpenVZではメモリ操作に関連する部分での各種OSリソースごとのアカウンティング<sup>\*6</sup>処理のオーバーヘッドが大きいことが一因と考えられます。

(3) mmap処理時間

lmbenchでのmmap処理時間を図4に示します。これはファイルをメモリ領域へマッピングする処理の時間であり、Disk I/O性能の指標の1つです。

各仮想化方式のmmap処理時間を比較すると、物理環境にI/Oモデルがもっとも近いOpenVZの性能が優れているといえます。また、Xenの結果に関してはオーバーヘッドはあるものの、準仮想化方式を用いたため、完全仮想化モデルであるKVM環境と比較して良い結果を示しています。

(4) TCP通信

ネットワーク性能を評価する1つの指標である、TCPのレスポンスタイムの測定結果を図5に示します。ローカルホストアドレスに対しての通信に対して測定しています。Xen, KVMともにレスポンスタイムは物理環境と比較して低い性能を示しています。これは、ネットワーク処理に関連する部分に関してハードウェアの仮想化を行う機構から、ネットワーク処理にかかわる処理時間が増加することが一因として考えられます。OpenVZについても、ネットワーク処理部分でのアカウンティング処理に時間が取られることが性能低下の一因であると考えます。

■仮想化オーバーヘッドの評価観点

以上のように、測定項目によっては仮想化をしても性能が物理環境と変化のないCPU演算処理であったり、物理環境と性能が大きく異なってくるメモリアクセスやDisk I/O処理やネットワーク I/O処理であったりと、性

\*6 Containerごとのリソース使用量を数えるための機能。

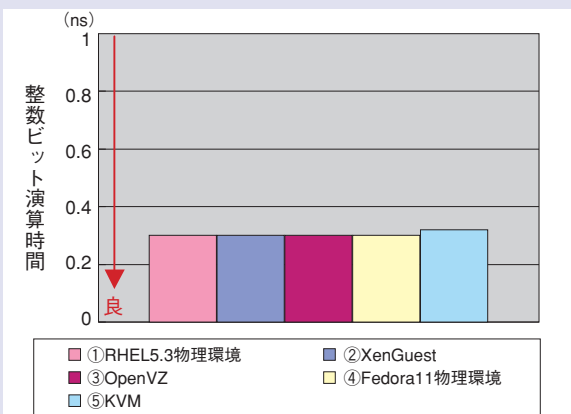


図2 整数ビット演算時間比較

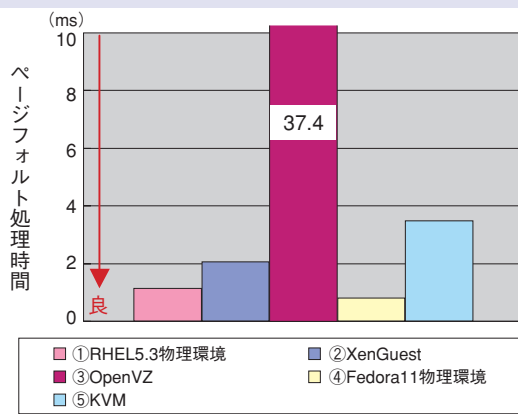


図3 ページフォルト処理時間比較

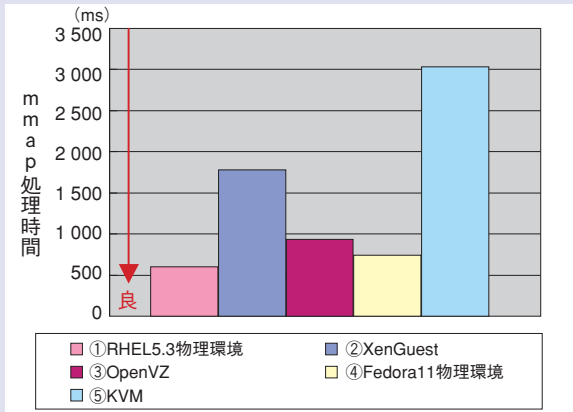


図4 mmap処理時間比較

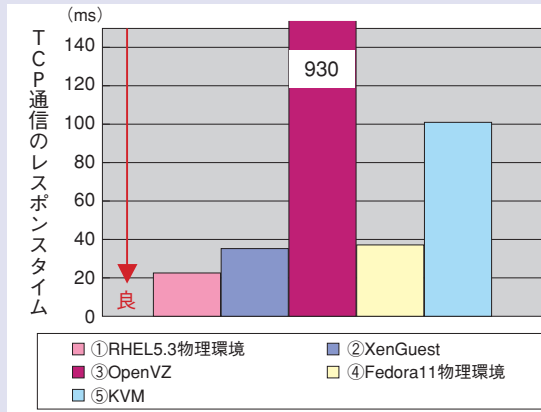


図5 TCP通信のレスポンスタイム比較

能評価結果はさまざまです。

アプリケーションはそれらの細かいOS処理の積み重ねで動作するため、アプリケーションに占めるOS処理の割合に依存したオーバーヘッドが現れます。そのため、下記の2点を把握することが重要になります。

- ① ゲスト上で動作させるアプリケーションが、どのようなOS処理（CPU演算、メモリ操作、Disk I/O、ネットワーク I/O）に直結する処理を行っているか。
- ② ①で洗い出した処理のうち、それらの処理がアプリケーションの処理全体に占める割合がどの程度か。

上記を把握することに加えて、アプリケーションのOS処理に相当する操作が、ImbenchのようなOSの細かい動作に関するどのベンチマーク項目に当てはまるかをマッピングすることによって、アプリケーション動作時の仮想化オーバーヘッドが大きいかまたは小さいかの大きな傾向を推測することができます。また、このマッピングは、性能が頭打ちになった場合のボトルネックを解析する際の手掛かりとしても有効です。

しかしながら、定量的な仮想化オーバーヘッドは単純には計算できないため、実アプリケーションを用いて性能測定をする必要があります。

## まとめ

仮想化技術適用時には、まずは機能面で要件を満たすかを検討する必要があり、特にCPUアーキテクチャや動作ゲストの確認は必須です。

また、サービスレベルやサービスモデルに依存して、構築しようとしているシステムにとって、故障影響レベルが

どこまで許容できるかを検討する必要があります。

そのうえでライブマイグレーション機能等の付加機能が、必要であれば考慮します。

また、仮想化製品の性能に関するオーバーヘッドは、下記に挙げるようなさまざまな要因が複雑に絡み合った結果です。

- ・ OS動作に依存する処理の割合
- ・ OS動作に依存する処理の順序
- ・ アプリケーションの負荷特性
- ・ 他ゲストの動作による影響

これらの要因のうち、何が支配的な要因になるかはアプリケーションと仮想化製品との組合せによってさまざまです。

今回提示したような基礎的な性能を参考にしつつ、実アプリケーションで所望の性能が得られるかを検証・評価することが重要です。

## 参考文献

- (1) <http://Imbench.sourceforge.net/>

## ◆問い合わせ先

NTT 研究企画部門 NTT OSSセンター  
 基盤技術ユニット OS-G  
 TEL 03-5860-5125  
 FAX 03-5463-5490  
 E-mail contact@oss.ntt.co.jp

このコーナーで取り上げて欲しいテーマをE-mailで編集部までお寄せください。  
 ●(社)電気通信協会内 NTT技術誌事務局  
 E-mail jimukyoku2008@tta.or.jp