

B-009

# セルフブート機能を持つマイクロベンチマークツールの開発

Development of Micro-benchmark tool with Self-boot function

岩元 祐太†  
Yuta Iwamoto

江藤 幹典†  
Mikinori Eto

梅野 英典†  
Hidenori Umeno

## 1. はじめに

近年の計算機の高性能化に伴い、通常通りマシン上で 1 つのオペレーティングシステム (Operating System: OS) を動かしたとしても、マシンにゆとりが生まれてきた。そこで、そのアイドル時間を有効利用するためのアプローチの 1 つとして、複数台の仮想計算機 (Virtual Machine: VM) を作り出し、複数の OS を動かす仮想化技術が注目され、様々な場で利用されている。例えば、システムソフトウェア開発の際のシミュレーションの単純化や仮想計算機を使用したサーバ統合によるサーバの効率的な運営などの利用がある。しかし、仮想計算機には仮想化に伴ってオーバーヘッドが生じてしまうというデメリットもある。

現在、仮想計算機システムソフトウェアとして、EMC 社の「VMware」、Microsoft 社の「Virtual PC」、ケンブリッジ大学で開発された「Xen」等が存在する。多くのユーザにとって、各仮想計算機システムの性能比較は興味のあることであろう。また、我々の研究室では独自の VMM の開発を目指しており、仮想計算機システムのオーバーヘッドを改善する手法の開発のために性能評価は肝要である。

しかし、仮想計算機のゲスト OS 上で既存のベンチマークツールを用いて性能比較を行うと、ゲスト OS によるスケジューリング等のオーバーヘッドが生じるため、純粋な VMM の評価を行うことができない。

そこで、本研究では、セルフブート機能を有し、ゲスト OS の機能に依存せずに動作する独自のベンチマークツールを開発した。本ツールは、仮想計算機システムの基本計算機インタフェースの動作を直接評価することに特化したマイクロベンチマークツールである。

## 2. 仮想計算機

### 2.1 仮想計算機の概要

仮想計算機とは、実計算機上に構成される仮想的な計算機である。この仮想計算機は、実計算機と相似のアーキテクチャを有しており、仮想計算機上で動作するプログラムは、その大部分をエミュレートすることなく、実計算機上で直接実行することができるという特徴を持つ。この直接実行は仮想計算機のアーキテクチャが実計算機のアーキテクチャとほぼ類似しているという特徴により可能となる。

この仮想計算機を同時に複数実行することを可能にするシステムを仮想計算機システムと呼び、仮想計算機システムを構成するための主要なシステムソフトウェアを仮想計算機モニタ (VMM: Virtual Machine Monitor) と呼ぶ。この仮想計算機システムの概念図を図 1 に示す。仮想計算機

システムを構成するためには、主要な計算機のリソースである CPU、メモリ、ディスク等の周辺機器を多重化し各仮想計算機に与える必要がある。この各リソースの多重化を VMM が行なう。この多重化のために VMM はプロセッサシミュレーション、メモリシミュレーション、I/O シミュレーションのシミュレーションを行なう。

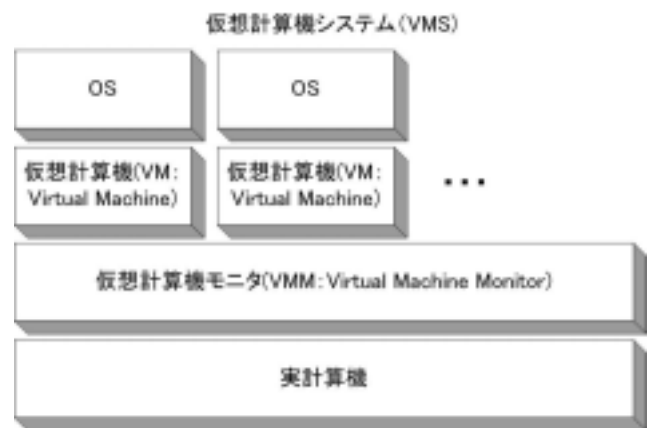


図 1. 仮想計算機システム (VMS)

## 3. マイクロベンチマークツール

### 3.1 マイクロベンチマークツールの概要

本研究で開発するベンチマークツールは、OS の影響を受けることなく、仮想化に伴うオーバーヘッドの正確な測定を目指す。そのためには、OS の機能に依存せず独立したベンチマークツールでなければならない。

そこで今回、セルフブート機能を有し、性能測定に必要な最低限の設定を OS に代わって行うマイクロベンチマークツールを開発した。本ツールは、命令単位の実行時間、ページテーブルエントリの更新によるオーバーヘッド等、様々な測定を可能である。さらに、ページングモードの ON/OFF、キャッシュの有無等の様々なシステム条件を任意に指定できるようにした。それにより、システム条件の変化が仮想化オーバーヘッドに及ぼす影響が評価可能になる。

### 3.2 マイクロベンチマークツールの実装

今回実装したマイクロベンチマークツールでは、評価指標としてタイムスタンプカウンタを用いる。タイムスタンプカウンタとは、Pentium 以降の x86 アーキテクチャ互換 CPU に搭載されているタイマであり、プロセッサのクロックサイクルごとにインクリメントされる。

また、本ツールでは以下のようにシステム条件を任意に指定できる。

† 熊本大学大学院自然科学研究科

- ページング

仮想化において、ページング機構やページテーブルの更新などは大きなオーバーヘッドを伴う。したがって、その部分の詳細な評価・比較を行えるように、ページングモードの ON/OFF を任意に切り替えられるようにした。

- 特権レベル

特権レベルの違いによって、仮想化オーバーヘッドにどのような差異が生じるかを評価・比較するために、各種ベンチマークテストをどの特権レベルで実行するか任意に指定することができる。

- CPU キャッシュ

CPU キャッシュが性能評価に影響を与えることは少なくない。特に命令の実行時間など、CPU キャッシュが測定に誤差を与えるような場合には使用すべきではない。そこで、CPU キャッシュの機能の ON/OFF を任意に切り替えられるようにした。

#### 4. 動作確認

開発したマイクロベンチマークを実際実計算機上と仮想計算機上で動作させた。今回使用した仮想計算機システムは、完全仮想化手法の商用の仮想計算機システムである。結果は表 1、表 2、表 3 に示す。

表 1 は、命令単位の実行時間の測定結果である。各命令 100 回ずつ実行したときの実行時間を測定した。今回用いた仮想計算機システムでは一般命令は直接実行され、実計算機上で実行した場合と同じ実行時間で処理されることが知られている。また、特権命令は仮想化されて実行される。本ツールを用いて、様々な一般命令や特権命令を測定した結果、表 1 のように一般命令では実計算機上と仮想計算機上での実行時間がほぼ同じとなり、特権命令では実計算機上と仮想計算機上での実行結果に差が生じた。このことより、本ツールは仮想計算機システムが行う仮想化を反映した実行時間の測定が可能であることが確認できる。

表 2 は、特権レベルの違いによる各命令の実行時間の変化を測定したものである。表 2 に示した PUSHF、SMSW はセンシティブ命令と呼ばれており、特権命令ではないが仮想化が必要となる命令である。また、これらはどの特権レベルで実行されるかによって振る舞いが異なる。表 2 を見ると、どちらの命令も特権レベル 3 では実計算機、仮想計算機でほぼ同じ実行時間となっている。しかし、特権レベル 0 では実行時間に明らかな差が見られる。このことより、前節に示した特権レベル移行機能が正しく動作していることが分かる。

表 3 は、ページングモード ON/OFF それぞれの条件で、ページテーブルエントリ (PTE) 5120 個を更新したときの実行時間である。今回用いた仮想計算機システムは、VMM 内にシャドーページテーブルという仮想計算機のメモリと実際のメモリを対応付けるテーブルを持っている。仮想計算機上で PTE が更新されれば、それに伴ってシャドーページテーブルも更新されるためオーバーヘッドが生じると考えられる。表 2 を見ると、実計算機上ではページング ON/OFF であまり差はなかったが、仮想計算機上ではかなりの差が測定された。これにより、前節に示したページングモード ON/OFF 切り替え機能が正しく動作していることが分かる。

また、CPU キャッシュ ON/OFF 切り替え機能も正しく動作することを確認している。

表 1. 実計算機・仮想計算機の実行時間比

	実計算機	仮想計算機	VM / 実機比
一般命令			
ADD	36	36	1.000
CWD	100	100	1.000
特権命令			
LGDT	20224	118056	5.837
STI	5096	820	0.1609

表 2. 特権レベルの違いによる実行時間の変化

	特権レベル 0		特権レベル 3	
	実計算機	仮想計算機	実計算機	仮想計算機
PUSHF	564	2516	564	608
SMSW	588	92508	588	576

表 3. PTE 更新の実行時間

	実計算機	仮想計算機
ページング ON	1.5E+06	3.4E+10
ページング OFF	1.5E+06	2.3E+06

単位はクロックサイクル

#### 5. おわりに

本稿では、仮想化に伴うオーバーヘッドの正確な測定を可能にするマイクロベンチマークツールの開発について述べた。動作確認の結果、今回開発したマイクロベンチマークツールは、様々なシステム条件の下で仮想化のオーバーヘッドを測定できることが確認できた。

今後は、

- グラフによる出力
- 新しい評価指標の模索
- I/O 待ち時間を測定可能にする

の 3 点を中心に機能拡充を目指す。

また、このマイクロベンチマークツールは、VT を用いた Xen 3.0 の完全仮想化環境では現在のところ動作しない。我々の研究室では完全仮想化の仮想計算機システムの開発を目指しており、そのアプローチとして VT を用いる手法と用いない手法の 2 通りが考えられる。どちらがより効率的に仮想化可能なのかを調査するためにも、本マイクロベンチマークツールが VT を用いた仮想化環境で動作することは重要である。早急に原因を解明し Xen 3.0 完全仮想化環境で動作可能にしたい。