Xen を用いた仮想化の実装と性能評価

松下　知明

1　はじめに

近年，システムの高速化・複雑化に伴い，性能を維持するためにサーバ数やストレージ数が増加傾向にある．また，サーバを構築する際，セキュリティの問題も考え，多くの異なるサーバを用意する必要がある．しかし，多くのサーバを用意するためには，管理コストや設置スペース，などの物理的な問題がある．この他にも，アイドル状態のサーバが増え，資源を有効的に活用できない問題もある．この問題を解決するために，仮想化技術が注目されている．仮想化技術は，1 台のサーバで複数の OS（オペレーティング・システム）を取り扱うことができる技術であり，ソフトウェアには VMware，VirtualPC，Xen[1)] など多くのソフトウェアがある．本報告では，Xen を用い，Xen のライブマイグレーションの実行と，Xen を用いないサーバと Xen を用いた場合の性能評価について述べる．

2　仮想化

2.1　仮想化技術とは

仮想化技術とは，物理的な資源を論理的な資源に変換することで物理的な制約から逃れ，より柔軟に資源を分割したり統合したりして利用できるようにする技術である．仮想化技術は「サーバ仮想化」と「ストレージ仮想化」の二種類に分けることができる．本報告では，サーバ仮想化について述べる．

2.2　サーバ仮想化

サーバ仮想化とは，1 台のサーバを複数の仮想的なコンピュータに分割し，それぞれが独立して別の OS やアプリケーションソフトを動作させることができる技術である．

仮想化の方法はプロセッサやメモリ，ディスク一式を複数の領域に分割する．これにより複数のサーバを用意する場合に比べ，物理的資源の管理にかかるコストが省け，需要に応じて柔軟に資源を配分することができる．しかし，仮想化によるオーバーヘッドのため，直接実行するよりも一部の性能面で劣る結果になる．本報告では，サーバ仮想化のひとつである Xen について述べる．

3　Xen

3.1　Xen とは

Xen は仮想マシン技術と呼ばれるサーバ仮想化の一種で，コンピュータを構成するリソース一式を仮想化する技術である．英国ケンブリッジ大学コンピュータ研究所で開発が始まり，現在では XenSource[1)] が中心になって開発が進められているオープンソース・ソフトウェアである．

3.2　Xen における仮想化方式

Xen のシステムでは仮想マシンは「Domain」と呼ばれる単位で管理され，Domain の中でゲスト OS が動作する仕組みとなっている．Xen の Domain には，「Domain0」と「DomainU」がある．

Domain0 は，物理ハードウェアにアクセスするためのデバイス・ドライバや他の Domain を管理する特権を持つ，ホスト OS の役割を果たしている．Domain0 以外のゲスト OS は，Domain はすべて DomainU である．

DomainU は物理ハードウェアに直接アクセスするためのドライバ類を持たず，Domain0 が備えるドライバにアクセスを依頼するためのドライバを持つ．

Xen の仮想化の方法には準仮想化と完全仮想化があり，Xen2.0 までは以下に示す準仮想化しかサポートされていなかったが，Xen3.0 からは完全仮想化もサポートされている．また，Xen の準仮想化には仮想マシンの移行機能が備わっている．

- 準仮想化（para-virtualization）
  Xen は準仮想化と呼ばれる実装手法を標準採用しており，この実装手法により，エミュレート時のオーバーヘッドを最小限に抑えることができる．しかし，OS のカーネルを Xen 用に修正する必要がある．

- 完全仮想化（full-virtualization）
  Xen3.0 はから実装された仮想化であり，カーネルに変更を加えずに OS をそのまま Xen 上で動作させることができる．ただし，VT(Virtualization Technology) 技術が実装されたプロセッサを使う必要がある．VT 技術は複数の OS を 1 つのプロセッサで同時に動かせる仮想化技術である．完全仮想化は，準仮想化に比べるとエミュレーションによるオーバーヘッドが増加するが，VT 技術が実装されたプロセッサを使うことで，カーネルを書き換えれない Windows などの OS も動作させることができる．

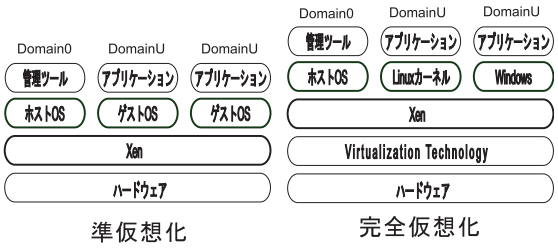

Fig.1　準仮想化と完全仮想化 (出典：自作)

3.3 Xen の利点

3.3.1 サーバの集約

Xen は，1 台のサーバに複数の OS を動作させることができ，省スペース化することができる．また，メモリ量やプロセッサ数などの資源も個別に割り当てれる．

3.3.2 高い実行性能

Xen はこれまでのエミュレーションと異なり，OS よりも下位のハードウェアに近い層で動作することにより，オーバーヘッドを極力少なくすることができる．

3.3.3 ライブマイグレーション

Xen のライブマイグレーション機能では，Fig. 2 のように稼働状態のまま仮想 OS を別のサーバに移動させることができる．

ライブマイグレーションが行われると，メモリの書き込みが少ない部分からコピーが始まり，差異が十分に小さくなるまで反復コピーされ，最後に一時停止し書き込みの多い部分をコピーする．現在の Xen では，HDD に持ってるデータまで自由に移動させることができないため，移動元と移動先の両方のサーバマシンからアクセスできる共有ディスクサーバが必要となる．

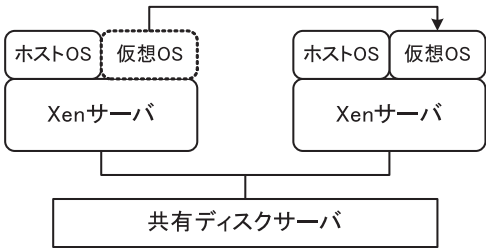

Fig.2　ライブマイグレーション (出典：自作)

4　Xen 環境の構築と性能評価

4.1 ライブマイグレーションの実行

Fig. 2 のように，2 台の Xen サーバと 1 台の共有ディスクサーバ (NFS) を用いてライブマイグレーションを行い，仮想 OS を他の Xen サーバ上に移行を行った．3 台のマシン環境を Table 1 に示す．

Table1　システム環境

プロセッサ	Pentium III 1GHz
Memory	512MB
HDD	32GB
network	Fast Ethernet

共有ディスクサーバ	Deabin(2.6.8-2-386)
Xen サーバ (Domain0)	Deabin(2.6.16-xen)
Xen サーバ (DomainU)	Debian(2.6.16-xen)

4.2 性能評価

Xen を用いていない OS と Xen カーネル上のホスト OS，ゲスト OS の性能評価には Linpack[2] を用いる．用いた環境を Table 2 に示す．

Linpack とは，「TOP500 Supercomputer Sites」の標準ベンチマークとして採用されている PC のベンチマークプログラムである．連立一次方程式を解くプログラムを用いて性能を測るプログラムで，浮動小数点演算の性能を計測することが可能であり，Table 3 にパラメータを示す．

- 問題サイズ (Ns)
  問題サイズは Linpack で N 次元連立方程式を解く際の問題の大きさである．一般にスワップ領域を使わないように，メモリの 80 %程度に設定し問題サイズを求める．
- ブロックサイズ (NBs)
  ブロックサイズは，小さくすることによって各プロセスのロードバランスが良くなるが，総ブロック数が増大することによりプロセス間の通信回数も増大する．今回のマシンでの最適な値は，過去の計測より 224[3] である．
- プロセスグリッド (P，Q)
  プロセスグリッドは, 問題の行列の分割方法を示している．P ＜ Q となる組み合わせが推奨されている．

Table2　システム環境

CPU	AMD Opteron 244 (1.8GHz)
Memory	1.0GB
HDD	36GB
network	Gigabit Ethernet

Xen を用いないサーバ	Deabin(2.6.8-2-386)
Xen サーバ (Domain0)	Deabin(2.6.16-xen)
Xen サーバ (DomainU)	Debian(2.6.16-xen)

Table3　パラメータ

問題サイズ (Ns)	10000
ブロックサイズ (NB s)	224
プロセスグリッド (P，Q)	1，1

4.3 評価結果

Table 4 に実験から得られた結果を示す．ここでは，プロセッサ数には 1，コンパイラには gcc3.3.5，ライブラリには ATLAS を用いた．理論値は 3.6GFlops である．

Table4　性能比較

	実効性能値	実効性能割合
Xen を用いないサーバ	2.913 GFLOPS	80.9%
Xen サーバ (Domain0)	2.858 GFLOPS	79.4%
Xen サーバ (DomainU)	2.920 GFLOPS	81.1%

サーバは理論値では 3.6GFlops の演算性能を持つ．実行性能割合はこの理論値に対する実行性能値の割合である．この結果により，Xen の仮想化 OS が，実サーバと同等の能力を発揮していることが判断できる．

5　今後の課題とまとめ

本研究では，近年注目されている仮想化技術とその仮想化技術の 1 つである Xen について述べた．Xen のインストールを行い，ライブマイグレーションの実装，Xen 環境での仮想化の影響によるオーバーヘッドの性能評価を行った．

今後の課題としては，ライブマイグレーションが実行されたときの停止時間と，ライブマイグレーションの実行で性能にどのくらい影響を及ぼすかを調査する．

参考文献

1) http://www.xensource.com/
2) http://www.netlib.org/benchmark/hpl/index.html
3) http://mikilab.doshisha.ac.jp/dia/research/report/2004/0811/002/report20040811002.html