

仮想計算機を用いた分散処理のための負荷分散手法に関する研究

中村 友太郎[†] 寺澤 卓也[†]

東京工科大学 大学院 バイオ・情報メディア研究科[†]

概要：仮想化技術の飛躍的な向上と普及により仮想マシン上でサービスが提供されてきている。本研究では複数の仮想マシンを用いて分散処理を行うことを想定し、これに適した負荷分散機構を提案する。構成の違う複数の実マシン上にそれぞれ複数の VM を置き、負荷のパターンを変えて検証した結果、負荷分散の動作を確認できた。

1.はじめに

仮想化技術の飛躍的な向上と普及に伴い、仮想マシン(VM)上でテスト環境の構築や、サービスの提供などが行われるようになった。仮想マシンが動作している実マシンの負荷は、複数の実マシン間で仮想マシンの移動(マイグレーション)を行うことで、分散・均衡を図ることができる。このことは佐藤らの研究[1]により、その有効性が示されている。

そこで本研究では、仮想計算機環境に負荷分散機構を付加することで効率的に実現する手法を提案する。具体的には仮想マシン群で分散処理を行うことを想定し、負荷分散機構を Xen(図 1) [2]環境上を実現する手法について検討する。

Xen は英国ケンブリッジ大学で開発されたオープンソースの Virtual Machine Monitor(以下 VMM)である。これはハードウェアのリソースを仮想化し、単数もしくは複数の、同じもしくは異なる OS(仮想マシン)を同時に並行して動作させることができる。この仮想マシンには dom0 と domU の 2 種類がある。図 1 に Xen の概要を示す。

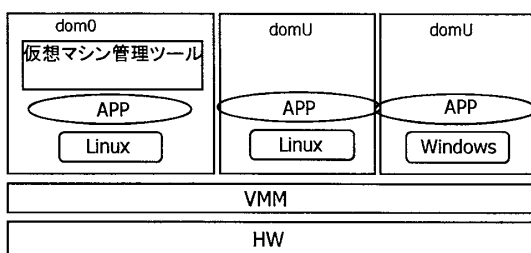


図 1 Xen の概要

Xen 仮想化の方式には準仮想化と完全仮想化の 2 種類がある。本研究ではより高い性能が得られるため準仮想化を選択した。

2.負荷分散手法

本研究では仮想計算機環境を用いた分散処理の負荷分散のためにマイグレーションを用いる。

その制御の方法として 2 種類の方法を検討・実装し、その比較をすることで最適な方法を見出す。1 つ目の方法は集中管理マシンを設置し、それが複数の実マシンの情報収集と負荷分散制御を行う [3]。2 つ目の方法は dom0 の OS 上のプログラムが協調して情報収集・負荷分散制御を行う。また、2 つ目の方法を発展させ、各 dom0 のカーネルの中で情報収集や負荷分散を行う方法(図 2)を提案する。カーネル内の実行によりシステムコールなどのオーバーヘッドが低減できると考える。

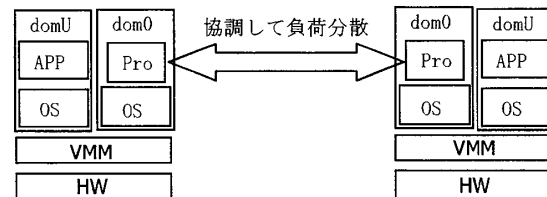


図 2 dom0 中で協調

本研究では実マシンの負荷を次の式で求めた。

$$\text{CPU 使用量} = \text{CPU クロック数} \times \text{CPU コア数} \times \text{CPU 使用率}$$

本来であれば L1, L2, L3 キャッシュ容量、FSB クロック数、アーキテクチャの違い等を考慮した方がより実的な CPU 性能を表すことができるが、異なるプロセッサの性能を正確に比較することは難しいため、今回は libvirt[4]を用いて取得できる CPU クロック数、CPU コア数、CPU 時間を用いて CPU の「使用量」を求めた。libvirt は仮想マシンの情報を取得する API を提供する、レッドハットを中心としたオープンソースプロジェクトである。本研究での CPU 使用率は dom0 から取得したものであり、domU の CPU 使用率も dom0 から取得したものである。また CPU 使用率は以下の式で求めた [5]。t はサンプリング時間、nr_cores はコア数を表している。cpuTime はナノ秒単位で取得している。

$$\text{CPU 使用率} = 100 \times (\text{cpuTime(now)} - \text{cpuTime(t seconds ago)}) / (t \times \text{nr_cores} \times 10^9)$$

マイグレーションを行うかどうかの判断は実マシンの負荷の値を比較して行う。この値が最も高い実マシン上で動作している domU の中で、最も負荷が高い domU をマイグレーション対象とした。また実マシンの負荷の値が最も低い実マシンをマイグレーション先の実マシンとした。そしてマイグレーション対象の domU をマイグレーション先へ移動させるようにした。また、マイグレーション先の実マシンに、マイグレーション対象となった domU の割り当てメモリ量分の空きがあるかどうかを事前にみている。

3. 実験と評価

上記の式を用いて実マシンの負荷を判断し、実マシン群の負荷分散を確認する実験を行った。異なるプロセッサ(クロック数、コア数)をもつ 4 台の実マシンを用意し、その上で domU を任意の数起動して負荷を様々な組み合わせでかけた。負荷としては単純な無限ループや π 計算を行った。負荷として並列計算を用いるアプリケーションは実行していない。また何らかの理由により人の手が介入した場合も想定した組み合わせも検証した。様々な組み合わせで検証を行った結果、dom0 が低負荷である場合は予測に沿った負荷分散結果が得られた。しかし dom0 の負荷が高い場合は負荷分散自体は行われるが、負荷が高い実マシンは負荷が高いままである結果になった。これは domU の負荷の高低にかかわらず、dom0 自体の負荷が高いためである。本来 dom0 は管理マシンであるため高負荷になることは考えにくいので、これは大きな問題ではない。また、本システム自体が実マシンに与える負荷はほとんどなかった。

4. 関連研究

PC 群の負荷を分散・軽減する手法として、プロセッサやネットワークの負荷予測に基づいて、動的にタスクを割り付ける手法[6]、演算コストの見積もり関数を用いて負荷のスケジューリングを行う方法[7]などが挙げられる。また[8]ではジョブスケジューリングの指標として CPU の負荷やメモリの空き容量、ネットワーク性能を考慮している。

5. まとめ

CPU の使用率は刻々と変わるものであり、本

方式の負荷分散が常に実マシン群全体にとって都合の良いように行われるとは限らない。これに対処するためには、情報取得を行う間隔や、そのときのみ単発で負荷が上がっているのか、徐々に上がっているのか、一定周期で上がっているのかなどの負荷の上がり方について考慮する必要がある。また、CPU 性能を数値化する際に CPU クロック数、CPU コア数、CPU 使用率のみで判断しているため、厳密に性能を数値化できていない。これをより正確に数値化するためには L1,L2,L3 キャッシュ容量とその有無、FSB クロック数、アーキテクチャの違いなどを考慮することが必要である。また、並列アプリケーションを実行した場合の検証や、アプリケーションの性質上、HDD に頻繁にアクセスする処理があった場合、それをマイグレーションしてしまうとリモートアクセスになってしまうような場合について対応していくことも今後の課題である。

参考文献

- [1] 佐藤賢斗、佐藤仁、松岡聡、「仮想クラスタを用いた Data-Intensive Application 実行環境の性能モデル構築と最適化に向けて」、先進的計算機基盤シンポジウム SACSIS2008、pp.46~47、2008
- [2] <http://www.xen.org/>
- [3] 窪田昌史、前田哲宏、北村俊明、「非均質環境向けの仮想計算機を用いた動的タスク分散システムの構築」、先進的計算機基盤シンポジウム SACSIS2007、pp.190~191、2007
- [4] <http://libvirt.org/>
- [5] <http://et.redhat.com/~rjones/virt-top/faq.html>
- [6] 小出洋、山岸信寛、武宮博、笠原博徳、「資源情報サーバにおける資源情報予測の評価」、情報処理学会論文誌：プログラミング、Vol.42、No.SIG3(PRO10)、pp.65-73(2001)
- [7] 綱代育大、田中淳裕、「仮想計算機環境における資源管理オーバーヘッドの評価」、情報処理学会研究報告 2006-EVA-17(4)、2005
- [8] 内裕和、小出洋、「並列分散プログラムのためのジョブマイグレーションスケジューリングの実装と評価」、情報処理学会研究報告 2005-HPC-102 (5) pp.27-32、2005

A Load Balancing Technique for Virtual Machines

†Tomotarou Nakamura Tokyo University of Technology

‡Takuya Terasawa Tokyo University of Technology