

平成 27 年度 卒業論文概要			
所 属	機械情報工学科	指導教員	光來 健一
学生番号	12237044	学生氏名	高山 都句子
論文題目	仮想マシンへの CPU 割り当ての変化に応じた最適な並列アプリケーション実行		

1 はじめに

近年, CPU のマルチコア化に伴い, 処理を分割して複数の CPU コアに割り当てることで同時に行う並列アプリケーションが増加している. また, クラウドコンピューティングの普及により, 仮想マシン (VM) 内で並列アプリケーションを動かすことも増えてきた. クラウドにおいてはサーバのメンテナンス時や負荷分散時に動作中の VM を別のホストに移動させるマイグレーションを行うことがある. この際に, マイグレーション先のホストの CPU が不足していると, VM にはマイグレーション前よりも少ない CPU しか割り当てられなくなる. その結果, VM 内で実行されている並列アプリケーションの性能が低下するが, 実験により, CPU 割り当ての減少分以上に性能低下が起きることが分かった.

本研究では, VM が利用可能な CPU の数を見積もり, CPU の不足時に並列アプリケーションを最適に実行する pCPU-Est を提案する.

2 CPU の不足による VM の性能低下

仮想化環境では, 図 1(a) のように, 物理的な CPU コア (物理 CPU; pCPU) は仮想化され, 仮想的な CPU (仮想 CPU; vCPU) として VM に提供される. CPU が十分にある場合には, 一つの物理 CPU は一つの仮想 CPU にだけ割り当てられる. VM 内の並列アプリケーションは複数のスレッドを用いることで, 複数の仮想 CPU を利用する.

2.1 CPU 不足時の対処

マイグレーション先のホストの CPU が不足している場合には以下の三つの対処が考えられる.

- VM に割り当てる物理 CPU を削減する
VM の持つ仮想 CPU 数は変わらないため, 図 1(b) のように一つの物理 CPU が複数の仮想 CPU に割り当てられる.
- VM が利用できる CPU の使用率を制限する
初期状態では合計で VM に割り当てられる物理 CPU 数 $\times 100$ (%) の CPU を使用することができるが, 図 1(c) のように CPU 使用率の上限値を設定することができる.
- 複数の VM 間で物理 CPU を共有する
図 1(d) のように物理 CPU を制限なく共有するため, VM が利用できる物理 CPU の割合は他の VM のワークロードに依存して変化する. 共有する VM の数が増えるほど利用可能な物理 CPU の割合が減る.

2.2 並列アプリケーションの性能低下

上記の三つの対処を行った時の並列アプリケーションの性能の変化を測定した. 実験には, AMD Opteron 6376 (16 コ

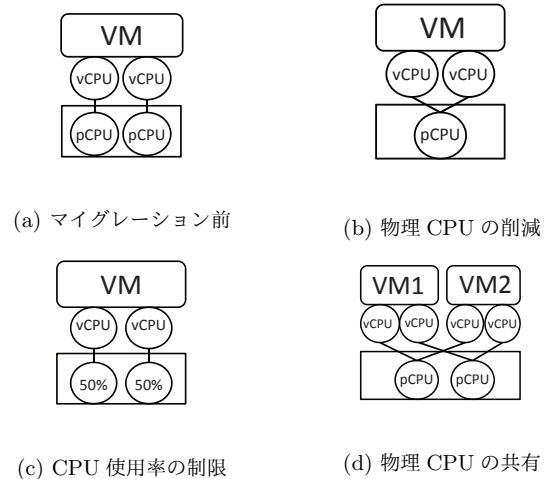


図 1: 物理 CPU の不足時の対処

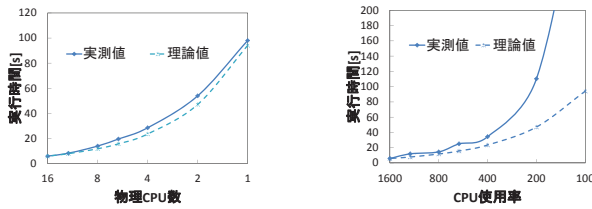
ア) の CPU を 2 基, メモリを 320GB 搭載したマシンを用いた. 仮想化ソフトウェアには Xen 4.4.0 を使用し, 16 個の仮想 CPU と 4GB のメモリを持つ VM を用いた. 並列アプリケーションとして, Tascell を用いて並列化されたフィボナッチ数計算 (fib) と NAS Parallel Benchmarks 中の BT を 16 スレッド用いて実行した. 物理 CPU を削減する場合には 16 個から減少させ, CPU 使用率を制限する場合には 1600% から減少させた. 物理 CPU を共有する場合には VM 数を 1 から増加させた. また, 物理 CPU の減少分だけ低下した時の性能を理論値とした.

図 2 に fib を用いた場合の実験結果を示す. どの対処を行った場合でも理論値より性能が低下していることがわかる. BT でも同様であり, 理論値からの性能低下は 16%~5243% であった. これは仮想 CPU のスケジューリングのオーバーヘッドが原因と考えられる. システム全体の仮想 CPU 数が物理 CPU 数より多い場合, スケジューリングによって仮想 CPU を切り替えるオーバーヘッドが生じる.

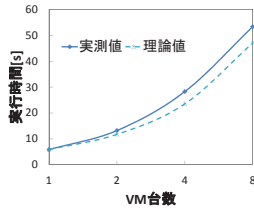
3 pCPU-Est

本研究では, VM が利用可能な物理 CPU 数を見積もり, 仮想 CPU の切り替えを減らす pCPU-Est を提案する. これにより, 物理 CPU の不足時に並列アプリケーションを最適に実行することができる. pCPU-Est は以下の三つの最適化手法を提供する.

- アプリケーションのスレッド数の最適化
利用可能な物理 CPU 数に応じて, アプリケーションが利用するスレッド数を調整する. スレッド数を利用可能な物理 CPU 数と同数以下にすることで, 仮想 CPU の切り



(a) 物理 CPU の削減 (b) CPU 使用率の制限



(c) 物理 CPU の共有

図 2: CPU 不足時の対処による fib の性能低下

替えを減らす。スレッド数は VM のユーザが容易に調整することができるが、実行中のアプリケーションのスレッド数を変更できるかどうかはアプリケーション依存である。

- VM の仮想 CPU 数の最適化

利用可能な物理 CPU 数に応じて、VM の持つ仮想 CPU 数を物理 CPU 数と同数以下にすることで、仮想 CPU の切り替えを減らす。VM の仮想 CPU 数を変更することはシステムの管理者にのみ可能であるが、他の VM への影響は小さいと考えられる。

- vCPU スケジューラのタイムスライスの最適化

vCPU スケジューラは物理 CPU を割り当てる仮想 CPU を定期的に切り替える。この間隔はタイムスライスと呼ばれ、タイムスライスを長くすることで仮想 CPU の切り替え頻度を減らし、オーバーヘッドを減らすことができる。タイムスライスの変更はシステム管理者のみが行うことができ、システム全体に大きな影響を及ぼす。

VM が利用可能な物理 CPU 数は以下のアルゴリズムに基づいて計算する。vCPU アフィニティが設定されており、仮想 CPU に特定の物理 CPU が割り当てられている場合は設定に使われている物理 CPU 数とする。vCPU アフィニティが設定されていない場合はシステム内の物理 CPU の総数とする。CPU 使用率の上限が設定されている場合は、その値を 100 で割って四捨五入した数とする。物理 CPU が複数の VM に共有されている場合には、VM に割り当てられている物理 CPU 数を共有している VM の数で割った値とする。利用可能な物理 CPU 数が VM の仮想 CPU 数を上回る場合は VM の仮想 CPU 数とする。この値を返すハイパーコールを作成し、VM 内から取得できるようにした。

4 実験

pCPU-Est を用いて、物理 CPU の不足時の三つの対処について最適化を行い、並列アプリケーションの性能の変化を調べ

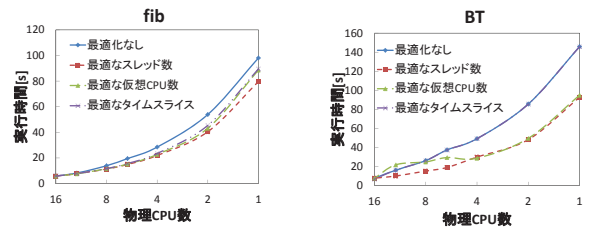


図 3: 物理 CPU の削減時の性能改善

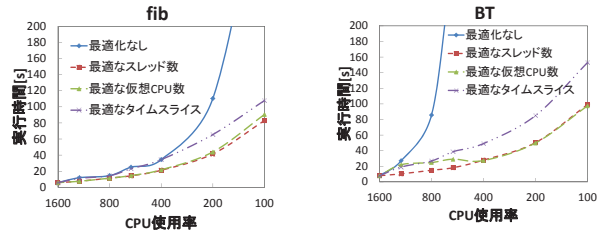


図 4: CPU 使用率の制限時の性能改善

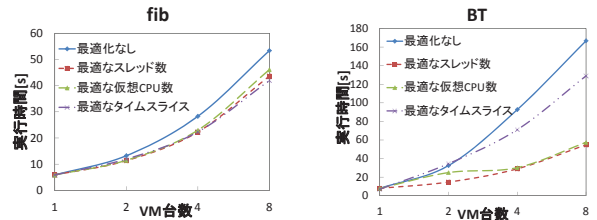


図 5: 物理 CPU の共有時の性能改善

た。この実験でも 2.2 節と同じ実験環境および並列アプリケーションを用いた。図 3 に VM に割り当てる物理 CPU を削減した場合の三つの最適化の効果を示す。スレッド数の最適化が最も性能低下を抑えることができ、最適化を行わなかった場合の 5%~96% 性能が改善した。仮想 CPU 数の最適化も効果的であるが、スレッド数の最適化ほど性能が改善しない場合があることも分かった。一方、タイムスライスの最適化は BT では効果がなかった。これはスレッド間での通信頻度が高いためだと考えられる。

図 4 に VM の CPU 使用率を制限した場合の最適化の効果を示す。スレッド数の最適化と仮想 CPU 数の最適化については図 3 と同様の効果があった。タイムスライスの最適化については図 3 とは異なり BT においても効果がみられた。また、図 5 に VM 間で物理 CPU を共有した場合の最適化の効果を示す。この場合も図 3、図 4 と類似の結果となったが、タイムスライスの最適化は fib では非常に効果的であったのに対し、BT では一定の効果しかなかった。

5 まとめ

本研究では、VM が利用可能な物理 CPU 数を見積もることにより、物理 CPU の不足時に並列アプリケーションの実行を最適化する pCPU-Est を提案した。アプリケーションのスレッド数の最適化が最も効果的であることが多く、次に仮想 CPU 数の最適化がよいことがわかった。今後の課題は、実行するアプリケーションに応じて最適化の効果を予測できるようにすることである。