

# 仮想計算機モニタとゲスト OS の協調による CPU 資源割付け方式

福島 滉章<sup>†</sup>      大月 勇人<sup>†</sup>      瀧本 栄二<sup>‡</sup>      毛利 公一<sup>‡</sup>  
<sup>†</sup>立命館大学大学院情報理工学研究科    <sup>‡</sup>立命館大学情報理工学部

## 1 はじめに

IaaS や VPS でサーバを構築し、運用する形態が一般的に利用されるようになった。この形態では、仮想化技術を用いて物理計算機上に複数のサーバを構築可能としている。しかし、従来の仮想計算機モニタ (以下、VMM) は、ゲスト OS 上で動作するプロセスの数や動作特性を考慮した資源割付けを行わない。これでは、仮想マシン (以下、VM) がどの程度資源を必要としているかわからないため、ある時点で、より処理能力を必要とする VM に資源が与えられず、一方の VM にのみ資源を偏って割り付けてしまう状態が起こり得る。そこで、各ゲスト OS で動作するプロセス群の動作特性を動的に検出することで、各 VM が必要とする資源を把握する。そして、余剰な資源を回収したり、余剰な資源を高負荷なゲスト OS の VM に割り付けることで、効率的な資源利用を目指す。本論文では、各ゲスト OS の動作特性を分類し、それに合わせて CPU 資源を動的に割り付ける CPU 割付け方式を提案する。

## 2 提案システム

### 2.1 概要

提案システムは、全てのゲスト OS が高負荷のとき、VM に割り付ける CPU 資源は、VPS と同様に指定された CPU 資源の割付けを保証する。一方で、各ゲスト OS の負荷にばらつきがあるとき、低負荷なゲスト OS の VM から CPU 資源を回収し、高負荷なゲスト OS の VM に割り付ける。これによって、低負荷なゲスト OS の VM が必要とする資源量を満たしつつ、高負荷なゲスト OS の VM に余力を与えることができる。すなわち、見かけ上は、システム全体で指定 CPU 資源以上の性能を発揮できる。

以上を実現するために、ゲスト OS は、CPU 使用率やプロセスの動作情報から動作特性を検出する。また、VMM は、各ゲスト OS の動作特性を動的に監視し、特性に応じた CPU 資源の割付けを行う。そのため、従来検出できなかった、CPU 使用率が低い状態を検出でき、より CPU 資源を必要としている VM を把握できる。

### 2.2 構成

本システムは、VMM として Xen を用いる。Xen 上の VM は Domain と呼ばれ、特権 Domain である Domain-0 とそれ以外の Domain-U がある。本システムでは、ゲスト OS に Linux を用い、Domain-0 で動作するゲスト OS をホスト OS、Domain-U で動作するゲスト OS をゲスト OS と呼ぶ。以下、動作の流れを述べる (図 1)。

1. CPU 情報を取得する。
2. 動作特性の分類を行う。

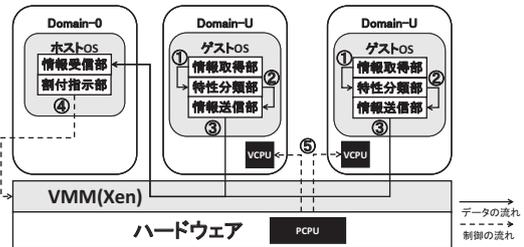


図1 システムの構成

表1 動作特性の分類

状態	CPU 使用率	LA	CS	WT
LACSHIGH	高い	高い	多い	-
LAHIGH	高い	高い	少ない	-
CSHIGH	高い	低い	多い	なし
IO	高い	低い	多い	あり
LACSLow	高い	低い	少ない	-
IDLE	低い	低い	少ない	-

表2 ゲスト OS 間の関係を考慮した動作特性の分類

状態	他ゲスト OS の状態	LA の比較
PriorityLA	LACSHIGH	大きい
CoreAvg	LACSHIGH	以下
PriorityGuest	CSHIGH 以下	-

3. ホスト OS に動作特性、CPU 情報の送信を行う。
4. 受信した情報を基に CPU 回収・割付けを指示する。
5. 各 VM に CPU を割り付ける。

## 3 動作特性の分類

本システムでは、各 OS の一般的な動作情報の指標である、CPU 使用率、ロードアベラージュ (LA)、コンテキストスイッチ数 (CS)、待ち時間 (WT) の 4 つを使用して動作特性を分類する。典型的な特徴を持つ特性に分類することで、特性に応じた資源割付けができる。表 1 に動作特性の分類を示す。なお、各特性の基準は UnixBench による予備実験の結果から決定した。表 1 の分類では、各ゲスト OS 自身で動作特性の検出を行うため、複数ゲスト OS の関係を考慮できない。これを解決するために、表 2 に示す VMM が検出するゲスト OS 間の関係を考慮した動作特性の分類を用いる。サーバ負荷の指標のひとつである LA を基準とし、LA が高いゲスト OS が高負荷な状況であると考え、負荷優先の割付けを行う。以上の状態の遷移方法をまとめた状態遷移を図 2 に示す。

図 2 の状態遷移を用いて、それぞれの動作特性の検出を行う。図 2 で検出した各特性について述べる。

「IDLE」は、処理を行うプロセスがわずかで CPU をほとんど使用していない状態を示す。

「LACSLow」は、少数のプロセスが処理を行っている状態を示す。LA、CS が高くないため、割り付けている CPU で十分であると判断できる。

「IO」は、待ち時間がある状態である。待ち状態のプロセスが存在するため、CPU 資源を増加して割り付け

CPU Resource Allocation Method by Cooperation between Virtual Machine Monitor and Guest OS

Hiroki FUKUSHIMA<sup>†</sup> Yuto OTSUKI<sup>†</sup> Eiji TAKIMOTO<sup>‡</sup> and Koichi MOURI<sup>‡</sup>

<sup>†</sup>Ritsumeikan University

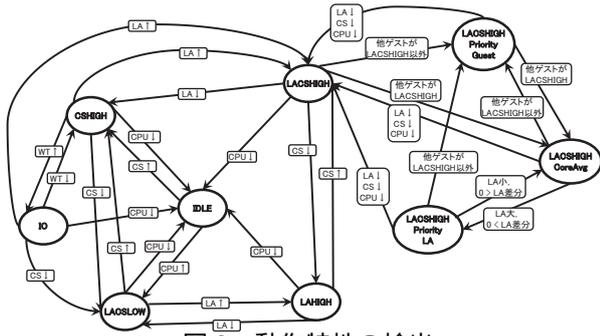


図2 動作特性の検出

ると、過剰に割り付けてしまう恐れがある。また、減少させると、実行状態のプロセスの処理が行えなくなる。

「CSHIGH」は、LAは増加していないため、プロセスは十分処理できる状態である。この場合、CPUを増加し、並列処理させることで、各プロセスのレスポンス向上が見込める。

「LAHIGH」は、優先度が高いプロセスがCPUを占有している状態である。この場合、CPUを増加することで、後続のプロセスが処理できる。

「LACSHIGH」は、割り付けているCPU資源では十分な処理ができていない状態である。すなわち、高負荷な状態である。CPU資源を増加して割り付けることで、多くのプロセスを並列処理ができる。

さらに、LACSHIGHの状態では、他のゲストOSとの関係において以下の状態を持つ。

「PriorityGuest」は、LACSHIGHの状態では他のゲストOSがCSHIGH以下の状態である。CSHIGH以下の状態は、割り付けているCPUで十分な処理が行えている。そのため、CPU資源を回収して割り付ける。

「CoreAvg」は、全てのゲストOSがLACSHIGHの状態である。各ゲストOSが高負荷な状態であるため、指定したCPU資源を割り付ける。

「PriorityLA」は、他のゲストOSと比べLAが高く、増加傾向にある状態である。つまり、各ゲストOSの中でもっとも高負荷な状態である。そのため、余剰なCPU資源を優先的に割り付ける。

## 4 評価

### 4.1 評価方法

評価環境(表3)に2つのゲストOSとして、表4に示すLAMPサーバを構成した。ホストOSに1コアのCPUを専有して割り付け、残りの7コアは2つVMで使用する。なお、OpenPNEには、1万ユーザ、8万日記を登録する。ベンチマークツールにJMeterを用いて、実際の利用環境を想定した計測を行う。JMeterは、Throughput Shaping Timerというタイマを使うことで、1秒あたりのリクエスト数をスケジューリングすることができる。これを用いて負荷に変化を与えて計測を行う。また、比較対象として、2つのVMで7コアを共有して割り付ける方式(7コア共有)、4コアと3コアで固定して割り付ける方式(固定割付け)と、本システム(Coop)の比較を行う。Coopは、4コアを優先して割り付けるVMと3コアを優先して割り付けるVMを用意する。評価軸は、各ゲストOSにおけるレスポンスタイムを用いる。

固定割付けは、固定コア以上の処理が行えず、低負荷時にIDLE状態のCPUが存在し、効率的ではない。また、7コア共有は、VCPU7コアを2つのVMが持って

表3 実計算機

資源	環境
CPU	Xeon X5482 3.20GHz (4コア)*2
RAM	64GB
Xen	4.2.4

表4 LAMPサーバ構成

SW	バージョン
OS	CentOS 6.4
Apache	2.3
MySQL	5.1.73
PHP	5.3.3
OpenPNE	3.8.6

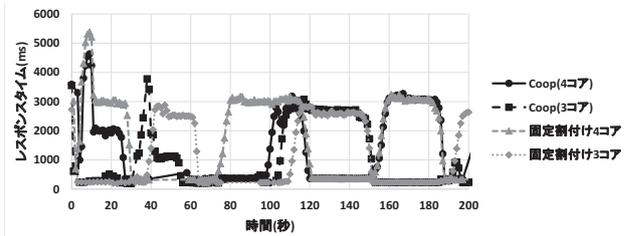


図3 Coopと固定割付けとの関係

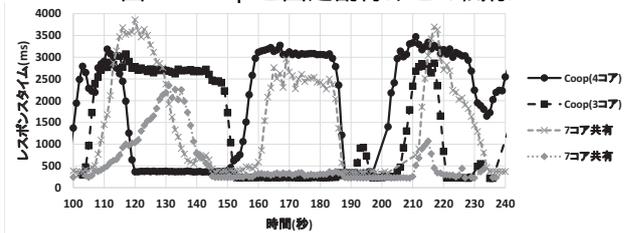


図4 Coopと7コア共有との関係

いるため、VMMは、VCPU14コアに対してPCPU7コアをスケジューリングを行う。CPUスケジューリングによるオーバーヘッドがネックとなる恐れがある。本システムは、固定割付け時の処理性能を保証しつつ、低負荷なゲストOSのVMからCPU回収し、高負荷なゲストOSのVMに割り付けることでCPUを効率的に使用する。また、VCPUとPCPUを同数に割り付けることで、VMMが行うCPUスケジューリングは発生しない。本システムによるCPU割付けにより、システム全体として性能向上が見込めることを示す。

### 4.2 評価結果

レスポンスタイムの推移をCoopと固定割付けとの関係(図3)と、Coopと7コア共有との関係(図4)で示す。図3より、0~60秒の範囲では、片方のゲストOSに偏らせて負荷を与えている。よって、Coopは、CPU資源をより負荷あるゲストOSのVMに割り付けるため、速いレスポンスが返すことができる。また、100秒以降は、両ゲストOSに負荷が与えられているため、指定したCPU資源を優先して割り付ける。よって、固定割付けと同等の性能を保証できている。図4は、7コア共有割付けの片方のレスポンスだけが良く、他方は悪い結果である。これは、片方に偏ったCPU割付けがされていたためと考えられる。Coopは、指定CPU資源を優先して割り付けるため、偏ったCPU割付けを行わない。そのため、ゲストOS間で偏った性能差が出ることはない。以上の結果より、本システムは、固定割付け方式と同等のCPU資源を保証しつつ、IDLE状態のCPUを回収して割り付けることで、高い性能を発揮することができた。

## 5 おわりに

本論文では、仮想計算機モニタとゲストOSの協調によるCPU資源割付け方式について述べた。