2J3-3

仮想マシンの組替えによる負荷変動への自律適応

Recombining Virtual Machines to Autonomically Adapt to Load Changes

網代育大
Yasuhiro Ajiro

NEC サービスプラットフォーム研究所
Service Platforms Research Laboratories, NEC Corporation

A large number servers have been virtualized and are running in virtual machines for the efficient utilization of the CPU resource. Virtual machines are consolidated into a physical server under the constraint that the sum of their resource utilizations are below the threshold of the physical server. Since the utilizations of virtual machines usually fluctuate according to the time and the change in business operations, it is reasonable to reconsolidate virtual machines in response to the load changes. We propose a heuristic algorithm to dissolve the overloaded states of physical servers by moving a minimal number of virtual machines because the virtual machine migration is a costly process. This algorithm is evaluated with randomly generated utilization data having several patterns of load imbalance.

1. はじめに

企業の各部門がサーバの低価格化にともなって次々にサーバを導入してきた結果、多くの企業は 100 から 10,000 台以上ものサーバを保有するようになった。近年、その運用管理コストが企業の IT 投資の 7〜8 割を占めることが問題視されている。一方で、CPU の高性能化やマルチコア化により、多くのサーバは CPU 資源を有効に使いきれていない状況である。このような背景から、仮想化技術を用いて多数の低負荷なサーバを少数の高性能サーバに収容するサーバ統合が盛んに行われ、現在では多くのサーバが仮想マシン (VM) 上で稼働するようになった。

サーバ統合時にどのサーバ同士を組み合わせて 1 台の物理サーバに収容するかは、ベクトルパッキング問題として定式化できる。ある計算機資源の使用率 (例えば CPU 使用率) が x を示す VM と使用率 y を示す別の VM を同一の物理サーバ上で稼働させたとき、物理サーバの資源使用率はこれらの和 $x + y$ となる。しかしながら、計算機資源は有限であるため、資源使用率が許容値を超えるとシステム性能の極端な劣化を招く。様々な負荷をもつ多数の VM が与えられたとき、これらを収容するための最小の物理サーバを求めるのが、ベクトルパッキング問題における目的である。

ベクトルパッキング問題は NP 困難問題の 1 つとしてオペレーションズリサーチの分野で古くから知られており、詰込み効率の理論的な解析や多くのヒューリスティックアルゴリズムの提案がなされている。我々のグループは、複数資源 (CPU の他にディスクやメモリ、ネットワーク) の使用率とその相関関係を考慮したアルゴリズムを提案し、複数資源を考慮する多次元のベクトルパッキング問題に対しては資源使用率の高い VM を余力の大きな物理サーバに割り当てる負荷分散的なアルゴリズムの方が詰込み効率がよいという知見を得ている [Ajiro 07]。また、VM の稼働時には VM 数に応じたオーバヘッドがかかり、特に稼働 VM 数が CPU のコア数を超えるとオーバヘッドが増大する傾向が見られる [VMw 06a, 網代 06]。各物理サーバに収容される VM の数をおおよそ等しくできる負荷分散的なアルゴリズムは、無駄な仮想化オーバヘッドの削減にも効果がある。

統合計画を立てる際の VM の資源使用率は一定期間の最大値等を基に見積もられるが、一般に使用率は月末や締め日といった時期的な要因や組織改編等のイベントによって大きく変動する。したがって、こうした負荷の変動に合わせて VM の配置を変更する (VM を組み替える) のが、計算機資源の有効活用の観点からは合理的である。VMware VMotion や DRS (Distributed Resource Scheduler) [VMw 06b] のように VM を他の物理マシンに移行するためのシステム的な仕組みは整いつつあるが、VM の移動はそれなりのコストとリスクを伴う処理である。

本研究では、一部の VM の負荷が変動し、物理サーバの資源使用率が閾値を超えた、あるいは超えることが予測された場合に、最低限の VM の組替えによって閾値の超過を解消するためのアルゴリズムを提案し、まずは資源数が 1 の場合の評価を行なった。このとき、VM が負荷分散的なアルゴリズムで統合されていることには、さらにもう 1 つの利点がある。VM を統合する際や組み替える際は、VM の資源使用率が極力小さい方が成功率が高いが、負荷分散的なアルゴリズムによって各物理サーバには大小様々な資源使用率をもつ VM が混在することになるため、使用率の小さい VM だけを組み替えられるという利点である。

2. 仮想マシンの組替え

たとえば夜間等に大部分の VM の負荷が大幅に減少する場合は、VM を再度統合し直すことで物理サーバの台数を削減し、電気代等のコストを削減することができる。逆に負荷が大幅に増加する場合はサーバの増設による対処が不可欠である。しかし、負荷の変動が一部の VM や物理サーバで局所的に発生している場合に、すべての VM を統合し直すのは無駄が大きい。そこで我々は、閾値の違反が少数の物理サーバで発生している場合に少ない VM の移動で違反を解消するための、反復深化探索に基づく VM 組替えアルゴリズムを提案する。

連絡先: 網代育大，日本電気 (株) 中央研究所
〒 211-8666 神奈川県川崎市中原区下沼部 1753
y-ajiro@cd.jp.nec.com

閾値の超過が解消されるまで、過負荷な物理サーバから VM の割当てを解除;
$m \leftarrow \infty$;
for $i \leftarrow 0$ **to** *MAX_MOVING* **do**
  各物理サーバから i 個の VM の割当てを解除;
  割当てを解除した VM を第 2.1 節のアルゴリズムを使って再配置した結果 (台数) を m' に代入;
  if $m' < m$ **then**
    $m \leftarrow m'$
  fi
end for

図 1: 組替えアルゴリズム

2.1 負荷分散に基づく統合アルゴリズム

VM 組替えアルゴリズムの紹介をする前に、まず我々の提案している最小負荷経路選択 (Least-Loaded) に基づく VM 統合アルゴリズムについて簡単に説明する。最小負荷経路選択は、Web サーバの前段に配備する負荷分散装置等で使われるポピュラーなアルゴリズムである。その仕組みは、到着したリクエストをもっとも負荷の軽いサーバに転送するだけの単純なもので、同一の機能をもった複数のサーバにリクエストを振りわけるのに用いられる。

サーバ統合の場合は、VM の個数や負荷があらかじめわかっているため、VM を資源使用率の大きい順にソートし、もっとも使用率の大きな VM から順にもっとも資源に余裕のある物理サーバに割り当てるという処理を繰返し行なうことで、各物理サーバの負荷を均等にする。しかしこれには物理サーバの台数があらかじめ決まっていなくてはならないため、ある台数の物理サーバに統合できない場合は、物理サーバの台数を追加して再度統合をやり直す再試行処理が必要である [Ajiro 07]。

2.2 組替えアルゴリズム

負荷変動によって物理サーバ間に負荷の不均衡が発生し、資源使用率が閾値を超過する過負荷な物理サーバと資源に比較的余裕のある低負荷の物理サーバが現れた場合を考える。過負荷なサーバから閾値の超過分の VM だけを低負荷の物理サーバに移行することで閾値の違反を解消できれば話は単純であるが、このような組替えが常に成功するわけではない。そこで、超過分の VM の再配置に失敗した場合に、各物理サーバから資源使用率の低い一定数の VM の割当てをいったん解除し、超過分の VM とともに再配置することを考える。各物理サーバから割当てを解除できる VM の個数 *MAX_MOVING* が与えられたとき、必要な物理サーバの台数 m を返す組替えアルゴリズムを図 1 に示す。なお、VM の割当て解除はすべて、資源使用率の低い VM から順に行なう。アルゴリズムは VM の割当て解除数に関する反復深化探索の手法を用いている。

3. 実験

評価のため、VM の資源使用率に関する擬似データを生成した。生成にあたっては次の 2 つの仮定をおいた。

1. 負荷変動の前後で全 VM の使用率の総和が一定。
2. 各 VM の使用率の分布は一様分布であり、負荷変動の前後で平均値が変化しない。

表 1: 物理サーバ 6 台の (想定) 負荷パターン

パターン	各物理サーバの資源使用率					
	A	B	C	D	E	F
1	125	75	100	100	100	100
2	125	87.5	87.5	100	100	100
3	125	95	95	95	95	95
4	150	50	100	100	100	100
5	150	75	75	100	100	100
6	150	90	90	90	90	90

仮定 1 は、ある VM の負荷が増大し、これを収容する物理サーバの資源使用率が閾値を超える一方で、別の (複数の)VM の負荷は減少して計算資源に余裕ができることを意味している。また仮定 2 から、負荷変動に伴う様々な閾値の超過状態が分布の平均値と VM の配置によって特徴づけられる。なお、企業内やデータセンタで運用される物理サーバの資源使用率が、正規分布よりも一様分布に近い分布をもつことは文献 [Rolia 05] 等から読み取れる。

実験では、6 台の物理サーバの閾値を 100 とし、表 1 に示す 6 パターンの閾値超過状態を生成した。パターン 1, 4 は物理サーバ A の超過分と B の余裕分とが等しい状態、パターン 2, 5 は A の超過分と B, C の余裕の合計、パターン 3, 6 は A の超過分と残りのサーバ B–F の余裕の合計が等しい状態である。なお、現実の CPU 使用率等の閾値は 50–80% の値をとることが多いが、ここでは簡単のため閾値を 100 とし、物理サーバ A の閾値超過分が閾値の 25% および 50% の状態を想定している。

次に、物理サーバに収容される各 VM の資源使用率を平均 25.0 または 12.5 の一様乱数によって生成した。それぞれ [0, 50)、[0, 25) の範囲の一様乱数を VM 数の分だけ生成し、第 2.1 節の統合アルゴリズムを用いて表 1 の資源容量をもつ 6 台の物理サーバへの統合案を求めることで、各パターンの閾値超過状態を作り出した。各 VM の平均使用率が 25.0 の場合は、6 台の物理サーバに平均 4 つの VM が収容されるため、24 個分の VM に対する使用率データを生成した。平均使用率 12.5 の場合は、同様に VM 48 個分のデータを生成した。しかしながら、統合後の物理サーバ A の使用率が 125 や 150 に遠く及ばない場合があるため、パターン 1, 2, 3 では統合後のサーバ A の資源使用率が 120 以上のデータだけ、パターン 4, 5, 6 では 145 以上のデータだけをサンプルとして抽出した。

各 VM の平均使用率として 25.0 を想定したときに実際に抽出された VM の平均使用率は、どの負荷パターンにおいても約 24.0、サーバ A の平均使用率は、パターン 1–3 に対しては 122、パターン 4–6 に対しては 147 であった。各 VM の平均使用率 12.5 を想定した場合では、実際に抽出された VM の平均使用率は 12.1 あるいは 12.2、サーバ A の平均使用率は、平均使用率 25.0 を想定したときと同様であった。

このようにして抽出したサンプルデータを 6 種類の負荷パターンと 2 種類の平均使用率に対してそれぞれ 100 個ずつ、合計 1,200 サンプルを用意し、各サンプルに対して図 1 の組替えアルゴリズムを適用した。移動 VM 数の許容値 (図 1 の *MAX_MOVING*) を収容 VM 数に対して充分大きな値に設定し、6 台の物理サーバに収容するのに物理サーバあたりいくつの VM の移動 (同図の i) が必要だったかを調べた結果を表 2 に示す。移動数 ∞ の列は、第 2.1 節のアルゴリズムでは 6 台

表 2: 物理サーバ 6 台への統合に必要な VM の移動数。表中の値は各負荷パターンに対する 100 サンプルの内訳。

(VM の平均使用率 25.0 の場合)

パターン	移動数							
	0	1	2	3	4	5	≥6	∞
1	76	10	5	3	0	0	0	6
2	15	49	22	7	0	1	0	6
3	4	11	50	15	6	0	0	14
4	78	18	1	3	0	0	0	0
5	47	29	18	3	0	1	0	2
6	0	38	45	6	1	0	0	10

(VM の平均使用率 12.5 の場合)

1	93	4	3	0	0	0	0	0
2	80	11	5	1	1	0	2	0
3	8	38	48	5	0	0	1	0
4	92	4	2	0	0	1	1	0
5	85	10	4	1	0	0	0	0
6	3	37	53	6	1	0	0	0

への統合がそもそも不可能なサンプルの個数を表している。

実験は Core2 Duo E6400 (2.13 GHz) CPU と 2 GB メモリを備えた x86 PC 上に Ubuntu Linux 7.1 を搭載し、開発実行環境として JDK 1.6 を用いて行なった。サンプル 1 つ分の VM の統合や組替えにかかった時間は数ミリ秒であった。

4. 考察

まず、VM の平均使用率 25.0 の場合、負荷パターン 1, 4 や 5 に対しては、移動数 0、すなわち閾値を超過しているサーバ A 以外の VM を移動せずに 6 台での組替えに成功している。それ以外のパターン 2, 3, 6 では、サーバ A 以外の VM の移動を必要とするものの、多くの場合、物理サーバあたり 2 つの VM(つまり全 VM の約半分) を組み替えることでサーバを 6 台に抑えることができた。もっとも結果の悪いパターン 3 においても、元々6 台に統合できなかった 14 サンプルを除く 86 サンプルのうちの 65 サンプル、全サンプルの約 75%は 2 VM の移動による組替えに成功した。

次に、平均使用率が 12.5 の場合は、パターン 1, 2, 4, 5 に対して大多数のサンプルが移動数 0 での組替えに成功している。平均使用率 25.0 の場合と違ってパターン 2 の結果がよいのは、サーバ B, C の資源の余裕分が 12.5 であり、これが VM の平均使用率と同程度であるためと考えられる。このことは、各 VM の資源使用率が一様分布に近い場合、VM の平均使用率以上の余裕を物理サーバにもたせることが、少ない VM の移動での組替えを成功させる上で重要であることを示唆している。一方、パターン 3, 6 においては、全サンプルの 93–94% が 2 VM (全 VM の 1/4) までの移動によって 6 台への組替えに成功した。

5. 関連研究

Khanna らは VM を収容する物理サーバ群の使用率の分散 (バラつき) を大きくすることによって、資源に余裕のあるサーバを確保し、負荷変動時の VM の移動を容易にするアプローチとそのためのアルゴリズムを提案している [Khanna 06]。また、負荷予測に基づいて VM を統合し、統合後の SLA(資源要求を満たしている期間の割合) を評価した研究があり、統合には使用率のピークが異なっている 2 VM 同士を統合していくヒューリスティクス [Rolia 05] や遺伝的アルゴリズム [Gmach 07] が用いられている。一方、プロセッサへのタスク割当てがベクトルパッキング問題として定式化されることは以前から知られており、Beck らは古典的なヒューリスティックアルゴリズムである FFD やその変形を文献 [Beck 96] で評価している。

6. まとめと今後の課題

VM の負荷変動に伴う閾値の超過に対し、少数の VM の移動で閾値の超過を解消するための、反復深化探索に基づく VM 組替えアルゴリズムを提案し、資源が 1 種類の場合について移動の削減効果を評価した。また、1 台の物理サーバの資源に VM の平均使用率程度の余裕があれば、非常に少ない VM の移動で組替えが成功しやすい可能性があることがわかった。しかしながら、実際の VM やサーバは CPU の他にディスクやメモリといった複数の資源を備えているため、各資源使用率の相関関係やどのようなタイミングで組替えを実行するかも考慮しなくてはならない。これらを含めた評価が今後の課題である。

参考文献

[網代 06] 網代育大, 田中淳裕：仮想計算機環境における資源管理オーバヘッドの評価, 情処システム評価研究会研究報告 (EVA-17), pp. 17–22 (2006)

[Ajiro 07] Ajiro, Y. and Tanaka, A.: Improving packing algorithms for server consolidation, in *Proc. 33rd Int. CMG Conference (CMG 2007)*, pp. 399–406 (2007)

[Beck 96] Beck, J. E. and Siewiorek, D. P.: Modeling multicomputer task allocation as a vector packing problem, in *Proc. Ninth Int. Symp. on System Synthesis (ISSS'96)*, pp. 115–120, IEEE (1996)

[Gmach 07] Gmach, D., Rolia, J., Cherkasova, L., and Kemper, A.: Capacity management and demand prediction for next generation data centers, in *Proc. IEEE Int. Conf. on Web Services (ICWS 2007)*, pp. 43–50 (2007)

[Khanna 06] Khanna, G., Beaty, K., Kar, G., and Kochut, A.: Application performance management in virtualized server environments, in *Proc. 10th IEEE/IFIP Network Operations and Management Symposium (NOMS 2006)*, pp. 373–381 (2006)

[Rolia 05] Rolia, J., Cherkasova, L., Arlitt, M., and Andrzejak, A.: A capacity management service for resource pools, in *Proc. Fifth Int. Workshop on Software and Performance (WOSP 2005)*, pp. 229–237, ACM (2005)

[VMw 06a] ESX server performance and resource management for CPU-intensive workloads, VMware technical resource (2006), available at http://www.vmware.com/vmtn/resources/

[VMw 06b] Resource management with VMware DRS, VMware technical resource (2006), available at http://www.vmware.com/vmtn/resources/