

VM クラスタにおけるサービスの発熱傾向を考慮したスケジューリング

伊勢 駿介†

廣津 登志夫†

† 法政大学情報科学部

1 はじめに

現在一般的になっているクラウド環境では、一台の物理マシン(ホスト)上に複数の仮想マシン(VM)を稼働させ、複数のユーザが一つのホストを共用する形態が一般的になっている。この際、稼働しているVMはユーザごとのそれぞれ異なる目的で利用されるため、配置によってはサーバラックの一部にホットスポットと呼ばれる熱の不均衡が生じることがある。高温なサーバを冷却する際には、サーバに急激な温度変化が生じることや周囲のサーバが過度に冷却されることがある。それによりクーラーの冷却効率が低下することやサーバが推奨稼働環境からはずれることで障害発生率が上昇することがあるため、データセンタではサーバ間の温度差が少ない状態が望ましい[1]。本研究ではVMのCPU利用傾向をVM作成時に属性としてユーザに申告させ、ホストの機種別に実測したCPU使用率とCPU温度のプロファイルと組み合わせることでVMのスケジューリングを行うことで、マイグレーションによる性能低下を抑えながらサーバ間の温度差の軽減を試みる。

2 既存研究

Kaurらの研究[2]では、VMのスケジューリングにおいてホストのCPU温度を考慮することでサーバ温度の均一化を図っている。システムを構成する各ホストのCPU温度を監視し、VM起動時には配置先としてCPU温度がメーカーの定めるクリティカル温度との差が一番大きいホストを指定し、CPU温度がクリティカル温度を超過した場合は超過しなくなるまでそのホスト上にあるVMをVM起動時と同様の方法で1台ずつ低温なサーバにマイグレーションしている。しかし、この手法ではその時点におけるVMの温度のみを見ているため、あるホストに負荷の変動が大きいVMが集まった場合に温度が安定せずにマイグレーションが続けて発生することや、負荷の低いVMが連続で追加された場合に低温のサーバが生じることが考えられる。

3 サーバ負荷と発熱の関連性

サーバの負荷がどのように発熱に影響を与えるのかを調査する。サーバの発熱は最終的に排気ファンによ

て外部に排出されるため、サーバ前部の吸気温度および後部の排気温度の差をサーバの発熱と捉える。負荷に応じて温度が変動する部品に負荷をかけたときのサーバの発熱を調査することで両者の関連性が分かる。CPUは計算で温度が上がると考えられるため、実際の使用状況を模倣するようなプログラムを用意する。なお、CPUの実行ユニットは一般的なプログラムではすべて使われるわけではないため、同時マルチスレッディング(SMT)機能を用いて利用効率の向上を図るが、科学技術計算のようなCPUの実行ユニットを効率よく利用するプログラムでは競合により性能が低下する。そのため、その違いを反映できるようなプログラムを用意する必要がある。

本研究では調査対象として4コア8スレッドのCPUを搭載したコンピュータを用いた。CPUに負荷をかけるため、一般的なプログラムを想定したプログラム(P1)およびSMT利用時の性能低下を想定したプログラム(P2)を用意した。P1では、整数および浮動小数の演算を実行しながら実行速度を測定するために一定回数実行する度にログに記録を残す。P2では、整数および浮動小数の演算の代わりにsqrt()関数を用いる。また、P1の一般的なプログラムとしての妥当性を調べるために、3層サーバクライアントシステムで構成されたチャットツールのLet's Chatとユーザを想定したスクリプト(P3)を用意した。NICに負荷をかけるために、pktgenを用いて64byteのショートパケットを生成して他のサーバに送った。HDDに負荷をかけるために、ddコマンドでHDDに連続したデータの書き込みを行った。CPUの負荷試験ではP1を0スレッドから8スレッド(CPUのスレッド数)まで増やしながら実行した結果、サーバの発熱は最大で5℃変化した。NICおよびHDDの負荷試験ではサーバの発熱に有意な変動は現れなかった。これらの結果からサーバの発熱の主要因はCPUの温度変化であるとわかる。以降、CPUの温度変化に着目する。

P1およびP2でCPU温度および性能低下の有無を調べるため、それぞれ0スレッドから8スレッドまで30分に1スレッドずつ増やしながら実行した。このときのCPU温度の変化およびログに記録された1スレッドあたりの実行回数の比をそれぞれ図1(左側がP1、右側がP2)および図2示す。また、P3を実行したときは同じCPU使用率においてP1と同じ温度になった。

A Thermal-aware Allocation Scheme using VM service classification

†Shunsuke ISE †Toshio HIROTSU

†Faculty of Computer and Information Sciences, Hosei University

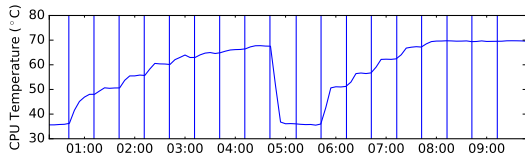


図 1: P1 と P2 の温度変化の差

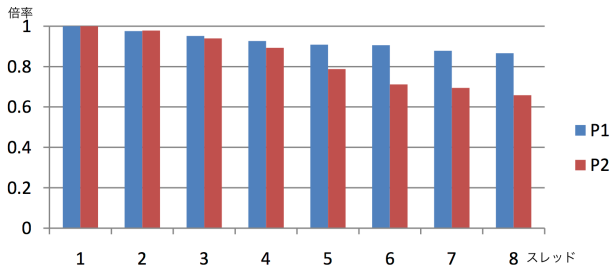


図 2: P1 と P2 のスレッド数ごとの実行回数の比

図 1 から P1 では CPU 温度が 35 °C から 68 °C まで上昇していて、3 スレッド実行したところで中間の 51 °C を超えた。P2 の場合はより急激に上昇して 5 スレッド実行したところで上昇が止まった (温度上限とする)。どちらの場合も CPU は使用率が低いときのほうが温度が上がりやすいと判明した。図 2 において P2 は温度上限に達した 5 スレッド目以降では実行回数の低下が目立つため、温度上限では CPU の実行ユニットが有効に使われた状態となることが考えられる。

4 VM の発熱傾向を考慮した制御

AWS や OpenStack のようなクラウドサービスでは、用途別に VM のディスクイメージおよびサイズを細かく分類してユーザに選択させる。本研究では VM の想定される CPU 利用傾向を Light, Normal および Heavy の 3 種類に分け、VM を起動するユーザに申告させる。Light VM はファイルサーバやオフィスワークを想定した VDI に代表される CPU 依存でない用途が想定される VM で、CPU 温度に対する影響は小さく、変動も少ない。Normal VM は Web サービスやメールサーバに代表される CPU 使用率が利用状況によって変動する用途が想定される VM で、CPU 使用率の予想が困難である。Heavy VM は人口知能や画像処理に代表される CPU 使用率が常時高い値を示す用途が想定される VM で、CPU 使用率の変動が少ないことを利用してホストの CPU 温度を安定して上昇させることができる。

これらの VM の区別を用いて、以下のようなポリシーでスケジューリングを行う。図 1 の P1 の温度が上昇幅の半分まで上昇するコア数までを Heavy VM に優先的に割り当てる (ポリシー 1)。物理コア数と同じ個数までコアを Normal VM に優先的に割り当てる (ポリシー 2)。図

1 の P1 の最高温度またはメモリ上限までコアを Light VM に割り当てる (ポリシー 3)。CPU が温度上限に達した場合は Light VM をマイグレーションする (ポリシー 4)。ポリシー 1 では複数ホスト間で熱発生を均衡化させるためには CPU 使用率の変動が小さいと想定される VM を均等に配置する。ポリシー 2 では CPU 使用率の変動が大きい Normal VM を余裕を持たせて配置する。ポリシー 3 では CPU 使用率とその変動が小さい Light VM を、余剰リソースを減らすために配置する。ポリシー 4 では性能低下の可能性を回避するため、図 1 の P1 の最高温度以下まで Light VM をマイグレーションする。VM 配置するホストを決定するため、各ホスト毎に VM の種類ごとに割り当てた総コア数を管理し、VM が起動したときはその VM の種類に割り当てたコア数が一番少ないホストに配置する。

5 考察

既存手法ではそのときの VM の温度のみを見ているため、あるホストに Normal VM が集まった際に温度が不安定になる可能性がある。本手法では Heavy VM をどのホストにも配置するため、全体の CPU 温度が上昇し、新たに VM を配置したときの変化量が減ると考えられる。VM マイグレーションは Light VM を優先的に行うので、マイグレーション先で負荷が変動して再度マイグレーションすることを防ぐことができると考えられる。また、本手法では CPU 温度の温度上限を調査することで、CPU の実行ユニットの限界まで VM を割り当てないようにして性能低下が発生する可能性を減らせると考えられる。

6 結論

データセンタにおけるホットスポットの発生を可能な限りサービス品質を維持したまま抑える VM 配置手法として、VM を想定される CPU 使用率で 3 種類に分類し、CPU の温度変化の特徴と組み合わせたスケジューリングポリシーを提案した。

※本研究は JSPS 科研費 JP15K00138 のを支援による

参考文献

- [1] A. Berl, E. Gelenbe, M. D. Girolamo, G. Giuliani, H. D. Meer, M. Q. Dang, and K. Pentikousis, “Energy-efficient cloud computing,” *The Computer Journal*, vol. 53, no. 7, pp. 1045–1051, 2010.
- [2] A. Kaur and S. Kinger, “Temperature aware resource scheduling in green clouds,” *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, no. 1919–1923, 2013.