

Adaptive Scheduling on Power-Aware Managed Data-Centers using Machine Learning

著者: Josep Ll. Berral, Ricard Gavaldà, Jordi Torres.

出典: *Proc. IEEE/ACM 12th International Conference on Grid Computing, GRID '11, pp. 66-73, 2011*

発表者: 高性能コンピューティング学講座 本多・近藤研究室 1353026 藤井 淳

1 はじめに

近年データセンタにおいてエネルギーコストの増大が問題になっている。データセンタでは次々にジョブが来る。これらのジョブを適切なホストに配置出来れば、最小限のホスト台数でデータセンタを運用することができる。しかし、どのようなジョブが来るかが予想できないため実際には最適な配置は困難である。

本研究ではクラウド環境での Web サービスにおいて、データセンタでの電力効率を向上させることで金銭的な利益を増加させるようにタスクスケジューリングをするフレームワークを提案する。想定環境としてユーザに VM を貸し出し、それぞれのホストの上で複数の VM を動かす Amazon EC2 のようなデータセンタを想定する。ユーザはこの VM 上で Web サーバを動かしているとする。

2 データセンタの数理計画モデル

2.1 Service Level Agreement

できるだけ少ないホスト台数でデータセンタを運用したいが、一方でユーザの期待する応答時間以内に結果を返す必要がある。そこでユーザ満足度を示す評価尺度として SLA (Service Level Agreement) 導入する。SLA は通常次の式で定義される。

$$SLA(Job_i) = \max(\min(1 - \alpha \frac{RT - RT_0}{RT_0}, 1), 0) \quad (1)$$

RT_0 はユーザの期待する応答時間を示し、 α は RT_0 の何倍までを許容するかの数値である。SLA は $0 \leq SLA \leq 1$ の範囲をとる。

2.2 数理モデル

以下に提案するデータセンタのモデルを最適化問題として定式化したものを示す。

Maximize:

$$Profit = \sum f_{revenue}(Job_i, SLA(Job_i)) - f_{powercost}(Power)$$

Output:

$Schedule[H, J]$, Integer Binary ; the Schedule

$GivenRes[J]$, Integer; resources for Job_i

Subject To:

- (1) $\forall_i \in J : \sum_{h \in H} Schedule[h, i] = 1$
- (2) $\forall_h \in H : \sum_{i \in J} GivenRes[i] \cdot Schedule[h, i] \leq Resources(h)$
- (3) $Power = \sum_{h \in H} f_{Pwr}(\sum_{i \text{ runs in } h} GivenRes[i])$
- (4) $\forall_i \in J : f_{MinRes}(i) \leq GivenRes[i] \leq f_{ReqRes}(i)$

$$(5) \forall_i \in J : \hat{RT}_i = f_{RT}(Load_i, ReqRes_i, GivenRes_i)$$

$$(6) \forall_i \in J : SLA(i) = f_{SLA}(\hat{RT}_i, RT_{i,0}, \alpha)$$

2.3 最大化関数

最大化したい目的関数 $Profit$ は、データセンタでの金銭的な利益であり、ジョブを実行することで得られる収益から電力コストを引いた値とする。 $f_{revenue}$ は SLA に応じた収入を計算する関数であり、 $SLA(job) \cdot Revenue(job)$ で定義する。 $f_{powercost}$ 関数は消費電力から電力コストを計算する関数であり、電力使用量・電気料金 で定義する。

2.4 出力

出力はどのホストでジョブを実行するかを組み合わせた $Schedule[H, J]$ と、ジョブの実行にどれだけのリソースを割り当てるかの組み合わせ $GivenRes[J]$ である。

2.5 制約条件

(1) は各ジョブは必ずひとつのホストで実行されることを示す。(2) はあるホストに割り当てた全ジョブのリソースの総量は、そのホストのすべてのリソースの総量以下であることを示す。(3) はジョブに割り当てたリソースの総量によってそのホストでの消費電力が決まることを示している。式中の f_{Pwr} は電力が CPU の使用数に関係する一次関数としてモデル化できることがわかっている [1]。(4) はジョブに割り当てるリソースはジョブの実行に最低限必要なリソース以上、SLA を最大にするリソース以下であることを示し、 f_{MinRes} , f_{ReqRes} は求めたい関数である。(5) はジョブの応答時間は VM 上の Web サーバ、割り当てたリソースによって決まることを示しており f_{RT} は求めたい関数である。(6) は SLA が応答時間と、ユーザの期待する応答時間によって決まることを示し、 f_{SLA} は式 (1) で定義する。

今回はこの関数のうち、(4) の f_{MinRes} , f_{ReqRes} と (5) の f_{RT} の 3 つの関数を機械学習を用いて予測する。

2.6 機械学習を用いたシステムモデルの予測

学習データから、ジョブの CPU 使用量、ホストにおけるジョブの応答時間の 2 つのシステムモデルを作成し、 f_{MinRes} , f_{ReqRes} , f_{RT} の 3 つの関数を作成する。想定環境ではホスト上でユーザの VM が動いており、VM 上では Web サーバが動いているとした。この Web サーバへのリクエスト数などによってジョブの違いが生じる。そこで 4 つの CPU のマシンの上で VM 動かし、その上で Apache を用いて実際の Web サイトを運用し、その Web サイトにきたデータを用いてモデルを作成する。

ジョブの CPU 使用量

CPU 使用量はリクエスト数、リクエストあたりの平均バイト数、リクエストあたりの平均処理時間に関係することがわかった。そこで、機械学習アルゴリズムとして M5P アルゴリズムを用いてモデルを作成した。

結果として、CPU 使用量の平均二乗誤差が 9.9%、標準偏差は 0 もしくは 1CPU となり良い精度で予測することができた。これを f_{MinReq} として定義する。

ホストにおけるジョブの応答時間

応答時間は秒間リクエスト数、リクエストあたりのバイト数、使える帯域幅、リクエストあたりの時間、物理マシンの CPU を使用量、VM に割り当てた CPU 量、VM 上の Web サーバが要求する CPU 量の 7 つに関係することがわかった。そこで機械学習アルゴリズムとして線形回帰を用いて予測を行った。

結果として、平均二乗誤差が $3.0 \cdot 10^{-5}$ であり、標準偏差は $4.0 \cdot 10^{-3}$ となり、良い精度で予測することができた。これを f_{RT} として定義する。SLA の値は 1 以上にはならないため、SLA がちょうど 1 になるような応答時間を f_{RT} により予測し、その場合の CPU 使用量を求めるのを f_{ReqRes} と定義する。

2.7 モデルから作成された最適化問題

機械学習により作成したシステムモデルを用いることで、データセンタをモデル化することができた。作成したモデルでは目的関数と制約条件はすべて 1 次式となり、最適化問題は NP 完全問題である混合整数線形計画問題となった。

そこで作成したシステムモデルを用いて高速に解を求めるヒューリスティック法を提案する。CPU 使用量は f_{ReqRes} 関数を用いて CPU の最大使用量を予測する。First Fit はジョブの CPU 使用量を予測し、最初に見つかった配置可能なホストに配置する方法である。λ-Round Robin は、最初に 1 台以外のホストの電源を切っておき、稼働中のホストの λ% 以上のリソースが使用された時に新しいホストの電源を入れ、予測した CPU 使用量を元に最初に見つかったホストに割り当てる方法である。今回は λ を 30 として行った。Desc BF は予測した CPU 使用量の多いジョブから、利益が最大になるホストに割り当てていく方法である。

3 評価実験

作成したデータセンタモデルの電力コストを変化させた場合の結果の変化と、最適解とヒューリスティックアルゴリズムによって解いた解の差の 2 つについてシミュレーションにより評価を行った。評価は 4 つの CPU のホストが 20 台あるデータセンタを想定している。リソース使用量は既知ではないが、どのようなジョブが来るかはわかっているものとした。シミュレーションはクラウドシミュレータである EEFSSim[1] に、ジョブの集合として実際の Web サービスから作成した Li-BCN[2] の平日 4 日分のデータを用いて行った。

3.1 電力コストを変化させた場合の結果の変化

電力コストを変化させた場合の総消費電力量の変化について図 1 に示す。また、電力コストを変化させた場合の SLA の変化について図 2 に示す。この図から、電力コストを上昇させた場合に総消費電力は減少しているが、SLA

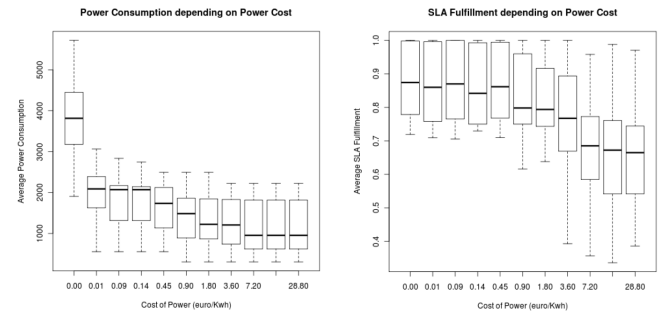


図 1: 電力コストを変えた場合の総消費電力の変化 図 2: 電力コストを変えた場合の SLA の変化

の値の悪化していくことがわかる。これは、利益がマイナスになるとデータセンタの経営が成り立たないため、SLA の値を低下させても利益を確保するようにしているためである。

3.2 最適解とヒューリスティック解の比較

表 1: 最適解とヒューリスティックアルゴリズム

Parameters	Method	Power KWh	Profit	Avg QoS	Used CPUs	Used hosts
0.09 €/Kwh	First Fit	290.2	177.290	0.519	3523	921
	λ-RR	358.9	241.048	0.716	3917	1237
	Desc BF	203.4	321.002	0.866	2230	660
0.17 €/Job	ILP Solver	169.8	326.935	0.878	1568	571
RT ≤ 2 · RT ₀	First Fit	293.2	69.871	0.515	3563	970
	λ-RR	230.7	132.610	0.604	3985	1161
	Desc BF	139.8	255.980	0.818	1504	456
0.17 €/Job	ILP Solver	151.3	270.543	0.862	1483	503

First-Fit、λ-Round Robin、Desc BF (Descending Best-Fit) の 3 つのヒューリスティックアルゴリズムと比較を行った。

電力コストを 0.09 €/kWh、0.45 €/kWh の 2 種類について実験を行った。1 時間あたりの VM の料金は 0.17 €/h とした。

結果を表 1 に示す。これは 10 回配置を行った時の平均値である。最適解が最も良い結果となっているが、Desc BF と結果は変わらない数値になっている。最適解を求めるには約 4 時間かかってしまい現実的ではないが、Desc BF では約 4 秒で結果がわかるため現実的な計算時間で結果を求められることがわかった。

4 おわりに

本稿ではデータセンタでの SLA を考慮しつつ省電力化にするタスクスケジューリングを行うフレームワークを提案した。データセンタのシステムモデルを機械学習を用いて予測することによって、データセンタのモデルを作成し、ヒューリスティックアルゴリズムを用いて解くことで、現実的な時間で最適解に近い結果を求めることができた。

参考文献

- [1] F. Juliá, J. Roldán, R. Nou, O. Fitó, Vaqué, G. Í., and J. Berral. EEFSSim: Energy Efficiency Simulator, 2010.
- [2] J. Berral, R. Gavaldà, and J. Torres. Li-BCN Workload 2010, 2011. <http://www.lsi.upc.edu/jlberral/documents/libcn-2011.pdf>.