講演番号： 206

OpenStack による実プロファイルに基づく 有限資源の動的割り当ての提案

D－23

Proposal of Using OpenStack Dynamic Resource Allocation based on Real Profiles

水村 直人† 高橋 淳二† 戸辺 義人†
Naoto MIZUMURA† Junji TAKAHASHI† Yoshito TOBE†
† 青山学院大学理工学部情報テクノロジー学科
† Department of Integrated Infomation, Aoyama Gakuin University

1. はじめに

本稿では OpenStack を用いてユーザのサーバアクセス数に基づき有限リソースの動的割り当てを行うシステム：ORDAR(Openstack Resource Dynamic Allocation based on Real profile)を提案する．

2. 関連研究

動的リソース割り当てによるシステムの最適化を行う研究は[1]に示されている．[1]は仮想サーバのコストとシステム全体の最適化の整数計画問題を解くことで，コストを抑えつつ性能効率を最大にする仮想サーバの設定を行う手法が提案されている．本研究は仮想空ディスク容量が十分に存在しない環境におけるシステムの最適化を行う点において異なる．

3. システム設計

まずユーザの OpenStack アカウント情報を OS 環境変数から読み取る．次にサーバ数 n_i $(i = 1,2,3\cdots, I)$，フレーバ数 f_k $(k = 1,2,3\cdots, K)$，物理サーバの総 RAM 数と CPU 数を M，C と定義する．また，仮想サーバがサービスを行う上で最低限必要な vRAM 量 vM と VCPU 数 vC を，フレーバ ID = 0 のフレーバの vRAM 量，vCPU 数を入れる．vM と vC の計算式を(3.1)に示す．

$$vM = f_{(ID=0)}.vRAM,\quad vC = f_{(ID=0)}.vCPU \qquad (3.1)$$

フレーバとは仮想サーバを作成する為のスペックテーブルであり，vCPU 数，vRAM 量等を事前に設定する必要がある．その後 j 回目のパケットキャプチャを T 秒間行う．この時プロミスキャスモードでパケットを受信し，TCP/UDP パケットの宛先ポート番号が 80 のものを{ 宛先 IP アドレス：受信回数 }の形で受信情報辞書に格納する．T 秒経過したら辞書内の値を読み取り，サーバ n_i のアクセス回数 A_{ij} から，サーバ n_i へのアクセス頻度 q_{ij} を(3.2)で算出する．

$$q_{ij} = \frac{A_{ij}}{\sum_i A_{ij}} \qquad (3.2)$$

その後，総リソース量から vM，vC とサーバ台数との積に各仮想サーバへのアクセス頻度 q_{ij} を掛けることで，リサイズ後の vRAM 量 m_{ij} と vCPU 数 c_{ij} をサーバ毎に算出する．m_{ij} と c_{ij} の式を(3.3)と(3.4)に示す．

$$m_{ij} = (M - vM \times n_i) \times q_{ij} \qquad (3.3)$$

$$c_{ij} = (C - vC \times n_i) \times q_{ij} \qquad (3.4)$$

算出した m_{ij} と c_{ij} に加えフレーバ ID や仮想ディスク量を指定してフレーバを作成する．その後リサイズ前後の vRAM 量の比較を行い，リソース量が減少するサーバから先に対象サーバとリサイズ後のフレーバの ID を指定することでリサイズを行い，その後 j+1 回目のパケットキャプチャを行う．

4. 評価

3 台の仮想サーバを作成し，ORDAR を用いてリソースの動的割当を行った場合と総リソースを各サーバで等分した場合とで、サーバへ複数の GET リクエストを送信しその応答時間を比較する．仮想サーバでは GET リクエストを受信次第 16 文字のランダムな英数字を生成し python のリストの末尾に追加する作業を 50 万回行う．結果を図 1,図 2 に示す．

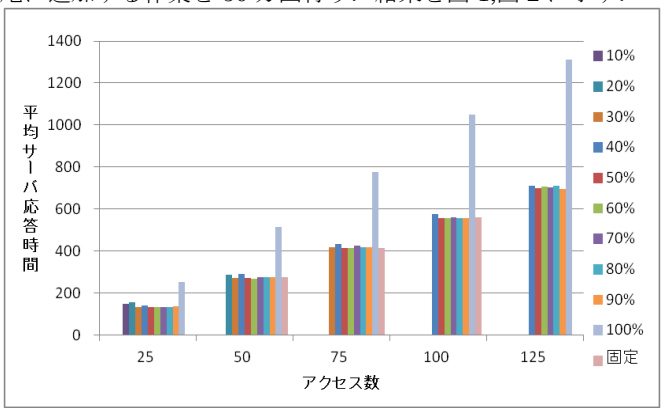

図 1 各リソース時のアクセス数に対する応答時間

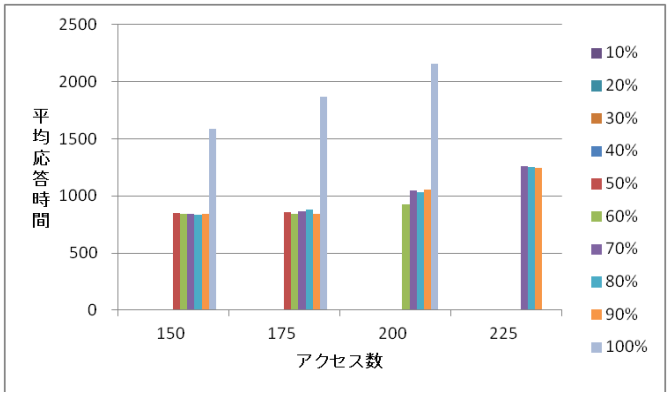

図 2 各リソース時のアクセス数に対する応答時間

5. 結び

本稿では，OpenStack を用いてユーザのサービス利用率に基づいた動的リソース割り当てを行うシステム ORDAR を提案した．今後は使用前の設定の簡略化を進める．

参考文献

[1] P.Kokkinos, et al, “Cost and Utilization Optimization of Amazon EC2 Instances”, *IEEE Sixth International Conference on Cloud Computing*, *pp.518-525*, 2013