Task aware resource allocation for maximizing throughput in cloud environment with heuristic knowledgebase approach

Prof. Bakul Panchal
Department of Computer Engineering L. D. College of Engineering, Ahmedabad (Gujarat-India)

ABSTRACT

In Cloud Environment, Resources available to the client on demand with pay per usage. Higher throughput with minimal execution time can reduce the budget cost for client as well as can never violate Service Level Agreement (SLA). Specific Task should be allocated to proper Virtual Machine can generate efficient result. Our study suggests a better approach to achieve this efficiency using empirical analysis for task by generating knowledgebase heuristic task database. In first step our approach suggest, before allocating a task for execution on Virtual Machine, find out task characteristic, estimate execution time by matching with self-generated heuristic database. During second step find out efficient virtual machine who is capable to do this task with higher throughput in minimum execution time. Better enhancement should be achieved using adaptive threshold value to compare task with heuristic database. This approach can optimize tradeoff between Quality of Service for task and resource utilization.

Keywords
Artificial Neural Network, QSAR, DFT, HUMO, LUMO, NCI

I. INTRODUCTION

Cloud computing is poised to revolutionize computing as a service where IT becomes a computing utility delivered over the Internet. Cloud computing utilizes massively scalable computing resources delivered as a service using Internet technologies, which allows these computational resources to be shared among a vast number of consumers to allow a lower cost of ownership of information technology.

“Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.”

This definition includes cloud architectures, security, and deployment strategies. In particular, five essential elements of cloud computing is clearly articulated:

- **On-demand self-service:**
  Cloud service is available any time on-demand on rent base for particular time slot. Resources are always available and on based on client demand it is available. Resources can be CPU, Network storage, Updated software-service.

- **Broad network access:**
  Cloud service is available on internet. It provides cloud service access to client with various heterogeneous platforms.

- **Resource pooling:**
  A cloud service provider contains huge data centers which use multi-tenancy characteristics of cloud environment to fulfill client on-demand resource requirement. Resource polling is dynamically achieved using virtualization technology. The result of a pool-based model is that physical computing resources become 'invisible' to consumers.

- **Rapid elasticity:**
  Scale up or scale down of resource configuration can be immediately applied because of rapid elasticity property of cloud environment. For consumers, computing resources become immediate rather than persistent: there are no up-front commitment and contract as they can use them to scale up whenever they want, and release them once they finish scaling down. Moreover, resources provisioning appears to be infinite to them, the consumption can rapidly rise in order to meet peak requirement at any time.
• **Measured Service:**
  Although computing resources are pooled and shared by multiple consumers (i.e. multi-tenancy), the cloud infrastructure is able to use appropriate mechanisms to measure the usage of these resources for each individual consumer through its metering capabilities.

• **Collaborative Environment:**
  Cloud environment provide collaborative working environment for team members located at geographically different location.

“Cloud is a parallel and distributed computing system consisting of a collection of inter-connected and virtualised computers that are dynamically provisioned and presented as one or more unified computing resources based on service level agreements established through negotiation between the service provider and consumers.”

**Service Model**
Cloud computing provide virtualized environment in four major cloud service forms as shown in the Figure 1. Services are as explained below.

<table>
<thead>
<tr>
<th>SaaS</th>
<th>Software-as-a-Service</th>
<th>Google Apps, Microsoft Software-Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>PaaS</td>
<td>Platform-as-a-Service</td>
<td>IBM IT Factory, Google AppEngine, Force.com</td>
</tr>
<tr>
<td>IaaS</td>
<td>Infrastructure-as-a-Service</td>
<td>Amazon EC2, IBM Blue Cloud, Sun Grid</td>
</tr>
<tr>
<td>dSaaS</td>
<td>data-Storage-as-a-Service</td>
<td>Nirvanix SDN, Amazon S3, Clavensafe, dNet</td>
</tr>
</tbody>
</table>

**Figure 1:** Cloud computing service layers with examples

**Objectives**
Our objective is to select best VM for optimal CPU Utilization. Virtual machine selection criteria depends on available knowledgebase of similar task executed in past. Knowledgebase contains task execution time, % of CPU utilization for that task. Select appropriate virtual machine to execute same type of task by analyzing available knowledgebase information and generate quick output for user.

The list of objectives that need to be attained includes following:
- Study of Literatures on Cloud computing
- Design of proposed system architecture
- Design proposed algorithm
- Testing different Scenarios
- Results analysis of the proposed algorithm

**II. METHODS AND MATERIAL**

2.1 System Architecture

**Working of System:**
AS shown in above figure 2 when user submit the task, controller handle the incoming task and check whether same type of task has been been executed in past as per the available knowledge base. If controller found the task details, it can estimate the CPU requirement and execution time for incoming task. Based on this data, controller will select Virtual Machine from the available list of Virtual Machine as per the algorithm policy.

**2.2 Proposed Algorithm**
1. User task execution request
   Find out task characteristics Task-Name.
2. Find out similar type of task which was executed in past?
   If No go to step 3 else go to step 5
3. Store incoming task details and go to step 4
   Task-ID, Task-Name, Task-Arrival-Time
5. Find out task execution summary from available record.
6. Retrieve available Virtual Machine details in descending sorted order on basis of current CPU utilization.
7. Match these VMs status with required CPU utilization of task and allocate VM using Following criteria :
   a. If Task_CPU_Requirement < VM_CPU_Available[]
   then Find nearest %CPU availability value with combination of executed task estimated time on that
machine to select VM and schedule task on that Virtual Machine.
b. If Task_CPU_Requirement > VM_CPU_Available[]
   then
   Calculate average time to complete assigned task
   based on estimated task time for CPU availability of
   each VMs and schedule task on VM which has
   minimum value of average time.
8. At end of task execution
   Store Task-End-Time, Average-CPU-Utilization, Required-Exec-Time.

Note: VM-CPU-Usage for every available VMs is being
update at regular interval.

Assumption:
1. Consider SAAS scenario provides specific services
   with private cloud environment.
2. Number of Virtual Machine has same type of
   hardware configuration.
3. All Virtual Machines are always up.

III. RESULT AND DISCUSSION

3.1 Result Analysis

Table 1: Task Execution Scenario

<table>
<thead>
<tr>
<th>Task</th>
<th>VM1</th>
<th>VM2</th>
<th>VM3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>60</td>
<td>50</td>
<td>40</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>60</td>
<td>40</td>
</tr>
<tr>
<td>3</td>
<td>50</td>
<td>40</td>
<td>60</td>
</tr>
<tr>
<td>4</td>
<td>60</td>
<td>50</td>
<td>40</td>
</tr>
<tr>
<td>5</td>
<td>40</td>
<td>50</td>
<td>60</td>
</tr>
<tr>
<td>6</td>
<td>50</td>
<td>60</td>
<td>40</td>
</tr>
</tbody>
</table>

Table 2: User Requested Task History

<table>
<thead>
<tr>
<th>Task</th>
<th>VM1</th>
<th>VM2</th>
<th>VM3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>V1</td>
<td>T1</td>
<td>60</td>
</tr>
<tr>
<td>2</td>
<td>V2</td>
<td>T2</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>V3</td>
<td>T3</td>
<td>40</td>
</tr>
<tr>
<td>4</td>
<td>V4</td>
<td>T4</td>
<td>30</td>
</tr>
<tr>
<td>5</td>
<td>V5</td>
<td>T5</td>
<td>60</td>
</tr>
</tbody>
</table>

3.2 Comparison with Existing Resource Allocation Policy

The effect of the service broker and load balancing
techniques are analysed defining

3.3 Observations

The effect of the service broker and load balancing
techniques are analysed defining the simulation
parameters as follows:

I have described that we can analyze the effect of
various service brokers and load balancer in the Cloud
computing using Cloud Analyst. The output of various
service brokers and load balancer is analyzed to evaluate
the performance of algorithm.

We can observe from the analysing test cases results that
• Threshold based load balancer provides efficient load
  balancing.
• Threshold based load balancer results in less response
time with proper selection of threshold value.
• Threshold based load balancer requires less processing
time.
• Threshold based load balancer is less costly.
• Throttled load balancer with optimized response time service broker results in less response time while both data centers having equal capacity.
• Throttled load balancer with Closest data center service broker results in less response time while closest data centers having different/less capacity then other.
• In both of the above stated cases Throttled load balancer gives better results.

We can also configure different scenarios using more number of Data Centers and VMs to test results.

IV. CONCLUSION AND FUTURE THOUGHTS

Cloud computing is a new and promising paradigm delivering IT services as computing utilities. As Clouds are designed to provide services to external users, providers need to be compensated for sharing their resources and capabilities. In this paper, they have proposed architecture for market-oriented allocation of resources within Clouds. They have also presented a vision for the creation of global Cloud exchange for trading services. Moreover, they have discussed some representative platforms for Cloud computing covering the state-of-the-art.

Data Centres are known to be expensive to operate and they consume huge amounts of electric power. For example, the Google data centre consumes power as much as a city such as San Francisco.

As Clouds are emerging as next-generation data centres and aim to support ubiquitous service-oriented applications, it is important that they are designed to be energy efficient to reduce both their power bill and carbon footprint on the environment.

To achieve this at software systems level, we need to investigate new techniques for allocation of resources to applications depending on quality of service expectations of users and service contracts established between consumers and providers.

As Cloud platforms become ubiquitous, we expect the need for internetworking them to create market-oriented global Cloud exchanges for trading services. Several challenges need to be addressed to realize this vision.

They include: market-maker for bringing service providers and consumers; market registry for publishing and discovering Cloud service providers and their services; clearing houses and brokers for mapping service requests to providers who can meet QoS expectations; and payment management and accounting infrastructure for trading services.

Some of these issues are explored in related paradigms such as Grids and service-oriented computing systems. Hence, rather than competing, these past developments need to be leveraged for advancing Cloud computing. Also, Cloud computing and other related paradigms need to converge so as to produce unified and interoperable platforms for delivering IT services as the 5th utility to individuals, organizations, and corporations.

REFERENCES