Self-adaptive Resource Allocation using Feedback and Ranking in Cloud Computing by Crawler

Kishan M. Patel¹, Asst. Prof. Mohammed Husain Bohara²

¹ Student Computer Science & Engineering Department, Parul Institute of Engineering and Technology, kishan053@gmail.com
² Computer Science & Engineering Department, Parul Institute of Engineering and Technology, mohhamed.jeeranwala@gmail.com

Abstract

Cloud computing is hot cake nowadays and primary factor of cloud computing is virtualization. It deals with on demand dynamic resource allocation for providing reliable and guaranteed services to the consumer. It is based on pay-as-you-use model to public. The tasks of resource allocation are meeting customer demands and application requirements. To get more efficient resources from cloud environment, it required feedback system which provides information of best resources. This report described a new methodology for finding the best resources from the cloud. Proposed approach will get maximum utilization of resource using feedback. Every resources (VMs) have its log of usage and wastage and this log file is consider as feedback. In the proposed approach, the crawler is crawled for log file of all available resources, analyzed log files and discovered the best resource by analyzed log files (feedback) and rank of those resources.

Keywords - resource allocation; virtual machine allocation; ranking; cloud computing; adaptive; crawler

I. INTRODUCTION

In the simplest terms, Cloud Computing means storing and accessing data and programs over the Internet instead of your computer's hard drive. In general, Cloud is a place where provider provides resources and consumer consume it.

Cloud Computing has several research areas but our focus on Resource Allocation in Cloud Computing. In cloud computing various cloud consumers demand variety of services as per their dynamically changing needs. So it is the job of cloud computing to avail all the demanded services to the cloud consumers. But due to the availability of finite resources it is very difficult for cloud providers to provide all the demanded services. From the cloud providers’ perspective cloud resources must be allocated in a fair manner. So, Resource Allocation is a vital issue in Cloud Computing.

Traditional resource allocation techniques are not adequate for cloud computing as it is based on virtualization technology with distributed nature. Cloud computing introduces new challenges for manageable and flexible resource allocation due to heterogeneity in hardware capabilities, workload estimation and characteristics in order to meet Service Level Objectives of the cloud consumers’ applications. The ultimate goal of resource allocation in cloud computing is to utilize resources properly so maximize the profit for cloud providers and to minimize the cost for cloud consumers.

Reference [1] describes so many resource allocation strategies are in world like negotiation based, policy based, loyalty based, agent based, SLA based, etc. In this paper used agent based resource allocation that is Crawler in this case. Proposed system also depends on feedback for better resource allocation.

In existing feedback and ranking resource allocation system [2], generally found that system is dependent on consumer for feedback. There is always possibility of untrustworthy in feedback so for those system need trust calculation for every consumer and it is so complicated. If consumer didn’t provide feedback to resource then it is unfair to that resource.

Proposed system is described a new approach to self-adaptive resource allocation using feedback and ranking system. In this system, Consumers use different resources (VMs) from cloud and resources (VMs) will generate Log file of its usage and wastage and which consider as feedback. Proposed system has Crawler named application that is responsible for crawling, analyzing and ranking to resources and also find best VM for particular task.

Objective of proposed system is to achieve maximum utilization of efficient resources based on its capacity, previous usage and ranking. And also crawling for best VM based on its previously usage that log file can describe.

II. Related Work

Qiang Li, Qinfen Hao, Limin Xiao and Zhourjun Li [4] proposed VM-base architecture for adaptive management of virtualized resources in cloud computing. Authors also designed a resource controller named Adaptive Manager that dynamically adjusts multiple virtualized resource utilization to achieve application Service Level Objective (SLO) using feedback control theory. Adaptive Manager is a multi-input, multi-output (MIMO) resource controller which controls CPU scheduler, memory manager and I/O manager based on feedback mechanism. To periodically measure application performance each Virtual Machine has sensor module which transmits information to the adaptive manager. Authors adopted Kernel based Virtual Machine (KVM) as a tool for infrastructure of virtual machine.

T. R. Gopalkrishnan Nair and Vaidehi M [5] presented a model, named as Ruled Based Resource Allocation (RBRA) which deals with the efficient resource utilization in M-P-S (Memory-Processor-Storage) Matrix Model. Authors say that resource allocation rate should be greater than resource request rate. Major components of the system are: cloud priority manager, cloud resource allocation, virtualization system manager and end result collection. To analyse the performance of the cloud system authors considered the Cloud Efficiency Factor. However, authors also identified other parameters of Cloud System for future work.
Justin Y. Shi, Moussa Taifi and Abdallah Khreishah [6] explored a simple quantitative Timing Model method for cloud resource planning. For the same they considered the estimated resource usage times in steady state. Authors had calculated Speed up for Parallel Resource Planning based on Parallel Matrix Multiplication. To investigate multiple important dimensions of a program’s scalability, authors proposed quantitative application dependent instrumentation method instead of qualitative performance models. Authors had mainly focused on application inter dependencies for cost effective processing.

Yichao Yang, Yanbo Zhou, Lei Liang, Dan He and Zhili Sun [7] focused on efficient data and network (combined) resource utilization for data intensive applications like IPTV. Authors proposed Cloud Infrastructure Service Framework (CISF) to achieve QoS requirements of cloud consumers. They introduced a Service-oriented Resource Broker (SRB) for guaranteed data transmission in cloud computing to discovery, select, reserve and assign data and network resources. Firstly the collected user requirements are given to Resource Requirement Interpreter to produce abstract resource requirement information. This information is then passed to the Resource Discovery Unit to produce list of resource combination which is passed to Resource Combination Ranker to assign priority. Finally Resource Reservation Unit makes coordinated resource reservation to resource gatekeepers through reservation interface.

Chenn-Jung Huang, Chih-Tai Guan, Heng-Ming Chen, Yu-Wu Wang, Shun-Chih Chang, Ching-Yu Li and Chuan-Hsiang Weng [8] proposed resource allocation mechanism based on Support Vector Regression (SVR) and Genetic Algorithm (GA). Authors designed Application service prediction module with Support Vector Regression (SVR) to estimate the number of resource utilization according to the Service Level Agreement (SLA) of each process. Then authors designed global resource allocation module with Genetic Algorithm (GA) to redistribute the resources to the cloud consumers.

Amit Nathani, Sanjay Chaudhary and Gaurav Somani [9] proposed an algorithm in a scheduler named Haizea for resource allocation policies like immediate, best effort, advanced reservation and deadline sensitive. Haizea is a resource lease manager that uses resource leases as resource allocation abstraction and implements these leases by allocating Virtual Machines (VMs). Authors main goal was to minimize resource rejection rate and reshuffle cost in order to provide all the above mentioned resource allocation policies for IaaS cloud. Authors also used two concepts named swapping and backfilling for deadline sensitive resource allocation policy. Authors mainly considered four lease parameters for their experiments: start time, duration, deadline and number of nodes.

Congfeng Jiang, Xianghua Xu, Jilin Zhang, Yunfa Li and Jian Wan [10] raised the effective resource allocation problem based on real time knowledge of workload and performance feedback of running services. Authors had proposed stochastic model of resources in virtualized environments. Authors had also proposed resource allocation and scheduling heuristics algorithms with service level agreement constraints. To improve the effectiveness of the future incoming dynamic workload, the performance of the targeted machine had been considered as a performance feedback mechanism to the source. This feedback mechanism improves the resource allocation method proposed by authors themselves.

Baomin Xu, Chunyuan Zhao, Enzhao Hu and Bin Hu [11] proposed job scheduling algorithm based on Berger Model with dual fairness constraints. Authors had mainly oncentered on fairness of resource allocation and cloud consumers’ satisfaction to the provided services. Based on parameters like completion time and bandwidth, cloud consumers’ tasks had been classified. According to the characteristics and preferences of tasks, resources were assigned to the cloud consumers. Authors implemented their algorithm on CloudSim toolkit and compared with optimal completion time algorithm. Results show that algorithm based on Berger Model is better.

Weiwei Lina, James Z. Wangeb, Chen Liangc and Deyu Qia [12] proposed a threshold based dynamic resource allocation scheme for cloud computing. Authors mainly focused on application level resource allocation instead of mapping between physical resources and virtual resources for better utilization of resources. A threshold is used to optimize the decision of resource reallocation. The proposed algorithm consists of two procedures: Datacenter-resides at the datacenters central computer and Broker-runs on user’s machine with the application. Both procedures interact with each other for dynamic resource allocation. The proposed algorithm is implemented by using CloudSim Toolkit.

Rosy Aoun, Elias A. Doumith and Maurice Gagnairein [13] proposed a model named as Mixed Integer Linear Program (MILP) for resource provisioning for enriched services in cloud environment. Authors stated that several basic services offered at IaaS level can be arranged together by the cloud providers for providing sophisticated services to the cloud consumers. Two original services, distributed data storage and multicast data transfer are jointly considered in addition to the traditional computing, centralized storage and point to point data transfer services. However, authors have considered the impact of four types of services: computing, storage, point to point data transfer and point to multipoint data transfer. The numerical results were given by considering 18-node backbone network.

### III. PROPOSED SYSTEM

Proposed system is described a new approach to self-adaptive resource allocation using feedback and ranking system. In this system, Consumers use different resources (VMs) from cloud and resources (VMs) will generate Log file of its usage and wastage and which consider as feedback.

In existing system [2], feedback coming from consumer to central system and then it get analyzed, But, in proposed approach,

- First to crawl for all log files.
- Analyze all log files and that will be feedback to system.
- Calculate completion time for all VMs based on feedback.
- Do ranking of all VMs based on feedback.
- Find best VM having highest rank and minimum completion time.

Crawling means to search for all resources’ log file. This is done with help of resource map that have track of all...
resources (VMs). And after that it will get analyzed. Ranking is one of the way, to indicate the system how VM is efficiently used by Consumer. This rank provides us to identify its accuracy and usability. Ranking may be integer assign to Resources based on performance of resources. Like Rank “1” is highest rank and “8” is lowest. And it’s discovered by crawler.

Figure 1. Flow diagram of proposed system

Above figure illustrate workflow of the proposed system. In proposed system when system get initialized, system will create log files for all available VMs and create resource map that having location of all log files. In next step it will allocate all task to VM as per First Come First Serve method then it check that any of submitted task finished its execution and if it’s true then it will search all log files as per resource map and analyzing them and do ranking to all VMs.

After that it will find first waited task on all VM and find VM that having best rank and minimum total completion time for that first waited task. This process will finished when all task is finished their execution. Here equation for finding total completion time of task on VM is below.

\[ CT = \frac{\text{LengthOfTask(MI)}}{\text{VMCapacity(MIPS)}} + \text{WaitingTime} \]

Here this system adapt itself because it uses log file as feedback to take decision and that log file is change run time as per VMs’ behavior.

Here rank describes performance of VM and used to take decision of resource allocation.

IV. EXPERIMENTAL RESULTS

This proposed system is implemented in CloudSim toolkit for simulation purpose. Experimental setup is described below.

Here for experiment we take three VMs having below speed in MIPS (million instruction per second).

<table>
<thead>
<tr>
<th>VM Id</th>
<th>Speed (MIPS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>1</td>
<td>500</td>
</tr>
<tr>
<td>2</td>
<td>1000</td>
</tr>
</tbody>
</table>

Tasks (Cloudlets) are taken heterogeneous in length, here we take 20 tasks having different length.

Here we take “Improved min-min algorithm for resource allocation method” for comparison. Resource Allocation with improved Min-Min Algorithm \(^{(3)}\) executes Min-Min in the first round. In the second round it chooses the resources that produce makespan and find the tasks which run on that resource. Find the light load resource. Find the tasks next maximum completion time of every task which run on Rj and algorithm identifies the resource with heavily load by choosing the resources produce makespan by Min-Min and maximum completion of task is compared with the makespan that is produced by Min-Min and also check the new MCT of Rk is less than makespan on which task will be swapped if condition is fulfilled then the task is rescheduled on that resource which fill the condition and produce it and then update the ready time of the both resource. Otherwise the next maximum time of tasks are selected and again repeats the steps. The process stops if all resources and all tasks assigned in those for rescheduling thus the tasks possible resources are rescheduled on the resource which have minimum load.

Figure 2. Comparison of total completion time
The proposed system will take feedback from log file that is automatically generated by system so no need to trust calculation. Proposed system also consider all resources and use log file as feedback. Proposed system will use application named crawler that is responsible for analyze log file and provide rank to resources. In future proposed system will implemented in real environment.

**Figure 2** shows total completion time for all cloudlets for proposed and Improved Min-Min method.

**Figure 3. Comparison of utilization of VMs**

Figure 3 describes that the proposed system increases the utilization of VM that has higher capacity and lower the utilization that have low capacity.

**V. CONCLUSION**

This paper describes the weightiest technique used for Resource Allocation using feedback and ranking. In existing system[2], generally feedback is given by consumers so there is always an issue for untrustworthy. Existing resource allocation system based on feedback mechanism only consider those resources that having feedback from consumer. Existing system also need trust calculation for identify malicious feedback. But proposed system will take feedback from log file that is auto generated by system so no need to trust calculation. Proposed system also consider all resources and use log file as feedback. Proposed system will use application named crawler that is responsible for analyze log file and provide rank to resources. In future proposed system will implemented in real environment.

**REFERENCES**


