Development of a Smart Job Allocation Model for a Cloud Service Provider

Sourav Banerjee
Department of Computer Science and Engineering
Kalyani Govt. Engineering College
Kalyani, Nadia, West Bengal, 741235, India
souravacademia@gmail.com

Mainak Adhikari, Utpal Biswas
Department of Computer Science and Engineering
University of Kalyani
Kalyani, Nadia. West Bengal 741235, India
onlinemainak@yahoo.com, utpal01in@yahoo.com

***Abstract*— Cloud Computing is known as a provider of dynamic services. It utilizes a very large, scalable and virtualized resource over the Internet. So many industries have joined this bandwagon nowadays. One of the major research issues is to maintain good Quality of Service (QoS) of a Cloud Service Provider (CSP). The QoS encompasses different parameters, like, smart job allocation strategy, efficient load balancing, response time optimization, reduction in wastage of bandwidth, accountability of the overall system, etc. The efficient allocation strategy of the independent computational jobs among different Virtual Machines (VM) in a Datacenter (DC) is a distinguishable challenge in the Cloud Computing domain and finding out an optimal job allocation strategy guided by a good scheduling heuristic for such an environment is an NP-complete problem. So different heuristic approaches may be used for better result. The related works with other NP- complete problems have shown that solutions guided by heuristic approaches can often be improved by applying local scheduling procedure for allocating independent jobs in the virtual machines (VM) inside a Datacenter (DC), which, when combined with fast construction heuristics, can find out the shorter schedules on benchmark problems than the other solution methodologies found in different literatures, and in significant less time. This paper highlights a smart job allocation strategy for a CSP by applying Round-Robin (RR) scheduling policy.**

Keywords*— *Quality of Service (QoS), Cloud Service Provider (CSP), Virtual Machine (VM), Datacenter (DC).

I. Introduction

Cloud Computing [1] [2] is nothing but a fusion of virtualization and distributed computing; where immensely scalable IT related capabilities are provided to the multiple external customers "as a service" using internet technologies. The cloud is a metaphor for the Internet and is an abstraction for the complex infrastructure that it hides. It is a process of delivering computing "as-a-service" rather than a product, whereby the shared resources, software, and information are provided to the users and the other devices as a utility over the network. The Cloud environment [3] allows the users to use the applications without installing any specific software on local machine and access their personal data or information from a storage located at any remote machine [5] [8] with the help of Internet. The Cloud [4] provides the illusion of infinite computing resources; its elasticity frees the applications designers from the confinement of a single system. The Cloud computing [6] provides computation, software applications, data access, data management and huge storage without requiring the knowledge of the location of the resources, i.e., location transparency and the other details of the computing infrastructure. In this domain, the background activities like job allocation, load sharing, load balancing, process migration, distributed shared memory access is completely abstracted from the user's purview. Here, the End users or the customers can access the cloud based applications [7] as well as infrastructure through logging in to a Cloud interface. The Cloud application providers strive to give the same or better service and performance than if the software programs were installed locally on end-user machines. The effective scheduling [11] of independent jobs in a heterogeneous environment like cloud environment is an important issue. To make the cloud services effective in that environment, one of its requirements is to provide an efficient job allocation strategy. The job allocation policy is responsible for mapping jobs; submitted to the cloud environment onto available resources in such a way that it will improve the overall performance. There are so many job allocation mechanisms applied by resources manager in distributed computing to allocate the jobs to the different resources optimally and the job scheduling is a part of job allocation procedure. A proper scheduling policy may lead to a good allocation of jobs to the suitable resources or VMs and that may eventually lead to improve the overall system performance. While some of these algorithms try to optimize the total completion time that may eventually lead to the make span [14] improvement.

There have been many algorithms used to schedule jobs on their desired resources; some of these algorithms are used in grid computing [15] which is a large scale distributed system concerned with resource sharing and coordination for problem solving. Three well known examples of such algorithms intended to be applied in cloud computing environment are Max-min, Min-min and RASA [9]. Each of these algorithms estimates the completion and execution time of each submitted job on each available resource.

This paper highlights a novel job allocation procedure in Cloud environment by applying ETC (expected time to compute) matrix [12]. The remaining part of this paper is organized as follows: section 2 represents the related works and different classical task scheduling algorithms. In section 3,

978-1-4799-3264-1/14/$31.00 ©2014 IEEE

the proposed model for job allocation in a cloud data center is described with an example to illustrate the prominence of the proposed model and section 4 concludes this proposed work and represents the future scope of this whole work.

II. RELATED WORK

In cloud computing, many different Job scheduling algorithms are used. Today, the researchers are attempting to defend the huge challenge of proper allocation of jobs to the suitable resources as well as VMs following the existing job scheduling algorithms that are compatible and applicable for Cloud like environment. The job scheduling is a process of proper allocation of the jobs to their intended resources [10]. In cloud computing, the scheduling is a problem of allocating a set of submitted jobs from different users on a set of computing resources to minimize the overall completion time for a specific job of a system. There are so many parameters associated with the scheduling challenges such as, load balancing, system throughput, service reliability, service cost, system utilization, network bandwidth, response time etc. A good job allocation strategy may lead to the enhancement of QoS of the Cloud Service Provider. The Expected Time to Compute [12] [13] parameter plays a vital role in the resource selection procedure in a heterogeneous environment. The ETC matrix contains (n x m) entries, where n is the number of jobs and m is the number of processors as well as VMs which is shown in Table. 1. The expected running time of each individual job on each VM is stored in the ETC matrix. The column in ETC matrix represents resource as well as VMs and the row contains the expected computation time for a single job on each of the available VMs.

TABLE I. ETC MATRIX

Jobs	VM_1	VM_2
J_1	5	3
J_2	2	4
J_3	8	12
J_4	3	9

The job scheduling is a decision making procedure through which the best match between jobs and resources are estimated. So, the job scheduling is an NP-complete problem [10] [19]. Few heuristics are mentioned here those are related to job scheduling in Cloud like heterogeneous environment [12].

OLB Opportunistic Load Balancing assigns each job in arbitrary order to the processor with the shortest schedule, irrespective of the ETC on that processor. OLB is intended to try to balance the processors, but it does not take execution times into account it finds rather poor solutions.

MET Minimum Execution Time assigns each job in arbitrary order to the processor on which it is expected to be executed fastest, regardless of the current load on that processor. MET tries to find good job-processor pairings, but because it does not consider the current load on a processor it will often cause load imbalance between the processors.

MCT Minimum Completion Time assigns each job in arbitrary order to the processor with the minimum expected completion time for the job. The completion time of a job j on a processor p is simply the ETC of j on p added to p's current schedule length. This is a much more successful heuristic as both execution times and processor loads are considered.

Min-min establishes the minimum completion time for every unscheduled job (in the same way as MCT), and then assigns the job with the minimum completion time (hence Min-min) to the processor which offers it this time. Min-min uses the same intuition as MCT, but because it considers the minimum completion time for all jobs at each iteration it can schedule the job that will increase the overall make span the least, which helps to balance the processors better than MCT.

Max-min is very similar to Min-min. Again the minimum completion time for each job is established, but the job with the maximum minimum completion time is assigned to the corresponding processor. Max-min is based on the intuition that it is good to schedule larger jobs earlier on so they won't 'stick out' at the end causing a load imbalance. However experimentation shows that Max-min cannot beat Min-min on any of the test problems used here.

III. PROPOSED WORK

The previous works have not considered the selection of a suitable processor as well as a VM for a particular job in the system. This paper has considered the VM affinity for a particular job at a certain time stamp with respect to the estimated computing time on a certain VM. The proposed model and its working principle is described below with a suitable example. Few important components of the proposed work are mentioned below:

Cloud User (CU), Cloud Service Provider (CSP), Data Center (DC), Data Center Broker (DCB), Servers as physical resources, VMs, Virtual Center Manager (VCM), a Virtual Machine Management Layer (VMML), Job Scheduler (JS), Global Queue (GQ), Requested Queue (RQ), Local Queue (LQ).

Our proposed model is scalable. But for the sake of simplicity we have considered here two servers and four VMs in this example. Now the above mentioned modules and their working methodologies are described chronologically-

Cloud User (CU) [7] – This module sends a batch of jobs to get executed in CSP through via Internet from different parts of the globe.

Cloud Service Provider (CSP) [7] - This module consists of different sub-modules, shown in fig. 1. Initially the requested jobs are stored in GQ and then they are scheduled to different DCs using help of different policies adopted in DCB module.

Data Center Broker (DCB) [8] - It schedules the jobs arriving from GQ depending upon the policy of system administrator, to the suitable DC.

Data Center (DC) [8] - DC module contains several servers as well as resources, which executes the jobs. DC schedules the jobs which are initially stored in RQ to the different servers as well as physical resources using VCM, shown in fig.3.

Virtual Center Manager (VCM) [16] - VCM module provides a convenient centralized management cockpit to the data center. It aggregates the underlying physical resources from the multiple servers and provides a central collection of simple and flexible resources for the system administrator when provisioning virtual machines in the virtual environment. VCM schedules and allocate the jobs from RQ to suitable Servers.

Server- Server is the most important component of the DC, which contains a number of virtual machines for delivering concurrent heterogeneous requests.

Virtual Machine Management Layer (VMML) [16] - This module resides in the virtual layer of each server that monitors the load of each VM inside a server and schedules and manages the jobs onto the VMs in a particular server. It stores the job initially to the Local Queues (LQ).

Job Scheduler (JS) [7] - JS allocates the jobs from LQ to VM depending upon the policy applied by system administrator.

We have proposed a matrix, i.e., MTC (Minimum Time to Compute) matrix, which is basically derived from ETC matrix. By applying the MTC matrix; VMML finds out the suitable VMs for the requested jobs and then allocate to the proper VM.

The proposed model is described below with a suitable example.

A. Proposed Model

Initially, the CU sends a batch of request to the CSP, which is stored in the GQ. Then, the DCB selects the DC which is suitable for serving the requests within an optimal time period. The basic block diagram of our proposed model is shown in fig. 1. The work flow model is described in fig. 2. In fig. 3 the several components inside a datacenter is mentioned and fig. 4 contains different components and job allocation strategy inside a server. After selecting the suitable DC, the jobs are shifted to the RQ inside each DC. Then the VCM, in fig. 3, allocates the jobs to the appropriate server in a particular datacenter with the help of the VM allocation policy; applied in the Distributed Resource Scheduler (DRS) [16]. In this example, the server contains a maximum of four VMs and a VMML. VMML schedules the jobs among these VMs based upon its availability and the VM affinity with the help of the developed MTC matrix. This paper mainly describes the procedure of job allocation to a suitable VM by the VMML. After the completion of the VM allocation procedure, the jobs are stored in the respective LQs, see fig. 4 and then the VMs start execution for accomplishing the jobs according to the scheduling policy, implemented by the system administrator. Here, the Round-Robin scheduling policy has been applied that may overcome the starvation. The time quantum can be customized by the system administrator. Now we are going to describe the operation of VMML and the job allocation to the VMs with the help of a suitable example in the next subsection.

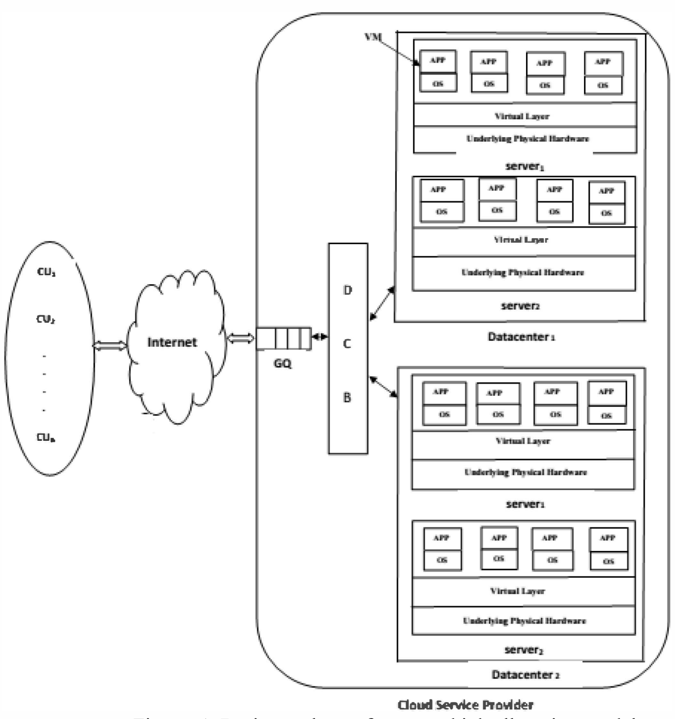
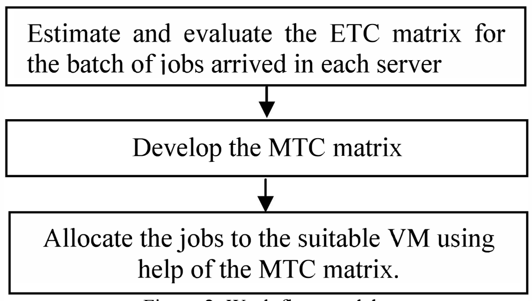


Figure 1 Basic topology of proposed job allocation model

Now the workflow model is described below



Estimate and evaluate the ETC matrix for the batch of jobs arrived in each server
Develop the MTC matrix
Allocate the jobs to the suitable VM using help of the MTC matrix.

Figure 2. Work flow model

The different components of scheduling models are described below.

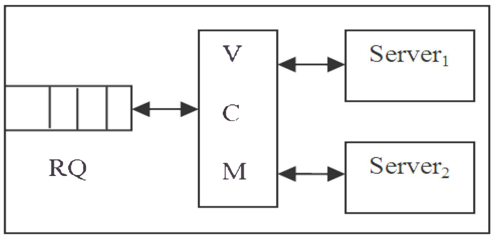


Figure 3. Basic scheduling operation inside a Datacenter.

The scheduling strategy is defined below-

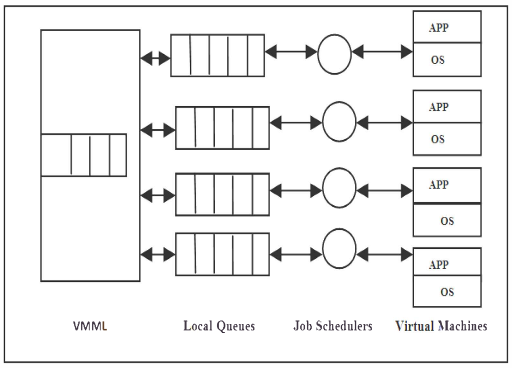


Figure 4. Allocation to different VMs inside a Server

B. Example

Here in this example we have considered a single datacenter, a server and four VMs. The VMML has the capacity of containing a batch of 20 jobs and each server may create a maximum of four VM. So the VMML first finds out the expected time to compute for each job and constructs the ETC matrix for the jobs using the help of the calculation of estimated execution time [17] of jobs. The ETC matrix of 20 tasks is shown in Table 2 and then the MTC matrix is generated; shown in Table 3. By applying the result of MTC matrix, the jobs will be dispatched to the LQ of intended VM and finally scheduled to the VM following Round Robin Scheduling [7] strategy applied in Job scheduler module, shown in fig. 4.

TABLE 2. ETC MATRIX OF JOBS

Jobs	**VM_1**	**VM_2**	**VM_3**	**VM_4**
J_1	4	1	3	8
J_2	2	4	3	5
J_3	7	6	5	4
J_4	7	5	4	8
J_5	8	10	12	5
J_6	7	2	6	5
J_7	8	6	5	7
J_8	8	12	10	11
J_9	10	8	3	5
J_{10}	10	15	12	6
J_{11}	3	2	4	8
J_{12}	8	10	8	7
J_{13}	5	15	6	7
J_{14}	7	5	10	15
J_{15}	10	12	7	15
J_{16}	10	12	15	8
J_{17}	7	8	10	12
J_{18}	4	3	5	6
J_{19}	7	8	6	8
J_{20}	4	6	6	8

This section describes the procedure of finding out the MTC matrix for jobs. The table 3 shows the MTC matrix of jobs.

TABLE 3. MTC MATRIX OF JOBS

Jobs	**VM ID**	**Burst Time**
J_1	VM_2	1
J_2	VM_1	2
J_3	VM_4	4
J_4	VM_3	4
J_5	VM_4	5
J_6	VM_2	2
J_7	VM_3	5
J_8	VM_1	8
J_9	VM_3	3
J_{10}	VM_4	6
J_{11}	VM_2	2
J_{12}	VM_4	7
J_{13}	VM_1	5
J_{14}	VM_2	5
J_{15}	VM_3	7
J_{16}	VM_4	8
J_{17}	VM_1	7
J_{18}	VM_2	3
J_{19}	VM_3	6
J_{20}	VM_1	4

The average waiting time and response time is calculated here, by applying Round Robin scheduling policy with a time quantum of 2ms.

MTC matrix for VM_1:

Jobs	**Burst Time**
J_2	2
J_8	8
J_{13}	5
J_{17}	7
J_{20}	4

Response time for each job in VM_1:

Jobs	**Response Time**
J_2	0
J_8	2
J_{13}	4
J_{17}	6
J_{20}	8

Average Response Time: - (0+2+4+6+8)/5 = 4.0

Waiting Time of each job in VM_1:

Jobs	**Waiting Time**
J_2	0
J_8	17
J_{13}	16
J_{17}	19
J_{20}	14

Average Waiting Time: - (0+17+16+19+14)/5 = 13.2

MTC matrix for VM_2:

Jobs	**Burst Time**
J_1	1
J_6	2
J_{11}	2
J_{14}	5
J_{18}	3

Response Time for each job in VM_2:

Jobs	Response Time
J_1	0
J_6	1
J_{11}	3
J_{14}	5
J_{18}	7

Average Response Time: - (0+1+3+5+7)/5 = 3.2

Waiting Time of each job in VM_2:

Jobs	Waiting Time
J_1	0
J_6	1
J_{11}	3
J_{14}	8
J_{18}	9

Average Waiting Time: - (0+1+3+8+9)/5 = 4.2

MTC matrix for VM_3:

Jobs	Burst Time
J_4	4
J_7	5
J_9	3
J_{15}	7
J_{19}	6

Response Time for each job in VM_3:

Jobs	Response Time
J_4	0
J_7	2
J_9	4
J_{15}	6
J_{19}	8

Average Response Time: - (0+2+4+6+8)/5 = 4.0

Waiting Time of each job in VM_3:

Jobs	Waiting Time
J_4	8
J_7	15
J_9	12
J_{15}	19
J_{19}	19

Average Waiting Time: - (8+15+12+19+19)/5 = 14.6

MTC matrix for VM_4:

Jobs	Burst Time
J_3	4
J_5	5
J_{10}	6
J_{12}	7
J_{16}	8

Response Time for each job in VM_4:

Jobs	Response Time
J_3	0
J_5	2
J_{10}	4
J_{12}	6
J_{16}	8

Average Response Time: - (0+2+4+6+8)/5 = 4.0

Waiting Time of each job in VM_4:

Jobs	Waiting Time
J_3	8
J_5	16
J_{10}	17
J_{12}	22
J_{16}	22

Average Waiting Time: - (8+16+17+22+22)/5 = 17.0

Finally, Table 4, depicts the waiting time and response time for each VM.

TABLE 4: RESPONSE TIME AND WAITING TIME FOR EACH VM

VM ID	Average Response Time	Average Waiting Time
VM_1	4.0	13.2
VM_2	3.2	4.2
VM_3	4.0	14.6
VM_4	4.0	17.0

IV. CONCLUSION AND FUTURE WORK

In this paper, we have discussed about the job allocation to the different VMs inside a Cloud Data Center with the help of the ETC matrix, MTC matrix and a classical scheduling policy. It is a part of our whole work. The ETC and MTC matrices help to map the jobs to the appropriate VMs, which will reduce the overall response time and waiting time of the jobs.

We are now trying to develop our system in such a way that will help the VMML module to identify intelligently that how many jobs may be served by a single VM at a certain time stamp. This will improve the overall makespan [18] of the Cloud Service Provider. We are presently investigating on the VM migration criteria which allow migration of VM from one server to another, i.e., live migration to the other server with the help of the Vmotion [16] Distributed Service in the Cloud environment. Another two major challenges are to investigate and improve the different allocation policies included in DCB and VCM module to achieve better Quality of Service.

ACKNOWLEDGMENT

This work is partially supported by DST-PURSE scheme of University of Kalyani.

REFERENCES

[1] Kaiqi Xiong, Harry Perros "Service Performance and Analysis in Cloud Computing" 978-0-7695- 3708-5/09 $25.00 © 2009 IEEE page- 693-700.

[2] Borja Sotomayor, Rubén S. Montero and Ignacio M. Llorente, Ian Foster "Virtual Infrastructure Management in Private and Hybrid Clouds" 1089-7801/09/$26.00 © 2009 IEEE.

[3]. M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia "A Berkeley View of Cloud computing" Technical Report No. UCB/EECS-2009-28, University of California at Berkley, USA, Feb. 10, 2009.

[4] Francesco Maria Aymerich, Gianni Fenu1, Simone Surcis "An Approach to a Cloud Computing Network" 978-1-4244-2624- 9/08/$25.00 ©2008 IEEE 113 page 113-118.

[5] Xu Lei, Xin Zhe, Ma Shaowu, Tang Xiongyan "Cloud Computing and Services Platform Construction of Telecom Operator" Broadband Network & Multimedia Technology, 2009. IC-BNMT '09. 2nd IEEE International Conference on Digital Object Identifier, pp. 864 – 867.

[6] Luqun Li "An Optimistic Differentiated Service Job Scheduling System for Cloud Computing Service Users and Providers" 2009 Third International Conference on Multimedia and Ubiquitous Engineering page-295-299.

[7] Mainak Adhikari, Sourav Banerjee, Utpal Biswas "Smart Task Assignment Model for Cloud Service Provider" Special Issue of International Journal of Computer Applications (0975 – 8887) on Advanced Computing and Communication Technologies for HPC Applications - ACCTHPCA, June 2012.

[8] Rajkumar Buyya, Rajiv Ranjan, Rodrigo N. Calheiro "Modeling and Simulation of scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities".

[9] Saeed Parsa and Reza Entezari-Maleki , "RASA: A New Grid Task Scheduling Algorithm", International Journal of Digital Content Technology and its Applications, Vol. 3, pp. 91-99, 2009.

[10] P. Brucker, "Scheduling Algorithms", Fifth Edition, Springer Press, 2007.

[11] D.I. George Amalarethinam and P. Muthulakshmi, "An Overview of the scheduling policies and algorithms in Grid Computing ", International Journal of Research and Reviews in Computer Science, Vol. 2, No. 2, pp. 280-294, 2011

[12] Graham Ritchie and John Levine, "A fast, effective local search for scheduling independent jobs in heterogeneous computing environments".

[13] Tracy D. Braun, Howard Jay Siegel, and Noah Beck, "A Comparison of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems", Journal of Parallel and Distributed Computing 61, 810_837 (2001).

[14] El-Sayed T. El-kenawy, Ali Ibraheem El-Desoky, Mohamed F. Al-rahamawy, "Extended Max-Min Scheduling Using Petri Net and Load Balancing", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-4, September 2012.

[15] L. Mohammad Khanli, and M. Analoui, "Resource Scheduling in Desktop Grid by Grid-JQA," The 3rd International Conference on Grid and Pervasive Computing, IEEE, 2008.

[16] White Paper- VMware Infrastructure Architecture Overview, VMware.

[17] Jaehyung Yang, Ashfaq Khokhar, Sohail Sheikht, Arif Ghafoor "Estimating Execution Time For Parallel Tasks in Heterogeneous Processing (HP) Environment", 0-8186-5592-5194 $3.00 Q 1994 IEEE

[18] D.I. George Amalarethinam, F.Kurus Malai Selvi, "A Minimum Makespan Grid Workflow Scheduling Algorithm", 978-1-4577-1583-9/ 12/ $26.00 © 2012 IEEE

[19] M. R. Garey and D. S. Johnson. Computers and Intractabzlzty: A Guade to the Theory of NP- Completeness. W. H. Freeman and Company, New York, 1979.