An Efficient Resource Management in Cloud Computing

Bashir Yusuf Bichi, Tuncay Ercan

Yasar University, Department of Computer Engineering
Izmir, Turkey

Abstract— Today the field of cloud computing is gaining more recognition to the public and sometimes refers to as public utility which allows the user to focus on his actual business without wasting valuable time on installation and maintenance of other important devices, as the cloud service providers takes full responsibility of installation and maintenance of such devices. Cloud computing is meant to scale up or down, enhance quality of service (QoS), cost effective and also simplified user interface so that the user can appreciate the benefits behind cloud computing idea. In this paper we develop a new framework based on Max-Min algorithm which aims at distributing load to a set of virtual resource system using a particular balancing technique.

Index Terms— Load balancing, Max-min algorithm, Makespan, Min-min algorithm, Task Scheduling, Resource Allocation.

1 INTRODUCTION

The management of resources requires putting a limited access to the pool of shared resources. No matter what kind of resources you are dealing with, it also controls the status of current resource consumption. Resources in Information Communications Technologies (ICT) are the fundamental elements like hardware part of the computer systems, data communications and computer networks, operating system and software applications. Since the number of these resources is limited, it is important to restrict access to some of them. So, we can ensure an SLA (Service Level Agreement) between the customers who are requesting resources and providers who are the owners of the systems.

Over the past decade, available ICT systems used in the development of internet and distributed systems gave computer users an opportunity to access and exploit the different resources contained in those systems. Recently, cloud computing is seen as new term in the area of computing, which is a technological adaptation of distributed computing and internet. The main idea behind cloud computing is to allow customer access to computing resources through the web services in an efficient way. Cloud based network services are provided by virtual hardware, which can scale up or down according to the incoming users' requests. Cloud computing provides different types of services like software as a service (SaaS), infrastructure as a service (IaaS) and platform as a service (PaaS). However, it presents a number of management challenges, because customers of these cloud services should have to integrate with the architecture defined by the cloud provider, using its specific parameters for working with cloud components. As the clients in the cloud ecosystem are increasing, it's good to find an efficient way to handle the clients' task by providing an effective and efficient way to balance the different task onto a given virtual resource.

The other part of this paper is organized as follows: Section 2 discussed on some previous studies related to task scheduling algorithms. In Section 3 we introduce some concept with regard to the technique for task scheduling and resource allocation in cloud computing, section 4 discuss on the issues pertaining load balancing, mathematical formulation, and the proposed algorithm. In section 5, mathematical simulation and results are discussed, and lastly concluding remarks are given in Section 6.

1. RELATED WORK

As the use of cloud environment is increasing, task scheduling needs to be more scalable to the user demands. [1][2] Uses a technique known as improved max-min algorithm and enhance max-min algorithm respectively with the aim of distributing the load among the available resources. [1] is a modification of [2] in which both uses the max-min task scheduling algorithm. This paper employs the technique in [1] to propose another algorithm that will help in balancing load across the virtual resources and to allow for scalability when handing task with the aim of improving the performance of the system. Load balancing over resources in the cloud environment is used to achieve minimum load when using resources, different methods are use to achieved such balance as stated in [7, 8, and 9]. Based on these methods, we take interest in max-min algorithm and shows how load can be balanced across different resources in the cloud environment.

2. CLOUD COMPUTING AND TASK SCHEDULING

Cloud computing fall in parallel and distributed computing, which is a collection of computers that are interconnected and virtualized as one computing resources, and the client get access to the resources following agreement between the provider and the client otherwise known as service level agreement (SLA) [11].

As mention previously cloud computing offers software, platform, and infrastructure as a service respectively. The software as a service includes providing software such as Mail (e.g. Gmail, Yahoo mail), social network sites, Google drive, and so on, to the customers or clients. The infrastructure as a service deals with VM, storage, network, load balancer and so on as a service to the client and lastly the platform as a service deals with database like sql, oracle, web services, runtime (e.g. java) and so on as a service to the client. The clients get access to these services through various devices as shown in the figure below [3] [4].

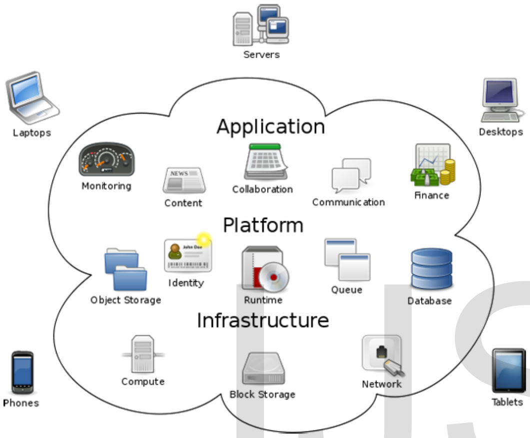

Fig. 1 Service types in Cloud Computing

Task scheduling is a well known concept as it is a vital aspect in cloud computing. It allows for scheduling virtual resources over the cloud to keep a balance load across the resources [47] as indicated in the figure below (fig. 2.)

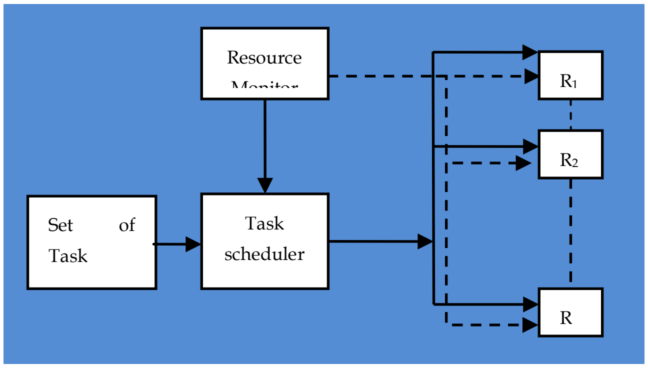

Fig. 2 Task Scheduling for Different Resources

3. Virtual Resources and Allocation

As it is shown in Fig. 2 above, the users send task to the cloud environment with different requirement to the cloud service providers. The requirement can be tasks with different set of data size and processing power, the task scheduler will then match the tasks with available resources (virtual resources) that are available.

Resources in cloud computing cover all useful entities which can be use through the cloud platform. These resources include storage, memory, network bandwidth, and virtual machine [3]. The resources can be virtualized and provisioned from the existing physical resources in the cloud environment. The parameters that are virtualized include; the CPU, memory, disk etc. The provisioning can be done by mapping these virtualized resources to their corresponding physical ones. Resource allocation in cloud computing is all about assigning available resources to a needing cloud application. Dynamic resource management is seen as a very active research area in the field of cloud computing. The cloud computing resources costs vary depending upon the type of configuration for using such resources. Therefore an efficient use of these resources is considered as a prime interest for both the customer/client and the cloud provider. Resource allocation in cloud computing takes place in two levels [5];

- If an application is uploaded to the cloud, a load balancer assigns the requested instances to a physical machine to balance the computational load of multiple applications across physical computers.
- If an application receives multiple incoming requests, such requests are assigned to a specific application instance to balance the computational load across a set of virtual instances of some applications

Resource allocation exhibits some benefits irrespective of the organization size or business market. It also have some limitations, below are some set of advantages and limitations of resource allocation [6];

Advantages

- Users do not have to install software or hardware to access the applications, develop application and to host the application over the net.
- No limitation of place and medium, application and data can be reached anywhere in the world and on any system.
- Users do not need to purchase the hardware and software systems.
- Cloud providers can share their resources over the internet during scarcity of resources.

Limitations

- Users do not have control over the resources since they rent the resources from a remote server.
- Migration issue occurs when a user decides to switch to other providers.
- In public cloud, security is major issue as clients/customers are concerned that their data could be hacked.
- Peripherals devices like printer and scanner might not work with cloud as many require software to be installed locally.

4. Load Balancing and Task Scheduling Algorithm

Load balancing is a technique used to distribute processing load (i.e. large processing load) to smaller processing nodes (i.e. resources) to enhance the overall performance within the system in a distributed environment as shown in fig.2 above. The idea of load balancing is to avoid loading up a resource during task scheduling so that all the resources will be allocated with a task evenly across a given virtual environment. Various load balancing algorithms exist as stated in [7] with the aim of distributing the task's load across resources. Some of these algorithms include;

4.1 Min-Min Algorithm

This algorithm has all the relevant information needed in advance. The algorithm uses some parameters to obtain the information it needs. Some of these parameters are; ETC (Expected Time Compute), MET (Minimum Execution Time), MTC (Minimum Completion Time) etc. The Min-Min algorithm selects a task with minimum completion time and maps it with a node with a minimum completion time [8].

4.2 Max-Min Algorithm:

Max-min algorithm chooses large task to be executed firstly before executing small once [10]. This algorithm works almost the same way as the Min-Min algorithm except in Max-Min the task with maximum value is selected from the set of execution time of tasks and maps it to a node with minimum completion time. The ready time of the node is updated by adding the execution time of the task [7, 8].

As the cloud users sends task to the cloud environment with different requirement to the cloud service providers. The requirement can be tasks with different set of data size and processing power, the task scheduler will then match the tasks with available resources (virtual resources) that are available. Some mathematical relations are given in [9] to analyze resources scheduling in cloud computing which are employed and used in this paper are given below.

The set of VMs V with their respective processing power is given as;

$$V = \{v_j(c_j) \mid j = 1,2,\ldots w\} \qquad (1)$$

The set of tasks is also given as

$$T = \{t_i(s_j, q_j) \mid i = 1,2,\ldots y\} \qquad (2)$$

Where

c_j = processing speed (MIPS)

t_i = given task i

s_i =Data file size of a given task (Mb)

q_i = Processing power (MI) of a task t_i

With above equations (1) and (2) the expected execution time (EET) for a given task by a virtual resource can be obtained as;

$$EET = Size\ of\ task\ (MI)/Computing\ Power\ of\ resource\ (MIP) \qquad (3)$$

Now with equation (3) above, another metric can be obtained, which is the completion time (CT) of task t_i by a given resource.

$$CT_{(i,j)} = EET_{(i,j)} + r_i \qquad (4)$$

Where r_i indicate the starting time of the execution of task t_i.

Using (4), another important metric can be obtained, which is called the makespan, define as a measure of the throughput of the heterogeneous computing system [7].

$$makespan = Max_{(i\in w, j\in y)}(CT_{i,j}) \qquad (5)$$

This paper employs a known scheduling algorithm called an improved max-min algorithm from [1] and then based on this algorithm we propose another algorithm that will help in balancing load across the VMs' resources to improve the performance of the system.

4.3 Improved Max-Min Algorithm

The Max-min algorithm allocated task t_i to resource v_j such that large tasks have higher priority. For instance for a given large task, the max-min algorithm execute smaller task concurrently while running large tasks. Therefore, the largest task determines the total makespan for other resources. The improved max-min algorithm is given below [2].

```
for all submitted tasks in Meta-task; $t_i$
      for all resources; $v_i$
      $C_{ij} = E_{ij} + t_i$
      Find task $t_k$ costs maximum execution time
      Assign task $t_k$ to resource $v_j$ which gives minimum
completion time
Remove task $t_k$ from Meta-tasks set.
Update $t_j$ for selected$v_j$.
Update $c_{ij}$ for all j.
While Meta-task not Empty
      Find task $t_k$ costs maximum execution time.
      Assign task $t_k$ to resource $v_j$ which gives minimum
completion time
Remove Task $t_k$ form Meta-tasks set.
```

4.4 Proposed Algorithm

The improved max-min algorithm is reliable and proved to be efficient in scheduling the set of tasks to the available resources. However to make sufficient use of resource a proposed algorithm was introduced which is based on the improved max-min algorithm but small changes are made to make sure that all resources are used sufficiently and to minimize the use of these resources if few once can perform the task. The proposed algorithm is shown in the pseudo code below;

```
For all submitted tasks in Meta-task; Ti
      For all resources; Rj
                $C_{ij} = E_{ij} + t_j$
            Find task $T_k$ costs maximum execution time
            Assign task $T_k$ to its corresponding resources $R_j$
Remove task $T_k$ from Meta-tasks set.
Update $r_j$ for selected $R_j$.
Update $C_{ij}$ for all j.
Pivot= $T_k$;
For all updated task in Meta-task; Ti
      For all updated resources; $R_j$
            Find task $T_h$ costs maximum execution time
                  Assign task $T_h$ to its corresponding
            resource $R_j$
Remove task $T_h$ from Meta-tasks set.
Update $r_j$ for selected $R_j$.
Update $C_{ij}$ for all j.
2pivot= $t_h$
While Meta-task not Empty
      Find task $T_g$ costs maximum execution time.
            If 2pivot+$t_g$ ≤Pivot then
                  Assign task $T_g$ to previous resource $R_j$
                  which gives minimum completion time
      Remove Task $T_g$ form Meta-tasks set.
        Update $r_j$ for Selected $R_j$.
      Update $C_{ij}$ for all j.
      Updata 2pivot.
```

In the algorithm the total makespan is made to be a pivot 1 value for the first step and another pivot 2 value is assigned during the second step of the execution. Then during the next execution step the second pivot value and the completion time of the current state are summed up together. If they are greater than the first pivot value, then a new resource is allocated to that task. By given this criteria, the resources can be used in a balanced manner and fewer resources can be used, the remaining resources will not be involved to minimize the use of such resources. The aim of the above algorithm when compared to the improved Max-Min algorithm is to make effective used of the available resource during scheduling.

The flowchart for the above pseudo code is given in the figure (Fig. 3)

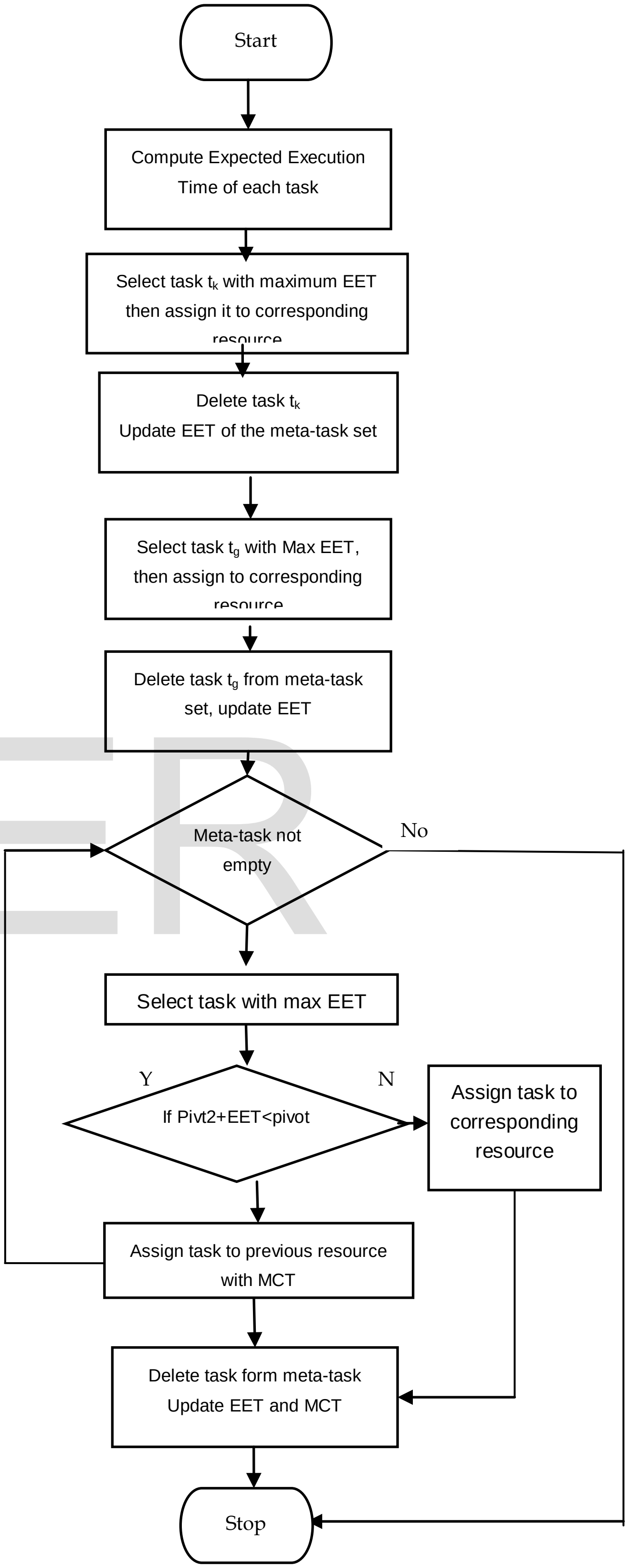
Start
Compute Expected Execution Time of each task
Select task t_k with maximum EET then assign it to corresponding resource
Delete task t_k Update EET of the meta-task set
Select task t_g with Max EET, then assign to corresponding resource
Delete task t_g from meta-task set, update EET
Meta-task not empty
No
Select task with max EET
Y
N
If Pivt2+EET<pivot
Assign task to corresponding resource
Assign task to previous resource with MCT
Delete task form meta-task Update EET and MCT
Stop

Fig. 3: Proposed Algorithm Flowchart

5. PROPOSED ALGORITHM RESULT AND ANALYSIS

Scenario: Below is a theoretical analysis of some predefined meta-task values and resources used to carry out the scheduling process. The tables below shows the meta-task values and the resources used.

Task	Size of task (MI)	Data volume (Mb)
T1	512	200
T2	1028	500
T3	420	300
T4	330	410
T5	550	328

Table1: Tasks values

The table (table 2) below holds the processing speed and the bandwidth of the resources on a network system.

R	Processing speed (MIPS)	Bandwidth (MbPS)
R1	128	100
R2	1256	120
R3	284	150

Table 2: Resource processing speed and bandwidth

Given the above values, Matlab is employed to compute the expected execution time of each task and the results are tabulated and analyzed as given in table 3 below;

	R1	R2	R3
T1	4	2	1.802
T2	8.031	4.015	3.619
T3	3.281	1.640	1.478
T4	2.578	1.289	1.161
T5	4.296	2.148	1.936

Table 8: Expected execution time of task

From the above tables i.e. table 8: T_i with maximum execution time is selected and then is assigned to the corresponding resource R_i. The Gantt chart in (fig. 3) shows how the allocation was performed based on the max-min idea, the task are allocated to all the available resources within the scheduler.

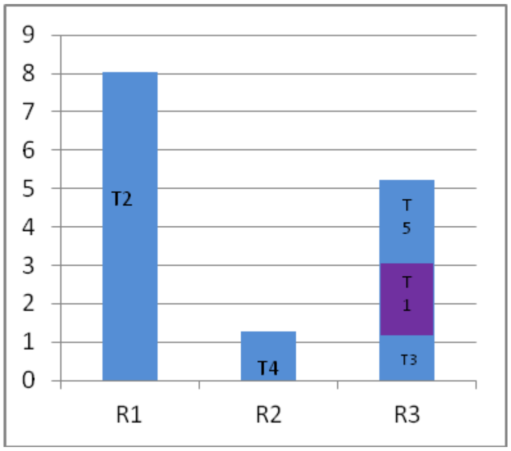

Fig. 3: Chart for Resource Allocation for Max-Min Algorithm

In contrast to the proposed max-min scheduling algorithm below shows how the allocation is performed in the figure (fig. 4) below.

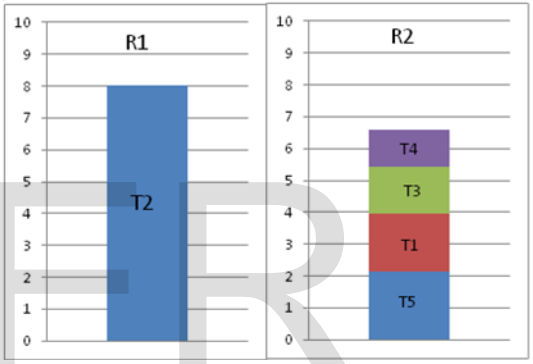

Fig. 4: Chart for Resource Allocation for proposed Max-Min Algorithm

From the chart (fig. 4), the largest task has a maximum makespan of 8.031 and it's scheduled to resource R1. The maximum makespan, is considered as the maximum throughput for other resources. This makes it possible to balance different smaller tasks to run concurrently on different resources across the system and also to use the resources wisely when needed. Another important factor which is based on the on-demand characteristics of cloud computing is that, the number of resources used is also minimized and a resource can be put into use when there is a demand for that resource. Based on the results obtained, instead of assigning the load to the three resources, it's possible to assign the task to only two resources, thereby increasing the efficiency of the system, thus we can have many task running concurrently on some resources and other resources can be put in use only when the need arise.

6. Conclusions

In conclusion Cloud Computing is an on-demand service, therefore, efficient on-demand allocation of VM is needed. In this paper, technique to handle on-demand allocation is analyzed. Allocation of resources can be performed efficiently within a cloud environment by balancing the load across the various virtual machine resources, by employing an efficient technique for load balancing such as the max-min algorithm that was used in this paper.

The usage of max-min technique made it possible to handle resources in an efficient and balanced manner. Thus, for a better service to be experienced in a field of cloud computing, a proper and efficient allocation techniques need to be adopted.

References

[1]. Upendra Bhoi, Purvi N. Ramanuj. Enhance Max-Min Task Scheduling Algorithm in Cloud Computing. International Journal of Application or Innovation Engineering & Management. 2013.

[2]. O. M. Elzeki, M. Z. Reshad and M. A. Elsoud. Improved Max-Min Algorithm in Cloud Computing. *International Journal of Computer Applications (0975 – 8887) Volume 50 – No.12, July 2012*.

[3]. Y Yuan, W-Cai Liu. Efficient resource management for cloud computing 2011.

[4]. Ryan Knight, The new role of XML in cloud data integration Using XML to integrate Salesforce data with enterprise applications. June 2009.

[5]. R Shelke, R Rajani. Dynamic resource allocation in Cloud Computing. 2013.

[6]. Ronak Pate, Sanjay Patel, Survey on Resource Allocation Strategies in Cloud Computing. *International Journal of Computer Applications (0975 – 8887) Volume 50 – No.12, July 2012*

[7]. S. Swaroop Moharana, D. Rajadeepan. Analysis of Load Balancer in Cloud Computing. International Journal of Computer Science and Engineering Vol.2 2013.

[8]. D. Manan Shah, A. Amit Kariyani, L. Dipak Agrawal. Allocation of Virtual Machines in Cloud Computing using Load Balancing Algorithm. International Journal of Computer Science and Information Technology & Security. Vol. 3 2013.

[9]. Yichao Yang, Yanbo Zhou. Heuristic Scheduling Algorithms for Allocation of Virtualized Network and Computing Resources. Journal of Software Engineering and Application 2013.

[10]. Pinal Salot, A survey of various scheduling algorithm in cloud computing environment, IJRET | FEB 2013.

[11]. Patel, Pankesh, Ajith H. Ranabahu, and Amit P. Sheth. "Service level agreement in cloud computing." (2009).