



## REVISED LOAD BALANCING ALGORITHM IN CLOUD COMPUTING

Settu Bharti <sup>(1)</sup>, Naseeb Singh <sup>(2)</sup>

<sup>(1)</sup> Research Scholar, Department of Computer Science Engineering, AIET, Faridkot  
settu.bharti@yahoo.com

<sup>(2)</sup> Assistant Professor, Department of Computer Science Engineering, AIET, Faridkot  
naseebdhillon@hotmail.com

### ABSTRACT

Cloud computing is one of the latest and upcoming paradigm that offers huge benefits such as reduced time to market, unlimited computing power and flexible computing capabilities. It is a model that provides an on-demand network access to a shared pool of computing resources. It comprises a large number of concepts primarily Load Balancing, Scheduling, etc. This paper discusses load balancing as a mechanism to distribute the workload evenly to all nodes in the system to achieve a higher resource utilization and user satisfaction. It helps in allocation and de-allocation of instances of applications without failure. This paper reports a new load balancing technique and its comparison with existing algorithm providing better results.

### Keywords

Cloud Computing; Load Balancing; Virtual Machine; Data Center; Data Center Broker; Cloudlet; FCFS; SJF.



# Council for Innovative Research

Peer Review Research Publishing System

Journal: INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY

Vol. 14, No. 12

[www.ijctonline.com](http://www.ijctonline.com), [editorijctonline@gmail.com](mailto:editorijctonline@gmail.com)



## INTRODUCTION

Cloud computing is an attracting technology in the field of computer science. In Gartner's report[1], it says that the cloud will bring changes to the IT industry. The cloud is changing our life by providing users with new types of services. Users get service from a cloud without paying attention to the details[2]. NIST gave a definition of cloud computing as a model for enabling ubiquitous, 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[3]. More and more people pay attention to cloud computing[4, 5]. Cloud computing is efficient and scalable but maintaining the stability of processing so many jobs in the cloud computing environment is a very complex problem with load balancing receiving much attention for researchers.

Cloud computing has been adopted by organization which includes, social networking websites, online application design by Google app managers and by Google doc which are some of the important implementation and a step ahead in cloud computing. Some clouds are also designed for online software testing. This all suggests that cloud computing will change the way we interact with the resources via Internet.

Cloud models used virtualization technology; this technology helps in slicing a single data centre or high power server to act as multiple machines. It depends on the hardware configuration of the data centre or server in how many virtual machine they can be divided. To implement virtualization additional software is also required. This software is the system software an operating system, can be from windows for example windows server 2008 or Hyper-V or for an open source environment like Linux.

According to the National Institute of Standards and Technology (NIST) [13], cloud computing exhibits several characteristics:

**On-demand Self-service-** A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.

**Broad Network Access-** Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).

**Resource Pooling-** The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, and network bandwidth.

**Rapid Elasticity-** Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand, To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.

**Measured Service-** Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

As cloud computing is in its evolving stage, so there are many problems prevalent in cloud computing [11],[16]. Such as:

- Ensuring proper access control (authentication, authorization, and auditing).
- Network level migration, so that it requires minimum cost and time to move a job.
- To provide proper security to the data in transit and to the data at rest.
- Data availability issues in cloud.
- Legal quagmire and transitive trust issues.
- Data lineage, data provenance and inadvertent disclosure of sensitive information is possible .

And the most prevalent problem in Cloud computing is the problem of load balancing.

**Load balancing** is the pre requirements for increasing the cloud performance and for completely utilizing the resources. Load balancing is centralized or decentralized. Load Balancing algorithms are used for implementing. Several load balancing algorithm are introduced like round robin algorithm a mining improvement in the performance. The only differences with this algorithm are in their complicity. The effect of the algorithm depends on the architectural designs of the clouds [4]. Today cloud computing is a set of several data centres which are sliced into virtual servers and located at different geographical location for providing services to clients. The objective of paper is to suggest load balancing for such virtual servers for higher performance rate.

In general, load balancing algorithms follow two major classifications:

- Depending on how the charge is distributed and how processes are allocated to nodes (the system load);
- Depending on the information status of the nodes (System Topology).



In the first case it is designed as centralized approach, distributed approach or hybrid approach and in the second case as static approach, dynamic or adaptive approach.

#### a) Classification According to the System Load

- **Centralized approach:** In this approach, a single node is responsible for managing the distribution within the whole system.
- **Distributed approach:** In this approach, each node independently builds its own load vector by collecting the load information of other nodes. Decisions are made locally using local load vectors. This approach is more suitable for widely distributed systems such as cloud computing.
- **Mixed approach:** A combination between the two approaches to take advantage of each approach.

#### b) Classification According to the System Topology

- **Static approach:** This approach is generally defined in the design or implementation of the system.
- **Dynamic approach:** This approach takes into account the current state of the system during load balancing decisions. This approach is more suitable for widely distributed systems such as cloud computing.
- **Adaptive approach:** This approach adapts the load distribution to system status changes, by changing their parameters dynamically and even their algorithms. This approach is able to offer better performance when the system state changes frequently. This approach is more suitable for widely distributed systems such as cloud computing.

## RESEARCH METHODOLOGY

The main objective of this paper was to answer the question: in identical cloud environments, which load balancing architecture: centralized, decentralized or hierarchical architecture will give the best results in terms of response time and server load?

To answer this question a robust evaluation framework [7] was implemented which includes the following steps:

- Identifying the critical elements in the design of cloud load balancers which are: number of nodes, load balancing algorithm and the architecture of the load balancers.
- Varying the number of nodes, in order to simulate representative sample of cloud environments.
- Selecting simple load balancing algorithms, Round Robin algorithm and opportunistic load balancing algorithm, to enable assessing the efficiency of each load balancing architecture.
- Identifying suitable performance measures: response time and server load.
- Implementing three identical scenarios, one for each load balancing architecture: centralized, decentralized and hierarchical architecture.
- Running the simulations and collecting the results.
- Repeating the simulation several times to improve the accuracy and accept the mean outcome.

## SIMULATION IN CLOUD: CLOUDSIM

In CloudSim, cloud computing infrastructures and application services allowing its users to focus on specific system design issues that they want to investigate [8]. Simulation in a CloudSim means implementation of actual environment towards benefit of research. The users or researcher actually analyse the proposed design or existing algorithms through simulation. Resources and software are shared on the basis of client's demand in cloud environment. Essentially, dynamic utilization of resources is achieved under different conditions with various previous established policies. Sometime it is very much difficult and time consuming to measure performance of the applications in real cloud environment. In this consequence, simulation is very much helpful to allow users or developers with practical feedback in spite of having real environment. In this research work, simulation is carried out with a specific cloud simulator, CloudSim [7]. Figure 1 shows Layered CloudSim architecture.

A brief description of these vital components and the working relationship between them is presented in the following [7].

**Data centre:** Data centre encompasses a number of hosts in homogeneous or heterogeneous configurations (memory, cores, capacity, and storage). It also creates the bandwidth, memory, and storage devices allocation.

**Virtual Machine (VM):** VM characteristics comprise of memory, processor, storage, and VM scheduling policy. Multiple VM can run on single hosts simultaneously and maintain processor sharing policies.

**Host:** This experiment considers VM need to handle a number of cores to be processed and host should have resource allocation policy to distribute them in these VMs. So host can arrange sufficient memory and bandwidth to the process elements to execute them inside VM. Host is also responsible for creation and destruction of VMs.



**Cloudlet:** Cloudlet is an application component which is responsible to deliver the data in the cloud service model. So the length, and output file sizes parameter of Cloudlet should be greater than or equal to 1. It also contains various ids for data transfer and application hosting policy.

### PROPOSED ALGORITHM

The proposed algorithm maximizes the utilization of cloud resources, and aims to reduce task waiting time. Fig.1 shows main procedures of proposed algorithm. The traditional grouping-based scheduling algorithm does not consider the network bandwidth and tasks file size [3]. The traditional grouping-based scheduling algorithm only takes into account the fine-grained jobs where processing requirements are small and almost equal [1] [3] [11] [12]. However in practice, tasks are uncertain and very random, and therefore the size of tasks cannot be very lightweight or processing requirements are not pre-established. In this paper, proposed grouping-based scheduling algorithm is suitable for both very lightweight jobs and the tasks which processing requirements are random and unpredictable.

**Improvement:** Traditional task grouping scheduling does not take into account in situations such that a task processing requirement (MI) is over each resource total MI occasionally; under this circumstance, grouping will not continue. In the proposed algorithm, once a task MI is over each resource in total MI, the task is grouped itself as a new grouped task with its own MI and a unique ID. This grouped task is allocated to the sequential resource.

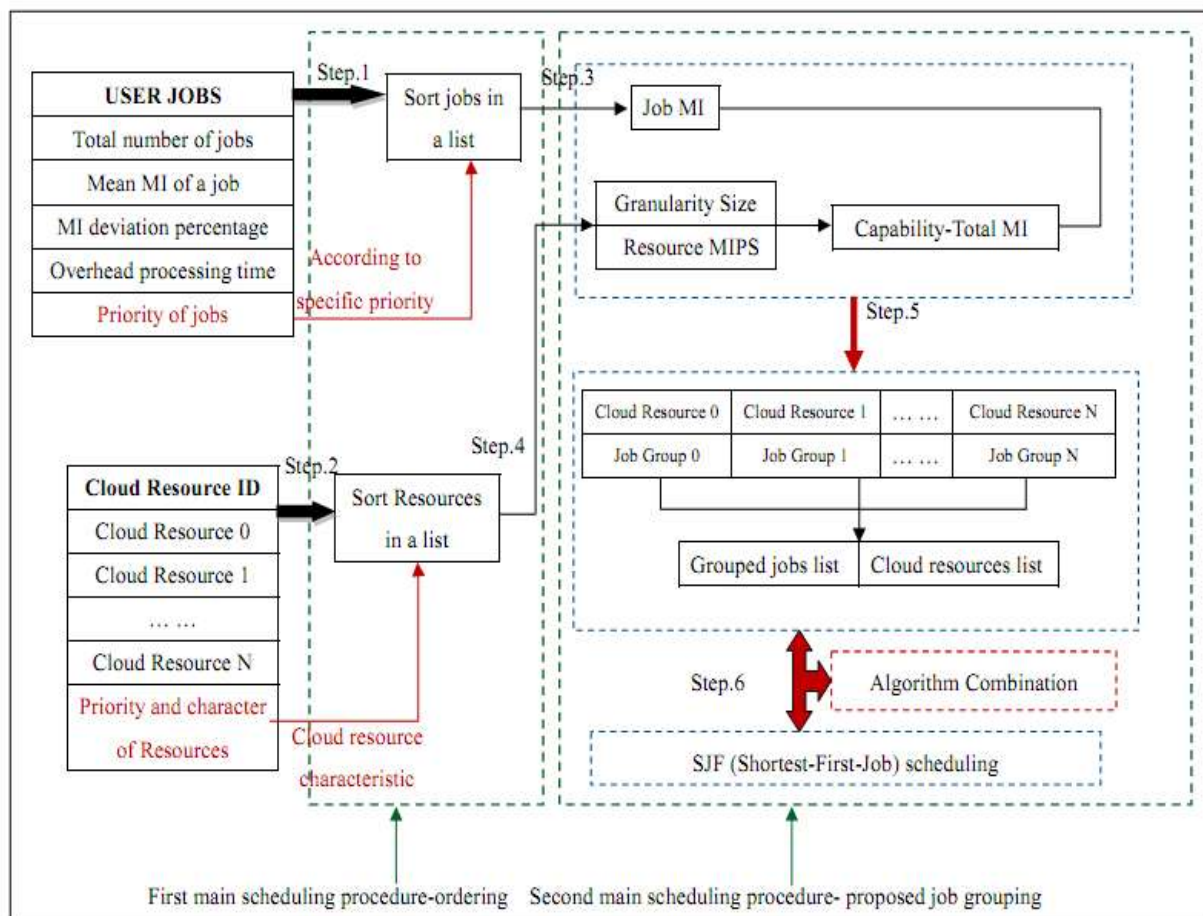


Figure 1. Flow Chart of Proposed work.

### Algorithm

- 1: sort (groupedcloudletList, MI in ascending order)
- 2: for k := 0 to groupedcloudletListsize-1 do
- 3:     calculate CT(l,j) of C(j) for VM(j)  
 $CT(i,j) = ET(i,j) + r(j)$ , where  $ET(i,j)$  is the expected execution time of task  $t(i)$  on machine  $m(j)$  and  $r(j)$  is the ready time of  $m(j)$  i.e. the time when  $m(j)$  becomes ready to execute  $t(i)$ .
- 4: end for



```
5: for k := 0 to groupedcloudletList size-1 do
6:     find the minimum expected completion time of each task in MT(meta task table) and the resource that will obtain
   it.(tasks are collected into a set called meta task(MT))
7: end for
8. sort(groupedVMList, MIPS in descending order)
8: pre cloudletFileSize = 0
9: pre cloudletOutputSize = 0
10: total resMIj = ResList
11: while total jobMI≤total resMI
21:   i--;
22:   if total jobMI>total resMI then
23:     total jobMI = total jobMI-cloudletList
24:     max cloudletFileSize = pre cloudletFileSize
25:     max cloudletOutputFileSize =precloudletOutputFileSize
26:     print maxsize
27:   end if
28: Create a new task whose MI equals total jobMI
29: Set a unique ID (groupid) to the created task
30: Insert this task into groupedcloudletList
31: Put the task on the groupedcloudletList
32: Insert the corresponding resource ResList into groupedvmList
37: Put the corresponding resource ResList on groupedvmList
38: end while
```

## EXPERIMENTAL RESULTS

For the simulation, we focus on average waiting time, the total processing time, and processing cost. (1) The waiting time and total finishing time is computed in milliseconds. CloudSim is a new generation and extensible simulation platform which enables seamless modeling, simulation, and experimentation of emerging Cloud computing infrastructures and management services to be accomplished [16] [17] [18]. CloudSim is used to verify the correctness of the proposed algorithm. CloudSim toolkit is used to simulate heterogeneous resource environment and the communication environment. Cloud resources are depicted as some parameters, such as VM ID, processing elements, processing capability (MIPS), bandwidth, RAM, cost and so on in TABLE 1. In this experiment, 5 cloud resources are created to simulate the experiments. The bandwidth, processing capability of resources are main factors that we consider in this experiment.

VM ID	MIPS	RAM	COST
1	125	256 MB	2.4
2	250	512 MB	2.4
3	500	1GB	2.4
4	750	2GB	2.4
5	1000	2GB	2.4

**Table 1. Cloud Resource Parameters**



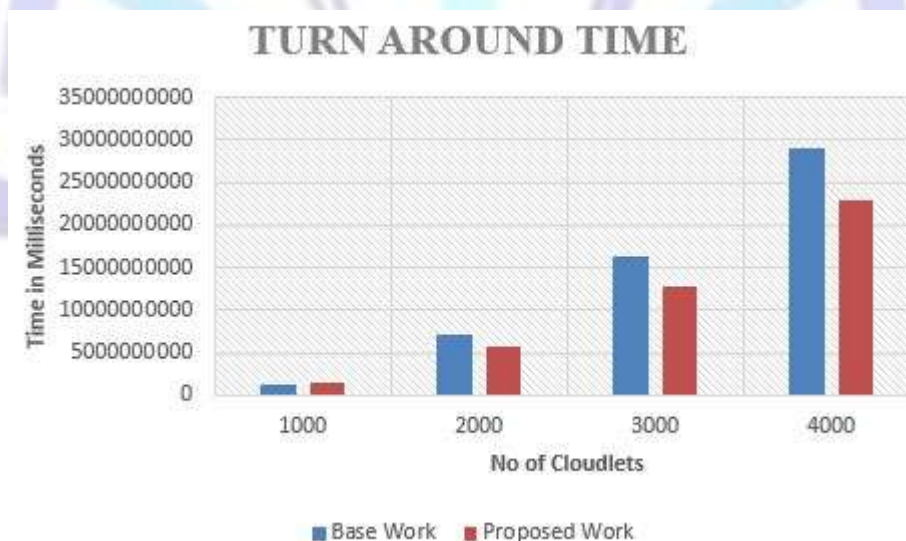
All the tasks are submitted to the datacenter broker. In CloudSim, tasks are packaged as cloudlets, which contain jobs requirements (MI), size of job input and output data (in bytes) and other various parameters related with execution when tasks are deployed to corresponding resources by broker. These cloudlets simulate Cloud-based application services, such as content transfer, social networking, and so on. Each application complexity is described by computational requirements. Therefore, each application has a pre-defined processing requirements component which is inherited from cloudlets and amount of data transfer which is related with input file size and output file.

The experiments are conducted several number of times by taking different number of cloudlets like 1000,2000,3000 and so on. The average waiting time, total processing time, processing cost have been computed as below in Table 2.

Number of Cloudlets	AVERAGE WAITING TIME	TOTAL PROCESSING TIME	TURN AROUND TIME	TOTAL PROCESSING COST
1000	1423593	6574192	1213754645	8578061.59
2000	2855504	15154273	7248722372	19570254.48
3000	4284476	22740240	16315021771	47376575.76
4000	5713630	31314433	28991224603	60754638.00
5000	7145543	42894512	45292056932	102946829.52
10000	14293706	85794912	181159072107	205907789.52
15000	21444630	128701200	407618132599	308882878.80
20000	28595736	171595713	724573272315	411829710.24
30000	42894823	257402400	1630343841030	617765760.05

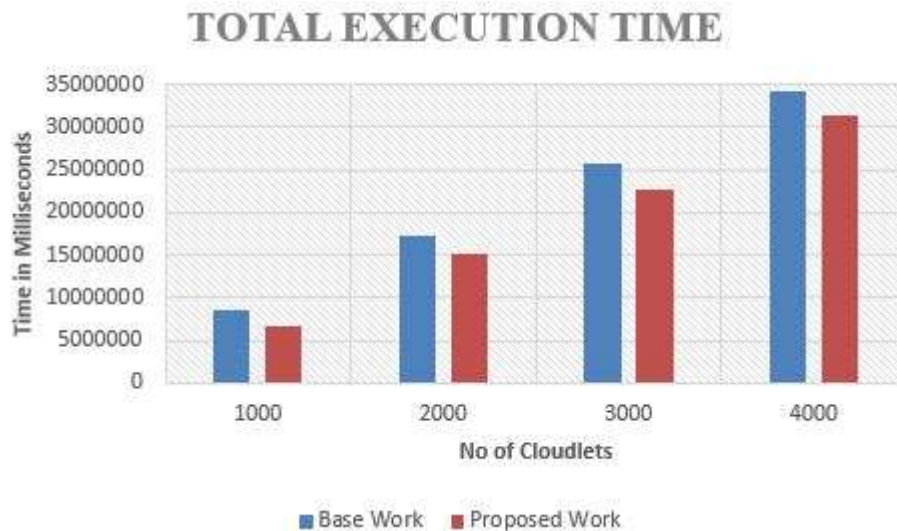
**Table 2. Experimental Results for different number of Cloudlets**

**Turn Around Time:** It is the total time taken between the submission of a cloudlet for execution and the return of the complete output to the client. Figure 2 shows the turn around time taken by the base algorithm and the proposed algorithm and it is clear from the bar chart that we have successfully reduced the turn around time in this research work.



**Figure 2. Turn Around Time for base work and proposed algorithm.**

**Processing Time/ Execution Time:** The time from the submission of a request to the time of first response by the cpu. It is the amount of time it takes to get first response for the request submitted. Figure 3 depicts the total processing time / execution time for the base algorithm and proposed algorithm for the given number of cloudlets.



**Figure 3. Total Execution Time for base work and proposed algorithm.**

**Average Waiting Time:** Waiting time is the amount of time that is taken by a process in ready queue. Average waiting time is the time for which the cloudlets were kept in the queue for their execution.



**Figure 4. Average waiting time for base work and proposed algorithm.**

The processing cost is calculated based on the actual CPU time when the tasks are accomplished to execute on the cloud resources and the cost of resources per second. From the figure 5, it is clear that the processing cost occurred to the client is lesser in the proposed algorithm than in the base algorithm. The processing cost is the main factor which has been occurred to the client. Lesser the processing cost, more will be the benefit to the client and thereby increasing the overall client's weightage at the cloud provider. The experiment has been conducted on multiple cloudlets and on multiple VM's which depict that the proposed algorithm has been working consistently on any type of configuration and parameters.



Figure 5. Processing cost for base work and proposed algorithm.

## CONCLUSION

Load Balancing is the most important aspect of cloud computing. It helps in reducing the dynamic workload across the all nodes for the achievement of higher user and resource satisfaction. It helps in reducing the overhead, response time and increasing scalability. This paper depicts the cloud computing in detail. It basic fundamentals of the cloud and its characteristics like on-demand access, resource provisioning, etc. Secondly, it elaborates the fundamental concept of load balancing defining its goals and challenges. Then, it provides the Existing Work as well as the proposed work on the simulated environment. Lastly the experimental setup used for the creation of the virtual system and the resources used are specified in the system along with the graphical and tabular results of the algorithms. Thus, the proposed algorithm reduces the waiting time of the machines along with the overheads in the system resulting in lesser cost than existing approach. The future work includes modifications in the system leading to greater resource utilization and performance.

## REFERENCES

- [1] O.M. Elzeki, "Improved Max-Min Algorithm in Cloud Computing". International Journal of Computer Applications(0975-8887) Volume 50-No.12, July 2012.
- [2] Vishwas Bagwaiya, "A HYBRID APPROACH USING THROTTLED AND ESCE LOAD BALANCING ALGORITHMS IN CLOUD COMPUTING.
- [3] Amandeep Kaur Sidhu. "Analysis of load balancing techniques in cloud computing". International Journal of computers & technology volume 4 No. 2, March-April, 2013, ISSN 2277-3061.
- [4] Ektemal Al-Rayis. "Performance Analysis of load balancing Architectures in Cloud computing" 2013 European Modeling Symposium. 978-1-4799-2578-0/13\$31.00@2013 IEEE.
- [5] Haozheng Ren, "The load balancing Algorithm in cloud computing Environment" 2nd International Conference on computer science and network technology, 2012.
- [6] Tushar Desai. "A survey of various load balancing techniques and challenges in cloud computing" International Journals of scientific and technology research volume 2, Issue11,Nov2013.
- [7] Upendra Bhoi. "Enhanced max-min Task scheduling Algorithm in cloud computing". International Journal of Application or Innovation in Engineering & management(IJAIEM), April 2013.
- [8] Klaitheem Al Nuaimi, "A survey of load balancing in cloud computing challenges and algorithm". 2012 IEEE second symposium on network cloud computing and applications.
- [9] Gytis Vilutis, "Model of load balancing and scheduling in cloud computing". Proceedings of the ITI 2012 34th Int.Conf. on Information Technology Interfaces, June 25-28,Cavat,Croatia.
- [10] L.D.Dinesh Babu<sup>a,\*</sup>. "Honey Bee Behaviour inspired load balancing of tasks in cloud computing environments, Jan 30,2013.
- [11] Kumar Nishant, " Load Balancing of Nodes in Cloud Using Ant Colony Optimization". 2012 14th International Conference on Modelling and Simulation.
- [12] Mr. M. Ajit, " VM Level Load Balancing in Cloud Environment", IEEE – 31661.