



# An Enhanced Scheduling in Weighted Round Robin for the Cloud Infrastructure Services

<sup>1</sup>R. Bhaskar, <sup>2</sup>Rhymend Uthariaraj, D. Chitra Devi

<sup>1</sup>Final Year M.E (SEOR), <sup>2</sup>Professor & Director, <sup>3</sup>Research Scholar

Ramanujan Computing Centre, Department of Computer Science and Engineering, Anna University, Chennai, India  
Email: <sup>1</sup>crbhaskar@gmail.com, <sup>2</sup>rhymend@annauniv.edu, <sup>3</sup>chitradevi.chithu@gmail.com

**Abstract - Infrastructure as services (IaaS) in Cloud computing offers the features of increasing the hardware resources (CPU cores, RAM, hard disk or storage space, and network bandwidth) on-demand in no time or utilizing the additional configured VMs to execute the business applications in a shortest time as possible by utilizing the dynamic scaling. The scheduling algorithm in the IaaS allows the applications to respond in a shortest response time by utilizing the configured resources very efficiently and it can have a greater impact in performance of the IaaS Cloud Environment by reducing the load balancing actions. This work introduces an enhanced scheduling in Weighted Round Robin algorithm for the cloud infrastructure based on the heterogeneous resources and heterogeneous with interdependent jobs combined with its unpredicted arrivals and the varied instruction length of the requested jobs.**

**Keywords-Cloud Computing, Scheduling, Dynamic Schedule, Scheduling Algorithms.**

## I. INTRODUCTION

Cloud computing is an efficient and scalable infrastructure for computations, but maintaining the workload balancing on the participating resources is a very complex task. So, the researchers are putting more attention in the complex problem of scheduling and load balancing in the cloud computing environment due to the un-predictable arrival pattern of the variable length jobs with different set of data. Additionally, the capacities of each participating node in the clouds may differ. These complexities increase the dependencies on the dynamic scheduling of Weighted Round Robin in the cloud environment.

Scheduling schemes in Load balancing can be either static or dynamic. Static schemes do not use any of the dynamic system information. Instead, it simply allocates the incoming jobs in a round robin/weighted round robin fashion on the available resources. But the dynamic scheme additionally uses the system information to send the jobs to the least utilized VM and avoiding the

overload on the higher utilized VM. The optimal allocation of jobs into VMs by the dynamic scheduler can reduce the effort of the load balancer and in-turn it can reduce the number of job migrations between the VMs.

The Enhanced scheduling of Weighted Round Robin given in this work uses both the static scheduling based on the resource capabilities combined with the arrived job lengths and the dynamic scheduling based on the run time load in the resources, the participating resource capabilities and the job lengths.

The static scheduler used once at the initialization of the environment due to the entire participating resource's idle state on that initialization stage. The dynamic scheduler handles the jobs, which arrived during the run time of the environment. This dynamic scheduler allocates the jobs into the appropriate environments based on the weighted round robin, load on the participating resources and the length of the arrived jobs.

The main objective of this dynamic scheduling is to minimize the response time of the jobs by optimally utilizing the participating virtual machines using the combination of static and dynamic scheduling by identifying the length of the jobs, resource capabilities, effectively predicting the under-utilized VMs and avoiding the overload on any of the VMs.

Section 2 gives a brief overview of the Related Work. Section 3 discusses our dynamic scheduling model implementation to obtain the optimal results. In Section 4, we present and discuss analytical as well as simulation results. The findings are summarized in Section 5, and we also outline the directions for future work.

## II. RELATED WORKS

There are many research work had happened on scheduling for the cloud environment. But still the many new architectures and changes in the cloud environment creates a challenge to the scheduling algorithms. The goal of scheduling is to improve the participating Virtual Machine performance in the cloud environment by evenly distributing the load among these various resources to achieve the optimal resource utilization, maximum throughput, minimum response time and avoiding overload.

Currently, there are many scheduling algorithms such as Round Robin, Weighted Round Robin, Equally Spread Current Execution Algorithm, and Ant Colony algorithm. But these algorithms had not met all the probabilities of the homogenous and heterogeneous environments with the homogenous or heterogeneous jobs on its unpredicted arrival patterns and widely varying loads [7], [8]. A static load balancing mechanism for distributed systems is proposed in [10]. In [11], the authors used a game theoretic approach, to present a dynamic solution for load balancing in Grid systems to spread the load on each computing node. They presented this algorithm to minimize the average completion time of tasks by modelling the load balancing as a minimization problem.

The workloads patterns can be parallel jobs [1] or it could be small bags-of tasks (BoTs) [2] and sometimes it may be workflows [3], [4] by comprising many sequential tasks may arrive in the environment. Additionally the workflow may also contain thousands of tasks [13], and these workflows indeed require the cloud environment to complete all the tasks in the shorter time using the optimal dynamic scheduling. Also, the scheduling with load balancing should be able to handle all the types of workloads to deliver the optimal results with respect to the response time and resource utilization.

There has been a plenty of research activities in identifying & improving the performance of virtual resources in cloud computing environments [5], [6]. The major parts of workflow scheduling approaches such as HEFT [9] address a single objective, typically the makespan.

Our algorithm is inspired by the Weighted Round Robin method and it additionally considers the particularities of requested tasks, such as its length and inter-dependencies of the tasks, when making the accurate allocation decisions to the appropriate VMs in the Cloud environments to minimize the execution time of the jobs.

## III. ENHANCED SCHEDULING IN WRR

The Enhanced Scheduling implementation in Weighted Round Robin algorithm consists of four major modules.

- Static Scheduler (Initial Placements)
- Dynamic Scheduler (Runtime Placements)
- Task Inter-dependent Scheduler
- Resource Monitor

The Static Scheduler has the function to find the most suitable VM and assign the tasks to VMs during the no load condition on all the VMs using the weighted round robin algorithm. The Dynamic scheduler has the function to allocate the runtime arrival jobs to the most suitable VMs based on the least utilized VM at that particular job arrival time. Resource monitor communicates with all the VM's resource prober and collects the VM capabilities, current load on each VM, number of jobs in execution/waiting queues in each VM to decide the appropriate VM to the jobs. The Task Requirement Estimator identifies the length of the tasks to be executed and transfer the estimated results to the Load Balancer for its operative decisions.

### A. System Architecture

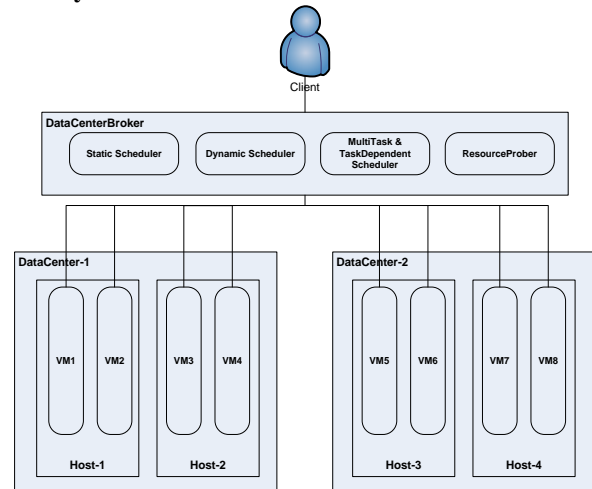


Fig 3.1: System Architecture

A Cloud can have many nodes and these nodes may be located in different geographical locations. But, this experiment setup considers a single data center cloud with multiple nodes and multiples vms on each host. The Datacenter broker acts as an entry point of the system and all the client's requests enter into the cloud environment through the datacenter broker. It handles all the traffic routing between user bases and data centers and the scheduling and load balancing.

The main enabling technology for cloud computing is virtualization. Virtualization generalizes the physical infrastructure, which is the most rigid component, and

makes it available as a soft component that is easy to use and manage. Host Resource is a physical computer connected in a computer network. A virtual machine monitor monitors a system of virtual machines (sometimes called hardware virtual machines), which allow the sharing of the underlying physical machine resources between different virtual machines, each running its own operating system. A virtual machine (VM) is software based, fictive computer. It is a software implementation of a machine (i.e. a computer) that executes programs like a physical machine. Every VM must have its own Operating System and it will act as like a independent machine using its VM allotted resource capabilities. Applications are the user's functional programs to meet their business requirements. These applications will run on its VM's Operating System.

**B. Enhanced Scheduling in Weighted Round Robin**

The proposed Enhanced Scheduling in Weighted Round Robin algorithm is the most optimal algorithm and it allocates the jobs to the most suitable VMs based on the VM's information like its processing capacity, current load on the VMs, and the length of arrived tasks with its priority. The static scheduling of this algorithm uses the processing capacity of the VMs, the number of incoming tasks and the length of each task to decide the allocation on the appropriate VM. The dynamic scheduling (at run time) of this algorithm, additionally uses the load on each of the VM along with the information mentioned above to decide the allocation of the task to the appropriate VM. There is a probability at run time that in some of the cases, the task may take longer execution time than the initial calculation due to the execution of more number of cycles (like a loop) on the same instructions based on the complicated run time data. In such situations, the load balancer rescue the scheduling controller and re-arrange the jobs according to the idle slot available in the other un-utilized/under-utilized VMs by moving a waiting job from the heavily loaded VMs.

**C. Static Scheduler in EWRR**

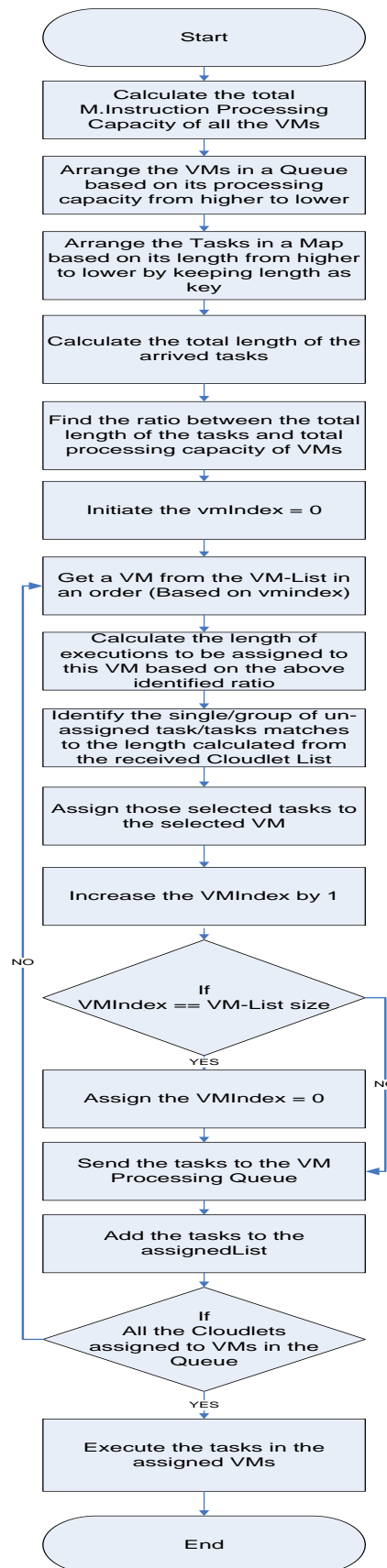


Fig 3.2: Flow Chart of Static Scheduler in EWRR

**D. Dynamic Scheduler in EWRR**

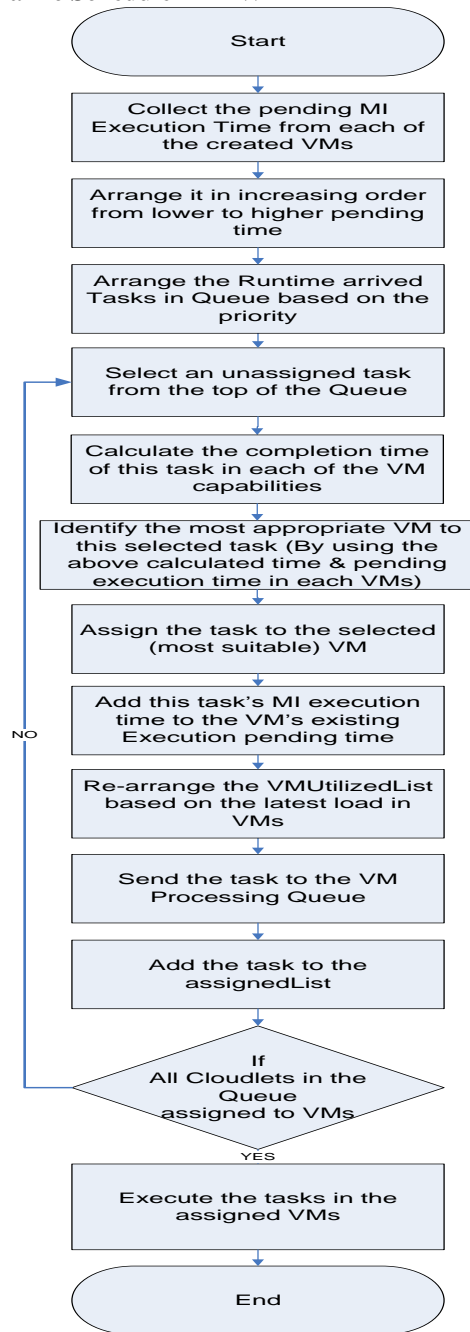


Fig 3.3: Flow Chart of Dynamic Scheduler in EWRR

**E. Simulation Tool**

CloudSim-3.0.3 is the simulation environment for the Cloud Computing Research and Project Works. It supports both system and behaviour modelling of Cloud system components such as data centers, hosts, virtual machines (VMs) and resource provisioning policies. It supports modelling and simulation of Cloud computing environments consisting of both single and inter-networked clouds (federation of clouds). It exposes custom interfaces for implementing scheduling and load balancing policies of jobs into vms and provisioning

techniques for allocation of VMs under inter-networked Cloud computing scenarios also. It can leverage virtualized services even on the fly based on requirements (workload patterns and QoS) varying with time.

In this work, the custom static and dynamic scheduling has been implemented to achieve the higher performance & utilization of VMs under varying load patterns. This intern produced the faster response time to the client's request on the application jobs.

**IV. EXPERIMENT & RESULTS**

The performance of the enhanced scheduling in Weighted Round Robin algorithm has been examined based on the results of simulation done using the cloudsim. The classes of the cloudsim simulator have been extended to utilize the newly written algorithm. In the following illustrations, the response time of combined tasks are examined in the Round Robin, Weighted Round Robin, and Enhanced Weighted Round Robin under the different load conditions and resource conditions.

**A. Cloud Setup Configuration**

TABLE I. CLOUD SETUP CONFIGURATION

Sl No	Entity	Quantity
1	Data Center	1
2	No of Hosts in DC	500 Hosts (200 Nos - 4 Core HyperThreaded Opteron270, 200 Nos - 4 Core HyperThreaded Opteron2218 and 100 Nos 8 Core HyperThreaded XeonE5430)
3	No of Process Elements	8/8/16
4	PE Processing Capacity	174/247/355 MIPS
5	Host Ram Capacity	16/32GB
6	No of VM	10 to 100 with an increment of 10
7	No of PE in VM	1
8	VM's Processing Capacity	150/300/90/120/93/112/105/225
9	VM RAM capacity	1920 MB
10	VM Manager	Xen
11	No of PE in Tasks	1
12	Task Length / Instructions	500000 to 200000000

**B. Comparison of Overall Execution Time**

The following table contains the execution time of different algorithms on different resource (number of Virtual Machines) conditions and varied load conditions.

TABLE II. EXECUTION COMPLETION TIME (SPACE SHARED)

SI No	No of VMs	Round Robin (secs)	Weighted Round Robin (secs)	Enhanced Weighted Round Robin with Job Length (secs)
1	10	12544	137771	113859
2	20	93018	89979	65732
3	30	93515	73933	51278
4	40	92439	73241	43515
5	50	92439	85821	39045
6	60	92439	77323	38676
7	70	92632	70693	35557
8	80	92632	62546	34873
9	90	92632	71261	34873
10	100	92632	70860	34873

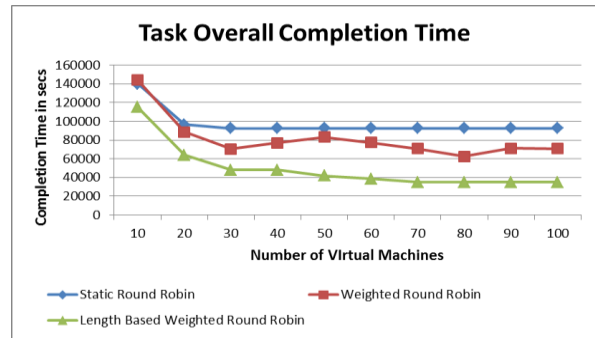


Fig 4.2: Execution Completion Time (Time Shared)

**C. Analysis**

The following is the order of highest to lowest performance of the algorithms in the provided heterogeneous environment.

- Enhanced Weighted Round Robin with Job Length
- Weighted Round Robin
- Round Robin

This result proved that the Enhanced Weighted Round Robin by Job Length delivers a faster completion time than the other two scheduling algorithms (Round Robin and Weighted Round Robin) in the heterogeneous resources (VMs) and heterogeneous jobs. The Enhanced Weighted Round Robin's static scheduler algorithm considers the job length along with processing capacity of the heterogeneous VMs to assign the job. So, the lengthy jobs get assigned to the higher capacity VMs in the heterogeneous environments. This helps in to complete the jobs in a shorter time. The dynamic scheduler considers the load of all its configured VMs and its tentative completion time of the current load has been identified. Then, the scheduler calculates the arrived job's estimated completion time in each of the configured VM and adds this calculated timing with the existing load's completion time on each VM. Now, it identifies the least possible completion time from the above calculations for any particular job in one of the VM and then the job has been assigned to this VM. So this algorithm is most suitable to the heterogeneous environment data centers.

The Weighted Round Robin considers the ratio of the VM capacity to the total VM capacity and it assigns the proportionate number of arrived Jobs into the VM. So it performs in the next level. But in case, if any lengthy jobs assigned to the low capacity VMs based on the above calculation, then this will delay the execution completion time.

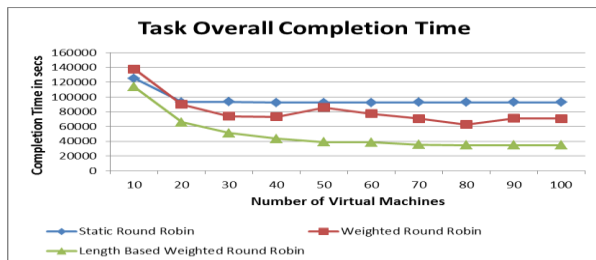


Fig. 4.1: Execution Completion Time (Space Shared)

TABLE III. EXECUTION COMPLETION TIME (TIME SHARED)

SI No	No of VMs	Round Robin (secs)	Weighted Round Robin (secs)	Enhanced Weighted Round Robin with Job Length (secs)
1	10	139860	143839	115114
2	20	96576	88643	63803
3	30	92439	70488	47947
4	40	92439	77119	47845
5	50	92439	83330	41817
6	60	92439	77323	38476
7	70	92632	70693	34873
8	80	92632	62546	34873
9	90	92632	71261	34873
10	100	92632	70860	34873

The simple Round Robin algorithm has not considered any variables about the environment, VM capabilities and the job lengths. It simply assigns the jobs to the VM lists one after another in an ordered manner. So its completion time of the jobs is higher than the other two algorithms.

## V. CONCLUSION & FUTURE WORK

In this work, the Enhanced weighted round robin algorithm considers the capabilities of each VM and the task length of each requested job to assign the jobs into the most appropriate VMs. The static scheduler algorithm pays attention to the initial placement of the jobs, which distributes the job requests to the participating VMs evenly based on the VM's capabilities and the length of the requested job. The dynamic scheduler considers the load of all its configured VMs and its tentative completion time of the current load has been identified along with the arrived job's estimated completion time in each of the configured VM. After this, the least possible completion time has been identified from the above calculations for this particular job in one of the VM and then the job has been assigned to this VM to complete the tasks with minimum possible time.

The same above algorithm can be extended further to utilize the load balancing in the heavily loaded scenarios for the task migrations. Currently, the multi-level inter dependency tasks has been considered. These tasks can be extended further for the multi-process parallel execution capable tasks using multiple processor elements in the CPU.

## REFERENCES

- [1] U. Lublin and D. G. Feitelson, "Workload on parallel supercomputers: modeling characteristics of rigid jobs," *J.Par.&Distr.Comp.*, vol. 63, no. 11, pp. 1105–1122, 2003.
- [2] A. Iosup, O. O. Sonmez, S. Anoep, and D. H. J. Epema, "The perf. of bags-of-tasks in large-scale distributed sys," in *HPDC*. ACM, 2008, pp. 97–108.
- [3] D. Thain, J. Bent, A. C. Arpaci-Dusseau, R. H. Arpaci-Dusseau, and M. Livny, "Pipeline and batch sharing in grid workloads," in *HPDC*. IEEE, 2003, pp. 152–161.
- [4] S. Ostermann, A. Iosup, R. Prodan, T. Fahringer, and D. H. J. Epema, "On the characteristics of grid workflows," in *CGIW*, 2008, pp. 431–442.
- [5] Alexandru Iosup, Simon Ostermann, M. Nezhir Yigitbasi, Radu Prodan, Thomas Fahringer and Dick H.J. Epema "Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing", *IEEE Trans. on Parallel and Distributed Systems*, vol. 22, no. 6, June 2011.
- [6] Hamzeh Khazaei, Jelena Mistic, and Vojislav B. Mistic, "Performance Analysis of Cloud Computing Centers Using M/G/m/m + r Queuing Systems", *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 5, May 2012.
- [7] K. Xiong and H. Perros, "Service Performance and Analysis in Cloud Computing," *Proc. IEEE World Conf. Services*, pp. 693-700, 2009.
- [8] J. Baker, C. Bond, J. Corbett, J.J. Furman, A. Khorlin, J. Larsonand, J.M. Leon, Y. Li, A. Lloyd, and V. Yushprakh, "Megastore: Providing Scalable, Highly Available Storage for Interactive Services," *Proc. Conf. Innovative Data Systems Research (CIDR)*, pp. 223-234, Jan. 2011.
- [9] H. Topcuoglu, S. Hariri, and M.-y. Wu, "Performance-Effective and Low-Complexity Task Scheduling for Heterogeneous Computing," *IEEE Trans. Parallel and Distributed Systems*, vol. 13, no. 3, pp. 260-274, Mar. 2002.
- [10] D. Grosu and A.T. Chronopoulos, "Algorithmic Mechanism Design for Load Balancing in Distributed Systems," *IEEE Trans. Systems, Man and Cybernetics, Part B*, vol. 34, no. 1, pp. 77-84, Feb. 2004.
- [11] A. Danak and S. Mannor, "Bidding Efficiently in Repeated Auctions with Entry and Observation Costs," *Proc. IEEE Int'l Conf. Game Theory for Networks*, pp. 299-307, May 2009.
- [12] Hamid Mohammadi Fard, Radu Prodan, and Thomas Fahringer "A Truthful Dynamic Workflow Scheduling Mechanism for Commercial Multicloud Environments", *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, June 2013.
- [13] G. Juve, A. Chervenak, E. Deelman, S. Bharathi, G. Mehta, and K. Vahi, "Characterizing and profiling scientific workflows," *Future Generation Computer Systems*, vol. 29, no. 3, pp. 682–692, Mar. 2013.

