Pown KAMARAJAPANDIAN, PhD Candidate
E-mail: kamarajapandianp@gmail.com
Assistant Professor Pandian CHITRA, PhD
E-mail:pccse@tce.edu
Thiagarajar College of Engineering, Madurai
Tamil Nadu, India

HJSA: A HIERARCHICAL JOB SCHEDULING ALGORITHM FOR COST OPTIMIZATION IN CLOUD COMPUTING ENVIRONMENT

Abstract. *Cloud is emerging day-by-day in the distributed environment and facing innumerable tackles, one amongst is Scheduling. Job scheduling is a vital task in cloud computing as the customer has to pay for used resources depends upon the time and cost. In existence, scheduling algorithms are established in the job length and the speed of the resources. The job execution in the cloud necessities multiple nodes to execute the single job. This approach is not sufficient to predict the optimal cost in the multi node execution platform. The cost of the network transmission is also not considered for scheduling cost. To overwhelm these complications, a Hierarchical Job Scheduling Algorithm (HJSA) is proposed. The major objective of the proposed work is to schedule the jobs with respect to the parameters of transmission cost, transfer cost, and execution cost of each job. Subsequently, it also foresees the multiple resources for job completion at the specific time. This is considered as the deadline of the workflow that provided by the customer in the cloud environment. To accomplish the deadline, the jobs are allocated using application splitting jobs to the small level task. A novel computational algorithm is introduced for predicting the optimum resources to complete the job with the defined cost and time. The experimental analysis depicts the lower time and cost, and also the higher reliability and throughput than the existing techniques.*

Keywords: *Cloud Computing, Job Scheduling, Hierarchical Job Scheduling Algorithm (HJSA), Deadline, Application Splitting Jobs, Computational Algorithm.*

JEL Classification: O30

1. Introduction

Cloud computing is a system of both parallel and distributed environments. A cloud comprising of group of virtualized and integrated computers. To utilize a cloud service, the most significant issues are (1) time slot, and (2) price determination. The cloud resources are provisioned into virtual machine instances by the cloud providers. Scheduling is a major task in a cloud computing environment. Based on trusting the correctness of status of resource information, the tasks are scheduled in distributed systems. The resource providers (i.e., Data centers) submits this type of information to a centralized database. This is further manageable to cloud schedulers, where the information is complete and accurate.

Cloud-based application service contains web hosting, social networking, content delivery and more. To quantify the provision's performance, the scheduling and allocation policies are defined in the cloud computing environment. Cloud necessitates the hardware, software, and network. Cloud user have the requirements of QoS and its application have different performance and workflow. Currently, there is a rapid increasing in the growth of cloud providers. An individual providers increase their own profit in multi cloud environment. This does not maintain the efficiency of other customers or providers. The resources are frequently provisioned isolated in a virtualized environment. Resource provisioning is said to be better using the following services such as IaaS, PaaS, and SaaS. Figure.1 describes the cloud scheduler that builds consistency checks on the cloud resources to ensure the resource optimization. This scheme is designed to authorize and schedule the job and provides the optimal resources. After authorizing the requests, it is further forwarded to the cloud scheduler by schedule request. Subsequently, the request returns back to the cloud user by service request.

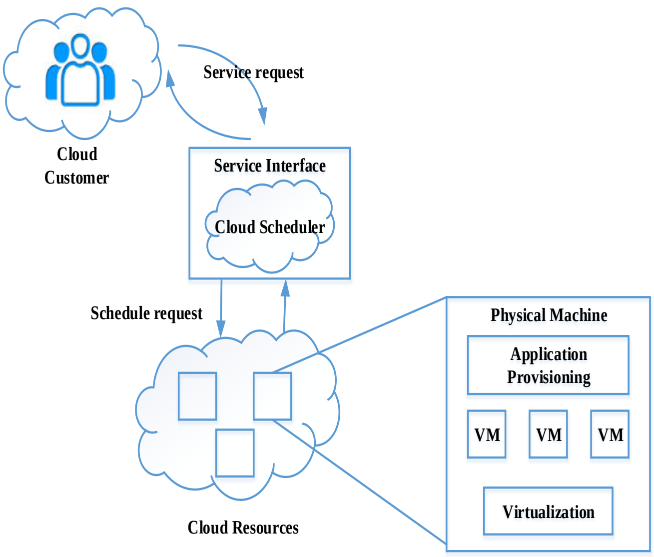

Figure 1. The cloud scheduler builds consistency checks on the cloud resources to ensure the resource optimization

The difficulties in the existing work are (a) the scheduling approach that only based on the job length and the speed of the resources, (b) needs several nodes for executing a single job, (c) the cost of transmission, and (d) not adequate to forecast the optimum cost in the platform of multiple node execution. The above mentioned complications are overwhelmed with the proposed work using a hierarchical job scheduling algorithm (HJSA).The job execution is done with the parameters such as transmission cost, transfer cost, and execution cost of each job. The deadline is predicted for a certain workflow by the cloud scheduler in the cloud environment. This process is accomplished by employing application splitting jobs to the smaller task level. A novel computational algorithm is proposed to forecast the best resource for job completion with defined cost and time.

The rest of the paper is systematized as follows. Section II briefly overview the relevant literature works in the concept of task/job scheduling techniques in the cloud computing environment. Section III involves the detailed explanation about the proposed hierarchical job scheduling methodology. Section IV describes the implementation details. Section V summarizes with a brief conclusive remark and discussion on future works.

2. Related Work

This section summarizes the relevant literature work in the workflow scheduling, price and time slot approaches for cloud services, and deadline constrained based workflow scheduling. *Fard et al* introduced a pricing model and truthful systems for dynamic scheduling. This mechanism was employed for a single task in multi cloud environment. The workflow scheduling was optimized in terms of makespan and monetary cost. Eventually, the complexity, truthfulness and the efficiency were analyzed[1].*Amir et al* described a dynamic market based model for grid resource allocation. An efficient bidding algorithm was also introduced based on the strategies of myopic equilibrium. Depends upon the success rate, this system offers a better submission of bids and implementation in the grid system[2].*Son et al* contributed a new price and time-slot negotiation (PTN)function for cloud service. The utility function was employed for characterizing the various agent’s satisfaction levels and time slots. The proposed burst mode was designed to improvise the negotiation speed and the aggregated use of price and time-slot. This provides an efficient tradeoff algorithm by managing the several proposals in a burst mode [3].

Liu et al presented the priority-based workload consolidation methodology. This schedule parallel jobs in data centers in which responsiveness were enhanced using under-utilized node computing. It comprises of foreground and background virtual machine (VM) tier. The performance enhances the accuracy of CPU usage estimation[4].*Rao et al* offered a new fuzzy control method for virtualized resource allocation. A two-layer quality of service (QoS) provisioning framework was designed based on the fuzzy controller. The fuzzy controller outperformed CPU, memory, and disk bandwidth allocation[5]. *Zaman et al* addressed the complications of dynamically delivering VM instances in clouds. Higher profit was produced by determining the VM allocation in terms of combinatorial auction based mechanism. This causes greater revenue in case of high demand cases and offers better performance in case of low demand cases[6].

Zhang et al provided a characterization of workload and machine heterogeneity in clusters. Harmony, a heterogeneity-aware framework was introduced for altering the number of machines. This yields high energy saving that improves the delay of task scheduling [7]. *Rodero et al* investigated an energy-aware online provisioning scheme for HPC applications .From the perspective of an energy, the workload-aware, dynamic, and proactive provisioning were explored[8].*Buyya et al* focused on the advancement of dynamic resource provisioning and allocation scheme. This considered the characteristics of QoS for allocation and scheduling methods. It accomplishes higher performance in terms of response time and cost saving consequences [9].

Xu et al proposed a novel strategy of low power task scheduling for large-scale cloud data centers. Winner tree was presented for creating the data nodes that considered as the leaf nodes of the tree. To minimize the consumption of energy, the optimum winner was chosen. Also, this system describes the comparison of task scheduling strategy where data center of the cloud manages several tasks subsequently[10].*Beaumont et al* considered the problem of resource allocation. This design both VMs allocation and also independent task scheduling difficulties. It helps to discover an allocation and accomplished the optimum throughput [11].*Warneke et al* intended the challenges for effective parallel processing in the cloud environments. Nephele framework was presented for manipulating the resource allocation. This enhances the resource utilization and minimize the processing cost[12].

Abirami et al described a novel linear scheduling of tasks and resources (LSTR), which was a scheduling algorithm. This algorithm focused to eliminate the conditions of starvation and the deadlock. The technique of virtualization yield better performance, system throughput and also the utilization of the cloud resources[13].*Polo et al* proposed a scheduling system for multi job MapReduce environment. This monitors the average length of task for all resources in all nodes simultaneously. It also used for estimating the expected completion time for each and every job. This system reduces the network traffic for a certain workload [14].*Zuo et al* established an integer programming method for the resource allocation. The solution guaranteed the user-level QoS and enhance the credibility of IaaS providers. Using a self-adaptive learning particle swarm optimization strategies, the task was scheduled[15].

Chiu et al presented an AVL-tree based cloud computing environment for organizing the cloud devices. Task scheduling was employed to offer better performance for the cloud environment[16].*Xue et al* proposed a load balancing optimization algorithm based on ant colony (ACO-LB). This provides an effective resource for certain tasks and enhance the resource utilization rate. It handles the load balance of VMs and reduce the makespan of task scheduling. Directed acyclic graph (DAG) was introduced to schedule the workflow. Cloudsim, a simulation platform was employed for implementing and comparing the ACO-LB with the several policies and algorithms [17].

Liu et al studied the strategy of cloud task scheduling. Before scheduling, clustering methodology was proposed for preprocessing the cloud resources. An earliest finish time duplication (EFTD) algorithm was presented to minimize the waiting time on the processor [18].*Liu et al* analyzed the mathematical model for both cloud task scheduling and VM allocation procedure in a cloud environment. For solving the optimization problem, the thermodynamic evolution algorithm was proposed. In this algorithm, the gene entropy and individual energy were described along with the integer encoding[19].*Lu et al* introduced a new heuristic workflow scheduling approach, namely, concurrent level based workflow scheduling (CLWS). The sequential tasks were optimized along with the time dependency using Markov decision process (MDP). It offers better and efficiency performance than the other algorithms [20].*Tang et al* presented a hierarchical reliability-driven scheduling (HRDS)approach for a grid system. It was observed as a hierarchical infrastructure and was executed by both local and global level scheduling [21].*Zhu et al* developed a new energy-aware rolling horizon scheduling algorithm, namely, EARH for the real-time, aperiodic, and independent tasks. This algorithm enhances the scheduling quality in various workloads. To test its performance, the EARH will implement in a real cloud environment [22].

3. A Hierarchical Job Scheduling Algorithm (HJSA) For Cost Optimization in Cloud Computing Environment

This section describes the proposed job scheduling methodology for optimizing the cost in a cloud environment.Figure.3 depicts the overall workflow of a hierarchical job scheduling approach in the cloud along with the properties of the virtual machines (VMs).The major components of the proposed work are explained as follows:

3.1 Job Scheduling

Scheduling is the group of procedures for managing the workflow that to be accomplished by a computer system. In the existing distributed environment, there has been several types of scheduling algorithm. Job scheduling is one among them and the benefits behind this is to attain better throughput of the system. In cloud, job scheduling is a multi-faceted complexity and cost optimization is a major objective in the large scale environment. The cloud provider assists to maximize their profit, whereas the customers optimize the cost. The jobs that require to be scheduled in the cloud increased proportionally as the number of cloud customers in the cloud environment increased. Therefore, a Hierarchical Job Scheduling Algorithm (HJSA) is introduced as crucial for scheduling the jobs on the cloud systems.

Many algorithms employed for job scheduling are service oriented and also be dissimilar in various environments. Job scheduling algorithms aim at decreasing the makespan of jobs with the reduction of resources proficiently. User applications will run on virtual machines in the cloud, where it dynamically allocates the resources. The load distribution system has to distribute jobs to the data centers in the VMs. Some applications may acquire high execution time to compute the jobs that are complex and necessitate a more memory storage. Nevertheless, a HJSA utilize better execution time, system throughput and also the optimum cost.

3.2 HJSA Algorithm

Job scheduling is a vital process and it has various stages such as cloud environment creation, job collection, estimating time and cost for each VMs, cost optimization, and job submission. Initially, the cloud environment is created and the parameters speed, workload, cost of execution, and cost of bandwidth (BW) are composed. The broker of data center determines the resources and further gathers information related to it. The job collection and acquisition of cost obtain from the customer. The process of a hierarchical job scheduling is diagrammatically explained in Figure.2. The distributed jobs are scheduled by the cloud scheduler in the virtual machines. For each and every application, the job scheduling comprises the hierarchical stages: the initial stage is to split the applications, and the next stage is to obtain the optimum VMs for the particular tasks, and finally the optimum resource is achieved by the cloud scheduler[21]. The following algorithm of the proposed work is explained by two important levels as:

- The first level of the proposed method is to choose the optimal VM for all data centers.
- The second level is to choose the optimal data center for the execution of job in the cloud.

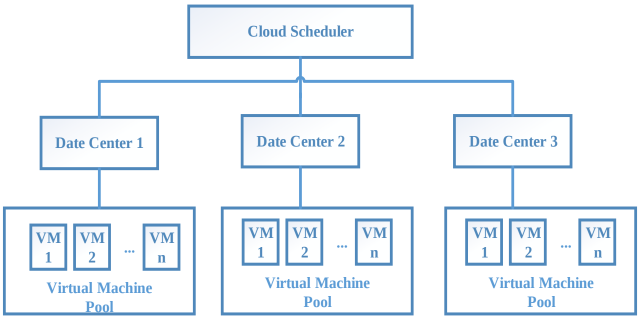

Figure 2. The process of obtaining the optimum virtual machine from data center of the cloud

A Hierarchical Job Scheduling Algorithm (HJSA)

Input: Application List AP_{List}
Output: Scheduled Datacenter List

```
While (AP_List ≠ empty)
    Jobs[n] ← AP_i;
    SV_List = Ø;//SV_List←Selected Virtual Machine List.
    D_List ← Datacenter List;
    While(Jobs ≠ Ø)
        D_OTCT [] = Ø;D_OCTC [] = Ø;
        For each datacenter D_List do
            VM_List ← D_List.getVM(i);
            V_NTT[] = Ø;V_ETC [] = Ø; V_EWT [] = Ø;
            V_OTCT [] =Ø; V_OCTC [] = Ø;
            For each Virtual Machine VM_List do
                V_NTT[j] = NTT for Job in j^th Virtual Machine;
                V_ETC [j] = ETC for Job in j^th Virtual Machine;
                V_EWT [j] = EWT for Job in j^th Virtual Machine;
                V_OTCT [j] = OTCT for Job in j^th VirtualMachine;
                V_OCTC [j] = OCTC for Job in the j^th VirtualMachine;
            End For
            Index X←OptimalSelection(V_OTCT[],V_OCTC []);
            D_OTCT [i] ← V_OTCT [x];
            D_OCTC [i] ← V_OCTC [x];
        End For
        Index X←OptimalSelection(D_OTCT [],D_OCTC[]);
        SV_List ←D_List.getVm(X);
        Remove Job from Jobs;
    End While
    Remove Application from AP_List;
End While
```

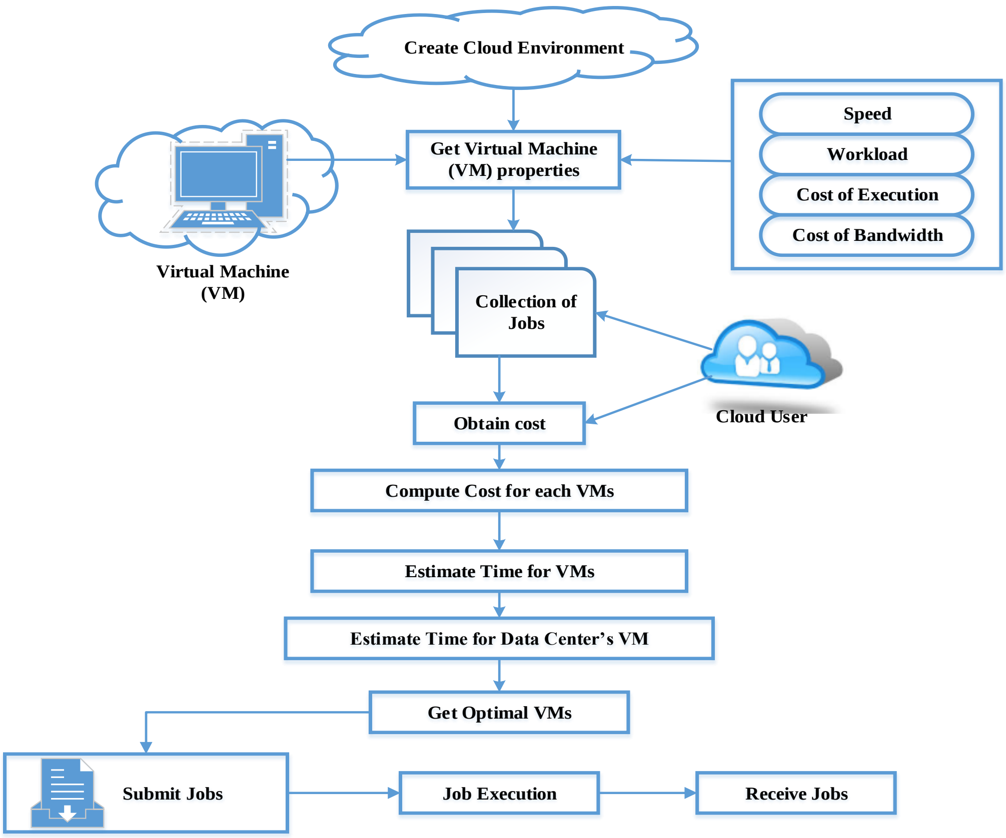

Figure 3. An overall flow diagram of a hierarchical job scheduling algorithm (HJSA) with respect to the properties of virtual machines (VMs) in cloud computing environment

The cost attained from the customer is estimated for each virtual machine. Subsequently, the time is computed for virtual machines and to the data centers of virtual machines. Lastly, the optimal virtual machine is preferred for the cloud environment. The optimized cost is calculated by the network transmission time, expected completion time, expected waiting time, and overall task completion time.

Table1. Description of the used variables in the Proposed Algorithm

S.NO.	VARIABLE	DESCRIPTION
1.	s_j	Size of job
2.	N_{Bw_i}	Network bandwidth
3.	S_i	Speed of the resource
4.	A_{s_i}	Allocated job size
5.	NP_i	Network Price
6.	P_{E_i}	Price for Execution
7.	P_{DSQ_i}	Price for data storage in queue

3.3 Optimization of Cost of the Cloud

For each virtual machines in data centers, the following metrics are estimated for accomplishing the optimal VM. It is measured by the given resources and jobs of the cloud.

3.3.1 Network Transmission Time

The network transmission time (NTT) is defined as the ratio of j^{th} job size to the i^{th} bandwidth of the network, where i is the given resources and j is the given cloud task or job.

$$NTT_{ij} = \frac{s_j}{N_{Bw_i}} \qquad (1)$$

3.3.2 Expected Time to Complete

The expected time to complete (ETC) for j^{th} job in i^{th} resource is defined as the ratio of j^{th} job size to the i^{th} speed of the resource.

$$ETC_{ij} = \frac{s_j\ (MI)}{S_i\ (MIPS)} \qquad (2)$$

where i is the given resources and j is the given cloud task or job.

3.3.3 Expected Waiting Time

The expected waiting time (EWT) is the summation of the ratio of the allocated job size of resources, which is not completed in *j*th job size to the *i*th speed of the resource, where *i* is the given resources and *j* is the given cloud task or job.

$$EWT_{ij} = \sum \frac{A_{s_i}}{S_i} \quad (3)$$

3.3.4 Overall Task Completion Time

The overall task completion time (OTCT) is denoted as the addition of network transmission time (NTT), expected time to complete (ETC), and expected waiting time (EWT).

$$OTCT = NTT + ETC + EWT \quad (4)$$

3.3.5 Overall Cost for Task Completion

The overall cost for task completion (OCTC) is defined as the addition of the *i*th resource unit network price along with the *j*th job network transmission time for *i*th resource, the *i*th resource unit price for execution with the expected time to complete the *j*th job in *i*th resource, and the *i*th resource unit price for data storage in queue with the expected waiting time for the *j*th job in *i*th resource.

$$OCTC_{ij} = \left(NP_i \times NTT_{ij}\right) + \left(P_{E_i} \times ETC_{ij}\right) + \left(P_{DSQ_i} \times EWT_{ij}\right) \quad (5)$$

Table 1 describes the variables that denoted in the proposed algorithm. After estimating the above parameters, the optimal virtual machine is selected with minimum cost and minimum time. Then, the optimal data center is chosen with minimum time and minimum cost. Ultimately, the status of selected virtual machine and data center are updated and thus the optimal cost is produced.

4. Performance Analysis

This section analysis the performance of cost optimization along with the job scheduling approach in the cloud in terms of throughput, reliability, optimal cost, load distribution, and the average execution time. For modelling the framework of a cloud computing environment, a CloudSim tool is used for executing the proposed algorithm. CloudSim is a novel, discovered, and expandable simulation framework, which initially test the performance of time efficiency, reliability, and throughput.

4.1 Optimal Execution Time

The execution time of a given job is denoted as the time consumed by the system that executes the task. Otherwise, considered as the CPU time and it also includes the execution of run-time or services of the system. Figure.4 depicts the performance analysis of execution time with respect to the job size to existing Energy-Aware Rolling Horizon (EARH)scheduling algorithm [22]and the proposed Hierarchical Job Scheduling Algorithm (HJSA). The existing EARH is for real-time independent tasks in a cloud, whereas the proposed HJSA is for independent jobs in a cloud. The execution time of HJSA is lesser than the EARH, where HJSA gradually increase the time consumption.

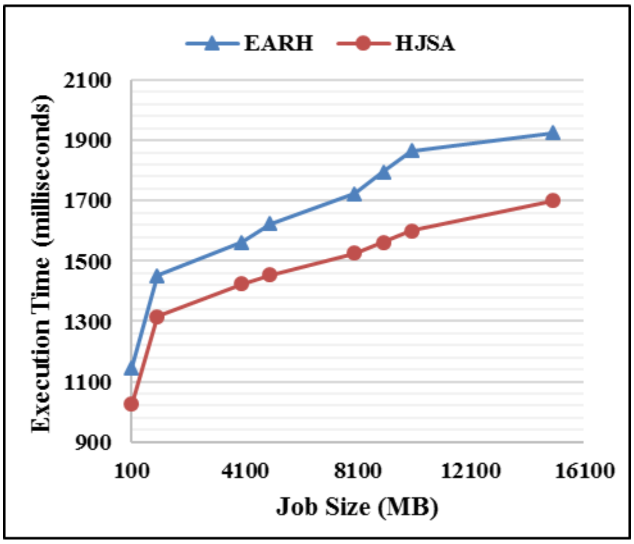

Figure 4. The result of execution time in milliseconds with respect to the job size in MB for EARH (existing) and HJSA (proposed) method

4.2 Throughput

Throughput is the number of data transfer from one location to another location in a specified quantity of time. The data transfer rate is used for measuring the performance of disk drives and networks in relation to the throughput. Figure.5 defines the throughput of a system with respect to the number of virtual machines for existing EARH [22]and the proposed HJSA. The throughput of HJSA is greater than the EARH approach.

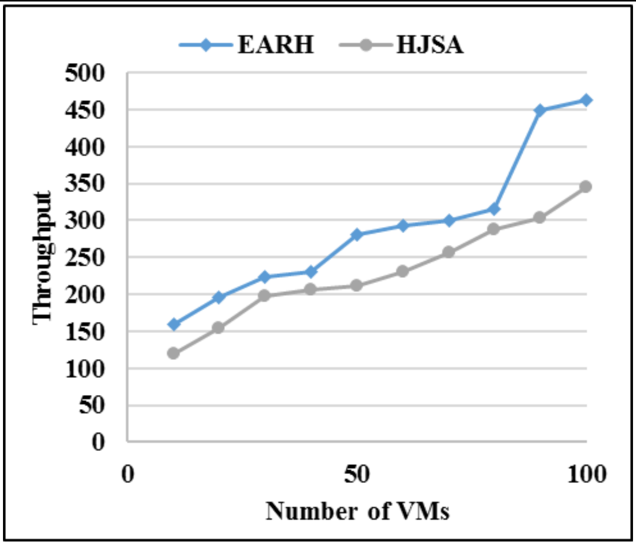

Figure 5. The result of throughput with respect to the number of VMs for EARH (existing) and HJSA (proposed) method

4.3 Optimal Cost

Figure.6 shows the optimal cost with respect to the makespan for existing EARH [22]and the proposed HJSA. The optimal cost of HJSA is lesser than the EARH approach, where the shown proposed algorithm gradually decreases in cost.

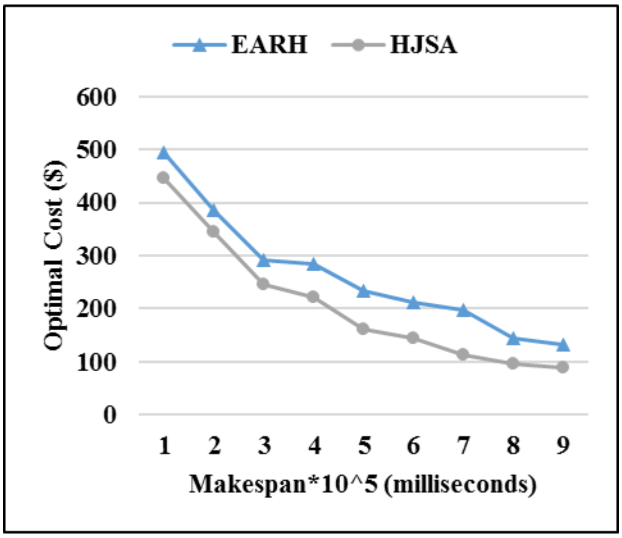

Figure 6. The result of optimal cost with respect to the makespan for EARH (existing) and HJSA (proposed) method

4.4 Reliability

The term reliability is referred as the component of computer-related hardware or software that perform persistently depends upon its specifications.

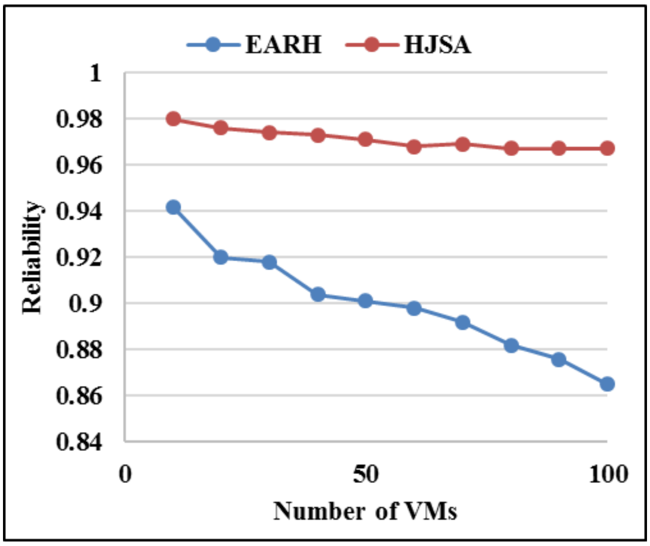

Figure 7. The result of reliability with respect to the number of VMs for EARH (existing) and HJSA (proposed) method

The cloud system reliability (SR) can be estimated by averaging the reliability of all applications, and is termed as:

$$SR = \frac{\sum_{n=1}^{k} R[E_{A_n}]}{k} \tag{6}$$

where $R[E_{A_n}]$ is the distribution of the reliability probability of application A_n, and k is the number of applications [21]. Figure.7 describes the reliability of a system with respect to the number of VMs for existing EARH[22]and the proposed HJSA. The reliability of HJSA is greater than the EARH approach.

4.5 Average Load Distribution

Figure.8 expresses the average load distribution with respect to the number of virtual machines. A load in the proposed HJSA is evenly distributed to the virtual machines, here, VM1, VM2, VM3, VM4, VM5, and VM6. For VM1 to VM4 has the constant load distribution, whereas in VM5 and VM6 has the value of 16 as its load distribution value.

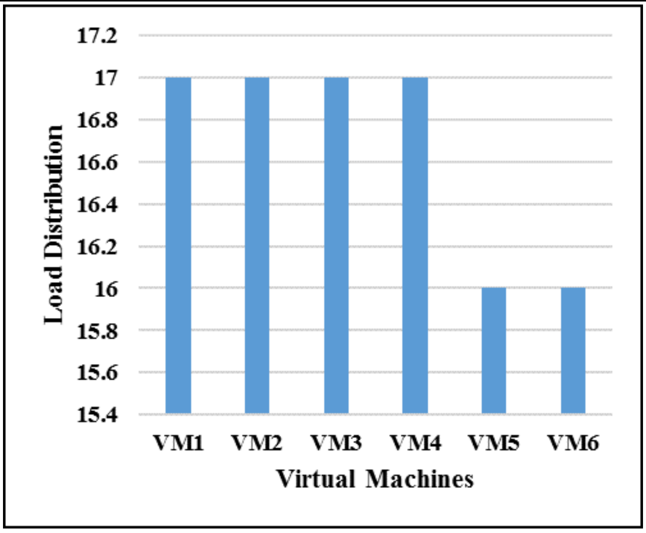

Figure 8. The result of average load distribution with respect to the virtual machines (VMs)

5. Conclusion and Future Work

Cloud computing paradigm assures a cost-efficient solution for the business application via a pay-as-you-go pricing model. Job scheduling is the most important challenge in the parallel and distributed environment. In the proposed work, a Hierarchical Job Scheduling Algorithm (HJSA) is introduced. This approach overcomes the prediction of optimal cost in the multi node execution platform. Correspondingly, the cost of the network transmission (i.e., not considered for scheduling cost) is also overwhelmed by the proposed system. It also forecasts the multiple resources for job completion at the specific time, which considered as the deadline of the workflow. The jobs are allocated to attain the deadline using application splitting jobs to the small level task. A novel computational algorithm is presented for predicting the optimum resources to complete the job with the defined cost and time. The performance analysis depicts the lower time and cost, and also the higher reliability and throughput than the existing techniques.

In future enhancement, the proposed work will extend with the Fault Tolerant Workflow Scheduling algorithm (FTWS). Based on the priority of the tasks, it will offer fault tolerance by employing replication and resubmission of tasks.

REFERENCES

[1] **Fard, H. M., *et al.*,(2013), *A Truthful Dynamic Workflow Scheduling Mechanism for Commercial Multicloud Environments*;** *Parallel and Distributed Systems, IEEE Transactions on,***24**(6): 1203-1212;

[2] **Danak, A. and Mannor, S.(2011), *Efficient Bidding in Dynamic Grid Markets;*** *Parallel and Distributed Systems, IEEE Transactions on,* **22** (9) :1483-1496;

[3] **Son, S. and Sim, K. M. (2012),*A Price-and-Time-Slot-Negotiation Mechanism for Cloud Service Reservations*;** *Systems, Man and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* **42** (3), 713-728;

[4] **Liu, X. *et al.*,(2013), *Priority-Based Consolidation of Parallel Workloads in the Cloud;*** *Parallel and Distributed Systems, IEEE Transactions on,* **24** (9), 1874-1883;

[5] **Rao, J. *et al.*,(2013), *Qos Guarantees and Service Differentiation for Dynamic Cloud Applications*;** *Network and Service Management, IEEE Transactions on,* **10** (1), 43-55;

[6] **Zaman, S. and Grosu, D.(2013) , *A Combinatorial Auction-Based Mechanism for Dynamic VM Provisioning and Allocation in Clouds*;** *IEEE Transactions on Cloud Computing,* **1**(2), pp. 129 -141;

[7] **Zhang, Q. *et al.*, (2013) *Dynamic Heterogeneity-Aware Resource Provisioning in the Cloud;*** Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on, pp. 510-519;

[8] **Rodero, I. *et al.*,(2010), *Energy-efficient Application-aware Online Provisioning for Virtualized Clouds and Data Centers;*** In: *Green Computing Conference, 2010 International*, pp. 31-45;

[9] **Buyya, R.*et al.*,(2010), *Energy-Efficient Management of Data Center Resources for Cloud Computing: A Vision, Architectural Elements and Open Challenges***; *arXiv preprint arXiv:1006.0308*;

[10] **Xu, X. *et al.*,(2013), *Low-power Task Scheduling Algorithm for Large-scale Cloud Data Centers***; *Systems Engineering and Electronics, Journal of,* **24** (5), 870-878;

[11] **Beaumont, O.*et al.*,(2013) , *Heterogeneous Resource Allocation under Degree Constraints;*** *Parallel and Distributed Systems, IEEE Transactions on,* **24** (5), 926-937;

[12] **Warneke D. and Kao, O. (2011) , *Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud*;** *Parallel and Distributed Systems, IEEE Transactions on,* **22** (6), 985-997;

[13] **Abirami S.and Ramanathan,S. (2012) , *Linear Scheduling Strategy for Resource Allocation in Cloud Environment;*** *International Journal on Cloud Computing: Services and Architecture (IJCCSA),* **2**(1), 9-17;

[14] **Polo, J.*et al.*, (2013),*Deadline-Based MapReduce Workload Management;*** *IEEE Transactions on Network and Service Management,* vol. **10** (2), 231-244;

[15] **Zuo,X.*et al.*,(2013), *Self-Adaptive Learning PSO-Based Deadline Constrained Task Scheduling for Hybrid IaaS Cloud;*** *IEEE Transactions on Automation Science and Engineering,***11**(2), 564 - 573;

[16] **Chiu, C.-F. *et al.*,(2014), *Task Scheduling Based on Load Approximation in Cloud Computing Environment*;** In: *Future Information Technology*, ed: Springer, pp. 803-808;

[17] **Xue, S. *et al.*,(2014) , *An ACO-LB Algorithm for Task Scheduling in the Cloud Environment;*** *Journal of Software,* **9** (2), 466-473;

[18] **Liu, Z. *et al.*, (2014), *Resource Preprocessing and Optimal Task Scheduling in Cloud Computing Environments;*** *Concurrency and Computation: Practice and Experience*; **27**(13), 3461–3482;

[19] **LIU, B.-x. *et al.*,(2013) , *Task Scheduling and Virtual Machine Allocation Strategy Based on Thermodynamics Evolutionary Algorithm in Cloud Computing Environment*;** *Science Technology and Engineering,* vol. 15,p. 53;

[20] **Lu, G. *et al.*,(2014) , *QoS Constraint Based Workflow Scheduling for Cloud Computing Services*;** *Journal of Software,* **9**(4), 926-930;

[21] **Tang, X.*et al.*,(2012) ,*A Hierarchical Reliability-Driven Scheduling Algorithm in Grid Systems*;** *Journal of Parallel and Distributed Computing,* **72** (4), 525-535;

[22] **Yang,L.*et al.*,(2014.), *Real-Time Tasks Oriented Energy-Aware Scheduling in Virtualized Clouds*;** *IEEE Transactions On Cloud Computing*, **2**(2), 168 -180.