

Predictive Probabilistic Resource Availability based Cloud Workflow Scheduling (PPRA)

*Chitra S¹, Dr. Prashanth C S R²

¹(Computer Science and Engineering, New Horizon College of Engineering/ VTU, India)

²(Computer Science and Engineering, New Horizon College of Engineering/ VTU, India)

Abstract: Cloud Computing provides access to a shared pool of computing resources such as servers, storage, computer networks and services, which can be rapidly provisioned and released, for the execution of various scientific and business applications. Scheduling scientific workflows modeled by Directed Acyclic Graphs is an NP complete problem. In cloud environment, there are fluctuations in resource availability due to shared resources and vastly varying workloads. The performance variations in virtual machines, have an impact on task execution times and data transfer times. This has a great impact on the performance of scheduling algorithms. Schedulers should map tasks to cloud resources considering the effective utilization of underlying resources. We propose a new static workflow scheduling algorithm called Predictive Probabilistic Resource Availability based Cloud Workflow Scheduling (PPRA) with the objective of minimizing makespan considering the probability of resource availability of cloud resources. This algorithm is compared with existing algorithms which assume full resource availability while making scheduling decisions. The proposed algorithm is found to be more reliable and performing better than existing HEFT and ECTS algorithms, in terms of minimized makespan.

Keywords: Cloud Computing, Directed Acyclic Graphs, Makespan, Resource Availability probability, Scientific Workflows

I. Introduction

Cloud computing is a collection of virtualized computing resources that are dynamically provisioned, and released based on service-level agreements established between the service provider and consumers through negotiation [1]. Cloud computing delivers infrastructure, platform, and software as subscription-based services in a pay-as-you-go model to consumers. These services are called Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) respectively[2]. It provides access to a shared pool of computing resources such as computer networks, servers, storage and services, which can be rapidly provisioned and released, for various applications[3]. Cloud computing technology has the characteristics such as massive scalability, dynamic infrastructure, universal access, fine grained usage controls and pricing, standardized platforms and Management Support Services[4]. The High performance applications like Weather Forecasting, Earthquake Analysis, Bio Informatics, Astronomy, and a host of other engineering and scientific applications that require enormous computing capabilities can be processed in cloud environment as a series of computational or data manipulation steps, or workflows with dependencies between tasks[5]. These applications can be CPU intensive, memory intensive or I/O intensive depending on the nature of the problem. Workflows are modeled as Directed Acyclic Graphs (DAGs), where users' computation is represented by nodes and data or control flow dependencies between the jobs are represented by directed edges [6]. Scheduling algorithms are an essential component of workflow management systems that orchestrate the execution of tasks in a set of compute resources while preserving the data dependencies. Schedulers map tasks to cloud resources considering the effective utilization of underlying resources. Cloud resources include virtualized compute resources called Virtual Machines (VMs) having a predefined CPU, memory, storage, and bandwidth capacity for lease. Different resource bundles (i.e. VM types) are available at varying prices. They can be elastically acquired and released and are generally charged per time frame or billing period[7]. Scheduling algorithms should also take into consideration the dynamic nature of cloud platforms and the uncertainties that arise due to the performance variations of cloud resources. There could also be delay introduced due to VM provisioning and deprovisioning. IaaS cloud resources can be effectively utilized by predicting the future workload or the future resource utilization pattern[8].

We propose a new static scheduling algorithm called Predictive Probabilistic Resource Availability based Cloud Workflow Scheduling (PPRA). It allocates tasks to the resource that has earliest finish time considering the resource availability probability. The probability of resource availability of each Virtual Machine for a certain prediction interval at every instant of time is considered based on the resource usage prediction models. The scheduling decision is based on earliest finish time that depends on the resource availability factor. The performance of the proposed algorithm is studied by comparing with existing

Heterogeneous Earliest Finish Time (HEFT)[9] and Earliest Completion Time based Scheduling(ECTS)[10] algorithms. Section 2 specifies problem statement. Section 3 gives an overview of application model, resource model and workflow scheduling algorithms as background. Section 4 contains Related work. In Section 5 the proposed Predictive Probabilistic Resource Availability based Cloud Workflow Scheduling algorithm(PPRA) is discussed. In Section 6, Simulation Study is discussed. Results are discussed in Section 7. In Section 8 conclusions are given.

II. Problem Statement

Precedence constrained parallel applications (workflows) require large amounts of bandwidth and powerful computational resources. Effective and efficient scheduling algorithm is very essential to use the cloud resource model for executing such data and compute intensive applications. There are certain existing static task scheduling algorithms in literature such as HEFT, ECTS that schedule tasks based on earliest finish time. They do not consider the temporal availability of resources in making scheduling decisions. This contributes to delays in task execution under reduced resource availability. This has an impact on the makespan, thus affecting performance and reliability.

The problem can formally be defined as "To develop efficient heuristic to schedule the tasks representing a parallel application, modeled by Directed Acyclic Graph on a network of heterogeneous processors such that precedence constraints are satisfied and minimizing overall execution time (finish time) considering the probability of resource availability".

III. Background

In IaaS clouds, large scale scientific workflows can be deployed on leased compute resources called Virtual Machines (VMs) that have a predefined CPU, memory, storage, and bandwidth. Different VM types are available at varying prices to suit a wide range of application needs. IaaS providers also offer storage services and network infrastructure to transport data in, out, and within their facilities[11]. Figure 1 shows the reference cloud architecture consisting of physical resource such as servers, storage and network. With virtualization, there are virtual machines, data stores and virtual switches, which offers elasticity of resources. These resources can be pooled.

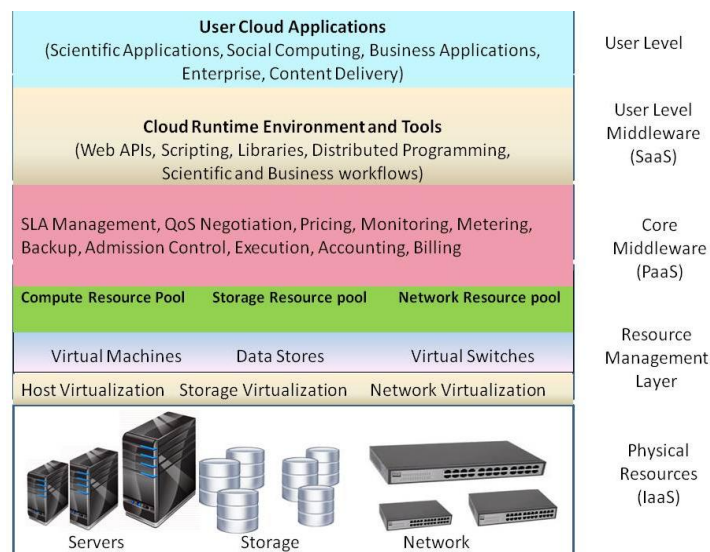


Figure 1: Cloud Architecture

Characteristics such as the shared nature, virtualization, and heterogeneity of non-virtualized hardware in clouds result in a variability in the performance of resources. For example, VMs deployed in cloud data centres do not exhibit a stable performance in terms of execution times. Performance variation is also observed in the network resources which affects data transfer times. It is difficult for schedulers to estimate runtimes and make accurate scheduling decisions to fulfill QoS requirements. The type of host where the VM is deployed, its current load and utilization of resources, the overhead of the virtualization software, and network congestion all have an effect on application execution times. Schedulers need to be aware of this variability in order to recover from unexpected delays and achieve their performance and cost objectives [11].

Workflows are modeled by Directed Acyclic Graphs DAGs with a set of task nodes and set of edges representing the dependencies representing data flow from one to another. These applications can be CPU, memory, or I/O intensive or a combination of these, depending on the nature of the problem. The Scheduling strategies are classified as Static, Dynamic and Hybrid. In the Static algorithms, the task to VM mapping is done in advance and will not be altered during runtime. Some static algorithms adapt to the dynamic nature of cloud environments by using runtime prediction, probabilistic QoS guarantees, and resource performance variability aware strategies. In Dynamic algorithms task to VM mapping is during runtime based on the current state of the system and the workflow execution. In Hybrid algorithms, there are two approaches: In runtime refinement, first a static task assignment is determined before runtime. This may change during the execution based on current state of system, and in the sub-workflow static approach, static decisions are made for a group of tasks dynamically[11].

IV. Related Work

A. Heterogeneous Earliest Finish Time (HEFT) [9]

HEFT algorithm has two phases: Task prioritization and Processor selection. Tasks are prioritized for scheduling based on upward ranking. Tasks are arranged in decreasing order of upward rank to generate the task priority list. Tie-break is random. In the processor selection phase, the selected task is assigned to processor with minimum Earliest Finish Time (EFT) using insertion-based approach. There is a possible insertion of a task in an earlier idle time slot between two already scheduled tasks on a processor, however ensuring that precedence constraints are maintained. The complexity of the HEFT algorithm is $O(e \times r)$ where e is the number of edges and r is the number of processors. For a dense graph, it is $O(v^2 \times r)$ where v is number of nodes and r is the number of processors.

B. Expected Completion Time based Scheduling Algorithm for Heterogeneous Processors (ECTS)[10]

In this paper, the authors propose a list scheduling heuristic for a bounded number of processors based on the expected completion time of task for heterogeneous distributed environment. The application is modelled by Directed Acyclic Graph (DAG) and the target execution system is heterogeneous distributed environment consisting of a certain number of processors in a fully connected network topology. This algorithm has Task Prioritization and Processor selection phases. The Task Prioritization phase consists of two stages. In the first stage the priority of each task at each level is computed based on their average computation cost and maximum data arrival cost, and in the second stage, tasks are selected from all levels based on their priority. The Average Computation Cost (ACC) of a task is calculated as the ratio of sum of computation cost of the task on each processor by the number of available processors. Maximum Data Arrival Cost (MDAC) of a task is the Maximum Data Arrival Cost (MDAC) of a task is the maximum time taken for the task to receive data among its parents. In the task selection stage, all tasks at each level are sorted in non-increasing order of their Expected Completion Time value, and then prioritized. The Expected Completion Time (ECT) of a task is sum of the average computation cost and maximum data arrival cost of the task. In the second phase, the selected task is allocated to the processor on which EFT is minimum using the insertion-based scheduling policy. The task is assigned to processor which minimizes its EFT compared to other processors. If two processors are producing same EFT for a selected task, then a processor can be selected randomly, or that which is lightly loaded, or that which is based on minimum processor utilization. The comparison metrics used for evaluation of the algorithm are Schedule Length Ratio (SLR) and Speedup.

C. STAR:SLA-Aware Autonomic Management of Cloud Resources [12]

In Cloud environment, uncertainty and dispersion of resources pose challenges in the efficient management of resources, caused due to heterogeneity, dynamism, failures leading to performance degradation. Autonomic system keep the system stable in unpredictable conditions and adapt quickly in new environmental conditions like software, hardware failures etc. SLA-aware autonomic Technique for Allocation of Resources (STAR) for effective scheduling of resources was developed to propose an autonomic resource management technique for execution of heterogeneous workloads by considering self-management, optimization of QoS parameters such as execution time, cost, latency, reliability and availability to analyze the impact of QoS parameters on SLA violation rate, reduction in SLA violation rate and improved user satisfaction by fulfilling their QoS requirements, implementation and performance evaluation in cloud environment.

D. Efficient Resources Provisioning Based on Load Forecasting in Cloud [13]

Cloud computing environment should provide reliable QoS defined in terms of Service Level Agreements (SLA) while maximizing resources utilization. Modern applications often experience highly variable workloads causing dynamic resources usage patterns. The consolidation of VMs can lead to performance degradation when an application encounters an increasing demand resulting in an unexpected rise

of resources usage. This may lead to SLA violation increasing response times, time outs, or failures. Overprovisioning may help to ensure SLA, but it leads to inefficiency when the load decreases. The optimal strategy is to timely adjust resources provisioning according to the actual demands of the application. The precondition of this approach is to find out the future workload. The proposed multi-step-ahead load prediction method, KSwSVR, is based on statistical learning technology, support vector regression (SVR), which is suitable for the complex and dynamic characteristics of the cloud computing environment. KSwSVR always has the minimum prediction error and is very stable. VMs are added, released, or migrated according to variation of load. Each process involves significant overhead. If the load changes dramatically, there will be frequent SLA violations. KSwSVR, based on statistical learning technology is suitable for the complex and dynamic characteristics of the cloud computing environment. It integrates an improved support vector regression algorithm that gives more weightage to important historical data and Kalman smoothing technology is employed to eliminate the noise of resources usage data coming from measurement error.

E. Resource Management in Clouds[14]

Due to the scale of modern data centers; the heterogeneity of resource types and their interdependencies; the variability and unpredictability of the load; as well as the range of objectives of the different actors in a cloud ecosystem, resource management is hard. Resource Management is the process of allocating servers, storage, networking to a set of applications, to meet application performance objectives, the infrastructure providers and the users of the cloud resources. Resources should be used effectively satisfying SLAs with users. Cloud resources are computer resources in the form of Physical Machines (PMs), each comprised of one or more processors, memory, network interface and local I/O, which together provide the computational capacity of a cloud environment hosting VMS. The physical machines must be interconnected with a high-bandwidth network, typically built upon Gigabit Ethernet or InfiniBand technologies. For all parallel applications deployed on cloud, communication overhead imposed by the data center networking technologies and networking protocols and persistent storage services of different types, ranging from virtual disks and database services to object stores, each service having varying levels of data consistency guarantees and reliability have an impact on overall performance. A hypervisor deployed on a physical machine will control the sharing of CPU, local I/O and network interfaces between the virtual machines it hosts and also dynamically adjust the share of resources allocated to virtual machines.

F. A Resource Prediction Model for Virtualization Servers[15]

As cloud infrastructures use virtual resources on demand, monitoring and predicting resources becomes essential. Resource prediction is used to forecast short-term resource usages based on historical. Virtual infrastructure resource usage is dynamic in nature. It is also difficult to monitor and predict its capacity resources. The authors have used Markov chain based prediction model to predict resource usage. They have used state based probability model for predicting CPU usage ranging between 0 and 100%. They have partitioned the measurement level in intervals and defined each interval as states of their model.

V. Proposed Work

We propose a new static workflow scheduling algorithm called Predictive Probabilistic Resource Availability based Cloud Workflow Scheduling (PPRA). It allocates tasks to the resource that has Earliest Finish Time(EFT) considering the resource availability probability. The probability of resource availability of each Virtual Machine for a certain prediction interval at every instant of time is considered based on the resource usage prediction models. The scheduling decision is based on earliest finish time that depends on the resource availability factor. For the duration of execution of task, the probabilities of resource availability are averaged. The EFT is adjusted depending on the probability of resource availability as the task execution times increase with reduced resource availability. The earliest finish time of task on specific VM is adjusted by a factor which is the ratio of maximum of the average probabilities of various VMs to the average probability of the current virtual machine VM. The VM on which the Earliest Finish Time (EFT) is the minimum, considering the probability of resource availability, will be chosen for task allocation.

The proposed PPRA comprises of two phases namely, Task Prioritization phase and Processor Selection phase. The tasks are grouped into different levels based on the precedence constraints and dependencies. The Expected Completion Time ECT is based on the Average Computation Cost ACC and Maximum Data Arrival Cost MDAC of each task. In each level the tasks are arranged in the decreasing order of ECT. The tasks are selected from each level according to their priority and a task sequence is produced for execution.

Resource provisioning determines the number of VMs required, their type, configuration. Also the probability of resource availability of each VM is predicted for a certain interval that the application is to be executed. In the scheduling or task allocation stage, the selected task sequence is assigned to the VMs that

minimize their execution time taking into account the probability of resource availability factor, using the insertion-based scheduling policy with precedence constraints among tasks maintained. In the VM selection phase, for each task, the probability of resource availability values for each VM, during the task execution interval are averaged. For the specific task, EFT considering the resource availability probability is calculated using the formulae:

$$EFT_{pr}(n_i, p_j) = EFT(n_i, p_j) * pr_max / pr(n_i, p_j)$$

The task is allocated to the VM with the least EFT.

ALGORITHM PPRA

BEGIN PPRA

// N is set of vertices of DAG

// P is set of VMs

// $RdyTskLst$ is the ready task list

// $pr(vm_j)$ is probability of availability of VM vm_j for a certain prediction interval

// $avgppa(vm_j)$ average probability of vm availability factor for the duration of task execution time for the task

// pr_max is the maximum of probability of availability of all the processors for a given task n_i .

For each level Li in the DAG do

For each task ti in the level Li do

Find Average Computation Cost (ACC) as $ACC(t_i) = \bar{w}_i = \sum_{j=1}^m w_{ij} / m$

Find Maximum Data Arrival Cost (MDC) as $MDA(t_j) = \max_{t_i \in pred(t_j)} c_{ij}$

Find Expected Completion Time (ECT) as $ECT(t_j) = ACC(t_j) + MDAC(t_j)$

End For

Sort all tasks in descending order of ECT

Assign levelwise priority for each task

End For

Return the priority of tasks

End

Generate probability of availability matrix pr with random samples from Normal Distribution with mean 0.5 and standard deviation 0.2

$RdyTskLst \leftarrow StartNode$

While $RdyTskLst$ is NOT NULL do

$ni \leftarrow Node$ in the $RdyTskLst$

For each VM vm_j in P

Compute Earliest Start Time (EST) as $EST(n_i, p_j) = \max (T_{avail[j], \max_{n_m \in pred(n_i)} (EFT(n_m, vm_k) + c_{m,i}))$

Compute Earliest Finish Time of t_j on vm_k using the insertion based policy as $EFT(n_i, vm_j) = w_{i,j} + EST(n_i, vm_j)$

Computer Earliest Finish Time of t_j on vm_k considering probability of resource availability as

$$EFT_{pr}(n_i, p_j) = EFT(n_i, p_j) * pr_max / pr(n_i, p_j)$$

where pr_max is the maximum among the various probability of VM availability

End For

Assign task ti to the vm vm_k that minimizes its finish time

End For

End

Update $T_Avail[p_j]$ and $RdyTskLst$

End While

END PPRA

VI. Simulation Study/Experimental Setup

Our simulated framework first executes Random Directed A-cyclic Graph Generator Program [16] for generation of Random DAGs that model parallel applications. Based on the number of tasks nodes in the application and number of VMs requested for the execution of application, the maximum number of edges was determined as $maxnodes = nodes * (nodes - 1) / 2$. For our performance study, the number of edges for a certain DAG was generated randomly in the range $maxnodes/3$ to $maxnode/2$. Communication to computation cost ratio (CCR) is the ratio of the average communication cost to the average computation cost. \bar{w}_{DAG} is chosen randomly. \bar{w}_i is selected randomly from a uniform distribution with range $[0.2 * \bar{w}_{DAG}, \bar{w}_{DAG}]$ where \bar{w}_{DAG} is the average computation cost of the given graph and is set randomly. Heterogeneity factor β for processor speeds depends basically on the range percentage of computation costs on processors, higher the percentage, higher the

degree of heterogeneity. This is taken from the set [0.1,0.5,1] as maximum 1 to have highest heterogeneity in task execution times.

The computation cost of each task t_i on each Virtual machine vm_j is randomly set from the range given in equation

$$\bar{w}_i * \left(1 - \frac{\beta}{2}\right) \leq w_{ij} \leq \bar{w}_i * \left(1 + \frac{\beta}{2}\right)$$

Communication cost weighted sparse matrix was generated randomly, based on number of task nodes. If the graph has multiple entry nodes, then dummy edges with weights 0 are added from dummy node that forms the entry node to the multiple entry nodes. Likewise, if the graph has multiple exit nodes, then dummy exit node is included with dummy edges from the multiple exit nodes with weights 0.

Graphs of varying number of nodes 10,20,30,40,50 were generated for testing the performance of the proposed algorithm PPRA, followed by execution of the existing algorithms specified in related work for comparison such as HEFT and ECTS. Resource availability probability for the given number of VMs is generated for a prediction interval of 1500 time units choosing random samples from Normal distribution with mean 0.5 and standard deviation 0.2. Application Graphs such as Montage, Epigenomics, Cybershake and Gauss Elimination graphs were generated for testing the proposed PPRA algorithm performance and comparing with existing algorithms HEFT and ECTS.

VII. Results and Discussion

The performance of the proposed algorithm PPRA is studied in comparison with existing algorithms HEFT and ECTS that do not consider resource availability factor based on the following performance metrics.

(i) Makespan : Makespan or Schedule Length is the Actual Completion Time of exit task node in the DAG representing a parallel application.

(ii) Schedule Length Ratio (SLR) : A large set of Random DAGs with different properties are used for experiment. So the Schedule Length is normalized to a lower bound called Schedule Length Ratio (SLR) which is the ratio of makespan to summation of minimum computation cost of critical path nodes. It is given by the expression

$$SLR = \frac{\text{makespan}}{\sum_{n_i \in CP_{MIN}} \min_{p_j \in Q} \{w_{i,j}\}}$$

(iii) Speedup : It is ratio of sequential execution time of parallel application to its parallel execution time. It is the ratio of minimum of summation of computation costs of tasks on processors to makespan of the schedule. It is given by the expression

$$Speedup = \frac{\min_{p_j \in Q} \{\sum_{n_i \in V} w_{i,j}\}}{\text{makespan}}$$

The comparative results of the proposed algorithm PPRA with HEFT and ECTS are presented. Randomly generated DAGs of sizes 10, 20, 30, 40 and 50 nodes were generated and used to test the performance of PPRA algorithm, while comparing with HEFT and ECTS. The PPRA algorithm was also tested with real life application scientific workflows such as Montage in area of Astronomy, Epigenomics in the area of Bio-Informatics, Cybershake in the area of Earthquakes and Gauss graph. The algorithms were executed to determine the performance metrics of Average Makespan, Average Schedule Length Ratio and Average Speedup for PPRA, HEFT and ECTS algorithms.

Test Case 1: The performance of proposed algorithm PPRA is studied for random Directed Acyclic Graphs of various sizes of task nodes and compared with existing HEFT and ECTS algorithms. Since HEFT is robust, has low running time, and is tested to give stable performance over wide range of graph structures[9], it is considered for comparison. ECTS algorithm performs better than HEFT and employs level sorting in task prioritization phase which facilitates parallel execution of task nodes of same level and hence is also considered for performance comparison. Randomly generated DAGs of sizes 10,20,30,40 and 50 nodes are considered for performance testing of algorithms. For each random DAG generated, our algorithm was executed 200 times with different probability of resource availability. The availability matrix was generated for a certain prediction interval by with random samples from normal distribution. The probable delay due to availability factor being considered for HEFT and ECTS schedules were determined. From the results, we observe makespan of RSRA is better than HEFT and ECTS in this scenario which considers processor availability factor. The proposed PPRA algorithm is more reliable as the resource availability factor is incorporated in our scheduling decisions while yet meeting the objective of minimizing the makespan and hence the cost. Comparative results for the performance metrics Makespan, Schedule Length Ratio and Speedup are shown below. Figure 2 shows the comparison of RSPA with HEFT and ECTS based on Average Makespan. Figure 3 shows the comparison based on Average SLR and Figure 4 shows the comparison based on Average Speedup.

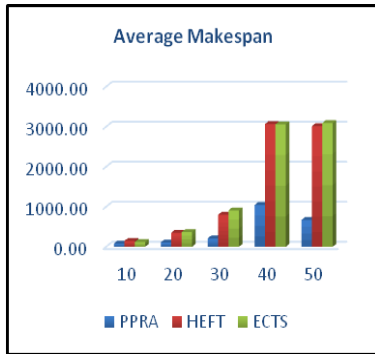


Figure 2 :Comparison based on Average Makespan

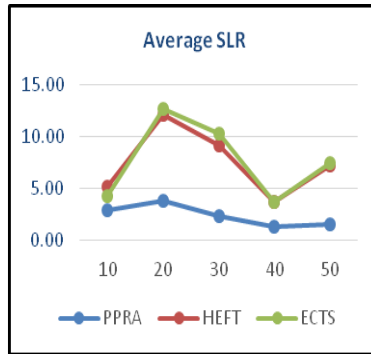


Figure 3: Comparison based on Average SLR

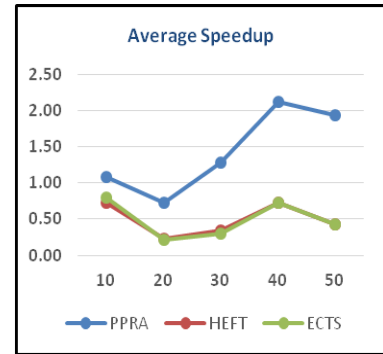


Figure 4 : Comparison based on Average Speedup

Test Case 2:The performance of proposed PPRA algorithm is compared with HEFT and ECTS for the Gauss Elimination Sample Graph [9] shown in Figure 5. The results in terms of Average Makespan, Average SLR and Average Speedup are shown in Figures 6,7,8 respectively. The total number of tasks of size m is $(m^2 + m - 2) / 2$. The sample graph was generated with the sequential algorithm for Gauss Elimination for size 5 consisting of total number of nodes 14.

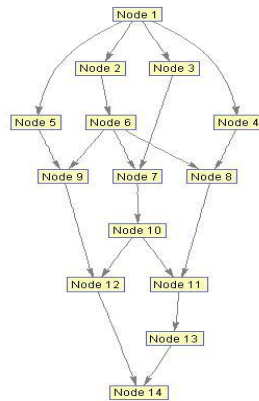


Figure 5 : Gauss Elimination Graph

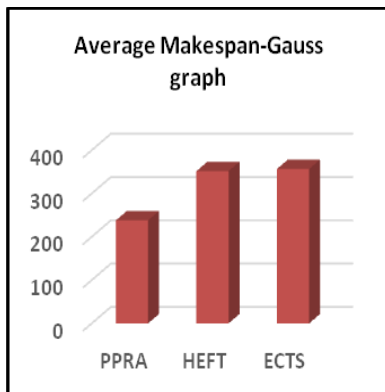


Figure 6:Comparison based on Average Makespan for Gauss graph

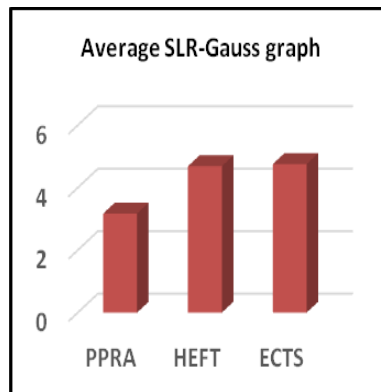


Figure 7 :Comparison based on Average SLR for Gauss graph

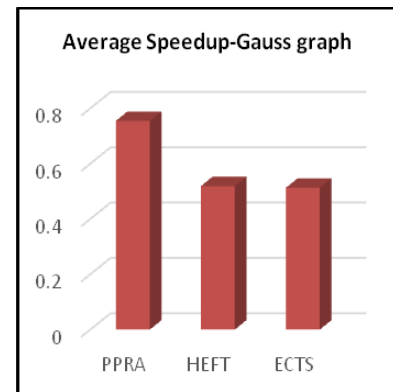


Figure 8 : Comparison based on Average Speedup for Gauss graph

Test Case 3: Cybershake[5] is an example for scientific workflow that is a data and memory intensive earthquake hazard characterization application used by the Southern California Earthquake Centre. The workflow begins by generating Strain Green Tensors (SGTs) for a region of interest via a simulation. These SGT data are then used to generate synthetic seismograms for each predicted rupture followed by the creation of acceleration and probabilistic hazard curves for the given region. A sample CyberShake workflow is depicted in Figure 9. The performance of proposed PPRA algorithm is compared with HEFT and ECTS for the Cybershake Sample Graph based on Average makespan, Average SLR and Average Speedup shown in Figures 10,11,12 respectively.

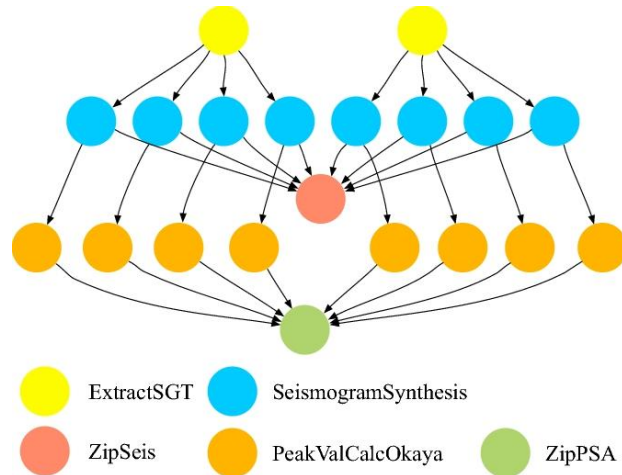


Figure 9: Sample Cybershake Graph

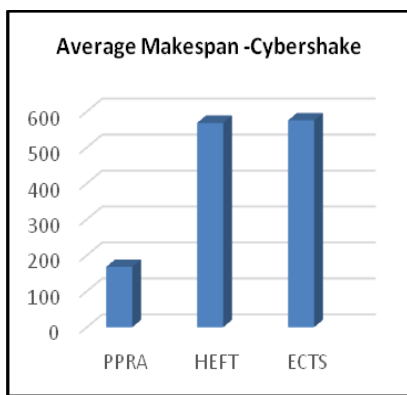


Figure 10 : Comparison of Average Makespan for Cybershake

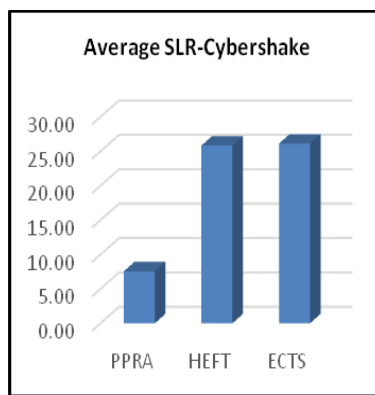


Figure 11 : Comparison of Average SLR for Cybershake

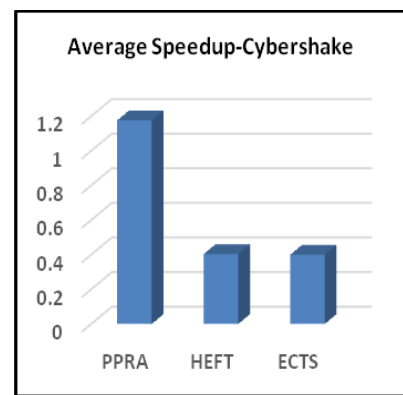


Figure 12 :Comparison of Average Speedup for Cybershake

Test Case 4: Montage workflow is an I/O intensive astronomy application used to create custom mosaics of the sky based on a set of input images. A composite image of a region of the sky that is too large to be taken by astronomical camera or images that have been measured with different wavelengths and instruments can be generated. The structure of sample Montage workflow is shown in Figure 13. The performance of proposed PPRA algorithm is compared with HEFT and ECTS for the Montage workflow based on Average Makespan, Average SLR and Average Speedup, is shown in Figures 14,15 and 16.

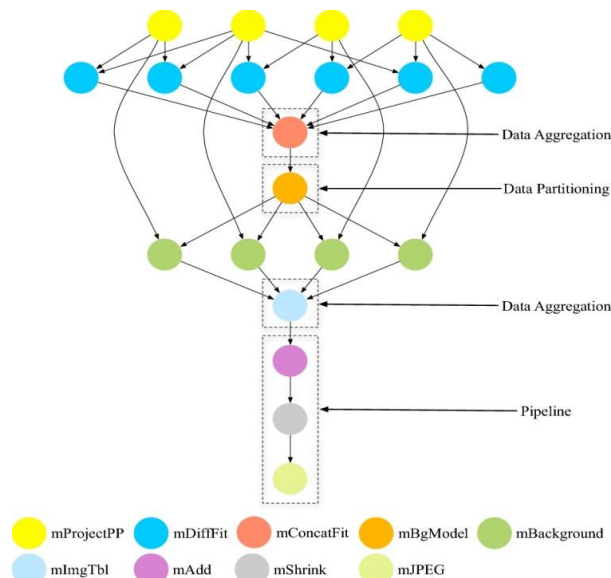


Figure 13 : Sample Montage Workflow

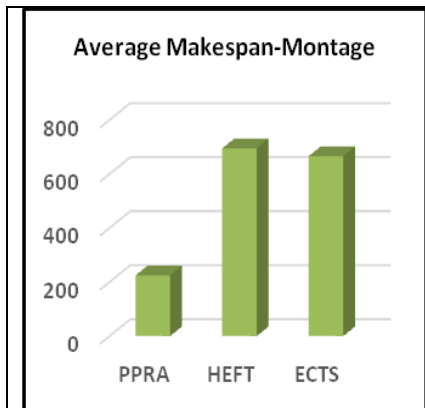


Figure 14 :Comparison based on Average Makespan for Montage graph

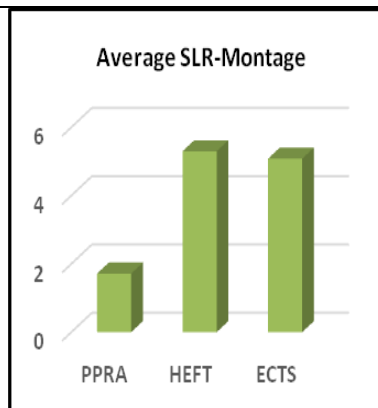


Figure 15 :Comparison based on Average SLR for Montage graph

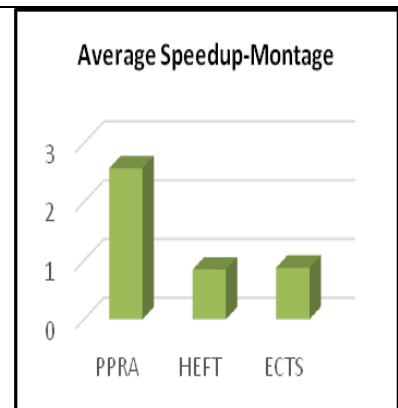


Figure 16 : Comparison based on Average Speedup for Montage graph

Test Case 5: The Epigenomics workflow of Bioinformatics field, is a CPU intensive application that automates the execution of various genome sequencing operations. This workflow is created by the USC Epigenome Center and the Pegasus Team. A sample Epigenomics workflow is shown in Figure 17. The performance of proposed PPRA algorithm is compared with HEFT and ECTS for the Epigenomics workflow sample based on Average Makespan, Average SLR and Average Speedup shown in Figures 18,19,20.

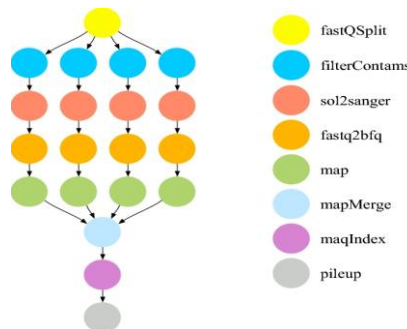


Figure 17 :Sample Epigenomics Workflow

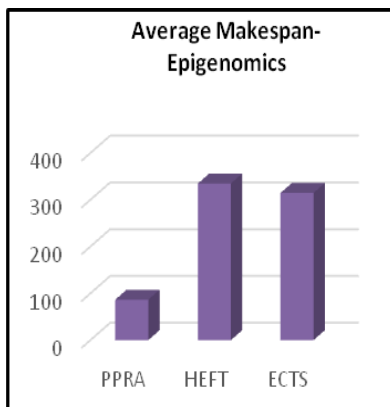


Figure 18 :Comparison based on Average Makespan for Epigenomics

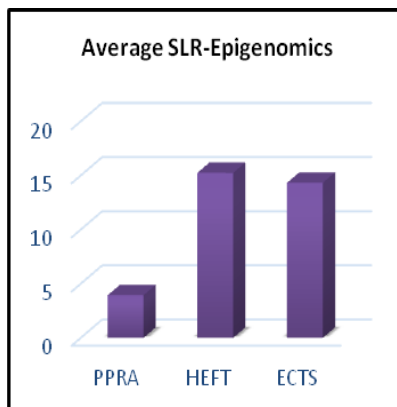


Figure 19 :Comparison based on Average SLR for Epigenomics

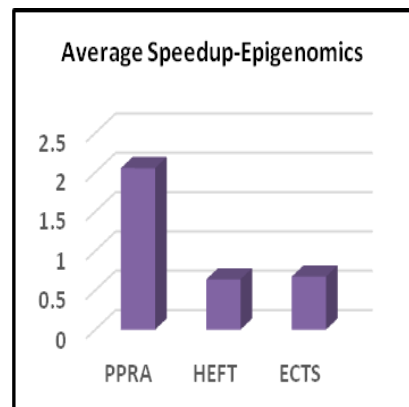


Figure 20 :Comparison based on Average Speedup for Epigenomics

VIII. Conclusions

Cloud resources exhibit variations in performance due to resource sharing, virtualization, and heterogeneity. Performance degradation of underlying hardware and load variations affect task execution times. Performance variation in network resources affect data transfer times. Since the resources are not dedicated and can be used by other users simultaneously, there are load variations on resources, resulting in fluctuations in resource availability that affects the schedules. In this paper, we propose static scheduling algorithm called Predictive Probabilistic Resource Availability based Cloud Workflow Scheduling (PPRA).The proposed algorithm performs better than HEFT and ECTS algorithms in a scenario where resource availability factor is

considered. We have presented four test cases, in the comparative experimental study of performance of PPRA. When algorithms such as PPRA, ECTS, HEFT were executed, in an environment considering resource availability factor, the performance metrics such as makespan, SLR and speedup were found to be better in PPRA than in others, for random graphs with varying sizes. This comparative study of performance of our proposed algorithm PPRA with HEFT and ECTS for real life application graphs Gauss Elimination graph, Cybershake, Montage and Epigenomics. For various sizes of graphs with 10,20,30,40,50 nodes, the performance improvement of PPRA when compared with HEFT, is ranging between 43% and 78%. When PPRA is compared with HEFT, on an average, the overall percentage reduction in makespan for graphs of various sizes is 66%. When PPRA is compared with ECTS algorithm, the percentage reduction in makespan ranges from 31% to 78% for various sizes of graphs. On an average, the overall percentage reduction in makespan for graphs of various sizes when PPRA is compared with ECTS is 64%. Also there is considerable percentage reduction in makespan for real life application graphs by about 67% for Montage workflow, 73% for Epigenomics, 70% for Cybershake and 33% for Gauss graphs. The simulation results show that PPRA algorithm performs better and is more reliable in environment where resource availability is continuously varying than other algorithms that assume full resource availability while scheduling tasks.

References

- [1]. Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems* 2009; 25(6):599–616.
- [2]. Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. F. De Rose and Rajkumar Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms", *SOFTWARE – PRACTICE AND EXPERIENCE*, *Softw. Pract. Exper.* 2011; 41:23–50, Published online 24 August 2010 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/spe.995
- [3]. *Cloud Computing Theory and Practice*, Dan C. Marinescu, Morgan Kaufmann, Elsevier, 2013
- [4]. Dan Sullivan, "A Definitive Guide to Cloud Computing", Real Time Publishers
- [5]. Maria A. Rodriguez, Rajkumar Buyya, "A Taxonomy and Survey on Scheduling Algorithms for Scientific Workflows in IaaS Cloud Computing Environments", *CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE*, *Concurrency Computat.: Pract. Exper.* 0000; 00:1–32, Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cpe.4041
- [6]. Weiwei Chen, Ewa Deelman, "WorkflowSim : A Toolkit for Simulating Scientific Workflows in Distributed Environments" The 8th IEEE International Conference on eScience 2012 (eScience 2012), Chicago, Oct 8-12, 2012
- [7]. Maria A. Rodriguez and Rajkumar Buyya, "Budget-Driven Resource Provisioning and Scheduling of Scientific Workflow in IaaS Clouds with Fine-Grained Billing Period", *ACM Trans. Autom. Adapt. Syst.* 9, 4, Article 39 (March 2015), 22 pages. DOI: 0000001.0000001
- [8]. Qazi Zia Ullah, Shahzad Hassan, Gul Muhammad Khan, "Adaptive Resource Utilization Prediction System for Infrastructure as a Service Cloud", *Computational Intelligence and Neuroscience*, Hindawi
- [9]. Haluk Topcuoglu, Salim Hariri and Min-You Wu, "Performance Effective and Low Complexity Task Scheduling for Heterogeneous Computing", *IEEE Transactions on Parallel and Distributed Systems*, Vol.13, No.3, March 2002, : 260-274
- [10]. R. Eswari and S. Nickolas, "A Level-wise Priority Based Task Scheduling for Heterogeneous Systems", *International Journal of Information and Education Technology*, Vol. 1, No. 5, December 2011
- [11]. M. A. Rodriguez and R. Buyya, "A Taxonomy and Survey on Scheduling Algorithms for Scientific Workflows in IaaS Cloud Computing Environments", *CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE*, *Concurrency Computat.: Pract. Exper.* 0000; 00:1–32, Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cpe.4041
- [12]. Sukhpal Singh, Indrveer Chana, and Rajkumar Buyya, "STAR: SLA-Aware Autonomic Management of Cloud Resources", *IEEE TRANSACTIONS ON CLOUD COMPUTING*, VOL. 4, NO. X, 2017
- [13]. Rongdong Hu, Jingfei Jiang, Guangming Liu, Lixin Wang, "Efficient Resources Provisioning Based on Load Forecasting in Cloud", *Hindawi Publishing Corporation, e Scientific World Journal*, Volume 2014, Article ID 321231, 12 pages <http://dx.doi.org/10.1155/2014/321231>
- [14]. Brendan Jennings and Rolf Stadler, "Resource Management in Clouds: Survey and Research Challenges", *Journal of Network and System Management*, Volume 23 Issue 3, July 2015, Pages 567-619
- [15]. Sayanta Mallick, Gaétan Hains, "A Resource Prediction Model for Virtualization Servers", *Laboratoire d'Algorithmique, Complexité et Logique (LACL), Université de Paris-Est Créteil (UPEC), Faculté des Science et Technologie, France, December 2011*
- [16]. Yinfeng Wang, Zhijing Liu, Wei Yan, "Algorithms for Random Adjacency Matrixes Generation Used for Scheduling Algorithms Test", *International Conference on Machine Vision and Human-Machine Interface (MVHI)*, 2010

IOSR Journal of Computer Engineering (IOSR-JCE) is UGC approved Journal with SI. No. 5019, Journal no. 49102.

Chitra S. "Predictive Probabilistic Resource Availability based Cloud Workflow Scheduling (PPRA)." *IOSR Journal of Computer Engineering (IOSR-JCE)* 19.4 (2017): 54-63.