RESEARCH PAPER | PAPER ID: 2015/MY/IJCITE/V2-E5-06 | PAGE NO: 42 - 48

ISSN (Online): 2348-8557
International Journal of Computer Informatics & Technological Engineering
Volume (2) Issue (5) May 2015

DEADLINE CONSTRAINED SCIENTIFIC WORKFLOW SCHEDULING WITH RANKING AND REPLICATION OF TASKS

Nallakumar. R [1], Dr. N. Sengottaiyan [2], Sruthi Priya K.S [3]
[1]Teaching Fellow, [3] PG Scholar
Department of CSE, Anna University Regional Centre, Coimbatore, India 1
[2] Professor, Depatment of CSE,
Indira Institute of Engineering and Technology, Thiruvallur, Chennai, India

Abstract: ***The emergence of Cloud computing in distributed systems for service provisioning stimulates the analysts and researchers in executing exploratory scientific applications such as workflows. The most demanding issues in the Clouds are workflow scheduling, which expects to fulfil the Quality of Service of the user like a deadline along with the cost reduction of workflow execution. An already prevailing work on workflow scheduling algorithm is in view of Partial Critical Paths which endeavour's to reduce the workflow execution cost and attempts to execute within the user-defined deadline and it is done only in a single cloud where ample resources won't be present to fulfil the user demand. In that case cloud providers will continue to process the user demands in order to increase their revenue. To address this problem, a novel replication aware dynamic workflow scheduling is introduced to the thought of ranking of tasks along with task replication for multi cloud and it dynamically allocates the workflows across multiple cloud domains by satisfying the Quality of Service requirement of the users.***

Index-Terms: ***Workflows, Partial Critical Paths, Task replication, Deadline.***

I. INTRODUCTION

Cloud Computing is one of the maturing areas that has picked up prominence in the late years among each one of those clients who make utilization of the Internet. Furthermore it is a blend of Grid and Utility Computing. High unwavering quality, extendibility, adaptability and Virtualization are the qualities of this upcoming innovation. It makes utilization of a pay-per-use utilization model to impart and convey immense no of computational assets to the clients with the acquirement of virtualization technologies. Cloud Computing serves the needs of the clients in different business and experimental applications. Utilization of these additionally diminishes the expense of equipment and programming as the assets are being rented just for a specific time of time[1] as requested by the end client. Cloud Computing is developing with an incredible prominence as of late because of its special trademark qualities such as scalability, reliability, security, cost-effective system. Numerous qualities of Cloud are really a mixture of Utility and Grid Computing. Cloud administrations propound on the premise of "Pay for what you use" via open(public) or private Cloud. Software As a Service (SaaS), Platform As a Service (PaaS), Infrastructure As a Service (IaaS) are the three service models of Cloud Computing technology.

Cloud serves in a huge number of clients crosswise over diverse stages. Any client who is recognized has the full rights to get to their records and applications from any Personal Computer arranged in wherever they have entry to steady association with the Internet. The innovation or Infrastructure behind Cloud Computing is not demonstrated expressly to the clients. Private, Public, Hybrid and Community cloud are the different deployment models within the Cloud. With the idea of Virtualization in Cloud, adaptability is profoundly achievable. Security and Scheduling are the two sub areas in the cloud and are the significant exploration issues as of late.

There are numerous uses of Cloud among which "Google Cloud" is the one. The different qualities of Cloud Computing from Google's perspective are user-driven, Powerful, Accessible, Intelligent, Programmable.

A various scope of exploration issues is tended to be in Cloud specifically are security, fault tolerant, workflow Scheduling. Among these deadline constrained Workflow scheduling is the most extreme issue in the period of computing.. Workflow Scheduling, a typical NP-complete issue is the

ISSN (Online): 2348-8557
International Journal of Computer Informatics & Technological Engineering
Volume (2) Issue (5), May 2015
Page No: 42-48

procedure of mapping the tasks to the proper resources with the assistance of proficient algorithm in a manner to fulfill the Quality of Service (QoS) of the client alongside the optimization of the cost with deadline constraint. A workflow represents a set of tasks and their interdependencies in the form of a directed acyclic graph (DAG) G= (V, E) where V is the no of tasks in the workflow and E is the data dependencies between those tasks. Critical Path heuristics are broadly utilized as a part of Workflow Scheduling. The critical path of a workflow is the longest execution path between the entry and exit tasks of the workflow. The vast majority of these heuristics attempt to schedule critical tasks (nodes), i.e. the tasks belonging to the critical path, first, by assigning them to the resources that process them earliest, in order to minimize the execution time of the whole workflow.

The proposed algorithm is in light of a comparable heuristic, in which the critical nodes are first scheduled, yet the execution time is not minimized whereas the price of executing the critical path before the user-specified deadline is reduced. Ranking the workflow tasks is employed to achieve this. In the wake of scheduling all the critical nodes, each of them has a start time which is a deadline for its parent nodes, i.e. its (immediate) predecessors in the workflow. So, the same methodology can be carried out by taking into account each critical node in turn as an exit node with its start time as a deadline, and generating a partial critical path that ends in the critical node and that leads back to an already scheduled node. In the SaaS Cloud-Partial Critical Paths (SC-PCP) algorithm, this methodology proceeds recursively until all tasks are scheduled successfully.

II. RELATED WORK

Abrishami S. and Naghibzadeh M.[2]previously developed a two-phase scheduling algorithm for utility Grids named as Partial Critical Paths (PCP), which attempts to minimize the cost of workflow execution along with deadline constraint.At present they propose two workflow scheduling algorithms: a one-phase algorithm which is called IaaS Cloud Partial Critical Paths (IC-PCP), and a two-phase algorithm named as IaaS Cloud Partial Critical Paths with Deadline Distribution (IC-PCPD2). Ostermann S., Prodan R.. and Fahringer T.[3]used clouds to extend a Grid workflow middleware and showed on a real implementation that this can speed up executions of scientific workflows. The goal is to provide an infrastructure that allows the execution of workflows on conventional Grid resources which can be supplemented on-demand with additional Cloud resources, if necessary. Concentration is held on the extensions that brought the presentation to the resource management service to consider Cloud resources, comprising new Cloud management, software (image) deployment, and security components. Deelman E., et al.[4]proposed a Pegasus framework to map complex scientific workflows onto distributed resources. Pegasus enables users to represent the workflows at an abstract level without worrying about the particulars of the target execution systems. Pegasus provides an option to cluster jobs together in cases where a number of small granularity jobs are present. Deadline constrained workflow scheduling algorithms on the grids are proposed by Plankensteiner and Prodan [5].W. Cirne, F. Brasileiro,D. Paranhos, L.F.W. Go´es, and W. Voorsluys[6]proposed an efficient algorithm for the distributed systems such as the grid systems with the task replication feature.Graph based task replication method is proposed [7] by G. Kandaswamy, A. Mandal, and D.A. Reed. A comparison on the Job replication and load balancing in a grid environment is presented by[8] M. Dobber, R. van der Mei, and G. Koole. Deadline constrained workflow scheduling with the Partitioned Balanced Time Scheduling (PBTS) algorithm was proposed by Byun et al in [9]. Hamid Mohammadi Fard, Radu Prodan, and Thomas Fahringer[10]proposed an algorithm for Multicloud environment. Dynamic workflow scheduling was done by calculating the rank of tasks based on the load of the task.

III. EXISTING METHODOLOGY

In the Existing work, Enhanced IC-PCP with Replication (EIPR) algorithm is proposed to schedule the workflows in the cloud environment with the consideration of deadline constraint. The existing algorithm is in light of a comparable heuristic, in which the critical nodes are first scheduled, yet the execution time is not minimized whereas the price of executing the critical path before the user-specified deadline is reduced. In the wake of scheduling all the critical nodes, each of them has a start time which is a deadline for its parent nodes, i.e. its (immediate) predecessors in the workflow. So, the same methodology can be carried out by taking into account each critical node in turn as an exit node with its start time as a deadline, and generating a partial critical path that ends in the critical node and that leads back to an already scheduled node.
In the IaaS Cloud-Partial Critical Paths (IC-PCP) algorithm, this methodology proceeds recursively until all tasks are scheduled successfully.. This algorithm works in three phases[11]. Those are as follows.

- Combined provision of cloud resources and tasks
- Data transfer aware provisioning adjust
- Task replication

ISSN (Online): 2348-8557
International Journal of Computer Informatics & Technological Engineering
Volume (2) Issue (5), May 2015
Page No: 42-48

The problem associated with the existing methodology is that the multi cloud environment is not considered which will lead to a selfish behavior and User QoS(Quality Of Service) demands are not satisfied in the expected manner due to the consideration of single cloud environment.

IV.OUR CONTRIBUTIONS

In this work a novel replication aware dynamic workflow scheduling is proposed for multi cloud environment. as like existing work, this is also attempts to reduce a cost and execution time to meet user specified qos constraint (cost and deadline). this work will calculate the rank for each and every task in the workflow based on its load and its dependency among each other.

The rank is calculated by using the below formulae:

Rank(i)=

$$\begin{cases} load(a) + \max_{b \in succ(a)} \{comm(a,b) + rank(b)\}, & \text{if } a \neq exit \\ load(a), & \text{if } a = exit \end{cases}$$

A novel replication aware workflow scheduling across multi cloud:

In this algorithm, the scheduling of workflow across multiple cloud service providers is discussed. The tasks are scheduled in the appropriate VM's based on the ranking of tasks. The tasks are ranked by considering the workload of the tasks. For each unscheduled tasks rank will be computed and after ranking, the task will start to auction. Based on the auction value, the resource provider will bid his requirement. And efficient resource will be selected for scheduling the tasks on the VM's available in the resource.

Method: A novel replication aware workflow scheduling across multi cloud.

Input: The workflow application

Output: Scheduled workflow on Multi cloud.

Algorithm 1: A Novel Replication Aware Workflow Scheduling

Step 1: tasks ← [n]

Step 2: ranks ← φ

Step 3: tempTask ← exit

Step 4: while tasks ≠ φ do

a. Add (ranks, ComputeRank (tempTask)) ;
b. tempTask ← Tail(tasks);
c. Remove (tasks, tempTask);

Step 5: end

Step 6: tasks ← [n]

Step 7: sort (tasks, ranks)

Step 8: while tasks ≠ φ do

a. auctioneer ← Head(tasks)
b. w ← winnerCloud
c. submit (auctioneer, w)
d. Remove (tasks, auctioneer);

Step 9: End

In the following algorithm the ranking mechanism for tasks is presented. This algorithm shows the ranking process of the tasks based on its load. The highest rank is allotted to the task which is having high load. That is the longest path from it to the exit node.

Compute_Rank (task)

Input : Tasks

Output: Rank of task

Algorithm 2: Computation of Rank of the tasks

Step 1: If task = exit then

a.Return load (task)

Step 2:Else

a. Rank ← null
b .Succ ← succ (task)
c. While succ != null do
d. B← head (succ)
e. Add (ranks, Comm (task, b) + computeRank (b))
f. Remove (succ, b)
g. End
h. Return max(ranks)+ load (task)

Step 3:.End.

The system architecture diagram of the proposed work is as follows.

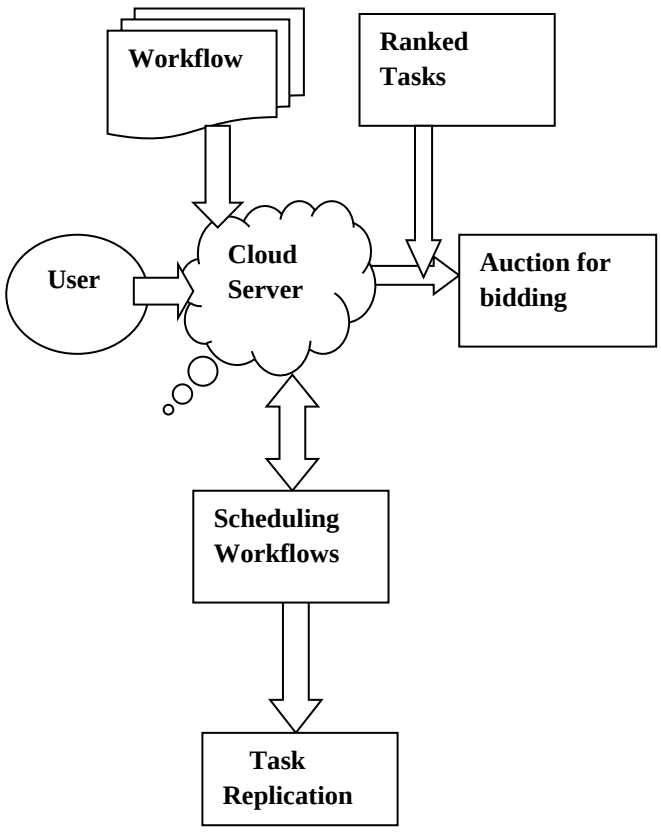

Figure 1. System Architecture Diagram

ISSN (Online): 2348-8557
International Journal of Computer Informatics & Technological Engineering
Volume (2) Issue (5), May 2015
Page No: 42-48

V.OPERATIONAL DESCRIPTION

The novel replication aware workflow scheduling scheme consists of seven modules as follows.

- **Creating cloud network**

A simulation toolkit empowers modeling and simulation of Cloud computing systems and application provisioning environments. The CloudSim toolkit assists in both system and behavior modeling of Cloud system components such as data centers, virtual machines (VMs) and resource provisioning policies. It implements generic application provisioning techniques that can be stretched with ease and limited effort. At present, it aids modeling and simulation of Cloud computing environments composed of both single and inter-networked clouds (federation of clouds). Additionaly, it exposes custom interfaces for executing policies and provisioning techniques for allocation of VMs under inter-networked Cloud computing scenarios.The cloud users, datacenters and cloud virtual machines as per our requirement are created. Job manager, Cloud controller are also created. The Job Manager receives the client's jobs, is responsible for scheduling them, and coordinates their execution. It is capable of communicating with the interface the cloud operator provides to control the instantiation of VMs. This interface is called the Cloud Controller. By means of the Cloud Controller the Job Manager can allocate or deallocate VMs according to the current job execution phase. Common Cloud computing terminologies are referred. The term instance type will be used to differentiate between VMs with different hardware characteristics.

- **Identifying dependency among tasks**

Among the set of submitted tasks, the dependency will be calculated i.e. the flow of execution of tasks. The dependency among the tasks is computed as a follow relationship. The DAG(Directed Acyclic Graph) will be constructed which will define the flow of execution of tasks. The nodes in the DAG represent the set of tasks and the edge represents the flow of execution among tasks, where the output of one task will be the input of another task.

A. Rank computation of tasks

After finding out the dependencies among the tasks, the rank will be computed for each and every task that is unscheduled. The rank is calculated based on the load of the tasks in the resource. This rank defines a longest path from the task to the exit task. The rank is calculated as follows:

$$\text{Rank}(i)=\begin{cases}\text{load}(i)+\max\limits_{j\in succ(i)}\{\text{comm}(i,j)+\text{rank}(j)\}, & \text{if } i\neq \text{exit}\\ \text{load}(i), & \text{if } i=\text{exit}\end{cases}$$

where load (i) represents the workload of task i and comm (i, j) is the output of tasks i to j. Function succ (i) returns the immediate successors of task i. The reason for using the makespan values for ordering the tasks is twofold: it is the structure-dependent objective that preserves the precedence relationships in the ordered list and it usually has a direct impact on the cost since the longer tasks need more resource usage. The workflow tasks are ordered according to their bottom level (B-level), defined in graph theory as the longest path from each task to the exit task, including the task itself.

- **Finding winner resource by auction**

The set of tasks based on its rank will start its auction. It will submit its input and output requirement to the scheduler. After the task submits its workload details, the resource will bid its capacity of processing task. That bid will consists of the time and cost that will be consume by the resource to process that task. The auctioneer will select the winner resource with the high computation capability. Each auction is based on some rules which define the identity of the winner. The most famous result in this area is the VCG which applies to utilitarian (sum of the valuations of the agents) objectives only.In this approach each task is as an auctioneer which starts an auction to select a proper resource for its execution. The task announces to the resources its workload, the dependencies with other tasks, and the required input and output. Each resource j bids by the strategy si,j = (ti,j, ci,j) which is a combination of its proposed time ti,j and proposed cost ci,j. The strategy si,j means that the resource j claims to complete the task i until the time ti,j with the cost of ci,j. Each resource is allowed to bid by more than one strategy in every auction.

B. Data transfer aware provisioning adjust

After selection of resources, the tasks will be allocated to the VM's based on the start time and end time of tasks. Based on the processing time of tasks, the VM's start time and end time will be decided. The VM's start time and end time will be mentioned based on the flow of tasks. That is based input and output relations of each task, the task allocation will be adjusted.If the first scheduled task in a VM is not an entry task, data from parent tasks have to be moved to the virtual machine before the task can run, and thus, the VM needs to be provisioned before the start time of its first task. This affects the start time of tasks and the total provisioning time of the VM, and may cause workflow execution delay and execution of VMs for an extra billing period.For each non-entry task scheduled as first task of a virtual machine, and for each non-exit task scheduled as the last task of a virtual machine, the algorithm meets the

ISSN (Online): 2348-8557
International Journal of Computer Informatics & Technological Engineering
Volume (2) Issue (5), May 2015
Page No: 42-48

required communication time by setting the start time of the machine D (i, j) earlier than st of the first task and or setting the end time of the machine D (i, j) later than the finish time of the last task, depending on where the extra time is required. Finally, the beginning of the first allocation slot of each virtual machine is anticipated by the estimated deployment and boot time for virtual machines, which was accounted for during the scheduling process.

C. Scheduling workflows

Based on the task and VM utilization time, the task will be scheduled to the appropriate VM. The scheduled task will compute the output before the start time of the tasks. This module is implemented through IC-PCP Scheduling Algorithm. Algorithm shows the pseudo-code of the overall IC-PCP algorithm for scheduling a workflow. After some initialization, the algorithm generates the fastest schedule for the input workflow in Step 3. This is a preliminary schedule, which obviously has the highest cost. In the next phases, the algorithm tries to refine this preliminary schedule by changing the selected service of each task, such that the overall execution time extends to the user's deadline, and the cost decreases as much as possible. A scheduled task is defined as a task whose selected service is finalized and which never changes in the next phases of the algorithm. Obviously, all tasks are still unscheduled in the preliminary schedule. Then, the algorithm computes the ESTs and the LFTs of all tasks according to the preliminary schedule. In Step 6, the algorithm checks if a solution exists, according to inference 2. After that, the dummy nodes, t_{entry} and t_{exit}, are marked as scheduled and finally the SchedulеParents procedure is called for t_{exit}, in Step 8. This procedure schedules all unscheduled parents of its input node. As it has been called for t_{exit} , it will schedule all workflow tasks.

G.Task Replication

If any of the resources completes its execution and remains idle, the running task will be replicated to it in order to reduce the overall completion time. The space replication approach is followed in our work. Space replication is defined as replicating the task across multiple virtual machines. The task replication process works as a semi-active replication technique for fault tolerance, with the difference that here tasks are replicated across performance- independent hosts rather than failure-independent locations. As in the case of replication for fault tolerance, in our approach replication is also managed by a single entity. The process operates as follows. Initially, the algorithm generates new VMs based on the available replication budget. For selection of the type of VM to be used for the new VM, the list of already provisioned VMs is sorted in descending order of number of scheduled tasks. Starting from the beginning of the sorted list, the first VM type whose cost is smaller than the available budget is chosen. The chosen VM is replicated, provisioning information is updated, the VM is moved to the end of the list, the available budget is updated, and the search is restarted. The process is repeated while the budget admits new instantiations. The new provisioned VM and the correspondent time slot reproduce the same start and end times than the original VM. Next, a list of all possible idle slots is created. This list includes both paid slots, which are slots were the VM is actually provisioned and no task is scheduled on it; and unpaid slots, which are the times between the start time and the deadline were the VM is not provisioned.

VI. RESULTS AND DISCUSSION

The performance evaluation is done to prove the efficiency of the proposed methodology by comparing it with the existing methodology in terms of metrics called the execution time, and the utilization. The graphical representation of the comparison of the proposed methodology is given in the following section.

Average execution time

The execution time or CPU time of a given task is defined as the time spent by the system executing that task, including the time spent executing run-time or system services on its behalf.

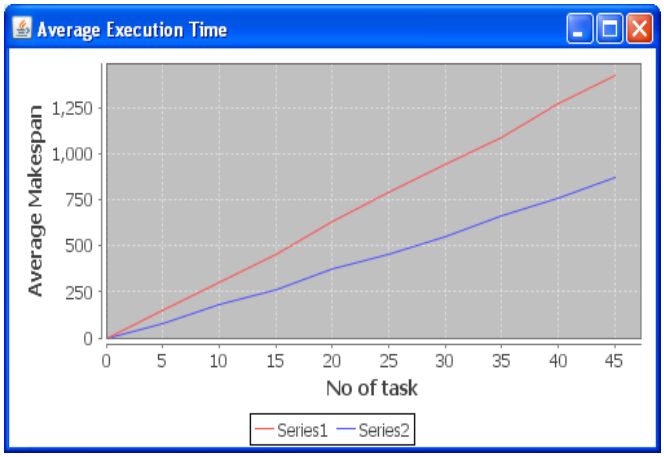

Figure 2. Average makespan

In the above graph, execution time comparison of proposed methodology with the existing approach is done against the various numbers of tasks. In the above graph, x axis denotes the number of tasks considered and the y axis plots the average execution time. From the graph, it has been proved that the

ISSN (Online): 2348-8557
International Journal of Computer Informatics & Technological Engineering
Volume (2) Issue (5), May 2015
Page No: 42-48

proposed methodology is improved in terms of less computation cost whereas in the existing methodology it is high.

Cpu Utilization

Bandwidth utilization is referred to as the percentage of bandwidth that are utilized by the tasks which are submitted by the users. It should be maximized in order to increase the performance and reputation of cloud service providers. The graph is plotted in following graph.

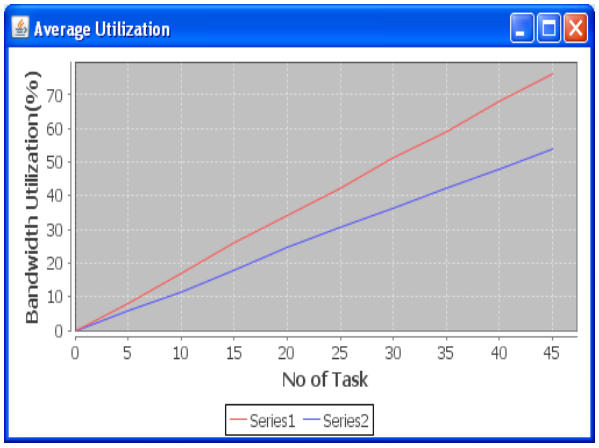

Figure 3.Band Width Utilization

In the above graph, bandwidth utilization comparison of proposed methodology with the existing approach is done against the various numbers of tasks. In the above graph, x axis denotes the number of tasks considered and the y axis plots the bandwidth utilization. From the graph, it has been proved that the proposed methodology is improved in terms of less computation cost whereas in the existing methodology it is high.

VII. CONCLUSION AND FUTURE WORK

Scientific workflows present a set of characteristics that make them suitable for execution in Cloud infrastructures which offer on-demand scalability that allows resources to be increased and decreased to adapt to the demand of applications. The scientific workflow scheduling is done across the multi clouds with the satisfaction of user QoS demands particularly like deadline and cost of execution of tasks. The workflow scheduling is done efficiently with the better coverage among multi clouds. The deadlock avoidance can be considered in the future research and can be extended to the distributed workflow schedulers which can coordinate each other to map the instances and jobs into multiple storage constrained sites while avoiding deadlock. It can also make applicable in reality if the number of workflows as well as the constituent jobs are large.

REFERENCES

[1]Ranjith Singh, Sarbjeet Singh(2013), "Score Based Deadline Constrained Workflow Scheduling Algorithm for Cloud systems", IJCCSA, (Vol 3).

[2] Abrishami S. and Naghibzadeh M. (2013)," Deadline-constrained workflow scheduling algorithms for Infrastructure as a service Cloud", Future Generation Computer Systems 29, 158–169.

[3]OstermannS.,ProdanR..andFahringer T.(2009),''Extending grids with cloud resource management for scientific computing'', 10th IEEE/ACMInternational Conference on Grid Computing, Banff, Alberta, Canada, pp. 42–49.

[4] Deelman E., et al. (2005), ''Pegasus: a framework for mapping complex scientific workflows onto distributed systems'', Sci. Program., 13, pp. 219–237.

[5] K. Plankensteiner and R. Prodan(2012) ''Meeting Soft Deadlines in Scientific Workflows Using Resubmission Impact,'' IEEE Trans.Parallel Distrib. Syst., vol. 23, no. 5, pp. 890-901.

[6] W. Cirne, F. Brasileiro,D. Paranhos(2007), L.F.W. Go´es, and W. Voorsluys,''On the Efficacy, Efficiency and Emergent Behavior of TaskReplication in Large Distributed Systems,'' Parallel Comput., vol. 33,no. 3, pp. 213-234.

[7] R. Sirvent, R.M. Badia, and J. Labarta(2009), ''Graph-Based TaskReplication for Workflow Applications,'' in Proc. 11th Int'l Conf.HPCC, pp. 20-28.

[8] M. Dobber, R. van der Mei, and G. Koole, '(2009)'Dynamic Load Balancing and Job Replication in a Global-Scale Grid Environment:A Comparison,'' IEEE Trans. Parallel Distrib. Syst., vol. 20,no. 2, pp. 207-218.

[9] E.-K. Byun, Y.-S. Kee, J.-S. Kim, and S. Maeng(2011), ''Cost Optimized Provisioning of Elastic Resources for Application Workflows,''Future Gener. Comput. Syst., vol. 27, no. 8, pp. 1011-1026.

[10]Hamid Mohammadi Fard, Radu Prodan, and Thomas Fahringer(2013), "A Truthful Dynamic Workflow Scheduling Mechanism for Commercial Multicloud Environments",IEEE transactions on parallel and distributed systems, vol. 24. [11]Rodrigo N.Calheiros.,Rajkumar Buyya(2014),''Meeting Deadlines of Scientific Workflows in Public Clouds with Tsks Replication'', IEEE Transactions On Parallel and Distributed Systems,Vol.25,No 7,July 2014.

[12] Bharathi S., Chervenak A., Deelman E., Mehta G., Su M.H. and Vahi K. (2008),''Characterization of scientific wworkflows'', The 3rd Workshop on Workflows in Support of Large Scale Science, Austin, TX, USA, pp. 1–10 .

ISSN (Online): 2348-8557
International Journal of Computer Informatics & Technological Engineering
Volume (2) Issue (5), May 2015
Page No: 42-48

[13] Buyya R., Yeo C.S., Venugopal S., Broberg J. and Brandic I. (2009), ‘‘Cloud computing and emerging IT platforms: vision, hype and reality for delivering computing as the 5th utility’’, Future Gener. Comput. Syst., 25, pp. 599–616

[14] Juve G., Deelman E., Vahi K., Mehta G., Berriman B., Berman B.P. and Maechling, (2009),‘‘Scientific workflow applications on Amazon EC2’’, 5th IEEE International Conference on e-Science, Oxford, UK.

[15] Kellerer H., Pferschy U. and Pisinger D. (2004), Knapsack Problems, Springer Verlag.

[16] Lang S.D(1999), “An extended banker’s algorithm for deadlock avoidance,” IEEE Transactions on Software Engineering, vol. 25, no. 3, pp.428–432, May/June .

[17] Yang Wang and Paul Lu(2012),” Maximizing Active Storage Resources with Deadlock Avoidance in Workflow-Based Computations”, IEEE Transactions on Software Engineering, vol. 25, May .

[18] Yu J., Buyya R.. and Tham C.K(2005), ‘‘Cost-based scheduling of scientific workflow applications on utility grids’’, First Int’l Conference on e-Science and Grid Computing, Melbourne, Australia, pp. 140–147 .

[19] Priya R.Lodha, Mr. Avinash P. Wadhe (2013), “Study of different types of workflow scheduling algorithm in cloud computing,International journal of Advanced Research in Computer Science and Electronics Engineering,Vol 2.

[20] Rodrigo N. Calheiros , Rajiv ranjan , Anton Beloglazov , Césara.A.F.DeRose,Rajkumarbuyya(2010),CiteSeerX “cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,II software: practice and experience.

[21]Abrishami S. and Naghibzadeh M. (2012),” Deadline-constrained workflow scheduling in software as a service Cloud”, Scientia Iranica, Transactions D: Computer Science and Engineering and Electrical Engineering doi:10.10.16/j.scient.2011.11.047.