Implementing Enhanced ICPCP Algorithm with Task Replication in Public Cloud

Chaudhari Arati D.
PG Student
SSVPS's BS Deore College of Engineering,
Dhule, 424005,
India

Mandre B. R.
Associate Professor,
SSVPS's BS Deore College of Engineering,
Dhule, 424005,
India

ABSTRACT
Cloud Computing has Large Scale Distributed Infrastructure which is accessible and scalable infrastructure. Cloud computing provides a pay as you go model in which the user has to pay for the services he uses. One of the characteristic of cloud is elasticity in which resources can be dynamically increases or decreases as per user requirement. The goal of this project is to execute the scientific workflows in public cloud within user define deadline and smallest possible cost. The deadline of the project can be meeting by provisioning more virtual machines that required. The algorithm Enhanced ICPCP uses the concept partial critical path which is defined in the ICPCP. The simulation result shows the algorithm reduces the execution time of different scientific workflows simulated using the cloudsim.

General Terms
Cloud Computing, Scheduling, Scientific Workflow.

Keywords
Cloud Computing, Scientific Workflows, Task Replication, Soft Deadline, Workflow Scheduling.

1. INTRODUCTION
To provide flexible and on demand services for number of companies, cloud computing becomes more attractive. Cloud computing can be defined as a type of parallel and distributed system which consist of collection of inter connected and virtualized computers that are dynamically provisioned and represented as one or more computing resources based on service level agreements which are established between service provider and customers [2].
Workflow application can be useful in many areas such as astronomy, bioinformatics, and physics, for the development of Scientific Application. These workflows contain hundred or thousand number of tasks which can be represented by Directed Acyclic Graph, in which node represents the Task and edge represents the relationship between tasks. Cloud computing has public model in which resources can be hire by paying charge for it, called as pay-per-use system. In this model resources are dynamically scalable according to the need of application. So, the resources for executing the workflow can be provisioned on demand and also its number can be increased until it contains enough budgets to support it [1]. Cloud computing provides Infrastructure as a Service model, in which user obtains the virtual machines as resource and deploy their workflow applications on it. Cloud makes this a suitable platform to execute deadline-constrained scientific workflows.

One of the constraints for workflow execution is the budget allotted for it, which becomes limitation for hiring the number of resources. This is because the cloud providers apply charges for resource utilization for time interval. Our work aims at scheduling the workflow, such that the execution is completed before the deadline and within the budget constraint [1].

2. RELATED WORK
In 2008, Yun Yang et al focused on scheduling for cost constrained transaction intensive cloud workflows by considering the characteristics of cloud computing. These types of workflows have large number of workflow instances which are bounded by budget for execution. The purpose of this algorithm is to minimize the cost under certain user-designated deadlines. The algorithm has characteristics as, they consider the pay per use feature of cloud workflow, enables the compromising of execution cost and time [9].

Lin and Lu [5] proposed SCPOR, a scientific workflow scheduling algorithm that is able to schedule workflows in need of elastically changing compute resources. But it misses the capability of considering the cost for utilization of resources.

To schedule the scientific workflows in Software as a Service cloud S. Abrishami et al proposed a new algorithm based on the partial critical path in 2012. This algorithm tries to minimize the workflow execution cost by meeting its deadline. This algorithm first schedules the critical path of the workflow and then finds the partial critical path to each task on critical path [3].

In 2013, Salid Abrishami, Mahmoud Naghibzadeh and Dick H. J. E pema proposed Deadline-Constrained Workflow Scheduling Algorithm for Infrastructure Service. In this paper, execution time is minimized by maintaining the user defined deadline. The author implements two different algorithms based on PCP. First is Cloud Partial Critical Path and second with Deadline Distribution [4]. The disadvantage of this algorithm is that they didn't consider the data transfer time during provisioning and scheduling.

The author Rajkumar buyya et al proposed a combine resource provisioning and scheduling for scientific workflow execution. They used Meta heuristic optimization algorithm, Particle Swarm Optimization to minimize the execution cost. This algorithm performs better for smaller sized workflow[6]. In order to reduce the limitations of previous algorithms, proposed algorithm takes replication of tasks in idle time of virtual machine such that the tasks are complete its execution before deadline.

3. SYSTEM MODEL
Scientific workflow application is represented by Directed Acyclic Graph (DAG) G=(t, e) where t is number of tasks in the workflow and e is the number of edges which represents relationship between task. Relationships are in the form of

edges $E_{i,j} = (t_i, t_j)$ where $(t_i, t_j) \in$ t which represents dependency of t_j on t_i. $Parent(t_j)$ represents the list of parent tasks of t_j and $Children(t_j)$ represents the list of child tasks of t_j . DL(G) is the soft deadline associated with each workflow. Resources provided by the cloud provider are virtual machines, which are denoted as VM = { $vm_1, vm_2, \ldots\ldots, vm_n$}. The cost associated with virtual machine is denoted as C = $\{c_1, c_2 \ldots\ldots, c_n\}$. Runtime matrix RM contains the runtime of each task. An element $RM_{j,k}$ of RM is the runtime of task j on resource k. RM_{min} is the minimum execution time of task in the matrix. Each edge $E_{i,j}$ of G has some value, is the Data Transfer Time (DTT) required to transfer data from parent task to child task.

Each task in the workflow has two important parameters known as Early Start Time(EST) and Latest Finish Time (LFT). Early Latest Time is the early time at which all of its parents finish their execution whereas Latest Finish Time is the late time at which all of its children get executed. EST and LFT of any task of workflow can be calculated by following formulae.

$$EST(t_j) = \begin{cases} 0, & if\ t_j = t_{entry} \\ \max\limits_{t_p \in Parent(t_j)} \left(EST(t_p) + RM_{min}(t_p) + DTT(e_{p,j})\right), & Otherwise \end{cases}$$

$$LFT(t_j) = \begin{cases} DL(G), & if\ t_j = t_{exit} \\ \max\limits_{t_c \in Children(t_j)} \left(LFT(t_c) - RM_{min}(t_c) - DTT(e_{c,s})\right), & Otherwise \end{cases}$$

The time at which task assigned for execution is denoted as $ST(t_j)$ that is Schedule Time t_j.

3.1 System Architecture

Figure 1 shows the system architecture of the proposed system. User has to provide the XML file representing Directed Acyclic Graph (DAG) structure of workflow. DAX files contains list of tasks, dependencies between tasks, their computation time and size of the input and output files generated by the tasks. The DAX file contains information about task as task id, its runtime and name of the task. The tasks are extracted from the XML file and partial critical path is obtained. It is provided as an input to the first step of EIPR algorithm. Optimized schedule is then found and replication precedence of task is optimized. Lastly based on the optimized schedule, tasks are submitted to the VMs in cloud.

3.2 Enhanced ICPCP with task replication Algorithm

Main objective of enhanced ICPCP with task replication algorithm is to increase the chances of completing workflow execution within user define deadline in public cloud.

The algorithm has following three main steps:-

1) Combined provisioning & scheduling.
2) Data transfer aware provisioning adjusts.
3) Task Replication.

3.3 Finding Partial Critical Path for the task

The Partial Critical Path algorithm minimizes the cost of deadline constraint workflow by assigning all tasks of PCP to the virtual machine.

PCP is determined by identifying the unassigned parents. Unassigned parent is a node that is not scheduled or assigned to any PCP. Further, PCP is created by finding the unassigned critical parent of the node, starting at the exit node, and repeating the same until there are no further unassigned parents for the critical parent. Algorithm 1 details the procedure to find the PCP of a node.

Algorithm 1: To find PCP for Task
Input: US: Unscheduled Tasks of workflow
Set PCP =NULL
while (! $isEmpty(US)$)
{
for $\left(t_p \in parents(t)\right)$
{
set $ready\ Time \leftarrow -1$
if $EST(t_p) + RM_{min}(t_p) + DTT(t_p, t) > readyTime$
{
$readytime \leftarrow EST(t_p) + RM_{min}(t_p) + DTT(t_p, t)$
$parenttask \leftarrow t_p$
}// End of if
}// End of For
$pcp \rightarrow parenttask \cup pcp$
Remove $parenttask$ from US
}// End of While
Return PCP

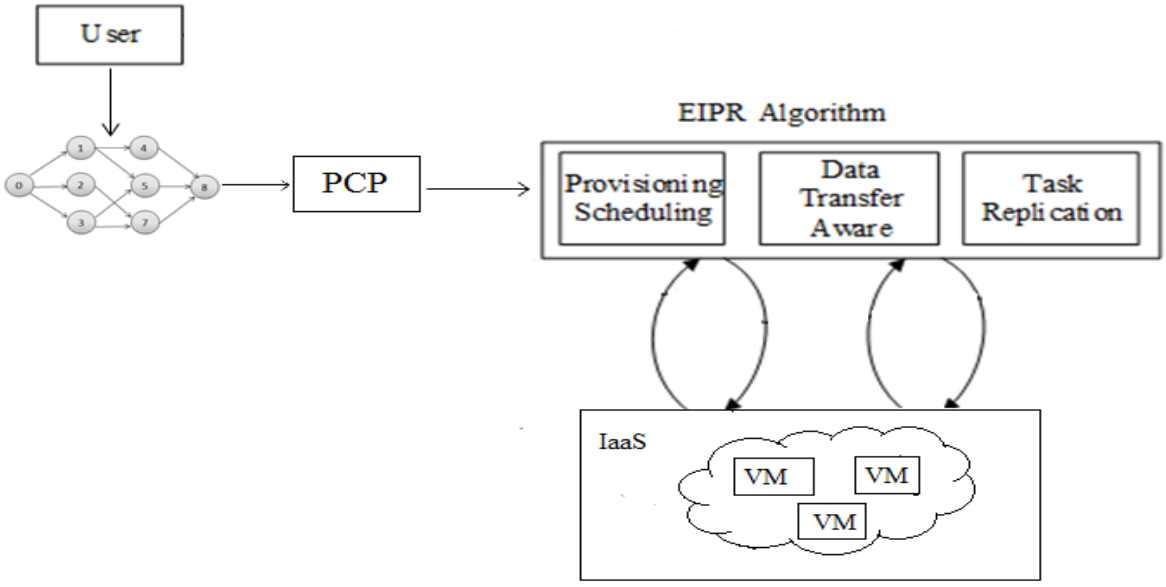

Fig 1: System Architecture

3.4 Combine Provisioning & Scheduling

Provisioning means finding the number and types of virtual machines required for the workflow execution whereas scheduling refers to find out an order of task to allocate to virtual machines. This step of algorithm requires the partial critical path of task as an input to the algorithm.

The received PCP is scheduled at start. After scheduling, check that if any task is violating its LFT or NOT. If it violates it's LFT then PCP is scheduled at end and again check for the LFT violation. If it again violates then assign the cheapest VM to each parent task list of each task of PCP otherwise the final schedule is obtained.

3.5 Data Transfer Aware Provisioning Adjust

The first step algorithm only finds the number & type of Virtual Machine. This step of algorithm determines the starting time and ending time of each virtual machine. For calculating those values, consider the data transfer time along with the start time and end time of scheduled task. To find out start time of VM consider all tasks allocated to it. For all tasks, take parent task list of each task and select the maximum data transfer time of parent task.

Start time = Start time - Data Transfer Time – Boot time

Assume boot time between 0 and 1.

To find out end time of virtual machine consider the child task list of all tasks allocated to it. Select maximum data transfer time of child task.

End Time=End Time + Data Transfer Time

3.6 Task Replication

The proposed algorithm tries to use idle time slots of provisioned virtual machines for replicating task. Replication refers to creating multiple copies of something. There are three different replication techniques such as Active replication, Passive replication and semi active replication. In Active replication, processing is done parallel on performance independent host only general agreement protocol masks the faulty replicas. In passive replication, processing is done at only single site; others will receive only status of the execution. Semi active replication is similar to the active replication only common decision is taken for all replicas. Proposed algorithm uses the semi active replication to increase the performance. Replication can be space replication or time replication. In space replication same task is executed on different machines whereas in time replication, task is executed multiple times on single machine. The proposed algorithm uses space replication. The detailed process of task replication is described in figure 2.

The Idle slots of virtual machine can exist due to dependency between task means child task has to wait for the data being generated by the parent task. Another reason to exist idle slot is partial use of provisioned time. The proposed algorithm will use idle slots exist by the second reason. These idle slots are paid slots or unpaid slots. Paid slots are the time the VM is provisioned but no task is allocated to it, and unpaid slots are the time between start and deadline of workflow where the VM is not provisioned. After finding the paid and unpaid slots replication precedence order has to find. Replication precedence order can be finding out based on three criteria which are 1. Ratio between execution time and lag time where lag time is calculated by $LFT(t) - ST(t) + R_{tv}$. 2. Execution time and 3. Number of children.

Rather than using these criteria to calculate replication precedence order, find the rank for each task to replicate. In this, ranking of task is find out by calculating the computational cost and data transfer cost. Computation cost is the time in seconds to execute task in vm. Data transfer cost is the time in seconds to transfer all files from each parent to each child. Finally the rank is the sum of average computation cost and maximum transfer cost for that task.

4. SIMULATION ON CLOUDSIM

CloudSim is a new open source toolkit developed using java that generalized, and advanced simulation framework allows simulation of Cloud computing and

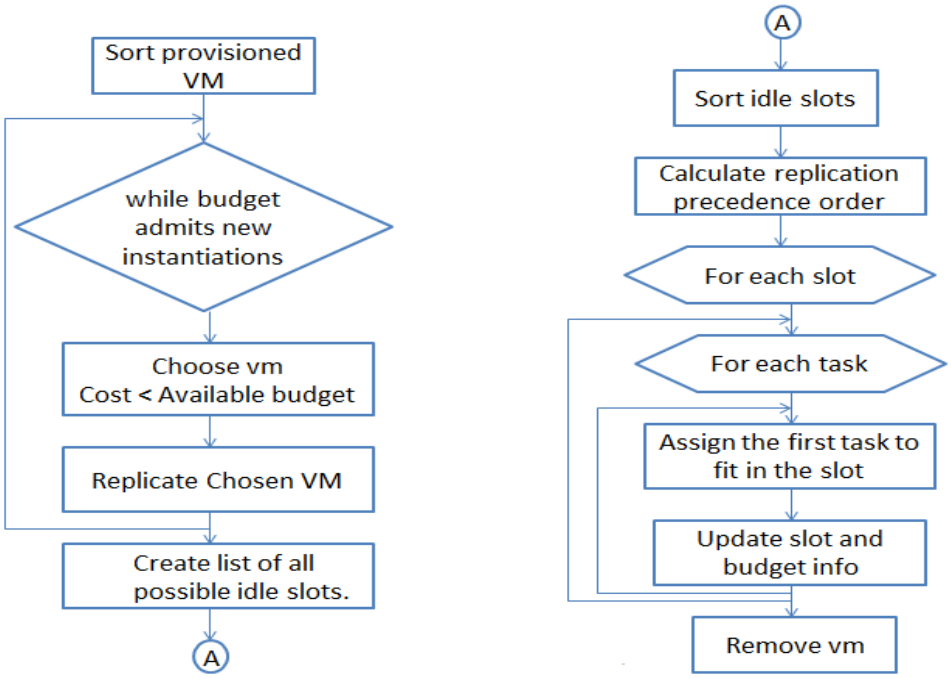

Figure 2: Task Replication Process

application services. CloudSim is a simulation tool for creating cloud computing environment and used as the simulator in solving the workflow scheduling problem. CloudSim allows us to create a data center with a set of hosts and number of virtual machines as resources. Each task of a workflow can be assigned to appropriate virtual machine once it's all parent tasks get executed.

4.1 Simulation Description

Virtual Machine - It is implemented virtual software of a computer that executes application programs same as a physical machine.

Cloudlet - Cloudlet is input job or set of tasks to be executed in cloud environment. Cloudlet has its own unique Cloudlet_id, and Cloudlet_length.

The result analysis was conducted on Dell PC with 2.0 GHz Intel i3 CPU and 4 GB of memory running windows 8 and CloudSim. Cloudsim is used to construct six virtual machines in single data center. The XML files of workflow are given as Input to algorithm. These workflows contain the number of tasks, and these tasks are provided for scheduling. Result evaluation is done based on the execution time required for different techniques such as scheduling without replication, with replication, increasing budget for replication, and by changing the order for replication.

Table 1 shows the execution time required by four different scientific workflows. In this table Without Rep denotes the execution of workflow without performing replication step. Bud=10 denotes Replication Budget and Rank Change denotes the execution of workflow with change in order of replication. Table 2 shows the execution cost required by the four different scientific workflows.

Table 1: total Execution time (sec) for workflow having tasks=50

	Montage	Cybershake	Sipht	Ligo
Without REP	175.7378	326.4874	2952.76	3835.0825
Rep	126.1080	322.73	2815.864	2605.020
Bud=10	122.9713	318.1585	2808.50	2409.505
Bud=15	90.31382	315.15	2804.57	2400.005
Rank_change	123.9384	310.4697	2805.94	2562.3836

Table 2: total Execution cost for workflow having tasks=50

	Montage	Cybershake	Sipht	Ligo
Without REP	547.92	1045.12	8858	12668.79
Rep	389.15	1033.86	8420	8253.42
Bud=10	394.91	1036.07	8424	8259.12
Bud=15	397.73	1039.14	8429	8263.24
Rank_change	382.15	1030.37	8417.87	8124.35

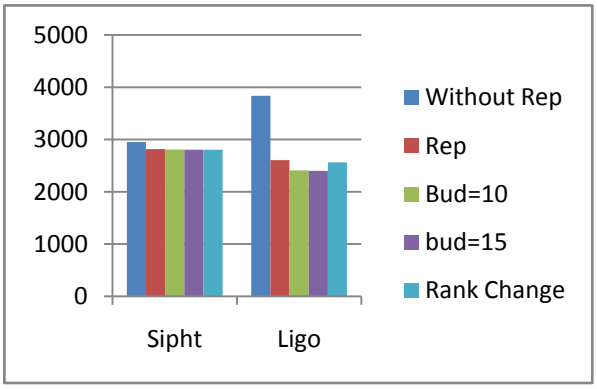

Figure 3: Total Execution time (sec) for Sipht and Ligo Workflow

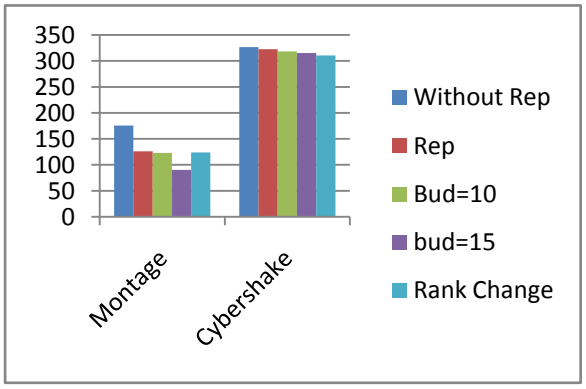

Figure 4: Total Execution time for Montage and Cybershake Workflow

5. CONCLUSION

Workflows are gaining the importance in scheduling by moving it from Grid computing to Cloud computing. Existing algorithms for executing of scientific workflows in Clouds either try to minimize cost or to minimize execution time. This paper presents enhanced ICPCP with task replication algorithm to complete workflow execution within user defined deadline and smaller possible cost. The algorithm completes the execution within deadline by replicating tasks on virtual machine when they are not in use. As shown in the simulation result, algorithm completes execution with the replication. If replication budget is increased, faster the algorithm completes its execution. Also if replication order of task gets changed by using its data transfer cost and computation cost, the algorithm performs better than previous.

As a future work, capabilities of Enhanced ICPCP algorithm with Task Replication can be increase by replicating task in multiple clouds.

6. REFERENCES

[1] Bowman, M., Debray, Rajkumar Buyya Rodrigo N. Calheiros,"Meeting Deadlines of Scientific Workflows in public Clouds with Tasks Replication,"*IEEE*, vol. 25, no. 7, pp. 1787-1796, July 2014.

[2] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, ‘‘Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility,’’ Future Gener. Comput. Syst., vol. 25, no. 6, pp. 599-616, June 2009.

[3] Mahmoud Naghibzadeh , Dick H.J. Epema Saeid Abrishami, "Cost-Driven Scheduling of Grid Workflows Using Partial Critical Paths, " *IEEE*, vol. 8, p. 23, August 2012.

[4] M. Naghibzadeh, and D. Epema S. Abrishami, "Deadline-Constrained Workflow Scheduling Algorithms for IaaS Clouds," *Future Generation Computer System*, vol. 29, no. 1, pp. 158 - 169, January 2013.

[5] C. Lin and S. Lu, ‘‘SCPOR: An Elastic Workflow Scheduling Algorithm for Services Computing,’’ in Proc. Int’l Conf. SOCA, 2011, pp. 1-8.

[6] LinlinWu , Siddeswara Mayura Guru , Rajkumar Buyya Suraj Pandey, "A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments," in *IEEE*, Perth, WA, April 2010, pp. 400 - 407.

[7] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, C´esar A. F. De Roseand ,Rajkumar Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," SOFTWARE – PRACTICE AND EXPERIENCE, pp. 23-50, 2011.

[8] The XML files that describe the workflow applications are available via the Pegasus project: https://confluence.pegasus.isi.edu/display/pegasus/WorkflowGenerator.

[9] Y. Yang, K. Liu, J. Chen, X. Liu, D. Yuan and H. Jin, “An Algorithm in swindew-C for Scheduling Transaction-Intensive Cost-Constrained Cloud Workflows” in Proceeding of 4th IEEE International Conference on e- Science, pp. 374-375, 2008.

[10] CLOUDS. [Online]. http://www.cloudbus.org/cloudsim/