REPLICATION-BASED TASK DEPLOYMENT APPROACH FOR EFFICIENT SCHEDULING OF SCIENTIFIC WORKFLOWS IN PUBLIC CLOUD

B.R. Mandre

SSVPS BS Deore College of Engineering, Dhule, India
bmandre@gmail.com

ABSTRACT: *Pointing at the current problems concerning workflow scheduling management for lowering the hiring cost of cloud services, this paper has focused on task scheduling and proposed an innovative approach based on task replication with a change order mechanism. Most previous works, generally, utilize a series of algorithms through optimizing target hosts with a process cycle and then choosing the optimal target hosts to achieve minimum turnaround time. However, it does not assure optimum cloud service cost. Based on this argument, the paper introduces an enhanced approach for cost optimization by studying the task dependencies in the task network of the workflow job and replicating the tasks on idle virtual machines. The workflow represented as task network shows a path with minimum idle time; known as the critical path. It basically employs the modified critical path and task replication. The simulation outcome shows that; the proposed algorithm decreases the execution time and cost for the given scientific workflow.*

Keywords: Cloud Computing, Workflow Scheduling, Task Deployment, Critical Path

1. INTRODUCTION

Cloud computing has been emerging as one of the most powerful, valuable, and encouraging research directions after grid computing and distributed computing [3, 4]. It provides cloud services concerning infrastructure, software, and platforms for users through the Internet. The services operate on an on-demand & commercial basis. The most significant mode is the Infrastructure as a Service (IaaS) which focuses on supplying user services with a large number of physical hosts deployed in the cloud center. The cloud center has scheduler software which is responsible for efficiently managing resources by allocating the required computing resource to the requesting task. The computing resource is a physical host and its computing time. Each host has different commercial pricing based on computing power and memory. Cloud center is constantly receiving task requests and it is required to handle these tasks by choosing a particular host with sufficient resources. The task's budget is also considered while choosing the host [5].

Any process either scientific or business can be represented in the form of workflow. This workflow model of scientific application can be found in the areas of astronomy, physics, and bioinformatics. The basic way to represent the workflow is by describing it as a directed acyclic graph (DAG). It represents tasks as nodes and task dependencies as vertices [2]. In practice, an average scientific workflow has thousands of tasks. To execute these workflows, a large-scale computing infrastructure is required. The public cloud can be one of the most suitable infrastructures for these workflows.

The public cloud in IaaS represents a pay-per-use system which is an important concern regarding pricing in the commercial cloud. For simplicity, we refer to IaaS cloud providers as cloud providers. The Cloud also provides dynamic scaling in the response to the application needs. In turn, it avails of many features like on-demand provisioning of the resources based on budgeting and optimizing billing costs.

The tasks of workflows are variable in nature. The main issue in the workflow management system is workflow scheduling because it is challenging to identify the available resource from the central pool of resources at the time of execution of the workflow. Workflow scheduling represents the discovery of a correct execution sequence for the workflow tasks, that is, execution that follows the constraints which represent the business logic of the workflow. Mapping and management of workflow tasks' execution on shared resources is done with the help of workflow scheduling [6, 7].

The cloud environment does not provide regular performance for execution time and data transfer time. While executing High-Performance Computing applications in a public cloud environment, cloud performance varied for execution time and data transfer time. This variation requires the constraints to be applied in the provisioning and scheduling stage. It is needed to enable soft deadlines to be met. Therefore, we need to propose the application of task replication to reduce the effect of performance variation of Cloud resources in the workflow execution time. Consider the classical motivational example of workflow and cloud Virtual Machine model given by Abrishami et al. [2, 8].

The main contributions of this article are as follows:

- The optimal solution of task scheduling is proposed for scientific workflows.
- The development of a new scheduling algorithm with Improved IaaS Cloud-Partial Critical Paths with Replication techniques.
- The proposed algorithm is evaluated using extensive simulation experiments on various performance metrics.

The rest of this paper is organized as follows: in the second section, the related work of the approaches for workflow scheduling for the scientific workflows will be briefly introduced. In the third section, the proposed system is described. The fourth section describes the design and implementation of the proposed. The fifth section shows simulation experiments and results, arguing that the proposed approach has high efficiency. And the conclusion of the whole paper is made in the sixth section.

2. Related Work

The main aim of the task scheduling algorithm is to perform the effective and optimal execution of the workflow tasks. There is no specifically defined solution available for the problem of workflow task scheduling. In this process of scheduling the tasks, generally, list scheduling algorithms are used. The list scheduling algorithms focus on two distinct phases – one for prioritizing the tasks and the second one for allotment of processors for the tasks.

S. Abrishami proposed an algorithm based on QoS on workflow scheduling on utility Grids [1]. It is called the Partial Critical Paths (PCP) algorithm. The goal of this algorithm is to create a schedule that reduces the total execution cost of a workflow, with satisfying a user-defined deadline. The algorithm comprises two main stages: Deadline distribution and Planning. The Deadline Distribution allocates the overall deadline of the workflow across individual tasks. In the Planning stage, it plans each task on the cheapest service that can execute the task before its sub-deadline.

There are three key distinctions between the cloud model that is commercial Clouds and the model of utility grids. The first is the dynamic or on-demand resource provisioning feature in the Clouds that allows this scheduling system to decide the nature and quantity of resources required, whereas in the utility Grids the resources are set and limited resources, with limited timeslots available. Another difference is the uniform bandwidth between the services of a Cloud provider, in contrast to the diverse bandwidth among service providers that are part of the Grids for utility use. Another difference is the pay-as-you-go pricing model that is currently used by commercial Clouds which charges users based on the number of time intervals they've utilized.

In 2014, Murillo and Prodan developed the “Multi-Objective, Heterogeneous Earliest Finish Time (MOHEFT)” method to be an enhancement to the familiar DAG scheduler HEFT [9]. This is a heuristic-based algorithm that calculates a set of Pareto methods that users can choose the most suitable one. MOHEFT creates numerous transitional workflow plans, with each step analogous. The efficiency of these results is assured through dominance relationships, and their variety is guaranteed through the use of the metric called "crowding distance". This algorithm is universal in terms of the variety and number of goals it is able to handle but costs and makespan were improved when executing workflow software on Amazon cloud.

In 2015, Malawski et al. propose mathematical models that optimize how much it costs to schedule workflows when there are deadline constraints. The proposed method is an overall optimization of data and task positioning by defining the scheduling issue as a "mixed integer Program". Alternate forms of this algorithm have been described. The first one is for "coarse-grained workflows" where the activities run for approximately an hour. The other is for the second designed for "fine-grained workflows", with numerous tasks that are short and have deadlines of less than one hour [10].

In 2015, the Security-aware and budget-aware (SABA) algorithm has been developed to plan workflows within multi-cloud environments [11]. The authors outline the concept of immovable and moveable datasets. Data that is movable does not have safety constraints, and therefore can be transferred among data centers and can be replicated when needed. Data that is not movable, on the other hand, are limited to one data center and are not able to be replicated or moved because of safety or price issues. The algorithm has three major stages. The prioritization and clustering phase is where activities and information are allocated to distinct data centers according to the workflow's irremovable datasets. Furthermore, the priority assignment is made to tasks based on the computation and I/O expenses in relation to a basic type of Virtual Machine.

Considering these variances, S. Abrishami et al. adapt the Partial Critical Path algorithm and propose two new workflow scheduling algorithms applicable to IaaS Clouds [2]. They are called the IaaS Cloud-Partial Critical Paths (IC-PCP) and the another is IaaS Cloud-Partial Critical Path with Deadline Distribution (IC-PCPD2). The IC-PCP is a single-stage algorithm employing a similar strategy to the deadline distribution phase of the original PCP algorithm, except that it actually schedules each workflow task, instead of assigning a sub-deadline to it.

Kaur et. al proposed a Deep-Q learning-based heterogeneous earliest-finish-time (DQ-HEFT) algorithm, which closely integrates the deep learning mechanism with the task scheduling heuristic [11]. The experiment results obtained on the workflows simulator demonstrate the efficiency of the approach compared with existing algorithms. This technique achieves significantly better makespan with a higher volume of data and can run faster compared with the existing workflow scheduling algorithms in a cloud computing environment.

3. Proposed Problem and Formulation

The IaaS cloud provides the facility to deploy tasks on the physical hosts in the resource pool. In general, the cloud data center chooses suitable physical hosts at random. This random allotment may result in a decline in responsiveness, quality of service, and computing power.

Clearly, different task deployment strategies may lead to diverse load allocation cloud systems. It may cause different execution efficiency, Quality of service, pricing, and external service capability.

Therefore, it is necessary to design and implement an efficient and load-balancing task deployment strategy in the cloud data center.

The proposed work will focus to handle the problem of completing workflow execution within a defined deadline. The Improved IC-PCP with Replication (IIPR) algorithm solves the above issues by replicating tasks on virtual machines when they are not in use. The algorithm takes workflow XML files as input with a number of tasks. This XML file contains the description of tasks and the parent-child relationship between these tasks. The algorithm schedules each task on a virtual machine, so that the task completes its execution without violating its Latest Finish Time. At last, the idle slots of virtual machines are found and the tasks are replicated on idle virtual machines when they are not in use. To replicate a task on VM replication precedence order for the task is found by considering the ratio between execution time and lag time, the execution time of the task, and cardinality of the task.

Advantages of Proposed System

- Replication has the benefit of reducing the deviations of execution time caused by performance variations.
- IIPR algorithm decreases the total execution time of workflows with the increasing budget offered for replication.

Components of The Proposed System

The following figure describes the different components with the flow of execution of the proposed system.

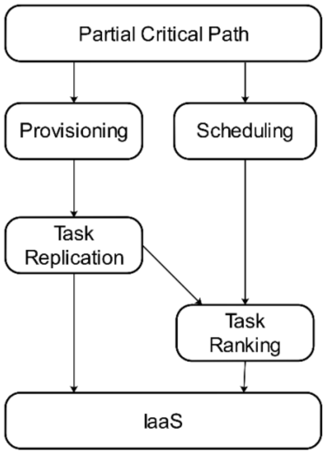

Figure 1: Components of the proposed system

It includes workflow selection and workflow size selection. Here the proper simulation parameters are input to the system including the workflow name and number of jobs in the workflow. According to this name and size, the DAX presentation of that workflow is selected. After selecting the workflow job information is exacted from the DAX. The second block uses these tasks' information to generate a Partial Critical Path. The partial Critical Path of the task is found by using the Early Start Time and Latest Finish time of the task. The task having the largest execution time and data transfer time is selected as a part of PCP. Finally, the PCP is provided as input for the IIPR algorithm. The algorithm founds the optimized schedule with task replication.

Find Partial Critical Path

The critical path of a workflow is the execution path between the entry and the exit nodes of the workflow with the longest execution time. The critical path determines the execution time of the workflow. The critical parent (CP) of $t_j$is the parent t_p, whose sum of the start time, data transfer time, and execution time to t_j is maximum among other parent nodes oft_j. The partial critical path (PCP) of node t_j is a group of dependent tasks in the workflow graph.

PCP - Partial Critical Path is calculated by identifying the unassigned parent's task nodes. An unassigned parent node can be defined as a node that is not scheduled or might not have been assigned to any PCP. It is the algorithm that minimizes the cost of deadline constraint workflow. It is done by assigning all tasks of the Partial Critical Path to the virtual machine.

Provisioning and scheduling

Provisioning refers to discovering the number and sorts of virtual machines needed for the workflow execution while scheduling refers to discovering an order of tasks to allot to VMs.

Data Transfer Aware Provisioning

The first phase of the algorithm discovers the number & type of Virtual Machine. It calculates the starting time and ending times of each VM. For determining values of starting and ending time, we need to use data transfer time with the start time and end time of the task.

Task Replication

Task replication is the process of creating replicas of tasks. The replication of tasks can be accomplished concurrently, in which all replicas of the task begin to execute simultaneously. When tasks are replicated simultaneously and the child tasks begin their execution according to the type of schedule. Task replication can be achieved through the replication of tasks within either an idle phase of resources or exclusively on the new resources.

4. Proposed Algorithm

The aim of the Improved IC-PCP using the Replication (IIPR) algorithm is to improve the chances of successfully completing the process of a scientific workflow application within a defined timeframe within the public Cloud environment, which usually has high availability but significant performance differences, through the application in task replicating. At the highest degree, the proposed algorithm is based on the following three steps in a distinct order:

1. The provision is combined from Cloud tasks and resources.
2. Provisioning of data that is aware of transfer changes
3. Task replication.

Partial Critical Path

The PCP algorithm reduces the cost of deadline restraint workflow. This is done by allocating all tasks on PCP to the virtual machine. Now let's see the details of finding the PCP.

Algorithm 1: To determine PCP for Task

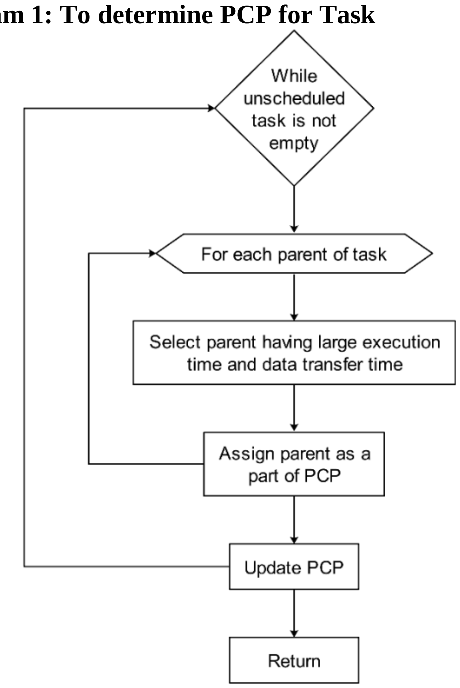

Figure 2: Flow Diagram for Partial Critical Path

Provisioning and scheduling

This process is required to determine the partial critical path of the workflow to be an element of the algorithm. The provisioning and scheduling issues are closely connected due to the fact that the availability of VMs influences the scheduling process, while the scheduling influences the time to finish for virtual machines. Therefore, better efficiency in the scheduling and provisioning process can be achieved if both issues are addressed as a whole instead of separately.

Data Transfer Aware Provisioning

It determines the beginning time and the end time of every VM. To calculate these numbers, look at the duration of the data transfer with the start and end times of the scheduled tasks. For each non-entry-related task scheduled as the first step of a virtual machine and for every task that is not scheduled as the final job of the virtual machine the algorithm will meet the required time for communication by setting the time for the start that the virtual machine DTT(i, j) earlier than ST of the initial task, or setting the deadline for the device DTT(i, J) later than the end time of the previous task, based on where the additional time is required..

To determine the start time of VM examine all tasks assigned to it. For each task, you must take your parent task list for each task, and choose the longest time for data transfer of the parent task.

Also, consider the boot time to be in the range of 0 and 1. To determine the end time of the virtual machine, you should look at the list of child tasks for the tasks assigned to it. Choose the maximum time for data transfer of the task.

Task Replication

The IIPR algorithm tries to make use of idle time slots on virtual machines to perform the replication tasks. Replication is the process of creating several copies of something.

There are three types of methods of replication, Active, passive, and semi-active replication.

When Active Replication is used, the processing occurs in parallel on an independent host.

In Passive replication the processing takes place at one location; other sites will take only the status of the execution.

Semi-active replication does work comparable to active replication but only common sense is used across all replicates. IIPR algorithm employs semi-active replication in order to boost efficiency.

In the space replication process, the same task is executed by different machines, while when it comes to time-replication, the same task is performed multiple times on one machine.

The IIPR algorithm uses space replication.

5. RESULTS AND ANALYSIS

For the implementation of the IIPR with task replication algorithm, in which a public cloud environment is used. The algorithm has three main steps which are combined provisioning and scheduling, data transfer aware provisioning adjust, and task replication. The combined provisioning and scheduling step require the partial critical path of the task. PCP of the task is the list of critical tasks having the largest execution time and data transfer time. The proposed algorithm is implemented using the CloudSim package which is used to provide the simulation of the cloud environment.

Table 1: Virtual Machines with Description

VM Id	MIPS	RAM	Bandwidth	Cores
1	1665	1741	1000	1
2	2220	3799	2000	1
3	1775	7680	1500	2
4	3330	15360	3000	4
5	3885	15360	5000	4
6	4440	30720	6000	8

Each workload is evaluated with three varying numbers of tasks. These varying numbers of tasks are named as application size. Table details the number of tasks composing each application in each of the sizes: medium, and large.

Table 2: Number of tasks for workflow

Workflow Name	Medium	Large
Montage	50	100
CyberShake	50	100
LIGO	50	100
SIPHT	60	100

The result analysis was conducted on a PC with a 2.0 GHz Intel i5 CPU and 16 GB of memory running windows 7 and CloudSim simulator. The CloudSim simulator can be utilized to build six virtual machines in one data center. The XML workflow files are input to the algorithm. The workflows are comprised of a number of tasks. These tasks are used for the purpose of scheduling.

Table 3: Execution cost for workflow (tasks=50)

Work flow ⇨ Method	Montage	Cybershake	Sipht	Ligo
Without REP	547.02	1045.02	8855	12668
Rep	389.11	1033.16	8425	8253.02
B=10	394.21	1035.07	8420	8259.02
B=15	397.03	1038.14	8422	8263.04
Order change	382.05	1032.37	8417.07	8124.05

Table 3 shows the time needed by four different workflows in science. In this table, Without Rep is the term used to describe the execution of workflows without the replication step. B=10 is a reference to the Replication Budget. Order Change indicates the workflow execution by changing the sequence of replication.

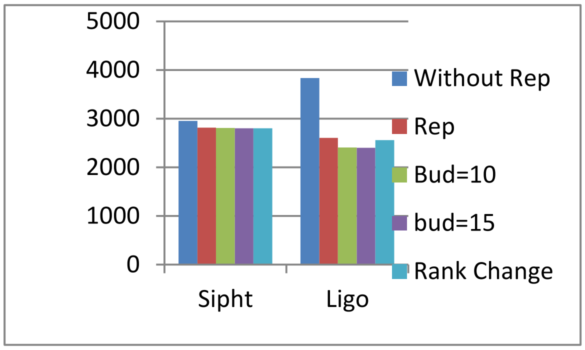

Figure 4: SIPHT and LIGO – Total Execution time (msec)

It clearly shows that the workflow execution with replication is more efficient than that without replication in both the sample scientific workflows. It also shows that if the pricing budget is more, then the time required for the workflow execution is less.

6. CONCLUSION

This paper presents IIPR with a task replication algorithm. This algorithm is mainly concerned with scientific workflows. The main parameters considered are the execution within a user-defined deadline and lesser possible cost. The result shows the execution cost essential for the four scientific workflows. As shown in the table, execution cost gets reduced if task replication is done. The cost of the execution gets increased if we increase the replication budget. The cost of execution is also get reduced by changing the order of the replication.

REFERENCES

1. S. Abrishami and M. Naghibzadeh, "Deadline-constrained workflow scheduling in software as a service Cloud," Scientia Iranica, Transactions D: Computer Science & Engineering and Electrical Engineering," pp. 680-689, 2012.
2. S.Abrishami, M.Naghibzadeh, and D.Epema, "Deadline-Constrained Workflow Scheduling Algorithms for IaaS Clouds," Future Generation Computer System, vol. 29, no. 1, pp. 158 - 169, January 2013.
3. Amid Khatibi Bardsiri Seyyed Mohsen Hashemi, "Cloud Computing Vs. Grid Computing," ARPN Journal of Systems and Software, vol. VOL. 2, pp. 188-194, May 2012
4. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A View of Cloud Computing," Communications of the ACM, vol. 53, no. 4, pp. 50-58, 2010.
5. J. M. Bahi, S. Contassot-Vivier, and R. Couturier, "Dynamic Load Balancing and Efficient Load Estimators for Asynchronous Iterative Algorithms," IEEE Trans. Parallel and Distributed Systems, vol. 16, no. 4, pp. 289-299, 2005.
6. L. F. Bittencourt, and E. R. Madeira T. A. Genez, "Workflow scheduling for SaaS/PaaS cloud providers considering two SLA levels." in Proceedings of the IEEE Network Operations and Management Symposium (NOMS), pp. 906-912, 2012.
7. J. Yu and R. Buyya, "A taxonomy of scientific workflow systems for grid computing," SIGMOD Record, vol. 34, no. 3, pp. 44-49, 2005.
8. E.K. Byun, Y.S. Kee, J.S. Kim, and S. Maeng, "Cost Optimized Provisioning of Elastic Resources for Application Workflows," Future Generation Computer System, vol. 27, no. 8, pp. 1011 - 1026, October 2011.
9. J. J. Durillo and R. Prodan, "Multi-objective workflow scheduling in amazon ec2 Cluster Computing." IEEE, vol. 17, no. 2, pp. 169-189, 2014.
10. K. Figiela, M. Bubak, E. Deelman, and J. Nabrzysk M. Malawski, "Scheduling multilevel deadline-constrained scientific workflows on clouds based on cost optimization." Scientific Programming, vol. 2015.
11. Kaur, Avinash, et al. "Deep-Q learning-based heterogeneous earliest finish time scheduling algorithm for scientific workflows in cloud." Software: Practice and Experience 52.3 (2022): 689-709
12. Poola et al, Saba, "A security-aware and budget-aware workflow scheduling strategy in clouds." Journal of Parallel and Distributed Computing, vol. 75, pp. 141-151, 2015.