Modified Scheduling Service workflow in Hybrid Cloud

Thesis submitted in partial fulfilment
of the requirements for the degree of

Master of Technology

in

Computer Science and Engineering

by

Toseef Ahmed Ansari

(Roll No: 213CS1137)

under the guidance of

Prof. Pabitra Mohan Khilar

Department of Computer Science and Engineering
National Institute of Technology, Rourkela
Rourkela-769 008, Odisha, India

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.

Certificate

This is to certify that the work in the thesis entitled " ***Modified scheduling workflow in Hybrid Cloud"*** submitted by ***Toseef Ahmed Ansari*** is a record of an original research work carried out by him under our supervision and guidance in partial fulfillment of the requirements for the award of the degree of Master of Technology in Computer Science and Engineering, National Institute of Technology, Rourkela. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Prof. Pabitra Mohan Khilar
Assistant Professor
Department of CSE
National Institute of Technology
Rourkela-769008

Place: NIT,Rourkela-769008
Date: 24 - 05 - 2015

Acknowledgment

First of all, I would like to express my deep sense of respect and gratitude towards my supervisor Prof. Pabitra Mohan Khilar, who has been the guiding force behind this work. I want to thank him for introducing me to the field of Cloud Computing and giving me the opportunity to work under him. His undivided faith in this topic and ability to bring out the best of analytical and practical skills in people has been invaluable in tough periods. Without his invaluable advice and assistance it would not have been possible for me to complete this thesis. I am greatly indebted to him for his constant encouragement and invaluable advice in every aspect of my academic life. I consider it my good fortune to have got an opportunity to work with such a wonderful person.

I wish to thank all faculty members and secretarial staff of the CSE Department for their sympathetic cooperation.

During my studies at N.I.T. Rourkela, I made many friends. I would like to thank them all, for all the great moments I had with them.

When I look back at my accomplishments in life, I can see a clear trace of my family's concerns and devotion everywhere. My dearest mother, whom I owe everything I have achieved and whatever I have become; my beloved father, for always believing in me and inspiring me to dream big even at the toughest moments of my life; and my sister; who was always my silent support during all the hardships of this endeavour and beyond.

Toseef Ahmed Ansari

Abstract

Workflow is used to represent variety of application which requires massive data computation and storage. To overcome this need of data computation and storage cloud computing has emerged as one of the best solution for on demand resource provider. But sometimes the resources available to us may not be sufficient, so the need arises to gather more sources from other clouds. This is done by using the Hybrid cloud. Hybrid cloud is combination of public and private cloud. The private cloud is owned by the user thus there are no extra charges for using the resources available in it, whereas public cloud is owned by others so we have to pay for the using the resource as per the uses. The use of the hybrid cloud provides elasticity to the user. While using the hybrid cloud, two most important question arises. The first one is how to divide the workflow. And the second one is that which resource we need to borrow from the public cloud so that it can meet our requirement within the specified deadline. The modified scheduling service workflow for hybrid cloud give the less makespan for the DAG than the original algorithm and provide us the best resources that we need to borrow from public cloud so as to have enough processing power to schedule the workflow within given deadline.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Service Model

Deployment Model

Hybrid Infrastructure

Motivation

Problem Statement

CHAPTER 1

Introduction

1.1 Introduction

The most efficient way for data computation and storage is the use of cloud. The main benefit of using the cloud is that it provides elasticity to computing environment. It is able to adapt to the user current need as cloud environment provide the borrowing of the resources by the user as per there need. And it also reduces the initial investment that we make to buy the resources that are required by the user to perform the task. Not only that it also saves the cost of maintenance and also lowers the operation cost and it also provides scalability.

The abstract way of representing the computing resources in a subset or a logical group is called virtualization. this abstraction have benefit over the original configuration. Through virtualization we are creating an interface for virtual machine which contain the virtual resources like network connection, physical memory, central processing units and peripherals. The virtual machines created have their own operating system, network services and applications. In other words they are similar, they are independent. Because of the virtualization same machine can have application isolation, server consolidation as well as hardware normalization. The most used virtualization technology is Xen [1]. Cloud user can have different software and hardware configuration which are provided by the cloud provider. Because of Xen, one user of the cloud can manage and configure the machines without affecting the other users.

The abstraction in cloud computing is implemented by using virtualizaton. Because of this the resources can be represented in logical group, which much better than the original configuration. For vitualization, Xen [1] is the most used technique. Because of this the hardware can be abstracted by creating the interface to the VMs. Each VM has its own resources like physical memory, interconnection, CPUs and peripherals. Not only, this each VM has its own application and operating system.

The Fig. 1.1 shows the service models and deployment model that are present in the cloud.

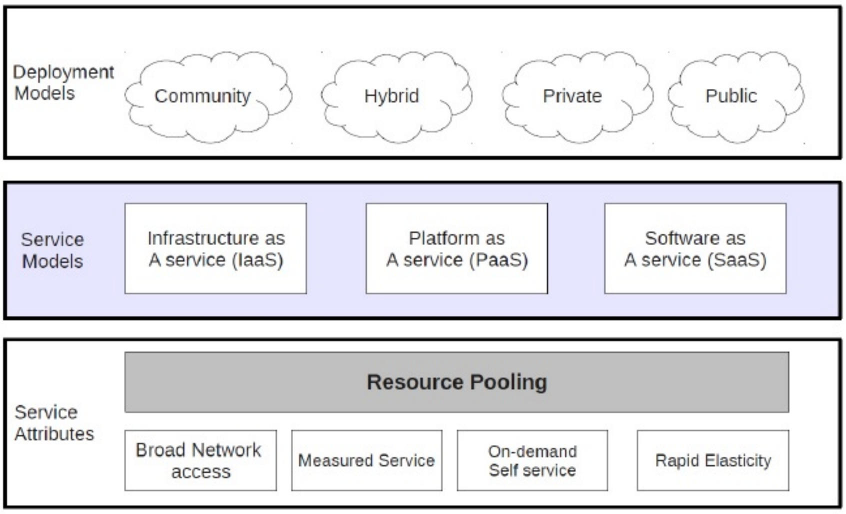

Figure 1.1: The NIST Model of Cloud Computing

1.2 Service Models

While deploying the cloud, each vendor provide service according to them. The service provided by them add a type of service in cloud. The service models are generally represented as XaaS or $\langle$ Something $\rangle$ as a Service.

There are three types of universally accepted services [9]. They are:

1.2.1 Software as a Service (SaaS)

The first one among the services is the Software as a Service. This is widely used service as it provides the preinstalled applications along with the infrastructure. For accessing this service, the just need have a device with a web browser.

In this service architecture the software to be used is hosted on the centralized network server. SaaS is the most popular service in cloud computing as it provides flexibility along with the services. And its maintenance cost is also low. The examples of SaaS are google docs, yahoo mail, etc.

SaaS is so popular because it provide the service at lowered cost as the software can be accessed at a much cheaper price than the buying price of the software. And the user need not have to worry about the installation and upgradation of the application.

1.2.2 Platform as a Service (PaaS)

The second service model that is commonly available is Platform as a Service. This service gives the user/developer the freedom to host their own application on an available infrastructure. The main advantage of using this service is that, now the user/developer do not have to worry about the underlying hardware. They can use any of the available artificial language supported by cloud service provider to create their own application in that environment. Few examples are SalesForce.com, force.com [6].

1.2.3 Infrastructure as a Service (IaaS)

The final model available is Infrastructure as a Service. In this the basic hardware resources are available to deploy our own framework. The costumer will customize their entire framework with the assistance of virtual machines, memory, virtual network etc. The consumer doesn't have to manage the physical resources, as they're supplied with virtual resources which can be managed programmatically. It delivers the computing infrastructure as a totally outsourced service. A number of the businesses that give infrastructure as a service are Google, IBM, Amazon.com etc. Virtualization allows IaaS suppliers to supply virtually unlimited instances of servers to customers and build efficient use of the hosting hardware.

1.3 Deployment Model

The aim and location of the cloud is defined by the Deployment model [9]. Based on the ownership, the cloud is classified into four deployment model [9] private cloud, public cloud, community cloud and hybrid cloud.

1.3.1 Private Cloud

Private Cloud is owned by a private firm. Sometimes, the Private Cloud is also referred as Company or internal cloud. This is mainly provide the services to a restricted variety of individuals behind a firewall. The private cloud can be on or off campus and can be managed by that organisation or third party. This offers organization the price advantages of virtualization.

1.3.2 Public Cloud

A pubic cloud is nothing but a cloud that is organised and managed by two or more organisation. It can be seen as an extension to the private cloud. Public cloud provides the resources like application and storage to the public over the internet. The services offered here are either free or available as pay per use basis. This is preferred by many because its cheap, fulfil the needs of the user and also save the resources as the resources which are required are used.

1.3.3 Community Cloud

Community cloud is nothing but a cloud formed by multiple organisations with common computing needs. These needs can be audit necessities, or can be related to performance necessities, like hosting applications that need a fast latent period. Like private cloud, community cloud can also be hosted on or off campus. And it can also be managed by the organisation building it or by a third party.

1.3.4 Hybrid Cloud

A hybrid cloud is formed by combining at least one private and one public cloud. Hybrid cloud is formed by a seller who incorporates a private cloud and form a partnership with the public cloud supplier or vice-versa. For using the hybrid cloud, an organisation manages and provide the resources that are available to itself and also have few resources available from outside. Hybrid cloud is used by the organisations when they are short on the resources to complete some task and the task is really important to them that they cannot expose it to third party.

1.4 Hybrid Cloud Infrastructure

Today there is a need of on-demand computing. And it requires flexibility, availability and scalability. To have these requirements we have to either add new resources or we have update the resources while keeping in mind that the executing processes are not affected. The hybrid cloud infrastructure requires the combination of public clouds and service oriented grid which are implemented using a dynamic service deployer, which makes the grid work as a private cloud.

The hybrid system is a combination of private cloud and public cloud [3], of which we can access and use a few resources. While using the hybrid cloud, the workflow management should provide the facility of submitting the task on the private resources first. The workflow manager should be able to utilize the resources available in private cloud to their best utilization. And it should also has some dynamic deployment facility. Not only this, it should be able to communicate with the public cloud resources so that they can be obtained as and when required. The public resources can hired when the local resources are not able to complete the workflow within some necessities. In this way, we are able to increase the computational power of private cloud without adding the new resources so that the on demand computing requirements can be met. By providing these requirements, the hybrid cloud infrastructure supports the decision making done by the scheduling algorithm.

1.5 The Cloud Interconnection

When we need to borrow the resources from public cloud we use the DDVR [3]. It has two group services. The first one is the DDVR. It takes care of the functionalities and also communicates with the infrastructure. The second one is Cloud Interface Service (CIS). It is used to interface with the public cloud. It allows the transparent access to the resources in public cloud which are same as for the resources available in the private cloud. For the transparent access, the CIS encapsulates the specific particularities of each cloud type (Nimbus, Amazon, Eucalyptus, etc.). One instance of couple DDVR/CIS is required for binding the between workflow manager and the cloud resources for each resource available. To use the cloud resources and request a resource a series of communication takes place. First the GPO communicates with the DDVR, then DDVR communicates with the cloud through CIS. Then the cloud returns the local value of th VM hired. And now the DDVR can start using it by starting the GT4 and it also creates a DDVR instance in the cloud resource.

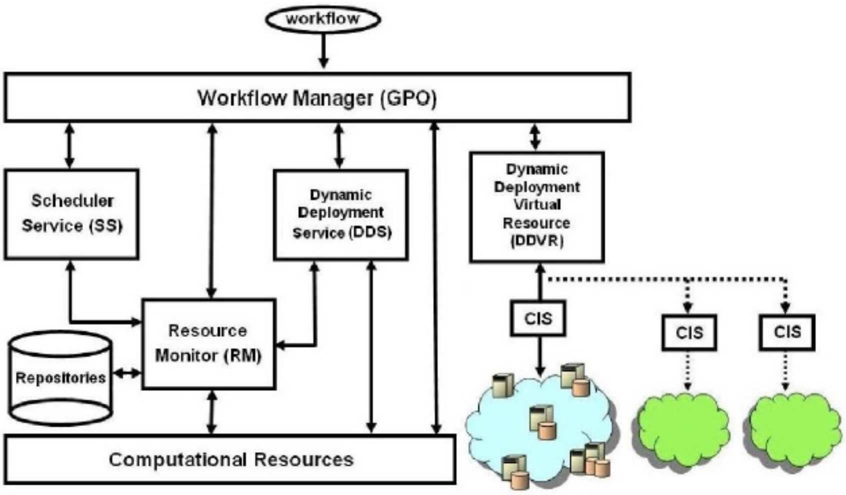

Figure 1.2: Hybrid Architecture

The DDVR/CIS allows the access to different clouds in an independent and simultaneous manner. Not only that it also provide the flexibility and scalability to the infrastructure. Besides that, it also provide the dynamic intiation of the services that are not available in public cloud. This is achieved because it has transparent access o public cloud which is provided by DDVR/CIS.

1.6 Problem in Hybrid Cloud

Hybrid Cloud requires careful determination of so as to have the best split between public and private cloud. Here the problem arises because the workflow consists of dependent tasks. The problem also includes the dividing of work on heterogeneous resources with heterogeneous link and money charged for using the public cloud.

While using the hybrid cloud, we have to carefully split the wokflow so that the task can be scheduled on the public and private cloud components. This splitting of the workflow is one of the major problem that arises in hybrid cloud because the task present in the workflow are dependent. Not only that, the resources available are heterogeneous in nature and the links connecting them are also heterogeneous in nature. The resources that are borrowed from public cloud have some cost. So we have to keep them in mind also and try to minimize the cost for using them. So in this thesis, we are presenting a way to divide the workflow of dependent tasks on private and public resources so that the workflow can be completed within a deadline D. And this work also tries to minimize the cost for using the public resources.

1.7 Motivation

The problem of scheduling the task for computation is not new. It is one of the basic problem still exists. Sometimes the resources available to us are not enough for scheduling the task within a constraint. This constraint can deadline or budget or any other user specified QoS, but in most of cases it is deadline. So for scheduling the workflow within a deadline we have to either add new resources or we can borrow the resources available to other users. But installing new resources is a costly business and it may time to install them. So the other option available is to borrow the resources from others in pay per use basis. This is where the concept of hybrid cloud comes in play.

In hybrid cloud the scheduling problem gets a bit complicated. Because now we have to think about how to divide the workflow so that it can scheduled on private and public resources. If we schedule most of the task on public resources then we may end up paying much more than we have anticipated. And if we give less task to the public resources than we miss deadline. So we have to transfer minimal task to the public resources so that the task can be completed within the deadline and we have to pay minimal for using the public resources.

1.8 Problem Statement

The main problem that arise while using the hybrid cloud is that how we will divide the workflow between the public and private resources. Our main concern here is to divide the workflow so that cost for using the public resources can be minimized and we can complete the task within a deadline and also the communication between the resources can be minimised.

Chapter 2

Introduction

CHAPTER 2

Task Scheduling Strategies in Cloud Computing

2.1 Introduction

In this chapter, we have discussed the algorithm for used for scheduling the task in a cloud environment [4]. First we will see the various type of the task scheduling Strategies used in cloud computing. Then in the next section we will review few basic algorithms which are simple to implement but can only be used for independent tasks. Then in the next section we will review the few scheduling algorithm that work on the workflow or dependent tasks which are normally represented using DAG.

2.2 Types of Task Scheduling Algorithm

2.2.1 Cloud Service Scheduling

The cloud server scheduling [4] is divided into two parts: User level and system level. The system level implemented within the datacenter and this is used for the resource management. The user level works between provider and customer for service provisioning.

Datacenter has many physical machines. And it receives millions of task from the users. It is the duty of the datacenter to assign the tasks to the physical machines. This assigning of the task is nothing but scheduling of the tasks. And it affectd the performance of the datacenter. There are many factors for scheduling like SLA, QoS, fault tolerance, resource sharing, real time constraints, reliability, etc.

2.2.2 User Level Scheduling

There are two type of scheduler for regulating the supply and demand for cloud resources which comes under the User Level Scheduling [4]. They are: Market-based and auction-based. When the resources are virtualized and have to be delivered to user as a service, we use the Market-based resource allocation policy. It is used to deal with dynamically fluctuating resource demand. The auction-based is used when the service provider provides distinct type of VMs.

2.2.3 Static and Dynamic Scheduling

When all the features are already known before scheduling, we go for static scheduling [4]. It requires the prefetching of the data. And we can also pipeline the different stages of task execution. It also gives less runtime overhead. While in the dynamic scheduling, all the informations not available before starting the execution. The allocation of task is done dynamically as the execution time of the task is not known.

2.2.4 Heuristic Scheduling

Optimization problems are considered as NP-Hard problems. To solve these problems we can use approximation method, enumeration method or heuristics method [4]. When the number of instances are small, we go for the enumeration method. In this method, optimal solution can be found by enumerating all the possible solution and comparing them one by one. But when the number of instances are large, we go for the heuristics method, as the exhaustive enumeration is not possible. Here instead of having the optimal algorithm, we go for the suboptimal algorithm which find a good solution in less time. Whereas the approximation method is used to find the approximate solution to a optimization problem. When the exact polynomial time for algorithm is known we use the approximation method.

2.2.5 Real Time Scheduling

Real time scheduling [4] is used minimize the average response time and also to increase the throughput. For maximizing the total utility, the task are scheduled non-preemptively. At the same time, we have to keep an on the two different time utility functions, profit and penalty. This type of scheduling penalizes for the abortion and also if the deadline is missed. Not only that, it also rewards when the task completes early.

2.2.6 Workflow Scheduling

A workflow is represented using a directed acyclic graph [4]. In the DAG, each node represents a task and the edges represents the inter task dependency. Workflow consists of set of tasks. And these task may communicate with each other. Scheduling a workflow is one of the major issue in management of workflow execution.

2.3 Task Scheduling Algorithm for Independent Task

This section include the simple task scheduling algorithm that can be used on the independent tasks. These are the few basic algorithms.

2.3.1 Min-min Algorithm

Min-min algorithm [10] [8] is one of the simplest algorithms used to schedule the task on the available resource. This algorithm is based on completion time of the task. For this algorithm we first find out the completion time of each task on each of the available resource. So for m task and n resources available we get a matrix of $O(mn)$. For scheduling, we find the minimum entry in this matrix and then we find the corresponding task and resource for that entry. Now we schedule that task on that resource. In order to find the next minimum value in second step, we update the available time of this resource and also eliminate the row corresponding to that task. And we repeatedly do this until all the tasks are scheduled.

One of the major drawback of using the min-min algorithm is that it assign the smaller tasks to the faster machines. And if in the batch of tasks, a larger task is present, then the makespan is determined by the execution time of this larger task. And if the number of small is really large then the larger task may starve. Another drawback is that it can only be implemented on the independent tasks.

2.3.2 Max-min Algorithm

Max-min algorithm [10] [5] was developed so as to overcome the disadvantage of the Min-min algorithm. This algorithm is also based on the completion time of the task. In this algorithm also we first find the completion time of each task on each available resource. Then we find out the maximum entry available in the matrix. The row corresponding to this entry give us the task to be scheduled. Now for the resource we find out the minimum value that is available in this row. Now we schedule the task on this particular resource. For next round we update the available time of the resource and also delete the row corresponding to the task scheduled. And again find out the next highest value. And this is repeated until all the task are scheduled.

The disadvantage of this scheduling algorithm is that it may also lead to starvation. Sometimes it may happen that the task with smaller completion time may not get scheduled. This algorithm also works only for independent tasks.

2.3.3 RASA Algorithm

The algorithm is built to overcome the disadvantage of the two well-known task scheduling algorithms, Min-min and Max-min. RASA [10] try to use the advantage of both the algorithm to cover the disadvantage. The parameter used here is again the completion time. For implementing the RASA, we have execute the Min-min and Max-min algorithms, alternatively. Min-min strategy focus on the smaller task while Max-min strategy focus on the larger tasks. And by using them alternatively, concurrency can be supported between the large and small task.

It is observed that the if the number of resources available are even, it is better to use the Max-min algorithm in first round and Min-min algorithm in the second round. Whereas, if the number of resources are odd then Min-min algorithm is preferred in first round followed by Max-min algorithm in second round.

Since RASA uses two primary algorithms alternatively it is able to hide the disadvantages of two algorithms with advantages of each other. And like Min-min and Max-min algorithm, RASA is implemented on independent task only.

2.3.4 Improved Max-min Algorithm

Improved Max-min algorithm [5] is an extension of the Max-min algorithm. Instead of using the estimated completion time of the tasks, it uses the estimated execution time of the tasks as parameter.
For this we first compute the execution time of each task on each resource available. From that matrix we find the maximum value in matrix. The row for that entry will give us the task to b scheduled. Now we find the minimum value available in that row, the column corresponding to this entry will give us the resource for that task. Before going for the next iteration we update the available time of that resource and then the matrix. And also the corresponding row is deleted. This process is repeated until all the task are scheduled.

2.3.5 Improved Min-min Algorithm

Min-min algorithm shortcoming is that the long tasks may starve. Therefore, here is an improved min-min algorithm [8] that not only based on the min-min algorithm but also the three other constraints. The three constraints are quality of service, the dynamic priority model and the cost of service.

- **QoS Constraint:**

  QOS indicates service performance of any combination of attributes. In cloud computing, QOS is the satisfaction standard of the users use cloud computing services. Cloud computing has the commercial characteristics, in order to make it has value, these attributes must be available, management, validation and billing, and when use them, they must be consistent and predictable, some attributes are even playing decisive roles. To classify and definition of the QOS purpose is to make agents manage and allocate resources, according to different types of QOS.

- **Priority Constraint:**

  The priority of task is divided into static priority and dynamic priority. Static priority is defined before the tasks being executed and it remained the same priority in the scheduling stage. Static priority can reflect the willing of the user, but, the system is dynamic, static priority is difficult to reflect the changes of the system.

- **The Cost of Service Model:**

  Cloud computing is a market-oriented applications, user pays according to the services that he uses. The users can select the services which are cheap.

2.3.6 Load Balanced Min-min Algorithm

The Min-min algorithm is a good and simple algorithm which produces the minimal makespan. While using the Min-min algorithm, some resources are very busy in processing the tasks while others are free. So to balance this load, a Load balanced Min-min (LBMM) [7] algorithm is used. This new algorithm implemented in two phase. In the first round simple Min-min algorithm is used. But since the load on the machines is not balanced, we go for the second phase. In this phase, we balance the load on the machines by rescheduling some task on the slightly loaded machines.

2.4 Task Scheduling Algorithm for Workflow

2.4.1 Scheduling Service Workflow for Cost Optimization in Hybrid Cloud

The Scheduling service workflow for cost optimization in Hybrid cloud [3] is one of the scheduling algorithm used in hybrid cloud. In hybrid cloud, the user/organisation has its own grid which work as a private cloud. And the organisation can also borrow the resources from the available public cloud on pay per use basis. Here it is important to decide when and what resource we need to borrow so that our constraint of deadline can be met and the cost for using the additional resources should be minimal. Using this algorithm, we can decide what task we can migrate to public resources and also which resource we can borrow from the public cloud.

This algorithm has two phase. In the first phase, the workflow or the dependent tasks are scheduled in the private cloud on the private resources using the Path clustering heuristics (PCH) algorithm. From this we get the makespan for the scheduled workflow. Now we check the makespan with the deadline. If deadline is not matched, then we go for the second phase of the algorithm. In second phase we try to get the resources from the public cloud and try to schedule the part of the workflow on public resource. The resource are taken from the public cloud so that the cost for using the resource is minimised.

Path Cluster Heuristics (PCH) Algorithm: This algorithm is used in the first step of the above algorithm. The baseline of the Path Clustering Heuristic (PCH) [2] algorithm is to select a path from the DAG and schedule the nodes on this path onto the same processor.

2.4.2 Cost-Optimal Scheduling in Hybrid IaaS Clouds for Deadline Constrained Workloads

This [12] is another algorithm which is implemented in hybrid clouds to optimize the cost for using resources that are available in public cloud and that too within a deadline. This algorithm uses linear programing to optimize the cost. In this algorithm each task is associated with the individual deadline. This algorithm shows that the binary integer program can support in the decision making for scheduling and it can also be partially atomization of the process.

2.4.3 Hybrid Heuristic for Scheduling Data Analytics Workflow Applications in Hybrid Cloud Environment

This algorithm [11] is a heuristics algorithm. This algorithm not only optimizes the cost of execution, but also satisfy the user constraints and requirement such as data placement, budget and deadline. This approach uses the Genetic Algorithm. It generates the initial schedule by combining the advantage of workflow level optimization with genetic algorithm. Then the deadline and budget are distributed to task by using the budget and deadline distribution algorithm. Using the Dynamic Critical Path

(DCP) algorithm, it can be used dynamically for just in time tasks. And it also fulfils the QoS requirement while utilizing the task level constraints.

2.4.4 Cost-Efficient Scheduling Heuristics for Deadline Constrained Workloads on Hybrid Clouds

This algorithm [13] optimizes the resources allocation in private and multiple public cloud, to provide a cost optimal solution within a deadline or with minima throughput. It optimizes the use of resources for batch workloads by using different heuristics scheduling. It also analyses the impact of parameters of the workload on the cost reduction that can be achieved using this algorithm.

2.5 Observations

Table 2.1: Observation on Scheduling algorithm for Independent Task

Algorithm	Parameter	Description
Min-min algorithm [10] [8]	• Completion time	• It find the resource for the task that will give the minimum completion for that task.
Max-min algorithm [10] [5]	• Completion time	• It finds the task that will take maximum time to process then it will find the resource that will give minimum time for that resource.
RASA [10]	• Completion time	• It will run the Min-min and Max-min algorithm alternatively till all the task are scheduled.
Improved Max-min algorithm [5]	• Execution time	• It is same as the Max-min algorithm, the only difference is that it will use the execution time instead of completion time.
Improved Min-min algorithm [8]	• Quality of service • Dynamic priority • Cost of service	• It uses the three addition parameter than the Min-min algorithm to schedule the task on the resources.
Load Balanced Min-min algorithm [7]	• Completion time • Load Balancing	• The scheduling is done in two phases. In first phase it will use Min-min algorithm and in the second phase it will balance the load on available resources so that almost all resources can be utilized equally.

Table 2.2: Observation for workflow scheduling algorithm

Algorithm	Parameter	Description
Hybrid Cloud for optimized Cost [3]	• EST • EFT • Deadline	• It schedule task on resources in the private cloud. if deadline is missed then we will try to borrow the resources from the public cloud so that the cost for using the public resources can be minimised.
Cost Optimal Scheduling in Hybrid IaaS Cloud for Deadline Contained Workloads [12]	• Individual deadlines	• This algorithm uses linear programming to optimize the cost of using the public resources. And each task in the workflow is associated with its own deadline.
Hybrid Heuristic for Scheduling Data Analytics Workflow Application in Hybrid Cloud [11]	• Budget • Deadline • Data placement	• Here the initial scheduling is done using the genetic algorithm. And then secondary algorithm based on Dynamic Critical path (DCP).
Cost Efficient Scheduling Heuristics for Deadline Constrained Workload on Hybrid Cloud [13]	• Deadline	• It try to optimize the resource allocation in public and private cloud so that the cost for using the resources can also be minimized.

2.6 Conclusion

In this chapter we have about few of the task scheduling algorithms that are used in a cloud computing. In Section 2.3 we have seen some the algorithm which are quite simple to work with. But they can only be implemented on the independent tasks. They work on parameters like execution time, completion time, load balancing, QoS etc. Whereas the algorithms discussed in section 2.4 works on the hybrid cloud. They are little bit complex, as they work on a given workflow or dependent task. And here we have decide which which task are executed on private resources and which tasks are executed on public resources. And also we have to decide which public resource are there that we want to hire so that the workflow can be comlpeted within a given deadline.

Chapter 3

Introduction

Modified Algorithm

Input

Initial Schedule

Output

CHAPTER 3

Proposed Algorithm

3.1 Introduction

The present algorithm for scheduling the workflow in hybrid cloud is given by L.F. Bittencourt et.al. This scheduling algorithm first schedules the task on the available resources that are present in a private cloud. In this initial scheduling algorithm it uses Path clustering heuristics (PCH) [2] algorithm. This algorithm first form the clusters from the workflow. The concept for forming the cluster is that the nodes in same cluster will be scheduled on the same virtual machine. This concept is used so that the time for intercommunication between the different virtual machine can be reduced as the there is no need to use the communication link between as it is the same virtual machine. This concept reduces the time wasted on while communicating between virtual machines.

After this initial schedule the makespan is checked against the deadline. If it misses the deadline then we will try to move some cluster from the private to public resources. For doing this we will find the resource that we will do our work faster as well as it will be less costly. Keeping these two constraints in mind we will decide which resource we have to borrow from the public cloud. Thus it will reduce the cost as well as the workflow is also completed within the deadline.

The reason for modifying this algorithm is as follows:

1. While using this scheduling algorithm, if the number of virtual machines are

enough and the clusters other than the first one are small in size, then the makespan will depend mainly on the first cluster that is formed. Because this cluster start with the starting node and end with last node and covering one node at each level. Thus the first cluster formed is largest cluster.

2. We have also seen that the last node present in the first cluster had to wait for the its previous nodes to complete the execution before it starts its execution.

Based on these to cases we have modified the initial scheduling algorithm so that the last node does not have to wait much for its predecessors.

So in the modified workflow scheduling, in the first round of forming the cluster, the node having the highest priority is selected as the head node. The second node is selected by finding the minimum value of the sum of P_i and EST_i from successor of the previously selected node. In this way we go on selecting the node until there are no successor node available.

Since we are selecting the minimum value from the sum of P_i and EST_i,we will the first cluster which will be smaller in size. Because this cluster will contain the nodes present in the shortest path of the DAG.

For the rest of the nodes, the cluster will be formed in the usual fashion,i.e., finding the cluster head by finding the maximum value of the priority from the rest of the nodes. And the rest of nodes in the are added by finding the maximum value of the sum of P_i and EST_i until there is no non clustered successor available. In this way the cluster are formed that are now ready to be scheduled on the available virtual machines.

As for scheduling the cluster on the virtual machines that are present in the private cloud, the first cluster will be scheduled on the scheduled the slowest machine available in the cloud. And as for the rest of the clusters, they will be scheduled on the fastest machines available.

This concept of scheduling will result in less waiting time of the final node since most of the previous nodes are scheduled on the faster machines. Therefore, we are getting a better performance in the private cloud. Now if the deadline is missed while utilising the private resources, we have to go for the public resources. Since the performance in first phase (private cloud) is improved, we will be able to see the improvement in the second phase also as the number of nodes to be rescheduled here will be less. Thus we are getting better results than the original scheduling algorithm.

3.1.1 Modified Algorithm

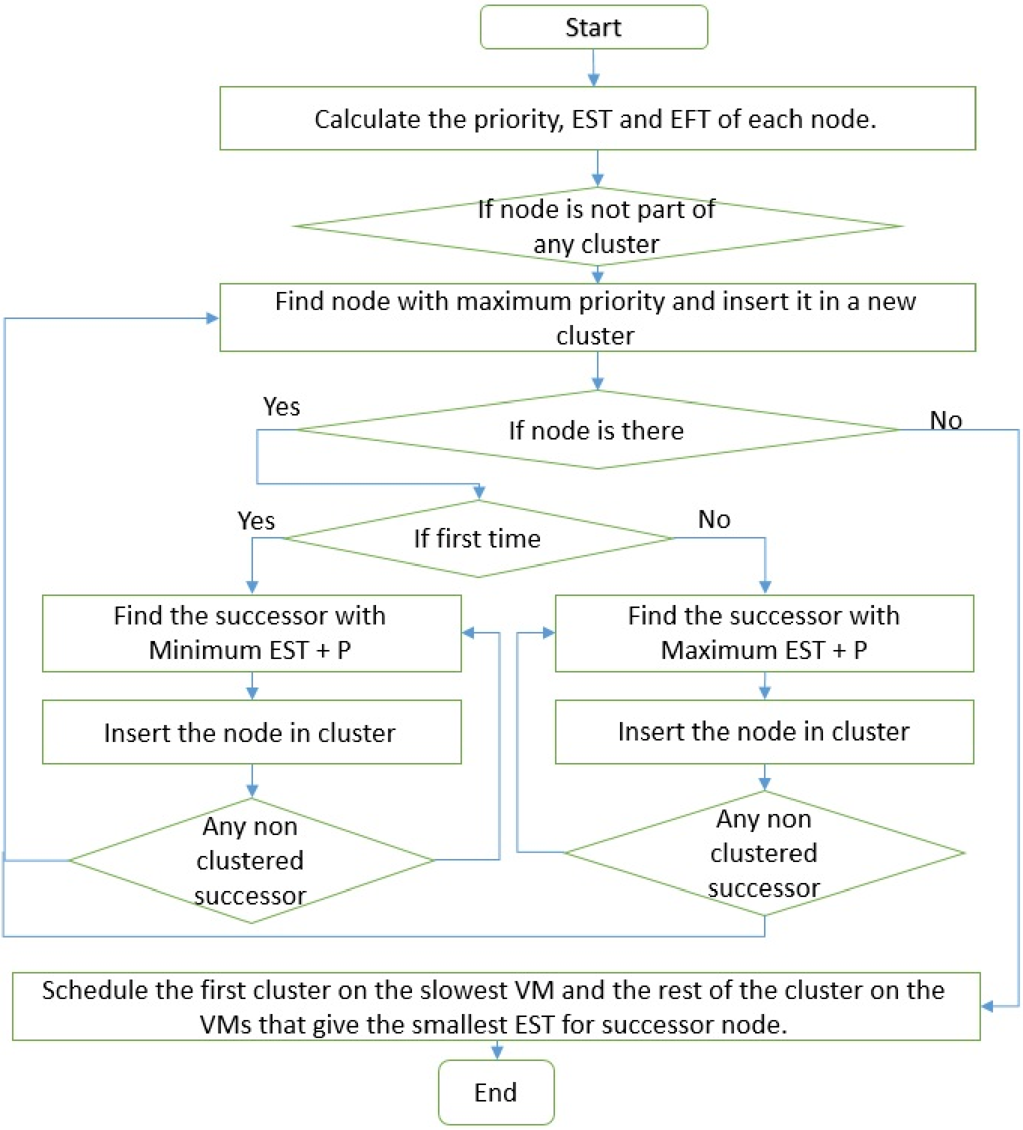
Start
Calculate the priority, EST and EFT of each node.
If node is not part of any cluster
Find node with maximum priority and insert it in a new cluster
Yes
If node is there
No
Yes
If first time
No
Find the successor with Minimum EST + P
Find the successor with Maximum EST + P
Insert the node in cluster
Insert the node in cluster
Any non clustered successor
Any non clustered successor
Schedule the first cluster on the slowest VM and the rest of the cluster on the VMs that give the smallest EST for successor node.
End

Figure 3.1: Input DAG

The Fig. 3.1 shows the flowchart for the modified algorithm that is proposed in this thesis. It is explained above, how it works.

3.1.2 Input

For implementation we are taking a dynamic acyclic graph (DAG) as input. This DAG will have n number of task that are linked to each other. Each task will have some length which will correspond to number of line of code that task has. The task will be interconnected and the edge connecting the two task will show the communication cost.

The private cloud will have m resources that have the computation power of p_{ri} and these resources are connected to each other with communication links $l_{i,j}$.

We will also have parameter deadline D, while we be sufficient enough for DAG. If the makespan of the DAG is greater than the deadline then the scheduling algorithm will try to get some resources from the public cloud. So that the DAG can be scheduled within the deadline.

The public cloud which will give lend us the resources on pay per use basis will have pun number of resources. Each resource will be associated with computation power of pu_{ri} and these resources are connected to each other with communication link $lpu_{i,j}$. In addition to that each resources in the public pool will have a price $cost_i$ associated to it. This cost will give us the price that we have to pay so as to use that particular resource.

Now the hybrid cloud set will have the communication link between each pair of resource whether that resource is part of public or private cloud. This is to ensure that each pair of resource have a direct link so that there two resources can communicate and transfer the task or cluster between them.

3.1.3 Initial Schedule

The initial scheduling will be done by PCH algorithm [2]. For implementing this algorithm we have to first calculate few parameters.

First we have to calculate the computation cost of each task that is present in the workflow on each processing resource that is available.

- **Computational Power:**

$$w_{i,r} = \frac{instructions}{p_r}$$

where $w_{i,r}$ is the computation cost of task i on resource r.

- $suc(n_i)$**:** is the immediate successor of node i.
- $pre(n_i)$**:** is the immediate predecessor of node i.

- **Priority :** $P_i = \begin{cases} w_{i,r} & if\ no\ successor \\ w_{i,r} + max_{\forall n_j \epsilon suc(n_i)}(c_{i,j} + P_j) & otherwise \end{cases}$

  $where$P$_i$ gives the priority level of node i.

- **Earliest Start Time:** $EST(n_i, r_k) = \begin{cases} Time(r_k) & if\ i = 1 \\ max\{Time(r_k), ST_i\} & otherwise \end{cases}$

  $where$ ST$_i = max_{\forall n_h \epsilon pred(n_i)}(EST(n_h, r_k) + w_{h,k} + c_{h,i})$.

  $EST(n_i, r_k)$ is the earliest start time of node i in resource r. $Time(r_k)$ is time of resource r when it will be available.

- **Estimated Finish Time:**

$$EFT(n_i, r_k) = EST(n_i, r_k) + w_{i,k}$$

where $EFT(n_i, r_k)$ is the estimated finish time of node i on resource r.

After calculating these initial parameter, we are ready to schedule the workflow using the modified PCH algorithm. For this we will find the node with highest priority and put it in a cluster. Then we will find its successor having the minimum value of the sum of EST_i and p_i and keep that node in that cluster until the last node is reached. Now from the second iteration, we will find node with highest priority in the remaining node and we will put it in another cluster. Then we will find the successor which will have maximum value of the sum of EST_i and p_i and put them in that cluster until the last node is reached. We keep on creating cluster until all node are part of one or the other cluster.

For scheduling we will now schedule the first cluster on the machine with least processing power. And we will update the available time of the resource. After that for the rest of the cluster we will find the resource that will us the minimum EST for that cluster. And this process is repeated until all the cluster are scheduled on the resources.

In the end we will get makespan for the workflow. If the makespan is less than the deadline, then the scheduling is completed and we can start actual processing. But if the deadline is missed then we have go for second phase of the algorithm. And we have to borrow the resources from the public cloud so that the makespan of the workflow can be less than the deadine.

3.1.4 Output

The output will give us the starting time and ending time of each task on the resource that it will be scheduled. The resource can be from the public or private cloud based on the scheduling algorithm. It will also give us the overall makespan for scheduling the DAG. And it will also tell us the amount that we will have to pay to avail the resources from the public cloud.

Chapter 4

Simulation and Implementation

Result and Discussion

CHAPTER 4

Simulation and Results

4.1 Simulation and Implementation

4.1.1 Experiment 1

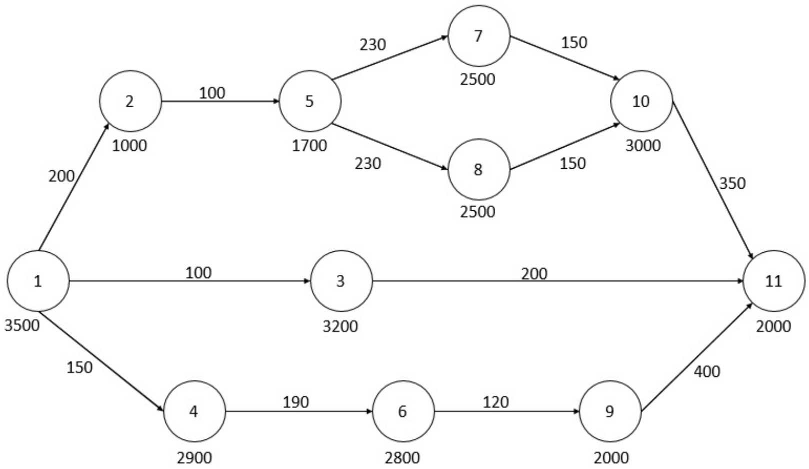

Figure 4.1: Input DAG

For our first experiment we will take DAG of 11 node. From the Fig. 4.1, we can see that each node will have some length associated with it. It is shown below the node. The length is nothing but the number of line code present in each task. There will be some intercommunication present between the node. This is shown on the edges joining the nodes. This intercommunication is nothing but the output of a node that is passed to its successor as input.

- **Length:** The length associated with each task is given below.

$$3500,\ 1000,\ 3200,\ 2900,\ 1700,\ 2800,\ 2500,\ 2500,\ 2000,\ 3000\ and\ 2000$$

- **Communication matrix:** The communication matrix is given below.

$$\begin{array}{ccccccccccc}
0 & 200 & 100 & 150 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 100 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 200 \\
0 & 0 & 0 & 0 & 0 & 190 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 230 & 230 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 120 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 150 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 150 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 400 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 350 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

First we will schedule a single DAG using the original and the modified algorithm to show the difference between them.
For this we will schedule a DAG consisting of 11 task.
The mips of the resources available in private cloud is [133 130]
The deadline for completing the tasks is 120
We have six public VMs available to us with mips [150 152 148 146 150 150].
The cost of using the public VMs per unit time is [45 50 40 35 45 45] in rupees.
After implementing the PCH Algorithm [2], the clusters are formed which are shown in Fig 4.2. Clusters formed are:

- **Cluster 1** : 1, 2, 5, 7, 10 and 11
- **Cluster 2** : 4, 6 and 8
- **Cluster 3** : 8
- **Cluster 4** : 3

But by implementing the modified PCH Algorithm, the cluster formed as showm in Fig. 4.3 are:

- **Cluster 1** : 1, 3 and 11
- **Cluster 2** : 2, 5, 7 and 10
- **Cluster 3** : 4, 6 and 9
- **Cluster 4** : 8

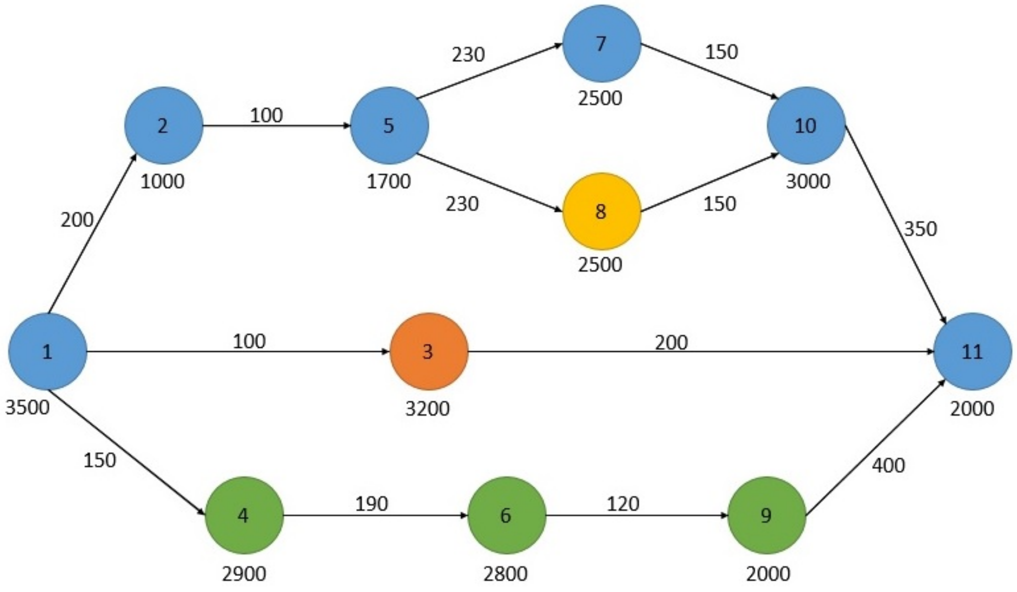

Figure 4.2: Clusters formed by PCH

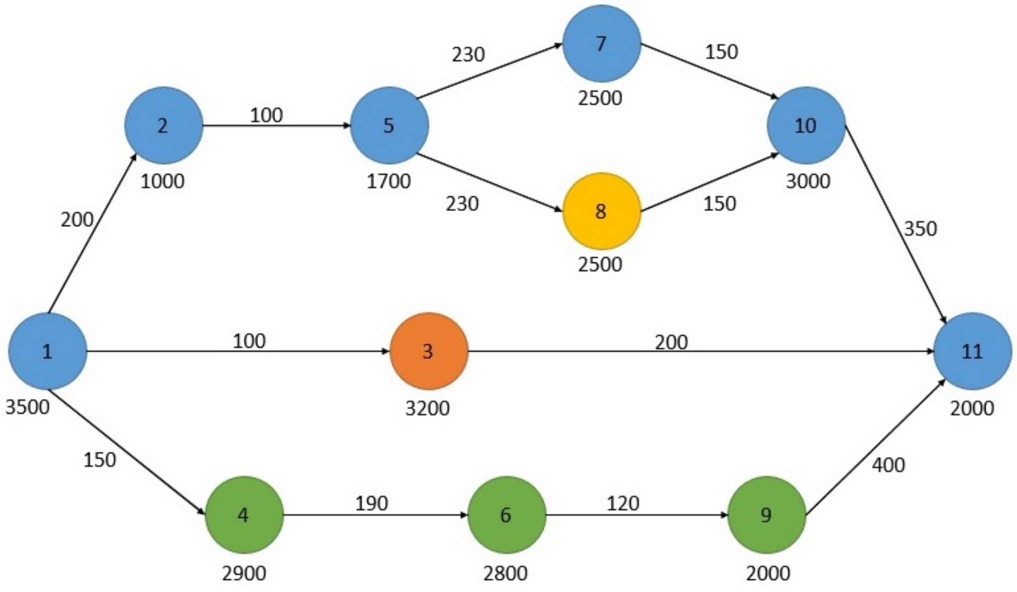

Figure 4.3: Cluster formed by modified PCH

The makespan generated by scheduling this DAG using PCH and modified PCH is shown in Fig. 4.4: Since the deadline s missed in both the cases, we have reschedule the task from private cloud to public resources.

While using PCH algorithm, node 4 and 6 are rescheduled to the public resources. They will acquire two public VMs of mips 150 each. So the cost for using these VMs is Rs 1700.

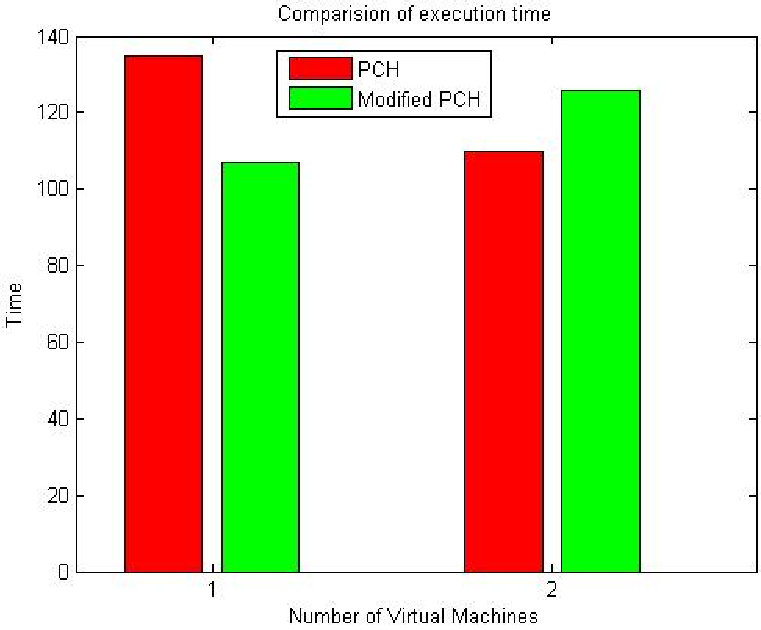

Figure 4.4: Running time of VMs in private cloud

For modified PCH algorithm, we have send just node 2 and 5 to the public resource. So it will acquire two public VM of mips 150. So the cost for using these VMs is Rs 810.

4.1.2 Experiment 2

In the second experiment, instead of taking a single DAG and scheduling it, here we have taken multiple DAG with the number of nodes starting from 10 to 110. And thus we have compared the makespan of the original and modified PCH algorithm. And final we have compared the cost incurred by original and modified workflow service algorithm in hybrid cloud.

- For this experiment we have randomly generated the DAG (with number of nodes in range of 10 to 110.)
- The number of private virtual machines are one-fifth of the number nodes in DAG.
- The MIPS for the private virtual machine is generated randomly in the range of [10,1000].
- The deadline is taken as the 70% of the makespan generated by PCH algorithm.

- The number of public virtual machines are half the number of nodes in DAG.
- The MIPS for the public virtual machines is generated randomly in the range of [10,1000].
- The cost for using the public virtual machine is in the range of [5,100].

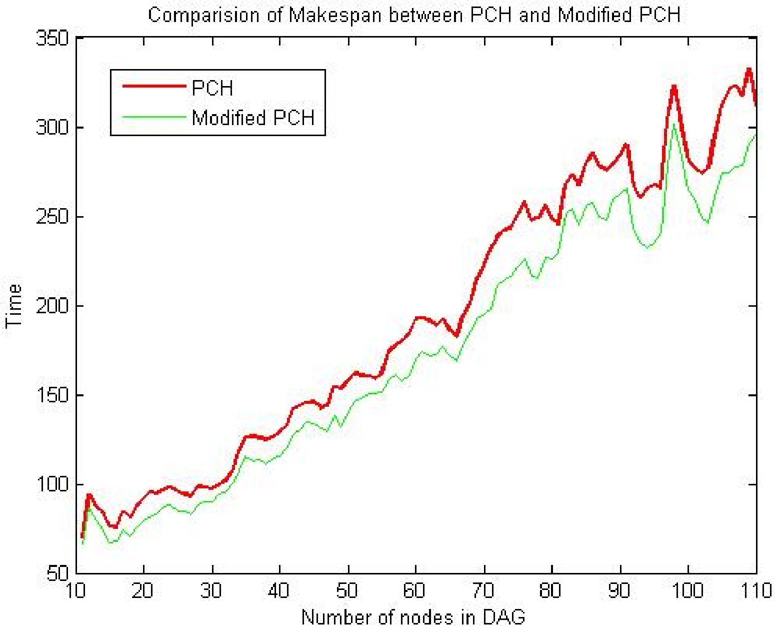

Figure 4.5: Comparison of makespan

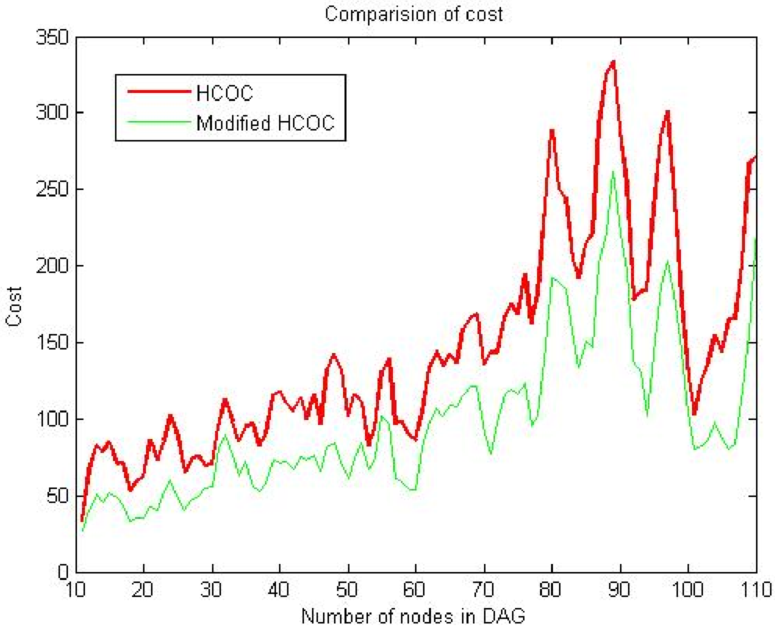

Figure 4.6: Comparison of Cost

4.2 Result and Discussion

Thus from the experiments shown in section 4.1, we can conclude that the modified PCH algorithm mostly generates makespan which is less than the original PCH algorithm. And since it is a initial algorithm that is used in Scheduling service workflow in Hybrid cloud, the modified PCH algorithm when used in this algorithm also gives the better results. Since the makespan is less in modified PCH, the number of cluster shifted to the public resources are less. Thus we have to use less public resources which in turn can be seen in the result that we are now spending less on public resources.

Chapter 5

CHAPTER 5

Future Work and Conclusion

5.1 Contribution

In this we have proposed a modified algorithm which helps us in reducing the execution time of the DAG on the private cloud. This is done by reducing the waiting time of the last node in the DAG. For reducing the waiting time, we are finding the shortest path in the DAG from starting to ending node instead of the longest path in the first round. And storing the node present in the path in a cluster. From the second round we are doing the same, i.e., find next longest path from the nodes which are not selected in the previous iteration. And again node present in the path are stored in cluster.

Now these clusters are scheduled on the resources available in private cloud. For scheduling the cluster, the first cluster is scheduled on the slowest machine, whereas the rest of the clusters are scheduled on the fastest machine available for them ASAP.

Because of these modification, the waiting time of the last node, which is present in the first cluster, is reduced. Thus we are getting a low makespan than the original algorithm. Since the makespan is lowered, we are difference between the deadline and makespan is also lowered. Because of this there are less number of task which we have to reschedule to the public resources. This less number of task are directed associated with the cost. Thus we are able to save the cost.

5.2 Future Work

Our modified algorithm only work with single core machines, so in future we are going to extend it the multiple core machines.

Bibliography

[1] ABELS, T., DHAWAN, P., and CHANDRASEKARAN, B., "An overview of xen virtualization," *Dell Power Solutions*, vol. 8, pp. 109–111, 2005.

[2] BITTENCOURT, L. F., MADEIRA, E. R., CICERRE, F., and BUZATO, L., "A path clustering heuristic for scheduling task graphs onto a grid," in *3rd International Workshop on Middleware for Grid Computing (MGC05)*, 2005.

[3] BITTENCOURT, L. F., SENNA, C. R., and MADEIRA, E. R., "Scheduling service workflows for cost optimization in hybrid clouds," in *Network and Service Management (CNSM), 2010 International Conference on*, pp. 394–397, IEEE, 2010.

[4] CHAWLA, Y. and BHONSLE, M., "A study on scheduling methods in cloud computing," *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, vol. 1, no. 3, pp. 12–17, 2012.

[5] ELZEKI, O., RESHAD, M., and ELSOUD, M., "Improved max-min algorithm in cloud computing," *International Journal of Computer Applications*, vol. 50, no. 12, pp. 22–27, 2012.

[6] GRAY, M., "Cloud computing: Demystifying iaas, paas and saas," *Retrieved July*, vol. 17, p. 2011, 2010.

[7] KOKILAVANI, T. and AMALARETHINAM, D. D. G., "Load balanced min-min algorithm for static meta-task scheduling in grid computing," *International Journal of Computer Applications*, vol. 20, no. 2, pp. 43–49, 2011.

[8] LIU, G., LI, J., and XU, J., "An improved min-min algorithm in cloud computing," in *Proceedings of the 2012 International Conference of Modern Computer Science and Applications*, pp. 47–52, Springer, 2013.

[9] MELL, P. and GRANCE, T., "The nist definition of cloud computing," 2011.

[10] PARSA, S. and ENTEZARI-MALEKI, R., "Rasa: A new task scheduling algorithm in grid environment," *World Applied sciences journal*, vol. 7, pp. 152–160, 2009.

[11] RAHMAN, M., LI, X., and PALIT, H., "Hybrid heuristic for scheduling data analytics workflow applications in hybrid cloud environment," in *Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), 2011 IEEE International Symposium on*, pp. 966–974, IEEE, 2011.

[12] VAN DEN BOSSCHE, R., VANMECHELEN, K., and BROECKHOVE, J., "Cost-optimal scheduling in hybrid iaas clouds for deadline constrained workloads," in *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, pp. 228–235, IEEE, 2010.

[13] VAN DEN BOSSCHE, R., VANMECHELEN, K., and BROECKHOVE, J., "Cost-efficient scheduling heuristics for deadline constrained workloads on hybrid clouds," in *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*, pp. 320–327, IEEE, 2011.