ISSN:2320-0790

COMPUSOFT
An International Journal of Advanced Computer Technology

Enhancement of Fault Tolerance in Cloud Computing

Pushpanjali Gupta, Rasmi Ranjan Patra

CSA, CPGS
OUAT, Bhubaneswar
Odisha, India

Abstract: In recent years researchers are trying to work out scientific applications in cloud so that it decreases the infrastructure cost and increases the span of team and finally innovative ideas towards applications is increased. But the cloud is still not as much reliable, controllable as grid. So in the evolving Cloud computing environment there is a great need of fault tolerance mechanism for the system to work effectively even in the presence of failure. Moreover Big Organizations are also opting for using Hybrid Cloud instead of private Cloud. Thus, in this paper we propose an approach of using a new framework in Cloud so as to use Cloud for scientific applications as well makes the public Cloud trustworthy platform. There is a progressive approach introduced to provide an effective way to achieve high fault tolerance in Clouds by enabling a new workflow planning method to balance performance, reliability and cost for critical scientific applications and focus mainly on use of distributed resources for workflow execution mainly in serial and concurrent manner.

Keywords— Cloud Computing, Fault Tolerance, Workflows, Instances in Cloud.

I. INTRODUCTION

Since the last decade there is a huge demand for large scale data storage, high speed computation in an environment which needs to be very reliable, flexible and low cost. Moreover, it's not an easy task to have a private Cloud for performing all of your operations. Recently it is found that researchers are interested in using cloud for performing scientific applications and even the big organizations are on the verse of switching over to hybrid cloud. The large scale availability of virtual instances in public Cloud Computing Infrastructure provides an effective environment for the performance of different tasks online. This makes it easy for different team members of a research team to access the information global wide, more members can join in and a big bright idea can be generated. But in spite of having many benefits in Cloud, there are some problems associated with Cloud which can be risky and may lead to big failures if not tackled in time. So there should be some reliable mechanisms to handle the failures and provide a worthwhile environment to the user which fits to their desirable interface.

Consider an example where you are accessing MS-WORD online using a hybrid instance. You have to write an application and send it urgently. You have written it and just before saving and sending, the instance you were using is terminated by system. Such cases are intolerable. To recover from this, you are provided with a back up mechanism which is again not cost efficient. This is beneficial for Cloud Providers but not user. So this makes cloud unreliable.

The main contributions to Cloud Computing Infrastructure through this paper are as follows:

- Enable a new workflow planning method to balance performance, reliability and cost considerations for critical applications and focus mainly on use of distributed resources for workflow execution.
- Propose a robust, adaptable method to run large-scale computational applications using reserved instance to reduce cost, increase scalability, reliability and tolerate the failure without increasing the completion time.
- Propose a use of replication & resubmission method for distributed Cloud storage systems, which is highly-efficient, fault-tolerant and suitable for global wide replication.

The rest of the paper is organized as follows:
Section II discusses the related work in this field. Section III describes proposed model. Section IV brings the conclusion and future scope of the paper.

II. RELATED WORK

In cloud computing fault tolerance is much important issue to improve reliability, performance of computing resources. The failures (hardware, network, and instance) in cloud decrease its performance thus decreasing the QoS. The public cloud used for large computation need to provide the much needed QoS so that it can be used by time sensitive critical applications.

The papers [1] [2] [3] [4] [5] discuss some aspects of fault tolerance, cloud computing. The authors have given some concepts and mechanisms used to implement some mechanisms in cloud computing so as to enhance fault tolerance. The authors of the remaining papers discuss various ideas for tolerating fault in cloud computing and improving the use of cloud computing in scientific applications.

In [6] the authors provide an enabling technology for executing deadline-driven, fault-tolerant workflows. The integrated cyber infrastructure from the LEAD and VGrADS system components provides a strong foundation for next-generation dynamic and adaptive environments for scientific workflows. Programming grid and cloud systems for e-Science workflows and managing QOS in these environments is challenging. VGrADS' virtual grid abstraction simplifies these tasks, unifying workflow execution over batch queue systems (with and without advanced reservations) and cloud computing sites (including Amazon EC2 and Eucalyptus). VgES provides a single interface to query and manage execution across distributed sites based on QOS properties of the resources. This approach uses similar support for each workflow but also manages multiple workflows. In this paper, the authors have used replication based fault-tolerance techniques based on resource (slot) reliability and application performance models. The Virtual Grid Execution System provides a uniform interface for provisioning, querying and controlling the resources. It allows coordinating resources across the components in a workflow set, and managing fault tolerance for the overall workflow set. The qualitative resource abstraction in VGrADS provides the ability to aggregate resources from different sources and under different policies .The vgES acquires resources and ensures their availability during time intervals expressed by the user. Internally, vgES maintains a database populated with static information about the entire collection of HPC and cloud resources available, and translates high level requests for resource collections into the most appropriate set of real resources at any given time. When vgES receives a request to construct a virtual grid, it performs a database search to find the best collection of resources to provision, and returns a virtual grid abstract data structure. VgES provides the view of resources in the form of a Gantt chart that describes how the resources will be provisioned over time.

In paper [7], the authors analyzed the pricing schemes of Amazon Elastic Compute Cloud (EC2) and found the disturbance effect that the price of the spot instances can be heavily affected due to the large number of spot instances required. So they proposed a dynamic approach which schedules and runs large-scale computational applications on a dynamic pool of cloud computational instances. There is the use hybrid instance, including both on-demand instances for high priority tasks and backup and spot instances for normal computational tasks so as to further reduce the cost without significantly increasing the completion time. This proposed method takes the dynamic pricing of cloud instances into consideration, and it reduces the cost and tolerates the failures for running large-scale applications in public clouds. The approach is robust and adaptable for large-scale computational applications.

Most of the authors [6][7][8] have suggested the use of replication and resubmission mechanism in order to enhance fault tolerance .But replicating data uses lot of resources, and resubmission increases computation time. So they have proposed a way to replicate and resubmit a task in such a way that it solves the problems. But the solution provided is not applicable always as it requires the parameters to be given by user.

In order to complete a task within the time a soft deadline (expected execution time) [6] is provided by the mechanisms, but it is not always reliable and not easy to calculate the time as it requires historic data collection which
is not easy to collect as cloud service provider do not give the information of their infrastructure.

Some authors [7] have considered the use of different types of reservation of instances for performing the execution. But the problem lies in the availability of instances as per requirement and at the time of requirement. Some authors [8] have suggested the use of reservation of virtual instances in teragrid instead of cloud in order to complete the execution within deadline, but resources in teragrid require being reserved before 48 hours of execution.

So if we reserve the instances in cloud for performing computation, it won't require us to reserve the instance much before the computation. We can execute multiple workflows at a time using 2 or more instances at a time .In this way the computation will be completed within deadline, and increase performance. For the scheduling we can assign priority to workflows, and maintain queue which can be shuffled as per requirement. If we provide a reliable atmosphere in cloud then scientific applications can be easily carried out in cloud.

Here is an approach which considers the use of replication & resubmission for scheduling multiple workflows, by assigning priority using FCFS & executing the workflows using MLQ.

III. PROPOSED MODEL (FRAMEWORK DESIGN)

The different techniques of achieving the fault tolerance make use of replication and resubmission methods.

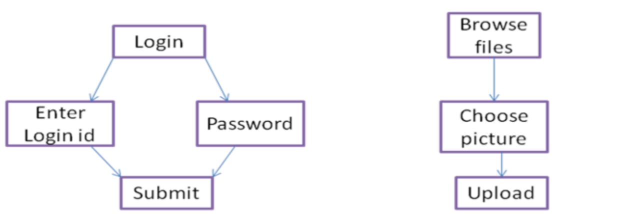
However replication and resubmission if done arbitrarily decreases the reliability & performance of Cloud. So there is a need of applying fault tolerance techniques in a customized manner.

To enhance Cloud infrastructure we use a framework consisting of reserved instance of cloud. The reserved instance is much easier to use. All you need is to give the details of the type of instance you need and the term of its usage. You will be allotted with the instance which is according to your requirement completely or approximately. You can use the instance whenever you need it, and when you don't use it is being used by other users. So, there is no wastage of resources.

There are different sets of workflows which may be dependent on one another. So the workflows are first scheduled using FCFS (First Come First Serve) and MLQ (Multi Level Queue) techniques.

A: Dependencies between Workflows

In this case we consider the dependencies between workflows.



(a) Login Workflow (b)Upload Workflow

Fig-1

In Fig-1, let's take an example of a job of setting a new picture as your profile picture in Facebook. It needs two sets of workflows: (a) login to your account, (b) upload picture. Here (b) depends on (a). So even if the workflows do not arrive in order, we assign priority to the workflows basing on dependencies and execute them serially.

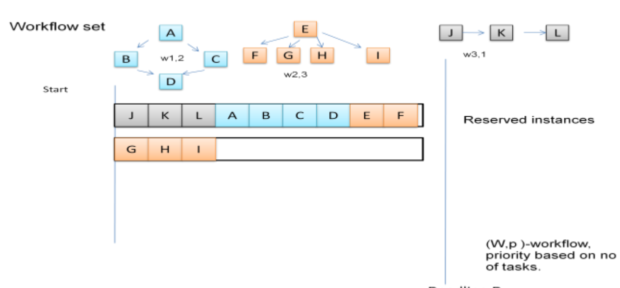


Fig-2

Case-1: Handling serial processing of workflow (Fig-2)

When there are a number of dependent workflows arriving at a time, we can handle the processing of workflows by assigning priority according to their dependency as in the form of fig-2.The instance is reserved as per our requirement and used for performing computation.

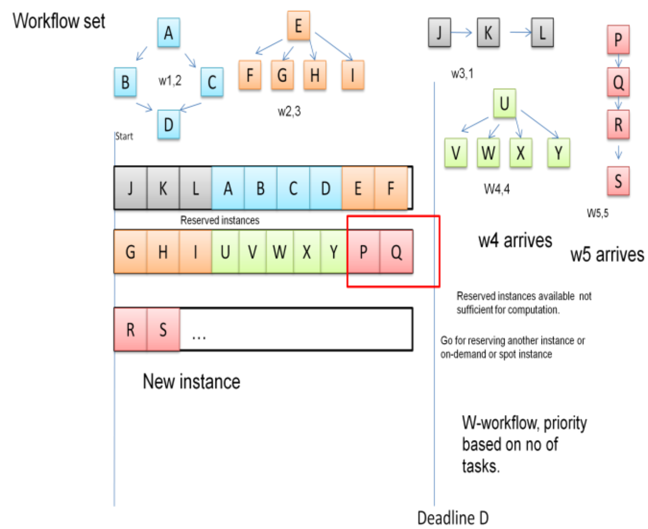


Fig-3

Case-2: If reserve instance not available, go for on-demand/spot instance/hybrid instance (in serial processing) (Fig-3)

Although we reserve instance as per our need, but if we find a lot of workflows are arriving for performing computation and the instance reserved is not sufficient, we can immediately go for the another instance as in fig-3.

B: No dependencies between workflows.

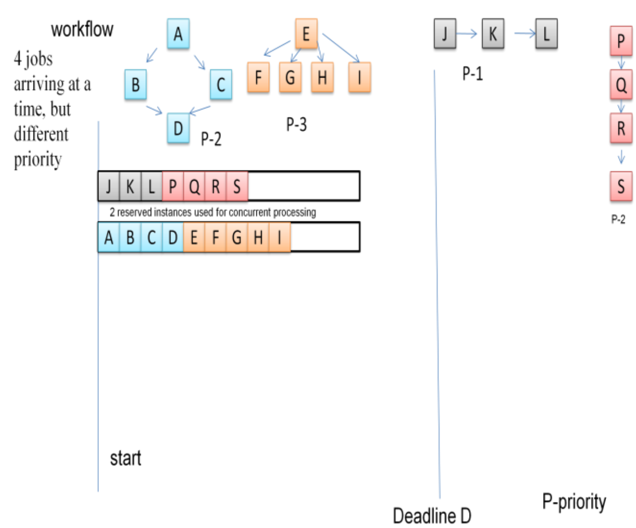


Fig-4

Case-1: Concurrent processing of workflow (Fig-4)

If more than one independent workflow arrives simultaneously then they can be executed concurrently using MLQ as in fig-4.

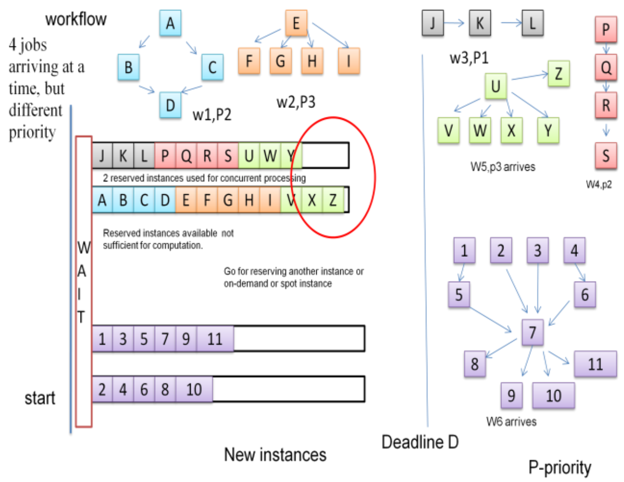


Fig-5

Case-2: If reserve instance not available, go for on-demand/spot instance/hybrid instance (in concurrent processing) (Fig-5)

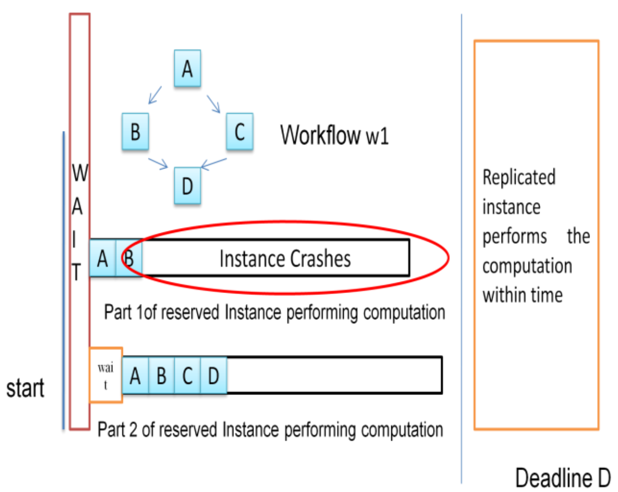


Fig-6

Case-3 Handling Failure of instances (Fig-6)

For our computation we reserve instance which is physically not present in a single place. So while executing workflows we use a part of it, one of the other parts is used to replicate the completed task. So if we are executing our task in part1 of instance, then we are replicating the executed task in part2. If in the middle part1 instance crashes, we use part2 to perform our computation. Here the benefit is that since part2 contains executed tasks, we don't need to execute the complete workflow, we execute from the last checkpoint, which saves the time. After completion of computation the instances are set free.

IV. CONCLUSION & FUTURE WORK

Fault being a major issue in cloud computing needs to be resolved in a convenient way. So there is greater need of Fault Tolerance mechanisms to have reliable, robust and better cloud. We have suggested the reservation of instances instead of teragrid for executing multiple critical workflows serially as well as concurrently using MLQ to reduce cost and time of completion. The workflows are given priority basing on the expected time of completion as well as arrival time. The executed workflows are replicated and used for execution using checkpoint mechanism thus decreasing resource consumption and time of completion leading to complete workflow execution within the deadline. In this way using this framework multiple sensitive scientific workflows can be executed in cloud thus making public cloud reliable. Big organization opting for hybrid cloud can use public cloud for performing different computation and use their private cloud for data & information storage thus proving large area in less effective cost. For storage of data and information a different set of reserved instances are used. In this case the instances are replicated using forwarding traffic techniques. Here if two instances need to replicate their data at a time, we use priority basing on sensitivity of queries. Thus we find this framework suitable for use in pubic cloud. So now cloud can be made suitable for working of scientific application & use in hybrid cloud. Our future work will include improving the framework and performing the simulation and evaluation of the model in order to prove the efficiency of the protocol.

REFERENCES

[1] Daniel Nurmi, Rich Wolski, Chris Grzegorczyk Graziano Obertelli, Sunil Soman, Lamia Youseff, Dmitrii Zagorodnov:" The Eucalyptus Open-Source Cloud-Computing System", 978-0-7695-3622-4/09 $25.00 © 2009 IEEE .DOI 10.1109/Ccgrid.2009.93.

[2] Ravi Jhawar, Vincenzo Piuri, and Marco Santambrogio: "Fault Tolerance Management in Cloud Computing: A System-Level Perspective" Digital Object Identifier 10.1109/Jsyst.2012.2221934, IEEE Systems Journal, Vol. 7, No. 2, June 2013

[3] Daniel Warneke,And Odej Kao:" Exploiting Dynamic Resource Allocation For Efficient Parallel Data Processing In The Cloud" IEEE Transactions On Parallel And Distributed Systems, Vol. 22, No. 6, June 2011, Digital Object Identifier No. 10.1109/Tpds.2011.65.

[4] Expert Group Report Public Version 1.0 Rapporteur for This Report: Lutz Schubert [Ustutt-Hlrs] Editors: Keith Jeff Ery [Ercim], Burkhard Neidecker-Lutz [Sap Research]:"The

Future of Cloud Computing Opportunities for European Cloud Computing Beyond 2010"

[5] Amazon Elastic Compute Cloud (Amazon Ec2). Http://Aws.Amazon.Com/Ec2/.

[6]Lavanya Ramakrishnan,Daniel Nurmi ,Anirban Mandal, Charles Koelbel,Dennis Gannon,T. Mark Huang,Yang-Suk Kee,Graziano Obertelli,Kiran Thyagaraja,Rich Wolski,Asim Yarkhan,Dmitrii Zagorodnov." VGRADS: Enabling E-Science Workflows on Grids and Clouds with Fault Tolerance".

[7] Sifei Lu, Xiaorong Li, Long Wang, Henry Kasim, Henry Palit, Terence Hung, Erika Fille Tupas Legara, Gary Lee:" A Dynamic Hybrid Resource Provisioning Approach For Running Large-Scale Computational Applications On Cloud Spot And On-Demand Instances", 2013 19th IEEE International Conference On Parallel And Distributed Systems 1521-9097/13 $31.00 © 2013 IEEE DOI 10.1109/.116 658 657.

[8] Yanzhen Qu, Naixue Xiong; "RFH: A Resilient, Fault-Tolerant and High-Efficient Replication Algorithm for Distributed Cloud Storage", 2012 41st International Conference on Parallel Processing, 0190-3918/12 $26.00 © 2012 IEEE DOI 10.1109/Icpp.2012.3

AUTHORS PROFILE

Pushpanjali Gupta received Master In Computer Application with 1st Class With distinction from O.U.A.T, Odisha, India in 2014.She is working in the area of Cloud Computing for last 2 years.

Rasmi Ranjan Patra received Master In Computer Application With 1st Class With distinction from O.U.A.T, Odisha, India in 2001, M.Tech in Computer Science and Technology from C.E.T, Bhubaneswar, India in 2010 and Ph.D Degree in 2013 from Utkal University, India. He is working as Assistant professor in Department of Computer Science and Application under Orissa University of Agriculture and Technology (O.U.A.T).He has Published many papers at national /international Journals and Conferences in the areas of Sensor Network, Soft Computing, Cloud computing and Big data. Mr. Patra has authorized one book in Computer Science area.