International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 1 Issue 8, October - 2012

Checkpoint-Based Fault Identification in Cloud Computing Tasks

G. Malathy[1], Dr.Rm. Somasundaram[2], Dr.K.Duraiswamy3
[1]*Assistant Professor/CSE, KSR Institute for Engineering and Technology, Thiruchengode, Tamil Nadu, India.*
[2]*Professor/CSE, SNS College of Engineering, Coimbatore, Tamil Nadu, India.*
[3]*Dean/CSE,K.S.Rangasamy College of Technology, Thiruchengode, Tamil Nadu, India.*

Abstract

Cloud computing is a computing paradigm in which the various tasks are assigned to a combination of connections, software and services that can be accessed over the network. The computing resources and services can be efficiently delivered and utilized, making the vision of computing utility realizable. In various applications, execution of services with more number of tasks has to perform with minimum intertask communication. The applications are more likely to exhibit different patterns and levels, and the distributed resources organize into various topologies for information and query dissemination. To ensure effective performance, fault tolerance and identification should be taken into consideration. The method to make the most of a diverse set of tasks with fault identification from the available resources in cloud efficiently is proposed in this paper. For the fault identification in the scheduling of tasks, checkpoints are inserted to identify the occurrence of fault with less computation and less time is required. The checkpoint fault identification method is evaluated in CloudSim, a toolkit for modeling and simulating cloud computing environments and the evaluation improves the performance of the system.

1. Introduction

Large scale computing environments such as clouds propose to offer access to a vast collection of heterogeneous resources. A cloud is a parallel and distributed structure consisting of a group of interconnected and virtualized computers that are dynamically accessible as one or more unified computing resources [3]. The shared resources, software, and information provided through the cloud to computers and other devices are normally offered as a metered service over the Internet. A user in the cloud system need not know about the place and other details of the computing infrastructure. Thus the user can comfortably concentrate on their tasks rather than utilizing time and knowledge on knowing the resources to manage the tasks. Internet is one basis of the cloud computing, therefore an unavoidable issue with Internet is that the network bottlenecks often occur when there is a large amount of data to be transferred. In this case, the complexity of resource management stick on to users and the users have normally limited management tools and authentication to deal with such issues [2]. Clouds are classified into three categories named public clouds, private clouds, and hybrid clouds [18]. Public clouds are publicly available remote interface for masses creating and managing resources, private clouds gives the local users a flexible and responsive private infrastructure to manage the workloads at their own cloud sites and the hybrid cloud enables the supplementing local system with the computing capacity from an external public cloud. Public cloud services like Google's App Engine are open to all anywhere round the clock. An example for private cloud is the usage of GFS, MapReduce, and BigTable by Google inside the enterprise. The following are the features of cloud computing.

- High flexibility
- High security
- Easy to maintain
- Location independent
- Reduction in capital expenditure on hardware and software

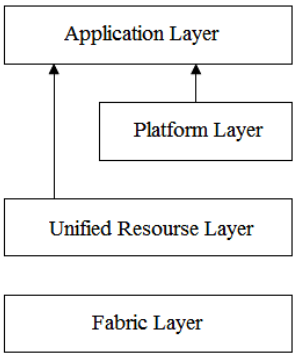

Fig 1 Architecture of Cloud Computing

Fig1 shows the four layer architecture of cloud computing and clouds are viewed as a large pool of computing and storage resources that are accessed through standard protocols with an abstract interface [6]. Computing resources, storage resources, and network resources are hardware level resources which are present in the fabric layer. Resources are virtualized to upper layer and end users as integrated resources are done in the unified resource layer. To develop and deploy platform on top of unified resources, a collection of specialized tools, middleware and services are added with the platform layer. The various applications that would be executed in the cloud environment are included in the application layer. The delivery mechanism in cloud computing is considered as services and is categorized in three different levels named; software service, platform service and infrastructure service [1]. The Software as a Service (SaaS) is a software delivery model in that the applications are accessed by simple interface like web browser over Internet. Examples of SaaS-based services are web Mail, Google Docs, Facebook, etc. The Platform as a Service (PaaS) gives a high-level integrated environment to build, test, deploy and host customer-created or acquired applications. Examples of PaaS-based service are Google App Engine, Engine Yard, Heroku, etc. Infrastructure as a Service (IaaS) ensures processing, storage, networks, and other fundamental computing resources to the users. Examples of IaaS-based services are Amazon EC2, IBM's Blue Cloud, Eucalyptus, Rackspace Cloud, etc.

Clustering is a primary and cost-effective platform for executing parallel applications that computes large amount of data with the nodes of a cluster through the interconnected network. Clustering is traditionally been used in many data mining applications to group together the statistically similar data elements. The algorithms used for clustering must not assume the existence of a standard distribution of certain parameters [23]. The performance of the cluster for scientific applications by the use of fully utilizing computing devices with idle or underutilized resources requires the scheduling and load balancing techniques in an effective way. Some of the applications of cluster based services include 3D perspective rendering technique, molecular dynamics simulation, etc. Moreover, the performance between the effective speed of processor and the various network resources continues to grow faster, which raises the need for increasing the utilization of networks on clusters using various techniques [21].

The rest of the paper is organized as follows. Section 2 reviews about the related literature and section 3 focuses on the detailed description of the proposed fault identification in tasks using checkpoint based cloud computing approach. Section 4 details the experimental setup and analysis of the proposed approach. Finally, conclusion is given in section 5.

2. Related Work

In this section, we review the prior work on improving the design strategy in cloud computing. Qian et al [14] proposed the use of cloud resources for a class of adaptive applications, where application-specific flexibility in computation is required with fixed time-limit and resource budget. The adaptive applications are maximized with Quality of Service (QoS) very precisely and by dynamically varying the adaptive parameters the value of application-specific benefit function is obtained. A multi-input multi-output feedback control model based dynamic resource provisioning algorithm is developed that adopts reinforcement learning to adjust adaptive parameters to guarantee the optimal application benefits within the time constraints.

Sandeep Tayal [16] proposed a task scheduling optimization for the cloud computing system based on Fuzzy-GA which makes a scheduling decision by evaluating the entire group of task in a job queue. The fuzzy sets were modeled to imprecise scheduling parameters and also to represent satisfaction grades of each objective. GA with various components are developed on the technique for task level scheduling in Hadoop MapReduce. To obtain better balanced load execution time of tasks assigned to processors are predicted using scheduler and making an optimal decision over the entire group of tasks.

Seokho et al [17] proposed a service-level agreement while making reservations for cloud services. The presented multi-issue negotiation mechanism supports both price and time-slot negotiations between cloud agents and tradeoff between price and time-slot utilities. The agents make multiple proposals in a negotiation round to

generate aggregated utility with variations in individual price and time-slot utilities.

Soumya et al [19] proposed an initial heuristic algorithm to apply modified ant colony optimization approach for the diversified service allocation and scheduling mechanism in cloud computing framework. The proposed optimization technique is used to minimize the scheduling throughput to service all the diversified requests according to the different resource allocator available under cloud computing environment.

Lei et al [13] proposed a public cloud usage model for small-to-medium scale scientific communities to utilize elastic resources on a public cloud site. Also, implemented an innovative system named DawningCloud, at the core of which a lightweight service management layers running on top of a common management service framework. The system has been evaluated and found that DawningCloud saves the resource consumption to a maximum amount.

Rajkumar et al [15] presented the vision, challenges, and architectural elements of service level agreement-oriented resource management. The architecture supports integration of market-based provisioning policies and virtualization technologies for flexible allocation of resources to applications. The performance results obtained from the working prototype system shows the feasibility and effectiveness of service level agreement-based resource provisioning in cloud systems.

Ganesh et al [7] investigated the use of a divisible load paradigm to design efficient strategies to minimize the overall processing time for performing large-scale polynomial product computations in compute cloud environments. For post-processing a compute cloud system with the resource allocator distributing the entire load to a set of virtual CPU instances is processed. Finally through simulation the performance of the strategy is quantified.

Ghalem et al [8] proposed an algorithm to improve the quality of service of real world economy and to extend and enrich the simulator CloudSim by auction algorithms inherited from GridSim simulator. The work satisfies the users by reducing the cost of processing cloudlets and improved implementation on GridSim to reduce the time auction and to assure a rapid and effective acquisition of computing resources.

Hong-Ha et al [9] considered the problem of scheduling lightpaths and computing resources for sliding grid demands in Wave Division Multiplexing (WDM).

On each demand a joint scheduling algorithm decides the start time, reserve an amount of computing resources and provide a primary lightpath. For obtaining an Integer Linear Programming (ILP) formulation is developed and to achieve scalability heuristic algorithms based on joint resource scheduling is used.

Khawar et al [12] proposed a pilot job concept that has intelligent data reuse and job execution strategies to minimize the scheduling, queuing, execution and data access latencies. By this approach, significant improvements in the overall turnaround time of a workflow can be achieved. This is evaluated using CMS Tier0 data processing workflow, and then in a controlled environment.

Jianhua et al [10] proposed a resource scheduling strategy based on genetic algorithm to produce best load balancing and reduces dynamic migration. To measure the overall load balancing effect of the algorithm an average load distance method is introduced. The method solves the problems of load imbalance and high migration cost after system virtual machine being scheduled.

Thomas et al [20] introduced a model for estimating the business impact of operational risk resulting from changes. The model takes into account the network of dependencies between process and services, probabilistic change-related downtime, uncertainty in business process demand, and various infrastructural characteristics. The model is evaluated using simulations based on the industrial data.

Cenk Erdil [4] described a general purpose peer-to-peer simulation environment that allows a wide variety of parameters, protocols, strategies and policies to be varied and studied. For proof utilization of the simulation environment is presented in a large-scale distributed system problem that includes a core model and related mechanisms.

Xiao et al [21] proposed a communication-aware load-balancing technique that is capable of improving the performance of communication-intensive applications by increasing the effective utilization of networks in cluster environments. Also a behavior model for parallel applications is added with the load-balancing technique with large requirements of network, CPU, memory and disk I/O resources.

Young et al [22] investigated the problem of scheduling workflow applications on grids and presents a novel scheduling algorithm for the minimization of application completion time. The performance of grid resources changes dynamically and the accurate estimation of performance is

difficult, and the proposed rescheduling method deal the unforeseen performance fluctuations effectively.

Ke et al [11] proposed a compromised-time-cost scheduling algorithm which considers the characteristics of cloud computing to accommodate instance-intensive cost-constrained workflows by compromising execution time and cost with user input enabled on the fly. Simulations show that the algorithm can achieve a lower cost than others while meeting the user-designated deadline or reduce the mean execution time than others within the user-designated execution cost.

Dharma et al [5] proposed a data replication algorithm that is not only a provable theoretical performance guarantee, but also can be implemented in distributed manner. This is based on a polynomial time centralized replication algorithm that reduces the total data file access delay by at least half of that reduced by the optimal replication solution.

3. Checkpoint-based Fault Identification

The runtime variations in cloud system may significantly affect the tasks execution time. For a huge time critical or time consuming tasks, parameters like delay and losses are not acceptable and an efficient fault tolerance mechanism should be considered. In a distributed system environment providing fault tolerance with optimizing resource utilization and tasks execution time is a challenging job. To accomplish this checkpoints are inserted at locations with regular interval through which the fault can be analyzed with less time.

Fig2 shows the proposed structure of checkpoint-based fault identification approach used in the cloud computing system. The system consists of geographically dispersed cloud clients over the entire system and a service unit.

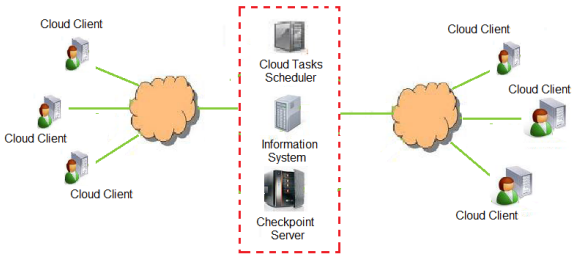

Fig 2 Architecture of Checkpoint-based Fault Identification approach

The service unit consists of cloud tasks scheduler, information service system, and checkpoint server. The cloud tasks scheduler is responsible for tasks resource matchmaking. The information service system collects the tasks and resource status information required for the cloud tasks scheduler. The checkpoint server is used to place checkpoint data in the tasks at appropriate locations. The cloud tasks scheduler invokes the matchmaking procedure within the predefined scheduling interval. The information service system collects the changes in resource status with delay to reflect modification in propagation time occurring in exact deployments.

The checkpoint approach even though saves the computation time for the faulty tasks, some factors like runtime overhead, latency, and recovery delay will be increased. The runtime overhead is the time delay obtained from the interruption of tasks execution to perform the checkpoint operation. The latency is the time interval between the checkpoint generation and its availability on the checkpoint server. The recovery delay is the time to download a failed tasks checkpoint from the checkpoint server to the cloud resources where the tasks are rescheduled to run. However, this paper concentrates on the checkpoint approach for obtaining the faulty tasks in the cloud environment.

4. Simulation Results

This section describes the implementation method for the proposed reservation cluster-based cloud computing approach. For the simulation hundred cloudlets are considered. Implementation is carried out on Cloudsim, because the rich set of simulation facilities in Cloudsim empowers us to implement and evaluate the checkpoint approach for fault identification of tasks in the heterogeneous distributed computing environments. Fig 3 shows the implementation screenshot of tasks executed with the resources in the cloud system using CloudSim.

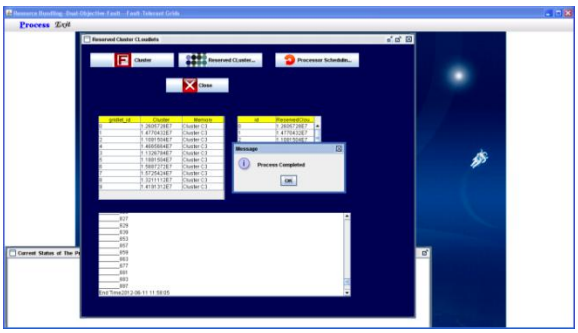

Fig 3 Checkpoint-based Fault Identification implementation screenshot using CloudSim

For the simulation, the task is retrieved by the cloud task scheduler and based on the information available the task is executed into sub process by placing checkpoints with the help of checkpoint server. So the sub processes are executed in parallel based on the available resources and if some fault occurs it is reported to the cloud task scheduler. So

the task scheduler reallocates the sub process to another cloud client by verifying the availability of resources in the information service system. If there is no checkpoint approach, the entire task will be executed by a node in the system and analyzed for fault. This will take more time and if there is some fault, then the total time is wasted. Instead of this if the task is scheduled into sub process by placing checkpoints in the task, and the sub process is given to various cloud nodes based on the information available in the information service system for the availability of resources in the cloud system. In this way processing can be efficiently done in the cloud system. The performance metrics used for the analysis is the resource usage and the execution time. The resource usage is defined as the average amount of usage of resources in the cloud system. The execution time is defined as the average time taken to complete the task. Figure 4 shows the runtime screen shot of task in the cloud environment using the checkpoint-based fault identification approach.

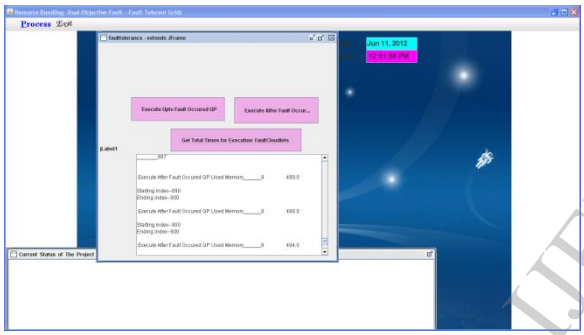

Fig 4 Analysis of execution time

5. Conclusion

This paper addresses checkpoint-based fault identification in cloud computing system. This fault tolerant approach reduces the time for predicting a fault because the checkpoint approach places checkpoint at different locations and verifies the result instead of verifying the result after executing the entire task. Analysis has been done using the CloudSim simulator, in which the maximum cloudlets used in the cloud system is 100 and checkpoint is placed by the checkpoint server on the task, and the sub process is executed by the resources in the cloud system. Thus the occurrence of faults is easily identified, without waiting for the entire task to execute and rescheduling in carried out. The rescheduling of the sub process is to done as future work with an optimized system so that the entire system performance can still be improved.

References

[1] M. Armbrust, A. Fox, and R. Griffith, "Above the Clouds: A Berkeley view of Cloud Computing Technical Report UCB/EECS-2009-28", EECS Department, University of California, Berkeley, Feb. 2009.

[2] 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, Issue 4, pp. 50-58, 2010.

[3] C. S. Buyya, R. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility", Future Generation Computer Systems, Vol. 25, No. 6, pp. 599-616, June 2009.

[4] Cenk Erdil D., "Simulating peer-to-peer cloud resource scheduling", Peer-to-Peer Network Applications, Springer, 12 November 2011, DOI 10.1007/s12083-011-0112-8.

[5] Dharma Teja Nukarapu, Bin Tang, Liqiang Wang, and Shiyong Lu, "Data Replication in Data Intensive Scientific Applications with Performance Guarantee", IEEE Transactions on Parallel and Distributed Systems, Vol. 22, No. 8, pp. 1299- 1306, August 2011.

[6] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud Computing and Grid Computing 360 degree compared", Proceedings of Grid Computing Environments Workshop (GCE'08), pages 1-10, 2008.

[7] Ganesh Neelakanta Iyer, Bharadwaj Veeravalli, and Sakthi Ganesh Krishnamoorthy, "On Handling Large-Scale Polynomial Multiplications in Compute Cloud Environments using Divisible Load Paradigm", IEEE Transactions on Aerospace and Electronic Systems, Vol. 48, No. 1, pp. 820-831, January 2012.

[8] Ghalem Belalem, Samah Bouamama, and Larbi Sekhri, "An Effective Economic Management of Resources in Cloud Computing", Journal of Computers, Vol. 6, No. 3, pp. 404-411, Academy Publisher 2011.

[9] Hong-Ha Nguyen, Mohan Gurusamy, and Luying Zhou, "Scheduling Network and Computing Resources for Sliding Demands in Optical Grids", Journal of Lightwave Technology, Vol. 27, No. 12, pp. 1827-1836, June 15 2009.

[10] Jianhua Gu, Jinhua Hu, Tianhai Zhao, and Guofei Sun, "A New Resource Scheduling Strategy Based on Genetic Algorithm in Cloud Computing Environment", Journal of Computers, Vol. 7, No. 1, pp. 42-52, January 2012.

[11] Ke Liu, Hai Jin, Jinjun Chen, Xiao Liu, Dong Yuan, and Yun Yang, "A Compromised Time Cost Scheduling Algorithm in SwinDeW-C for Instance Intensive Cost Constrained Workflows on a Cloud Computing Platform", The International Journal of High Performance Computing Applications, Vol. 24, No. 4, pp. 445-456, 2010.

[12] Khawar Hasham, Antonio Delgodo Peris, Ashiq Anjum, Dave Evans, Stephen Gowdy, Jose M. Hernandesz, Eduardo Huedo, Dirk Hufnagel, Frank van Lingen, Richard McClatchey, and Simon Metson, "CMS Workflow Execution Using Intelligent Job Scheduling and Data Access Strategies, IEEE Transactions on Nuclear Science, Vol. 58, No. 3, pp. 1221-1232, June 2011.

[13] Lei Wang, Jianfeng Zhan, Weisong Shi, and Yi Liang, "In Cloud, Can Scientific Communities Benefit from the Economies of Scale", IEEE Transactions on Parallel and Distributed Systems, Vol. 23, No. 2, pp. 296-303, February 2012.

[14] Qian Zhu, and Gagan Agrawal, "Resource Provisioning with Budget Constraints for Adaptive Applications in Cloud Environments", IEEE Transactions on Services Computing, 27 December 2011, DOI: 10.1109/TSC.2011.61.

[15] Rajkumar Buyya, Saurabh Kumar Garg, and Rodrigo N. Calheiros, "SLA-Oriented Resource Provisioning for Cloud Computing: Challenges, Architecture, and Solutions", Proceedings of the International Conference on Cloud and Service Computing, IEEE, 2011.

[16] Sandeep Tayal, "Tasks Scheduling Optimization for the Cloud Computing Systems", International Journal of Advanced Engineering Sciences and Technologies, Vol. 5, Issue 2, pp. 111-115, 2011.

[17] Seokho Son, and Kwang Mong Sim, "A Price and Time-Slot Negotiation Mechanism for Cloud Service Reservations", IEEE Transaction on Systems, Man, and Cybernetics – Part B: Cybernetics, Vol. 42, No. 3, pp. 713-728, June 2012.

[18] B. Sotomayor, R. Santiago Montero, I. Martin Llorente, and I. Foster, "Virtual Infrastructure Management in Private and Hybrid Clouds", IEEE Internet Computing, Vol. 13, No. 5, pp. 14-22, September/October 2009.

[19] Soumya Banerjee, Indrajit Mukherjee, and P. K. Mahanti, "Cloud Computing Initiative using Modified Ant Colony Framework", World Academy of Science Engineering and Technology, No. 56, pp. 221-224, 2009.

[20] Thomas Setzer, Kamal Bhattacharya, and Heiko Ludwig, "Change Scheduling based on Business Impact Analysis of Change-Related Risk", IEEE Transactions on Network Service Management, Vol. 7, No. 1, pp. 58-71, March 2010.

[21] Xiao Qin, Hong Jiang, Adam Manzanares, Xiaojun Ruan, and Shu Yin, "Communication-Aware Load Balancing for Parallel Applications on Clusters", IEEE Transactions on Computers, Vol. 59, No. 1, pp. 42-52, January 2010.

[22] Young Choon Lee, Riky Subrata, and Albert Y. Zomaya, "On the Performance of a Dual-Objective Optimization Model for Workflow Applications on Grid Platforms", IEEE Transactions on Parallel and Distributed Systems, Vol. 20, No. 9, pp. 1273-1284, September 2009.

[23] S. Zhong, and J. Ghosh, "A comparative study of generative models for document clustering", Proceedings of the SDM Workshop on Clustering High Dimensional Data and Its Applications, 2003.