Resource Management: Cost Optimal Scheduling with Resource Replication in Cloud Environment

Trilok Sinha, Tinku Kumar, Rohit Patidar
Department Of Information Science Engineering
JSS Academy Of Technical Education
Bangalore, India
trilok0406@gmail.com

Nagamani N P (Asst. Professor)
Department Of Information Science Engineering
JSS Academy Of Technical Education
Bangalore, India
Nagamani1326@gmail.com

Abstract— ***Coping with uncertainty is a challenging and complex problem particularly in hybrid cloud environments—private cloud plus public cloud. For providing an efficient way of load balancing the paper proposes a time scheduler algorithm for the cloud computing. The scheduling is done through two phases. The first phase is direct assignment in the best case and worst condition in the second time in a time slicing phase. To describe the proper methodology the paper first gives a suitable introduction followed by literature review. Then a proposed method is described with the simulation part followed. VM Replicated depending upon the task duration time. Task is assign to replicated VM, divided on duration time of tasks, completion in minimal time. Cost Expansive is limited. Cloud Provider gets more VM cost beneficial. Lowest resources and efficient throughput Job submits to VM based on Cost optimal Algorithm. Here we added VM Replication; this paper gives advancement in the cloud computing in the matter of high processing speed. The process has a least calculation over load on the load balancer.***

Keywords–replication; scheduling; assignment; data center

I. Introduction

Cloud computing is really changing the way, how and where the computing is going to be performed. cloud computing provides a user access to computer resources like machines, storage, operating systems, application development environments, application programs over a network through Web services. While the user not necessarily know the actual physical location and organization of the equipment hosting these resources—be it in the next room or spread across the globe. Cloud computing has got a lot of attention to be used as a computing model for a different kind of application. But the people are still bothering to use it for many applications. But now some researchers and cloud service providers are working to give the power of cloud and associated benefits to the applications. Some of the cloud operators have started real time cloud support. Cloud support for real time system is really important. Because, today we found a lot of real time applications around us. These applications range from small mobile phones to larger industrial controls and from mini pacemaker to larger nuclear plants. Most of them are also safety critical systems, which should be reliable. In general, real-time applications is any information processing system which has to respond to externally generated input stimuli, within a specified period of time [12]. So the correctness depends not only on the logical result, but also the time it was delivered [8]. Failure to respond is as bad as the wrong response [13]. These systems have two main characteristics by which they are separated by other general-purpose systems. These characteristics are timeliness and fault tolerance [6]. By timeliness, we mean that each task in real time system has a time limit in which it has to finish its execution. And by fault tolerance means that it should continue to operate under fault presence [7]. Use of cloud infrastructure for real time applications increases the chances of errors. As the cloud data centers (virtual machines) are far from the transceiver (job submitting data center, actuator or sensor). Many real time systems are also safety critical systems, so they require a higher level of fault tolerance [14]. Safety critical real time systems require working properly to avoid failure, which can cause financial loss as well as casualties [10]. So there is an increased need to tolerate the fault for such type of systems to be used with cloud infrastructure

To take maximum benefit from cloud computing, developers must design mechanisms that optimize the use of architectural. The Virtual Machine’s (VMs) has emerged as an important issue because, through virtualization technology, it makes cloud computing infrastructures to be scalable and using the replication technique the same virtual machine can be replicated, having the same working efficiency. For providing an efficient way of load balancing the paper proposes a time scheduler algorithm for the cloud computing. The scheduling is done through two phases. The first phase is direct assignment in the best case and worst condition in the second time in a time slicing phase. VM Replicated depending upon the task duration time. Task is assign to replicated VM, divided on duration time of tasks, completion in minimal time.

The rest of the paper is organized as follows. In next section literature review, then a problem definition along with proposed method is described

II. Literature Review

Adrian Coles and Bica Mihai in the paper proved the fact that the asynchronous adaptive replication algorithm can improve the performance, client response time and reduce network bandwidth of high availability systems in situations where the environment changes are very often. A drawback of the system was the buffering time. In present, the buffering time is too large and for a small number of modified pages it can be greater even than the network transfers. Some improvements can be made by implementing a better buffering technique.

Brendan Cully, Geoffrey Lefebvre, Dutch Meyer build on top of the well-known Xen hypervisor [5], is an attempt to address the abovementioned challenge. Virtual machines are protected from crash-stop failures by very frequent synchronization using primary-backup approach. It is assumed that both replicas are in the same local area network (LAN). As already described, instead of trying to replay the exact sequence of events on the backup machine, the synchronization is done by copying the new state of the primary to the backup. They described that the most naive way would be to copy the whole VM state from primary to the backup after each single VM instruction executed on the primary. A more efficient way is to make incremental updates, e.g. send only the difference between the previous and the new state. It is also natural to send updates at a higher granularity than single instructions, because synchronization after every minor change in the VM state is impractical. This is taken into account by the authors.

The authors [19], addressed in this work the main problems facing the large number of scientists rely on Bags-of-Tasks (BoTs) in mixtures of computational environments such as grids and clouds, the lack of tools available for selecting efficient scheduling strategies for user-defined utility functions. This paper's ExPERT BoT scheduling framework chooses the Pareto-efficient strategies through a range of replication strategies for running BoTs on mixtures of environments with varying reliability, cost, and speed. For any user provided modifiable function, ExPERT finds the best strategy in a large, sampled strategy space.
The authors [19] validated ExPERT's expected accuracy and showed its viability through a variety of experiments in real and simulated environments. The author found that ExPERT are achieving a 72% cost reduction and a 33% shorter build span compared with commonly-used old scheduling strategies such as combining the reliable and unreliable resources.

The author [20] investigated the problem of virtual resource management for database systems in cloud environments. The author practiced machine learning techniques to learn a system performance model through a data-driven approach. The model captures relationships between the systems resources and database performance. Based on the learned predictive model, the author designed an intelligent resource management system, Smart service level agreement (SLA).

Smart SLA considers some factors in cloud computing environments such as SLA cost, client workload, infrastructure cost, and action cost in a general way. Smart-SLA achieves optimal resource allocation in a situational and intelligent fashion. Practical studies on input data and day to day workloads demonstrated that such an intelligent resource management system has great potentials in improving the profit margins of cloud service providers.

The authors [21] first develop a deterministic model, using a mixed integer linear program, to facilitate resource rental decision making. the author investigates planning solutions to a resource market featuring time-varying pricing. The author conducts time-series analysis over the spot price trace and examines its predictability using Auto-Regressive Integrated Moving-Average (ARIMA). The authors also develop a stochastic planning model based on multistage recourse. The author discover that spot price forecasting does not provide this paper's planning model with a crystal ball due to the weak correlation of past and future price, and the stochastic planning model better hedges against resource pricing uncertainty than resource rental planning using forecast prices.
The authors [22] the author pointed out many challenges in addressing the problem of enabling SLA-oriented resource allocation in data centers to satisfy competing applications demand for computing services. In particular, the user applications are becoming more complex and need multiple services to execute instead of a single service. These complex user applications often require collaboration among multiple organizations or businesses and thus require their specific services to function successfully. Moreover, fast turnaround time is needed for running user applications in today's increasingly competitive business environment.
By addressing SLA-oriented resource allocation in data centers, the author provide a critical link to the success of the next ICT era of Cloud computing. The author also showed how the proposed framework can be effectively implemented using the Aneka platform. We envision the need for a deeper investigation in SLA oriented resource allocation strategies that encompass customer-driven service management, computational risk management, and autonomic management of Clouds in order to improve the system efficiency, minimize violation of SLAs, and improve profitability of service providers.

III. Problem Definition And Proposed Solution

A. Problem Definition

Most of the cloud scheduling algorithms based on availability (time based) of computational resources. It does not managing all the heterogeneous cloud resources. It might be reduce scheduling computational resources times but not cost. Support for dynamic, reconfigurable on demand, secure and highly customizable computing storage and networking environments these all based on time. Resource management challenges such are Performance throughput, multiple domains, Availability of computational resources, Handle of conflicts between common resources demand, Fault tolerance, and Inter domain compatibility.
Managing resources at large scale while providing performance isolation and efficient use of underlying hardware is a key challenge for any cloud management software. The basic approach involves the user accessing a resource when it's idle on a random basis. When the processes need a processor to execute its job, the availability of the processor is checked. When a processor is idle, the process is randomly assigned. But this is detrimental to the efficiency of

the execution because, it doesn't check the type of resource needed for a particular process.

B. *Proposed Solution*

The below figure shows the diagram for the proposed system-
The architectural; diagram clearly shows that the system is started with the user request. The time duration is extracted from the user request. If the request can be processed by a single virtual machine then the work is precede to the virtual machine. In the other case if the duration of the work cannot be processed in a single machine then the work is processed in a number of virtual machines

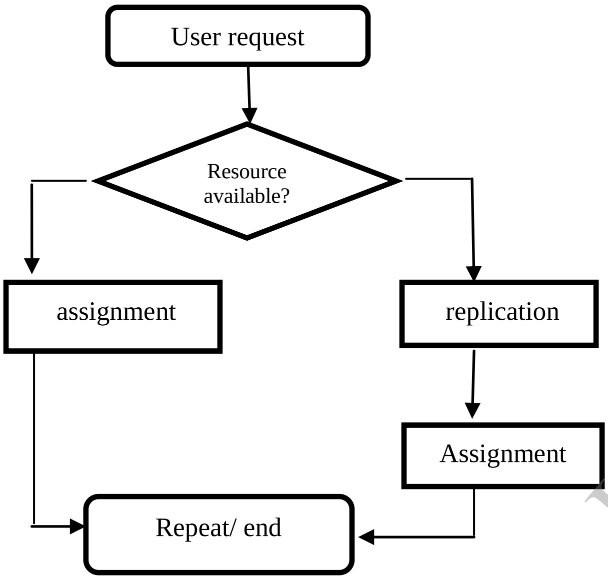

Fig. 1. Architecture Diagram

In present scenario, it is hard to achieve the required process configuration this can be attaining by optimal cost scheduling algorithm it also allow processor replication. Processors replicated depending upon the user process duration time. Scheduling algorithm allocates the resources if available, otherwise the request is put on hold. It is not possible for a cloud server to satisfy all the requests due to finite resources at a time. Each and every processor has some process to evaluate the process and for that the grid user need to pay for cloud server. The concept that has been put forward in this paper involves the use of optimization for identifying the nature of the process and judiciously assigning the resources to each process depending upon the amount of processing power required for that process. Replication of the processors is way to Higher Throughput and reduces the process time. This paper identifies the issues in resource management and scheduling in the emerging cloud computing context and briefly discusses techniques for scheduling using computational economy concept.

Generally Virtual Machines in the Datacenter are running at the time of assigned the task. VM performance is good at the time of small task (duration time).

i. *ASSINGMENT PHASE*

The below figure show the diagram of the module that decides the assignment of work to a virtual machine in a best condition. The below modular diagram clearly shows the initialization of the network starts with the incoming request. The module flows to the iteration phase. Then the load balancer extracts the time duration form the request and matches with the time durations available with the VM wares. If the time duration is present in any of the vm wares; the work is assigned to that VM ware directly. Else the process is subjected to the second phase without committing it.

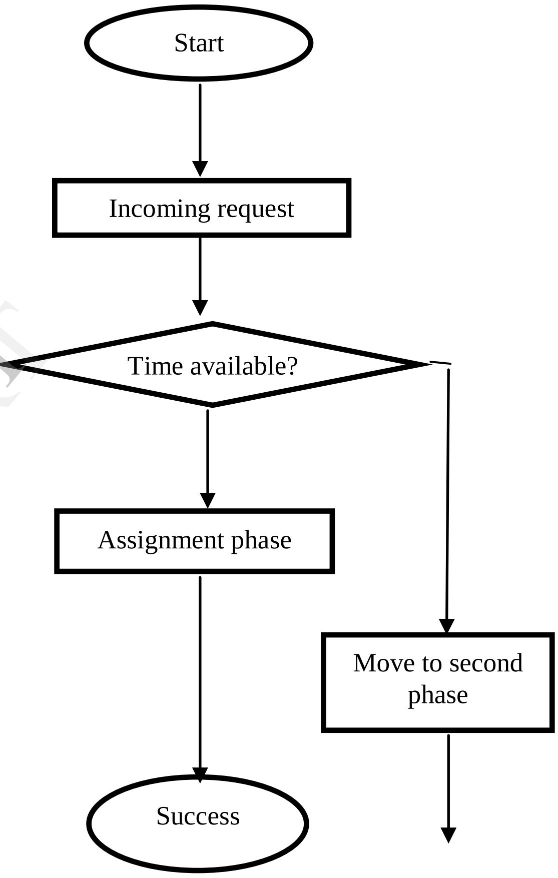

Fig. 2. Modular Diagram For Best Condition

For the technique of job matching, we are using some below formulas.

When a task is coming to the load balancer; it first checks whether the duration of the task is available in the virtual machines or not. This is done through an iterator, which is moved through all the virtual machines. If some duration is available the task is directly assigned to the virtual machine.
Normal: Duration time- 3hrs

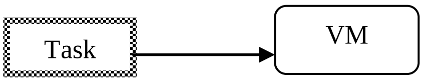

Fig. 3. Task Scheduling Diagram

ii. REPLICATION PHASE

In this phase the duration of the job is analyzed. The process is done through a number of replications. The architectural diagram is given below. According to the diagram the phases starts with the migration value is one. The phase one sets the migration value to 1. After checking the migration value is more than one the process flows to adding an module operator. The modulo operator is exactly 1 more in value to the migrate value. After that it is necessary to check the value is below thresh hold or not. If it is less than thresh hold, the load balancer can't do more modulo so that the standard of the work will we degraded. After the process of modulo; it is checked whether the division of the process is reality or not. If division is not reality then it is thrown an error to the user. On the other hand, the process is once again travelling availability checking phase. If the resource is not available then the migrate value is increased by one and flows once again enter in to the whole process. After going through the module the task is assigned to virtual machines as given below. Here the time duration of the work is 3 hours. The work is forwarded to three virtual machines. Each having the work duration of 1 hour each ,the detail is given below.

VM Replication:
An overall example is given below where Duration time- 3hrs

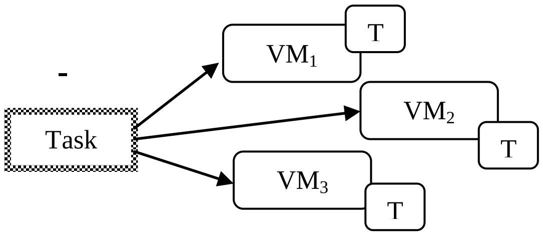

Fig. 4. Task Scheduling Diagram

- Normal → Task (Job) Duration time is 3 hours directly scheduled or assign to VM.

- VM Replication→ Task (Job) Duration time is 3 hours, VM replicate three VMs are VM$_1$, VM$_2$, VM$_3$ and Task Scheduled or assign to T→VM1, T→ VM2, T→ VM3. T is Task divided based on duration time.

- VM replication based on the Task Job duration time.

Duration Time = Number of VM Replication

CONCLUSION

For providing an efficient way of load balancing the paper proposes a time scheduler algorithm for the cloud computing. The scheduling is done through two phases. The first phase is direct assignment in the best case and worst condition in the second time in a time slicing phase. To describe the proper methodology the paper first gives a suitable introduction followed by literature review. Then a proposed method is described with the simulation part followed. VM Replicated depending upon the task duration time. Task is assign to replicated VM, divided on duration time of tasks, completion in minimal time. Cost Expansive is limited. Cloud Provider gets more VM cost beneficial. Lowest resources and efficient throughput Job submits to VM based on Cost optimal Algorithm. Here we added VM Replication,

This paper gives advancement in the cloud computing in the matter of high processing speed. The process has a least calculation over load on the load balancer.

.

REFERENCE

[1] "Vmware high availability." [Online]. Available: http://www.vmware.com/products/high-availability/

[2] B. Cully, G. Lefebvre, D. Meyer, M. Feeley, N. Hutchinson, and A. Warfield, "Remus: high availability via asynchronous virtual machine replication," in NSDI'08: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation. Berkeley, CA, USA: USENIX Association, 2008, pp. 161–174.

[3]Bradford,R.,Kotsovinos,E.,Feldman,A.,and Schioberg H. Live wide-area migration of virtual machines including local persistent state. In VEE '07: Proceedings of the 3rd international conference on Virtual execution environments (New York, NY, USA, 2007), ACM Press, pp. 169–179.

[4] Bressoud,T.C.,and Schneider, F. B. Hypervisor-based fault-tolerance. InProceedings of the Fifteenth ACM Symposium on Operating System Principles(December 1995), pp. 1–11.

[5] Bressoud,T.,and Schneider, F. B. Hypervisor-based fault tolerance. InProceedings of the fifteenth ACM symposium on Operating systems principles(New York, NY, USA, 1995), SOSP '95, ACM, pp. 1– 11

[6] Clark, C., Fraser, K., Hand, S., Hansen,J.G.,JUL, E., Limpach, C., Pratt, I., and Warfield, A. Live Migration of Virtual Machines. InNSDI'05: Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation(Berkeley,CA, USA, 2005), USENIX Association, pp. 273–286

[7] Cully, B., Leefbvre, G., Meyer, D., Feeley, M., Hutchinson,N.,Andwarfield A. Remus: high availability via asynchronous virtual machine replication. In Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation(2008), NSDI'08, pp. 161–174.

[8] DU,Y.,Andyu, H. Paratus: Instantaneous Failover via Virtual Machine Replication. InProceedings of the 2009 Eighth International Conference on Grid and Cooperative Computing(2009), GCC '09, IEEE Computer Society, pp. 307–312

[9] DU,Y.,YU, H., Shi, G., Chen, J., and Zheng, W. Microwiper: Efficient Memory Propagation in Live Migration of Virtual Machines. In Proceedings of the 2010 39th International Conference on Parallel Processing(Washington, DC, USA, 2010), ICPP '10, IEEE Computer Society, pp. 141–149

[10] Dunlap,G.W.,Luccheti, D. G., Fetterman,M.A.,Andchen,
P. M. Execution replay of multiprocessor virtual machines. InProceedings of the fourth ACM SIGPLAN/SIGOPS international conference on Virtual execution environments(2008), VEE '08, pp. 121–130.

[11] Hariharan, R., and Sun, N. Workload Characterization of SPECweb2005. http://www.spec.org/workshops/2006/papers/02 Work loadchar SPECweb2005Final.pdf, 2006

[12] Huang,W.,Gao, Q., Liu, J., and Panda, D. K. High performance virtual machine migration with RDMA over modern interconnects. In

Proceedings of the 2007 IEEE International Conference on Cluster Computing(Washington, DC, USA, 2007), CLUSTER ’07, IEEE Computer Society, pp. 11–20

[13] Kivity, A., Kamay,Y.,Laor, D., Lublin, U., and Ligouri,A. kvm: the Linux virtual machine monitor. InOttawa Linux Symposium (July 2007), pp. 225–230.

[14] Lee, D., Wester, B., Veeraraghavan, K., Narayanasamy, S.,Chen, P. M., AND Flinn, J. Respec: efficient online multiprocessor R playvia speculation and external determinism. ASPLOS ’10, ACM, pp. 77–90

[15] LU, M., and Ckerchiueh, T. Fast memory state synchronization for virtualization-based fault tolerance. InDependable Systems Networks, 2009. DSN ’09. IEEE/IFIP International Conference on(2009), pp. 534 – 543

[16] Mcdougall, R., and Anderson, J. Virtualization performance: perspectives and challenges ahead. Sigops Oper. Syst. Rev. 44(December 2010), 40–56

[17] Nelson, M., Lim, B. H., and hutchins, G. Fast transparent migration for virtual machines. InATEC ’05: Proceedings of the annual conference on USENIX Annual Technical Conference (Berkeley, CA, USA, 2005), USENIX Association, p. 25.

[18] Scales,D.J.,Nelson, M., and Venkitachalam, G. The design of a practical system for fault-tolerant virtual machines.SIGOPS Oper. Syst. Rev. 44(December 2010).

[19] Orna Agmon Ben-Yehuda, Assaf Schuster, Artyom Sharov, Mark Silberstein, Alexandru Iosup “**ExPERT: Pareto-Efficient Task Replication on Grids and Clouds”** Technion - Computer Science Department - Tehnical Report CS-2011-03 – 2011

[20] Pengcheng Xiong yz1, Yun Chi z2, Shenghuo Zhu z3, Hyun Jin Moon z4, Calton Pu y5, Hakan Hacıg¨um¨us, “Intelligent Management of Virtualized Resources
for Database Systems in Cloud Environment”, 2011

[21] H. Zhao and X. Li,” *Resource Management in Utility and Cloud Computing”*SpringerBriefs in Computer Science, 2013

[22] Rajkumar Buyya1,2, Saurabh Kumar Garg1, and Rodrigo N. Calheiros, “SLA-Oriented Resource Provisioning for CloudComputing: Challenges, Architecture, and Solutions”, 2011 International Conference on Cloud and Service Computing