A Survey on Improving Performance of Real Time Scheduling for Cloud Systems

Mrs Rekha A Kulkarni
Assistant Professor
PICT Pune

Prof. Dr. Suhas H Patil
COE
Bharati Vidyapeeth University

Abstract

Cloud computing provides a distributed computing environment in which there is a pool of virtual, dynamically scale-able heterogeneous computing and storage platforms. The computing power and the storage are provided as Services on demand to external user over the internet. Cloud computing aims at making computing as a service whereby shared resources, software, and information are provided to users who are requesting the service as a utility over a network. One of the key technology which plays an important role in Cloud data-center is resource scheduling. Real time processing on the cloud systems requires to satisfy timing constraints of real time systems. In the scenario of time dependent application deadline misses are undesirable. One of the challenging scheduling problems in Cloud data center is to consider allocation and migration of re-configurable virtual machines (VMs) and integrating features of hosting physical machines. While using the facilities provided by cloud environment in the area of real time processing the challenge is that of meeting the deadlines. A real-time scheduler must ensure that processes meet deadlines, regardless of system load or inherent delays caused by the various components of the cloud environment. One solution to this is by selecting the scheduling policy at run time. In this paper the various Real time scheduling algorithms have performance of real time scheduling policies in cloud system with the dynamic selection of scheduling algorithms with the help of fuzzy logic.

Keywords: Cloud Computing, Real Time Scheduling , Fuzzy Logic, Performance, Multi Core.

I. INTRODUCTION

Nowadays, use of real-time cloud systems is dramatically increasing. With the changed design scenario of processors which were having one processor unit previously currently have multiple processing cores laying on the same machine. So even the desktop processing scenario has changed with the introduction of muticore architectures and less is known about how to schedule multiprocessor-based real-time systems than that for uni-processors. Optimal scheduling of real-time tasks and resources on cloud systems is known to be computationally intractable for large data sets. Any practical scheduling algorithm in cloud systems presents a trade-off between performance and computational complexity.

Scheduling theory for cloud computing has been studied extensively over the years. The LATE scheduling algorithm, [2] tries to improve the response times. Agrawal [2] algorithm has tried to improve the Hadoops scheduling by using shared scans. Fairness scheduling has been explored in a number of frameworks such as the default Hadoop's fair-sharing scheduling, the Quincy been studied and dynamic selection of scheduling policy based on fuzzy logic has been proposed. Based on the information collected by the system at the time of scheduling. This is possible by identifying and using the appropriate parameters in selecting the scheduling policy. In the proposed system we have tried to identify the parameters and the effect of these different parameters on the selection policy. The proposed system will try to improve the deadlines. In the proposed system it has been studied to develop a cloud infrastructure that not only supports latency-sensitive and large-scale data processing but also provides timing guarantees.

To improve the accuracy of scheduling techniques, one direction is to investigate parametrized techniques for large-scale data analytic s in the cloud environment. The idea is to incorporate with the analysis useful information that may affect a task's execution time as parameters to achieve more accurate estimates. These parameters include (i) data-related information, e.g., the size etc. Unpredictability is the biggest bane of any real-time scheduler. The idea is to incorporate with the analysis useful information that may affect a task's execution time (or its dependent tasks) as parameters to achieve more accurate estimates.

Fuzzy logic is extension of Boolean logic dealing with the concept of partial truth which denotes the extension to which proposition is true. Whereas classic logic holds that everything can be expressed in terms of binary numbers. Fuzzy logic deals with Boolean truth value with scheduling , and scheduling techniques based on delay scheduling and copy-compute splitting . Grid-based scheduling techniques have also been adopted into the cloud setting.

All these techniques, however, targeted non real-time applications where the primary Objectives are to increase throughput and minimize average response time instead of meeting include pure data-intensive systems or they may also contain data/compute-intensive systems. In that, the former type of systems devote most of their time to data manipulation to facilitate huge data analysis and massive input/output operations.

Elasticity refers to the capability of the cloud to utilize system resources as per the needs and usage. This implies that more capacity can be added existing system. The resources may shrink or grow according to the current state of the cloud. The cloud system should support heterogeneous infrastructure. Homogenous configuration is not always possible for data-intensive systems. In such an environment, issues such as differing computation power across cloud machines, varying disk speeds, and networking hardware with dissimilar capacity are not infrequent. Consequently, a cloud may have to encounter varying delays.

VMs can be rescaled as demand for the applications they host changes. However, from a global scheduling perspective a more powerful tool is live migration of VMs, whereby running VMs are paused, serialized and transferred to a different PM, where they are once again scheduled for execution. Live migration offers many benefits, including degree of truth. Degree of truth is often employed to capture the imprecise modes of reasoning that play an essential role in the human ability to make decisions in an environment of uncertainty and imprecision. Fuzzy Inference Systems (FIS) are conceptually very simple. They consist of an input, a processing, and an output stage. With the use of fuzzy inference engine it will be possible to decide on which can be best scheduling policy for the given situation.

II. RELATED WORK

Nowadays, use of real-time cloud systems is dramatically increasing. Less is known about how to schedule multiprocessor-based real-time systems than that for uni-processors . Optimal scheduling of real-time tasks and resources on cloud systems is known to be computationally intractable for large data sets. Any practical scheduling algorithm in cloud systems presents a trade-off between performance and computational complexity.

Data Intensive computing refers to computing of large scale data. Gorton et al. describe types of applications and research issues for data intensive systems. Such systems may either robustness analysis. The intrinsic uncertainty in dynamic real-time improving fault management, simplifying system maintenance and, from the resource management perspective, enabling global scheduling objectives, such as balancing load across PMs and consolidating VMs on a minimal number of PMs.

In this paper, we propose an dynamic scheduling policy based on based on fuzzy inference model which is more suitable for embedded soft real-time systems in an uncertain environment.

III. PROPOSED SYSTEM

All practical real-time scheduling algorithms in cloud systems present a trade-off between their computational complexity and performance. In real-time systems, tasks have to be performed correctly and timely. Finding minimal schedule in multiprocessor systems with real-time constraints is shown to be NP-hard. The practical scheduling algorithms in real-time systems have not deterministic response time. Deterministic timing behavior is an important parameter for system robustness analysis. The intrinsic uncertainty in dynamic real-time systems increases the difficulties of scheduling problem.

The critical real time applications typically have deadline requirements, and any delay is considered as failure for the whole deployment. For instance, traffic control centers periodically collect the state of roads by sensor devices. Database updates recent information before next data reports are submitted. If anyone consults the control center about traffic problems, a real-time decision should be responded to help vehicle operators choose appropriate control actions.

With the above analysis, the ability to satisfy timing constraints of such real-time applications plays a significant role in cloud environment. However, the existing cloud schedulers are not perfectly suitable for real-time tasks, because they lack strict requirement of hard deadlines. A real-time scheduler must ensure that processes meet deadlines, regardless of system load or inherent delays which are caused by various components of cloud systems.

Our Proposed system S will be represented as : S={ I,O,F,Fa,Su}

Where I=Set of inputs {t1……..tn}{p1…..pn}

O=Set of output {tn - pn} finding an optimal mapping which improves performance gurantees of the real time application.

F= { Set of functions f1,f2,f3…}

Where f1 gathering of information about various resources in cloud

F2 generation of scheduling policy

F3 based on the parameters collected from f2 deciding which is the best policy for given application.

Su : Success case when scheduling policy improves the performance by meeting the dead line of real time task.

Fa : failure case when deadline of time critical applications are not being met.

IV. CONCLUSION

Meeting the deadline for Resource and Timing Constraints is a Challenge in a cloud system. While working with shared environment users will compete for resources to meet their timing deadlines. In such a scenario, resource scheduling plays a critical part in meeting an application's expectations. In large clusters, tasks complete at such a high rate that resources can be reassigned to new jobs on a timescale much smaller than job duration.

The job to be scheduled next according to fairness might not have data on the nodes that are currently free. In a heterogeneous environment like cloud where processing speeds are likely to vary among nodes, high processing nodes are expected to complete more tasks. In such a scenario, the concept of equally distributing data in order to provide data locality is likely to create network

bottleneck. For heterogeneous systems, effective data placement strategies are required in order to ensure efficient task scheduling. So the proposed system tries to select a scheduler at run time which can meet the timing requirements based on the information available locally and globally.

REFERENCES

[1] Kyoungho An, Strategies for Reliable, Cloud-based Distributed Real-time and Embedded Systems 2012 31st International Symposium on Reliable Distributed Systems 1060-9857/12
[2] An Empirical Analysis of Scheduling Techniques for Real-time Cloud-based Data Processing Linh T.X. Phan Zhuoyao Zhang Qi Zheng Boon Thau Loo Insup Lee
[3] A Toolkit for Modeling and Simulation of Real-Time Virtual Machine Allocation in a Cloud Data Center
[4] Data-Intensive Cloud Computing: Requirements, Expectations, Challenges, and Solutions Jawwad Shamsi · Muhammad Ali Khojaye ·J Grid Computing (2013) 11:281–310
[5] Resource Management in Clouds: Survey and Research Challenges Brendan Jennings • Rolf Stadler.