ISSN:2394-3386

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 7
July 2017

Real-Time Tasks Scheduling using Agent-Based Concept

Swapnil Gholap
M.E. Scholar
Y.T.I.E.T.,Bhivpuri road, Karjat, Maharashtra, India.

Vaishali Londhe
Assistant Professor, Y.T.I.E.T., Bhivpuri road, Karjat
Maharashtra, India.

Gayatri Naik
Assistant Professor
Y.T.I.E.T., Bhivpuri road, Karjat, Maharashtra, India.

ABSTRACT

Cloud computing has the potential to dramatically change the landscape of the current IT industry Cloud computing is a model for enabling ubiquitous network access to a shared pool of configurable computing resources. To improve the performance of cloud computing, it is important to not only measure the profit when completing a job in time, but also account for the penalty when a job is aborted or discarded. Note that, before a task is aborted or discarded, it consumes system sources including network bandwidth, storage space, and processing power, and thus can directly or indirectly affect the system performance. Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in third-party data centres. It relies on sharing of resources to achieve coherence and economies of scale, similar to a utility (like the electricity grid) over a network. At the foundation of cloud computing is the broader concept of converged infrastructure and shared services. But scheduling a real time task is a main role for a cloud provider. In this project, the system introduces a novel agent-based scheduling mechanism to maintain its quality of service and enhance the system's performance.

I. INTRODUCTION

Proponents claim that cloud computing allows companies to avoid upfront infrastructure costs, and focus on projects that differentiate their businesses instead of on infrastructure. Proponents also claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and enables IT to more rapidly adjust resources to meet fluctuating and unpredictable business demand. Cloud providers typically use a "pay as you go" model. This can lead to unexpectedly high charges if administrators do not adapt to the cloud pricing model.

The present availability of high-capacity networks, low-cost computers and storage devices as well as the widespread adoption of hardware virtualization, service-oriented architecture, and autonomic and utility computing have led to a growth in cloud computing. Companies can scale up as computing needs increase and then scale down again as demands decrease. To improve the performance of cloud computing, one approach is to employ the traditional utility accrual (UA) approach.

First proposed to associate each task with a Time Utility Function (TUF), which indicates the task's importance. Specifically, the TUF describes the value or utility accrued by a system at the time when a task is completed. Based on this model, there has been extensive research results published on the topic of UA scheduling. While Jensen's definition of TUF allows the semantics of soft time constraints to be more precisely specified, all these variations of UA- aware scheduling algorithms imply that utility is accrued only when a task is successfully completed, and the aborted tasks neither increase nor decrease the accrued value or utility of the system.

In this project, the scheduling process based on agent technique, multiple individual agents are capable of composing a multi-agent system, in which these agents interact with each other to accomplish the goal of the system. Meanwhile, interaction technology among agents is of great importance, and the corresponding rules in the interactions can be designed by users, which make agent-based technology flexible enough to meet totally different requirements while scheduling.

To facilitate interactions, the ability to cooperate, coordinate, and negotiate with each other is required. Cooperation is the process when several agents work together and draw on the broad collection of their knowledge and capabilities to achieve a common goal. Coordination is the process of achieving the state in which actions of agents fit in well with each other. Negotiation is a process by which a group of agents communicate with one another to try to come to a mutually acceptable agreement on some matter. Agent-based scheduling is to employ these operations so as to finish task allocation on cloud resources.

II. LITERATURE SURVEY

The Xiaomin Zhu et.al, has advocated the scheduling method in their paper [1] that novel agent-based scheduling mechanism in cloud computing environment to allocate real-time tasks and dynamically provision resources. A bidirectional announcement-bidding mechanism and the collaborative process consist of three phases, i.e., basic matching phase, forward announcement-bidding phase and backward announcement-bidding phase.

In Cloud Computing and Emerging IT Platforms [2] paper R. Buyyaa et.al, elaborated the significant advances in Information and Communications Technology (ICT) over the last half century, there is an increasingly perceived vision that computing will one day be the 5th utility (after water, electricity, gas, and telephony). This computing utility, like all other four existing utilities, will provide the basic level of computing service that is considered essential to meet the everyday needs of the general community. Then, we present some representative Cloud platforms, especially those developed in industries, along with our current work towards realizing market-oriented resource allocation of Clouds as realized in Aneka enterprise Cloud technology. The benefit of this is high-end computing systems such as Clouds are used for hosting applications containing short-lived or long-lived processing tasks but A confirmed reservation must not end after the deadline and cost more than the budget.

The QoS Guarantees and Service Differentiation for Dynamic Cloud applications [3] by Jia rao et.al, says Cloud elasticity allows dynamic resource provisioning in concert with actual application demands. Feedback control approaches have been applied with success to resource allocation in physical servers. However, cloud dynamics make the design of an accurate and stable resource controller challenging, especially when application-level performance is considered as the measured output. Application-level performance is highly dependent on the characteristics of workload and sensitive to cloud dynamics. To address these challenges, we extend a self-tuning fuzzy control (STFC) approach, originally developed for response time assurance in web servers to resource allocation in virtualized environments. Here it is used in to track the utilization of CPU and allocate CPU resources correspondingly to maintain the utilization to a specified value but These metrics include but are not limited to performance metrics (e.g., response time and throughput).

In energy-aware resource allocation heuristics for efficient management of data centers for cloud computing paper [4] Anton Beloglazovet.al, says we need Green Cloud computing solutions that can not only minimize operational costs but also reduce the environmental impact. In this paper, they define an architectural framework and principles for energy-efficient cloud computing. Based on this architecture, we present our vision, open research challenges, and resource provisioning and allocation algorithms for energy-efficient management of cloud computing environments. The proposed energy-aware allocation heuristics provision data center resources to client applications in a way that improves energy efficiency of the data center, while delivering the negotiated Quality of Service (QoS). The advantage of this is that the knowledge can be used to develop workload-aware resource allocation algorithms, which can be incorporated into energy-efficient resource management strategies in data. But that is not suitable for a large-scale data center environment that has to be able to quickly respond to changes in the workload.

A Novel Fault-tolerant Scheduling Algorithm [8] the Xiao Qin et.al, investigate an efficient off-line scheduling algorithm in which real-time tasks with precedence constraints can tolerate one processor's permanent failure in a heterogeneous system with fully connected network. The tasks are assumed to be non preemptable, and each task has two copies that are scheduled on different processors and mutually excluded in time. It perform ability are described and used for performance assessment of eFRD in comparison with the relevant and quantitatively comparable algorithms.

The Borja Sotomayor et. al, presented in open source solution for virtual infrastructure management[12] paper that Open Nebula, an open source virtual infrastructure manager that can be used to deploy virtualized services on both a local pool of resources and on external IaaS clouds, and Haizea, a resource lease manager that can act as a scheduling backend for Open Nebula providing features not found in other cloud software or virtualization-based datacenter management software, such as advance reservations and resource preemption, which we argue to be specially relevant for private and hybrid clouds. Open Nebula is a virtual infrastructure manager that can be used to deploy and manage virtual machines, either individually or in groups that must be co-scheduled, on local resources or on external public clouds. sBut The primary aim of these private cloud deployments is not to sell capacity over the Internet through publicly-accessible interfaces.

In Truthful Dynamic Workflow Scheduling Mechanism[10] H. M. Fard introduce a pricing model and a truthful mechanism for scheduling single tasks considering two objectives: monetary cost and completion time. With respect to the social cost of the mechanism, i.e., minimizing the completion time and monetary cost. They extend the mechanism for dynamic scheduling of scientific workflows. They theoretically analyze the truthfulness and the efficiency of the mechanism and present extensive experimental results showing significant impact of the selfish behavior of the cloud providers on the efficiency of the whole system. This service is directly used in the brokerage layer implementing our proposed pricing model and scheduling mechanism by selecting the most adequate resources in terms of execution time and monetary cost for users but the proposed algorithms designed for utility Grids are not sprightly useable in clouds.

III. PROJECTED BLOCK DIAGRAM OF SYSTEM

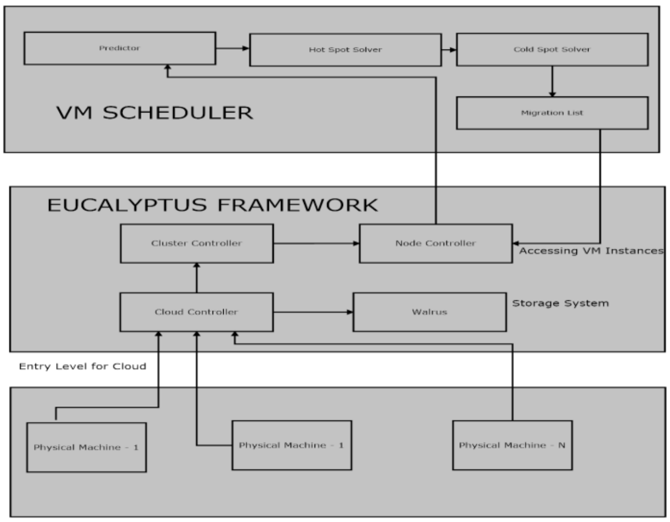

Figure 1: System Architecture of Real Time Tasks Scheduling Using Agent Based Concept

In proposed system, the system proposes an agent-based dynamic scheduling algorithm named Real-Time Tasks Scheduling Using Agent Based Concept. To allocate real-time tasks and dynamically provision resources, the system proposes an agent-based scheduling mechanism and designs a bidding mechanism based on an improved contract net protocol and then develops an agent-based scheduling algorithm for independent, real-time tasks.

This proposed system designed selection strategies matching selection strategy to determine the contractors and investigated a dynamic scaling up method used for our system to further enhance the schedulability. The agent-based scheduling algorithms can be classified into two categories i.e. threshold-based algorithms and market-based algorithms. In the first category, scheduling algorithms are developed from the threshold model in insect colonies. The system uses agents namely task agent and VM agent in the agent-based scheduling algorithms to do this process. Specifically, the proposed system integrates the aforementioned bidding mechanism and the selection strategies based on threshold. Moreover, the system efficiently considers the schedulability, priority, and load balancing.

In the process of interactions, the VM agent set constantly generates or updates VM agents based on the real VM configuration. That is to say, if a new VM is created, then a new VM agent is built; if a VM is cancelled, then the corresponding VM agent is removed; if a VM's performance such as CPU, RAM is varied, the relevant VM agent information is updated. Once VMs' information is changed, the VM agent set forwards it to the manager agent on its board. Now, we are in the position to discuss how the bidding mechanism works. It basically includes basic matching phase, bidding phase, and threshold matching phase.

The proposed system has many advantages like it has the ability to allocate tasks through negotiation, which brings great advantages for dealing with dynamically arrived tasks in distributed systems, it makes it be perfect platform to process real-time tasks, the scale of VMs can dynamically vary according to the current state of the system, t he performance of the system is high.

IV. METHODOLOGY

A) Prediction Module

Predict the future resource needs of VMs. As said earlier, our focus is on internet applications. One solution is to look inside a VM for application level statistics, e.g., by parsing logs of pending requests. Doing so requires modification of the VM which may not always be possible. Instead, we make our prediction based on the past external behaviors of VMs. Prediction module is helpful for bidding mechanism. So we come to know how much recourse will be required by VMs. So that if resource quota of one virtual machine is full then we can allocate task to another virtual machine.

B) Hot Spot Mitigation Module

We sort the list of hot spots in the system in descending temperature (i.e., we handle the hottest one first). Our goal is to eliminate all hot spots if possible. Otherwise, keep their temperature as low as possible. For each server p, we first decide which of its VMs should be migrated away. We sort its list of VMs based on the resulting temperature of the server if that VM is migrated away. We aim to migrate away the VM that can reduce the server's temperature the most. In case of ties, we select the VM whose removal can reduce the load of the server the most. For each VM in the list, we see if we can find a destination server to accommodate it. The server must not become a hot spot after accepting this VM. Among all such servers, we select one whose load can be reduced the most by accepting this VM. Note that this reduction can be negative which means we select the server whose skewness increases the least. If a destination server is found, we record the migration of the VM to that server and update the predicted load of related servers. Otherwise, we move onto the next VM in the list and try to find a destination server for it.

C) Migration Module

Our algorithm executes periodically to evaluate the resource allocation status based on the predicted future resource demands of VMs. We define a server as a hot spot if the utilization of any of its resources is above a hot threshold. This indicates that the server is overloaded and hence some VMs running on it should be migrated away. The temperature of a hot spot reflects its degree of overload. If a server is not a hot spot, its temperature is zero. We define a server as a cold spot if the utilizations of all its resources are below a cold threshold. This indicates that the server is mostly idle and a potential candidate to turn off to save energy.

D) Trust Allocation Module

The physical machines provide a set of virtual machines which are configured dynamically according to user requests. When the limited physical machines are provided to users from a pool of resources, the provided resources have two types; one is the dedicated resources and the other is the undedicated resources to give some extra margin in case of sudden request In this Cloud system environment, if a new user requests resources when all of the resources are already assigned, then the undedicated resources allocated to others are provided to the new users via dynamic reconfiguration. Here we calculate the trust model based on the historical information. By analyzing the vm load and the load of resource the trust value is calculated.

VI. APPLICATION

The Wireless Real Time Tasks Scheduling Using Agent Based Concept System has following applications-

- Significantly improve the Real-Time scheduling existing approaches.
- Gain high scalability.
- Its ability to handles the large amount of data.
- Can be use for medical applications
- Can be use for weather reporting system.
- Used for military purpose where response and execution time is plays important role.

REFERENCES

[1] Xiaomin Zhu, *Member, IEEE*, Chao Chen, Laurence T. Yang, *Senior Member, IEEE*, and Yang Xiang, *Senior Member, IEEE*ANGEL: Agent-Based Scheduling for Real-Time Tasks in Virtualized Clouds

[2] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, I. Brandic, “Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility,” Future Generation Comput. Syst., vol. 57, no. 3, pp. 599-616, 2009.

[3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, “A View of Cloud Computing,” Commu. the ACM, vol. 53, no. 4, pp. 50-58, 2010.

[4] J. Rao, Y. Wei, J. Gong, and C. Xu, “QoS Guarantees and Service Differentiation for Dynamic Cloud Applications,” IEEE Trans. Network and Service Management, vol. 10, no. 1, pp. 43-55, 2013.

[5] A. Beloglazov, J. Abawajy, and R. Buyya, “Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing,” Future Generation Comput. Syst., vol. 28, no. 5, pp. 755- 768, 2012.

[6] L. He, D. Zou, Z. Zhang, C. Chen, H. Jin, and S. A. Jarvis, “Developing Resource Consolidation Frameworks for Moldable Virtual Machines in Clouds,” Future Generation Comput. Syst., vol. 32, pp. 69-81, 2014.

[7] M. Dong, H. Li, K. Ota, and H. Zhu, “HVSTO: Efficient Privacy Preserving Hybrid Storage in Cloud Data Center,” Proc. 3rd Workshop on Communications and Control for Smart Energy Systems CCSES ’14) - INFOCOM ’14 Workshop pp. 529-534, 2014.

[8] X. Qin and H. Jiang, “A Novel Fault-Tolerant Scheduling Algorithm for Precedence Constrained Tasks in Real-Time Heterogeneous Systems,” Parallel Comput., vol. 32, no. 5, pp. 331-356, 2006.

[9] K. Plankensteiner, R. Prodan, T. Fahringer, A. Kertesz, and P. Kacsuk, “Fault-Tolerant Behavior in State-of-the-Art Grid Workflow Management Systems,” Technical Report TR-0091, Inst. On Grid Information, Resource and Workflow Monitoring Services, CoreGRID-Network of Excellence,2007.

[10] H. M. Fard, R. Prodan, and T. Fahringer, “A Truthful Dynamic WorkflowScheduling Mechanism for Commercial Multicloud Environments,” IEEETrans. Parallel and Distributed Systems, vol. 24, no. 6, pp. 1203-1212,2013.

[11] L. F. Bittencourt, E. R. M. Madeira, and N. L. S. da Fonseca, “Scheduling in Hybrid Clouds,” IEEE Communications Magazine, pp. 42-47, 2012.

[12] B. Sotomayor, R. S. Montero, I. M. Liorente, and I. Foster, "Virtual Infrastructure Management in Private and Hybrid Clouds," IEEE Internet Computing, pp. 14-22, 2009.
[13] M. Mishra, A. Das, P. Kulkarni, and A. Sahoo, "Dynamic ResourceManagement Using Virtual Machine Migrations," IEEE Communications Magazine, pp. 34-40, 2012.
[14] X. Liu, C. Wang, B. Zhou, J. Chen, T. Yang, and A. Y. Zomaya, "Priority-Based Consolidation of Parallel Workloads in the Cloud," IEEE Trans. Parallel and Distributed Systems, vol. 24, no. 9, pp. 1874-1883, 2013.
[15] Y. Mei, L. Liu, X. Pu, S. Sivathanu, and X. Dong, "Performance Analysisof Network I/O Workload in Virtualized Data Centers," IEEE Trans.Services Computing, vol. 6. no. 1, 2013.