International Journal on Data Science and Technology
2015; 1(1): 1-5
Published online January 25, 2016 (http://www.sciencepublishinggroup.com/j/ijdst)
doi: 10.11648/j.ijdst.20150101.11

Multiple Agents Based Scheduling and Monitoring in Cloud System

Devenedra Kumar Sahu, Pankaj Kawadkar

Dept. of Computer Science, Patel Institute of Engineering & Science, Bhopal, MP, India

Email address:
Devendrasahu.info@gmail.com (D. K. Sahu), kawadkarpankaj@gmail.com (P. Kawadkar)

To cite this article:
Devenedra Kumar Sahu, Pankaj Kawadkar. Multiple Agents Based Scheduling and Monitoring in Cloud System. *International Journal on Data Science and Technology.* Vol. 1, No. 1, 2015, pp. 1-5. doi: 10.11648/j.ijdst.20150101.11

Abstract: Cloud computing is associated with a new paradigm for the provision of computing infrastructure and services. It represents a shift away from computing as a product that is purchased, to computing that is delivered as a service to consumers over the Internet from large scale data centers or clouds. Clouds provide an infrastructure for easily usable, scalable, virtually accessible and adjustable IT resources that need not be owned by an entity but can be delivered as a service over the Internet. The cloud concept eliminates the need to install and run middleware and applications on users own computer by providing Infrastructure, Platform and Services to users, thus easing the tasks of software and hardware maintenance and support. In this paper we propose a concept of multi agent based batch scheduling and monitoring in cloud computing environment, where the number of agent are more than or equal to two with reducing the complexity of accessing and responding time.

Keywords: Grid, Cloud, Utility Computing, IaaS, SaaS, PaaS, Agent Based Scheduling, Multi Agent Based Scheduling, Batch Scheduling, Monitoring

1. Introduction

Heterogeneity in the computational requirement, dynamic choice and infrequent usages types of resources of the users in modern era has main challenge for service provider (application developer and hardware manufacturer). Secondly, Now computing power or connected computing power (with network) has more demanding and significant role in almost all areas of epoch including market analysis, searching, map, accounting, medical, trading, shopping, rescue operations and many more, the list is endless. Various devices (Computing) and application has been developed and developing to fulfill the common users need. However different users have different requirements of computational power and application and systems software. Hence demand of users is heterogeneous in nature so that varieties of application (hardware & software) have been developed to achieve the highest user satisfaction. Advancement of electronics and telecommunication field has done the job. Specialization has more promising than generalization due to expertise in specific job/function but it also has dark sides. Various requirements require numerous specialized devices (CPU, storage etc.) and software tools. Purchasing or licensing of all such required items (devices & applications) is not feasible to the organization or individuals in terms of the cost and installation. Secondly most of the resources are idle i.e. frequently not used. Hence the utility types of computing paradigm will play an import role. Cloud computing is a new computing paradigm based on utility computing model which will fulfill the user's requirement dynamically on rent basis.

According to the Lewis Chunningham [20] "Cloud computing is the internet to access someone else's software running on someone else's hardware in someone else's data center".

More comprehensive concept about cloud computing has been narrated and drafted by National Institute of Standard & Technology (NIST): According to NIST-"Cloud computing is a paradigm for facilitating expedient, on-demand network access to a shared cluster (pool/collection) of configurable computing power and resources (like applications, services, networks, servers, and storage,) that can be expeditiously provisioned and exemption with least management endeavor or without service provider interaction. This cloud paradigm endorsed availability and is possessed of five imperative characteristics, three service models, and four deployment models."

2. Proposed Concept

Modern era is reflection of human creative thinking and application of optimize solution for the problems mapped and simulated into the machines using technological skills and advancement on it. Cloud computing is another example of technological advancement which offers dynamic provisioning of the utility on rent basis to the subscriber. Cloud offers instant service (software, platform or infrastructure) to requisite dynamically. Sometime's subscriber need more resource (like network bandwidth, CPU or memory) or sometimes it require to less. A cloud service provider has deployed and manages to sufficient number of resource that has been shared to the entire subscriber as per the load requirement of the individuals. To achieve this task cloud service provider required highly efficient scheduling approach and the proper monitoring of the services provisioned or will be provision to subscriber. All the scheduling and monitoring assured the high reliability, automatic scalability, fault tolerance services in secure manner. Proposed agent based approach has provides the efficient and accurate solutions for efficient scheduling and monitoring in cloud computing. In the cloud computing. Agents are the self-executable code work on behalf of the humans. They are able to communicate i.e. social in nature, mobile i.e. can roam in the network, perform the task at remote stations and send back the results to source platform (where they been originated), agents are also clone themselves and one of the core property of the agent is autonomy i.e. autonomous and distributive in nature. Hence agent based solution has been proposed to meet the requirement of the modern cloud computing with pace of dynamic provision to insure shrink in shrink out (elasticity) of the cloud service provider to achieve highest scalability and reliability in extent of the maximum availability of the service to the requisites.

For implementation and evaluation of proposed approach public cloud has been chosen due to cost effective experimental setup. Outcome the results shows that the provisioning of SaaS (Software as a service) and its monitoring using agent has gives better result which is more efficient than existing approach. Integration of the agent in the propose system provides the cost effective and reliable with dynamic pace, solution for efficient scheduling (elasticity of the resource and services) and proposer monitoring of the cloud computing systems.

For developing proposed agent based system three types of public cloud and their services has been selected as test bed for better evaluation and measurement of the accuracy of the propose system. They are following with respective functionality in the proposed system

1. Codenvy – To develop an / are application i.e. SaaS (Software as a service). For the proposed system an java web application using JSP (Java Server pages) application has been chosen to develop on to the codevnySaaS cloud service provider.

2. Cloudbees – To deploy and test our SaaS application onto the cloud, propose system needs a platform i.e. Platform as a Service (PaaS). For this Cloudbees service provider has been integrated onto the developed SaaS application.

3. New Relic – To develop the core functionality of the proposed system. i.e. monitoring and scheduling using software agent New Relic service has been subscribed. In this the java agent has been customized to meet the monitoring and scheduling of the SaaS services.

3. Problem Identification and Proposed Solution

Propose system has surveyed and identified the problem domain that must be addressed in context of the cloud computing and consequently present the idea of agent integration. These are following

1. Service scheduling delay
2. Elasticity optimization
3. Better Provisioning of the SaaS
4. Fault tolerance

While author [1] and [2] proposed an agent based solution to solve the above listed QoS parameter that greatly affect the performance of cloud service especially SaaS.

But the main problem while looking [1] and [2] is the realization and effectiveness of the agent with cloud for better optimization of the service delivery. The main lacking point in the article [1] and [2] is validation of the proposed mechanism.

Additionally the requirements for such fast provisioning of the cloud has been discuss in the recent year in the article [3]. Our main research work is to enhance the agent based model for SaaS delivery in the cloud as depicted in the [1] and [2].

Detail Objective of the proposed agent based SaaS Service. Following goals has been achieved or solved with integrating of the Mobile Agent to Cloud Computing service realization

1. To Evaluate and delivered the cloud computing services (SaaS) using agent (for better and fast delivery) using public cloud such as "New Relic and cloud bees".
2. Deploying a web services under SaaS paradigm and evaluate the effectiveness of the web application in the cloud environment with the help of agent. For SaaS development Codenvy has been subscribed. In which jsp based application has been develop and deployed on cloudbees PaaS.
3. Evaluation and Public PaaS (platform as a Service) of the Cloud bees service integrating a SaaS deployment on it and delivering through agent.
4. Measuring the performance of the proposed analytical approach (influenced from Aneka) in cloud services such as public Cloud bees.

4. Proposed Algorithm

Provisions of service and resources in cloud PaaS is an important function that provides analytical statistics about the current view of cloud (running instance for a user or group of users).

Model for Proposed Work

Our proposed work is to schedule and monitor cloud SaaS application onto the cloud and evaluate the performance of the same using proposed agent based.

4.4 Proposed Algorithm for Provisioning Application and

Resources:

Algorithm for Scheduling (influenced from Aneka) developed onto the Cloudbees

Proposed Algorithm

```
Step1: Register user login
Step2: Sign in the user login
Step3: Select multi monitoring agent in cloud environment (for each subc_request with QoS constraints)
Step 4: Mapping of resources between cloud client, cloud agent and cloud service providers.
Step 5: Resources = Available_resources for the requested SaaS application
Jobs_pending = Number of jobs in the queue;
Cycle Time=30 Min;
Effort = (Jobs_pending /resources) × average_Jobs_Runtime;
If (cycle time ≤ 30)
{
    if (effort >Remaining_Time_application)
    {
        additionalResources = (Jobs_pending × AverageJobsRuntime) / Remaining_Time_application;
        CALL_Monitoring_ Multi_Agents (job_Id); // for resource provisioning
        Penalty= Job Finished Time - Total Time;
    }
    else
    toRelease = 0;
    Profit=Total_Time – Job_Finished_Time;
Step 6:
    if (Jobs_pending< resources)
    {
        toRelease ← Resources-Jobs_pending;
        Profit=Total_Time – Job_Finished_Time;
    }
    else
    {
        Jobs_pending = Jobs_pending + Jobs_running;
        Effort = (Jobs_pending /resources)×average_Jobs_Runtime;
        if (effort <Remaining_Time_application)
        toRelease ←Resources – (Jobs_pending×average_Jobs_Runtime)/Remaining_Time_application
    }
    CALL_Relaese_resources_Multi_Agent(job_Id);
    Penalty= Job_Finished_Time - Total_Time;
}
Else
{
    Print "Increase the cycle time";
}
Step7: End of Process
Step8: End of Process
Step9: End of Process
Step10: Sign out the user login
Step11: Complete the Process with maximum throughput
```

5. Experiments Platform and Dataset

Following public cloud service providers or tools (platform) and data set has been used for evaluating the effectiveness of the proposed monitoring and provision approach.

Table 1. *Experimental Setup.*

Data Set	Cloud Tools
JSP Response Page	Codenvy Cloudbees New Relic

Summary MAP of the SaaS using Agent – Figure 1 has shows the overall MAP of the proposed SaaS with the help of the customized new relic agent.

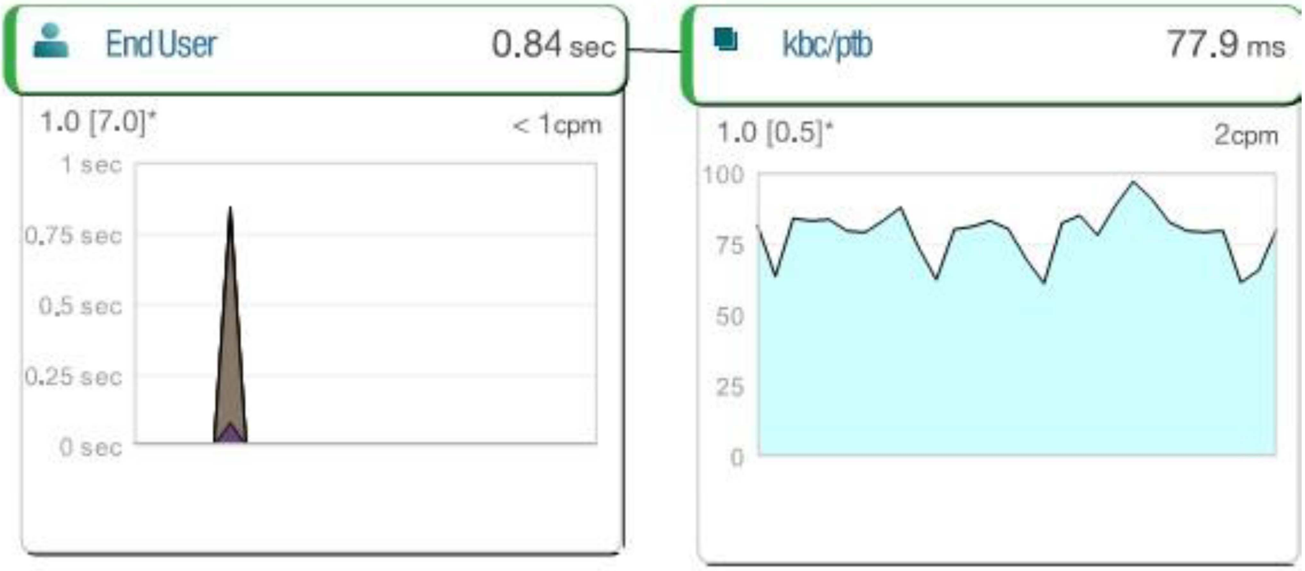

Figure 1. *Brief MAP of usages of the SaaS monitored using Agent.*

6. Performance Measurement

For evaluating the performance of the proposed agent system, the obtained results have been compared with [2] in which author has proposed “MABOCCF” the realization of the federation of different cloud (cloud interoperability) using agent. Author has choose two matrices to evaluate the performance of the it’s proposed MABOCCF technique – average user satisfaction and another one is average utilization ratio which has been derived from following fundamental (base) matrices

1. Number of tasks submitted at instant i (Ni)
2. Time to execute the task
3. Availability
4. Scalability

Author has compared the outcome of their experiment with Non - MABOCCF (NMABOCCF) technique.

Proposed multi agent based solution has influenced from [2] but it’s not the realization of cloud federation rather it has to evaluate the scheduling and monitoring of the SaaS (task) application in public cloud’s (cloud federation not interoperability). All the matrices of the performance checking has been same meaning as our proposed system generated like

1. Response time is same as to average utilization ratio in addition to CPU usages. Table 2 shows the response time of the deployed SaaS obtained results and has been compared with existing agent based method (in seconds)

Table 2. Response Time.

No. of task	NMBOCCF	MBOCCF	Proposed Multiple Agent Based
0	0 sec	0 sec	0 sec
20	1 sec	0.97 sec	0.7 sec
40	0.98 sec	0.97 sec	0.69 sec
60	0.98 sec	0.96 sec	0.65 sec
80	0.97 sec	0.95 sec	0.631 sec
100	0.963 sec	0.954649 sec	0.6171 sec

2. Average user satisfaction is same as to availability and scalability of the proposed system. Table 3 shows the user satisfaction of the deployed SaaS obtained results and has been compared with existing malty agent based method

Table 3. Average user Satisfaction Ratio (in %).

No. of task	NMBOCCF	MBOCCF	Proposed Multiple Agent Based
0	0	0	0
20	0.71	0.95	1
40	0.69	0.94	0.97
60	0.681	0.932	0.969
80	0.665	0.923	0.956
100	0.632389	0.923713	0.95123

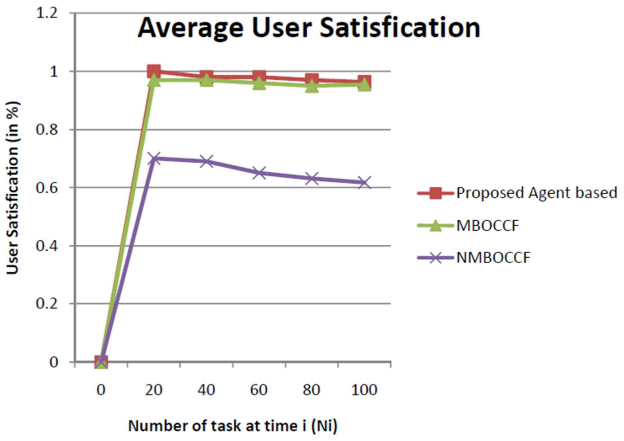

***Figure 2.** Average user satisfaction.*

Rather than performing analytical calculation, proposed system has been compared with the author’s MABOCCF and NMABOCCF method. Our response time and scalability availability and CPU show that the obtained result has closest to 100 (97-99%). Hence proposed system performs better than [2] and any other methods as shown in figure 3.

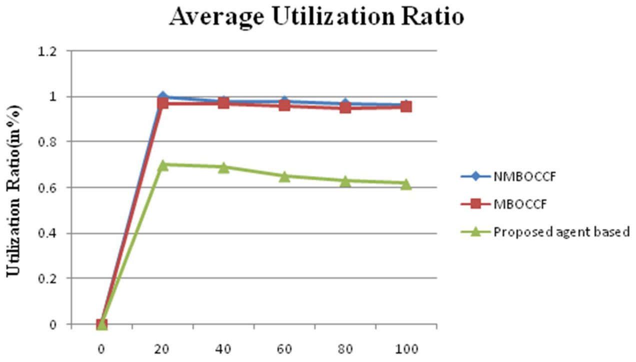

***Figure 3.** Average Utilization Ratio.*

7. Conclusion

This paper proposed and developed an agent based enhanced method for better scheduling (for resource like CPU, memory etc. for their granting/releasing) and their exact monitoring in the cloud in context of public cloud computing service provider such as Cloudbees. This paper presents the enhanced agent based solution to ensure better elasticity and monitoring solution. Analytical analysis is to collect statistics to check the required number of resources needs or used and provides dynamic indication to better elasticity achievement. Proposed agent based solution for guaranteed better elasticity and their efficient monitoring of the resources in the cloud which helps to gather analytical statistics of the resources currently held and will be used such a memory, number of instances and CPU. Proposed mechanism has influences from the working of Aneka framework. For evaluation of the propose agent based method data set (jsp) has been developed using jsp web pages and deployed onto the cloud evaluating the elasticity and its monitoring. The developed java web application (SaaS) has been developed with the help of codenvy SaaS developed platform. To deploying created SaaS application in the cloud a PaaS service has been required to be subscribed, for this cloudbees PaaS service has been chosen. Then for monitoring and scheduling with software agent New Relic service has been used to customized the agent functionality to meet the propose systems requirement. Proposed agent based methods obtained result has been found satisfactory and performs better than existing available solution. Following few areas has been chosen as future work as derivative of the proposed multi agent based solution where the current work can be taken further.

1. Security enhancement using Agent for following attack internal attacks and DoS (Denial of Service) attack.

2. Develop a security perimeter based on anomaly detection using Application Process Management by integrating the mobile Agent on them for the cloud computing paradigm.

References

[1] Gaoyun Chen, Jun Lu, Jian Huang, Zexu Wu, “SaaAS - The Mobile Agent based Service for Cloud Computing in Internet Environment”, IEEE, Sixth International Conference on Natural Computation (ICNC 2010), pp. 2935–2939, 2010.

[2] Zehua Zhang, Xuejie Zhang, “Realization of Open Cloud Computing Federation Based on Mobile Agent”, IEEE, pp. 642–646, 2009.

[3] Javier Espadas, Arturo Molina, Guillermo Jiménez, Martín Molina, Raúl Ramírez, David Concha, “A tenant-based resource allocation model for scaling Software-as-a-Service applications over cloud computing infrastructures”, Elsevier Science Direct, Future Generation Computer Systems 29 (2013), pp. 273–286, 2013.

[4] IBM Blue Cloud project [URL]. [Online] Available: http://www-03.ibm.com/ press/us/en/pressrelease/22613.wss/.

[5] Rizwan Mian, Patrick Martin, Jose Luis Vazquez-Poletti “Provisioning data analytic workloads in a cloud”, Elsevier, Future Generation Computer Systems. The Characteristics of Cloud Computing, 2012.

[6] Chunye Gong, Jie Liu, Qiang Zhang, Haitao Chen, Zhenghu Gong, “The Characteristics of Cloud Computing”, 39th International Conference on Parallel Processing Workshops, pp. 2010.

[7] Amit Nathani, Sanjay Chaudhary, Gaurav Soman “Policy based resource allocation in IaaS cloud”, Elsevier, Future Generation Computer Systems 28, 2012, pp. 94–103.

[8] Naidila Sadashiv, S. M Dilip Kumar, “Cluster, Grid and Cloud Computing: A Detailed Comparison”, IEEE, The 6th International Conference on Computer Science & Education (ICCSE 2011) August 3-5, 2011. SuperStar Virgo, Singapore 2011.

[9] Bernd Grobauer, Tobias Walloschek, Elmar Stöcker Siemens, “Understanding Cloud Computing Vulnerabilities”, IEEE, Copublished By The IEEE Computer and Reliability Societies, 2011.

[10] Jianwei Yin, Yanming Ye, Bin Wu, Zuoning Chen, “Cloud Computing Oriented Network Operating System and Service Platform”, 1st IEEE Workshop on Pervasive Communities and Service Clouds, IEEE, 2011.

[11] Dan Svantesson and Roger Clarke “Privacy and consumer risks in cloud computing”, Elsevier, computer law & securety review 26 (2010) 391e397.

[12] L. M. Vaquero, L. R. Merino, J. Caceres, M. Lindner, “A break in the clouds: Towards a cloud definition”, ACM SIGCOMM Computer Communication Review, Vol. 39, No. 1, 2009.

[13] “CLOUD COMPUTING” book authored by John W. Ritting house and James F Ransome.

[14] D. Chappell, "Introducing Windows Azure", David Chappell & Associates, 2009.

[15] I. Raicu, Y. Zhao, C. Dumitrescu, I. Foster, M. Wilde, Falkon: A fast and light-weight tasK execute ON framework, in: Proceedings of the 2007 ACM/IEEE Conference on Super computing, ACM, Reno, Nevada, 2007.

[16] Frederick Chong, Gianpaolo Carraro, "Architecture Strategies for Catching the Long Tail", Microsoft Corporation, April 2006. [Online] Available: http://www.msdn.microsoft.com/ enus/library/aa479069.aspx.

[17] T. Dierks, E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2”, 2008, IETF RFC 5246, [Online] Available: http://www.ietf.org/rfc/rfc5246.txt.

[18] A. Baumann, P. Barham, P.-E. Dagand, T. Harris, R. Isaacs, S. Peter, T. Roscoe, A. Schüpbach, A. Singhania, “The multikernel: a new OS architecture for scalable multicore systems”, In Proceedings of the ACM SIGOPS 22nd Symposium on OS Principles, 2009, pp. 29–43.

[19] A. Schüpbach, S. Peter, A. Baumann, T. Roscoe, P. Barham, T. Harris, R. Isaacs, “Embracing diversity in the Barrelfish manycore operating system”, In Proceedings of the Workshop on Managed Many-Core Systems (MMCS) 2008. ACM, June 2008.

[20] S. Peter, A. Schüpbach, P. Barhamy, A. Baumann,R. Isaacsy, T. Harrisy, T. Roscoe, "Design Principles for End-to-End Multicore Schedulers”, In 2nd Workshop on Hot Topics in Parallelism, Berkeley, CA, USA, June 2010.