ISSN 2320-9984
02
9 772320 998046
Volume 4 Issue 2 | June 2016
ISSN: 2320-9984 (Online)
International Journal of Modern Engineering & Management Research
Website: www.ijmemr.org

Proper Resource Utilization and Their Monitoring Using Advance Java Agent In Cloud Computing

Raj Vikram
M. Tech. Research Scholar
Vindhya Institute of Technology and Science (VITS)
Jabalpur (M.P.), [INDIA]
Email: mail2rajvikram@gmail.com

Sanjay Gupta
Associate Professor
Vindhya Institute of Technology and Science (VITS)
Jabalpur (M.P.), [INDIA]
Email: sanjiit@rediffmail.com

Abstract*—Cloud computing is terms as a new utility paradigm that's offers services dynamically to the subscribers. Cloud computing provide on demand services and it works on pay per use model. Mainly this model is converging the distribution of services like application, data and computational power from an elastic computational network of nodes. Like user PC (Personal Computers), Database and data centers, and remote servers or web server/ services. Set of such a scalable network of nodes is called cloud. And application that has been deployed or provisioned through/ from cloud is termed as a cloud application. Proper scheduling and monitoring is important in cloud computing environment. In cloud computing various methods have been proposed but they have their own advantages and disadvantages. Distributive nature of the cloud computing is the main challenge so proper scheduling of the services (mainly for SaaS) and their performance monitoring is required. Proposed advance java agent based approach presents the method for computing the total number of resources used and the solution for providing better elasticity and monitoring of the resources in the cloud which helps to gather information about the resources currently held. For implementation and evolution of the proposed advance java agent based work we use everdata cloud environment and new relic cloud performance monitoring tool.*

1. INTRODUCTION

Cloud computing is terms as a new utility paradigm that's offers services dynamically to the subscriber. Mainly this model is converging the distribution of services like application, data and computational power from an elastic computational networks of nodes. Like user PC (Personal Computers), Database and data centers, and remote servers or web server/ services. The set of such a scalable network of nodes is called cloud and an application that has been deployed or provisioned through/ from the cloud is termed as a cloud application.

Heterogeneity is the computational requirement, infrequent usages of resources and dynamic choice in modern era has main challenge for service providers (application developer and hardware manufacturer). Now computing power or connected computing power (with network) plays demanding and significant role in almost all areas including market analysis, map, searching, accounting, shopping, medical, trading and many more, the list is endless. Various computing devices and applications has been developed and developing to fulfill the users need. However different users have different requirements of application, computational power and systems software. Demand of users is heterogeneous in nature so verities of hardware & software have been developed to achieve the highest user satisfaction. So advancement of electronics and

telecommunication field has done the job. As we know the Specialization has more promising than the generalization due to it's expertise in specific job/function but it also has dark sides. Various user's require numerous specialized devices (CPU, storage etc.) and software tools. In terms of the cost and installation Purchasing or licensing of items like devices & applications is not feasible to the organization or individuals. Second thing is that most of the resources are idle i.e. frequently not used. Hence utility types of the computing paradigm will play an important role. Cloud computing is a modern computing paradigm based on utility computing model which can fulfill the user's requirement on rent basis.

Cloud computing provides very fast services which are an alternative to the conventional computing. However, this paradigm is somehow similar as, distributed computing, cluster computing, grid computing and utility computing. Cloud computing is a virtual paradigm, which is generally used for data sharing and performing computations over the scalable network of nodes. Such nodes include web services, end user computers, and data centers .These scalable networks of the nodes are called cloud or cloud nodes. Cloud application is an application which is based on such clouds or nodes. Cloud computing is the modern TCP/IP integrations of network technologies such as fast microprocessors, huge memory, high-speed network and reliable system architecture [4].Generally cloud computing services are organized into three groups:

- SaaS- Software as a Service
- PaaS-Platform as a Service and
- IaaS-Infrastructure as a Service

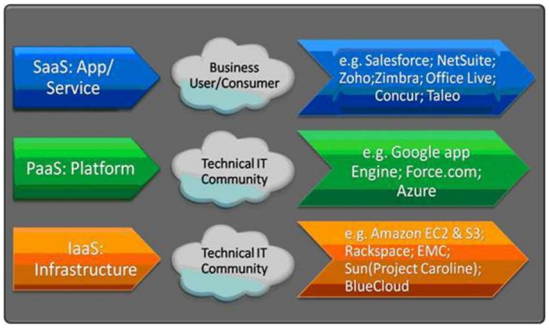

Figure 1: Architecture of cloud services

2. LITERATURE SURVEY

Cloud computing is fast growing as an alternative to conventional computing. However, the paradigm is somehow same as, utility computing, grid computing, cluster computing and distributed computing in approximately .Cloud computing fabricate a virtual paradigm for sharing data and computations over a scalable network of nodes. Examples of such nodes include end user computers, data centers, and web services. Such a scalable network of nodes is called cloud.

The work describes a distributed monitoring service, implemented in Java and JINI called MonaLISA (Monitoring Agents in A Large Integrated Services Architecture). An agent in MonaLISA represents a service (i.e., that can be used by other services or clients) that is discoverable, self-describing and able to collaborate and cooperate with other services in various monitoring tasks. Collected data is stored, per service, in a local relational database. The Data Collection Engine directs MonaLISA's function. Clients may request both real-time and historical data through use of various filtering mechanisms (e.g., predicates, Agent Filter) describe the Lattice monitoring framework, designed to be a base framework on top of which monitoring systems may be built. Though in agreement with most of the requirements we specify here, their focus is more on the actual probes for sensing low-level metrics (e.g. CPU utilization probe). Authors are more interested in the collection, aggregation, and distribution of application- and system-level metrics from third-party probes. The work introduces an

architecture for and implementation of a private cloud monitoring system. The architecture is quite high-level and is composed of three layers: an Infrastructure layer, an Integration layer and a View layer. The implementation is modular in design and consists of several components that are mostly focused on the integration layer of the architecture. Currently, it is compatible with Eucalyptus (as a IaaS implementation); however, it is mentioned that it could be extended to work with alternative IaaS implementations in the future.

T. Kanstren and Savola define a set of requirements for a distributed monitoring framework and a reference architecture that satisfies those requirements. The requirements include scalability, correctness, security, adaptation and intrusiveness. The architecture is a conceptual layered architecture and there is no reported realization of it.

An implementation of a distributed network monitoring framework was proposed.The authors showed how a three tier layered framework can be used for monitoring computer networks in geographically distributed locations. Compared with the above two approaches, our approach is better-suited to federated systems of clouds, though we share some common requirements such as scalability.

A cloud monitoring framework was proposed by Y. Sun et al.. The authors use a conceptual Service Oriented Architecture and focus mostly on message interchange among entities and on the integration of the framework with the existing system management processes. There is no implementation or evaluation of the architecture. Following works has need the next conceptual step, recognizing the importance of scalability and inter cloud monitoring on a loosely-coupled publishes-subscribe architecture. A. Lahmadi et al. present a benchmark effort for defining metrics for evaluating a performance management framework. Their metrics include overhead, delay and scalability in the context of networks and services.

3. PROPOSED WORK

Proposed Advance Java Agent based approach has provides the efficient and accurate solutions for efficient scheduling and performance monitoring in cloud computing. In the cloud computing Advance Java Agents are the self executable code work on behalf of the humans. They are capable to communicate i.e. social in nature, mobile i.e. can move in the network, perform the task at remote stations and send back the results to source station (where they been originated), one of the core property of the advance java agent is autonomy i.e. autonomous and distributive in nature. Hence advance java agent based solution has been proposed to meet the requirement of the modern cloud computing with dynamic provision (elasticity) of the cloud service provider to achieve highest scalability reliability and maximum availability of the services to the subscribers.

4. IMPLEMENTATION AND RESULTS

Snapshot of DATA SET

Fig.2 is the snapshot request/response of the deployed SaaS application developed using Java web onto the SaaS developer Cloud Platform.

Figure. 2 Jsp request page of dataset

For evaluating the performance of the proposed smart agent based system, the obtained results have been compared with base paper [2] in which author has proposed “Mobile agent based solution ” the realization of the federation of different cloud (cloud interoperability) using mobile agent. Author has choose mainly two matrices to evaluate the

performance of its mobile agent based technique–first one is average user satisfaction and second one is average utilization ratio which has been derived from the following fundamental (base) matrices –

- Number of tasks submitted
- Time required to execute the tasks
- Availability of the SaaS
- Scalability of the Saas

Author has compared there outcome with Non mobile agent based open cloud computing federation technique.

All the matrices of the performance checking has been same meaning as our proposed system like –

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

Average Utilization Ratio

Table 1 Average utilization ratio

No. of task	Proposed Agent based	MBOCCF	NMBOCCF
0	0	0	0
20	1	0.95	0.71
40	0.97	0.94	0.69
60	0.969	0.932	0.681
80	0.956	0.923	0.665
100	0.9523	0.923713	0.632389

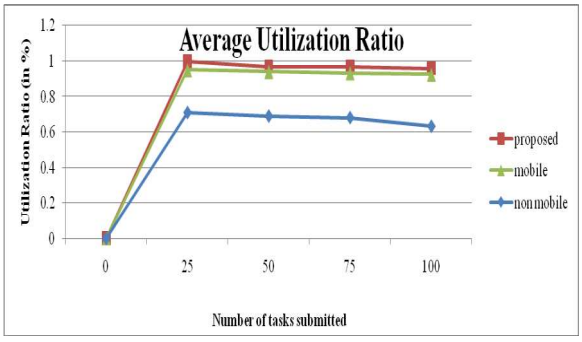

Figure. 3 Average utilization ratio

Average user satisfaction-Average user satisfaction is same as to the availability and scalability of the smart agent based system.

Table 2 shows the average user satisfaction of the deployed SaaS obtained results and it has been compared with the existing agent based technique-

Table 2 Avg. User Satisfaction Ratio

No. of task	Pro-posed Agent based	MBOCCF	NMBOCCF
0	0	0	0
20	1	0.97	0.7
40	0.98	0.97	0.69
60	0.98	0.96	0.65
80	0.97	0.95	0.631
100	0.963	0.54649	0.6171

Table 2 Avg. User Satisfaction Ratio

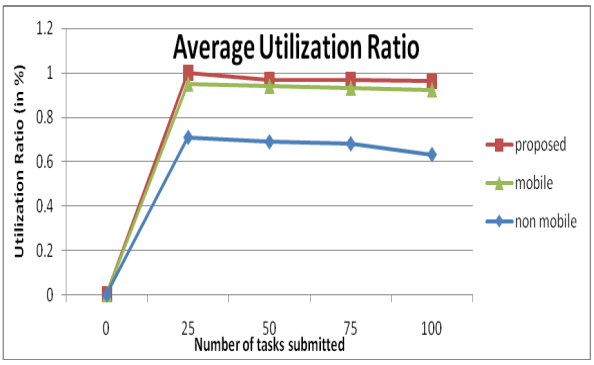

Figure 4: Average user satisfaction

5. CONCLUSION AND FUTURE WORK

This work presents the advance java agent based solution to ensure better elasticity and monitoring solution. Proposed advance java agent based solution for guaranteed better elasticity and proper monitoring of the resources in the cloud environment. For evaluation of the proposed advance java agent based method data set (jsp) has been developed using jsp web pages and deployed onto the everdata cloud environment to deploying created SaaS application in the cloud a PaaS service has been required to be subscribed, for

this Everdata PaaS service has been chosen then for monitoring and scheduling with advance java agent. New Relic service has been used for measuring the performance of advance java agent. Proposed advance 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 advance java agent based solution.

1. Security enhancement using Advance Java Agent from attacks.
2. Develop a security perimeter based on anomaly detection using APM (Application Process Management).

REFERENCES:

[1] Zehua Zhang, Xuejie Zhang, "Realization of Open Cloud Computing Federation Based on Mobile Agent", IEEE, pp.642 – 646, 2013.

[2] 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.

[3] Javier Espadas, Arturo Molina, Guillermo Jiménez, MartínMolina, Raúl Ramírez, David Concha, "A tenant-basedre source 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, ZhenghuGong, "The Characteristics of Cloud Computing", 39thInternational Conference on Parallel Processing Workshops, pp. 2010.

[7] Amit Nathani, Sanjay Chaudhary, Gaurav Soman "Policybased 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 6thInternational Conference on Computer Science & Education (ICCSE 2011) August 3-5, 2011. Super Star Virgo, Singapore2011.

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

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

[15] Frederick Chong, Gianpaolo Carraro," Architecture Strategies for Catching the Long Tail", Microsoft Corporation, April2006. [Online] Available: http://www.msdn.microsoft.com/enus/library/aa479069.aspx.

[16] 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.

[17] A. Baumann, P. Barham, P.-E. Dagand, T. Harris, R. Isaacs, S. Peter, T. Roscoe, A. Schüpbach, A. Singhania, "Themultikernel: a new OS architecture for scalable multicoresystems", In Proceedings of the ACM SIGOPS 22nd Symposium on OS Principles, 2009, pp. 29–43.

[18] A. Schüpbach, S. Peter, A. Baumann, T. Roscoe, P. Barham, T. Harris, R. Isaacs, "Embracing diversity in the Barrelfishmanycore operating system", In Proceedings of the Workshopon Managed Many-Core Systems (MMCS) 2008. ACM, June2008.

[19] 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.

[20] D. Malcolm, "The five defining characteristics of cloud computing", [Online] Available: http://www.news.zdnet.com/2100-9595_22-287001.html.

[21] P. Sharma, "What kind apps are best suited for 'Cloud deployment': 4 Solutions", [Online] Available: http://www.techpluto.com/cloud-computing-characteristics/.

[22] D. Amrhein, "Forget Defining Cloud Computing", [Online] Available: http://www.ibm.ulitzer.com/node/1018801.

[23] D. Wentzlaff, A. Agarwal, "The Case for a Factored Operating System (fos)", MIT CSAIL Technical Report, MIT-CSAILTR-2008-060, October 2008.

[24] D. Wentzlaff, A. Agarwal, "Factored Operating Systems (fos): The Case for a Scalable Operating System for Multicores", ACM SIGOPS Operating System Review (OSR), Apri