International Research Journal of Management Science & Technology



ISSN 2250 – 1959(0nline)
2348 – 9367 (Print)

An Internationally Indexed Peer Reviewed & Refereed Journal

www.IRJMST.com
www.isarasolutions.com

Published by iSaRa Solutions

Performance Analysis of Load Balancing For an Extended User Centric Cloud Computing Architecture Using Genetic Algorithm

Charu Singh

Department Of Computer Science, Govt. College Faridabad

Abstract

Cloud computing is an emerging high performance computing environment with a large scale, heterogeneous collection of autonomous systems and flexible computational architecture. It will thrive on how effectively the infrastructure is instantiated and available resources utilized dynamically. Load balancing, which is one of the main challenges in Cloud computing, distributes the dynamic workload across multiple nodes. This work analyzes load balancing strategy using Genetic Algorithm (GA) and then compares with other load balancing algorithms for an extended user centric cloud computing architecture, Hooda and Sharma [1]. The proposed algorithm improves the efficiency of data-centers and optimizes the latency by using Genetic Algorithm for load balancing. The proposed work has been simulated using the CloudSim simulator. The simulation result shows that the proposed algorithm outperformed the existing work, Hooda and Sharma [1].

Keywords: Cloud Computing; Load balancing; Genetic Algorithm; Extended user centric Cloud Computing Architecture.

1. Introduction

Cloud computing, the long-held dream of "computing as a utility", is emerging as a new paradigm of large-scale distributed computing driven by economies of scale, in which a pool of highly scalable, heterogeneous, virtualized, and configurable and reconfigurable computing resources (e.g., networks, storage, computing units, applications, data) can be rapidly provisioned and released with minimal management effort in the data centers [2-6]. Economically, the main appeal of cloud computing is that customers only use what they need, and only pay for what they actually use. Resources are available to be accessed from the cloud at any given time, and from any location via the internet [7]. Due to the exponential growth of cloud computing, it has been widely adopted by the industry and thus making a rapid expansion in availability of resources on the Internet. As the size of the cloud scales up, cloud computing service providers require handling of massive requests. The primary challenge then becomes to keep the performance same or better whenever such an outburst occurs. Thus in spite of the glorious future of Cloud Computing, many critical problems still need to be explored for its perfect realization [8]. One of these issues is load balancing. It is considered as one of the prerequisites to utilize the full resources of parallel and distributed systems. Load Balancing allows distribution of workload across one or more servers, data centers, hard drives, or other computing resources, thereby providing Cloud Service Providers (CSP) a mechanism to distribute application requests across any number of application deployments located in data centers. Load balancing mechanisms can be broadly categorized as centralized or decentralized, dynamic or static, and periodic or non-periodic. Load balancing is the key to success for cloud architectures.

In this paper Genetic Algorithm (GA) has been used as a soft computing approach, which uses the mechanism of natural selection strategy. Genetic Algorithms are a family of computational models

inspired by evolution. These algorithms encode a potential solution to a specific problem on a simple chromosome-like data structure and apply recombination operators to these structures so as to preserve critical information. Genetic algorithms are often viewed as function optimizers although the range of problems to which genetic algorithms have been applied is quite broad.

2. Related Work

Cloud architecture and computing is a potential area of research. From the point of view of some researchers, cloud computing as virtualization of previously existing data centers while some others consider data centers as backend resources. Fang et.al. [9] correlates cloud architecture with Business process management (BPM). Sharma et.al. [10] proposed cloud computing architecture that is based on the master-slave paradigm wherein in this, an intelligent and energy efficient Cloud computing architecture is proposed.

There has been little research on load balancing techniques in the cloud computing environment. Armstrong et. al. [11] uses Minimum Execution Time (MET) to assign order to each job in an arbitrary manner to the nodes on which it is expected to be executed fastest, regardless of the current load on that node. Some existing scheduling techniques like Min-Min, Round Robin and FCFS for load balancing also exist in literature. An intelligent method for load balancing has been proposed by Yang Xu et. al.[12] . It proposes a novel model to balance data distribution to improve cloud computing performance in data-intensive applications, such as distributed data mining. A few soft computing techniques like Ant Colony, Mishra and Jaiswal [13] are also reported in literature. The load balancing problems are complex and can be considered as computationally intractable problems. Such a problem cannot be formulated by linear programming hence it is quite difficult to find the globally optimal solution by using deterministic polynomial time algorithms or rules. GA, Goldberg [14] considered it one of the most widely used artificial intelligent techniques used primarily for effective search and optimization. It is a stochastic searching algorithm based on the mechanisms of natural selection and genetics. In Dasgupta et.al.[15] Genetic Algorithm (GA) has been used as a soft computing approach, which uses the mechanism of natural selection strategy.

Recently, a new architecture “extended user centric cloud computing architecture”, Hooda And Sharma [1] has been proposed. In this work the existing cloud computing architecture has been extended by dividing the data-centers into sub-data-centers according to their computing capabilities. Each sub data center provides the specific needs to specific users. Hence forth, request is diverted to the best supporting data-center. Resulting in a user’s request being executed at a more favorable destination with minimum response time. This work promises reduced latency to users of cloud computing hence, area of interest.

In this Paper GA has been applied to load balancing in extended user centric cloud computing Architecture. CloudSim Visual Modeler has been used for simulation and analysis of the algorithm. The performance of the algorithm is compared with the load balancing algorithm proposed by Hooda and Sharma [1].

3. Proposed Work

In the Extended User Centric Cloud Computing Architecture, Hooda and Sharam(2009), proposed Cloud architecture, where data centers are divided into three levels such as master data-center(MDC),slave data-centers(SDC) and sub-slave data-centers(SSDC).Master data center resides at the service provider’s premises and remaining two (SDC and SSDC) are scattered in various geographical locations, as shown in figure 1.

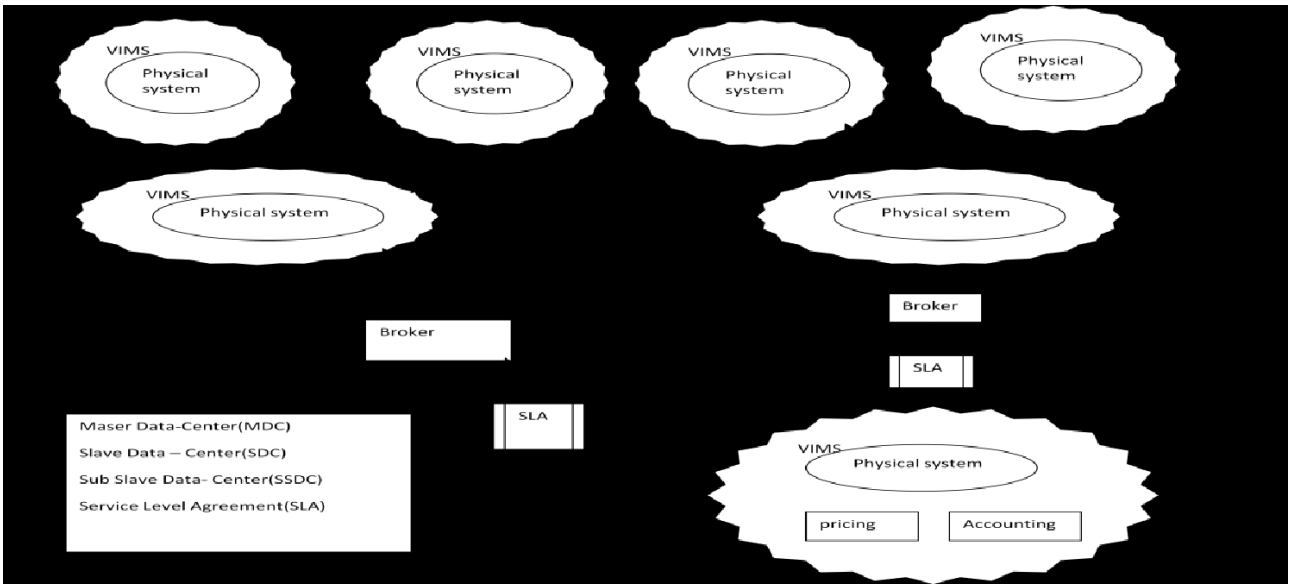


Figure 1 Cloud Architecture with Master, Slave and Sub-Slave Data Centers [1]

3.1 Algorithm Of Load Balancing By Using Genetic Algorithm Technique

Cloud computing is dynamic but at any particular instance the said problem of load balancing can be formulated as allocating *N* number of jobs submitted by cloud users to *M* number of processing units in sub slave data centers in a Cloud.

Each of the processing unit will have a processing unit vector (*PUV*) indicating current status of processing unit utilization. This vector consists of *MIPS,* indicating how many million instructions can be executed by that SSDC per second, α, cost of execution of instruction and delay cost *L*. The delay cost is an estimate of penalty, which Cloud service provider needs to pay to customer in the event of job finishing actual time being more than the deadline advertised by the master data center.

$PUV = f\,(MIPS, \alpha, L)$ (1)

Similarly, each job submitted by cloud user can be represented by a job unit vector (*JUV*). Thus the attribute of different jobs can be represented by equation 2.

$JUV = f\,(t, NIC, AT, wc)$ (2)

Where, *t* represents the type of service required by the job, Software as a Service (SAAS), Infrastructure as a Service (IAAS) and Platform-as-a-Service (PAAS). *NIC* represents the number of instructions present in the job; this is the count of instructions in the job determined by the processor. Job arrival time (*AT*) indicates the wall clock time of arrival of job in the system and worst case completion time (*wc*) is the minimum time required to complete the job by a SSDC unit.

The master data center needs to allocate these *N* jobs among the M number of SSDC such that response time function ζ as indicated in equation 3, is minimized.

$\zeta = w1.\ \alpha(NIC \div MIPS) + w2.\ L$ (3)

where *w*1 and *w*2 are predefined weights. It is very difficult to decide/optimize the weights, one criterion could be that more general the factor is, larger is the weight. Another logic is users preference or importance given to a particular Dasgupta et.al.[15] factor over the other. Here the later approach has been used and the optimization is then performed on the given set of weights. The weights are considered as w1 = 0.8 and w2 = 0.2 such that their summation is 1.

The Proposed Algorithm Is As Given Below:

Step 1: Randomly initialize a population of processing unit after encoding them into binary strings [*Start*].

Step 2: Evaluate the fitness value of each population using equation 3 [*Fitness*].

Step 3: While either maximum number of iteration are exceeded or optimum solution is found Do:

(a): Consider chromosomes with lowest fitness twice and eliminate the chromosome with highest fitness value to construct the mating pool [*Selection*].

(b): Perform single point crossover by randomly selecting the crossover point to form new offspring [*Crossover*].

(c): Mutate new offspring with a mutation probability of (0.05) [*Mutation*].

(d): Place new offspring as a new population and use this population for the next round of iteration [*Accepting*].

(e): Test for the end condition [*Test*].

Step 4: End.

3.2 Simulations and Analysis

In order to validate the effectiveness of the proposed algorithm, simulation experiments based on the platform of CloudSim were performed. CloudSim is a GUI based simulation tool. CloudSim facilitates modeling, simulation and other experimentation on cloud programmatically. It allows setting of parameters for setting a simulation environment to study any research problem of the cloud. Based on the parameters the tool computes the simulation result. A hypothetical configuration has been generated using CloudSim.

The entire geographical locations are divided into m number of area locations i.e. Area1,Area 2, ---,Area M where each area consists of a number of SSDCs (sub-slave data-centers) as shown in figure 2.

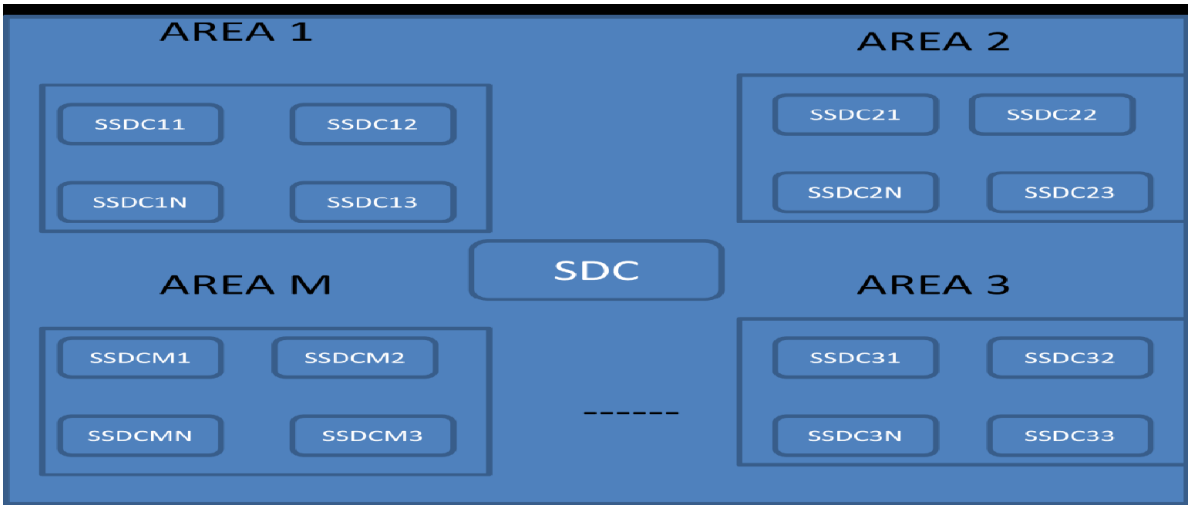


Figure 2 Create User's Instance based on Geographical Location

But for the sake of simplicity for simulation let us consider 4 areas having SSDC for all the user bases and it is assumed that there are a varied number of online registered users during peak hours, out of which only one tenth is online during the off-peak hours. Table 1 lists the details of user bases used for experimentation. Each simulated sub slave data centre host has a particular amount of virtual machines (VMs) dedicated for the application. Each of the Machines has 4 GB of RAM and

100GB of storage and each machine has 4 CPUs, and each CPU has a capacity power of 10000MIPS.

Table 1: Configuration of Simulation Environment

S.No.	User Base Region	Online users during Peak hours	Online users during off peak hours
1	Area 1	400000	4000
2	Area 2	80000	8000
3	Area 3	30000	4500
4	Area 4	15000	2000

3.3 Simulation Setup

For experimentation all user requests are processed by a single SDC having 25, 50 and 75 VMs of Cloud Configuration (CCs) in terms of SSDC allocated to the application. This simulation setup is described in table 2 with calculated overall average Response Time (*RT*) in *ms* for GA and for the referred work. A comparative performance analysis graph of the same is depicted in figure 3.

Table 2: Simulation scenario with calculated overall average response time (RT) in (ms)

S.No.	Cloud Configuration	SSDC Configuration	RT Using GA	RT without GA
1	CC1	CC1 Each with 25 VMs	349.56	368.76
2	CC2	CC1 Each with 50 VMs	315.88	359.05
3	CC3	CC1 Each with 75 VMs	302.45	348.22

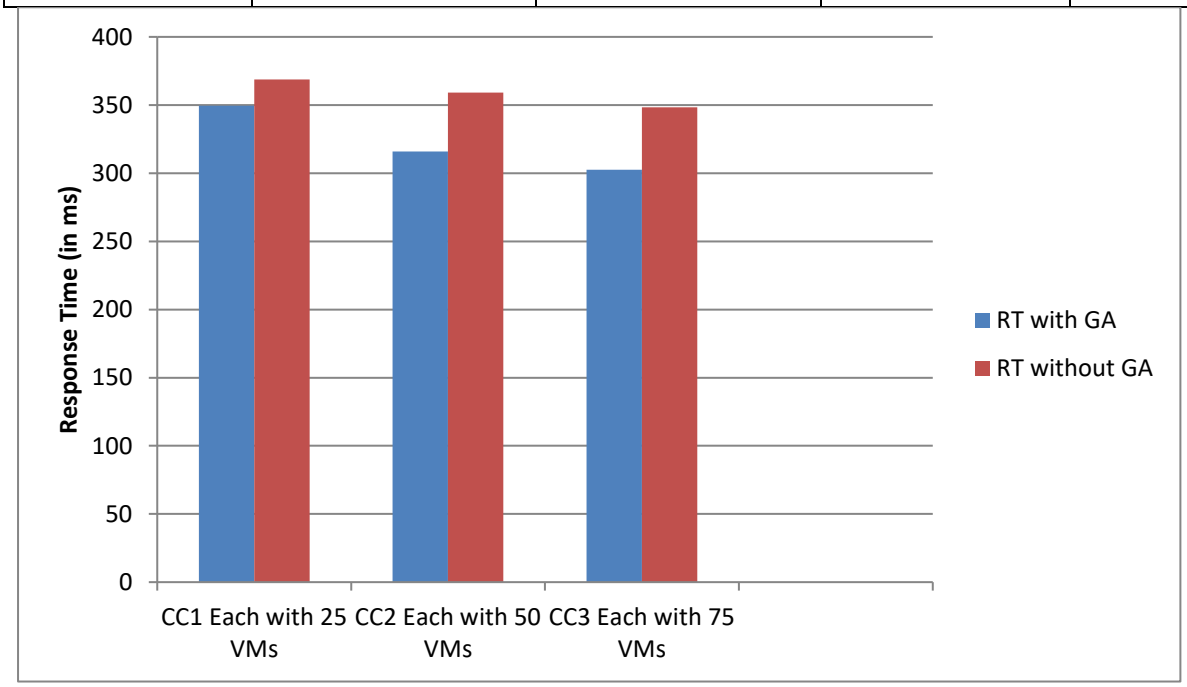


Figure 3: Comparative Performance analysis of proposed GA with Referred work [1]

4. Conclusion

In this paper, performance of a genetic algorithm based load balancing strategy for Extended User Centric Cloud Computing Architecture has been analyzed to provide an efficient utilization of resources in the cloud environment. Analysis of the result indicates that the proposed strategy for load balancing outperforms techniques of load balancing without GA and guarantees the QoS requirement of customer jobs. Here it has been assumed that all the jobs are of the same priority which may not be the actual case, this can be accommodated in the JUV and subsequently taken care of in fitness function. Also a very simple approach of GA has been used however variation of the crossover and selection strategies could be applied as a future work for getting more efficient and tuned results.

5. References

1. Susheela Hooda, A.K.Sharma, "An Extended User Centric Cloud Computing Architecture", *International Journal of Science and Research (IJSR), ISSN (Online): 2319-7064, Volume 3 Issue 6, 939-942, June 2014.*
2. Armbrust M, Fox A, Griffith R, Joseph A D, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A and Stoica I, "A view of cloud computing", Communications of the ACM, Vol. 53, No. 4, 2010, pp. 50-58.
3. Iosup, A., Ostermann, S., Yigitbasi, M.N., Prodan, R., Fahringer, T. and Epema, D.H.J, "Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing", IEEE Transactions on Parallel and Distributed Systems, Vol. 22, No. 6, 2011, pp. 931-945.
4. Arunadevi.M and R.S.D Wahidabanub, "Design of Power Efficient Schema for Energy Optimization in Data Center With Massive Task Execution Using DVFS", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 2, 2012, pp. 407-414.
5. Almutairi, A., Sarfraz M., Basalamah S., Aref W. and Ghafoor A, "A Distributed Access Control Architecture for Cloud Computing", IEEE Software Vol. 29, No. 2, 2012, pp. 36-44.
6. Junaid Qayyum, Faheem Khan, Muhammad LaL, Fayyaz Gul, Muhammad Sohaib and Fahad Masood, "Implementing and Managing framework for PaaS in Cloud Computing", IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 5, No. 3, 2011, pp. 474-479.
7. Sanjeev Narayan Bal, "Clouds for Different Services", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No 1, 2012, pp. 273-277.
8. A. Vouk, "Cloud computing- issues, research and implementations", in Proc. of *Information Technology Interfaces*, pp. 31-40, 2008.
9. Zhenyu Fang and Changqing Yin, "BPM Architecture Design Based on Cloud Computing," *Online Journal on Intelligent Information Management*, Vol. 2, May 2010, pp 329-333.
10. Rajkumar Sharma and Priyesh Kanungo "Vikram University", Ujjain, India rksujn@rediffmail.com," Patel College of Sc. & Technology", Indore, India, sept 2011.
11. T. R. Armstrong, D. Hensgen, "The relative performance of various mapping algorithms is independent of sizable variances in runtime predictions", in Proc. of *7th IEEE Heterogeneous Computing Workshop (HCW 98),* pp. 79-87,1998.
12. Yang Xu, Lei Wu, Liying Guo, Zheng Chen,Lai Yang, Zhongzhi Shi, "An Intelligent Load Balancing Algorithm Towards Efficient Cloud Computing", in Proc. of *AI for Data Center*

Management and Cloud Computing: Papers, from the 2011 AAAI Workshop (WS-11-08), pp.27–32, 2008.

13. Ratan Mishra and Anant Jaiswal, "Ant colony Optimization: A Solution of Load balancing in Cloud",in *International Journal of Web & Semantic Technology (IJWesT)*, Vol.3, No.2, pp. 33–50, 2012.
14. D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*,Addison-Wesley, 1989.
15. Kousik Dasgupta, Brototi Mandal, Paramartha Dutta, Jyotsna Kumar Mondal, Santanu Dam, "A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing", in *International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA), Procedia Technology 10 (2013), 340 – 347, 2013.*

ISARA INSTITUTE OF MANAGEMENT & PROFESSIONAL STUDIES

WWW.IIMPS.IN

EARN YOUR

MBA



Accreditation & Ranking

UGC / NCTE Approved.

INFO@IIMPS.IN

011-41005174

RESEARCH GATEWAY

STOP PLAGIARISM

1 Submit your content in Word File.

2 Get report in 48 hrs.

3 *Missing content or references will be fixed.

4 Citation for your work.

5 Get accurate user friendly report.

researchgateway.in | info@researchgateway.in

+91-9205579779

AROGYAM ONLINE

Arogyam Ayurveda

Holistic Healing through herbs

100% NATURAL

PARIVARTAN PSYCHOLOGY CENTER

COLOR PSYCHOLOGY : HOW COLOR AFFECT YOUR CHILD

BLUE — Calms your Child's Mind & Body

YELLOW — Promotes Concentration, Stimulates the Memory

PINK — Evokes Empathy, makes your Child Calm

RED — Excites and energizes your Child's body

GREEN — Improves Reading speed and Comprehension

www.parivartan4u.com

Confuse about your children's future?

भारतीय भाषा, शिक्षा, साहित्य एवं शोध

ISSN 2321 – 9726

WWW.BHARTIYASHODH.COM

INTERNATIONAL RESEARCH JOURNAL OF MANAGEMENT SCIENCE & TECHNOLOGY

ISSN – 2250 – 1959 (0) 2348 – 9367 (P)

WWW.IRJMST.COM

INTERNATIONAL RESEARCH JOURNAL OF COMMERCE, ARTS AND SCIENCE

ISSN 2319 – 9202

WWW.CASIRJ.COM

INTERNATIONAL RESEARCH JOURNAL OF MANAGEMENT SOCIOLOGY & HUMANITIES

ISSN 2277 – 9809 (0) 2348 - 9359 (P)

WWW.IRJMSH.COM

INTERNATIONAL RESEARCH JOURNAL OF SCIENCE ENGINEERING AND TECHNOLOGY

ISSN 2454-3195 (online)

WWW.RJSET.COM

INTEGRATED RESEARCH JOURNAL OF MANAGEMENT, SCIENCE AND INNOVATION

ISSN 2582-5445

WWW.IRJMSI.COM

JOURNAL OF LEGAL STUDIES,POLITICS AND ECONOMICS RESEARCH

WWW.JLPER.COM