

Enhanced Load Balancing Architecture using EE-GA

Jaskirat Singh
Research Fellow
Punjabi University, Patiala

Brahmaleen Kaur Sidhu
Asst. Professor
Punjabi University, Patiala

ABSTRACT

Cloud computing is the long dreamed vision of computing as a utility, where data owners can remotely store their data in the cloud to enjoy on-demand high-quality applications and services from a shared pool of configurable computing resources. In the meantime, the cloud environment represents various difficulties. Two players in distributed computing situations, cloud suppliers and cloud clients, seek after diverse objectives; suppliers need to amplify income by accomplishing high asset usage, while clients need to minimize costs while meeting their execution prerequisites. Nonetheless, it is hard to allot resources in a commonly ideal manner because of the absence of data sharing between them. In addition, continually expanding heterogeneity and variability of the surroundings poses considerably harder difficulties for both sides. This paper describes the work which mainly aimed at enhancing the load balancing architecture where firstly genetic algorithm is been implemented with simple architecture [1]. Secondly, genetic algorithm is been implemented with enhanced architecture named as E-GA where job grouping is done according to job's requirements. Finally the whole architecture is been enhanced by using job grouping method with enhanced genetic algorithm named as EE-GA. In enhanced genetic algorithm, artificial bee colony algorithm uses the output given by genetic algorithm as their input and provides efficient resources. Both E-GA and EE-GA have been successful in better resource utilization so that the jobs are handled in a more efficient manner and also time is saved [3]. All the comparison results prove that the EE-GA provides a more efficient way as compared to the others.

General Terms

Load Balancing, Cloud Computing, Genetic Algorithm, Job Grouping.

Keywords

Load Balancing, Enhanced Cloud Architecture, GA, Resource Utilization.

1. INTRODUCTION

Cloud computing is a long time ago envisioned vision of registering as an utility, where information proprietors can remotely store their information in the cloud to appreciate on-interest top notch applications and administrations from a common pool of configurable figuring assets. Cloud is another plan of action wrapped around new advances, for example, server virtualization that exploit economies of scale and multi-occupancy to decrease the expense of utilizing data innovation assets [8]. It likewise conveys new and testing security dangers to the outsourced information. Since cloud administration suppliers (CSP) are particular regulatory substances, information outsourcing really gives up the proprietor's definitive control over the destiny of their information.

A cloud computing and stockpiling arrangement furnishes clients and endeavors with different abilities to store and procedure their information in outsider information centers. It depends on sharing of assets to accomplish soundness and economies of scale, like an utility (like the power matrix) more than a system [6]. At the establishment of distributed computing is the more extensive idea of merged foundation and shared administrations.

Cloud computing, or in less difficult shorthand simply "the cloud", additionally concentrates on amplifying the adequacy of the common assets. Cloud assets are typically shared by numerous clients as well as alertly reallocated per request. This can work for apportioning assets to clients.

Load Balancing is a strategy to pass on workload more than one or more servers, framework interfaces, hard drives, or other enrolling assets. Normal server ranch utilization rely on upon broad, compelling (and indulgent) figuring gear and framework establishment, which are obligated to the common risks joined with any physical device, including hardware dissatisfaction, power and/or framework interruptions, and resource imperatives in times of claim [22]. Weight modifying in cloud differs from conventional thinking on weight altering auxiliary arranging and use by using item servers to perform the stack changing. This obliges new open entryways and economies-of-scale, and moreover showing its own specific uncommon game plan of troubles. Weight modifying is used to confirm that none of existing resources are unmoving while others are being utilized. To change load allocation, migrate the pile from the source center points (which have surplus workload) to the correspondingly delicately stacked destination centers. Right when apply load changing in the midst of runtime, it is called component weight conforming — this can be recognized both in a prompt or iterative route as showed by the execution center decision:

- In the iterative systems, the last destination center is determined through a couple of accentuation steps.
- In the quick procedures, the last destination center point is picked in one stage.

Besides, another kind of Load Balancing framework can be used i.e. the Randomized Hydrodynamic Load Balancing framework, a creamer procedure that adventures both prompt and iterative system [15].

2. EXISTING LOAD BALANCING TECHNIQUES

There are different systems to adjust the heap of distributed computing. Some of which are examined below:

- Honey Bee Foraging Algorithm: This entire algorithm is taking into account the procedure of bumble bees discovering the sustenance and disturbing others to go and eat the nourishment. To start with forager honey bees go and discover their nourishment. Subsequent to returning to their individual beehouse, they move. Subsequent to

seeing the quality of their move, the scout honey bees take after the forager honey bees and get the nourishment. The more fiery the move is, the more nourishment accessible is. So this entire procedure is mapped to over-burden or under stacked virtual servers [13]. The server forms the solicitations of the customers which is like the sustenance of the honey bees. As the server gets substantial or is over-burden, the honey bees hunt down another area i.e. customer is moved to some other virtual server. Thusly, this entire system meets expectations.

- Task Scheduling Algorithm in view of Load Balancing: Y. Tooth et al, talked about a two-level errand booking instrument taking into account burden adjusting to meet element prerequisites of clients and get high asset usage. It accomplishes burden adjusting by first mapping errands to virtual machines and afterward virtual machines to host assets.
- Throttled Load Balancing Algorithm: This calculation makes utilization of personality of virtual machines. Customer asks for the ID of virtual machine [17]. Throttled burden adjusting calculation gives back that ID to the client.
- Ant Colony Optimization Technique: In this procedure, a pheromone table was being outlined which was redesigned by ants according to the asset usage and hub choice formulae. Ants move in forward heading looking for the over-burden or under stacked hub. As the over-burden hub is crossed, then ants move back to fill the as of late experienced under stacked hub, so a solitary table is overhauled unflinching
- Role Based Access Control (RBAC): RBAC is a procedure used to diminish the heap of the cloud. In this, a part is doled out to every client so that their particular number of clients can get to restricted utilizations of the cloud. So by this approach, the assets are confined to the clients.
- Resource Allocation Scheduling Algorithm (RASA): In this calculation, virtual hubs are made first. At that point the normal reaction time of each virtual hub is found. At that point as indicated by the minimum stacked hub criteria, proficient virtual hub is discovered and ID of that hub is come back to the customer [23]. In this, Min-Min and Max-Min systems are taken after. In the event that number of assets accessible are odd, then Min-Min methodology is connected else Max-Min procedure is connected.

3. PROPOSED WORK

The proposed work is to implement enhanced load balancing architecture by using .NET Environment and also to implement a new Enhanced GA with ABC based load balancing mechanism to integrate it with grouping & priority of jobs. The basic design of the proposed system is given in figure 1.

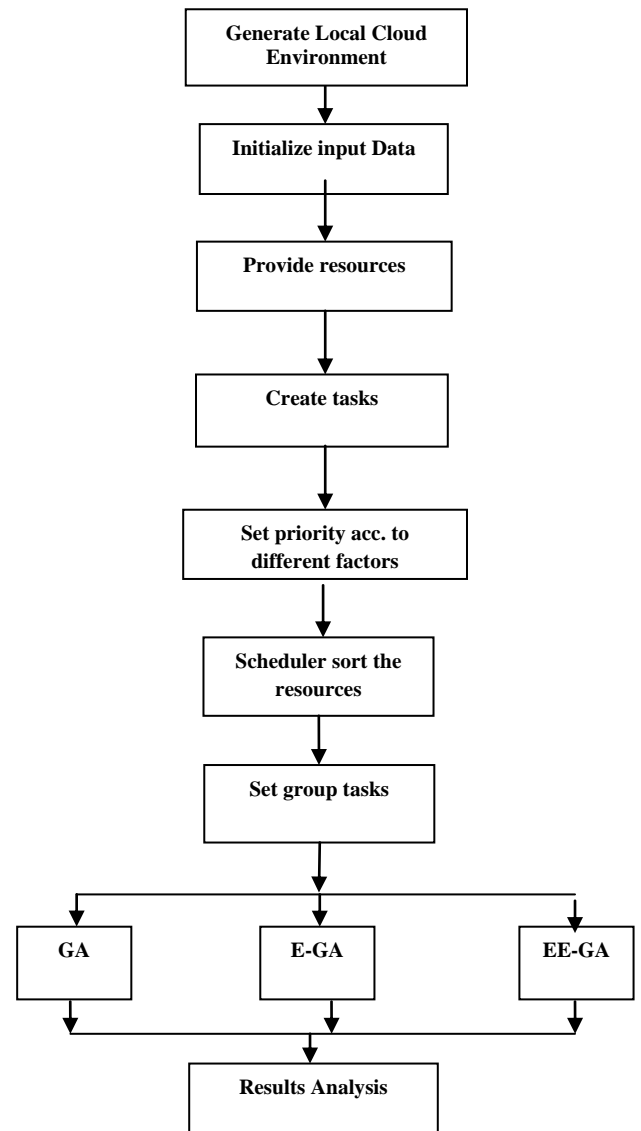


Fig 1: Basic Design of the System

In this a local cloud environment is generated using .Net technology. A client-server architecture is been created. In which client assign different types of jobs and send to the server for execution. Once the server receives the jobs it initializes the jobs and also initializes the resources available. Once the initialization is done it creates the tasks then priority is been assigned according to different factors. Now the scheduler sorts the resources according to their capability. Then the available jobs are set into groups and the execution is done by different algorithm i.e. GA, E-GA, and EE-GA. Finally the results are analyzed according to the time taken for the execution of the jobs.

3.1 Generate Cloud Environment

In this work using .Net technology a local cloud environment is generated for execution of different Tasks. A user has to have login account for accessing this cloud environment so that user can assign task to the server. In figure 2 the generated cloud environment can be shown.

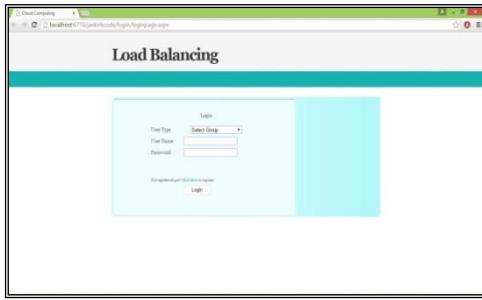


Fig 2: Local Cloud Environment

In this, Firstly select the group from which user belongs to and then user can login after authentication. Only Admin can see the full environment. It means only admin have the full access to see all the assigned jobs and to execute them, all existing resources and users. Figure 3 shows the Admin panel after login.



Fig 3: Admin Login Panel

In this admin can manage the resources. Checks the available jobs which are assigned by different users. Admin can assign its own jobs also. It has all the details of all the assigned jobs and executed jobs.

3.2 Initialize Input Data

In this user can submit their job by adding job type and also have option to add description about their job which will state some properties related to the job. Submitted jobs are added to the queue in the cloud system which will be later executed by the server. Figure 4 shows how jobs are been added.



Fig 4: Submitting Jobs

3.3 Initialization of Resources

In this the available resources are initialized by the cloud system which can be used for the execution of the jobs. It shows the list of all the resources with their status that they are free or busy. Figure 5 shows the list of some available resources.

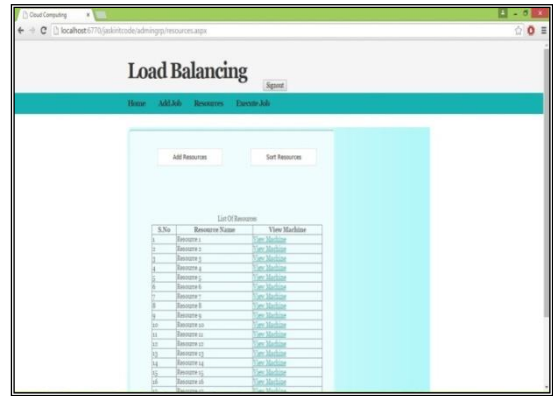


Fig 5: List of resources

3.4 Creation of Tasks

This defines that how jobs are set to the resources which are assigned by different users. Once the jobs are assigned there are different ways for execution of jobs by the admin. Figure 6 shows the different methods for execution of the jobs.

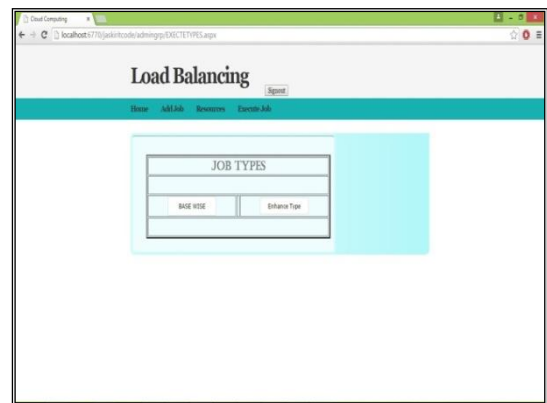


Fig 6: Execution Options

The admin chooses one of the options for performing execution of the jobs. These methods are based on different algorithms. Figure 7 shows the EE-GA based execution of data.

In this the available resources are assigned to the jobs once the resources are assigned the tasks are executed. If different resources are selected, jobs are executed in parallel way and if two same resources comes up it is executed in serial.

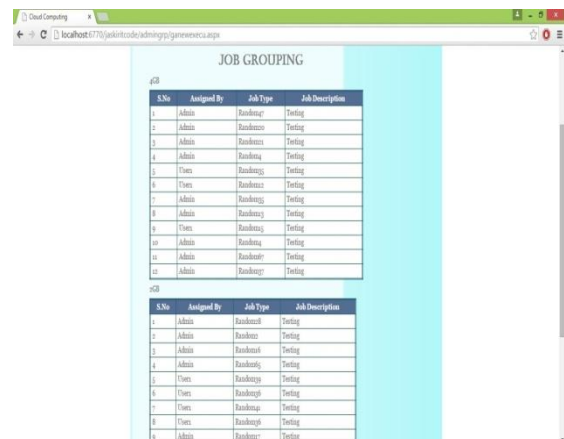


Fig.7: EE-GA based Resource Mapping

4. ALGORITHM LEVEL DESIGN

In this model on the one end resources are initialized. While initializing resources it gets R_id, R-name, R-RAM, and R_Temp. According to their temperature virtual machines will be created. According to the temperature of resource the virtual machines are created. If R_Temp<5 then 5 VM's will be created on one resource but if 10<R_Temp<5 then 3 VM's will be created but if R_temp>10 then 1 VM will be created on one resource. The whole information of the submitted jobs is saved i.e. J_id, J-name, J-desc, J-inst and Job_type. In this work job will be grouped according to their requirement type means if jobs are CPU-intensive jobs it will be in one group and if jobs are Memory-Intensive then it will be in other group. Scheduler checks the resource availability and maps the jobs according to their requirement with the intelligence of GA & EE-GA (hybrid of GA & ABC) algorithm.

The algorithm level design of the system is defined below and algorithm design of the selection method can be defined individually in the subsections.

```

Step 1: set usr_name = user name;
Set passwd = password;
Step 2: If ((select * from usertable where username=usr_name
and password=passwd) == true)
{
Set job = type_of_job
Set job_des= description about job
Submit & save into database.
}
Else
{
An Error message can be displayed ('User can't be authorized
by the system')
}
Step 3: if (usr_name == admin_usr)
{
Select method for execution: 'GA' 'E-GA' 'EE-GA'
Execute job by using above selected method.
}
Else
{
A message can be displayed ('User can only have a
authorization to submit a job')
}
Step 4: if (Jobs_execution == done)
{
Results saved in database.
}
Else
{
A message can be displayed ('check the selected resources for
execution')
}

```

5. RESULTS AND DISCUSSIONS

To analyze the performance of this new enhanced architecture, a comparative analysis can be done by using different methods which includes both GA, E-GA & EE-GA on local client/server environment. The comparative analysis of all these above method can be shown in a graphical manner on the basis of different parameters named as:

- Time
- Resource Utilization

5.1 Time

Scalability of the framework can be measured by calculating time of each process. Here Time can be defined as a time taken to execute job by the server. Total time can be calculated as:

$$Time = \frac{No. of instructions (job (i))}{MIPS rating (Machine (i))}$$

Table 1 gives the total time to execute 40 and 80 jobs on the cloud server with GA, E-GA & EE-GA.

Table 1. Time to Execute 40 & 80 jobs

| | GA | E-GA | EE-GA |
|---------|--------|--------|--------|
| 40 jobs | 117.29 | 81.67 | 76.68 |
| 80 jobs | 206.18 | 149.69 | 139.82 |

The above results shows that EE-GA takes less time as compare to GA and E-GA to execute 40 and 80 jobs respectively. Figure 8 the graphical representation of the results.

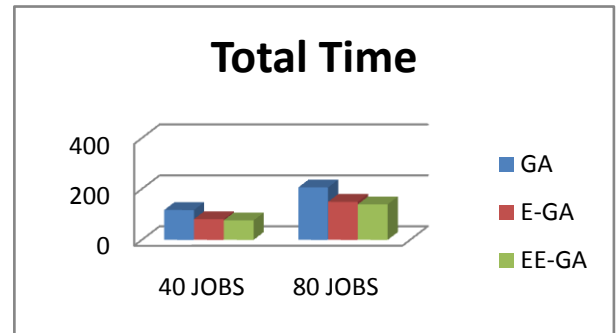


Fig 8: Total Time for Execution

5.2 Resource Utilization

To evaluate the performance of load balancing system, resource utilization is calculated which shows whether the load is balanced or not. Resource utilization gives the ratio of the over utilized, less utilized and proper utilize resources. Table 2 gives the resource utilization rate to execute 40 and 80 jobs on the cloud server with GA, E-GA & EE-GA.

Table 2. Resource Utilization for 40 & 80 jobs

| | 40 JOBS | | | 80 JOBS | | |
|---------------------------|---------|------|-------|---------|------|-------|
| | GA | E-GA | EE-GA | GA | E-GA | EE-GA |
| Over utilization | 20 | 17.5 | 15 | 22.5 | 20 | 12.5 |
| Less Utilization | 55 | 47.5 | 37.5 | 27.5 | 20 | 10 |
| Proper Utilization | 25 | 35 | 47.5 | 50 | 60 | 77.5 |

The above results shows that in EE-GA more number of resources are properly utilized as compare to GA and E-GA to execute 40 and 80 jobs respectively. Figure 9 the graphical representation of the results.

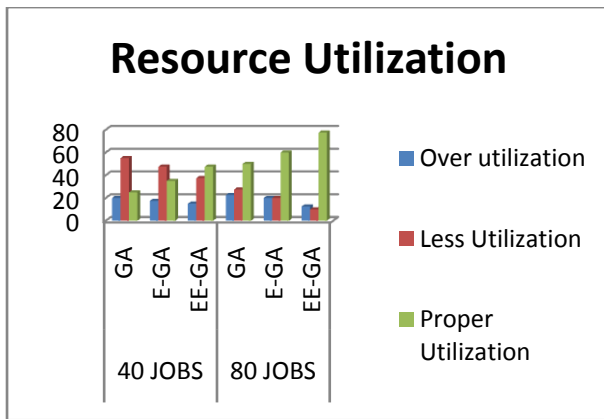


Fig 9: Resource Utilization

6. CONCLUSION & FUTURE SCOPE

There has been a sensational increment in the prevalence of cloud computing systems that lease processing assets on-interest, bill on a pay-as-you-go premise, and multiplex numerous clients on the same physical foundation. These cloud computing systems give a fantasy of unending computing resource to cloud clients so they can build or reduce their asset utilization rate as indicated by the requests. In this work, the enhanced genetic algorithm (both E-GA and EE-GA) have been successful in better resource utilization so that the jobs are handled in a more efficient manner and also time is saved. All the comparison results prove that the enhanced GA provides a more efficient way as compared to the genetic algorithm. In future, the present work may be extended to developing an algorithm that is more efficient that can handle extremely large amount of jobs in a more synchronized and reliable manner.

7. REFERENCES

- [1] David C. Wyld, "Moving to the cloud: An Introduction to Cloud Computing in Government", IBM centre for The Business of Government e-Government Series, 2009. Mell, P. and Grance, T. (2010), "The NIST Definition of Cloud Computing", NIST.
- [2] Rajkumar Buyya and Karthik Sukumar, "Platforms for Building and Deploying Applications for Cloud Computing", CSI Communications | May 2011, pp.6-11.
- [3] X. Wang, B. Wang and J. Huang, "Cloud Computing and its Key Techniques", in Proc. Of IEEE International Conference, pp.404-410, June 10-12, 2011
- [4] N. Carr and F.Y. Yu, "IT is no longer important: the Internet great change of the high ground Cloud Computing", CITIC Publishing House, October 2008.
- [5] J.Srinivas, K.Venkata Subba Reddy, Dr.A.Moiz Qyser, "Cloud Computing Basics", International Journal of Advanced Research in Computer and Communication Engineering Vol. 1, Issue 5, July 2012
- [6] Qi Zhang · Lu Cheng · Raouf Boutaba, "Cloud computing: state-of-the-art and research challenges", J Internet Serv Appl (2010) 1: pp.7–18
- [7] Dimpi Rani, "A Comparative Study of SaaS, PaaS and IaaS in Cloud Computing", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 6, June 2014, pp. 458-461
- [8] Pooja, Asmita Pandey, "Cloud Computing – An on Demand Service Platform", International Conference on Advances in Management and Technology (iCAMP – 2013), pp. 5-9
- [9] Introduction to Cloud Computing architecture, White Paper, 1st Edition, June 2009, Sun Microsystems, Inc.
- [10] J. Yu and R. Buyya, "A Taxonomy of Workflow Management Systems for Grid Computing, Journal of Grid Computing", vol. 34, no.3, pp.171-200, September 2005.
- [11] R.P. Padhy, "Load Balancing in Cloud Computing Systems", B.Tech Thesis, NIT, Rourkela, Computer Science and Engineering Department, Rourkela, 2011.
- [12] R. Buyya, J. Broberg and A. M. Goscinski, "Cloud Computing: Principles and Paradigms", pp. 664, February, 2011. ISBN: 978-0-470-88799-8.
- [13] S. Haider et.al., "Fault Tolerance in Distributed Paradigms", in Proc. of Fifth International Conference on Computer Communication and Management, IACSIT Press, Singapore, 2011.
- [14] Yogita Chawla and Mansi Bhonsle, "A Study on Scheduling Methods in Cloud Computing", International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 1, Issue 3, September – October 2012, pp.12-17
- [15] Harpreet Kaur, Amritpal Kaur, "A Survey on Fault Tolerance Techniques in Cloud Computing" in the International Journal of Science, Engineering and Technology (IJSET) (ISSN: 2348-4098), Volume 3 Issue 2 (April 2015), Pages Number – 411-415.
- [16] T. A. Dumitras and P. Narsimhan, "Fault-Tolerant Middleware and the Magical 1%", ACM/IFIP/USENIX Conference on Middleware, Grenoble, France, November-December 2005.
- [17] Vishonika Kaushal, Anju Bala, "Autonomic fault tolerance using haproxy in cloud environment," International Journal of Advanced Engineering Sciences and Technologies, vol. 7, 2010.
- [18] Zaipeng Xie, Hongyu Sun and Kewal Saluja, "A Survey of Software Fault Tolerance Techniques". 2004
- [19] R. Buyya and M. Murshed, "GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing. Concurrency and Computation: Practice and Experience, Wiley Press, 14(13- 15), Nov.-Dec., 2002.
- [20] Sheheryar Malik, Fabrice Huet, "Adaptive Fault Tolerance in Real Time Cloud Computing", IEEE World Congress on Services, 2011.
- [21] Alain Tchana, Laurent Broto, Daniel Hagimont, "Fault Tolerant Approaches in Cloud Computing Infrastructures", The Eighth International Conference on Autonomic and Autonomous Systems, ICAS, 2012.
- [22] Anju Bala, Inderveer Chana, "Fault Tolerance-Challenges, Techniques and Implementation in Cloud Computing", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 1, January 2012.
- [23] T. C. Bressoud and F. B. "Schneider. Hypervisor-Based Fault Tolerance". In Proceedings of the 15th ACM Symposium on Operating Systems Principles (SOSP 1995), pages 1–11, Dec. 1995.

- [24] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, " Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility. Future Generation Computer Systems", 25(6): Elsevier Science, Amsterdam, the Netherlands, pages 599-616, June 2009.
- [25] Dilbag Singh, Jaswinder Singh, Amit Chhabra, " Evaluating Overheads of Integrated Multilevel Checkpointing Algorithms in Cloud Computing Environment", I. J. Computer Network and Information Security, 2012, 5, pp. 29-38
- [26] Ekpe Okorafor, "A Fault-tolerant High Performance Cloud Strategy for Scientific Computing", 2011 IEEE International Parallel & Distributed Processing Symposium, pp.1525-1532
- [27] Kassian Plankensteiner, Radu Prodan, Thomas Fahringer, "A New Fault Tolerance Heuristic for Scientific Workflows in Highly Distributed Environments based on Resubmission Impact", 2009 Fifth IEEE International Conference on e-Science, pp.313-320
- [28] Haimantee Mahato, Anjali Munjal, Shreya Chinchalikar, " A System for Task Scheduling and Task Migration in Cloud Environment", IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p- ISSN: 2278-8727 Volume 16, Issue 2, Ver. IX (Mar-Apr. 2014), PP 115-118
- [29] Engelmann, G. R. Vall'ee, T. Naughton, and S. L.Scott. Proactive fault tolerance using preemptive migration. In Euromicro International Conference on Parallel, Distributed, and network-based Processing (PDP), pages 252–257, 2009.
- [30] Y. Huang, C. Kintala, N. Kolettis, and N. Fulton, "Software rejuvenation: Analysis, module and applications," in Proceedings of Fault-Tolerant Computing Symposium FTCS-25, June 1995
- [31] Ravi Jhavar, Vincenzo Piuri and Marco Santambrogio, Member of IEEE, "Fault Tolerance Management in Cloud Computing: A System-Level Perspective ", IEEE, 2012.
- [32] Anjali D. Meshram, A.S.Sambare and S. D. Zade, "Fault Tolerance Model for Reliable Cloud Computing ", International Journal on Recent and Innovation Trends in Computing and Communication, ISSN 2321 – 8169, Vol. 1, Issue: 7, July 2013.
- [33] Prasenjit Kumar Patra, Harshpreet Singh & Gurpreet Singh, "Fault Tolerance Techniques and Comparative Implementation in Cloud Computing" International Journal of Computer Applications, Vol. 64, No.14, February 2013.
- [34] A. Tchana and L. Broto and D. Hagimont, "Approaches to Cloud Computing Fault Tolerance," Computer, Information and Telecommunication Systems (CITS), pp 1-6, 2012.
- [35] Yue Gao, Sandeep K. Gupta, Yanzhi Wang, Massoud Pedram, "An Energy-Aware Fault Tolerant Scheduling Framework for Soft Error Resilient Cloud Computing Systems", 978-3-9815370-2-4/DATE14/©2014 EDAA IEEE
- [36] Jing Liu, Xing-Guo Luo, Xing-Ming Zhang, Fan Zhang and Bai-Nan Li, "Job Scheduling Model for Cloud Computing Based on Multi-Objective Genetic Algorithm", IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 1, No 3, January 2013, pp134-139
- [37] Giuseppe Portaluri, Stefano Giordano, Dzmityr Kliazovich, Bernab'e Dorransoro, "A Power Efficient Genetic Algorithm for Resource Allocation in Cloud Computing Data Centers", 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet), pp 58-63