iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 2 Issue: III Month of publication: March 2014

DOI:

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

Efficient Utilization of Nodes in a Gaming Environment by Greedy Strategies in Cloud Computing

Rajesh George Rajan[1], V.Jeyakrishnan[2]

M.Tech. Student, Dept of Computer Science, Karunya University, Coimbatore, India[1]
Assistant Professor, Dept of Computer Science, Karunya University, Coimbatore, India[2]

Abstract- Cloud computing can be offers dynamic and a better resources are to be given to the users according to their demand basis. In the existing environments of cloud computing the load distribution between the different virtual machines and virtual servers are becoming a challenging task. By using the gaming environments, the efficient utilization of nodes are to be done. In this paper the experimental result shows the efficient load distribution between the nodes and the better utilization of the nodes in the environments.

Keywords- Load Distribution, Greedy Heuristics, Greedy Heuristic with State, Positional Action Manager, Cloud Computing.

1. INTRODUCTION

Cloud computing is on demand service by which can be provide various types of services to our societies. There were various types of services that should be provided by the cloud computing they are platform as a service, infrastructure as a service and software as a service. Cloud computing was intended to enable computing across widespread diverse resources rather than on local machines or at remote service farms. Although there is no proper definition for cloud computing. Load balancing was identified as a major concern to allow cloud computing to scale up to increasing demand. Load balancing is the process of reassigning the total loads to the individual nodes of the collective system to make the best response time and also good utilization of the resources

Load distribution is the important method that can be helps for the better distribution of the load throughout the system. According to the load distribution, when various requests are coming from different clients, then if the server cannot withstand with the request, it will cause the problem so that the load becomes distributed over the cluster. In the distributed virtual environments, the massively multiplayer games are to be used for the distributed simulations. Mostly the distributed virtual environments rely on a centralized architecture that supports various users' functionalities .like synchronization, users login etc. When the users level is increasing simultaneously there were show some scalability limitation. As a result cluster based centralized architecture becomes maintained.

Load balancing was identified as a major concern to allow cloud computing to scale up to increasing demands. A distributed solution is required, as it is not practical or cost efficient in many cases to maintain idle service/hardware provision merely to keep up with all identified demands. Equally, when dealing with such complexity, it is impossible to fully detail all system future states. Therefore, it is necessary to allow local reasoning through distributed algorithms on the current system state.

Cloud Computing [1] allows us to solve the aforementioned scalability and hardware ownership problems because of on demand resource provisioning [2, 3]. The possibility of renting machines lifts the DVE operators from the burden of buying and maintaining hardware, whereas it offers the illusion of infinite machines, with good effects on scalability. Also, the pay-per-use model adheres to the seasonal access pattern of the DVE (e.g. more users in weekends than in the middle of the week). However, Cloud Computing may still be

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

costly for platform operators. Besides server time, bandwidth cost represents a major expense when operating a DVE [4]. When this cloud approaches are to be very feasible but its cost must be too higher for the distributed virtual environments. So that another concept of infrastructure for the distributed virtual environments is considered that is peer-to-peer concept. So various advantages are to be there for the peer system that is the network is able to self repair, robustness and also the major thing is the low cost that can be affordable to the organization. So these are the two orthogonal approaches that are to be combined.

According to the integration of the two different environments then the execution of the corresponding system becoming a challenging one. In the distributed virtual environments the advantages of these two methods are to be combined. In this paper the distributed virtual environment hybrid architecture, distribution concept of No Distribution, Greedy Heuristic and Greedy Heuristic with state are to be analyzed. And the rest was the results and conclusion.

2. HYBRID ARCHITECTURE

Hybrid architectures are used to exploit and combine the various user resources i.e. the peer and the servers that is the cloud. According to this section the overall structure of the distributed virtual environment are to be discussed. In the gaming environments there were various players and each players having their own states and their own position in order to maintain the proper functioning of the game. The players are to be connected to the server by means of the game client and the client show the representation of the corresponding game. When each position is to be updated by the client then the positional action manager can be updated. Similarly the state action manager can also update the position of the state.

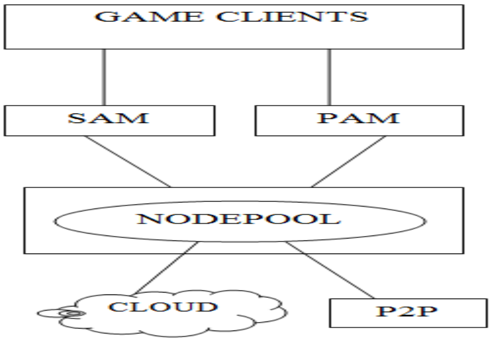

Fig.1 Overall Architecture

Fig. 1 presents the main components of our architecture. The two core distributed components are the Positional Action Manager (PAM) and the State Action Manager (SAM). PAM manages only the positions of the entities and organizes the VE according to principles of area partitioning, so that AOI resolution is simplified. Instead, SAM is organized according to a random object-to-sector assignment, and this allows us to handle the state of the entities without any transfer of the entities across servers due to positional action [5]. Such a transfer may anyway occur, but instead of being triggered by positional actions, it is performed to optimize the distribution of the entities among the nodes. In the positional action manager when a gaming environment starts in a network so that each player in the game can maintain the position and also when the player moves from one position to another position then the updating of the position done in the positional action manager. But when considering the state action manager when a player will be died or making any changes in that environment, then that changes of their state can be updated on the state action manager. In this node pool there were the combination of the cloud and peer nodes.

In the distribution concepts the major benefit when compared to the other system configuration is scalability. So the heuristic-based algorithms are to be used for the load distribution. In the initial stage No Distribution policy can be considered. According to the No Distribution policy there was no transfer of load between the nodes. So if too many requests are to be come on to that corresponding server then the fault tolerance can be occurred. This can be avoided by another algorithm as greedy approaches. Moreover in the real time the usage of the cloud node so that if any fault will occur, then a cloud node can be act as a backup virtual server.

I. *Greedy Heuristic:* One of the best optimization algorithms was this. Here also the searching operation was done to select the node that can perform the load distribution. The pseudo code of the greedy approach is shown below

```
1. Initially set the node pool becomes null
2. For each updating of cloud and peer nodes in to the
   pool calculate the score of the nodes
3. According to the basis of the score, nodes are
   arranged in descending order
4. Selection of the best node from the pool by
   comparing the load of the node
5. Load distribute through the best node
6. Update the load of the best node
7. Repeat the step 4 to 6
```

According to the logic of this greedy strategy these operations are to be performed. The selection of the best node is the one of the important part of the searching algorithms. In this the best node is select from the node pool containing different node values.

II *Greedy Heuristic with State*: This is similar to that of greedy heuristic. But the difference is the introduction of the state. That is there were a time was to be set for the selection and performance of a node. The algorithm is same as that of the greedy heuristics. The importance is that the continuous usage of the single node was cannot be allowed here. The pseudo code of this approach is as follows

```
1. Initially set the node pool becomes null
2. For each updating of cloud and peer nodes in to the pool calculate the score of the nodes
3. According to the basis of the score, nodes are arranged in descending order
4. Selection of the best node from the pool by comparing the load of the node
5. Load distribute through the best node
6. Update the load of the best node
7. Set a minimum interval of time for the selection node in order to avoid continuous selection
8. Repeat the step 4 to 6
```

3. RESULTS

When considering the results of these two approaches the greedy heuristic with state is better than that of the other. According to the greedy heuristic the pool having the nodes and the selection of the node was on the basis of the load. At each time when a node will select that nodes weight becomes reduced and utilization of that node becomes increased. In this approach the selection of a node becomes repeatedly, so that the participation of the other nodes in the pool becomes in very delay.

In the experiment we use minimum number of nodes. But when considering very large number of nodes with more amount of resources, then there were a possibility of node that was not participated in the load distribution. From this logic of greedy approaches the experiment was established.

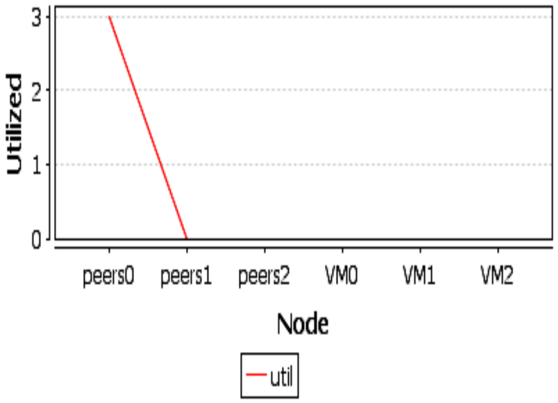

Fig.2 Utilization of nodes using greedy heuristic

In this fig.2 graph we see that the peer0 node should only participated the continuous three times in the distribution. So the other nodes are not takes part in the operation. So the selection can be continuous towards on a particular node or some nodes. But when considering the other greedy approaches like greedy heuristic with state the problem becomes solved.

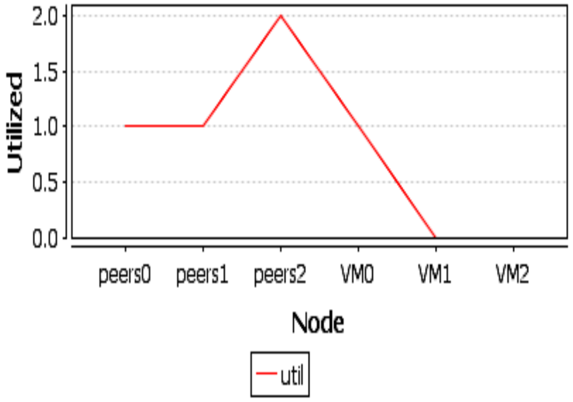

Fig.3 Utilization of nodes using greedy heuristic with state

In this fig.3 shows the approach of greedy heuristic with state, so that the utilization of the nodes becomes increased. Here the continuous utilization of a single node can be avoided. So each node in the node pool was getting a chance to participate in the load distribution. This is the peculiarity of the greedy heuristic with state.

4. CONCLUSION

In this paper we consider the greedy strategies like greedy heuristic and greedy heuristic with state. When considering the gaming environment the effective utilization of the nodes are become the one of the challenge. This can be reduced by the help of these greedy strategies. In this paper the greedy heuristic with state is the better algorithm for the effective utilization of nodes in the gaming environments.

REFERENCES

[1] R. Buyya, C. 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 Computer Systems 25 (6) (2009) 599–616.

[2] V. Nae, A. Iosup, S. Podlipnig, R. Prodan, D. Epema, T. Fahringer, Efficient management of data center resources for Massively Multiplayer Online Games, 2008 SC—International Conference for High Performance Computing, Networking, Storage and Analysis, 2008, pp. 1–12.

[3] V. Nae, R. Prodan, A. Iosup, T. Fahringer, A new business model for massively multiplayer online games, in: Proceeding of the Second Joint WOSP/SIPEW International Conference on Performance Engineering, ACM, 2011, pp. 271–282.

[4] K. Chen, P. Huang, C. Lei, Game traffic analysis: an MMORPG perspective,Computer Networks 50 (16) (2006) 3002–3023.

[5] Emanuele Carlini, Laura Ricci, Massimo Coppola, Flexible load distribution for hybrid distributed virtual environments: Future Generation Computer Systems 29 (2013) 1561–1572.