# Client-Server Assignment for Better Traffic Route in Internet Distributed Systems

Nirmal Zachariah,[1] Syamamol T[2]
Computer Science & EngineeringComputer Science & Engineering
Mangalam College Of Enggmangalam College Of Engg
Ettumanoor, Kerala, IndiaEttumanoor, Kerala, India

**Abstract-- An internet is basically a client-server system with lot of computer's are to be connected together in order to share resources and computation. This paper will introduce the basic concepts of connection between the client machines and server machines. Here the system consists of large number of machines who share data's and given equal priority thus no node will have larger computation than the other one. The load balancing and communication traffic are the two problems to be faced in the distributed system. In this work, here introduced many concepts to solve the above two problems. The method of client server assignment is usually introduced in social networks like Facebook, twitter etc. The basic example of this system is Email, by which the users can send and receive Emails. Here introduced an algorithm center selection problem by this the user can reduce the communication traffic and load balancing. Thus the efficiency of the client-server assignment can be increased to a larger extent.**

***Keywords--Distributed systems, client-server assignments, traffic routing, time complexity, center selection problem.***

## I. INTRODUCTION

An internet is totally a distributed system with many systems of different configurations are connected together to perform many tasks. Mainly the distributed systems are in a state of decentralized. Thus no node is given any importance or priority all nodes are given equal operations of computation for execution. The main problem that the users will be faced today is the cost in making all the system's to equally distributed model. Mainly a distributed system is made of a number of server's and client's to be connected together. The server machine is more computationally efficient than the client machines. The basic example of this model is the Email transmission considers the case with two user's A and B. If A want to send a message to B, then A will first send the message to the A's Email server, then A's Email server send message to B's Email server, Then B's Email server will send message to B thus the message transfer takes place in a Email system. If A and B is situated in same server then the message transfer will takes place very fast, otherwise if A and B is situated in two different Email server then the message transfer will be little bit slow than the earlier one. The advantage of the system are powerful server's by this the process of transfer of messages takes place in Email system, thus it will have a good processing speed, more scalable, and last can afford large number of Email's in it. Email system can be basically implemented in company's organization where employees are the different clients to send message between them. If the company wants to send any messages to the employees then they post the information in their Email server and thus all the employees can access that message from the Email server.

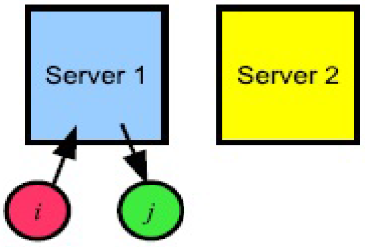


Fig1: Client i and j are logged in server 1. The messages between them are passed only via server 1.

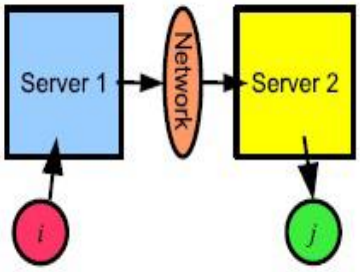


Fig 2: Client i is logged in server 1 and client j is logged in server 2. The messages are passed through server 1 and 2, which causes extra communication

The other method of client server operation is the Instant Message Service (IMS) by these users can directly send messages between two parties very easily. The IMS will be had a server i.e., similar to Email server's. The IMS uses XMPP protocol [1]. Here a user is set to a domain such as user@domain like abc@xyz, here abc is the username and xyz is the domain name. Thus if two user's want to send message then, user1 and user2 are two user's and dom1 and dom2 are two domains. If the message transfer will be takes place in the same domain then the message transfer will be very easy otherwise it will be delay due to that in the case when a single domain then the both user's will be deal led in one server (user1@dom1 → user2@dom1)thus user1 → dom1 → user 2, otherwise it had to contact to servers (user1@dom1 → user2@dom2) thus user1 → dom1 → dom2→ user 2. The number of user's will be increased then add more server's to the system and thus the system will be made more scalable. The most modern applications of the system include Facebook, twitter, LinkedIn etc. buy these social network sites users can send and receive messages, also can send files, photos like multimedia content can also be send between each user's in the network. Also the application can implemented on the auction sites like eBay, flip cart etc. or that the server system will be much efficient thus the system can capture all the recent changes that will be happening in the market. Thus all the client system and server system in the network will have a more co-ordination between each other.

## II. EXSISTING SYSTEM

For the implementation of the system first here developed a matrix format table, that table contain all the details of the clients and servers. By taking the values from the table the system can define the overall operations of the clients in the network. This matrix will used as the input for the operation to be carried out. Next task is to assign the different clients to different servers in the system. For that here introduced the general algorithm, by this the system can achieve the optimal client-server assignment. Thus consider M servers and split it into two as M and M-m. Thus all clients in the system will come under these two groups. The problem here is poor traffic routing and time complexity.

## III. PROBLEM DEFINITION

The problem that was to be faced in the existing system is time complexity and traffic routing. The problems can be explained detailed as follows:

### *A. Time Complexity*

In some cases with there will be about 100 clients in a system, so divide them into 50 each thus there will be about a combination of 100 C 50. Thus the system will have to evaluate that much combinations so it will take more time to evaluate all the combinations thus the problem of time complexity is occurred.

### B. Traffic Routing

In the case when there will be four clients C1, C2, C3, C4 and servers S1, S2 then C1 and C2 under S1, then C3 and C4 under S2, C1 and C2 can directly communicate with each other because they are at one server, if C1 and C4 want to communicate then C1 will first send message to S1 then S1 to S2 and then S2 to c4 thus a series of operations will be carried out in the system. Thus the system will take more time for the message transfer. Thus the system will had a poor traffic routing.

## IV. PROPOSED SYSTEM

### *A. SYSTEM ARCHITECTURE*

The below shown architecture is used for the implementation of the system. The architecture will be consist of 3 parts.

1. Reconnect and Redirection server
2. N servers
3. N clients

The overall working of the architecture is said to be described as following steps

1. Client 1 will send a connect request to Reconnect server.
2. Reconnect server will process that request and then send a reconnect message to client 1.
3. Thus client 1 can connect with the server 1 by sending a connect message to server 1 from client 1.
4. After the connection is established client 1 will send a message to client 2. Thus that message will be received at server 1 and server 1 will pass that message to client 2, because client 1 and 2 are connected to server 1 thus this message transfer is called as inter message transfer.
5. Thus the message from server 1 to client 2 will be received at client 2, initially the message was send by client 1 and it will pass to server 1 and then server 1 will pass that message to client 2.
6. When client 2 want to send a message to client 4 then client 2 will send that message to server 1 then server 1 will pass that message to server 2 thus the message will be received at server 2 and server 2 will send that message to client 4. This type of message transfer is called as the intra message transfer.
7. After a series of message transfer will be performed by the system then the reconnect servers will analyses all the type of message transfer that was said to be occurred in the system. Thus it will reconnect the servers and clients in the system who will be frequently contacted with each other. i.e.Server 1 and client 1 and 4 will be frequently contacted then reconnect server put client 1 and 4 in the under of server 1. Thus in 7) the reconnect server will send a reconnect message to client 4. Thus client 4 will be reconnecting to server 1.
8. Client 4 will be reconnecting to server 1 from server 2. Thus the transfer of message from client 1 and 4 will be made easier.
9. Thus the traffic of the system can be controlled and thus the traffic between clients and servers can be reduced to a very small extent.

### B. *CLIENT MACHINE*

The client machines will connect to the reconnection server, the reconnection server will be able to perform three types of operations:

1. The connect request from client machine is processed and then it will connect the client machine to the corresponding server machine. When the message transfer between a server and a client is said to be completed then type of connection is established between client and server is released. When there will be an existing connection, after sometimes the reconnect server will find a better server for the client machine and then the server of the client is remapped.
2. Receive the different messages from the message server and display the received messages.
3. Send Different type of messages to message server.

These are the three operations performed by the client machine.

### C. *MESSAGE SERVER*

In this model the first process is to establish a connection. When a connection is said to be established then the system started receiving of different messages. After the process of message receiving is completed, then the received messages will be routed to the concern machine in the system. These are the different operations performed by the message server.

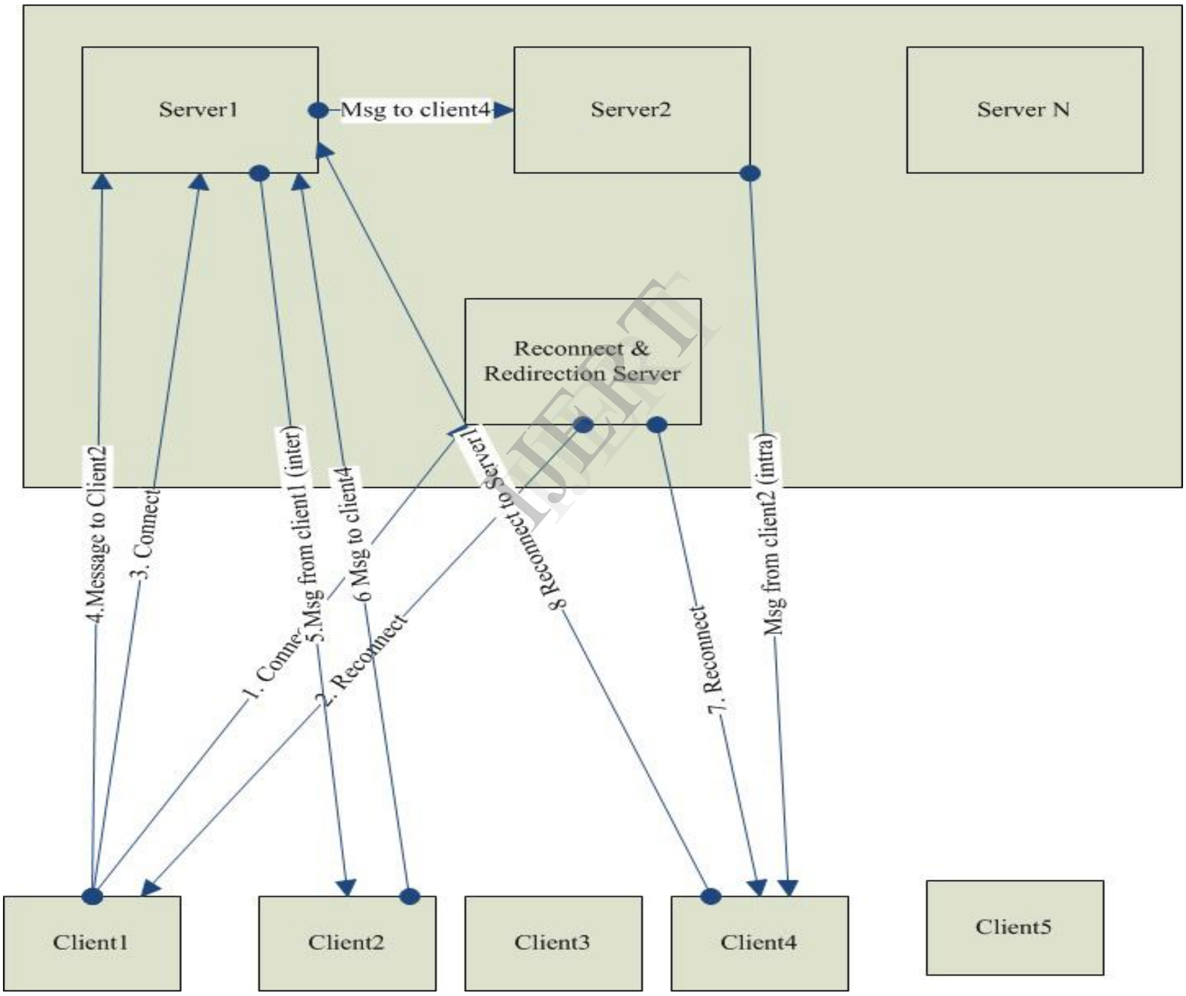


Fig. 3System Architecture

### *D. RECONNECT SERVER*

In this model the system, initially the connection from client is received to the reconnection server is received is received then, this model will connect the client machine to server, after sometime based on the information get from different clients and server machines the route will be remapped, based on the client machine is connected to low load servers and the process is continued.

After sometime the system collects the traffic information from the servers and clients then perform partition algorithm and collects the reconnect information from the clients and servers in the system and inform that the reconnect information to all clients in the system.

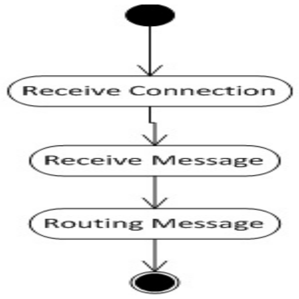


Fig 4: Message Server

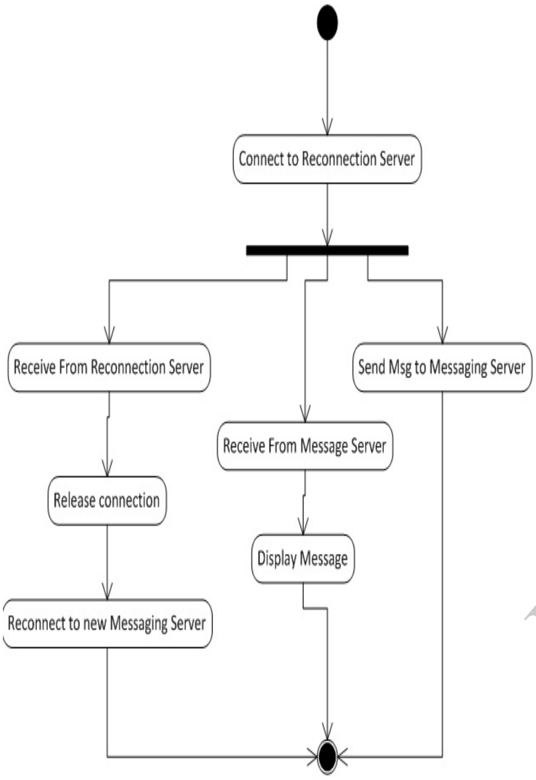

Connect to Reconnection Server
Receive From Reconnection Server
Send Msg to Messaging Server
Receive From Message Server
Release connection
Display Message
Reconnect to new Messaging Server

Fig5: Client Machine

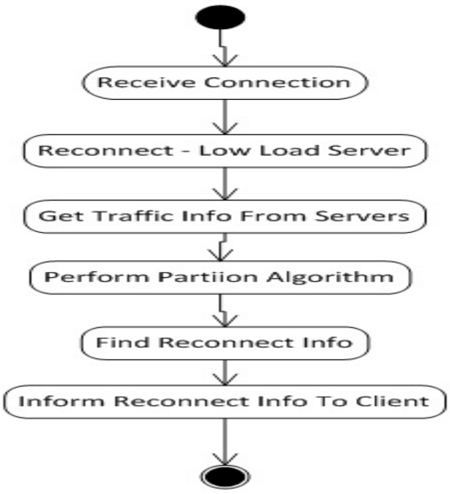

Receive Connection
Reconnect - Low Load Server
Get Traffic Info From Servers
Perform Partiion Algorithm
Find Reconnect Info
Inform Reconnect Info To Client

Fig 6: Reconnect Server

## V. IMPLEMENTATION

The major problems of the existing system was with the time complexity and better traffic routing so that the implementation of the system will be as to avoid all the problems of the existing system. The two goals can be achieved by using of an algorithm called center selection problem. Thus here formulate a mathematical algorithm based on that the above explained goals can be achieved.

### A. NOTATION

The following notation can be used to formulate the algorithm

$N_l$: List of nodes
$c_n$ : list of clients
$s_n$ : list of servers
$c_i$ : client
$s_i$ : servers

### B. GENERAL ALGORITHM

```
$c_n = \phi$ [initially]
Find a node € $N_l$ with highest traffic
Add to $c_n$
While $N(c_n)$ != $s_n$
For ( node $n_i$ in $N_l$- $c_n$ with minimum interaction with $c_n$)
Add to $c_n$
End(For)
End(While)

Grouping of nodes

For ( each node $c_i$ in $c_n$)
add $c_i$ to $s_i$
End(For)
For( each node in $N_l$ - $c_n$)
find node ci in $c_n$ with maximum interaction with $n_i$
add $n_i$ to $s_i$
End(For)
```

Here $N_l$ which represents the list of nodes in the system, assign a value $c_n$, which is initially zero(0) assigned to it. First step is to find a node from list of nodes with highest traffic to be experienced, continue until the value of $c_{n,}$ not equal to the number of server, $s_n$. Then find another node $n_i$ from $N_l$ - $c_{n,}$with minimum interaction with $c_n$. Then add that node to $c_{n,}$ thus the process of finding the server is completed. Next process is to find nodes to the server concerned i.e. select the nodes that come under a unique server. So that select some nodes with maximum interaction with server node and the nodes with maximum interaction withthe server node will come under that specified node. Thus the problem can be solved.

### *C. TRAFFIC ANALYSIS*

Consider the above table there will be about seven nodes to be represented. Here calculated the traffic with each node with other nodes in the table. Select a node with highest traffic. Let N6 be the node with highest traffic so assign N6 as server S1. Then find another node in the table with less traffic than N6 and it will be N7 name it as S2. Then find another node in the table with less traffic than N6 and N7, it will be N9 name it as S3. So here find out the servers in the system, then find the nodes that will have maximum interaction with these three nodes and assign that nodes under these three specific nodes, do this process until all nodes to be listed are grouped.

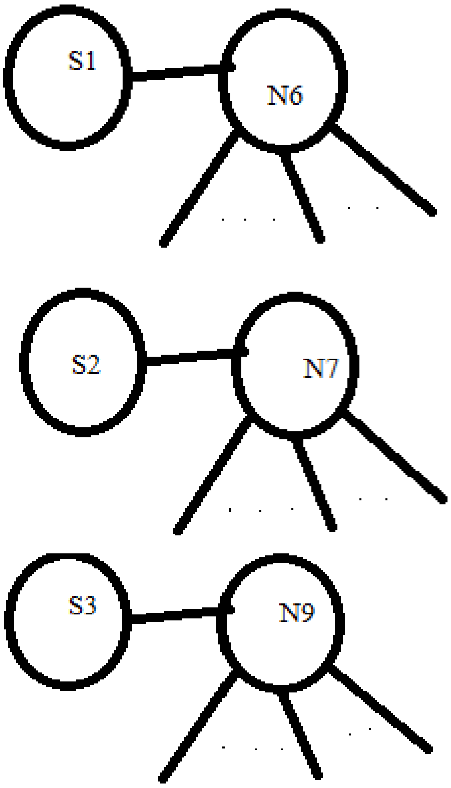
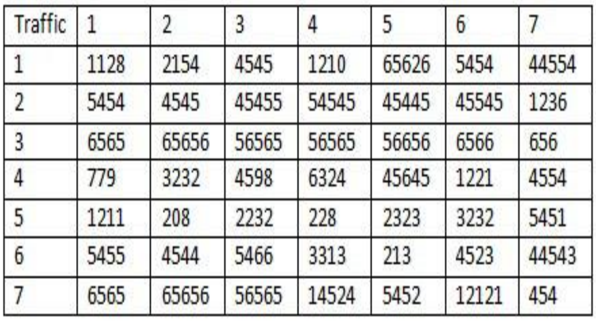


Fig 7: Generation of Nodes

Thus by this implementation the traffic can be reduced to a very much lower extent so that the communication between each node in the system is more compact.

Table 1: TRAFFIC TABLE



| Traffic | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| 1 | 1128 | 2154 | 4545 | 1210 | 65626 | 5454 | 44554 |
| 2 | 5454 | 4545 | 45455 | 54545 | 45445 | 45545 | 1236 |
| 3 | 6565 | 65656 | 56565 | 56565 | 56656 | 6566 | 656 |
| 4 | 779 | 3232 | 4598 | 6324 | 45645 | 1221 | 4554 |
| 5 | 1211 | 208 | 2232 | 228 | 2323 | 3232 | 5451 |
| 6 | 5455 | 4544 | 5466 | 3313 | 213 | 4523 | 44543 |
| 7 | 6565 | 65656 | 56565 | 14524 | 5452 | 12121 | 454 |

The execution of the tasks in the system is done very fast so that the time complexity of the system can be achieved better. Thus the two problem definitions of the system can be solved.

## VI. PERFORMACE EVALUATION

To measure the performance of the system, firstly compare the total data size for each node on different platforms. All the benchmarks are designed to be executed within the specified constrains. Thus the overall performance of the system can be evaluated. Here two graphs are shown, Normal method and randomized method graphs. Normal method is said to be carried out by entering values by the user so that the graph that was to be obtained will contain a continuously changing values in it. Thus the graph will be plotted as a discontinuous graph of continuously growing and decreasing style. In the case of randomized graph there valued are said to be gained from internet by extracting the traffic values of different sites, so that the traffic of the sites is said to be growing continuously of one site to another thus the system will gain a graph of exponentially growing style.

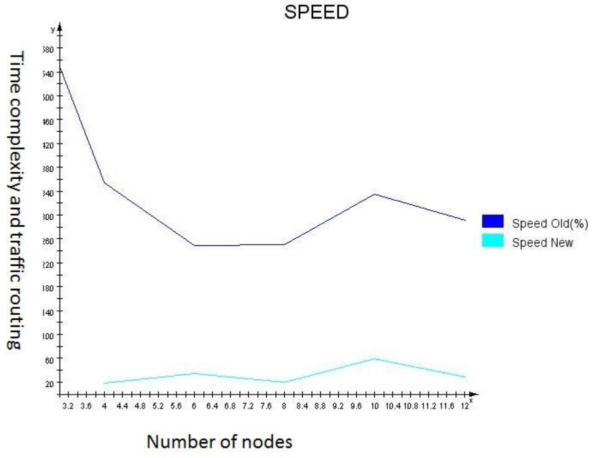


Fig 8: Normal Method

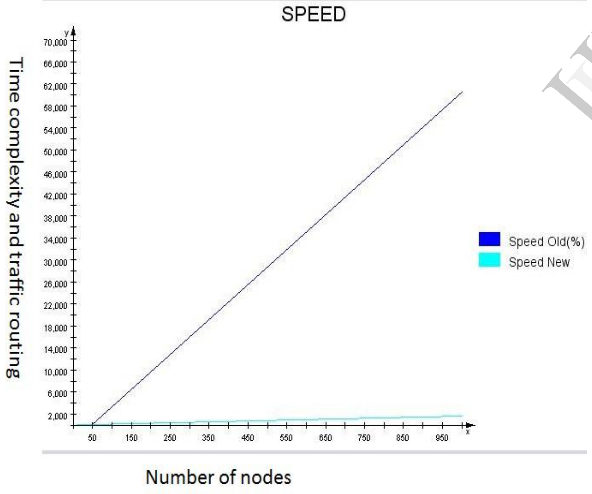


Fig 9: Randomized Method

## VII. CONCLUSION

In this work, here present a mathematical model and an algorithmic solution to the client-server assignment problem for optimizing the performance of a class of distributed systems over the Internet. In general, here find's an optimal client-server assignment for some pre-specified requirements on time complexity reduction and better traffic route, and propose center selection problem analysis for finding the approximate solution to the client-server assignment problems. Thus the interaction among different servers and clients in system are made easier and traffic free.Thus the client-server assignment mademore accurate.

## VIII.FUTUERSCOPE

The current system is based on center selection problem. Thus the time complexity and traffic routing is a main problem in this design. By this design the traffic only made completely secured the load balancing is still a problem by this design. So as a future work, the problem of load balancing is said to be solved. Then we can able to say that all problems in client server assignmentare being solved.

## IX. ACKNOWLEDGEMENT

The authors would like to thank all of their colleagues for their discussions and suggestions.

## X. REFERENCES

[1] [Online] Available:http://xmpp.org/about/

[2] J. Dean and S. Ghemawat, MapReduce: simplified data processing on large clusters, Commun. ACM, vol. 51, pp. 107113, January 2008. [Online]. Available: http://doi.acm.org/10.1145/1327452.1327492

[3] J. Shi and J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 888905, 2000.

[4] Z. Wu and R. Leahy, An optimal graph theoretic approach to data clustering: theory and its application to image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, no. 11, pp. 11011113, August 2002.

[5] K. Lang, Fixing two weaknesses of the Spectral Method, in Advances in Neural Information Processing Systems 18, 2006.

[6] Kevin Lang, Finding good nearly balanced cuts in power law graphs, Yahoo Research Labs, Tech. Rep., 2004.

[7] M. Kurucz, A. Benczur, K. Csalogany, and L. Lukacs, Spectral clustering in telephone call graphs, in Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis, ser. WebKDD/SNA-KDD 07. New York, NY, USA: ACM, 2007, pp. 8291.

[8] J. C. S. Lui and M. F. Chan, An efficient partitioning algorithm for distributed virtual environment systems, IEEE Trans. Parallel Distrib. Syst., vol. 13, pp. 193211, March 2002.

[9] P. Morillo, J. M. Orduna, M. Fernandez, and J. Duato, Improving the performance of distributed virtual environment systems, IEEE Transactions on Parallel and Distributed Systems, vol. 16, pp. 637649, 2005.

[10] Y. Deng and R.W. H. Lau, Heat diffusion based dynamic load balancing for distributed virtual environments, in Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology, ser. VRST 10. New York, NY, USA: ACM, 2010, pp. 203210.

[11] H. Nishida, Optimal client-server assignment for internet distributed system, Oregon State University, Tech. Rep., 2011.

[12] Ncut. [Online]. Available: http://www.seas.upenn.edu/ timothee/ software/ncut/ncut.html