An Efficient Distributed Interactive Applications In Client Assignment Problem

Sravya Boya[1], Divya Boya[2]
[1]*Assistant Professor, Arjun College of Technology and Science*
[2]*Assistant Professor, Arjun College of Technology and Science*

ABSTRACT: The World Wide Web is used by millions of people every day for various purposes includingemail, reading news, downloading music, online shopping or simply accessing informationabout anything. Using a standard web browser, the user can access information stored on Webservers situated anywhere on the globe.An ideal system in the distributed architecture provide equal responsibility and computational power but an engaged system it fails to provide that facility among the distributed network and it also provide a load balancing problem when two or more packet send at the same time and some of the data were loosed because of collusion. For this matter many investigators try to balance the load of the server, but it improved in slight level only and if sequence of data occurred it may cause congestion problem. To recover the problem here we propose a Heuristic Algorithm (HA) to fully solve the load balancing and traffic avoidance problem and we also focus on security problem with the help of DSA Algorithm.

KEYWORDS-Distributed System, DIA, CAP, Heuristic Algorithm, Load Balancing, Traffic Control and Security.

I. INTRODUCTION

An Internet distributed system consists of a number of nodes (e.g., computers) that arelinked together in ways that allow them to share resources and computation. An idealdistributed system is completely decentralized, and that every node is given equal responsibilityand no node is more computational or resource powerful than any other. However, for manyreal world applications, such a system often has a low performance due to a significant cost ofcoordinating the nodes in a completely distributed manner. In practice, a typical distributedsystem consists of a mix of servers and clients. The servers are more computational andresource powerful than the clients. Typical examples of such systems are e-mail, instantmessaging, e-commerce, etc. For sending a mail from node A to another node B, the data firstflows to email server of node A. Then the data flows towards the email server of node B and finally it reaches node B. Hence, it is the responsibility of email servers to send receives emailsfor the clients assigned to it. Clients do not communicate with each other directly. Serverscommunicate on behalf of their clients. Similar kind of process can be used for instantmessaging services and ecommerce services.

Client server assignment can be designed based on the following observations:

1. If two clients are assigned to the same server, the server will receive message fromone client and forward to another one. If they are on different servers, the sender client firstsends to its server. The sender's server will receive data and forward it to the receiver's server.

The receiver server will receive data and forward it to receiver client. Thus the totalcommunication between two frequently interacting clients increases if they are assigned to twodifferent servers. Assigning these two clients to a single server makes all information exchangelocally. It is more efficient to have all the clients assigned to few servers to minimize totalcommunication.

2. On the contrary having fewer servers, results in those servers being heavily loadedwhile others are left underutilized. If a server is heavily loaded it results in low performance dueto excessive resource usage on that server. Thus it is necessary to consider load balance on theserver while assigning clients. From the above observations it is clear that total communicationload and load

balancing are two contradicting features. Hence equilibrium must be maintainedbetween overall communication load and load balancing of the servers.

A classical example of such systems is Email. When a client A sends an email to anotherclient B, A does not send the email directly to B. Instead, A sends its message to its email serverwhich has been previously assigned to handle all the emails to and from A. This server relays A'semail to another server which has been previously assigned to handle emails for B. B then readsA's email by downloading it from its local server. Importantly, the email servers communicatewith each other on behalf of their clients. The main advantage of this architecture isspecialization, in the sense that the powerful dedicated email servers release their clients fromthe responsibility associated with many tasks including processing and storing emails, and thusmaking email applications more scalable.

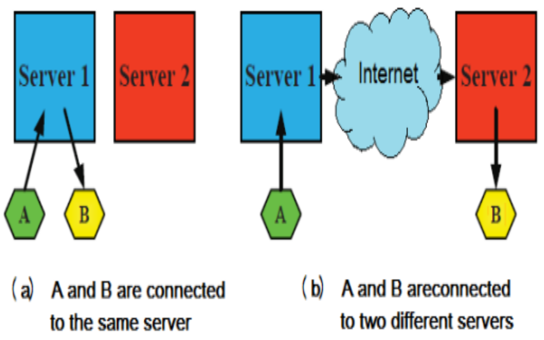

Fig1: A exampleof client assignments to servers

II. RELATED WORK

In paper [1] the author discussed the problem of latency inthe network and uses King tool. This tool that accuratelyand quickly estimates the latency between arbitrary end hostsby using recursive DNS queries in a novel way. It does notrequire the deployment of additional infrastructure.In paper [2] the author discussed about the mirror placementproblem as a case of constrained mirror placement wheremirrors can only be placed on a preselected set of candidates.Performance improvement in terms of client round-trip time(RTT) and server load when clients are clustered by theautonomous systems (AS) in which they reside. the numberof mirror sites (under the constraint) effective in reducingclient to server RTT and server load.

In paper [3] the propose is Game-independent, networkbased service, called Sync-MS, that balances the trade-offbetween response time and fairness. Sync-MS uses twomechanisms: Sync-out mechanism properly queue up themessage at player stations and deliver it to the gameapplication only after the same update message has arrived aall player stations. Sync-in mechanism enforce a sufficientwaiting period on each action message dynamically in orderto guarantee fair processing of all action messages.Thefairness requirement is to ensure that all clients have samechance of participation regardless of their network conditions.

In paper [4] Existing online multiplayer games typically use aclient-server model, which introduces added latency as wellas a single bottleneck and single point of failure to the game.Distributed multiplayer games minimize latency and removethe bottleneck, but require special synchronizationmechanisms to provide a consistent game for all players. Anew synchronization mechanism, trailing statesynchronization (TSS), which is designed around therequirements of distributed first-person shooter games.Trailing state synchronization (TSS) is designed to executecommands quickly while at the same time maintaining aconsistent copy of the game state at all players. Wheninconsistency does occur due to jitter, the application statecan be repair by trailing state synchronization.

In paper [5] the drawback is a novel distributed algorithmthat dynamically selects game servers for a group of gameclients participating in large scale interactive online games.The goal of server selection is to minimize server resourceusage while satisfying the real-time delay constraint. Developa synchronization delay model for interactive games andformulate the server selection problem. The proposedalgorithm, called zoom-in-zoom-out, allow the clients selectappropriate servers in a distributed manner

In paper [6] the author discussed about Collaborative virtualenvironments (CVEs) enable two or more people, separatedin the real world, to share the same virtual 'space'. CVEs iscompromised by one major problem: the delay that exists inthe networks linking users together .The 'Impact-PerceiveAdapt' model of user performance, which considers theinteraction between performance measures, perception oflatency and the breakdown of the perception of immediatecausality, is proposed as an explanation for the observedpattern of performance.

III. SYSTEM AND METHODOLOGY

To solve the aforementioned problems, our systemproposed a client-server assignment problem based ontotal communication load and security of the resourceswith the help of heuristic algorithm for the possibleoptimal solution. For the optimal solution, here thecombination of Greedy Assignment to assign clientsiteratively, Distributed greedy assignment for distribute global knowledge and optical location allotment algorithmto minimize the average incurred delay and also focused on security problem with the help of DSA Algorithm. Aheuristic method is used for making the decision at eachstep, to find the best method. For end-to-end transfer ortransfer among the nodes they were traffic occur becauseof sequence of data transfer from one node to another.

Forfree the traffic and to send the packets here we propose aGreedy Assignment method is used to select the candidateservice node with the major ―potential‖ value. When anode with the largest ―potential‖ value is selected, the―potential‖ value of next candidate service nodes and thecurrent gathered connection possibility of each client nodeare updated this technique were used to free the trafficamong the nodes. For distributing the packets among thenetwork when the selected node is busy means italternatively chooses the other node with the help of ourDistributed greedy assignment technique. Significantlyoutperform the intuitive Nearest-Server by using thistechnique. And finally we propose an optical locationallotment algorithm to avoid or minimize the deserveddelay among the network. And also for the security herewe propose a DSA Algorithm. Digital SignaturesAlgorithm is used to detect unauthorized modifications todata and to authenticate the identity of the signatory.

A. Heuristic algorithm

Heuristic algorithm is used to discover an optimal solutionfor transferring the data. This method is used to progressefficiency of client server assignment. Heuristicalgorithms are used in distribution system for allocatingserver, used to solve the NP hard problems. This algorithmprovides complete an optimal, accurate and completesolution. The client assignment problem is NP-complete;there is no polynomial time to find the optimal clientserver assignment. In existing approach they where lacksto discover the optimal solution in responsible time evenfor small number of clients and servers. Moreover, serverscan fail and may not be able to respond the incomingclient requests. For this reason, heuristic algorithms arepreferred to find near optimal solutions to the clientassignment problem.

B. Distributed Load Balancing Algorithm

In network, more number of packets is sending from clientto server and server to client it may arise delay intransmission because of over load of data among thenodes. Delay on the client side is the sum of network delayand congestion delay at the server. Here the DLBalgorithm used based on two conditions (i) leastconnection and (ii) least response time. In the firstcondition selection of sever based on which servercurrently has a least client connection and the secondcondition is based on the selecting server based on serverquick response time. By using this method congestion freetransmission will achieve for sending the packets. With thehelp of DLB algorithm it minimizes the maximumincurred delay using an approximation and optimalalgorithm. It minimizes the communication load betweenservers and it sends the packets in the manner ofcongestion free transmission.

C. Distributed Load Balancing Algorithm

```
// peer status update
prob_space[0]=0; load_diff = 0; load_diff_sum = 0;
for(j=1; j<=n; j++){
    if(load_i - peer[j].load){
       load_diff = load_i - peer[j].load;
       //insert the new difference
       build_prob_space(load_diff, prob_space);
       load_diff_sum = load_diff_sum + load_diff;
    }
    //normalize the vector elements
    update_prob_space(load_diff_sum, prob_space);
}

// balancing process
if(prob_space[] == NULL)  //no neighbors with lower load
    //serve locally the request
    serve_request();
else{
    float x = rand(); //random number generator
    int req_sent = 0; int i = 0;
    while(prob_space[i] == 1 or req_sent == 1){
        if(prob_space[i-1] <= x < prob_space[i]){
            //send request to the chosen peer
            send_to(peer[i-1].addr);
            req_sent = 1;
        }
        i++;
    }
}
```

D. Distributed greedy assignment technique

From the distribution network transferring of data isavailable at very frequent situation. In this situationsequence of data transfer may cause the packet loss,congestion, jamming data, etc. to solve these issues ourproposed technique were used to select the nearest serverto transfer the data from one node to another when thenode were busy it transfer to the next nearest node basedon our Distributed greedy assignment technique. The Fig 2shows the clear view of the transferring data in the busyserver architecture.

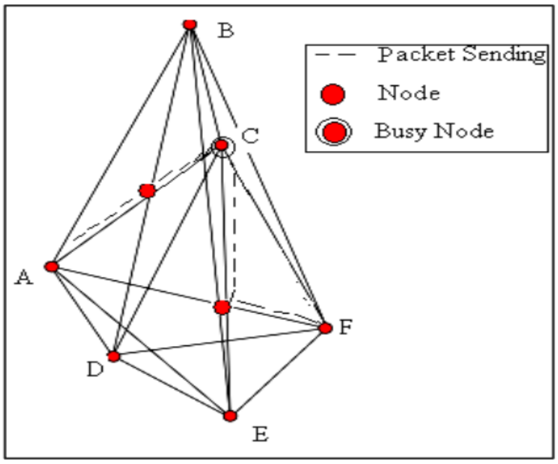

Fig 2: Selecting of nearest node

From the above diagram it shows that when a data istransfer from one node to another if the selected node isbusy in the network means it automatically send the datato the nearest node and it pass to the designation nodewithout loss of data. Here it avoids the congestion in thenetwork based on our Distributed greedy assignmenttechnique. Our proposed method proves better result whencompared to the existence one.

IV. METHODOLOGY

A. Content Delivery Network

The consistency requirement for continuous DIAs is toensure that all clients share the same view of theapplication state when their respective simulation timesreach the same value. This is automatically guaranteedamong the clients assigned to the same server because theyall inherit the application state from their assigned serverthrough state updates. Nevertheless, the application statesseen by the clients assigned to different servers may not beidentical at the same simulation time if the applicationstates maintained by their assigned servers are notconsistent. Since the state of a continuous DIA changesdue to both user operations and time passing, to ensureconsistency among the application states at the servers,each user operation must be executed by all servers at thesame simulation time.

B. Distributed-Modify Assignment

Distributed-Modify Assignment is performed in adistributed manner without requiring the global knowledgeof the network at any single server. It starts with an initialassignment. Then, the assignment is continuouslymodified for reducing the maximum interaction pathlength D until it cannot be further reduced. This process isreferred to as the assignment modification. One server iselected as a coordinator responsible for calculating D andselecting the server to execute the transfer modification.To compute D of the primary assignment, each servermeasures its distance which is said to be network latenciesto all the other servers. It also computes its distances to allthe clients that are allocated to it and preserve them as asorted list. Then, each server s broadcasts to all the otherservers its longest distance l(s) to its clients, and sends theinter server distances to the coordinator. The coordinatorcalculates D based on the received information.

C. Transaction-Least-Work-Left

The Transaction-Least- Work-Left (TLWL) algorithmaddresses load balancing issue by assigning differentweights to different transactions based on their relativecosts. Counters are preserved by the load balancerrepresenting the weighted number of transactions allottingto each server. Here in the transaction new calls areallocated to the server with the lowest contradictor. TLWLguess the server load which is based on the weightednumber of communication among the server which serveris currently handling the data. TLWL can be personalizedto workloads with other transaction types by usingdissimilar loads based on the overheads of the transactiontypes. In addition, the comparative costs used for TLWLcould be adaptively varied to progress the performance ofnetwork.

D. Load balancer

In the network, client sends the data which is said to beparser which is parsed the request to the designation nodewith the help of session recognition were implemented.This determines if the request communicate to an alreadyaccessible session by querying the Session State. If ithappens, the request is forwarded to the server to whichthe session was previously assigned. If it not exists, ourServer Selection module assigns the new session to aserver using TLWL algorithm. For numerous of the loadbalancing algorithms, this effort may be depend on LoadEstimates, it preserves the data for each of the servers. TheSender one is it forwards requests to servers and updatesLoad Estimates and Session State as needed. The Receiveralso receives responses sent by servers. The client toreceive the response is recognized by the SessionRecognition module which acquires this information byquerying the Session State. The Sender then sends theresponse to the client and updates Load Estimates andSession State as needed. The Trigger module updatesSession State and Load Estimates after a session hasexpired.

IV. CONCLUSION

The client assignment problem for interactivity enhancementin continuous DIAs is investigated. The interactivityperformance of continuous DIAs under the consistency andfairness requirements is modeled. To solve these problems wepropose a heuristic algorithm to avoid the abovementioned problems. With the help of HA the we providethe three combinations they were Greedy Assignment toassign clients iteratively, Distributed greedy assignmentfor distribute global knowledge and optical locationallotment algorithm to minimize the average incurreddelay in the network. Additionally we also focus on thesecurity of the network with the help of DSA Algorithm. Itis used to detect unauthorized modifications to data and toauthenticate the identity of the signatory.

REFERENCES

[1]. J.C.S. Lui and M.F. Chan, ―An Efficient Partitioning Algorithm for Distributed Virtual Environment Systems,‖ IEEE Trans.Parallel and Distributed Systems, vol. 13, no. 3, pp. 193-211, Mar .

[2]. Greenberg, S. and Marwood, D. Real Time Groupware as a Distributed System: Concurrency Control and its Effect on the Interface. In: Proc. ACM CSCW, Chapel Hill, NC, USA, pages 207–217, October 1994.

[3]. Patil, Bharati, and S. B. Patil. "A Novel Approach on Client Server Assignment Problem in Distributed System" [4]. Zhang, L.; Tang, X.; , "Optimizing Client Assignment for Enhancing Interactivity in Distributed Interactive Applications," Networking,IEEE/ACM Transactions on , vol.PP, no.99, pp.1, 0doi: 10.1109/TNET.2012.2187674.

[5]. Lu Zhang; Xueyan Tang; , "Client assignment for improving interactivity in distributed interactive applications," INFOCOM, 2011Proceedings IEEE , vol., no., pp.3227-3235, 10-15 April 2011doi: 10.1109/INFCOM.2011.5935173

[6]. Ucar, Seyhan, HuseyinGuler, and OznurOzkasap. "Online Client Assignment in Dynamic Real-Time Distributed Interactive Applications" Distributed Simulation and Real Time Applications (DS-RT), 2013 IEEE/ACM 17th International Symposium on, IEEE, 2013.

[7]. Nguyen, Cong Duc, FarzadSafaei, and Paul Boustead. Distributed server architecture for providing

immersive audio communication to massively multiplayer online games." Networks, 2004.(ICON 2004). Proceedings.12th IEEE International Conference on.Vol. 1.IEEE, 2004.

[8]. Hiroshi Nishida, Member, IEEE, and Thinh Nguyen, Member, IEEE-―Optimal Client-Server Assignment for Internet Distributed Systems‖-ieee transactions on parallel and distributed systems, vol. 24, no.3, march 2013.

[9] D. Delaney, T. Ward, and S. McLoone, "On Consistency and Network Latency in Distributed Interactive Applications: A Survey-Part I," Presence: Teleoperators and Virtual Environments, vol. 15, no. 2, pp. 218-234, 2006.

[10] M.R. Garey and D.S. Johnson, "Computers and Intractability: A Guide to the Theory of NPCompleteness," WH Freeman and Company, San Francisco, Calif, 1979.

[11] L. Gautier, C. Diot, and J. Kurose, "End-to-End Transmission Control Mechanisms for Multiparty Interactive Applications on the Internet," Proc. IEEE INFOCOM '99, pp. 1470-1479, 1999.

[12] K.P. Gummadi, S. Saroiu, and S.D. Gribble, "King: Estimating Latency between Arbitrary Internet End Hosts," Proc. Second ACM SIGCOMM Workshop Internet Measurement, pp. 5-18, 2002.

[13] Y. He, M. Faloutsos, S. Krishnamurthy, and B. Huffaker, "On Routing Asymmetry in the Internet," Proc. IEEE Global Telecomm.Conf. (GLOBECOM '05), 2005.

[14] C. Jay, M. Glencross, and R. Hubbold, "Modeling the Effects of Delayed Haptic and Visual Feedback in a Collaborative Virtual Environment," ACM Trans. Computer-Human Interaction, vol. 14, no. 2, article 8, 2007.

[15] M.R. Korupolu, C.G. Plaxton, and R. Rajaraman, "Analysis of a Local Search Heuristic for Facility Location Problems," J. Algorithms, vol. 37, no. 1, pp. 146-188, 2000.

Authors:

Sravya Boya Completed M.Tech (SE) and working as Asst. Professor in Arjun College of Technology and Science.

Divya Boya Completed M.Tech (SE) and working as Asst. Professor in Arjun College of Technology and Science.