ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

The Client Assignment Problem for Continuous DIA: Analysis, Algorithms, and Evaluation.

Koraveni Vijay
M.Tech Student
Department of CSE
Global Group Of Institutions
Batasingaram,
Ranga Reddy (Dist), India

Mr. S Dilli Babu, M. Tech
Assistant. Professor,
Department of CSE
Global Group Of Institutions
Batasingaram,
Ranga Reddy (Dist), India.

Mr. M V Narayana, M. Tech(Ph.D)
Associate Professor & HOD
Department of CSE
Global Group Of Institutions
Batasingaram,
Ranga Reddy (Dist), India.

Abstract*—Quality of user experience in Distributed Interactive Applications (DIAs) highly depends on the network latencies during the system execution. In DIAs, each user is assigned to a server and communication with any other client is performed through its assigned server. Hence, latency measured between two clients, called interaction time, consists of two components. One is the latency between the client and its assigned server, and the other is the inter-server latency, that is the latency between servers that the clients are assigned. In this paper, we investigate a real-time client to server assignment scheme in a DIA where the objective is to minimize the interaction time among clients. The client assignment problem is known to be NP-complete and heuristics play an important role in finding near optimal solutions. We propose two distributed heuristic algorithms to the online client assignment problem in a dynamic DIA system. We utilized real-time Internet latency data on the PlanetLab platform and performed extensive experiments using geographically distributed PlanetLab nodes where nodes can arbitrarily join/leave the system. The experimental results demonstrate that our proposed algorithms can reduce the maximum interaction time among clients up to 45% compared to an existing baseline technique.

***Index Terms*—distributed interactive application, client assignment, interactivity, consistency, fairness, NP-complete**

INTRODUCTION

Distributed Interactive Applications (DIAs) are network applications that enable interaction between clients geographically distributed around the world. Online games, military simulations and collaborative designs are some examples of DIAs . In a DIA, minimizing the communication delay is a crucial objective that attracts more clients to join the system. The communication delay in DIA is defined as the time duration between when a client triggers an operation and when this operation is transferred to other clients . Different architectures have been proposed to decrease the interaction time between clients which can be classified into three groups, namely client-server, peer-to-peer and mirrored distributed server architectures. In the client-server architecture, one server controls the application and each client connects to the system through that single entity. Consistency is one of the important advantages because each client is directly informed by the central server and each receives other clients' operations simultaneously. However, since clients can only connect to a central server, this server may become a bottleneck for the application. In peerto-peer architecture, instead of using a central server, clients are connected to each other and share the workload among them. Workload sharing can be done in different ways such as partitioning the environment into regions, and assigning each of the regions to one client. However, the problem in the peerto-peer system arises when the performance of a client is bad relative to the others. For example, in online games, clients may handle the processing of region assignments

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

instead of receiving game updates which decreases the user satisfaction.

When issuing an operation, a client first sends the operation to its assigned server. Then, the server forward the operation to all the other servers. On receiving the operation, each server calculates the new state of the application and sends a state update to all the clients assigned to it. Thus, the clients interact with one another through their assigned servers. The interaction time between any pair of clients must include the network latencies between the clients and their assigned servers, and the network latency between their assigned servers. These network latencies are directly affected by how the clients are assigned to the servers. In addition, the interaction time is also influenced by the consistency and fairness requirements of DIAs. Consistency means that shared common views of the application state must be created among all clients to support meaningful interactions .Fairness, on the other hand, is to ensure that all clients have the same chance of participation regardless of their network conditions . Maintaining consistency and fairness in DIAs usually introduces artificial synchronization delays in the interactions among clients due to diverse network latencies . These synchronization delays are also dependent on the assignment of clients to servers. Therefore, how to assign the clients to the servers in DIAs is of crucial importance to their interactivity performance.

RELATED WORK

The literature presents few studies that directly address the client assignment problem in DIAs. In [7] and [8], authors propose mirrored server placement algorithm for content distribution networks (CDN). In these works, the objective is to serve clients in a fast manner by redirecting incoming clients into one of the mirrored servers. Given the set of servers, authors investigate the placement of these servers to maximize the performance. In contrast to CDN, in DIA rather than finding optimal geographical server placement the idea is to find optimal client to server assignments. Moreover, each client in DIA is connected to one server and clients interact with each other through their assigned servers.

In [5] and [9], authors prove that the client assignment problem is NP-complete and there is no polynomial time algorithm to find the optimal solution. For that reason, they propose four heuristic algorithms. In Nearest-Server Assignment, clients are greedily assigned to their nearest server. In Longest-FirstBatch Assignment, the first client is assigned to its nearest server. All other clients which are not far away from this client are assigned to the same server since they will not increase the interaction delay. If that is not the case, then the client will be assigned to its nearest server and the interaction delay is updated accordingly. The Greedy Assignment works similar to Longest-First-Batch Assignment and the only difference is that they use a cost metric to decide which server to assign the client.

In Distributed-Greedy Assignment, the process starts with the initial assignment and continues to modify client assignments until the point where maximum interaction path cannot be reduced further. They utilize the Meridian [10] internet latency data in the evaluation of their algorithms. In [11] and [12], authors propose an approach to enhance the interactivity of DIAs by only considering the network latencies between client and server pairs. After the server placement, proposed algorithm uses the network latencies during the client assignment. However, as we show in Section VI, the interserver latencies also play a critical role in improving the interactivity in DIAs.

In [13], the proposed solution is based on a virtual environment that is partitioned into several zones and each zone is controlled by a server. Clients in the same zone can interact with each other and clients can move to other zones as well. They propose two algorithms namely, Initial Assignment 1http://www.planet-lab.org/ where zones are sorted based on the total weight of clients then assign the first zone to the first server, and Refined Assignment where they further reduce the Initial Assignment by reassigning the

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

clients whose communication delay to their current server exceed a pre-defined threshold. In [14], the assignment problem is mathematically modeled and an approximation algorithm is proposed. The study shows that finding the optimal client-server assignment with predefined requirements is NP-hard and relaxed convex optimization is proposed to find an approximate solution. The main idea behind the proposed optimization algorithm is to divide servers into two groups recursively until the point where no further split can be applied.

In [15], a partitioning algorithm is proposed to reduce the inconsistency in a multi-server distributed virtual environment. The main purpose is to efficiently distribute the network traffic generated by avatars among different servers in the system. By using the metric time-space inconsistency [16], the problem is formulated as a mixed integer programming problem. Alternating optimization is used to divide the problem into two sub-problems. In [17], the authors investigate the update scheduling for distributed virtual environment (DVE). The key idea is to keep the DVE consistent where state updates are applied based on their potential impacts on the consistency. They propose three algorithms that utilize current network delays and estimate inconsistencies that may occur in future and show that the proposed algorithms significantly outperform the intuitive update algorithms. Different from our work, where we aim to find near optimal client to server assignments, they focus on how to schedule particular updates by using network capacity and delay.

In [18] and [19], the existing algorithms in [5] and [9] are modified to handle dynamic network conditions. Since the Meridian [10] set does not consider the latency variation over time, authors collect pairwise latency data from Planetlab-AllPairs-Ping [20] over a one day period. By using collected Internet latency data, they experimentally evaluate the proposed algorithms with dynamic client join/leave. However, they still consider an offline version of the client assignment problem using latencies between clients and servers known beforehand hence not real-time. In contrast, we examine the online client assignment problem in this work.

EXISTING SYSTEM

Distributed interactive applications (DIAs), such as multiplayer online games and distributed interactive simulations, allow participants at different locations to interact with one another through networks. Thus, the interactivity of DIAs is important for participants to have enjoyable interaction experiences. Normally, interactivity is characterized by the duration from the time when a participant issues an operation to the time when the effect of the operation is presented to the same participant or other participants . We refer to this duration as the interaction time between participants. Network latency is known as a major barrier to provide good interactivity in DIAs . It cannot be eliminated from the interactions among participants and has a lower theoretical limit imposed by the speed of light.

DISADVANTAGES:-

1. Interaction between the client and server not much effective.
2. It has more Network latency which barries interactive of DIA.

PROPOSED SYSTEM

In this paper, we investigate the problem of effectively assigning clients to servers for maximizing the interactivity of DIAs. We focus on continuous DIAs that change their states not only in response to user-initiated operations but also due to the passing of time Several heuristic assignment algorithms are then proposed. Their approximation ratios are theoretically analyzed. The performance of the algorithms is also experimentally evaluated using real Internet latency data. The results show that our proposed Greedy Assignment and Distributed-Modify Assignment algorithms generally produce near optimal interactivity and significantly reduce the interaction time between clients compared to the intuitive Nearest-Server Assignment algorithm that assigns each client to its nearest server. Distributed-Modify Assignment also

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

has good adaptivity to dynamics in client participation and network latency.

ADVANTAGES:-

1. Reducing network latency for improving interactivity in DIAs.
2. Server calculation more effective than existing system.

IMPLEMENTATION

Nearest-Server Assignment:-

The first algorithm is called Nearest-Server Assignment, which intuitively assigns clients to their nearest servers. This algorithm can be implemented by having each client measure the network latencies between itself and all servers, and select the server with the lowest latency as its assigned server.

The assumption of the triangle inequality is commonly made when theoretically analyzing the performance of the approximation algorithms in facility location problems When assuming that the network latency satisfies the triangle inequality, we can show that Nearest- Server Assignment has a tight approximation ratio of 3. In the absence of the triangle inequality , Nearest-Server Assignment cannot achieve any bounded approximation ratio. Please refer to Appendix D of the online supplemental material for the detailed proof.

Greedy Assignment:-

The second algorithm Greedy assignment adopts a greedy approach to assign clients iteratively, starting with an empty assignment. In each step, the algorithm considers all the possibilities of assigning an unassigned client to a server. If a client c is selected to be assigned to a server s, then all unassigned clients that are not farther from s than c are also assigned to s as this would not increase the maximum interaction path length. To minimize the amortized increase in the maximum interaction path length, we use _l=_n as the cost metric for selecting which client to be assigned to which server. In each step, among all possible pairs of unassigned client and server, the pair ðc; sÞ resulting in the minimum cost _l=_n is selected and the corresponding clients are then assigned to s. The algorithm terminates when all clients have been assigned to servers.

Distributed-Modify Assignment:-

The third algorithm Distributed-Modify Assignment is performed in a distributed manner without requiring the global knowledge of the network at any single server. It starts with an initial assignment. Then, the assignment is continuously modified for reducing the maximum interaction path length D until it cannot be further reduced. This process is referred to as the assignment modification. One server is elected as a coordinator responsible for calculating D and selecting the server to perform the assignment modification. To calculate D of the initial assignment, each server measures its distances (network latencies) to all the other servers. It also measures its distances to all the clients that are assigned to it and maintain them as a sorted list. Then, each server s broadcasts to all the other servers its longest distance lðsÞ to its clients, and sends the interserver distances to the coordinator. The coordinator calculates D based on the received information.

Dealing with Server Capacity Constraints:-

So far, our proposed assignment algorithms have not assumed any capacity limitation at the servers. These "uncapacitated" algorithms are suitable for the scenario where each server site has abundant server resources or server resources can be added to these sites as needed . However, if the server capacity at each site is limited, assigning more clients to a server than its capacity may result in significant increase in the processing delay at the server, damaging the interactivity of the DIA. Therefore, we now discuss how to adapt each proposed assignment algorithm to deal with server capacity constraints.

- Nearest-Server Assignment: Each client chooses its server and makes the request to connect to the server independently. Each server accepts the client requests on a first-come-first-serve basis until it is saturated. A

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

client first attempts to choose the nearest server. If the nearest server is saturated, the client in turn tries the second nearest server, the third nearest server and so on, until its connection request is accepted by a server.

- Greedy Assignment: When selecting the pair of unassigned client and server in each step, the algorithm considers unsaturated servers only. After a client c is selected to be assigned to a server s in a step, if the algorithm cannot assign to server s all clients closer to s than c due to the capacity constraint of s, only a portion of these clients are assigned to server s to fill it to capacity. Accordingly, the calculation of _n is adjusted to reflect the capacity limitations of the servers.
- Distributed-Modify Assignment: At each assignment modification, a client is allowed to be reassigned to unsaturated servers only.

The approximation ratios previously analyzed for "uncapacitated" assignment algorithms are not applicable to "capacitated" assignment algorithms. Distributed-Modify Assignment has unbounded approximation ratio even without server capacity limitation. Thus, the same is also true when the server capacity is limited." Nearest-Server Assignment and the "capacitated" Greedy Assignment, respectively, when assuming that the network latency satisfies the triangle inequality. Please refer to Appendices G and H of the online supplemental material for the detailed proofs.

CONCLUSION

In distributed interactive applications, each client is connected to one of the servers and pushes/retrieves updates in the system through their connected servers. Thus, any interaction between two clients consists of both client to server latency and inter-server latency which is called an interaction path. Our objective is to minimize the maximum of these interaction paths between any of the client pairs in the system. Previous works, that addressed the same problem, all considered a static system with previously calculated Internet latency values. The problem is proven to be NP-complete. Three heuristic assignment algorithms are presented. Their approximation ratios are theoretically analyzed and their performance is experimentally evaluated using real Internet latency data. The results show that: 1) our proposed Greedy Assignment and Distributed-Modify Assignment algorithms significantly outperform the intuitive Nearest-Server Assignment algorithm; 2) Distributed-Modify Assignment requires only a small proportion of clients to perform assignment modifications for improving interactivity; and 3) Distributed-Modify Assignment has good adaptivity to dynamics in both client participation and network latency.

The interaction path from a client ci to another client cj can be considered as a directed path that is different from the interaction path from cj to ci. It is easy to show that if we change the definition of D to be the maximum length of all the directed interaction paths between clients, the consistency and fairness requirements can still be satisfied. herefore, the objective of the client assignment problem becomes to minimize the maximum length of all directed interaction paths. For the heuristic algorithms, we can simply use the lengths of the directed routing paths between clients and servers in the calculation without modifying the algorithms. However, the approximation ratios of the algorithms may change. We leave the detailed analysis to the future work.

REFERENCES

[1] "LotRO server list," http://lotro-wiki.com/index.php/List_ of_Worlds, 2013.
[2] "Planetlab All-Pairs-Pings," http://pdos.lcs.mit.edu/strib/, 2013.
[3] WoW Server List, http://www.wowwiki.com/Realms_list, 2013.
[4] L.D. Bricen˜ o, H.J. Siegel, A.A. Maciejewski, Y. Hong, B. Lock, M.N. Teli, F. Wedyan, C. Panaccione, C. Klumph, K. Willman, and C. Zhang, "Robust Resource Allocation in a Massive Multiplayer Online Gaming Environment," Proc. Fourth Int'l Conf. Foundations of Digital Games, pp. 232-239, 2009.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

[5] J. Brun, F. Safaei, and P. Boustead, "Managing Latency and Fairness in Networked Games," Comm. ACM, vol. 49, no. 11, pp. 46-51, 2006.
[6] E. Cronin, B. Filstrup, and A. Kurc, "A Distributed Multiplayer Game Server System," technical report, Univ. of Michigan, 2001.
[7] E. Cronin, S. Jamin, C. Jin, A.R. Kurc, D. Raz, and Y. Shavitt, "Constrained Mirror Placement on the Internet," IEEE J. Selected Areas Comm., vol. 20, no. 7, pp. 1369-1382, Sept. 2002.
[8] E. Cronin, A.R. Kurc, B. Filstrup, and S. Jamin, "An Efficient Synchronization Mechanism for Mirrored Game Architectures," Multimedia Tools and Applications, vol. 23, no. 1, pp. 7-30, 2004.
[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 NP-Completeness," 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.
[16] K.W. Lee, B.J. Ko, and S. Calo, "Adaptive Server Selection for Large Scale Interactive Online Games," Computer Networks, vol. 49, no. 1, pp. 84-102, 2005.
[17] Y.J. Lin, K. Guo, and S. Paul, "Sync-MS: Synchronized Messaging Service for Real-Time Multi-Player Distributed Games," Proc. IEEE 10th Int'l Conf. Network Protocols (ICNP '02), 2002.
[18] C. Lumezanu, R. Baden, N. Spring, and B. Bhattacharjee, "Triangle Inequality and Routing Policy Violations in the Internet," Proc. 10th Int'l Conf. Passive and Active Network Measurement (PAM '09), pp. 45-54, 2009.
[19] M. Marzolla, S. Ferretti, and G. D'Angelo, "Dynamic Resource Provisioning for Cloud-Based Gaming Infrastructures," ACM Computers in Entertainment, to be published.
[20] M. Mauve, J. Vogel, V. Hilt, and W. Effelsberg, "Local-Lag and Timewarp: Providing Consistency for Replicated Continuous Applications," IEEE Trans. Multimedia, vol. 6, no. 1, pp. 47-57, Feb. 2004.
[21] L. Qiu, V.N. Padmanabhan, and G.M. Voelker, "On the Placement of Web Server Replicas," Proc. IEEE INFOCOM '01, pp. 1587-1596. 2001.