International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 11, November-2014

Different Server Architecture for Mobile Presence Services

Dipti Madankar
Comp. Network
Smt. Kashibai Navale College of Engg.
Pune, India

Prof. A. A. Deshmukh
Comp. Network
Smt. Kashibai Navale College of Engg.
Pune, India

Abstract: **In today's internet world use of Smartphones and PDAs are increase enormously which leads to the use of internet communication by the means of instant messaging which is the crucial part of the online social networking. This paper focus on different server architectures needed for MPS. MPS are the services like facebook, AIM, Microsoft Msn etc. which provide the each and every update of the friends of an user who is using the service like log in/log out ,recent updation made by friends to user and user to every other friend listed in user's friend list, different activities etc. This paper enlists the different architectures for MPS and their brief comparison along with features and flaws.**

Index Term: Mobile presence services (MPS), Instant Messaging (IM).

I. INTRODUCTION

Because of the ubiquitous computing use of presence enabled applications like Facebook [1], AIM [6], Microsoft Msn [6], YMsg [6], GTalk, and Skype [7] have grown up rapidly. Because of the large availability of Smartphones and PDA which allow using wireless network technology, it is possible to share experiences on social network across the large distances. MPS are the services which provide presence enabled information of every user of it like availability, log in /log out, recent activities, current location. This information is very useful and responses of these services to such queries are very quick. In every such application there is a friend list called buddy list. These buddies are able to see the user activation status and all updates made by him/her and vice versa.

In this article we present an overview of different server architecture on which instant messaging [5] depends upon. As in last decade use of instant messaging increases, millions of users registered to instant messaging application. Every year there is a great hike seen in the use of these applications. One can say that it becomes a trend to communicate via instant messaging, which is very quick, cheap, multitasking and easy to understand. The only requirement for instant messaging services (IMS) is availability of internet and literacy. If we look over graph of Users increased within a decade we also can conclude that there are number of applications available in market which provide IMS. Every application comes with some changes in its technology or server architecture. It is due to market competition which leads to frequent changes in technology. We focus on these changes related to server architecture. We started with basic client-server architecture and ended to quorum based architecture. Every technology has two faces good and bad. We discussed both aspects and also described the ways to overcome flaws of every architecture.

A brief review of the paper is presented as follows. Next section focuses on related work. In section 3, we described a brief history and background of instant messaging. Section 4 contains different service architecture. In section 5 we conclude the paper.

II. RELATED WORK

In this section a brief description of previous work in this area is given. Aforementioned services in this area are AIMS, YMSG, and Microsoft MSN. Jenning III [6] presented functionality of these services with its client-server architecture. Zhen Xiao [8] described instant messaging traffic characterization. Skype[7] uses Global Index technology which is based on multi-tired network .As there is very less data available on Skype it is difficult to explain detailed view on Global Index technology. GTalk and Facebook [1] are the best example of distributed environment. In Ion Stoica's [12] Chord, working of distributed architecture explained with the help of distributed hash table. It is also a peer-to-peer network in distributed environment. Gnutella [4] is a decentralized architecture and the main representive of the unstructured P2P system [9] [10]. Jabber [3] is the example of distributed architecture which is more scalable than centralized architecture .Chi-Jen Wu [11] presents quorum based architecture which is better competitor to distributed architecture with great scalability, low search cost and less network communication latency.

III. BACKGROUND OF INSTANT MESSAGING

Aforementioned IMS is a very crucial part of online social networking. Every IM user has unique UserId and password via which he can log in to the IM application. Every user has his friend list termed as buddy list and every friend called as a buddy. Whenever a user logged in to the application his availability notification is send to the member of his buddy list. IM application allows user to exchange text messages, images, file transfer and video too. Beside chat messages IM client also send hint message and presence message. Hint messages and presence messages are the major reason for IM traffic. Chat messages occupy small portion of total traffic. As mentioned previously presence messages include activity updation, notification, location updation, and current status. Whereas hint messaging done at the time of editing of chat messages like typing, read, seen etc.

IV. SERVER ARCHITECTURE

A. Client-Server Architecture

Basic client server architecture [6] provides centralize control over user data and messages. This client server architecture again distilled into Symmetric and Asymmetric. In Symmetric approach (fig.1) there is single server who handles all the data of user. In short single server performs task of authorization server, mail server, buddy icon server, chat room server etc. Whereas in Asymmetric server (fig.2) for every task a different server is allocated. Each server perform different task step by step. Client server architecture has centralized control that's why there is no problem of server authentication. Aforementioned due to central server all the information stored at only one server which leads to the fast discovery of buddies. Hence response time is very less. But with increase in number of user scalability and load balancing issues arises with this architecture. If central server fails in middle of communication complete network will crash and takes time for recovery. It may possible that there is data loss too. AIM, Yahoo messanger and Microsoft MSN are the some examples of this architecture.

B. Decentralized Architecture

In decentralized architecture [4] each server is connected to its neighbor server and buddy search message will send to all neighbor servers. Fig.3 shows the decentralized architecture. Decentralized approach allows random walk. In random walk lastly visited node will be avoid in next iteration while searching the correct node. As buddy search message send to each node, it provide better buddy search options. If one server fails its traffic diverted to its neighbor node hence this architecture is robust to server failure. Along with robustness it has feature like high availability. Due to message flooding

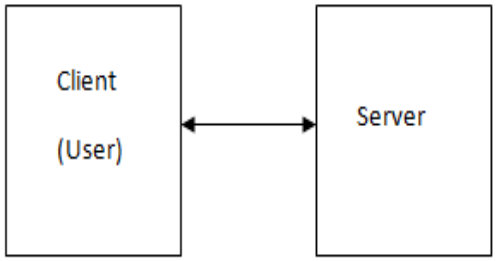

Fig. 1: Client server architecture symmetric approach

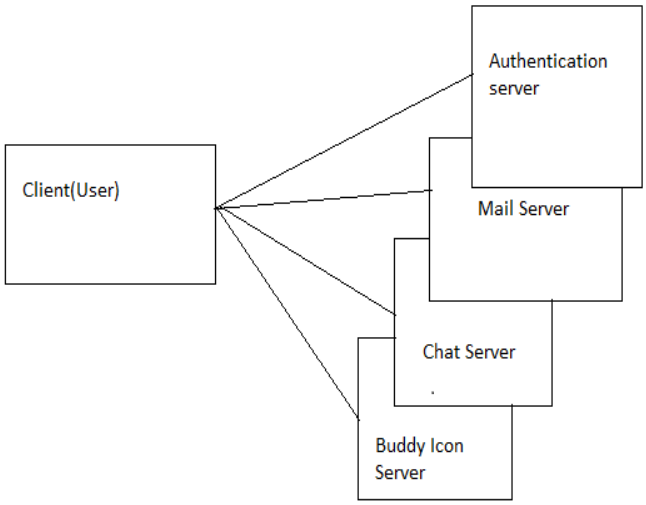

Fig.2: Client server architecture Asymmetric approach

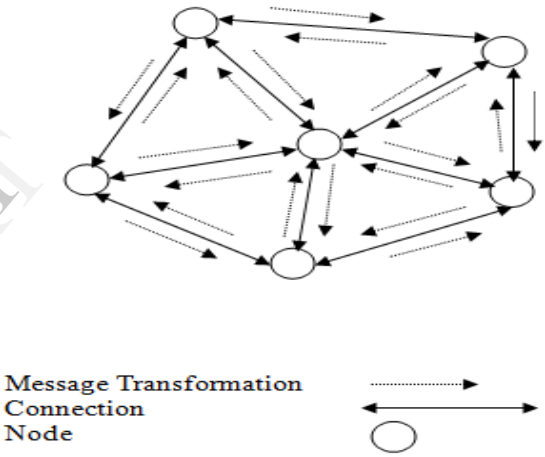

Fig. 3: Decentralized Architecture

there is wastage of bandwidth. As unnecessarily buddy search message send to the all other server. It also leads to message redundancy and hence load at each node linearly grows with size of user. Gnutella is first decentralized peer-to-peer network use for file sharing.

C. Distributed Architecture

Distributed architecture uses DTH i.e. distributed hash table [12] for connecting different nodes. DTH used to locate key-based objects over millions of hosts on the internet. Distributed approach provides scalable, flexible, efficient solution over flaws of centralized and decentralized approach. Along with this proper load balancing and availability are also the advantages of distributed approach. It overcomes issue of message redundancy at some extent. But still it is not enable to provide full user satisfaction and server authentication. Its look up cost is high compared to others and it is not that fast as centralized and decentralized. Facebook,

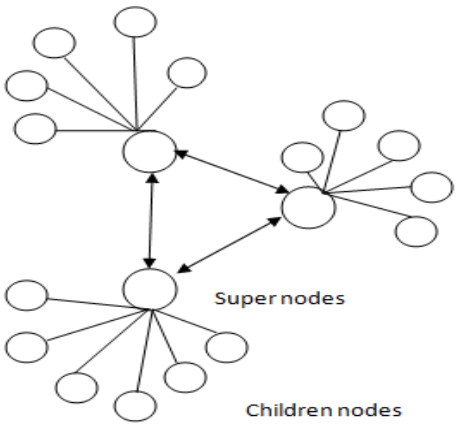

Fig. 4: Multi-Tired Architecture

GTalk are the applications uses distributed approach. GTalk is based upon Jabber [3] technology.

D. Multi-Tired Architecture

In multi-tired approach [7] there are super nodes and children nodes connected to super nodes. Fig. 4 shows the multi-tired architecture. Super nodes are powerful nodes which act as index servers and client queries are propagated through super nodes. Each super node carries 100 to 150 children and connected to 30 to 35 other super nodes. Multi-Tired approach provides more scalability than centralized approach and faster look-up than decentralized approach. Nodes which have more bandwidth and whose availability is more will act as super nodes. Each super node and its children are like mini centralized structure. In this approach if data is at multiple nodes then user can do parallel downloading. Parallel downloading here refer to the different portion of file from the different nodes. To identified identical copies of data concept of content hash is used. By same concept automatic recovery also possible when one super node becomes inactive or stop sending data or files. Hence multi-tired approach provide high performance. But there are security issues and user satisfaction is less. Multi-Tired architecture also called as hierarchical architecture.

E. Quorum-Based Architecture.

The main motto of this architecture is to provide scalability with efficient search of buddies over the internet. For this it uses direct search algorithm and one hop caching [11]. This architecture provide low search cost and minimum search latency along with less storage. There are three components of architecture:

- Presence Cloud server overlay
- One-hop cache
- Direct search

I would like to thank all of the staff members and my friends for their support and kindness towards me.

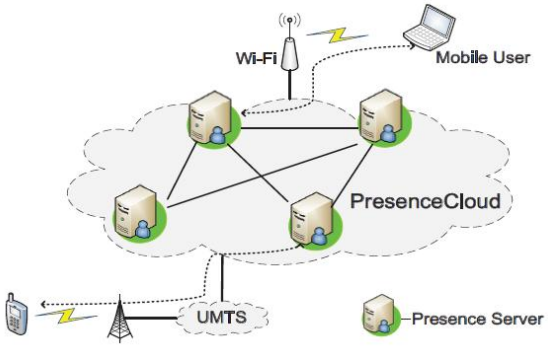

Fig. 5: Presence Cloud

Presence Cloud server overlay in fig. 5 organizes server into quorum i.e. into groups which allow diameter of 2 hops. One hop cache decreases number of transmitted messages and increases query accessing speed. Direct search ensures one hop search which helps to maintain low search latency. Hence quorum based architecture provide high scalability and performance even with increase in number of users performance does not affect adversely. But this architecture cannot provide efficient security which includes end user security and server authentication problem due to which this architecture is vulnerable to Man-In-Middle (MIM) attack and denial of service (DOS) attack. We can protect MIM by using end-to-end private key encryption technique. Because no authentication is provided to server, any server can choke its neighbor server by sending buddy search messages. This is DOS attack, to prevent this we can use distributed trust model [13].

V. CONCLUSION

In this paper, we have presented a brief introduction to different server architecture that supports to mobile presence services. Aforementioned traditional client-server model does not support large scale network. Whereas decentralized model overcomes issues related to client server but it leads to bandwidth wastage due to message redundancy. Distributed model use hash table to look up the nodes with précised use of bandwidth. Multi-tired architecture allow parallel downloading while quorum based server architecture provide high scalability, low search cost and less network latency. But all of three are not as fast as centralized and decentralized .Also security and user satisfaction level does not alleviate in these technologies.

ACKNOWLEDGMENT

In stern guidance of my guide Prof. A.A.Deshmukh who has bought my work in proper channel. Her guidance not only helped me but also boost my confidence which will be helpful to me in my future. I sincerely thanks to my guide.

REFERENCES

[1] Facebook, http://www.facebook.com
[2] Google latitude , http://www.google.com/intl/ensu/latitud/ intro.html
[3] Jabber, http://www.jabber.org.
[4] rfc-gnutella.sourceforge.net
[5] Mobile instant messaging, http://en.wikipedia.org/ wiki/Mobile _Instant_ messaging.
[6] R. B. Jennings, E. M. Nahum, D. P. Olshefski, D. Saha, Z.-Y. Shae, and C. Waters, "A study of internet instant messaging and chat protocols," IEEE Network, 2006
[7] Gobalindex, http://www.skype.com/intl/en-us/support/user-guides/ p2pexplained/.
[8] Z. Xiao, L. Guo, and J. Tracey,"Understanding instant messaging traffic characteristics," Proc. of IEEE ICDCS, 2007.
[9] Peer-to-peer session initiation protocol ietf working group, http://www.ietf.org/html.charters/p2psip-charter.html.
[10] K. Singh and H. Schulzrinne,"Peer-to-peer internet telephony using sip," Proc. of ACM NOSSDVA, 2005.
[11] Chi-Jen Wu, Jan-Ming Ho, Ming-Syan Chen, "Scalable server architecture for mobile presence services,"IEEE Trans. on Mobile Computing,2013
[12] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Bala krishnan,"Chord: A scalable peer-to-peer lookup service for internet,"IEEE/ACM Tran. on Networking, 2003.
[13] A. Abdul-Rahman and S. Hailes.,"A distributed trust model," Proc. of the workshop on New security paradigms, 1997.