International Journal of Research

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 07, July 2015
Available at http://internationaljournalofresearch.org

A Scalable Server Architecture for Mobile Presence Services support large scale Social Network Application

Noorbhasha Nabi & T G.Ramnadh Babu

[1]M.Tech (CSE), Department of Computer Science & Engineering
2Assistent Professor, Department of Computer Science & Engineering.
E-mail id: baburamnadh@gmail.com

Abstract-

Now a day's Social network applications are becoming increasingly popular on mobile devices. There are various mobile presence services for social network applications. Social networking services on the Internet are growing and increasing numbers of people are using these new ways to communicate and share information. Many users are communicating with both friends from outside the service as well as with people they have only been in contact with through a social networking service. A mobile existence overhaul is a necessary constituent of a social network application as it uphold each mobile user's presence information such as the current status, GPS location and network address. Also updates the user's online friends with the information repeatedly. If presence updates happen frequently, the massive number of messages dispersed by presence servers may lead to a scalability issue in a large-scale mobile presence service. Presence is a service that allows a user to be informed about the reachability, availability, and willingness to communicate of another user. Mobile presence service is a vital component of a social network applications due to mobile user's presence details such as positioning system location, network address, and online/offline status are continuously apprise to user's online buddies. If persistent updates occur about presence information and also when servers disperse expansive number of messages scalability problem arises in large scale mobile presence service. A presence cloud which is scalable server architecture is proposed which support large scale social network application where it searches for the presence of his/her buddies and reveals them about his/her arrival. Presence Cloud organizes presence servers into a quorum-based server-to-server architecture for efficient presence searching. It also leverages a directed search algorithm and a one-hop caching strategy to achieve small constant search latency.

Keywords- Presence Cloud; Social network service; Distributed presence servers

1. INTRODUCTION

Presence enabled applications can be provided by mobile devices and cloud computing environments, i.e., social network applications/ services, worldwide. From the last decade, presence enabled applications are growing rapidly. Some of the examples are Facebook, Twitter, Foursquare, Google Latitude, buddy cloud and Mobile Instant Messaging (MIM). In cloud computing environment mobile presence service is a vital component of social network application. Presence information tells the detail about mobile user's availability, activity and machine capacity. Service does binding of user id to his/her current presence information details. Each individual mobile user has a buddy list which includes details of whom he/she wants to interact with in social network services.

When a user does shipment from one level to other, this change is instinctively transmitted to each individual on the buddy list. Server cluster technology increases the search speed and decrease the report time. For example in social network application mobile user logs in through his/her mobile device, the mobile presence services searches and reveals each of them about user's friend list such as instant messaging system. Potential of presence cloud can be examined by using search cost and search satisfaction without impaired neither of them. When a user arrives presence server provoke a number of messages is search cost. Time it takes to examine the arrival of user's buddy list is search satisfaction. To help the users who are present worldwide, the services enhanced by Google and Facebook are proliferated among many servers. Presence server used in large scale social network services to ameliorate the coherence of mobile presence services.

Billion shared items every month and Twitter receives more than55million tweets each day. In the future, mobile devices will become more powerful, sensing, and media capture devices. Hence, we believe it is inevitable that social network services will be the next generation of mobile Internet applications. A mobile presence service is an essential component of social network services in cloud computing environments. The key function of a mobile presence service is to maintain up-to-date list of presence information of all mobile users. The presence information includes details about a mobile user's location, availability, activity, device capability, and preferences. The service must also bind the user's ID to his/her current presence information, as well as retrieve and subscribe to changes in the presence information of the user's friends. In social network services, each mobile user has a friend list, typically called a buddy list, which contains the contact information of other users that he/she wants to communicate with. The existing server architecture for buddy list in large scale geographical information Centre. Overloading of buddy search message on presence server leads to scalability problem. Presence cloud disseminates many users' information details among many presence servers on the internet, which is used as a building block of mobile presence services. For efficient buddy list search there is no single point collapse, since servers in presence cloud are organized in quorum based server to server architecture to gain small search delay using directed buddy search algorithm. Caching procedure is used to reduce buddy list search. The potential of three architectures such as presence cloud, mesh based scheme and distributed hash table are examined in terms of search response time and friend notification time.

2. RELATED WORK

In this section, we describe previous researches on presence services, and survey the presence service of Existing systems. Several IETF charters have addressed closely related topics and many RFC documents on instant messaging and presence services (IMPS) have been published, e.g., XMPP, SIMPLE. Jabber is a well-known deployment of instant messaging technologies based on distributed architectures. It captures the distributed architecture of SMTP protocols. Since Jabbers architecture is distributed, the result is a flexible network of servers that can be scaled much higher than the monolithic, centralized presence services. Recently, there is an increase amount of interest in how to design a peer-to-peer SIP. P2P-SIP has been proposed to remove the centralized server, minimize maintenance costs, and keep from happening failures in server based SIP deployment. To maintain presence information, P2P-SIP clients are organized in a DHT system, rather than in a centralized server.

3. PRESENCE CLOUD

The past few years has seen a veritable frenzy of research activity in Internet scale object searching field, with many designed protocols and proposed algorithms. Most of the previous algorithms are used to address the fixed object searching problem in distributed systems for different intentions. However, people are nomadic, the mobile presence information is more mutable and dynamic; anew design of mobile presence services is needed to address the buddy list search problem, especially for the demand of mobile social network applications. Presence Cloud is used to construct and maintain distributed server architecture and can be used to efficiently query the system for buddy list searches. Presence Cloud consists of three main components that are run across a set of presence servers. In the design of Presence Cloud, the ideas of P2P systems and present a particular design for mobile presence services has been refined. Presence Cloud is used to construct and maintain distributed server architecture and can be used to efficiently query the system for buddy list searches. Presence Cloud consists of three main components that are run across a set of presence servers. In the design of Presence Cloud, we refine the ideas of P2P systems and present a particular design for mobile presence services.

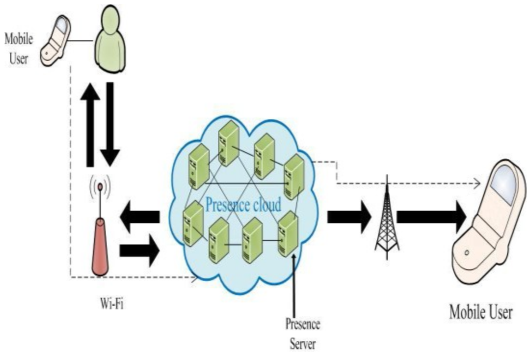


Fig.1. Presence Cloud Architecture

Presence servers which are present in presence cloud, where these presence servers are arranged in quorum based server to server architecture and also load on servers are balance in presence cloud sever overlay.

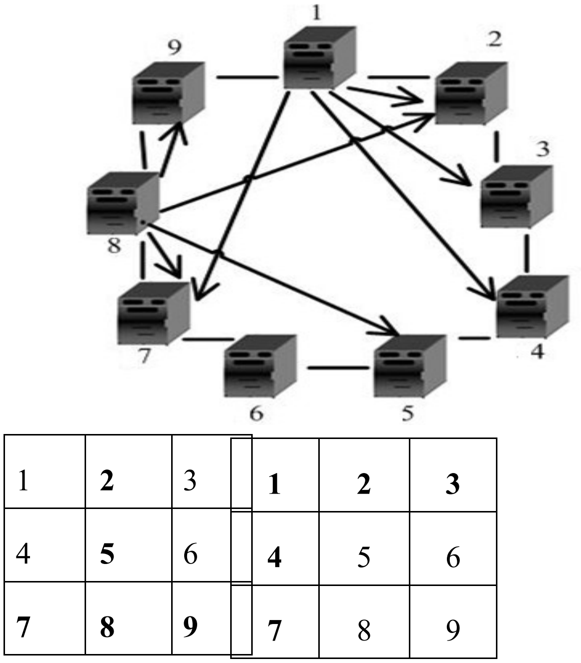
All these presence server keeps caches for buddies in order to increase query speed is one hop caching approach.

Finding small constant search delay results in directed buddy search by decreasing network traffic using one hop search strategy.

Architecture of presence cloud is shown in Figure1, Using 3G or Wi-Fi services mobile user access the internet and make a data link to the presence cloud. Using secure hash algorithm mobile users are intent to one of the presence servers. To transfer presence information details, the mobile user is authenticated to the mobile presence services and also opens a TCP link. Once path is set up, the mobile user request for the friend list to the presence server which is present in presence cloud. And finally the request is responded by the presence cloud after completing an efficient search of buddy’s presence information. The three key components of Presence Cloud are summarized below:

A. PRESENCE CLOUD SERVER OVERLAY

The Presence Cloud server overlay construction organizes the Presence server nodes into a server-to-server overlay, which provides a good low-diameter overlay property. Needs two hops to reach from one presence server node to other presence server node is the possession of low diameter and Presence cloud is based on grid quorum system. Size of presence server node is O √m, where m is the number of presence server in mobile presence services. By using grid quorum system presence server list is built and this presence server list maintains presence server node which has a set of presence server nodes.



1	**2**	3
4	**5**	6
7	**8**	**9**

1	**2**	**3**
4	5	6
7	8	9

Fig.2. Presence Cloud Server Overlay

For example in Fig 2, grid quorum is set to √9*√9.Presence server node 8 has a presence server list {2, 5, 7, 9} and presence server node a1 has presence server list {2, 3, 4, 7}. Thus presence server node a8 and a1 can built their overlay network according to their presence server list.

B. One Hop Caching

Presence Cloud requires a caching strategy to make an exact copy of presence information of users to improve the efficiency of the search operation. In order to adapt to changes in the presence of users, the caching strategy should be asynchronous and not require expensive mechanisms for distributed agreement. To duplicate the presence information details presence cloud requires caching strategy in order to enhance the efficiency of search operation. In presence cloud for the attached users, presence information details of user list are maintained by presence server node. Duplicating user list by presence server nodes are at most one hop away from itself. When connection is proven by neighbor's cache is updated and also updated periodically with their neighbors. If query accepted by presence server node it is not only respond with matches from cache where user list available by its neighbors.

Presence information changes for mobile users when user leaves presence cloud or due to failure. Response from presence server node broad casts its new presence to other neighboring presence server node for updates. Presence information remains constant and up to date throughout the session time of user is ensured by one hop caching strategy. Consequently, this one hop caching strategy ensures that the user's presence information could remain mostly up to date and consistent throughout the session time of the user. More specifically, it should be easy to see that, each node maintains roughly $2\ (\lceil \sqrt{n} \rceil - 1)$ x u duplicates of presence information, due to each node duplicates its user list at most one hop away from itself. Here, u is denoted the average number of mobile users in a node.

C. Directed Buddy Search

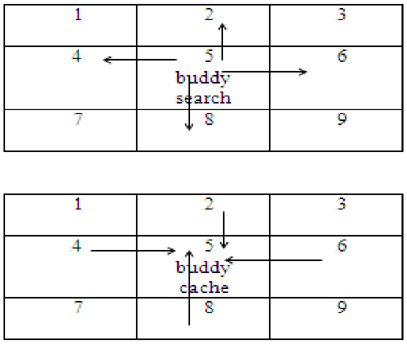


Fig.3. Buddy list search in Presence Cloud

Fig 3 shows, for mobile presence services it is important to reduce search time. Using two hop overlay and one hop caching strategy presence cloud endow response for large number of mobile users. One hop search used for queries in order to reduce network traffic one hop caching maintains user list of its neighbors to enhance response time by increasing in finding buddies.

System architecture for social network application using mobile presence services is shown in Fig.4, setting overall outlook of application in service layer where user registers for presence cloud and updates his/her profiler. In persistence layer performing CURDS operation such as create, read, update and delete. Presence cloud searches for presence information details such as online/offline, network address, global positioning system location when user joins the network.

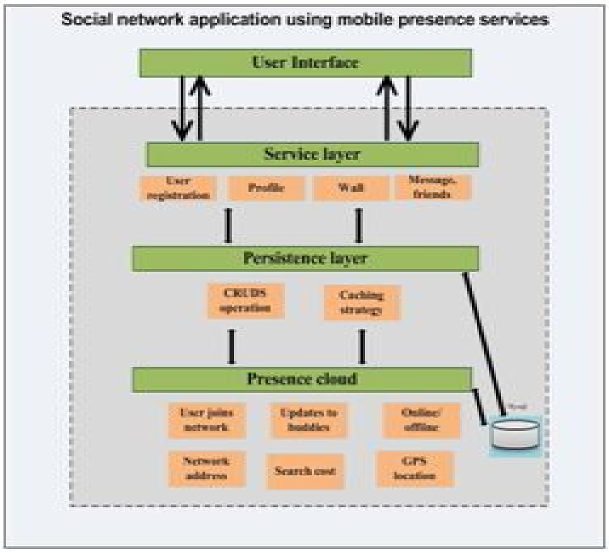


Fig.4. System Architecture for Mobile presence service in social network application

4. IMPLEMENTATION

In Fig.5, the process flowchart of the implementation is shown. The proposed system is implemented in 32 bit Windows operating system with 1 GHz Processor and 1GB RAM .The design environment is selected in java. User registers for presence cloud if it is success then he/she can view the friend list and send a request to the presence server which is present in presence cloud for buddy list search. Presence cloud makes an efficient search of buddy list and returns the result to the user. Then user send request to buddies using friend id if accepted then added to buddy list, otherwise buddies can ignore the request. Buddies send messages to their friend and also can view message, compose messages for their buddies using friend id and finally navigates to the main page.

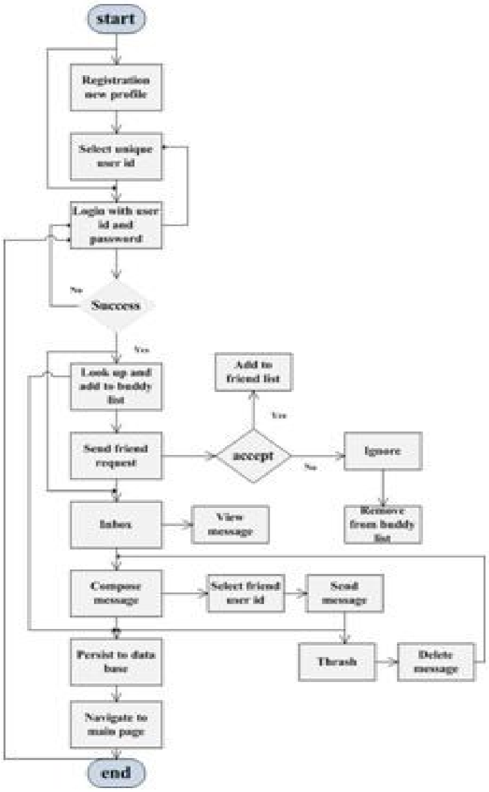


Fig.5. Process flowchart implementation

5. CONCLUSION

The scalable server architecture called as presence cloud is supported by many large scale social network mobile presence services. Total number of buddy search messages increases substantially with the user arrival rate and the number of presence servers. The growth of social network applications and mobile device computing capacity to explore the user satisfaction both on mobile presence services or mobile devices. Presence Cloud could certificate the presence server every time when the presence server joins to Presence Cloud.

6. REFERENCES

[1.] K.Peternel, L.Zebee and A.Kos, "Using Presence information for an effective Collaboration".Open Mobile Alliance, "OMA Instant Messaging and Presence Service," 2005

[2.] Z. Xiao, L. Guo, and J. Tracey, "Understanding Instant Messaging Traffic Characteristics," Proc. IEEE 27th Int'l Conf. Distributed Computing Systems (ICDCS), 2007

[3.] ChetanS, GautamKumar, K.Dinesh, MathewK, andAbhimanyaM.A,"Cloud computing for Mobile world". Google, http://www.google.com.

[4.] Chi, R. Hao, D. Wang, and Z.Z. Cao, "IMS Presence Server: a Traffic Analysis and Performance Modeling," Proc. IEEE Int'Conf. Net work Protocols (ICNP), 2008

[5.] Kundan Singh and Henning Schulzrinne Department of Computer Science, Columbia University {kns10,hgs}@ cs.columbia.edu, "SIPPEER : A session iniiation protocol (SIP)-baed peer-to-peer internet telephony client adaptor".

[6.] Jianming Zhao, Nianmin yao, Shaobin Cai, and Xiang li, "Tree-Mesh Based P2P Streaming data distribution schema", 2012.

[7.] Distributed hash tables http://www.cs.cmu.edu/~dga/15-744/S07/lectures /16-dht.pdf.

[8.] P. Saint Andre, "Interdomain Presence Scaling Analysis for the Extensible Messaging and Presence Protocol (XMPP)," a IETF Internet draft, 2008.

[9.] Brent Hecht, Jaime Teevan, Meredith Ringel Morris, and Dan Liebling, "Search Buddies: Bringing Search Engines into the Conversation", 2012

[10.] Michael Piatek, Tomas Isdal, Arvind Krishnamurthy, and Thomas Anderson "One hop Reputations for Peer to Peer File Sharing Workloads". Quorumbasedtechniqueshttp://www.en. wikipidiaorg/wiki/Quorum_ (distributed_computing).

[11.] D.Dhiravida vasantha nayaki and G.Mangayarkarasi,"Presence server for mobile ubiquity services in presence cloud,"March 2014.

[12.] Christian Voigt, Sandra Barker, Sharron King, Kit Macfarlane, Tim Sawyer & Sheila Scutter, "Conceptualising social networking capabilities: Connections, objects, power and affect", 2010.