Presence Cloud - a scalable mobile presence service in control message transmission

Mahesh V. Shastri
Computer Engineering Department
Pdm. Dr. V. B. Kolte College of Engineering,
Malkapur, India
maheshvshastri@gmail.com

Sachin B. Jadhav
Computer Engineering Department
Pdm. Dr. V. B. Kolte College of Engineering,
Malkapur, India
sachin.bjadhav@gmail.com

Abstract

Today Social networking services on the Internet are growing and increasing number of people are using these new ways to communicate and share information. At the same time mobile phones are becoming more powerful and increasingly offer high speed Internet connectivity. Because of this people expect these social networking services to be available on their mobile device. To maintain mobile user's presence information such as the current status (online/offline), GPS location and network address, and also updates the user's online friends with the information continually the mobile presence service is essential component. So when that presence update occurs frequently the scalability problem occurs at server side to resolve this problem we proposed scalable server architecture called as presence cloud, which enables mobile presence services to support large-scale social network applications. When a mobile user joins a network, Presence Cloud searches for the presence of his/her friends and notifies them of his/her arrival. Presence Cloud organizes presence servers into a quorum-based server-to-server architecture for efficient presence searching. It also efficiently uses a directed search algorithm and a one-hop caching strategy to achieve small constant search latency. We analyze the performance of Presence Cloud in terms of the search cost and search satisfaction level. The search cost is defined as the total number of messages generated by the presence server when a user arrives; and search satisfaction level is defined as the time it takes to search for the arriving user's friend list.

Keywords- mobile presence services, distributed presence servers, cloud computing.

I. INTRODUCTION

Cloud computing techniques for storage and processing of data on mobile devices, thereby reducing their limitations. According to ABI Research, "By 2015, more than 240 million business customers will be leveraging cloud computing services through mobile devices, driving revenues of $5.2 billion". While it must be noted that there were only 42.8 million Mobile Cloud Computing subscribers in 2008 [10].This underlines the importance of cloud computing for mobile. The end mobile device user will eventually be the benefactor of the Mobile Cloud Computing. Presence information [1] entitled applications such as Facebook, Twitter etc., which is produced by mobile devices and cloud computing [2] using internet [3].Way the members are engaged with their buddies on internet are changed by the social network services [4]. In order to interact with buddies across great distance participants can dispense the live event immediately using their mobile device. Mobile user's presence information details will be maintained by mobile presence service [5]. In cloud computing environment mobile presence service is a very important component of social network application. Presence information tells the detail like current status (online/offline), GPS location and network address**.** Service does binding of user id to his/her current presence information details. Each individual mobile user has a Friends 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 [6]. Potential of presence cloud [5] [7] 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 [3] [8] and Facebook [3] are proliferated among many servers. Presence server used in large scale social network services to ameliorate the coherence of mobile presence services.

In this section, examine 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 [9] 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 [10] based scheme and distributed hash table [11] are examined in terms of search response time and friend notification time.

II. EXISTING SYSTEM

In this section, we describe previous researches on presence services, and survey the presence service of existing systems. Well known commercial IM systems leverage some form of centralized clusters to provide presence services. Jennings III et al. presented taxonomy of different features and functions supported by the three most popular IM systems, AIM, Microsoft MSN and Yahoo! Messenger. The authors also provided an overview of the system architectures and observed that the systems use client-server-based architectures. Skype, a popular voice over IP application, utilizes the Global Index (GI) technology to provide a presence service for users. GI is a multi-tiered network architecture where each node maintains full knowledge of all available users. Since Skype is not an open protocol, it is difficult to determine how GI technology is used exactly. Moreover, Xiao et al. analyzed the traffic of MSN and AIM system. They found that the presence information is one of most messaging traffic in instant messaging systems. In, authors shown that the largest message traffic in existing presence services is buddy NOTIFY messages.

Here in the existing system when mobile users joins to a social network application such as an IM system through mobile device the mobile presence service look for and informs everyone on the user's buddy list. To make the most of a mobile presence service's search speed and reduce the notification time most presence services use server cluster technology. Presently more than 2.13 billion social network users around the globe on the Internet. Given the expansion of social network applications and mobile network capacity it is usual that the number of mobile presence service users will augment considerably.

III. PROPOSED SYSTEM

Aim of proposed system is to design architecture A scalable server-to-server architecture that can be used for mobile presence services. Presence Cloud is to allocate the information of millions of users in thousands of presence servers on the Internet. To Evade single point of failure no single presence server is supposed to maintain service-wide global information about all users. Presence Cloud categorizes presence servers into a quorum-based server-to-server architecture to assist proficient buddy list searching. It also leverages the server overlay and a directed buddy search algorithm to attain small constant search latency and employs a lively caching scheme that considerably decreases the number of messages generated by each search for a list of buddies.

In this a scalable server architecture called Presence cloud which provides services to 'n' number of users is presented. And presenting a precise design by improving the thought of peer to peer [12] [13] system while designing presence cloud. There are 3elements in presence cloud which run across presence servers such as presence cloud server overlay, one hop [14] caching approach, and directed buddy search [5].

Recently, there is an increase amount of interest in how to design a peer-to-peer SIP. Peer-to-peer SIP has been proposed to remove the centralized server, reduce maintenance costs, and prevent failures in server-based SIP deployment. To maintain presence information, Peer-to-peer SIP clients are organized in a Directed hash table system, rather than in a centralized server. However, the presence service architectures of Jabber[15] and Peer-to-peer SIP are distributed, the buddy-list search problem we defined later also could affect such distributed systems. It is noted that few articles in discuss the scalability issues of the distributed presence server architecture. Saint Andre analyzes the traffic generated as a result of presence information between users of inter-domains that support the XMPP[14]. Houri et al. Show that the amount of presence traffic in SIMPLE can be extremely heavy, and they analyze the effect of a large presence system on the memory and CPU loading. Those works in study related problems and developing an initial set of guidelines for optimizing inter-domain presence traffic and present DHT-based presence server architecture.

Recently, presence services are also integrated into mobile services. For example, 3GPP has defined the integration of presence service into its specification in Universal Mobile Telecommunications Service(UMTS). It is based on SIP protocol, and uses SIMPLE protocol to manage presence information. Recently, some mobile devices also support mobile presence services. For example, the Instant Messaging and Presence Services (IMPS) was developed by the Wireless Village consortium and was united into Open Mobile Alliance (OMA) IMPS in 2005. In, Chen et al. proposed a weakly consistent scheme to reduce the number of updating messages in mobile presence services of IP Multimedia Subsystem (IMS). However, it also suffers scalability problem since it uses a central SIP server to perform presence update of mobile users. In, authors presented the server scalability and distributed management issues in IMS-based presence service.

A. Presence Cloud server overlay

Presence Cloud server overlay organizes presence servers based on the concept of grid quorum system. So, the server overlay of Presence Cloud has a balanced load property and a two-hop diameter with $O(\sqrt{n})$ node degrees, where n is the number of presence servers.

The Presence Cloud server overlay construction algorithm organizes the PS nodes into a server-to-server overlay, which provides a good low-diameter overlay property. The low-diameter property ensures that a PS node only needs two hops to reach any other PS nodes.

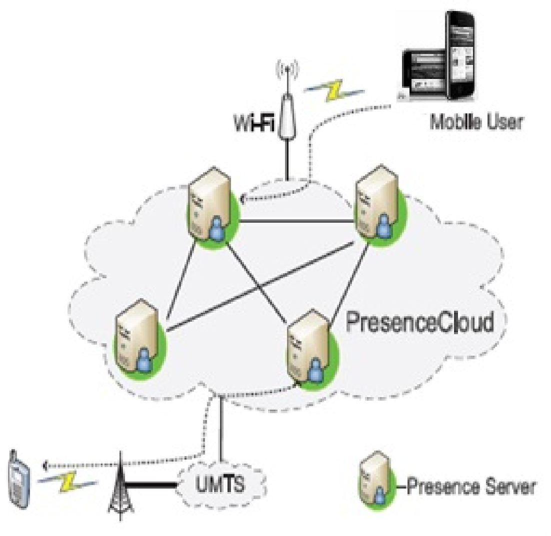

Fig. 1. Architecture for presence cloud

B. One-hop caching strategy

One-hop caching strategy is used to reduce the number of transmitted messages and accelerate query speed. All presence servers maintain caches for the buddies offered by their immediate neighbors.

To improve the efficiency of the search operation, Presence Cloud requires a caching strategy to replicate presence information of users. 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. In Presence Cloud, each PS node maintains a user list of presence information of the attached users, and it is responsible for caching the user list of each node in its PS list, in other words, PS nodes only replicate the user list at most one hop away from itself. The cache is updated when neighbors establish connections to it, and periodically updated with its neighbors. Therefore, when a PS node receives a query, it can respond not only with matches from its own user list, but also provide matches from its caches that are the user lists offered by all of its neighbors.

C. 3. Directed buddy search

Presence Cloud ensures a one-hop search; it yields small constant search latency on average.

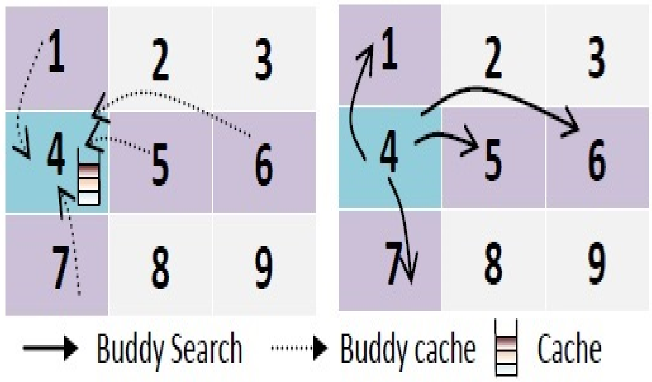

Fig. 2. Buddy list searching operations in Presence Cloud

We insist that minimizing searching response time is important to mobile presence services. Thus, the buddy list searching algorithm of Presence Cloud both with the two-hop overlay and one-hop caching strategy ensures that Presence Cloud can typically provide swift responses for a large number of mobile users. First, by organizing PS nodes in a server-to-server overlay network, we can therefore use one-hop search exactly for queries and thus reduce the network traffic without significant impact on the search results. Second, by capitalizing the one-hop caching that maintains the user lists of its neighbors, we improve response time by increasing the chances of finding buddies. Clearly, this mechanism both reduces the network traffic and response time. Based on the mechanism, the population of mobile users can be retrieved by a broadcasting operation in any PS node in the mobile presence service. Moreover, the broadcasting message can be piggybacked in a buddy search message for saving the cost.

IV. CONCLUSION

In this paper, we have presented Presence Cloud, a scalable server architecture that supports mobile presence services in control message transmission. We discussed the scalability problem in server architecture designs, and introduced the buddy-list search problem, which is a scalability problem in the distributed server architecture of mobile presence services By surveying some papers it is conclude that Presence Cloud achieves performance gains in the search cost without compromising search satisfaction. Overall, Presence Cloud is shown to be a scalable mobile presence service in large-scale social network services.

REFERENCES

[1] Facebook, http://www.facebook.com.

[2] Twitter, http://twitter.com.

[3] Foursquare http://www.foursquare.com.

[4] Googlelatitude, http://www.google.com/intl/enus/latitude/intro.html.

[5] Buddycloud, http://buddycloud.com.

[6] Mobile instant messaging, http://en.wikipedia.org/wiki/Mobile instant messaging.

[7] 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.

[8] Gobalindex, http://www.skype.com/intl/enus/support/guides/p2pexplained /

[9] Z. Xiao, L. Guo, and J. Tracey, "Understanding instant messaging traffic characteristics," Proc. of IEEE ICDCS,2007.

[10] C. Chi, R. Hao, D. Wang, and Z.-Z. Cao, "Ims presence Server: Traffic analysis and performance modelling," Proc. of IEEE ICNP, 2008.

[11] Instant messaging and presence protocol IETF working Group http://www.ietf.org/html.charters/imppcharter.html.

[12] Extensible messaging and presence protocol ietf Working group http://www.ietf.org/html.charters/xmpp charter.html.

[13] Sip for instant messaging and presence leveraging extension IETF working group. http://www.ietf.org/html.charters/simplecharter.html

[14] P. Saint-Andre., "Extensible messaging and presence protocol (xmpp): Instant messaging and presence describes instant messaging (im), the most common application of xmpp," RFC 3921, 2004.

[15] Jabber, http://www.jabber.org/.

[16] C. Chi, R. Hao, D. Wang, and Z.-Z. Cao, "Ims presence server: Traffic analysis and performance modelling," Proc. of IEEE ICNP, 2008.