RESEARCH ARTICLE OPEN ACCESS

An Understanding of Mobile Presence Server with Enhanced Scalability in Presence Cloud

N.Tejesh[1], D.Prasanth[2]
PG Student[1], M.Tech. Assistant Professor[2]
Department of Computer Science and Engineering,
Audisankara College of Engineering & Technology, Gudur.
AP-India

ABSTRACT
In Mobile presence service is a vital component of a social network applications due to mobile user's presence details such as global positioning system location, network address, and online/offline status are continuously apprise to user's online buddies. A mobile ubiquity services is an important element of cloud computing environments, for the reason it keeps an up-to-date list of presence information of mobile user. If presence updates occur often the number of messages distributed by presence server may lead to scalability problem and buddy list search problem in large-scale mobile presence services. To overcome the scalability problem proposed an efficient and ascendable server architecture called presence cloud. It organizes the presence server into quorum based server-server architecture for efficient searching. When a mobile user joins a network or internet, presence cloud searches the presence information. It also achieves small constant search latency by the directed search algorithm and one-hop caching strategy.
Keywords:- Social networks, mobile presence services, distributed presence servers, cloud computing.

I. INTRODUCTION

The Presence [1] entitled applications such as Face-book, Twitter etc., which is produced by mobile devices and cloud computing [2] nature due to the prevalence of 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 vital component of social network application. Because of the ubiquity of the Internet, mobile devices and cloud computing environments can provide presence-enabled applications, i.e., social network applications/services, worldwide. Face book, Twitter Foursquare Google Latitude, buddy cloud and Mobile Instant Messaging (MIM) are examples of presence-enabled applications that have grown rapidly in the last decade. Social network services are changing the ways in which participants engage with their friends on the Internet. They exploit the information about the status of participants including their appearances and activities to interact with their friends. Moreover, because of the wide availability of mobile devices (e.g., Smartphone's) that utilize wireless mobile network technologies, social network services enable participants to share live experiences instantly across great distances. 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 [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. 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 [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. RELEATED WORK

The rationale behind the design of Presence Cloud is to distribute the information of millions of users among thousands of presence server's on the Internet. To avoid single point of failure, no single presence server is supposed to maintain service-wide global information about all users. Presence Cloud organizes presence servers into a quorum based server-to-server architecture to facilitate efficient buddy list searching. It also leverages the server overlay and a directed buddy search algorithm to achieve small constant search latency and employs an active caching strategy that substantially reduces the number of messages generated by each search for a list of buddies. We analyze the performance complexity of Presence Cloud and two other architectures, a Mesh based scheme and a Distributed Hash Table (DHT)-based scheme. Through simulations, we also compare the performance of the three approaches in terms servers on the Internet. The design of Presence Cloud, a scalable server-to-server architecture that can be used as a building block for mobile presence services. The rationale behind the design of Presence Cloud is to distribute the information of millions of users among thousands of presence servers on the Internet. To avoid single point of failure, no single presence server is supposed to maintain service-wide global information about all users. Presence Cloud organizes presence servers into a quorum-based server-to-server architecture to facilitate efficient buddy list searching.

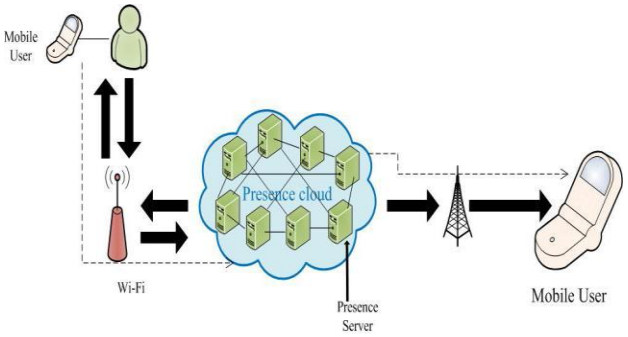

Fig1. Presence Cloud Architecture

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 which is the proposed work 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. Analyse the performance complexity of Presence Cloud and two other architectures, a Mesh based scheme and a Distributed Hash Table (DHT)-based scheme. Through simulations, we also compare the performance of the three approaches in terms of the number of messages generated and the search satisfaction which we use to denote the search response time and the buddy notification time.

The results demonstrate that Presence- Cloud achieves major performance gains in terms of reducing the number of messages without sacrificing search satisfaction. Thus, Presence Cloud can support a large-scale social network service distributed among thousands of servers on the internet. Presence Cloud is among the pioneering architecture for mobile presence services. To the best of our knowledge, this is the first work that explicitly designs a presence server architecture that significantly outperforms those based distributed hash tables. Presence Cloud can also be utilized by Internet social network applications and services that need to replicate or search for mutable and dynamic data among distributed presence servers.

The contribution is that analyzes the scalability problems of distributed presence server architectures, and defines a new problem called the buddy-list search problem. Through our mathematical formulation, the scalability problem in the distributed server architectures of mobile presence services is analyzed. Finally, we analyze the performance complexity of Presence Cloud and different designs of distributed architectures, and evaluate them empirically to demonstrate the advantages of Presence Cloud. Server architectures of existing presence services, and introduce the buddy-list search problem in distributed presence Architectures in large-scale geographically data centres.

III. 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 server overlay: It 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 node degrees, where n is the number of presence servers. One-hop caching strategy: It 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 neighbours Directed buddy search: It is based on the directed search strategy. Presence Cloud ensures a one-hop search, it yields a small constant search latency on average.

The primary abstraction exported by our Presence Cloud issued scalable server architecture for Mobile presence services, and can be used to efficiently search the desired buddy lists. We illustrated a simple overview of Presence Cloud in Fig. 1. In the mobile Internet, a mobile user can access the Internet and make a data connection to Presence Cloud via 3G or Wi-Fi services. After the mobile user joins and authenticates himself/herself to the mobile presence service, the mobile user is determinately directed to one of Presence Servers in the Presence Cloud by using the Secure Hash Algorithm, such as SHA-1. The mobile user opens a

TCP connection to the Presence Server (PS node) for control message transmission, particularly for the presence information. After the control channel is
Established, the mobile user sends a request to the connected PS node for his/her buddy list searching. Our Presence Cloud shall do an efficient searching operation and return the presence information of the desired buddies to the mobile user.

IV. EVOLUTION

A cost analysis of the communication cost of Presence Cloud in terms of the number of messages required to search the buddy information of a mobile user. Note that how to reduce the number of inter server communication messages is the most important metric in mobile presence service issues. The buddy-list search problem can be solved by a brute-force search algorithm, which simply searches all the PS nodes in the mobile presence service. In a simple mesh-based design, the algorithm replicates all the presence information at each PS node; hence its search cost, denote by QMesh, is only one message. On the other hand, the system needs n _ 1 messages to replicate a user's presence information to all PS nodes, where n is the number of PS nodes. The communication cost of searching buddies and replicating presence information can be formulated as Mcost = QMesh +RMesh, where RMesh is the communication cost of replicating presence information to all PS nodes. Accordingly, we have Mcost = O(n). In the analysis of Presence Cloud, we assume that the mobile users are distributed equally among all the PS nodes, which is the worst case of the performance of Presence- Cloud.

Here, the search cost of Presence Cloud is denoted as Qp, which is messages for both searching buddy lists and replicating presence information. Because search message and replica message can be combined into one single message, the communication cost of replicating, Rp (0). It is straight forward to know that the communication cost of searching buddies and replicating presence information in Presence Cloud is Pcost .However, in Presence Cloud, a PS node not only searches a buddy list and replicates presence information, but also notifies users in the buddy list about the new presence event. Let b be the maximum number of buddies of a mobile user. Thus, the worst case is when none of the buddies are registered with the PS nodes reached by the search messages and each user on the buddy list is located on different PS nodes.

V. CONCLUSION

In this paper in large scale social network services mobile presence services is supported by the scalable server architecture called as presence cloud. A scalable server architecture that supports mobile presence services in large-scale social network services. Presence Cloud achieves low search latency and enhances the performance of 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. The results of that Presence Cloud achieve performance gains in the search cost without compromising search satisfaction.

REFERENCE

[1] 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, vol. 20, no. 6, pp. 16-21, July/Aug. 2006.

[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] Chi, R. Hao, D. Wang, and Z.-Z. Cao, "IMS Presence Server:aTraffic Analysis and Performance Modelling," Proc. IEEE Int'Conf. Network Protocols (ICNP), 2008.

[4] Instant Messaging and Presence Protocol IETF Working Group,http: //www.ietf.org /html.charters/impp-charter.html, 2012.

[5] Extensible Messaging and Presence Protocol IETF Working Group, http://www.ietf.org /html.charters/xmpp-chart er.html,2012.

[6] Open Mobile Alliance, "OMA Instant Messaging and Presence Service," 2005.

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

[8]X. Chen, S. Ren, H. Wang, and X. Zhang, "SCOPE: Scalable Consistency Maintenance in Structured P2P Systems," Proc. IEEE INFOCOM, 2005.

[9]. Quorumb asedtechni queshttp:// www.en.wikip idiaorg/ wiki/Quoru m_(distributed _computing).

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

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

[12]. Juuso aleksi lehtinen, "Mobile peer-to-peer over session initiation protocol", August 4,2008.

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

[14]. Michael Piatek, Tomas Isdal, Arvind Krishnamurthy , and Thomas Anderson "One hop Reputations for Peer to Peer File Sharing Workloads".

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