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

A Scalable Mobile Presence Services with Efficient Study in Social Network Applications

E.Ravi Kumar
M.Tech Student,
Dept of CSE,
Pace Institute of Technology and Sciences, Ongole.

Ch.Koteshwara Rao
Associate Professor,
Dept of CSE,
Pace Institute of Technology and Sciences, Ongole.

Jagadeeswara Rao.Annam
Associate Professor & HOD,
Dept of CSE,
Pace Institute of Technology and Sciences, Ongole.

ABSTRACT:

Social network applications are becoming increasingly popular on mobile devices. A mobile presence service is an essential component of a social network application because it maintains each 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. If presence updates occur frequently, the enormous number of messages distributed by presence servers may lead to a scalability problem in a large-scale mobile presence service. To address the problem, we propose an efficient and scalable server architecture, called 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 leverages 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. The results of simulations demonstrate that Presence Cloud achieves performance gains in the search cost without compromising search satisfaction.

Index Terms:

Social Networks, Mobile presence services, PGP, PresenceCloud, Distributed presence servers.

INTRODUCTION:

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. 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 searc 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. The scalability problem in the distributed server architectures of mobile presence services is analyzed through our mathematical formulation.

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

Finally, we demonstrate the advantages of Presence-Cloud by analyzing the performance complexity of PresenceCloud and different designs of distributed architectures, and evaluate them empirically. The primary abstraction exported by our PresenceCloud is used to construct scalable server architecture for mobile presence services, and can be used to efficiently search the desired buddy lists.

In the mobile Internet, a mobile user can access the Internet and make a data connection to PresenceCloud via 3G or Wi-Fi services. After the mobile user joins and authenticates him/her to the mobile presence service, the mobile user is determinately directed to one of Presence Servers in the PresenceCloud by using the Secure Hash Algorithm, such as SHA-1 [6]. The Pretty Good Privacy Program (PGP) uses a "Web of Trust" model for establishing trust relationships between users. This could allow a user to indirectly and unknowingly trust others that the user does not know.

MODULES:

1. Presence Cloud server overlay.

2. One-hop caching strategy.

3. Directed buddy search.

Modules Description:

1. Presence Cloud server overlay:

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.

2. One-hop caching strategy:

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.

3. Directed buddy search:

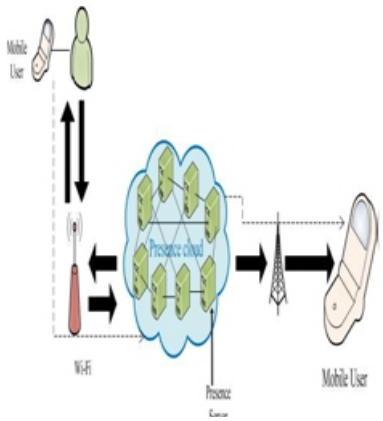
We contend that minimizing searching response time is important to mobile presence services. Thus, the buddy list searching algorithm of Presence Cloud coupled with the two-hop overlay and one-hop caching strategy ensures that Presence Cloud can typically provide swift responses for a large number of 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.

Proposed System:

Aim of proposed system is to design an architecture of disseminate server for coherence request to the system for buddy list search. In this project work a scalable server architecture 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 3 elements in presence loud which run across presence servers such as presence cloud server overlay, one hop [14] caching approach, and directed buddy search [15].

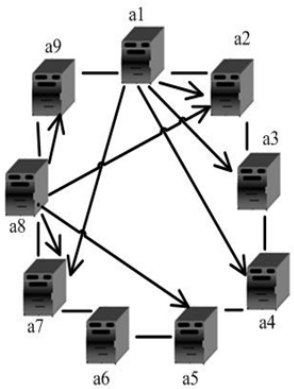
ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal



» 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.

Presence cloud server overlay:

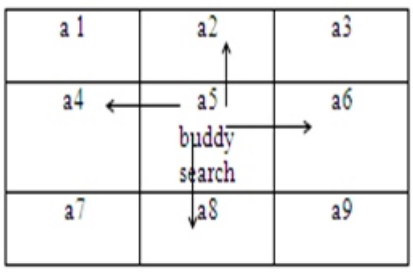


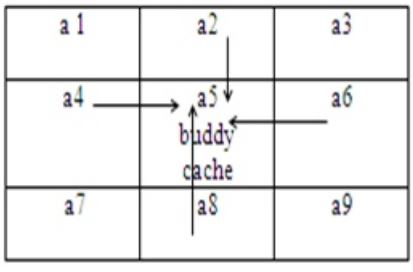
Presence server nodes are ordered in the form of server to server overlay in presence cloud server overlay and also endow low diameter overlay. Needs two hops to reach from one presence server node to other presence server node is the possession of low diameter and Presence cloud is based on grid quorum 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.

One hop caching:

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.

Directed buddy search:





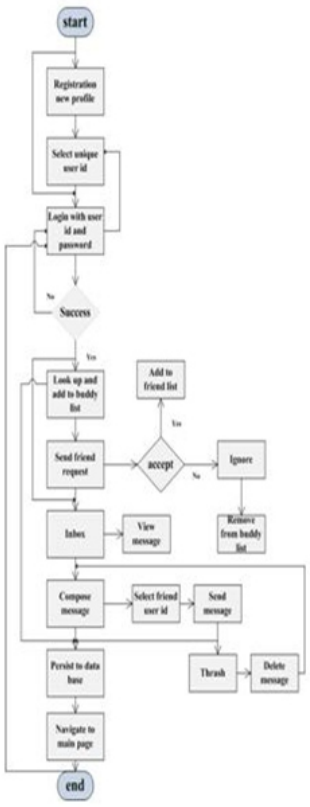
ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Implementation:

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.

Performance Analysis:

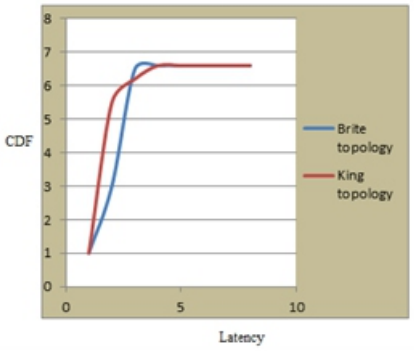
In king topology for king data set there is a real internet topology. Internet computation are derived using technique for king data set latency pattern. thebrite topology alpha is set to 0.15 and beta is set to 0.2 using Waxman model BRITE topology generator creates AS topology.



Process flowchart implementation

Performance metrics:

There are three metrics which computes the performance of server architecture. First, totalnumber of messages transferred between query creator and the presence server in presence cloud is the total search message. Second, average searching messages per arrived user is individual of user arrival prototype. Third, it searches for buddies when mobile user joins the network is average search latency.



RELATED WORK:

In this section, we describe previous researches on presence services, and survey the presence service of Existingsystems. Several IETF charters [7] have addressed closely related topics and many RFC documents on instant messaging and presence services (IMPS) have been published, e.g., XMPP [8], SIMPLE [9]. Jabber [10] 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 [11]. P2P-SIP [12] 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, P2PSIP clients are organized in a DHT system, rather than in a centralized server. Our research is essentially the paradigm in which we attack our problem. Just as in the electronic marketplace scenario, our agents do not have to rely on centralised mechanisms.

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

Trust information is propagated throughout the system via the interaction of the agents themselves, and the ,,quality of this information is calculated on the basis of the perceived trustworthiness of recommender. Trust is then revised upon receipt of new recommendations or new experiences. Other related work includes Pretty Good Privacy, which helped inspire the distributed nature of our model.

WEB OF TRUST:

The web-of-trust model [14] differs greatly from the hierarchical model. The hierarchical model is easily represented with computers as a tree, but the web-of-trust more closely relates to how people determine trust in their own lives. As people go through life and meet new people, they look to those they already trust to help make the determination if they should trust these new people. If people they trust already trust this new person, they are more likely to do the same. PGP implements this model by allowing people to sign each others keys assigning to each of the keys a level of trust and validity.

PGP assumes that a certificate binds a key to a person. That person chooses a name to identify them self, and also provide an email address as a global identifier. When a user signs another users key, the signing user is asserting that they have verified that the key truly belongs to the listed user and that it is valid. This method of signing keys leads to an ad-hoc network of trust. Because of the shape of the resulting graph, Phil Zimmerman called this system the "Web of Trust". The system allows users to specify how much trust to place in a signature by indicating how many independent signatures must be placed on a certificate for it to be considered valid.

HOW PGP WORKS:

PGP combines some of the best features of both conventional and public key cryptography. PGP is a hybrid cryptosystem. When a user encrypts plaintext with PGP, PGP first compresses the plaintext. Data compression saves modem transmission time and disk space and, more importantly, strengthens cryptographic security. Most cryptanalysis techniques exploit patterns found in the plaintext to crack the cipher.

Compression reduces these patterns in the plaintext, thereby greatly enhancing resistance to cryptanalysis (Files that are too short to compress or which don't compress well aren't compressed). PGP then creates a session key, which is a one-time-only secret key. This key is a random number generated from the random movements of your mouse and the keystrokes you type. This session key works with a very secure, fast conventional encryption algorithm to encrypt the plaintext; the result is cipher text. Once the data is encrypted, the session key is then encrypted to the recipient's public key.

This public key-encrypted session key is transmitted along with the cipher text to the recipient. Decryption works in the reverse. The recipient's copy of PGP uses his or her private key to recover the temporary session key, which PGP then uses to decrypt the conventionally-encrypted cipher text. The combination of the two encryption methods combines the convenience of public key encryption with the speed of conventional encryption. Conventional encryption is about 1, 000 times faster than public key encryption. Public key encryption in turn provides a solution to key distribution and data transmission issues. Used together, performance and key distribution are improved without any sacrifice in security.

You might think of a PGP certificate as a public key with one or more labels tied to it. On these 'labels' you'll find information identifying the owner of the key and a signature of the key's owner, which states that the key and the identification go together. (This particular signature is called a self-signature; every PGP certificate contains a self-signature). One unique aspect of the PGP certificate format is that a single certificate can contain multiple signatures. Several or many people may sign the key/ identification pair to attest to their own assurance that the public key definitely belongs to the specified owner. If you look on a public certificate server, you may notice that certain certificates, such as that of PGP's creator, Phil Zimmermann, contain many signatures. Some PGP certificates consist of a public key with several labels, each of which contains a different means of identifying the key's owner (for example, the owner's name and corporate email account, the owner's nickname and home email account, a photograph of the owner — all in one certificate).

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

Conclusion:

In large scale social network services mobile presence services is supported by the scalable server architecture called as presence cloud. Performance improved for mobile presence services and also low search latency is accomplished by presence cloud. Number of buddy search messages increased with user arrival rate. Presence cloud achieved performance gain in terms of search cost.

References:

1. K.Peternel, L.zebeeandA.Kos, "Using Presence information for an effective Collaboration".

2. ChetanS,GautamKumar,K.Dinesh,MathewK,andAbhimanyaM.A,"Cloud computing for Mobile world".

3. Facebook, http://www.facebook.com.

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

5.D.Dhiravidavasanthanayaki and G.Mangayarkarasi, "Presence server for mobile ubiquity services in presence cloud,"March 2014.

6. Instant Messaging and presence protocol IETF working group,http://www.ietf.org/html.charters/impp-charter.html,2012.

7. Chi-Jen Wu, Jan-Ming Ho, Member, IEEE, Ming-Syan Chen, Fellow, IEEE," A Scalable Server Architecture for Mobile Presence Services in Social Network Applications",2013.

8. Google, http://www.google.com.

9.Quorumbasedtechniqueshttp://www.en.wikipidiaorg/wiki/Quorum_(distributed_computing).

10. Jianmingzhao, Nianminyao, ShaobinCai, and Xiang li, "Tree-Mesh Based P2P Streaming data distribution schema",2012.

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

E.Ravi Kumar is pursuing his M.Tech (CSE), Pace Institute of Technology and Sciences,Ongole, Prakasam District, Andhra Pradesh.E-mail: rk.emana999@gmail.com

Ch.KoteshwaraRao ,M.Tech ,Associate Professor, Department of Computer Science & Engineering, Pace Institute of Technology and Sciences, Ongole, Prakasam District, Andhra Pradesh
E-mail: koteswararao143@gmail.com:

JagadeeswaraRao.Annam,M.Tech(P.hd),Associate Professor & HOD, Department of Computer Science & Engineering, Pace Institute of Technology and Sciences,Ongole, Prakasam District, Andhra Pradesh.