

## **Enhanced Scalability In Social Network Application Using Presence Service**

*Poonam G. Pachkawade , Prof. V. M. Deshmukh*

Master of Engineering 2<sup>nd</sup> yr.  
Information Technology  
Prof. Ram Meghe Institute of Technology & Research,  
Badnera, India  
[poonampachkawade@gmail.com](mailto:poonampachkawade@gmail.com)  
Head  
Information Technology Department  
Prof. Ram Meghe Institute of Technology & Research,  
Badnera, India  
[msvdeshmukh@rediffmail.com](mailto:msvdeshmukh@rediffmail.com)

### Abstract

Social network applications have become more and more in style on mobile devices. A mobile presence service is an important element of a social network application as a result of it maintains every mobile user's presence data, like the present standing (online/offline), GPS location and network address, and additionally updates the user's on-line friends with the data regularly. If presence updates occur often, the large range of messages distributed by presence servers could cause a measurability downside in an exceedingly large-scale mobile presence service. to handle the matter, we have a tendency to propose associate degree economical and ascendable server design, referred to as Presence Cloud, that allows mobile presence services to support large-scale social network applications. once 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 design for economical presence looking. It additionally leverages a directed search algorithmic program and a one-hop caching strategy to attain little constant search latency. we have a tendency to analyze the performance of Presence Cloud in terms of the search value and search satisfaction level. The search value is outlined because the total range of messages generated by the presence server once a user arrives; and search satisfaction level is outlined because the time it takes to go looking for the coming user's friend list. The results of simulations demonstrate that Presence Cloud achieves performance gains within the search value while not compromising search satisfaction.

Keywords: Social networks, mobile presence services, distributed presence servers, cloud computing.

### INTRODUCTION

In Mobile presence service may be a important element of a social network applications owing to mobile user's

presence details like world positioning system location, network address, and on-line/offline standing are ceaselessly apprise to user's online buddies. A mobile presence services is a vital component of cloud computing environments, for the rationale it keeps associate degree up -to- date

list of presence info of mobile user. If presence updates occur usually the quantity of messages distributed by presence server might cause quantifiability downside and pal list search downside in large-scale mobile presence services. to beat the quantifiability downside projected associate degree economical and scalable server design referred to as presence cloud. It organizes the presence server into gathering based mostly server-server design for economical looking. once a mobile user joins a network or web, presence cloud searches the presence info. It conjointly achieves tiny constant search latency by the directed search rule and one-hop caching strategy. The Presence entitled applications like Face-book, Twitter etc., that is created by mobile devices and cloud computing nature owing to the prevalence of web .Way the members are engaged with their buddies on web are modified by the social network services . so as to move with buddies across nice distance participants will dispense the live event at once victimization their mobile device. Mobile user's presence info details are going to be maintained by mobile presence service . In cloud computing surroundings mobile presence service may be a important element of social network application. due to the presence of the net, mobile devices and cloud computing environments will offer presence-enabled applications i.e. social network applications/services, worldwide. Face book, Twitter Foursquare Google Latitude, pal cloud and Mobile Instant electronic messaging (MIM) are samples of presence-enabled applications that have big speedily within the last decade. Social network services are dynamical the ways in which during which participants interact with their friends on the net. They exploit the knowledge regarding the standing of participants as well as their appearances and activities to move with their friends. Moreover, due to the wide convenience of mobile devices (e.g., Smartphone's) that utilize wireless mobile network technologies, social network services modify participants to share live experiences instantly across nice distances. Presence info tells the detail regarding mobile user's convenience, activity and machine capability. Service will binding of user id to his/her current presence info details. every individual mobile user encompasses a pal list which incorporates details of whom he/she needs to move with in social network services. once a user will cargo from one level to alternative, this transformation is instinctively transmitted to every individual on the pal list. Server cluster technology will increase the search speed and reduce the report

time. for instance in social network application mobile user logs in through his/her mobile device, the mobile presence services searches and divulges every of them regarding user's friend list like instant electronic messaging system .

## 2. LITERATURE SERVEY

Instant electronic communication (IM) may be a variety of on-line chat that offers period of time text transmission over the web. The network IM system: AOL Instant courier, Yahoo! courier (YMSG),and Microsoft Messenger(MSN).are mentioned. Most IM systems use centralized clusters to supply presence services. Jennings et al. [5] bestowed a Taxonomy of various options and functions supported by these 3 IM systems. The authors additionally provided an summary of the system architectures and determined that the systems use client-server-based architectures. All 3 industrial systems use server clusters for quantifiability. AIM and MSN take the uneven approach. AIM defines many styles of servers: login, BOS icon, user search, chat space setup, and chat space hosting. MSN defines 3 types: dispatch, notification, and central. In distinction, YMSG takes the parallel approach. purchasers want solely contact one variety of server so route all types of activities to that individual server. whereas every has been designed and enforced on an individual basis, the cluster exhibits similar characteristics with relevance network and system design. for instance, all of the IM protocols enable authenticating with a central server, partaking in camera messages, and conversing publically chat rooms. additionally, some IM systems enable file transfers, digital camera usage, victimization privacy controls, maintaining pal lists, voice chat sessions, and alternative choices. Most IM systems, together with the 3 use client-server design. IM suppliers usually host a group of servers that customers log in to and exchange messages with. In client-server design, since each management and knowledge ways bear the central servers, scaling the service to scores of users is tough. The quantifiability issue is especially tough for voice chat sessions. AIM uses client-server design for traditional operations however uses a peer-to-peer approach for voice-chat sessions. YMSG additionally uses client-server design for traditional operations similarly as voice-chat service. YMSG voice traffic is routed through a centralized voice-chat server.MSN additionally uses a client-server design for traditional operations and peer-to-peer for voice – chat communication. Most IM systems have mechanisms for maintaining lists

of friends. These are usually referred to as “buddy lists,” “allow lists,” and “block lists.” Recently, presence services are integrated into mobile services. For instance, 3GPP has outlined the mixing of presence service into its specification in UMTS. It’s supported SIP [8] protocol, and uses straightforward [8] to manage presence info. Recently, some mobile devices additionally support mobile presence services. For instance, the moment electronic communication and Presence Services was developed by the Wireless Village association and was united into Open Mobile Alliance (OMA) IMPS [10] in 2005. In [11], Chen et al. planned a edible consistent theme to scale back the quantity of change messages in mobile presence services of information processing transmission scheme (IMS). However, it additionally suffers quantifiability downside since it uses a central SIP server to perform presence update of mobile users. In [13], authors bestowed the server quantifiability and distributed management problems in IMS-based presence service. Recently, the IETF has commenced a shot to standardize IM and chat protocols. 2 competitive standards are being developed: one supported straightforward [8] and a second done supported XMPP [13]. Straight forward is associate extension to the Session Initiation Protocol (SIP) [8] that adds instant electronic communication and presence. SIP may be a text – primarily based control-plane protocol for establishing transmission sessions like box information processing. The Message Session Relay Protocol (MSRP) is a second message transport protocol outlined by the easy social unit. It’s a session-based protocol Skype, a preferred box information processing application, utilizes the worldwide Index (GI) technology [8] to supply a presence service for users. GI may be a multitier spec wherever every node maintains full data of all on the market users. of these IM services use central server design that results in quantifiability downside at server facet. thus to deal with the matter, economical and ascendible server design, referred to as Presence Cloud is planned by Chi-Jen et la, [1]. Presence Cloud organizes presence servers into a quorum-based server –to-server design for economical presence looking out. It additionally uses directed search rule and a one-hop caching strategy to realize little constant search latency. Overall, Presence Cloud is shown to be a ascendible mobile presence service in large-scale social network services.

### 3. PROPOSED WORK

Aim of projected system is to style an design of broadcast server for coherence request to the system for friend list search. during this project work a scalable server design that provides services to ‘n’ variety of users is given. And presenting a certain style by rising the thought of peer to see system whereas coming up with presence cloud. There are three parts in presence cloud that encounter presence servers like presence cloud server overlay, one hop caching approach, and directed friend search .

- Presence servers that are gift in presence cloud, wherever these presence servers are organized in gathering based mostly server to server design and conjointly load on servers are balance in presence cloud sever overlay.
- of these presence server keeps caches for buddies so as to extend question speed is one hop caching approach.
- Finding tiny constant search delay leads to directed friend search by decreasing network traffic victimization one hop search strategy.

Architecture of presence cloud that is that the projected work is shown in Figure1, victimization 3G or Wi-Fi services mobile user access the net and create a knowledge link to the presence cloud. victimization secure hash rule mobile users square measure intent to 1 of the presence servers. To transfer presence data details, the mobile user is documented to the mobile presence services and conjointly opens a communications protocol link. Once path is about up, the mobile user request for the friend list to the presence server that is gift in presence cloud. and eventually the request is responded by the presence cloud when finishing associate degree economical search of buddy’s presence data.

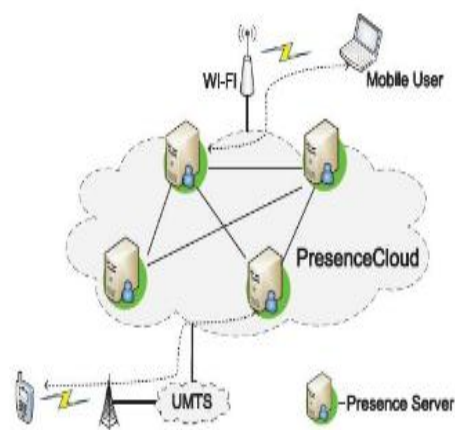


Fig 1. Architecture for presence cloud

### 3.1 Presence cloud server overlay

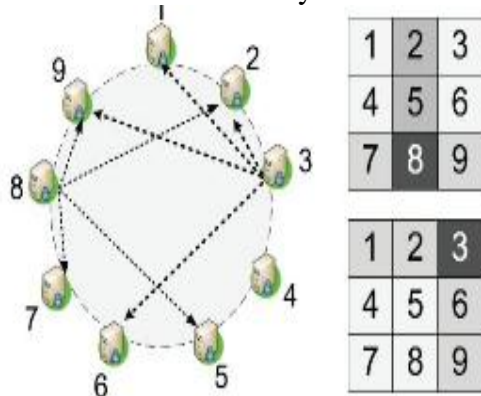


Fig 2. Presence cloud server overlay

Presence server nodes square measure ordered within the variety of server to server overlay in presence cloud server overlay and conjointly endow low diameter overlay. desires 2 hops to achieve from one presence server node to alternative presence server node is that the possession of low diameter and Presence cloud is predicated on grid assemblage system. Size of presence server node is  $O \sqrt{m}$ , wherever  $m$  is that the variety of presence server in mobile presence services. By victimization grid assemblage system presence server list is constructed and this presence server list maintains presence server node that features a set of presence server nodes. as an example in Figure2, grid assemblage is about to  $\sqrt{9} * \sqrt{9}$ . Presence server node a8 features a presence server list and presence server node a1 has presence server list . therefore presence server node a8 and a1 will engineered their overlay network consistent with their presence server.

### 3.2. One hop caching

To duplicate the presence data details presence cloud needs caching strategy so as to reinforce the potency of search operation. In presence cloud for the hooked up users, presence data details of user list area unit maintained by presence server node. Duplicating user list by presence server nodes area unit at the most one hop removed from itself. once association is well-ried by neighbor's cache is updated and additionally updated sporadically with their neighbors. If question accepted by presence server node it's not solely respond with matches from cache wherever user list on the market by its neighbors. Presence data changes for mobile users once user leaves presence cloud or thanks to failure. Response from presence server node broad casts its new presence to different neighboring presence server node for updates. Presence data remains

constant and up thus far throughout the session time of user is ensured by one hop caching strategy.

### 3.3. Directed pal search

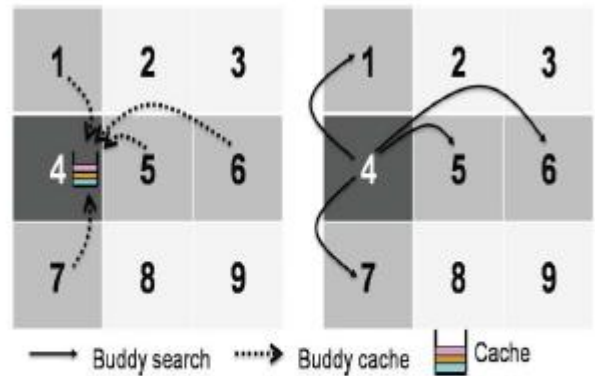


Fig 3. Directed buddy search

Figure three shows, for mobile presence services it's necessary to cut back search time. victimisation 2 hop overlay and one hop caching strategy presence cloud endow response for big variety of mobile users. One hop search used for queries so as to cut back network traffic one hop caching maintains user list of its neighbors to reinforce latent period by increasing to find buddies.

### ALGORITHM1. PRESENCE CLOUD MAINTENANCE algorithmic program

- 1: /\* sporadically verify postscript node n's plist \*/
- 2: Definition:
- 3: plist: set of the present postscript list of this postscript node, n
- 4: plist[].connection: the present postscript node in plist
- 5: plist[].id: symbol of the right association in plist
- 6: node.id: symbol of postscript node node
- 7: Algorithm:
- 8:  $r \leftarrow \text{Sizeof}(\text{plist})$
- 9: for  $i = \text{one to } r$  do
- 10:  $\text{node} \leftarrow \text{plist}[i].\text{connection}$
- 11: if  $\text{node.id} \neq \text{plist}[i].\text{id}$  then

```

12: /* raise node to refresh n's postscript list entries
*/
13: findnode=Find_CorrectPSNode(node)
14: if findnode=nil then
15: pslist[i].connection=RandomNode(node)
16: else
17: pslist[i].connection=findnode
18: end if
19: else
20: /* send a heartbeat message */
21: bfailed=SendHeartbeatmsg(node)
22: if bfailed= true then
23:pslist[i].connection=RandomNode(node)
24: end if
25: end if
26: finish for

```

Our algorithmic program is fault tolerance style. At every postscript node, an easy Stabilization () method sporadically contacts existing postscript nodes to keep up the postscript list. The Stabilization () method is in an elaborate way conferred within the algorithmic program . once a postscript node joins, it obtains its postscript list by contacting a root. However, if a postscript node n detects unsuccessful postscript nodes in its postscript list, it must establish new connections with existing postscript nodes. In our algorithmic program, n ought to decide a random postscript node that\ within the same column or row because the unsuccessful postscript node.

Algorithm 2: Directed buddy search algorithm:

1. A mobile user logins Presence Cloud and decides the associated notation node, q.
2. The user sends a chum List Search Message, B to the notation node alphabetic character.
3. once the notation node alphabetic character receives a B, then retrieves every atomic number 83 from B and searches its user list and one-hop cache to retort to the approaching question. And removes the responded buddies from B.

4. If B = nil, the chum list search operation is finished.

5. Otherwise, if B =nil, the notation node alphabetic character ought to hash every remaining symbol in B to get a grid ID, severally.

6. Then, the notation node alphabetic character aggregates these b(g) to become a brand new B(j), for every g Sj. Here, notation node j is that the intersection node of Sq intersection Sg. And sends the new B(j) to notation node j.

#### 4. CONCLUSION

In massive scale social network services mobile presence services is supported by the ascendable server design known as presence cloud. Performance improved for mobile presence services and additionally low search latency is accomplished by presence cloud. range of brother search messages magnified with user arrival rate. Presence cloud achieved performance gain in terms of search value. We have got bestowed Presence Cloud, a ascendable server design that supports mobile presence services in large-scale social network services. we\have got shown that Presence Cloud achieves low search latency and enhances the performance of mobile presence services. additionally, we have a tendency to mentioned the quantifiability downside in server design styles, and introduced the buddy-list search downside, that could be a quantifiability downside within the distributed server design of mobile presence services. Through an easy mathematical model, we have a tendency to show that the entire range of brother search messages will increase well with the user arrival rate and therefore the range of presence servers. The results of simulations demonstrate that Presence Cloud achieves major performance gains in terms of the search value and search satisfaction. Overall, Presence Cloud is shown to be a ascendable mobile presence service in large-scale social network services.

#### 5. REFERENCES:

- [1] Chi-Jen Wu, Jan-Ming Ho, and Ming-Syan Chen, "A Scalable Server Architecture for Mobile Presence Services in Social Network Applications", 2013.
- [2] 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.

- [3] Z. Xiao, L. Guo, and J. Tracey, "Understanding Instant Messaging Traffic Characteristics," 2007.
- [4] C. Chi, R. Hao, D. Wang, and Z.-Z. Cao, "IMS Presence Server: Traffic Analysis and Performance Modeling," 2008.
- [5] Instant Messaging and Presence Protocol IETF Working Group, <http://www.ietf.org/html.charters/imp-protocol.html>, 2014.
- [6] Extensible Messaging and Presence Protocol IETF Working Group, <http://www.ietf.org/html.charters/xmpp-charter.html>, 2012..
- [7] <http://www.jabber.org>, 2012..
- [8] A. Hourri, S. Parameswar, E. Aoki, V. Singh, and H. Schulzrinne, "Scaling Requirements for Presence in SIP/SIMPLE," 2009.
- [9] S.A. Baset, G. Gupta, and H. Schulzrinne, "Open VoIP: An Open Peer-to-Peer VoIP and IM System," 2008.
- [10] Open Mobile Alliance, "OMA Instant Messaging and Presence Service," 2005
- [11] W.-E. Chen, Y.-B. Lin, and R.-H. Liou, "A Weakly Consistent Scheme for IMS Presence Service," July 2009.
- [12] N. Banerjee, A. Acharya, and S.K. Das, "Seamless SIP-Based Mobility for Multimedia Applications,"
- [13] Kundan Singh and Henning Schulzrinne "SIPPEER : A session initiation protocol (SIP)-based peer-to-peer internet telephony client adaptor"
- .
- [14] Michael Piatek, Tomas Isdal, Arvind Krishna murthy , and Thomas Anderson "One hop Reputations for Peer to Peer File Sharing Workloads".
- [15] Brent Hecht, Jaime Teevan , Meredith Ringel Morris, and Dan Liebling, "Search Buddies: Bringing Search Engines into the Conversation", 2012