International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

"A Survey on Presence Cloud based solution for on demand data in wireless computing devices"

[1]monali.D.Akhare & [2]prof.N.M.Kandoi
[1]Master of Engineering,Computer Engineering Department.,Shri Sant Gajanan Maharaj College of Engineering,Shegaon
Sant Gadge Baba Amravati University,India
[2]Head, CSE.Shri Sant Gajanan Maharaj College of Engineering,Shegaon Sant Gadge Baba Amravati University,India
Email: [1]akhare.monali@gmail.com, [2]kandoin2011@gmail.com

Abstract

The main objective of this paper is to propose an on demand QoS routing algorithm. Since the requirements for various applications may vary time to time, the approach for QoS routing may not be proactive. The proposed approach has two phases namely route discovery phase and route maintenance phase. When a source node has to pass data to a destination node with QoS requirements it starts with the route discovery phase. Once the route is found, the data transfer will take place. While data transmission is going on, it is also required to maintain the path to the destination. This is very much desirable and required in mobile ad hoc networks and hence is done in the route maintenance phase.

Introduction

Many people are sharing digital resources through networks in order to facilitate, enhance or improve collaborative work. Information sharing is not only important to support collaborative work but it also represents the basis for design and implementation of solutions for typical design aspects of groupware applications, such as: floor control, group memory, shared objects replication and sessions and users management. Advances in mobile technology have extended the sharing information scenarios to Mobile Networks which has brought new challenges. We entitled applications such as Face-book, Twitter etc., which is produced by mobile devices and cloud computing nature due to the prevalence of internet .Way the members are engaged with their buddies on internet are changed by the social network services . 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. In cloud computing environment mobile presence service is a vital component of social network application.

The main goal in the design of the protocol was to reduce the overhead for routing. MAN is a collection of mobile nodes which communicate over radio. These kinds of networks are very flexible, thus they do not require any existing infrastructure or central administration. Therefore, mobile ad-hoc networks are suitable for temporary communication links. In this paper, we propose a 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. 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.

Literature Survey

Instant messaging (IM) is a type of online chat which offers real-time text transmission over the Internet. Most popular network IM system: AOL Instant Messenger, Yahoo! Messenger (YMSG), and Microsoft Messenger (MSN).Jennings et al. [5]presented Taxonomy of different features and functions supported by these three IM systems. The authors also provided an overview of the system architectures and observed that the systems use client-server-based architectures. All three commercial systems use server clusters for scalability. AIM and MSN take the asymmetric approach. AIM defines several types of servers: login, BOS icon, user search, chat room setup, and chat room hosting. MSN defines three types: dispatch, notification, and switchboard. In contrast, YMSG takes the symmetric approach. Clients need only contact one type of server and then route all kinds of activities though that particular server. While each has been designed and implemented separately, the overall group exhibits similar characteristics with respect to network and system architecture.

AIM uses client-server architecture for normal operations but uses a peer-to-peer approach for voice-chat sessions. YMSG also uses client-server architecture for normal operations as well as voice-chat service. YMSG voice traffic is routed through a centralized voice-chat server.MSN also uses a client-server architecture for normal operations and peer-to-peer for voice – chat communication. Most IM systems have mechanisms for maintaining lists of friends. These are typically called “buddy lists,” “allow lists,” and“ block lists.”Recently, presence services are also integrated into mobile services. For example, 3GPPhas defined the integration of presence service into its specification in UMTS. It is based on SIP [8]protocol, and uses SIMPLE [8] to manage presence information. Recently, some mobile devices also support mobile presence services. For example, the Instant Messaging and Presence Services was developed by the Wireless Village consortium and was united into Open Mobile Alliance (OMA)IMPS [10] in 2005.In [11], Chen et al. proposed a weakly consistent scheme to reduce the number of updating messages in mobile presence services of IP Multimedia Subsystem (IMS). In [13], authors presented the server scalability and distributed management issues in IMS-based presence service. Recently, the IETF has embarked on an effort to standardize IM and chat protocols. Two competing standards are being developed: one based on SIMPLE [8] and a second one based on XMPP [13].SIMPLE is an extension to the Session Initiation Protocol (SIP) [8] that adds instant messaging and presence. The Message Session Relay Protocol (MSRP) is an instant message transport protocol defined by the SIMPLE working group. It is a session-based protocol Skype, a popular voice over IP application, utilizes the Global Index (GI) technology [8] to provide a presence service for users.GI is a multitier network architecture where each node maintains full knowledge of all available users. All these IM services use central server architecture which leads to scalability problem at server side. So to address the problem, efficient and scalable server architecture, called Presence Cloud is proposed by Chi-Jen ET la, [1].

Presence Cloud organizes presence servers into a quorum-based server-to-server

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

architecture for efficient presence searching. It also uses directed search algorithm and a one-hop caching strategy to achieve small constant search latency. Overall, Presence Cloud is shown to be a scalable mobile presence service in large-scale social network services.

Aim & Objectives:

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 system while designing presence cloud. There are 3 elements in presence cloud which run across presence servers such as presence cloud server overlay, one hop caching approach, and directed buddy search.

1. 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.

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

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

Methodology

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. 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.

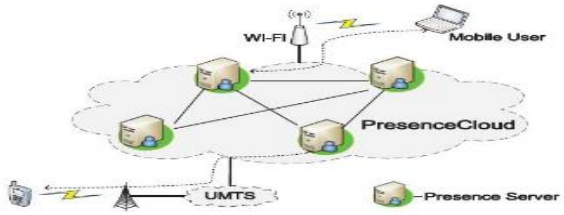

Fig 1. Architecture for presence cloud

2.1. 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.

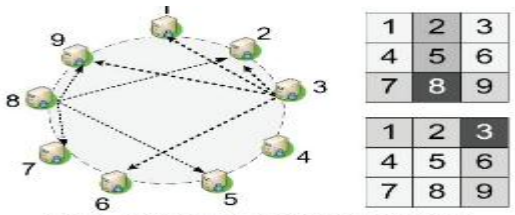

Fig 2. Presence cloud server overlay

2.2. 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

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

maintained by presence server node. Duplicating user list by presence server nodes are at most one hop away from itself. 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.

2.3. Directed buddy search

Figure 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.

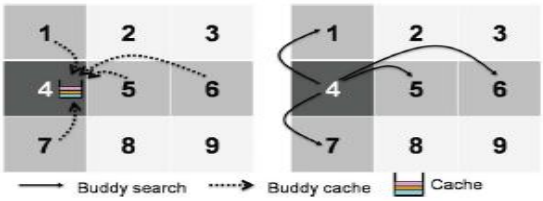

Fig 3. Directed buddy search

PRESENCE CLOUD MAINTENANCE ALGORITHM

1: /* periodically verify PS node n's pslist */

2: Definition:

3: pslist: set of the current PS list of this PS node, n

4: pslist[].connection: the current PS node in pslist

5: pslist[].id: identifier of the correct connection in pslist

6: node.id: identifier of PS node node

7: Algorithm:

8: r □ Sizeof(pslist)

9: for i = 1 to r do

10: node □ pslist[i].connection

11: if node.id ≠pslist[i].id then

12: /* ask node to refresh n's PS 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: end for

Our algorithm is fault tolerance design. At each PS node, a simple Stabilization () process periodically contacts existing PS nodes to maintain the PS list. The Stabilization () process is elaborately presented in the Algorithm. When a PS node joins, it obtains its PS list by contacting a root. However, if a PS node n detects failed PS nodes in its PS list, it needs to establish new connections with existing PS nodes. In our algorithm, n should pick a random PS node that is in the same column or row as the failed PS node.

Directed buddy search algorithm:

1. A mobile user logins Presence Cloud and decides the associated PS node, q.

2. The user sends a Buddy List Search Message, B to the PS node q.

3. When the PS node q receives a B, then retrieves eachbi from B and searches its user list and one-hopcache to respond to the coming query. And removes the responded buddies from B.

4. If B = nil, the buddy list search operation is done.

5. Otherwise, if B =nil, the PS node q should hash each remaining identifier in B to obtain a grid ID, respectively.

6.Then, the PS node q aggregates these b(g) to become a new B(j), for each g Sj. Here, PS node j is the intersection node of Sq intersection Sg. And sends the new B(j) to PS node j.

REFERENCES:

[1] Chi-Jen Wu, Jan-Ming Ho, Member, IEEE, and Ming-Syan Chen, Fellow, IEEE on "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," Proc. IEEE 27th Int'l Conf. Distributed Computing Systems (ICDCS), 2007.

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

[5]InstantMessaging and Presence Protocol IETF WorkingGroup,http://www.ietf.org/html.charters /impp-charter.html, 2014.

[6] Extensible Messaging and Presence Protocol IETFWorkingGroup,http://www.ietf.org/html.charters/xmpp-charter.html,2012..

[7] http://www.jabber.org, 2012.

[8] A. Houri, S. Parameswar, E. Aoki, V. Singh, and H. Schulzrinne, "Scaling Requirements for Presence in SIP/SIMPLE," IETF Internet draft, 2009.

[9] S.A. Baset, G. Gupta, and H. Schulzrinne, "OpenVoIP: An Open Peer-to-Peer VoIP and IM System," Proc. ACM SIGCOMM, 2008.

[10] Open Mobile Alliance, "OMA Instant Messaging and PresenceService," 2005

[11] W.-E. Chen, Y.-B.Lin, and R.-H. Liou, "A Weakly ConsistentScheme for IMS Presence Service," IEEE Trans. Wireless Comm.,vol. 8, no. 7, pp. 3815-3821, July 2009.

[12] N. Banerjee, A. Acharya, and S.K. Das, "Seamless SIP-BasedMobility for Multimedia Applications," IEEE Network, vol. 20,

[13]. 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 cllient adaptor" .

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

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