RESEARCH ARTICLE OPEN ACCESS

Study of Server Scalability Issues in Mobile Presence Services

Kishori Dharurkar[1], Dipak.Patil[2]
Department of Information Technology,
Amrutvahini college of Engineering, Sangamner, (M.S)
Dist. Ahmednagar (Maharashtra-India)

ABSTRACT
With the development of wireless telecommunication technologies, many customer services that are used in the computer-based Internet have also appeared in mobile phones. Mobile instant message (MIM) is a typical example. .MIM enables consumers, whether sitting at the computer or on the road, to connect instant message (IM) with existing communities and social services. A mobile presence service is an essential component of a MIM and social network applications. 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. This presence update occur frequently so enormous messages are distributed by presence server may lead to a scalability problem in a large-scale mobile presence service. This paper gives a survey of presence services of existing and proposed systems and their server scalability issues. Various Instant Messaging services (IM), VoIP services are discussed here.

.***Keywords:-*** Social networks, mobile presence services, Instant messaging.

I. INTRODUCTION

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. Presence is a powerful network capability that is useful for consumers as a rich communication tool, for enterprises to enhance productivity and for mobile operators to increase the value of the network. Presence complements new business models in open mobile eco-systems. Application developers for Android, iPhone or Windows Mobile can easily derive and use Presence to offer new social applications.

Facebook, twitter, Mobile Instant Messaging (MIM), WhatsApp etc. are the popular social , presence-enabled applications. These applications engage with their friends on the Internet, share live experiences instantly across great distances. For example, Facebook receives more than 25 billion shared items every month and Twitter receives more than 55 million tweets each day. In the future, mobile devices will become more powerful, sensing, and media capture devices.

The key function of a mobile presence service is to maintain an up-to-date list of presence information of all mobile users. Presence is a foundational element of unified communications. The presence information includes details about a mobile user's location, availability, activity, mood, device capability, and preferences. The service must also bind the user's ID to his/her current presence information, as well as retrieve and subscribe to changes in the presence information of the user's friends.

In social network services, each mobile user has a friend list, typically called a buddy list, which contains the contact information of other users that he/she wants to communicate with. The mobile user's status is broadcast automatically to each person on the buddy list whenever he/she transits from one status to the other. For example, when a mobile user logs into a social network application, such as an IM system, through his/her mobile device, the mobile presence service searches for and notifies everyone on the user's buddy list. To maximize a mobile presence service's search speed and minimize the notification time, most presence services use server cluster technology [5]. Currently, more than one billion people use social network services on the Internet [1].

Given the growth of social network applications and mobile network capacity, it is expected that the number of mobile presence service users will increase substantially in the near future. Thus, a scalable mobile presence service is deemed essential for future Internet applications. In the last decade, so many Internet services have been deployed in distributed paradigms as well as cloud computing applications. For example, the services developed by Google and Facebook are spread among as many distributed servers as possible to support the huge number of users worldwide. In this paper, we discuss the server architectures of existing and proposed presence services.

II. COMPARATIVE STUDY

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). are discussed.

Most IM systems use centralized clusters to provide presence services. Jennings et al. [5] presented a 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. For example, all of the IM protocols allow authenticating with a central server, engaging in private messages, and conversing in public chat rooms. In addition, some IM systems allow file transfers, Webcam usage, using privacy controls, maintaining buddy lists, voice chat sessions, and other options.

Most IM systems, including the three use client-server architecture. IM providers typically host a set of servers that customers log in to and exchange messages with.

In client-server architecture, since both control and data paths go through the central servers, scaling the service to millions of users is difficult. The scalability issue is particularly difficult for voice chat sessions.

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, 3GPP has 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). However, it also suffers scalability problem since it uses a central SIP server to perform presence update of mobile users.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. SIP is a text-based control-plane protocol for establishing multimedia sessions such as Voice over IP.

The Message Session Relay Protocol (MSRP) is an instant message transport protocol defined by the SIMPLE working group. It is a session-based protocol.

XMPP, the Extensible Messaging and Presence Protocol, is an alternative to SIMPLE. The basic syntax and semantics of XMPP were developed originally within the Jabber open-source community [13]. it is intended mainly for the purpose of building IM and presence applications. This IM protocol is also the protocol used in the commercial implementation of Google Talk and Facebook Chat. In October 2004, the XMPP working group at IETF published the documents RFC 3920, RFC 3921, RFC 3922 and RFC 3923, to standardize the core XMPP protocol. WhatsApp also uses a customized version of the XMPP. Upon installation, it creates a user account using one's phone number as the username (Jabber ID: `[phone number] @s.whatsapp.net`).

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 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

III. CONCLUSIONS

This paper provides survey of different presence enabled services with their system architectures. Popular IM services, VoIP, mobile presence services and chat communication using IETF standardized protocols such as SIMPLE and XMPP are discussed. Presence Cloud is a proposed scalable server is also discussed.

Out of all the systems, Presence Cloud seems to more scalable and efficient server for mobile presence enabled services.

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'l
Conf. Network Protocols (ICNP), 2008.
[5] Instant Messaging and Presence Protocol IETF Working Group,
http://www.ietf.org/html.charters/impp-charter.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. 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 Consistent
Scheme 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-Based
Mobility for Multimedia Applications," IEEE Network, vol. 20,