PRESENCE CLOUD SUPPORTING MOBILE PRESENCE SERVICES IN LARGE-SCALE SOCIAL NETWORK SERVICES

International Journal of Information Technology and Management

Vol. VI, Issue No. I, February-2014, ISSN 2249-4510

AN INTERNATIONALLY INDEXED PEER REVIEWED & REFEREED JOURNAL

International Journal of Information Technology and Management
Vol. VI, Issue No. I, February-2014, ISSN 2249-4510

Presence Cloud Supporting Mobile Presence Services in Large-Scale Social Network Services

Harshika Juvvadi[1], Reshma Gande[2], Srinivas Gadam[3], Hitesh Allam[4], Panduranga Thota[5]

[1]Student, Department of Computer Science and Engineering, Varadha Reddy College of Engineering, Warangal, India
[2]Student, Department of Computer Science and Engineering, Varadha Reddy College of Engineering, Warangal, India
[3]Student, Department of Computer Science and Engineering, Varadha Reddy College of Engineering, Warangal, India
[4]Student, Department of Computer Science and Engineering, Varadha Reddy College of Engineering, Warangal, India
[5]Senior Assistant Professor, Department of Computer Science and Engineering, Varadha Reddy College of Engineering, Warangal, India

Abstract – Social networking services on the Internet are growing and increasing numbers of people are using these new ways to communicate and share information. Many users are communicating with both friends from outside the service as well as with people they have only been in contact with through a social networking service. At the same time mobile phones are becoming more powerful and increasingly offer high speed Internet connectivity. Because of this people expect these social networking services to be available on their mobile device, as well as on their personal computer. Given the capabilities of today's mobile devices, it is possible to extend the existing phonebook with capabilities to support a variety of social networking services in addition to the existing communication options. By integrating the contacts gained from the social networking service into the mobile phonebook the user can reach these contacts easily. 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 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.

Key Terms : Cloud, Global Index, Service IM System, Overlay Network

---------------------------♦-----------------------------

I. INTRODUCTION

Well known commercial IM systems leverage some form of centralized clusters to provide presence services. Jennings III *et al.* presented a taxonomy of different features and functions supported by the three most popular IM systems, AIM, Microsoft MSN and Yahoo! Messenger. The authors also provided an overview of the system architectures and observed that the systems use client-server-based architectures. Skype, a popular voice over IP application, utilizes the Global Index (GI) technology to provide a presence service for users. GI is a multi-tiered network architecture where each node maintains full knowledge of all available users. Since Skype is not an open protocol, it is difficult to determine how GI technology is used exactly. Moreover, Xiao *et al.* analyzed the traffic of MSN and A Recently, there is an increase amount of interest in how to design a peer-to-peer SIP. P2PSIP has been proposed to remove the centralized server, reduce maintenance costs, and prevent failures in server-based SIP deployment. To maintain presence information, P2PSIP clients are organized in a DHT system, rather than in a centralized server. However, the presence service architectures of Jabber and P2PSIP are distributed, the *buddy-list search problem* we defined later also could affect such distributed systems. It is noted that few articles in discuss the scalability issues of the distributed presence server architecture. Saint Andre analyzes the traffic generated as a result of presence information between users of inter-domains that support the XMPP. Houri *et al.* Show that the amount of presence traffic in SIMPLE can be extremely heavy, and they analyze the effect of a large presence system on the memory and CPU loading. Those works in study related problems and developing an initial set of guidelines for optimizing inter-domain presence traffic and present a DHT-based presence server architecture. 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 protocol, and uses SIMPLE to manage presence information. Recently, some mobile devices also support mobile presence services. For example, the Instant Messaging and Presence Services (IMPS) was developed by the Wireless Village consortium and was united into Open Mobile Alliance (OMA) IMPS in 2005. In, 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, authors presented the server scalability and distributed management issues in IMS-based presence service IM system. They found that the presence information is one of most messaging traffic in instant messaging systems. In, authors shown that the largest message traffic in existing presence services is buddy NOTIFY messages.

II. RELATED WORK

The Internet's excellent scalability and robustness result in part from the end-to-end nature of Internet congestion control. End-to-end congestion control algorithms alone, however, are unable to prevent the congestion collapse and unfairness created by applications that are unresponsive to network congestion. To address these maladies, we propose and investigate a novel congestion-avoidance mechanism called Congestion Free Router (CFR). CFR entails the exchange of feedback between routers at the borders of a network in order to detect and restrict unresponsive traffic flows before they enter the network, thereby preventing congestion within the network. The fundamental philosophy behind the Internet is expressed by the scalability argument: no protocol, mechanism, or service should be introduced into the Internet if it does not scale well. A key corollary to the scalability argument is the end-to-end argument: to maintain scalability, algorithmic complexity should be pushed to the edges of the network whenever possible. Perhaps the best example of the Internet philosophy is TCP congestion control, which is implemented primarily through algorithms operating at end systems. Unfortunately, TCP congestion control also illustrates some of the shortcomings of the end-to-end argument. 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. Cloud computing has raised a variety of vital privacy and security problems [19], [25], [30]. Such problems area unit because of the very fact that, within the cloud, users' information and applications reside—at least for a precise quantity of time—on the cloud cluster that is closely-held and maintained by a 3rd party considerations arise since within the cloud it's not continually clear to people why their personal info is requested or how it'll be used or passed on to different parties. To date, very little work has been wiped out this house, specifically with relevancy answerability. Pearson et al. have projected answerability mechanisms to handle privacy considerations of finish users [30] so develop a privacy manager [31]. Their basic plan is that the user's non-public information area unit sent to the cloud in Associate in Nursing encrypted kind, and therefore the process is completed on the encrypted information. The output of the process is deobfuscated by the privacy manager to reveal the proper result. However, the privacy manager provides solely restricted options in this it doesn't guarantee protection once the information area unit being disclosed. In [7], the authors gift a stratified design for addressing the end-to-end trust management and answerability downside in federate systems. The authors' focus is incredibly totally different from ours, in this they principally leverage trust relationships for answerability, at the side of authentication and anomaly detection. Further, their resolution needs third-party services to finish the observation and focuses on lower level observation of system resources. With relevancy Java-based techniques for security, our strategies area unit associated with self-defending objects (SDO) [17]. Self-defending objects area unit Associate in Nursing extension of the object-oriented programming paradigm, wherever software package objects that provide sensitive functions or hold sensitive information area unit accountable for protective those functions/data. Similarly, we tend to additionally extend the ideas of object-oriented programming. The key difference in our implementations is that the authors still deem centralized information to take care of the access records, whereas the things being protected area unit control as separate files. In previous work, we tend to provided a Java-based approach to stop privacy outpouring from categorization [39], that may well be integrated with the United States intelligence agency framework projected during this work since they build upon connected architectures. In terms of authentication techniques, Appel and Felten [13] projected the Proof-Carrying authentication (PCA) framework. The PCA includes a high order logic language that permits quantification over predicates, and focuses on access management for net services whereas associated with ours to the extent that it helps maintaining safe, superior, mobile code, the PCA's goal is extremely totally different from our analysis, because it focuses on substantiating code, instead of observation content. Another work is by Mont et al. United Nations agency projected Associate in Nursing approach for powerfully coupling content with access management, exploitation Identity-Based coding (IBE) [26]. We additionally leverage IBE techniques, however during a} very totally different manner. we tend to don't deem IBE to bind the content with the foundations. Instead, we tend to use it to supply robust guarantees for the encrypted content and therefore the log files, like protection against chosen plaintext and ciphertext attacks. Additionally, our work could look the same as works on secure information beginning [5], [6], [15], however really greatly differs from them in terms of goals, techniques, and application domains. Works on information beginning aim to ensure information integrity by securing the information beginning.

III. EXPERIMENTAL RESULTS

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.

Fig-1 : Presence Cloud server overlay

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.

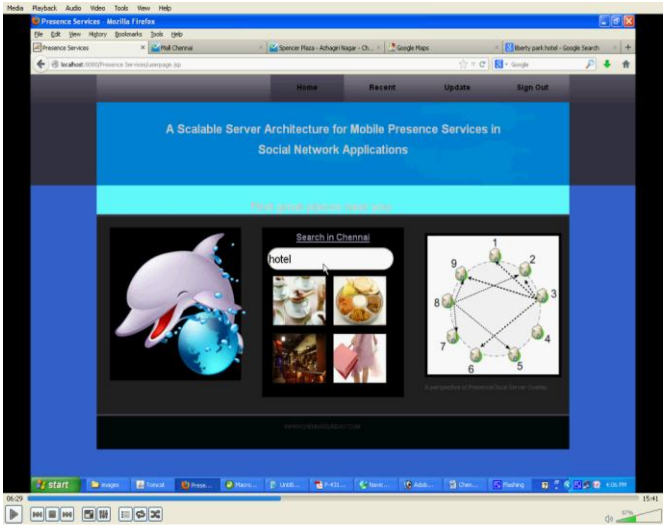

Fig-2.1 : Directed Buddy Search

Fig-2.2 : Directed Buddy Search

Fig-2.3 : Directed Buddy Search

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.

IV. CONCLUSION

In this paper, we have presented Presence Cloud, a scalable server architecture that supports mobile presence services in large-scale social network services. We have shown that Presence Cloud achieves low search latency and enhances the performance of mobile presence services. In addition, we discussed the scalability problem in server architecture designs, and introduced the buddy-list search problem, which is a scalability problem in the distributed server architecture of mobile presence services. Through a simple mathematical model, we show that the total number of buddy search messages increases substantially with the user arrival rate and the number of presence servers. The results of simulations demonstrate that Presence Cloud achieves major performance gains in terms of the search cost and search satisfaction. Overall, Presence Cloud is shown to be a scalable mobile presence service in large-scale social network services.

V. REFERENCES

[1] Facebook, http://www.facebook.com.

[2] Twitter, http://twitter.com.

[3] Foursquare http://www.foursquare.com.

[4] Google latitude, http://www.google.com/intl/enus/latitude/intro.html.

[5] Buddycloud, http://buddycloud.com.

[6] Mobile instant messaging, http://en.wikipedia.org/wiki/Mobile instant messaging.

[7] 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*, 2006.

[8] Gobalindex, http://www.skype.com/intl/en-us/support/user-guides/p2pexplained/.

[9] Z. Xiao, L. Guo, and J. Tracey, ”Understanding instant messaging traffic characteristics,” *Proc. of IEEE ICDCS*, 2007.

[10] C. Chi, R. Hao, D. Wang, and Z.-Z. Cao, ”Ims presence server: Traffic analysis and performance modelling,” *Proc. of IEEE ICNP*, 2008.

[11] Instant messaging and presence protocol ietf working group http://www.ietf.org/html.charters/impp-charter.html.

[12] Extensible messaging and presence protocol ietf working group http://www.ietf.org/html.charters/xmpp-charter.html.

[13] Sip for instant messaging and presence leveraging extensions ietf working group. http://www.ietf.org/html.charters/simplecharter.html.

[14] P. Saint-Andre., ”Extensible messaging and presence protocol (xmpp): Instant messaging and presence describes instant messaging (im), the most common application of xmpp,” *RFC 3921*, 2004.

[15] B. Campbell, J. Rosenberg, H. Schulzrinne, C. Huitema, and D. Gurle, ”Session initiation protocol (sip) extension for instant messaging,” *RFC 3428*, 2002.

[16] Jabber, http://www.jabber.org/.

[17] Peer-to-peer session initiation protocol ietf working group, http://www.ietf.org/html.charters/p2psip-charter.html.

[18] K. Singh and H. Schulzrinne, ”Peer-to-peer internet telephony using sip,” *Proc. of ACM NOSSDVA*, 2005.

[19] P. Saint-Andre, ”Interdomain presence scaling analysis for the extensible messaging and presence protocol (xmpp),” *RFC Internet Draft*, 2008.

[20] A. Houri, T. Rang, and E. Aoki, ”Problem statement for sip/simple,” *RFC Internet-Draft*, 2009.

[21] A. Houri, S. Parameswar, E. Aoki, V. Singh, and H. Schulzrinne, ”Scaling requirements for presence in sip/simple,” *RFC Internet- Draft*, 2009.

[22] S. A. Baset, G. Gupta, and H. Schulzrinne, ”Openvoip: An open peer-to-peer voip and im system,” *Proc. of ACM SIGCOMM*, 2008.

[23] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, ”Sip: Session initiation protocol,” *RFC 3261*, 2002.

[24] Open Mobile Alliance, OMA instant messaging and presence service, 2005.

[25] W.-E. Chen, Y.-B. Lin, and R.-H. Liou, ”A weakly consistent scheme for ims presence service,” *IEEE Transactions on Wireless Communications*, 2009.

[26] N. Banerjee, A. Acharya, and S. K. Das, ”Seamless sip-based mobility for multimedia applications.” *IEEE Network*, vol. 20, no. 2, pp. 6–13, 2006.

[27] P. Bellavista, A. Corradi, and L. Foschini, ”Ims-based presence service with enhanced scalability and guaranteed qos for interdomain enterprise mobility,” *IEEE Wireless Communications*, 2009.

[28] A. Houri, E. Aoki, S. Parameswar, T. Rang, , V. Singh, and H. Schulzrinne, ”Presence interdomain scaling analysis for sip/simple,” *RFC Internet-Draft*, 2009.

[29] M. Maekawa,”A*p n* algorithm for mutual exclusion in decentralized systems,” *ACM Transactions on Computer Systems*, 1985.

[30] D. Eastlake and P. Jones, ”Us secure hash algorithm 1 (SHA1),” *RFC 3174*, 2001.

[31] M. Steiner, T. En-Najjary, and E. W. Biersack, ”Long term study of peer behavior in the kad DHT,” *IEEE/ACM Trans. Netw.*, 2009.

[32] K. Singh and H. Schulzrinne, ”Failover and load sharing in siptelephony,” *International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, July 2005.

[33] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, ”Chord: A scalable peer-to-peer lookup service for internet,” *IEEE/ACM Tran. on Networking*, 2003.

[34] X. Chen, S. Ren, H. Wang, and X. Zhang, ”Scope: scalable consistency maintenance in structured p2p systems,” *Proc. of IEEE INFOCOM*, 2005.

[35] K. P. Gummadi, S. Saroiu, and S. D. Gribble., ”King: Estimating latency between arbitrary internet end hosts,” *Proc. of ACM IMW*, 2002.

[36] A. Medina, A. Lakhina, I. Matta, and J. Byers, ”Brite: An approach to universal topology generation,” *Proc of ACM MASCOTS*, 2001.

[37] R. Cox, A. Muthitacharoen, and R. T. Morris, ”Serving DNS using a peer-to-peer lookup service,” *Proc. of IPTPS*, 2002.

[38] V. Ramasubramanian and E. G. Sirer, ”Beehive: 0(1) lookup performance for power-law query distributions in peer-to-peer overlays,” *Proc. of USENIX NSDI*, 2004.

[39] A. Abdul-Rahman and S. Hailes., ”A distributed trust model,” *Proc. of the workshop on New security paradigms*, 1997.

[40] K.-T. Chen, C.-Y. Huang, P. Huang, and C.-L. Lei, ”Quantifying skype user satisfaction,” *Proceedings of ACM SIGCOMM*, 2006.

[41] P. Anick, ”Using terminological feedback for web search refinement: a log-based study,” *Proceedings of ACM SIGIR conference on Research and development in informaion retrieval*, pp. 88–95, 2003.