International Journal of scientific research and management (IJSRM)
||Volume||2||Issue||2||Pages||559-564||2014||
Website: www.ijsrm.in ISSN (e): 2321-3418

Autonomic Notification For A Context Aware System Using Push Server Technology

Vasudevan.R, Chandramohan.K, Manimegalai.S,

Computer science and Engineering, Gnanamani college of Engineering, Tamilnadu, India.

vaasudevan.r@gmail.com , chandramohancse@gmail.com

Electronics and Communication Engineering, Gnanamani college of Technology, Tamilnadu, India.

mani.megu1990@gmail.com

Abstract: *The method of autonomic notification for a context aware system using push server technology. Push server technology has developed under the Comet programming model. The Comet investigates basic problems associated with distributed computing in an internet environment with mobile objects .Comet, support mobility in two flavors: physical mobility and virtual mobility. Push technology has evolved to a great extent since its inception and there have been many additions of new features to the available solutions, in terms of reliability, performance and new standards. We propose an application to push data in real time which enables bidirectional flow of data and which is independent of any particular publisher/subscriber. Usually, the file systems are heavily guarded by firewalls and accessing them from a remote location poses problems, our application solves this problem and allows secure access while maintaining the integrity. Server application is called Push Notify which is based on publish/subscribe model and is independent, which enables it to be easily integrated with any file server of any university and with any communication client, to achieve the paradigm, "any publisher, any subscriber". Communication client for subscribers to receive notifications or alerts can be web browser extension or mobile device application.*

Keywords: push Notification, comet programming..,

1. INTRODUCTION

The classical style of the web called REST (Representational State Transfer) requires all communication between the browser and the server to be initiated by the client, i.e., the end user clicks on a button or link and thereby requests a new page from the server. In this scheme, each interaction between the client and the server is independent of the other interactions [2]. No 'permanent' connection is established between the client and the server maintains no state information about the clients. This scheme helps scalability, but precludes servers from sending asynchronous notifications. There are, however, many use cases where it is important to update the client-side interface as soon as possible in response to server-side changes. An auction web site where the user's needs to be averted that another bidder has made a higher bid, a stock ticker, a news portal. Today, such web applications requiring real-time event notifications are usually implemented using the Comet and push-based style, where the clients subscribe to their topic of interest, and the server publishes the changes to the clients asynchronously every time its state changes [1]-[2]. Comet enables the server to send a message to the client when the event occurs, without the client having to explicitly request [2].

2. MOBILE COMPUTING

Mobile computing is human–computer interaction by which a computer is expected to be transported during normal usage. Mobile computing involves mobile communication, mobile hardware, and mobile software [3].

Communication issues include ad hoc and infrastructure networks as well as communication properties, protocols, data formats and concrete technologies. Mobile deals with the characteristics and requirements of mobile applications. Display, collect, and transfer information from a mobile device to an information system using one or a combination of various data transfer methods. In the figure data communication technology

component is wired connection with information system component [3]-[2]. Mobile computing device component is wireless connection with data communication technology component [3].

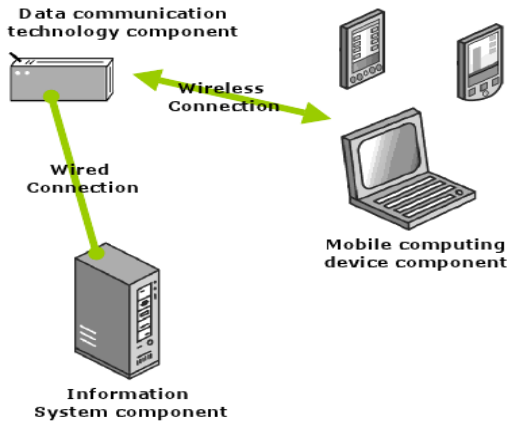

Fig.1 Mobile Communications

2.1 DEVICES OF MOBILE COMPUTING

Many types of mobile computers have been introduced since the 1990s including the,

- Personal digital assistant/enterprise digital assistant
- Smartphone
- Tablet computer [4]

2.2 SECURITY ISSUES INVOLVED IN MOBILE COMPUTING

Mobile security or mobile phone security has become increasingly important in mobile computing. It is of particular concern as it relates to the security of personal information now stored on the Smartphone [4].

More and more users and businesses use smart phones as communication tools but also as a means of planning and organizing their work and private life. Within companies, these technologies are causing profound changes in the organization of information systems and therefore they have become the source of new risks [4]-[5]. Indeed, smart phones collect and compile an increasing amount of sensitive information [3].

All smart phones, as computers, are preferred targets of attacks. These attacks exploit weaknesses related to smart phones that can come from means of communication like SMS, MMS, Wi-Fi networks, and GSM. There are also attacks that exploit software vulnerabilities from both the web browser and operating system. Finally, there are forms of malicious software that rely on the weak knowledge of average users [4].

2.3 MOBILE DATA COMMUNICATION

Wireless data connections used in mobile computing take three general forms so. Cellular data service uses technologies such as GSM, CDMA or GPRS, and more recently 3G networks such as W-CDMA, EDGE or CDMA2000. These networks are usually available within range of commercial cell towers [2]. Wi-Fi connections offer higher performance, may be either on a private business network or accessed through public hotspots, and have a typical range of 100 feet indoors and up to 1000 feet outdoors. Satellite Internet access covers areas where cellular and Wi-Fi are not available and may be set up anywhere the user has a line of sight to the satellite's location, which for satellites in geostationary orbit means having an unobstructed view of the southern sky. Some enterprise deployments combine networks from multiple cellular networks or use a mix of cellular, Wi-Fi and satellite [5]. When using a mix of networks, a mobile virtual private network not only handles the security concerns, but also performs the multiple network logins automatically and keeps the application connections alive to prevent crashes or data loss during network transitions or coverage loss [6].

2.4 MOBILE APPLICATION DEVELOPMENT

Developing application software for mobile devices requires considering the constrains of these devices. Mobile devices run on battery and have less powerful processors than personal computers. Developers also have to consider a lengthy array of screen sizes, hardware specifications and configurations because of intense competition in mobile software and changes within each of the platforms [6].

Mobile application development requires use of specialized integrated development environments made. Mobile applications are first tested within the development environment using emulators and later subjected to field testing. Emulators provide an inexpensive way to test applications on mobile phones to which developers may not have physical access [7].

3. MOBILE APPLICATION MANAGEMENT (MAM)

Mobile Application Management (MAM) describes software and services responsible for provisioning and controlling access to internally develop and commercially available mobile apps used in business settings on both company-provided and "bring your own" smart phones and computers. Mobile application management differs from mobile device management [6]-[7].

As the name suggests MAM focuses on application management it provides a lower degree of control over the device, but a higher level of control over applications. MDM solution manages the down to device firmware and configuration settings and can include management of all applications and application data [5]

3.1 BASIC ALGORITHM

Algorithm presents the entire scheme of the proposed method, which comprises a load-leveling mechanism and a load-balancing mechanism implemented on each server, and a server selection mechanism implemented on each device. The load-balancing mechanism determines the server load using a load index computed from the CPU utilization and memory utilization of each server [5]. Then the mechanism notifies devices of the result. The server selection organization determines which server should be connected using load information from the load-balancing mechanism [3].The load-leveling mechanism determines a target load index of each server by exchanging load indices among multiple servers. It then adjusts the load of each server [7]. The basic algorithm of the proposed method is described along with (a)–(i) as follows [7]-[8]

(a) Device connects to a server. When a device connects to a server for the first time, the server is selected arbitrarily using a random method, DNS round-robin fashion, etc.

(b) Server lets the response from the device wait until a request occurs.

(c) Server returns a request message to the device as a response to (a) when a request to the device takes place in the server.

(d) Device executes processing according to the content of the request message and then to returns the result to the server.

(e) Server passes a response to the application and simultaneously notifies the device of processing completion.

(f) Device connects to the same server to query for the next request.

(g) When a timeout of long polling takes place, the server notifies the device of the timeout.

(h) Server determines its own load condition based on its load index when notifying the device of the timeout, and adds the timeout notice to be sent to the device.

(i) When the server is not in a high load, the process returns to (a) and the device is then connect to the same server.

4. PUSH NOTIFY: PUSH SERVER APPLICATION

Push technology has evolved to a great extent since its inception and there have been many additions of new features to the available solutions, in terms of reliability, performance and new standards. In this paper, we propose an application to push data in real time which enables bidirectional flow of data and which is independent of any particular publisher/subscriber [7].

Server application is called Push Notify which is based on publish/subscribe model and is independent, which enables it to be easily integrated with any file server of any university and with any communication client, to achieve the paradigm, "any publisher, any subscriber". [8] Communication client for subscribers to receive notifications or alerts can be web browser extension or mobile device application.

4.1PERSISTENCY SUPPORT FOR MOBILE OBJECTS IN THE COMET HETEROGENEOUS ENVIRONMENT

Open distributed computing in an internetwork environment has gained considerable attention. The paper presents COMET, A common object management

environment that investigates basic problems associated with distributed computing in an internet environment with mobile objects. COMET persistency support is based on typed memory, manipulated by persistent actions [5]-[6]. The use of typed memory enables transparent handling of the presentation management issues that are unavoidable in a heterogeneous environment. Persistent actions on the other hand ensure that the typed memory is manipulated in a consistent manner. The typed memory approach presents a conceptually simple persistency model to the object programmer [8].

4.2 MAIN BLOCK DIAGRAM

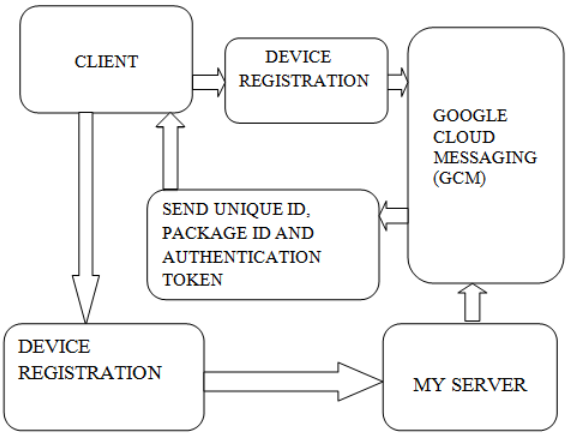

Fig.2 Main block Diagram

4.2 MOBILE PUSH NOTIFICATION

Push notifications on mobile and other platforms allow your cloud-based applications to send brief alerts and updates to a client application [6].Push notifications are a great way to keep the dialogue going between a consumer and a marketer. However, mobile experts agree that similar to all other marketing channels, the key is consistency – and including the medium into a multichannel strategy. Push notifications as a stand-alone mobile messaging channel has a place in the marketer's toolbox [7].

4.3 ANDROID PUSH NOTIFICATION

Google Cloud Messaging (GCM) replaces the beta version of Android Cloud to Device Messaging (C2DM). Android Cloud to Device Messaging (C2DM), which is now deprecated, is a push notification service that helps developers to send data from servers to their applications on Android devices (that were launched together with Android 2.2 by Google) [4]. The service provides a simple, lightweight mechanism that servers can use to tell mobile applications to contact the server directly, to fetch the updated application or user data [7].

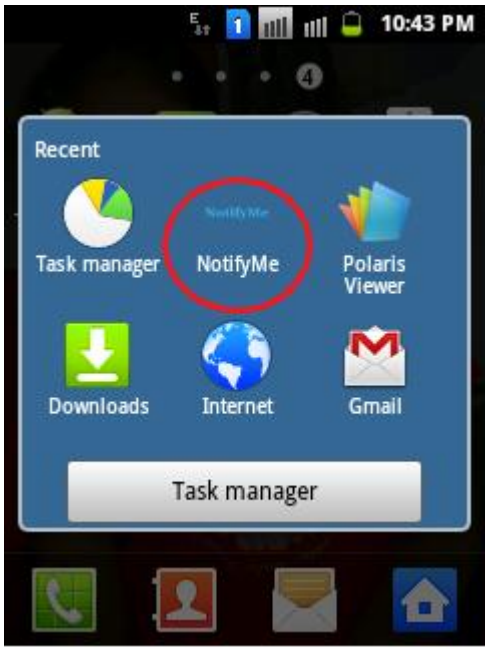

Fig.3 Android push notification

4.4 WINDOWS PHONE PUSH NOTIFICATIONS

The Microsoft Push Notification Service in Windows Phone offers third-party developers a resilient, dedicated and persistent channel to send data to a Windows Phone app from a cloud service in a power-efficient way [8].

- Your app requests a push notification URI from the Push client service
- The Push client service negotiates with the Microsoft Push Notification Service (MPNS) - and MPNS returns a notification URI to the Push client service
- The Push client service returns the notification URI to your app
- Your app can then send the notification URI to your cloud service
- When your cloud service has info to send to your app, it uses the notification URI to send a push notification to MPNS

Push notification and the payload attached to it, the info is delivered as raw data to the app and the app's tile is visually updated, or a toast notification is displayed [5]. MPNS sends a response code to your cloud service after a

push notification is sent, indicating that the notification has been received and will be delivered to the device at the next possible opportunity [6]. However, MPNS doesn't provide an end-to-end confirmation that your push notification was delivered from your cloud service to the phone [7].

4.5 COMETS AND PUSH TECHNOLOGY

Comet is a web application model in which a long-held HTTP request allows a web server to push data to a browser, without the browser explicitly requesting it. Comet is an umbrella term, encompassing multiple techniques for achieving this interaction. All these methods rely on features included by default in browsers, such as JavaScript, rather than on non-default plugging. The Comet approach differs from the original model of the web, in which a browser requests a complete web page at a time [6]-[8].

4.6 SYSTEM ARCHITECTURE

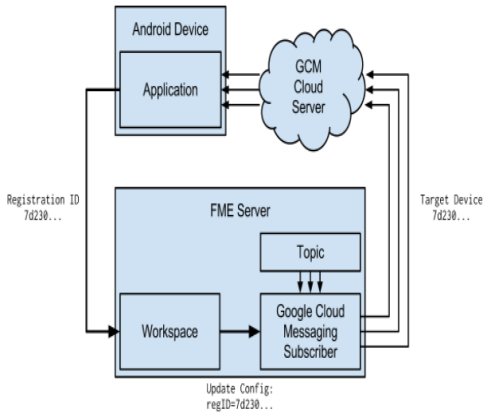

Fig.4 System Architecture

5. CONCLUSION

Push technology are quite advanced and powerful methods for achieving web-based real time event notification. We have analyzed and compared representative scenarios and proposed an initial architecture for a mobile push system. The P/S interaction scheme is the basis for our architecture. It is a well-established solution for the asynchronous interaction between frequently unavailable devices. Proposed push approach shows high data coherence and high network performance However, push brings some scalability issues; the server application CPU usage is 7 times higher as in pull. For larger number of users, load balancing and server clustering techniques are unavoidable. With the pull approach, achieving total data coherence with high network performance is very difficult. If the pull interval is higher than the publish interval, some data miss will Pull performs well only if the pull interval equals to publish interval. However, in order to achieve that, we need to know the exact publish interval beforehand. This work examines the possible solutions to this problem, provides solutions and analyzes the feasibility of those approaches. Based on the analysis of existing solutions, this work proposes desirable solution.

REFERENCES

1. T. Liao. Global Information Broadcast: "Architecture for Internet Push Channels," IEEE Trans. Internet Computing, 4(4):16–25, July/August 2000.
2. Blazevic, s. Giordano, and j.-Y. Leboudec, "A location based routing method for mobile ad hoc networks," IEEE Trans.Mobile Computing, vol. 4, no. 2, pp. 97-110, mar. 2005.
3. Z. Liu and T. D. Bui, "Dynamic mobile terminal location registration in wireless networks," IEEE Trans. Mobile Computing., vol. 4, no. 6,pp. 630–640, Nov. 2005.
4. J.Bacon, et.al. "Generic Support for Distribute Applications,"IEEETrans.Computer, 33(3):68–76, March 2000.
5. b.Karp and h.t.Kung,"gpsr:greedy perimeter stateless routing for wireless network,"IEEE.Trans.mobile computing, pp. 243-254, aug. 2000.
6. j. Hightower and g. Borriello, "location systems for ubiquitous computing,"IEEE Trans .Mobile computing, vol. 34, no. 8, pp. 57-66, aug. 2001.
7. G. Wu,S. Talwar, K.Johnsson, N.Himayat, and K.D. Johnson: "M2M: From Mobile To Embedded Internet," Communications Magazine, IEEE Vol. 49, Issue 4, pp.3 (2011-4).
8. P. Sutton, R. Arkins, and B. Segall. Supporting Disconnectedness – Transparent Information Delivery for Mobile and Invisible Computing. Proc.IEEEInt. Symposium on ClusterComputing and the Grid, IEEE CS Press, Los Alamitos, Calif., pp.87-90,may 2001.

Author Profile

R. Vasuedevan received the B.Sc. and MCA. degrees in Computer Science and Applications from MASC & GCT in 2009 and 2012, respectively. During 2012-2014, he stayed in Master Degree (Computer Science and Engineering) at Gnanamani College of Engineering from Namakkal.

Mr.K.Chandramohan received the MCA and ME Degree in Computer Science and Applications from KSR College, Erode and Mahendra Engineering College, Namakkal. During the 2009 – till date , he stayed in Head of the Department in Computer science and Engineering.

S. Manimegalai received the B.Tech degrees in Information Technology from PSNA College of Engineering and Technology in 2012. During 2012-2014, he stayed in Master Degree (Embedded Systems) at Gnanamani College of Technology from Namakkal.