[Volume 1 issue 3 April 2013]
Page No.88-93 ISSN :2320-7167
[INTERNATIONAL JOURNAL OF MATHEMATICS AND COMPUTER RESEARCH]

Implementation Of High Performance SEDA(Staged Event Driven Architecture) Server As MS Active Sync Proxy Server.

Anshuj Sharma[1] Ashish Kumar[2] Darshika Manral[3] Priya Dwivedi[4] Prof. Gajanan M. Walunjkar[5]

[1,2 ,3,4]BE (Information Technology),Army Institute Of Technology, Pune , Maharashtra, India,

anshuj0412@gmail.com

[5]Assistant Professor, IT Deptt. ,Army Institute Of Technology, Pune , Maharashtra, India,

gm_walunjkar@rediffmail.com

Corresponding Author Details: Anshuj Sharma #312A,Old Boys Hostel, Army Institute Of Technolgy, Dighi Hills,Alandi Road, Pune-411015, Maharashtra,India

ABSTRACT:

We propose to develop a web server based on Staged Event Driven Architecture which will then be further used as a MS Active Sync proxy server for synchronization of data and information between the Exchange ActiveSync-enabled devices.

In order to synchronize your mobile device with the email server, your mobile device needs to be connected to the Mobile device company's exchange server. But such solutions are specific to mobile device company, for example a person with a Nokia mobile cannot connect to Exchange server without going through company's firewall or Nokia's own Exchange server.

In this paper we would like to discuss our idea for providing ActiveSync Proxy server for helping the mobile users to synchronizing their data: emails, calendar or contacts without having to worry about the company's firewall or mobile device's company. But we aim to provide efficient communication over the network, so to achieve this we will be implementing a high performance server using SEDA which would then be used as a VPN server to cater our new idea of proxy server.

Keywords: [Stages; Events; Sync; Threads; Server; Network]

TITLE: Implementation of high performance SEDA(Staged Event Driven Architecture) server as MS Active Sync proxy server.

INTRODUCTION

1.1 NEED

Internet is providing services to millions of users each day. We all expect it to be robust and highly responsive. The number of concurrent sessions and hits per day actually lead to very high I/O and network requests, placing load on resources. As the demand for Internet services grows, new system design techniques must be used to manage this load. This systems challenge is magnified by three trends that increase the generality of services. First, services themselves are becoming more complex, with static content replaced by dynamic content that involves extensive computation and I/O. Second, service logic tends to change rapidly, which increases the complexity of engineering and deployment. Third, services are increasingly hosted on general-purpose facilities, rather than on platforms that are carefully engineered for a particular service. As these trends continue, we envision that a rich array of novel services will be authored and pushed into

the infrastructure where they may become successful enough to scale to millions of users. Several investigations are addressing the high-level aspects of service authorship, including naming, lookup, composition, and versioning.
We focus here on the performance aspect of the problem: achieving robust performance on a wide range of services subject to huge variations in load, while preserving ease of authorship.
So, we need an architecture that is capable of handling these challenges. As the Traditional web server focuses on thread per request model. Due to this OS have to switch between tasks and cause latency. Synchronization also required in the traditional model. Traditional web server show serious performance degradation at high loads. To overcome the present demand and drawbacks of traditional web server we will implement SEDA based web server.
These positives of SEDA server will help in making the MS Active Sync Proxy server more efficient,robust and high performing.

1.2APPLICATION

After developing the SEDA web server we will use it as MS ActiveSync proxy server for synchronization of data and information between the Exchange ActiveSync-enabled devices.
ActiveSync is a mobile data synchronization technology and protocol developed by Microsoft. ActiveSync ensures that you don't ever have to enter any information twice. If you enter it on your mobile device, it is synchronized to the mail2web Mobile Email Server which in this case will be a SEDA Web Server.
With ActiveSync you can synchronize Emails, Contacts, Tasks and Calendar items including Appointments, Events and Meeting Requests. Synchronization using ActiveSync can happen while your device is cradled or cabled to your computer, or wirelessly over the air using Wi-Fi, GPRS or other wireless technologies.

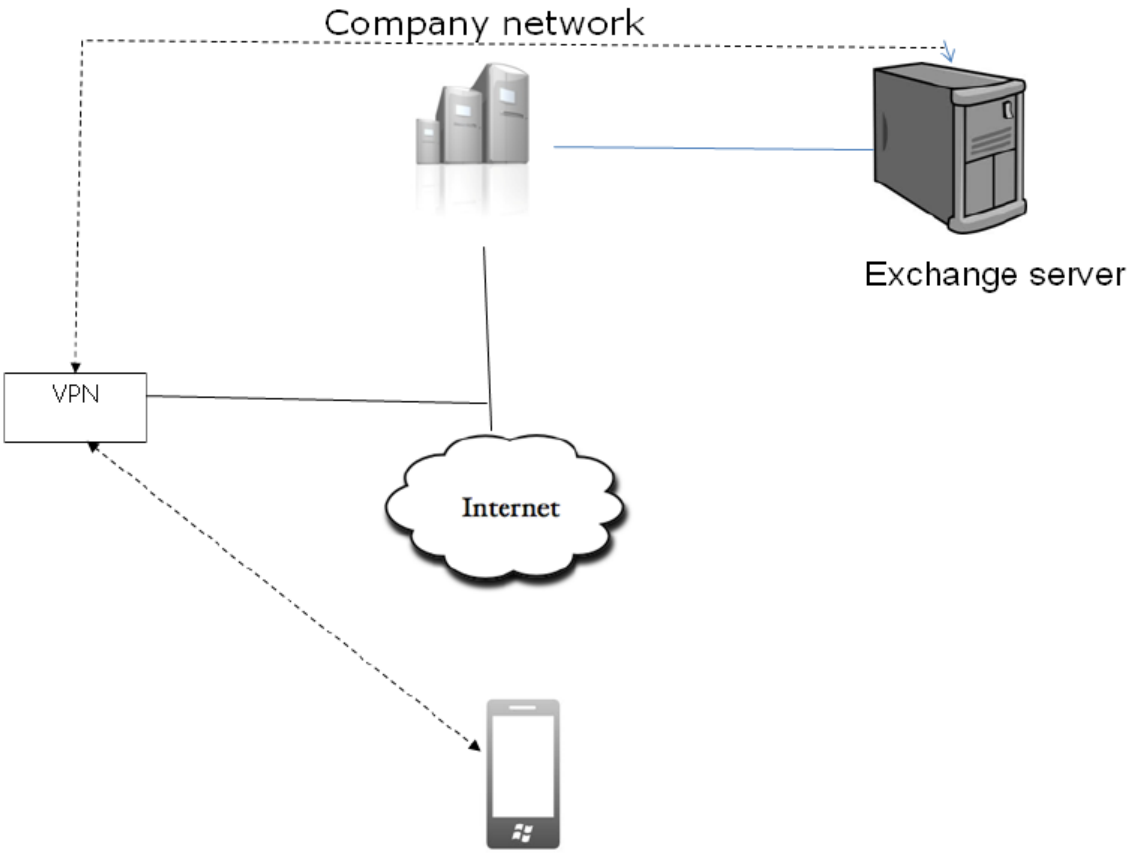

Figure 1: Mobile user accessing exchange server via VPN.
This model helps us to access our mails and contacts from company's exchange server directly via VPN without even entering company's private network.

1.3 DESIGN INTERFACE

Here we present a high-level overview of the main aspects of the design-

☐ **Efficient, event-driven concurrency:** To support massive degrees of concurrency, SEDA relies on event-driven techniques to represent multiple flows through the system. This design makes use of a small number of threads, rather than one thread per request. Non-blocking I/O primitives are used to eliminate common sources of blocking.

☐ **Dynamic thread pooling:** To relax the rigid scheduling and non-blocking requirements for event driven concurrency, SEDA uses a set of thread pools, one per stage, to drive execution. This not only frees the application programmer from having to implement event scheduling (as the operating system handles scheduling threads), but also allows event-handling code to block for brief periods of time, as additional threads can be allocated to a stage.

☐ **Structured queues for code modularity and load management:** By partitioning an application into a set of stages with explicit queues between them, application designers can focus on the service logic and concurrency management for individual stages, "plugging" them together into a complete service later. Queues decouple the execution of each stage, allowing stages to be developed independently.

Queues provide a point of control over the request stream in a service, as requests flowing across stages can be inspected and managed by the application. Likewise, admission control can be performed on a per-stage basis.

☐ **Self-tuning resource management:** Rather than mandate a priori knowledge of application resource requirements and client load characteristics, SEDA makes use of feedback and control to automatically tune various resource usage parameters in the system. For example, the system determines the number of threads allocated to each stage based on perceived concurrency demands, rather than relying on a hard-coded value set by the programmer or administrator

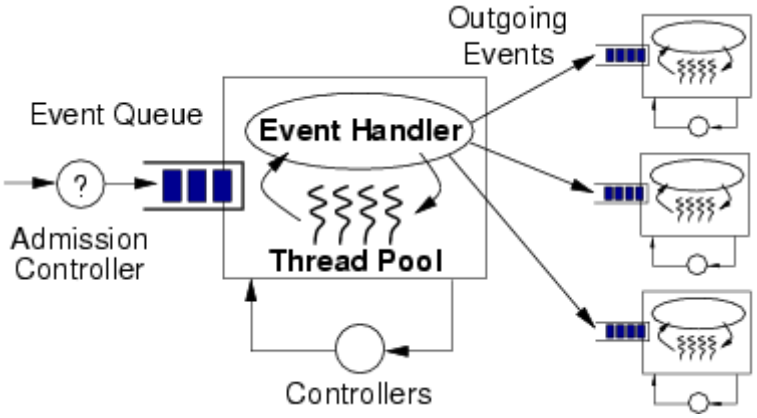

Figure 2: A SEDA Stage

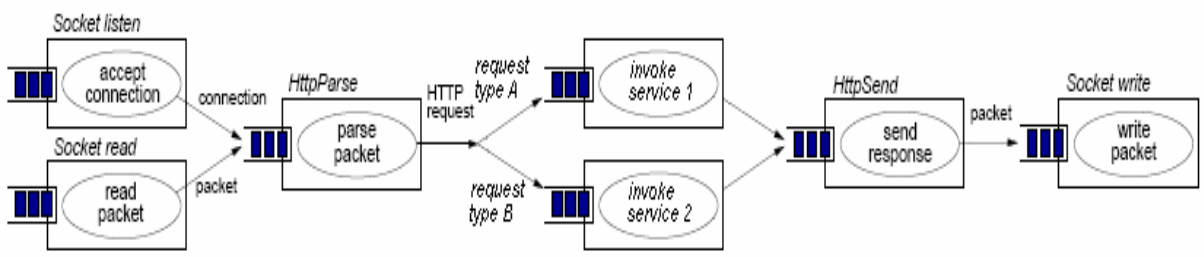

Figure 3: Our scenario mapped to the SEDA model

1.4 IMPLEMENTATION

After implementing our server, we will configure it as VPN server and use it for bringing our model to life. This is how we access a VPN server from a remote client which here will be our mobile device. A remote access VPN connection over the Internet enables a remote access client to initiate a dial-up connection to a local ISP instead of connecting to a corporate or outsourced network access server (NAS). By using the established physical connection to the local ISP, the remote access client initiates a VPN connection across the Internet to the organization's VPN server. When the VPN connection is created, the remote access client can access the resources of the private intranet. The following figure shows remote access over the Internet.

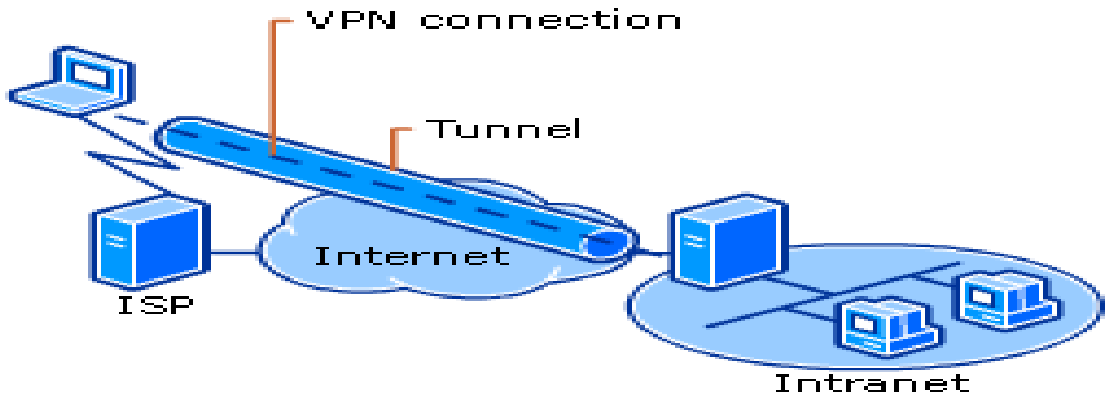

Figure 4: VPN Connecting a Remote Client to a Private Intranet

In this model as we have come up with the idea of providing a more open access of exchange server by any mobile device without going through company's network, or having to worry about the mobile device's company. The mobile device user will connect to the VPN which will provide it the direct access to the exchange server. Through this your mail exchange server will not even come on public IP and you will not even need to go through the organization's private network. This helps in providing more security to Private network and flexibility to users.

We have divided our server into seven stages. As shown in Figure 3.10, first two stages: Accept connection and Read packet for receiving the input packets coming from mobile device user. Then these requests are divided on the basis of whether they are meant for data on the server or meant to be forwarded, as done by a proxy server.

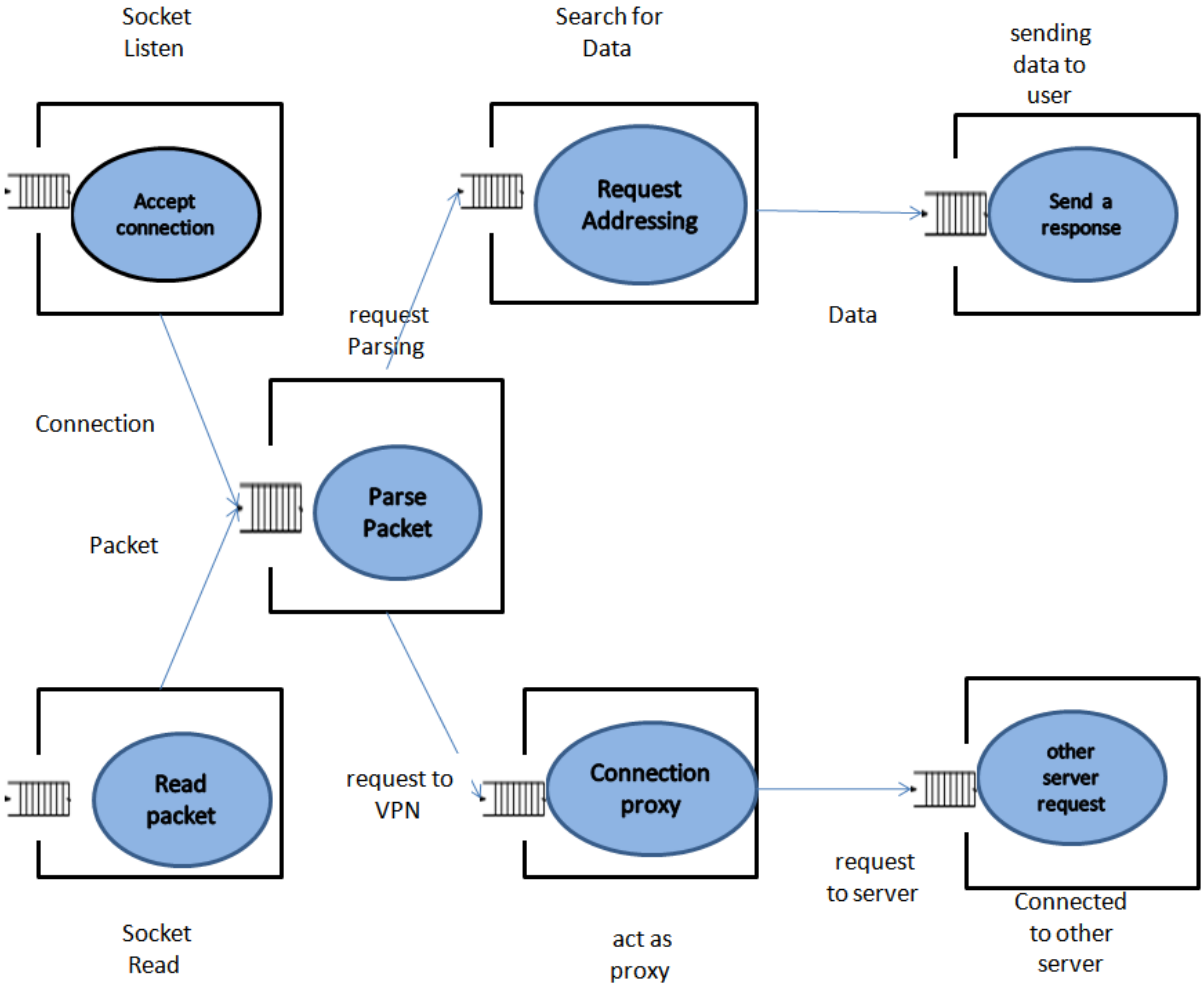

Figure 5: Staged event-driven (SEDA) server.

At Request Address stage, we look for response by the server itself, the other tasks handled by the VPN server. The next stage will be to send a response to the user. The other task handled by the server will be to forward the request of the user to Exchange server (after authentication) from where he will be able to synchronize the data.

This model does not involve entering into any organization's personal network via firewalls. It not only reduces our efforts but also provides security to company's data.

CONCLUSION:

This is how we implemented a Staged Event Driven Architecture (SEDA) server and then used it as a MS Active Sync proxy server which further helped us to access our mails and contacts from company's exchange server directly via VPN without even entering company's private network. Use of SEDA architecture helped in making the proxy server highly effective, concurrent, robust and at the same time highly responsive. Thus making SEDA a much better choice of architecture compared to multi-threaded server and Event driven server.

CONFLICTS OF INTEREST:

The problem with using SEDA server as a VPN server is that SEDA n/w depends on various parameters. For SEDA servers to perform better than multi threaded servers as well as Event driven architecture servers these parameters need to be tuned. SEDA did show "acceptable Performance", while providing good load Conclusion Measurement & Control is the key, as Opposed to fixed resource allocation.

Though the major challenges faced while using SEDA server was detecting overload conditions and to strategize and counter overload.

ACKNOWLEDGEMENTS:

On the outset we would like to thank our Head of Department Ms.(Prof.) Sangita Jadhav for giving us an opportunity to go ahead with the research work realated with the topic.
We would also like to thank our mentor and guide Mr. (Prof.) GM Walunjkar for making us understand the nitty -gritty of the topic and at the same time for providing us with all the necessary required equipments for the research work.

We would also like to thank all those people who directly or indirectly were a part in making this successful.

REFERENCES:

- **SEDA: An Architecture for Well Conditioned, Scalable Internet Services.** Matt Welsh, David Culler, and Eric Brewer Computer Science Division University of California, Berkeley fmdw,culler,brewerg@cs.berkeley.edu

- www.eecs.harvard.edu/~mdw/proj/**seda**/

- DESIGNING AND ACTIVE SYNC FOR POCKET PC USING WIRELESS TECHNOLOGY By Shalini Bhaskar.
- http://www.ijetae.com/files/Issue2/IJETAE_1211_16.pdf

- Concurrent programming for scalable web architectures By Benjamin Erb

- Evaluating the Scalability of Java Event-Driven Web Servers
  Vicenc¸ Beltran, David Carrera, Jordi Torres and Eduard Ayguade´
  {vbeltran, dcarrera, torres, eduard}@ac.upc.es
  European Center for Parallelism of Barcelona (CEPBA)Computer Architecture Department, Technical University of Catalonia (UPC)

- http://www.theserverside.com/news/1363672/Building-a-Scalable-Enterprise-Applications-Using-Asynchronous-IO-and-SEDA-Model
- Auto tune design and evaluation of SEDA Zhanwen Li, David Levy, Shiping Chen, John Zic