A Performance Study of Conventional and Bare PC Webmail Servers

Patrick Appiah-Kubi
Department of Computer and Information Sciences
Towson University
Towson, MD 21252, U.S.A.
appiahkubi@towson.edu

Ramesh K. Karne
Department of Computer and Information Sciences
Towson University
Towson, MD 21252, U.S.A.
rkarne@towson.edu

Alexander L. Wijesinha
Department of Computer and Information Sciences
Towson University
Towson, MD 21252, U.S.A.
awijesinha@towson.edu

***Abstract*-In this paper, we compare the performance of several Webmail servers and a bare PC Webmail server running without an operating system. The conventional Webmail servers used in the study are Icewarp, MailTraq and Hexamail running on Windows, and Atmail and Afterlogic running on Linux. Server performance is compared with respect to the typical email transactions (login, compose, and read), CPU utilization, and throughput. Performance under increased loads is measured by using a stress tool, and experiments are conducted in both LAN and WAN environments. The results indicate that bare PC Webmail server performance is consistent and predictable, whereas conventional Webmail server performance varies considerably depending on message size, server load and transaction.**

Keywords – Operating Systems; Bare Machine Computing; Webmail servers; Performance; Email.

I. INTRODUCTION

Web-based email or Webmail enables users to access email from any computer located anywhere using a Web browser such as Internet Explorer, Firefox or Google Chrome. Although popular Webmail systems such as gmail provide useful services, they are not designed for high performance or security. While there are some email/Webmail servers and systems that are designed for high-performance and/or security, they require an operating system (OS) to run. The performance of such systems is limited by the capabilities of the underlying OS and they are also susceptible to attacks that exploit the vulnerabilities of this OS.

In contrast, bare PC or bare machine servers and applications do not use an OS. Each application contains only essential functionality and has its own interfaces to the hardware. This eliminates OS overhead and enables the system to be optimized for performance by fully exploiting the capabilities of the underlying hardware. Moreover, bare PC applications are immune to conventional attacks that target a specific OS such as Linux or Windows. Many bare PC applications have been developed including Web servers, email servers and Webmail servers. Performance studies of bare PC applications serve to verify that the application provides the desired performance benefits and that it outperforms its OS-based counterparts when running on compatible systems. In this paper, we compare the performance of a bare PC Webmail server with several OS-based Webmail servers. The rest of this paper is organized as follows. Section II discusses related work and Section III provides a brief overview of the bare PC Webmail server. Section IV contains the performance results and Section V concludes the paper.

II. RELATED WORK

Atmail [2], MailTraq [17], Axigen [3], Afterlogic [1], Squirrelmail [23], Facemail [7], Adaptive Email [4], Petmail [21], Icewarp [12], Roundcube [22], Emailman [8], WinWebmail [24], and Hexamail [11] are just a few of the numerous Webmail systems in existence today. Some of these systems are designed for high performance, while others such as Webex [6] are designed for high reliability and availability. An email architecture to address problems associated with scalability and dependability due to conventional design approaches is proposed in [15]. There appear to be no studies that evaluate the performance of Webmail systems or servers. Techniques to improve performance of the Open Webmail system are discussed in [5]. In [13], an email server architecture, which is based on a spam workload and optimized with respect to concurrency, I/O and IP address lookups, is shown to significantly improve performance and throughput. The design and implementation of an email pseudonym server providing anonymity to reduce server threats and capable of reducing risks due to OS-based vulnerabilities is presented in [18]. The notion of semantic email is discussed in [19].

An email server that runs on a bare PC is the focus of [9] and [10]. However, the server does not support Webmail. Many bare PC applications including Web servers [16] and VoIP clients [14] have been previously developed. In [20], the design and implementation of a bare PC Webmail server are described and some preliminary performance results are presented. The present paper differs from earlier work in that it compares the performance of a bare PC Webmail server and several OS-based Webmail

 ISBN: 978-1-61208-133-5

servers in both routed LAN and WAN environments, and also under stress conditions.

III. THE BARE PC WEBMAIL SERVER

Only a brief overview of bare PC Webmail server internals is given here as its design and implementation were discussed in detail in [20]. Since bare PC applications run directly on the hardware without the support of an OS, they are self-supporting. The Webmail server includes lean implementations of the HTTP/TCP/IP/SMTP/POP3 protocols (that are intertwined with the server application), and an Ethernet driver.

CPU task and memory management and the concurrent processing of requests from multiple clients are done by the application itself, which is written in C++ except for some low-level assembly code. There are only 4 task types in the Webmail server application: The Main task, consisting of a loop that runs whenever no other task is running; the Rcv task that receives incoming packets and is used for Ethernet, IP, and TCP processing; and multiple Get and Post tasks that manage the processing of client requests. A given request or packet is processed as a single thread of execution. Once activated, a task runs to completion unless it has to wait for an ack or a timeout. Delay and Resume lists are used to efficiently manage the suspension and resumption of tasks. Get/Post are modeled using state transitions.

The application is initially booted from a USB flash drive and does not use a hard disk. The USB is also used for persistent storage of email messages and user information, but a separate server could be used for auxiliary storage in the future. The Webmail server currently runs on an ordinary PC (not a server machine). The main data structure used by the Webmail server (and all bare PC applications) is the TCB (Transmission Control Block) table that contains entries to enable the management of concurrent requests, associated data, and TCP/application state information. Get/Post tasks are placed in the Resume list when requests arrive and their active status is indicated by a flag in the TCB table. The Webmail server application includes a lean PHP parser that interprets client Get/Post data.

IV. PERFORMANCE RESULTS

A. LAN Setup

For the LAN studies, a dedicated test network consisting of five Ethernet switches (S1-S5) interconnected linearly by four Linux routers (R1-R4) was set up. The client (C) and Webmail server (WMS) were connected to the ends of the network so that messages between the client and Webmail server are routed along the following path:

C--S1--R1--S2--R2--S3--R3--S4--R4--S5--WMS

All switches were gigabit switches except for the 100Mbps switch (S1) used to connect the client to the network. The clients ran Windows XP and the OS-based Webmail servers ran Windows XP or Linux (CentOS). All machines were Dell Optiplex GX520s. OS-based Webmail server details are as follows: Afterlogic MailSuite Pro (Linux), MailTraq Server (XP), Atmail Server 6.20.3 (Linux), Icewarp Server 10.2.1 (XP), and Hexamail Server 4.0.1.002 (XP).

B. LAN Results

Fig. 1 below is derived from the Wireshark timestamps for each message in the sequence of messages exchanged during a login Get request. The difference in timestamps for a pair of consecutive messages such as (Get, Ack) or (Data, 200_OK) gives the delay between the pair. As expected, the performance for all servers during the initial TCP handshake is the same. There is a rise between the client Get request and the server Ack due to the server delay in processing the request. All servers show little variation in processing time for subsequent message pairs.

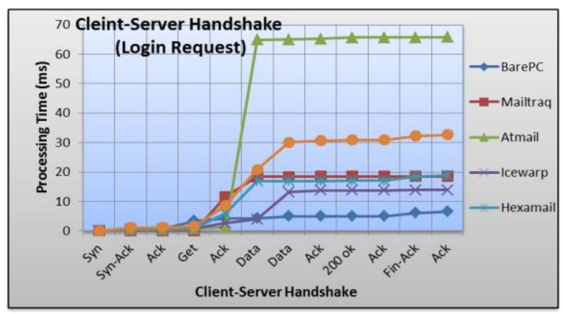

Figure 1. Login Get request message times

Similarly, Fig. 2 compares the processing time for a login Post request. The (Post, Ack) behavior for all servers except for Hexamail and MailTraq is the same. The (Ack, 302_Found) delay is visible for all servers except MailTraq. The bare PC server processing times for both Get and Post requests are minimal. Since different servers may do the work to process the requests during different steps, only the overall processing time should be compared. Fig. 3 shows the processing time for compose with varying message sizes. The varying behavior of the servers reflects the combination of TCP, HTTP and the mail server application. Hexamail has stable behavior for large message sizes, while Icewarp shows the most variation. The bare PC server has the highest processing delay for a message of 10,000 bytes, but shows a general reduction for larger sizes except for a

 ISBN: 978-1-61208-133-5

small rise at 20,000 bytes. Fig. 4 shows the processing time for receiving an inbox with 6 messages. While all servers complete processing in about 1.1 milliseconds on the average, the bare PC server requires less than 0.1 ms.

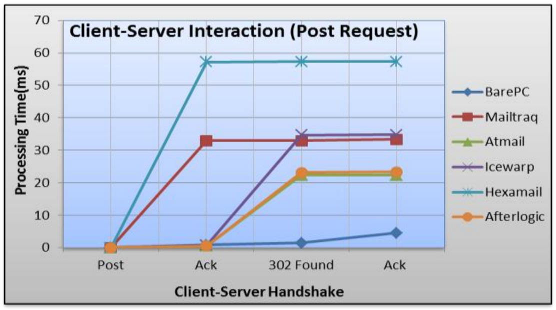

Figure 2. Login Post request message times

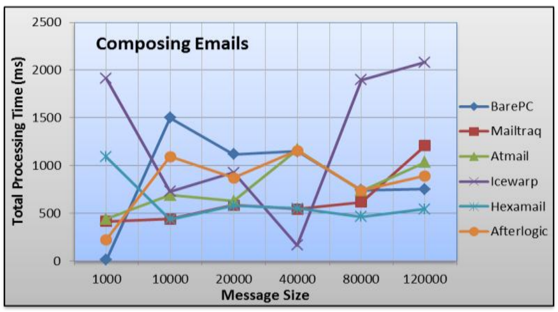

Figure 3. Processing time for compose (varying message sizes)

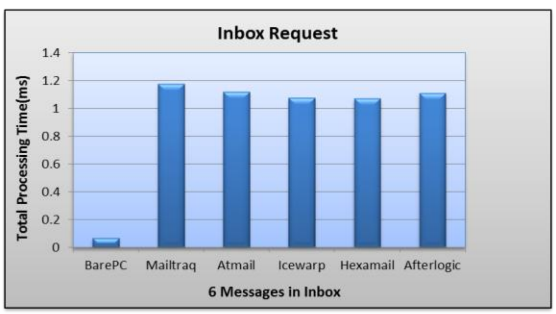

Figure 4. Processing time for an Inbox request (6 messages)

Fig. 5 shows the processing time to retrieve messages of sizes 1000-120,000 bytes. Once a message is retrieved into an inbox, it takes less time to process and transmit the message to the client. Processing time for the bare PC is minimal, and it has the smallest increase in processing time. Among the OS-based servers, Icewarp followed by Afterlogic have the lowest processing times (except for a 1000-byte message), while Icewarp and Hexamail have the smallest increase in processing time (the latter actually has the highest processing time). Fig. 6 shows the throughput measured during compose for increasing message sizes. The bare PC server throughput is highest and approximately twice the throughput of the best OS-based server Afterlogic. The low throughput of Icewarp reflects its large processing time in Fig. 3.

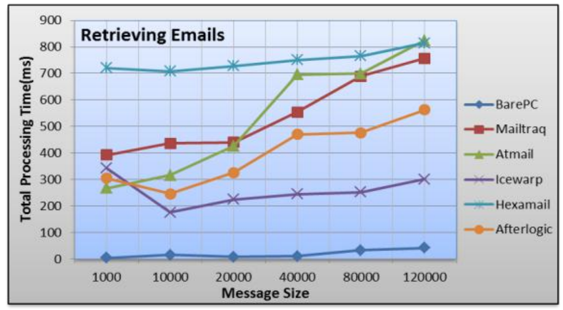

Figure 5. Processing time for read (varying message sizes)

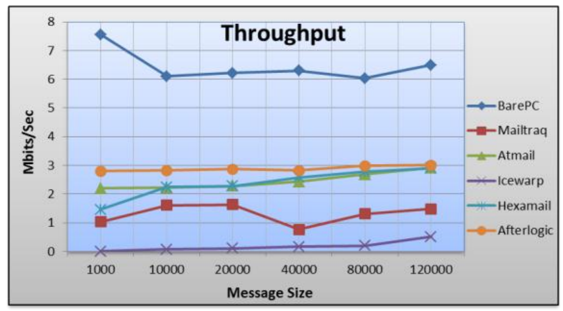

Figure 6. Throughput for compose (varying message sizes)

Further performance tests were conducted on the Webmail servers using the Web stress tool [25]. The tool was used to increase the number of users from 1 to 10 and determine the resulting impact on performance. Each test was run for 10 minutes and each user makes 100 requests/s. Fig. 7 illustrates the variation of server CPU utilization over time for a maximum of 10 users. The average CPU utilization of the Linux-based Afterlogic and Atmail servers and the bare PC server is less than 4%, while that of the Windows-based Mailtraq, Icewarp and Hexamail servers is between 8-12%. It is evident that more CPU processing is required by the Windows-based servers when processing concurrent requests. The figure also indicates that the CPU utilization of the bare PC server shows some slight initial variability compared to that of the Atmail server.

Fig. 8 shows the variation of server bandwidth over of time for 10 users. It can be seen that the bandwidth of all servers is relatively stable after the initial increase during the first 5 seconds. However, while there is little difference between the bandwidth of the OS-based servers (average < 60 kbps, maximum <700 kbps), the bandwidth of the bare PC server is significantly higher (average and maximum exceed 100 kbps and 12 Mbps respectively).

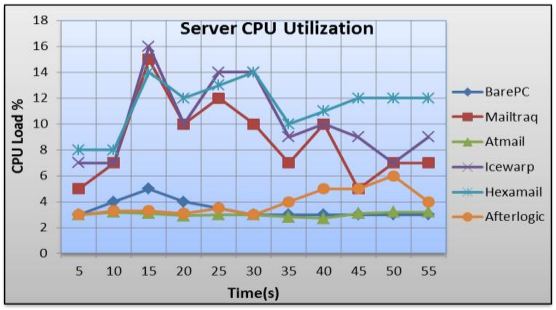

Figure 7. CPU utilization for 10 users

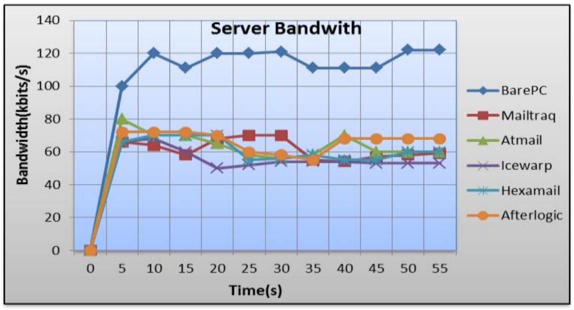

Figure 8. Bandwidth variation for 10 users

Figure 9 above is the average time (delay) to complete a Post request for a varying number of users. Even two users cause the delay to increase significantly compared to one user, but the delay stabilizes for three or more users. The bare PC has the least delay, while the Windows-based servers have the highest delay in this case. It can be seen that the delays for the bare PC server and Linux-based servers differ by almost 100 ms.

Fig. 10 shows the amount of time a user waits for the server to establish a connection in the presence of multiple users. The Linux and bare PC servers perform much better than the Windows servers, but the performance advantage of the bare PC server compared to the Linux servers is reduced since the performance of the latter improves significantly when there are 6-10 users. Figs. 11 and 12 show respectively the Webmail server processing times for a read request and the throughput for a compose request with and without stress. To create stress, the tool is used to generate 100 concurrent requests/s from 10 users and an additional client is used to generate the read or compose request involving an email message of 120,000 bytes. Although performance degrades under stress for all servers as expected, the bare PC server's performance with and without stress is significantly better than the performance of the OS-based servers. However, no simple relationship exists between throughput and processing time for the OS-based servers (for example, Icewarp and Mailtraq have the lowest throughput, but respectively low and high times).

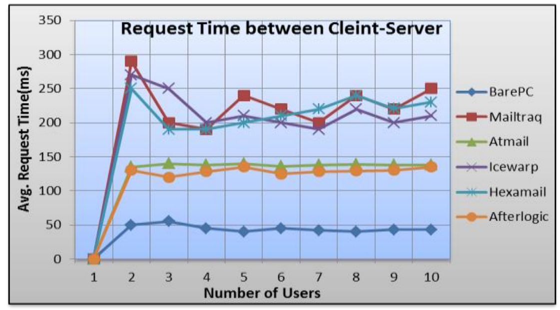

Figure 9. Post request completion time

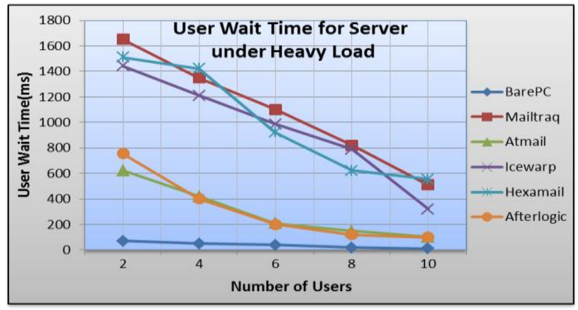

Figure 10. User wait time (increasing number of users)

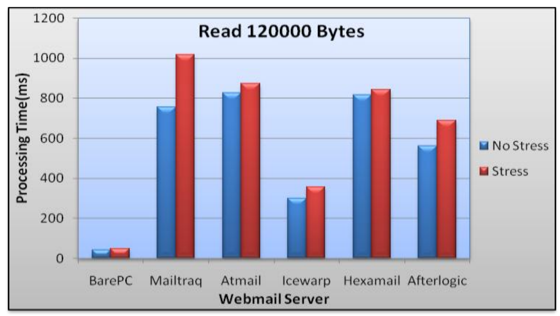

Figure 11. Message read time (120000-byte message)

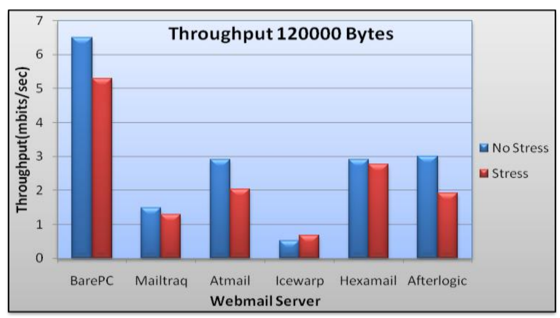

Figure 12. Throughput for compose (120000-byte message)

C. *WAN Setup*

For the WAN tests, an Internet connection was established between each Webmail server and a client PC located approximately 50 miles away with about 30 hops between the two destinations. To ensure consistency, tests were performed during a contiguous time and repeated several times to ensure that the results were stable and independent of varying network conditions. As before, a Wireshark

 ISBN: 978-1-61208-133-5

packet analyzer was used to capture the data. The machines used were the same as for the LAN studies.

D. WAN Results

Figs. 13 and 14 are derived from the Wireshark timestamps for each message in the sequence of messages exchanged over the WAN during a login Get or Post request (they correspond to Figs. 1 and 2 for the LAN tests). As before, the difference between cumulative processing times for a pair of consecutive messages such as Get-Ack for Get or Post-Ack for Post gives the delay between the pair. It can be seen that these delays for Get and Post requests are significantly less for the bare PC server than for the OS-based servers.

A closer examination of Figs. 13 and 14 reveals that the Get and Post delays for the OS-based servers vary considerably across message pairs. For example in Fig. 13, MailTraq has the highest Get-to-Ack time, Atmail has the highest Ack-to-Data time, and Icewarp has the highest Data-to-200_OK time. However, Atmail has the lowest Get-to-Ack and Data-to-200_OK times among the OS-based servers. Similarly, compared to the Windows servers, Afterlogic has lower Get-to-Ack and Data-to-200_OK times, but a higher Ack-to-Data time. In case of a login Post request (Fig. 14), it can be seen that MailTraq and Atmail have respectively the highest and lowest (next to the bare PC) Post-to-Ack time, whereas Atmail has the highest and MailTraq has the lowest (next to the bare PC and Icewarp) Ack-to-302_Found time.

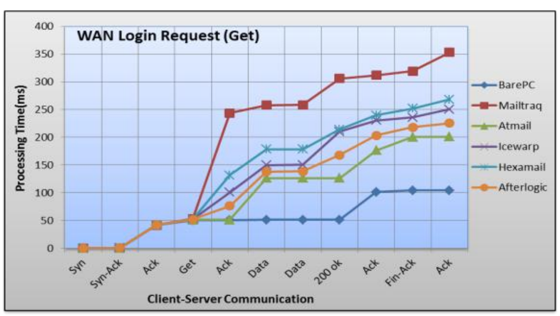

Figure 13. Processing time for login Get request

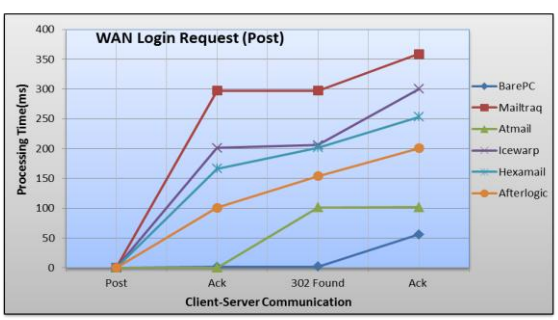

Figure 14. Processing time for login Post request

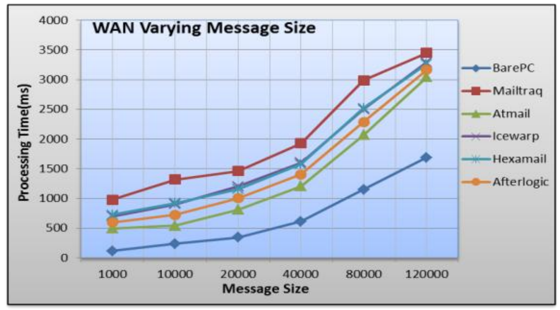

Figure 15. Processing time for compose (varying message sizes)

Figs. 15 shows the processing time over the WAN for compose, with message sizes varying from 1000 to 120,000 bytes. Processing time increases in an approximately linear manner as the message size increases. This was not the case for the corresponding LAN result in Fig. 3. However, it can be seen that the processing time for all servers increases at a higher rate for message sizes from 40,000-120,000 bytes.

Fig. 16 shows the processing time over the WAN for receiving an inbox containing 6 messages. The bare PC receives the inbox in 0.392 milliseconds, while the other servers require an average time of about 230 milliseconds. Fig. 17 shows the processing time on the WAN for reading individual emails of varying message sizes. The processing time for the bare PC server is stable up to 40,000 bytes and increases slowly thereafter for larger messages. The processing times on the other servers are stable up to 20,000 bytes, but rise sharply to 1400 milliseconds.

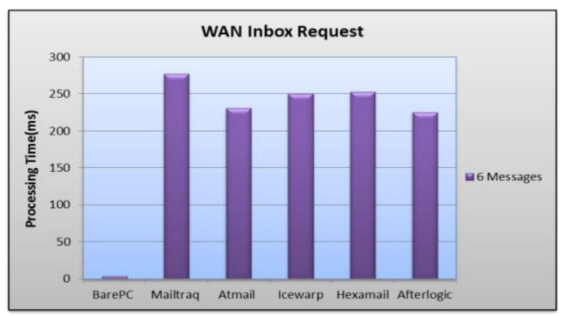

Figure 16. Processing time for an Inbox request (6 messages)

Figure 17. Processing time for read (varying message sizes)

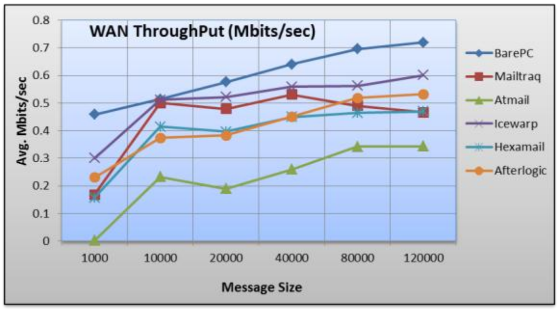

Figure 18. Average throughput (varying message sizes)

The throughput for varying message sizes was also captured during the Internet test and compared in Fig. 18. The average throughput for all servers is about 1.28 times better than the throughput of the Atmail server, whose performance in general on the previous tests was better than the other OS-based servers.

V. CONCLUSION

We conducted a performance study of six Webmail servers including a bare PC server with no operating system. Server processing times, CPU utilization and throughput in LAN and WAN environments and under stress conditions were compared with respect to common email transactions. The results show that performance of the OS-based servers is variable and no single server performs consistently better than the others on all tests. There appears to be no simple relation between LAN and WAN results even for the case of a single server. With a few rare exceptions, only a small drop in performance was seen under stress conditions. However, a detailed study under real workloads and conditions would be needed to determine the ability of servers to handle stress. As expected, the bare PC server performs significantly better on all tests with a few minor exceptions. This suggests that some of its novel design features could serve as a baseline for designing secure high-performance Webmail servers in the future.

REFERENCES

[1] Afterlogic Webmail Server, www.afterlogic.com Retrived on May 20th, 2010

[2] Atmail-Linux Webmail Server, www.atmail.com Retrived on October 5th, 2009.

[3] Axigen Email Sevrer, www.axigen.com Retrived on January 23th, 2009.C. Yeh and C. Mao, Adaptive e-mail intention finding mechanism based on e-mail words social networks. Workshop on Large Scale Attack Defense, 2007.

[4] C. Tung and S. Tsai, Tuning Webmail performance-The design and implementation of Open Webmail. National Cheng Kung University, www.openWebmail.org Retrived on May 20th, 2010.

[5] Cisco WebEx Mail: High availability design for your most mission critical application. www.ciscowebexmail.com. Retrived on May 20th, 2010.

[6] E. Lieberman and R.C. Miller, Facemail: showing faces of recipients to prevent misdirected email. 3rd Symposium on Usable Privacy and Security, 2007.

[7] Emailman Webmail Server, www.emailman.com Retrived on May 21th, 2010.

[8] G. Ford, R. Karne, A. L. Wijesinha, and P. Appiah-Kubi, The design and implementation of a bare PC email server, 33rd Annual IEEE International Computer Software and Applications Conference (COMPSAC), 2009.

[9] G. Ford, R. Karne, A. L. Wijesinha, and P. Appiah-Kubi, The performance of a bare machine email server, 21st International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD), 2009.

[10] Hexamail Webmail Server, www.hexamail.com Retrived on June 10th, 2010.

[11] IceWarp Mail Server, www.icewarp.com Retrived on May 20th, 2010

[12] A. Pathak, S. A. R. Jafri, and Y. C. Hu, The case for high-performance spam-Aware mail server architecture. 29th IEEE International Conference on Distributed Computing Systems, pp. 155-164, July 2009.

[13] G. H. Khaksari, A. L. Wijesinha, R. K. Karne, L. He, and S. Girumala. A peer-to-peer bare PC VoIP application. IEEE Consumer and Communications and Networking Conference, Las Vegas, Nevada, January 2007.

[14] E. Kageyama, C. Maziero, and A. Santin, A pull-based email architecture. ACM Symposium on Applied Computing, Fortaleza, Ceara, Brazil, 2008, pp. 468-472.

[15] L. He, R. Karne, and A. L. Wijesinha, "The design and performance of a bare PC Web server", International Journal of Computers and Their Applications, vol. 15, June 2008, pp. 100-112.

[16] Mailtraq email server-the complete email server, www.mailtraq.com Retrived on June 10th, 2010.

[17] D. Mazieres and M. F. Kaashoek, The design, implementation and operation of an email pseuodonym server. 5th Conference on Computer and Communications Security, 1998, pp. 27-36.

[18] L. McDowell, O. Etzioni, A. Halevy, and H. Levy, Semantic email. 13th International Conference on World Wide Web, New York, NY, USA, 2004, pp. 244 – 254.

[19] P. Appiah-Kubi, R.K. Karne, and A. L. Wijesinha, The design and Performance of a bare PC Webmail Server. 12th IEEE International Symposium of Advances on High Performance Computing and Networking. September 2010.

[20] Petmail Webmail Server, www.petmail.lothar.com Retrived on June 12th, 2010

[21] RoundCube Webmail Server, www.roundcube.com Retrived on November 30th, 2010

[22] SquirrelMail Webmail, www.squirrelmail.com Retrived on November 30th, 2010

[23] WinWebmail Server, www.winWebmail.net Retrived on October 20th, 2010

[24] Web Stress Tool 7, www.paessler.com Retrived on May 20th, 2010