ISSN XXXX XXXX © 2017 IJESC

Research Article Volume 7 Issue No.9

Using PHP and MySQL for SMTP Configuration and Database Creation

Madhurai Das
Student
Department of Computer Science and Engineering
SRM University, Ramapuram, Tamil Nadu, India

Abstract:
This paper has been presented to portray a basic idea of PHP using MySQL Database Connectivity. This will tell about how a database can be easily created using MySQL. Moreover, the database can be connected to PHP. It provides a variety of features to the users like they can use it to send an email by clearly specifying the sender and the receiver address and most importantly after fully completing the SMTP configuration. SMTP stands for Simple Mail Transfer Protocol. It is easy to configure SMTP with Gmail for its simple user interface or GUI. This helps the users to save time by reducing the manpower needed to type each and every mail individually. Now the database created helps us to remove the files, registers, notebooks or any sort of physical storage from the picture. Thus, in this way even if there is loss of files and notebooks, it will not affect the normal working process of the firm, company or the organization.

Keywords: Database, PHP, MySQL, Server, SMTP

1. INTRODUCTION

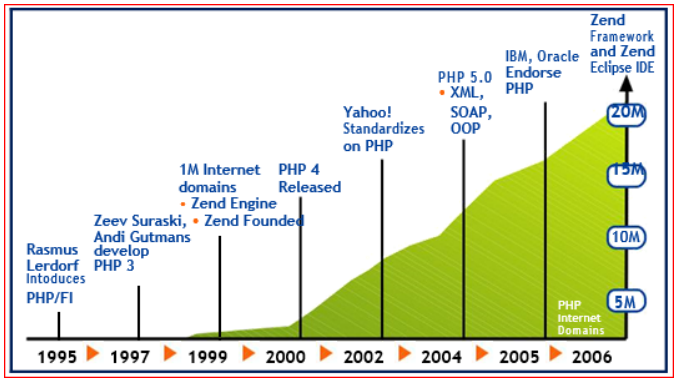
PHP is an important player in the software development market. Its popularity with both private software developers and corporate IT departments is having a growing impact on the market for commercial software as well. While PHP is heavily used in the development of web based software, it is not limited to this field. It is also used quite frequently to tie together different existing applications or application modules. Such is the case with Yahoo that uses PHP to "glue" together code written in different languages. The purpose of this paper is to summarize the history of PHP and offer an overview of its current status and market. We will then highlight the advantages of adopting PHP at the Enterprise level and how programmers and IT managers can benefit. Finally, we will give an outlook on the future development of PHP. This paper presents a performance study of the ISP (Internet Service Provider) mail servers. By using the SPEC-mail2001 benchmark to test a commercial mail service– MDaemon 5.0.1, we explore the both network and I/O performance of mail servers with user population's between 200 to 10,000. File system traces are also collected for analysis.

2. HISTORY OF PHP

The history of PHP is a long never ending story. The development of PHP started in 1995: Rasmus Lerdorf created a personal collection of Perl[1] scripts and transferred them into a package written in C. This package was called Personal Home Page tools or PHP for short. On June 8th, 1995 this package was published as PHP/FI, where FI stood for Form Interpreter. PHP/FI showed great similarities to Perl, while attempting to be much simpler to use. More than two years later, Rasmus Lerdorf published a more advanced version of the increasingly popular software called PHP 2.0 (or PHP/FI 2). In 1997, ZeevSuraski and Andi Gutmans began a complete re-write of PHP to make the language more powerful for eCommerce applications. They cooperated with Rasmus Lerdorf and changed the meaning of PHP to "PHP: Hypertext Preprocessor".[2] their efforts resulted in PHP 3.0 and were published in June of 1998. This collaboration gave PHP strong extensibility features that made it very easy to write additions and extensions for the language. At this point, initial buds of object orientation were integrated in the language's syntax. The dramatic growth of the WWW in the late 90s, created an enormous demand for scripting languages for dynamic website development. With the release of PHP 3, the Internet community found a tool that was easier to learn and handle than Perl (the defacto standard at the time) and PHP rapidly became the language of choice for dynamic content. By the end of 1998, the install base for PHP reached several hundred thousand websites. During the next two years, ZeevSuraski and Andi Gutmans, developed PHP 4. When it was published in May 2000, the main new features included simple object orientation and session handling capabilities. To increase the performance and stability of PHP execution on the server, the Zend Engine 1 was introduced as the heart of PHP installation. In May 2000, the number of domains that use PHP reached 2 million[3].

The PHP community, with the strong participation of ZeevSuraski and Andi Gutmans spent the next four years creating PHP 5. This long development cycle was mainly due to a very extensive testing phase. When PHP 5 was finally released in the summer of 2004, together with the Zend Engine 2 it introduced major enhancements: such as full support for object orientation, XML[4] integration, and the SOAP[5] protocol. By the summer of 2004, PHP had grown to an install base of more than 16 million domains.

The transition from PHP 4 usage to PHP 5 was slow in the beginning, however this changed with the release of PHP 5.1 in late 2005. In addition to smaller enhancements, PHP 5.1 brought a database abstraction layer called PDO[6]. PDO makes it much easier to use PHP with various databases from different vendors. By the end of 2005, PHP adoption soared to more than 23 million domains.



PHP – A Success Story

The PHP community, with the strong participation of ZeevSuraski and Andi Gutmans spent the next four years creating PHP 5. This long development cycle was mainly due to a very extensive testing phase. When PHP 5 was finally released in the summer of 2004, together with the Zend Engine 2 it introduced major enhancements: such as full support for object orientation, XML[4] integration, and the SOAP[5] protocol. By the summer of 2004, PHP had grown to an install base of more than 16 million domains.

The transition from PHP 4 usage to PHP 5 was slow in the beginning, however this changed with the release of PHP 5.1 in late 2005. In addition to smaller enhancements, PHP 5.1 brought a database abstraction layer called PDO[6]. PDO makes it much easier to use PHP with various databases from different vendors. By the end of 2005, PHP adoption soared to more than 23 million domains.

3. MANAGEMENT SUMMARY

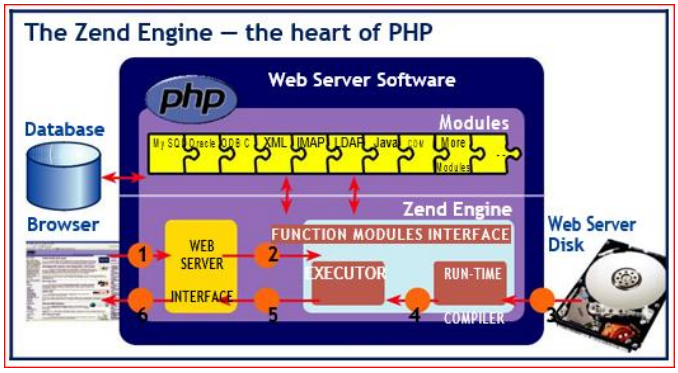
PHP is at the forefront of Web 2.0 and Service Oriented Architectures enabler technologies along with other open source projects MySQL, Apache or JBoss. With ten years of development behind it, PHP is a relatively young programming language.

Nevertheless, millions of developers world-wide use PHP to develop systems that power over 20 million websites. PHP is endorsed not only by its large open source community, but also by leading players and analysts in the IT market such as IBM, Oracle and Microsoft. PHP is ideal for Enterprise applications as it is easy to learn yet robust and flexible enough to power business critical applications.

4. PHP TODAY

Today, PHP is a full featured comprehensive programming language with solid object orientation support. While it was called a scripting language in the past, today it is more referred to as a dynamic programming language. Unlike traditional programming languages such as C/C++, PHP does not have to be compiled.

Instead the source code is interpreted at runtime The Zend Engine acts as a runtime interpreter that compiles the code in real time. Historically, interpreted languages suffered from lower performance (as compared with compiled languages). The PHP community has worked hard to overcome these issues, and today, it is possible for properly configured interpreted languages to approach execution speeds of compiled languages.



ZEND ENGINE

One of the major advantages PHP offers is **platform independence**. Currently, the list of supported operating systems includes Linux (for various CPU architectures), Microsoft Windows, Mac OS X, Sun's Solaris (SPARC and Intel), IBM AIX, HP-UX, FreeBSD, Novell Netware, SGI IRIX, IBM AS/400, OS/2 and RISC OS. Since the Zend Engine is open source, it is straightfoward to compile it for additional operating systems. It is no surprise that there are adaptations for all common UNIX versions, as well as for exotic environments like Amiga or BeOS.

Platform independence has a second facet:

Most PHP applications can therefore be used on every computer or internet capable device. Another benefit of PHP is **flexibility**. Since no compilation is needed, it is easy to make changes or bug fixes within minutes and to deploy new versions of the program frequently. Additionally, it is e**asy to prototype** new applications and concepts; typically compared to C or Java, PHP application development takes 50% of the time. Whilst PHP5 includes SQLite database as a standard install for rapid prototyping, PHP applications usually store their data in external databases, such as MySQL, Oracle or IBM DB2. PHP5 incorporates a database abstraction layer "PDO" for developing applications that support multiple databases. As PDO currently supports most of the commercial databases on the market and is part of an open source project, other drivers can be easily added as new technologies emerge.

Since PHP is an open source language, all information on how the language works, its syntax and the inner functions of the Zend Engine are publicly available and can be used free of charge under the PHP license. This is a BSD-style license, there are no licensing fees related to using PHP or the Zend Engine in applications whether commercial or otherwise and there are no limitations on commercial use as there are with other open source projects released under the GPL license.

4. PEOPLE BEHIND PHP

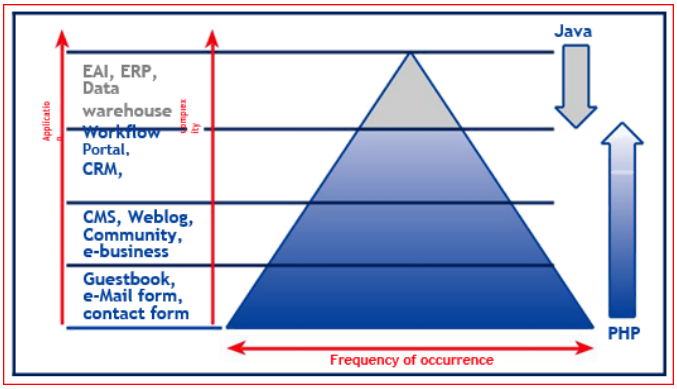
The development of PHP is driven by the PHP Group, a community of developers that contribute to the various parts and modules of PHP. The PHP Group is also the official licenser of PHP. It is hard to give an exact number of how many people work on PHP, since literally anyone can contribute to the project. It is easier to make a rough distinction between the people working on the core language and the Zend Engine and the larger number of people working on individual PHP extensions and libraries. There are also several companies, whose staff contributes to the development of PHP. Among those involved, Zend Technologies founded by ZeevSuraski and Andi Gutmans, Yahoo, ezSystems, OmniTI and others.

5. THE PHP MARKET

PHP ranks among the most popular programming languages today and since 2004 is the undisputed leader in the web programming market. According to a recent survey from Nexen.net, PHP has a market share of more than 30 percent[7]. The number of internet sites using PHP is around 20 million. However, this figure does not take into consideration the growing number of internal corporate servers used for intranet applications or development purposes - statistics about this usage is still unclear. There are more than 15.000 worldwide users of Zend Technologies' commercial products. The total number of professional software developers worldwide is estimated to be about seven million. According to a recent survey of the popular UK-based IT news site, The Register, it is estimated that more than 2.5 million of them have experience with PHP programming. More and more universities are taking into account PHP's growing role and their syllabi offer relevant courses. In the summer of 2004 Zend Technologies established a certification program together with Pearson-VUE. More than 1,000 PHP developers have become Zend Certified Engineers (ZCE) since then. PHP's penetration and market shares differ from region to region. Since the basic combination of Linux, Apache, MySQL and PHP (without additional commercial tools) provides an almost enterprise grade application platform without any licensing costs, PHP's market share is very high in developing countries. The Ukraine and Sao Tomé for example lead the list of countries with the highest market share for PHP with 69.7 and 68.2 percent respectively[8]. In the leading industrial nations, USA, Germany and Japan hold largest PHP markets, followed by France, UK, Canada, Italy and The Netherlands. There are a vast number of PHP programs available. A considerable part of them deal with dynamic website content management such as Typo3, ezPublish, Drupal and Joomla, eCommerce systems like Oxidesales, OS Commerce and Zen Cart, blog software like Wordpress, Serendipity, WIKIS like Tiki, Mediawiki (Wikipedia) or discussion forum applications like phpBB, vBulletin and FUDforum.

6. THE FUTURE OF PHP

In the past PHP open source principles have put greater overhead on individual developer choices. Unlike Java, every developer had the possibility to do as they pleased since there was little language and systems standardization. Over the past 18 months this gap has been minimized by the introduction of a set of best practices and standardizations and will continue this orientation in future with the growth of PHP frameworks such as Zend Framework Past conceptions held that PHP was only capable of elementary tasks such as guest books, simple Web sites etc. However PHP is also increasingly used for other types of applications up to a certain level of complexity. The important benefit of PHP is that it does not go around the function range of an application, but rather around the complexity of the connections in application. Therefore large applications can also be written with PHP, as long as they are divided into components. As this method of development is becoming more in demand by Ajax and SOA in the IT surrounding fields, PHP will consequently advance into additional domains. As a result, PHP goes hand in hand with the IT trends to divide large applications into small services and is clearly in line with the direction of IT management strategies in large enterprises. Bearing in mind the advantages of PHP and the technology evangelism activities of Zend Technologies' it seems a logical consequence that PHP would receive a strong endorsement from the IT industry Zend cooperates with Sun Microsystems to optimize the PHP/Java interoperability. By collaborating with IBM and Oracle, Zend has created **Zend Core™**, stable and supported stacks of PHP and database drivers' desgined for their major databases. Intel and SAP demonstrated their trust in the PHP market by investing into Zend Technologies through their venture capital funds. Finally, Microsoft realizes the potential of PHP and has announced a technical collaboration to improve Interoperability of PHP on the Windows Server Platform.



TECHNOLOGY FOLLOWS REQUIREMENT

PHP will increase in importance in Enterprise environments while keeping a leading position as an easy to use programming language for web applications.

7. SQL SERVER

These realities of business are driving the next wave of innovation on the Microsoft platform. To support those needs for today and tomorrow, we are investing heavily into several core areas as SQL Server continues to evolve:

Performance: SQL Server's integrated in-memory toolset goes far beyond isolated features andprovides support for improving performance dramatically in a wide range of scenarios.

Cloud services: New tools in SQL Server and Microsoft Azure make it even easier to scale to thecloud; to build patching, backup and disaster recovery solutions; and to access resources wherever they are — on-premises, private cloud or public cloud.

Security and compliance: As SQL Server progresses, we're adding new capabilities to protectdata both at rest and in motion, with new features like Always Encrypted and Row-Level Security.

Availability:

Already known for rock-solid, reliable performance, SQL Server is adding significantnew enhancements to AlwaysOn, better load balancing, and new features for flexible and efficient backups.

Scalabilty: New advancements in compute, storage and networking will provide direct impacton mission-critical SQL Server workloads.

8. MISSION CRITICAL APPLICATION PERFORMANCE WITH SQL SERVER

Today the dramatic shift toward complex, unstructured data types requires organizations to embrace back-end solutions

that support all data sources, while also providing high-performance, mission-critical capabilities. IT organizations need to balance this amplified importance of data with other pressures, such as globalization, conservative budgets, and ever-tightening compliance policies and regulations that are increasingly strict in many areas of the world. And they need to accomplish it all with higher levels of uptime and performance, with layered defenses that provide the utmost security. For SQL Server 2016, we are continuing to push the envelope on mission-critical performance with new innovations across many components covering performance, security, availability and scalability.

Performance

Data volume is increasing exponentially, and the ability of today's analysis and business intelligence tools to derive insight from that data is more important than ever. Modern database technologies must be able to take advantage of these amplified data streams across faster, parallel processors and great reservoirs of storage in order for businesses to compete.
With SQL Server 2014 and the upcoming SQL Server 2016, performance is enhanced with a number of new technologies, including in-memory, query store, JSON, and temporal support, to name a few.

In-memory online transaction processing

SQL Server was originally designed when it could be assumed that main memory was very expensive, so data needed to reside on disk except when it was actually needed for processing. This assumption is no longer valid as memory prices have dropped enormously over the past 30 years. At the same time, multicore servers have become affordable, so that today one can buy a server with 32 cores and 1TB of memory for under $50K. Because of this trend to much more available memory and many more cores, the SQL Server team at Microsoft began building a database engine optimized for large main memories and many-core CPUs.

30x performance gains

In-memory technology for SQL Server dramatically improves the throughput and latency of SQL Server OLTP capabilities. It is designed to meet the requirements of the most demanding transaction processing applications, and Microsoft has worked closely with a number of companies to prove these gains — Dell has achieved a 9x increase in performance. Beth Israel Deaconess Medical Center has cut query times by 75 percent while tripling its data storage from 30 days to 90 days. We are continuing with our design point of workload optimized in-memory, but now allowing you to gain real-time insights on operational data with the ability to run a columnar index over your in-memory or on disk row store. What's unique here is you can gain the speed of In-Memory OLTP and have the ability to gain operation analytics. Even if you are not using in-memory you can gain operational insights. Also you can apply our in-memory technology to more of your applications than ever before with expanded T-SQL surface area support. For SQL Server 2016, we are still keeping to the workload-optimized approach as customers want to optimize in-memory by workload. When it comes to In-Memory OLTP you will now be able to apply this tuned transaction performance technology to a significantly greater number of applications with expanded T-SQL surface area. In addition to providing up 30x performance gains you will now be able to gain real-time operational insights on your operational data. This data can be in-memory or on disk.

Security and compliance

The security landscape has changed dramatically over the years, but it's as important as ever. Today organizations must contend with an explosion in devices and device types, varying network technologies across the world, and data that resides in a multitude of formats and platforms — including the cloud. How do you protect data that is constantly in motion? How do you provide the right access to the right people at the right time? As SQL Server continues to evolve, we're adding new capabilities to protect data both at rest and in motion, with new features like Always Encrypted and Row-Level Security. Other new features enhance security in a multitenant environment, with fine-grain access control based on user attributes such as location, role and more. With new capabilities for 2016, SQL Server is continuing to evolve to provide the kinds of finely tuned, granular control that companies need today.

Transparent Data Encryption

SQL Server's Transparent Data Encryption (TDE) allows organizations to encrypt data when it is stored on a disk, and decrypt it when it is read into memory. TDE uses a database encryption key (DEK), which is stored in the database boot record for availability during recovery. The DEK is a symmetric key secured by using a certificate stored in the master database of the server or an asymmetric key protected by an EKM module. Simply put, TDE protects data at rest, meaning the data and log files. This enables software developers to encrypt data by using AES and 3DES encryption algorithms, without changing existing applications. Encryption and decryption operations are handled by the database engine in the background. Therefore, organizations do not have to make changes to their applications for SQL Server to secure their data. Because encryption is built into the database engine, it is transparent to applications and users — and it is included in SQL Server Enterprise edition. In addition, extensible key management works with TDE to store encryption keys outside of the database. With extensible key management, organizations can use a hardware device or a third-party encryption tool to create encryption keys. Storing the keys separately from the encrypted data makes it even harder for unauthorized users to gain access to encrypted data. For databases protected by TDE, backups of those databases are also encrypted.

9. ENHANCED FEATURES OF NEW SQL VERSIONS

TDE now supports storage of memory-optimized OLTP Tables. This allows for greater security along with the performance enhancements provided by memory-optimization. Dynamic Data Masking limits exposure to sensitive data by obfuscating it for nonprivileged users. This feature enables you to set up policies at the table and column level that provide multiple masking functions, such as obfuscating the first eight digits and displaying the last four digits of an ID or credit card number. Once the policies have been set up, these masks are applied in queries. You can allow certain privileged logins to see the data unmasked.

9. MAIL SERVERS

Mail servers play an important role in the information society today. As the Internet user population continues to grow exponentially, there is an increasing demand for high performance mail servers. Unfortunately, few results have been published on studying and analyzing on mail servers.

In order to understand mail server performance and its relationship with modern operating systems and com-puter architecture, we conducted a comprehensive performance evaluation of a stateoftheart mail server. We chose SPECmail2001 [17] as the standard benchmark and MDaemon 5.0.1 [2] as the ISP mail server under test. E-mail systems can be evaluated in different dimensions such as performance and scalability (in terms of users and per user data storage limits), data persistence and fault-tolerance, etc. In this paper we emphasize the performance aspect.

- In SMTP and POP sessions [1], the initial network connection setup usually is very expensive. Since e-mails are not "real-time", it is suggested that e-mails sent to the same remote mail server could be packed together into a single large mail to be sent.
- An important observation is that I/O latencies contribute up to 40–55% of the data transfer time of e-mail requests, especially when the user population is large. Such a high I/O overhead would seriously limit the scalability of ISP mail servers. We recommend to use a high performance I/O storage system optimized for such workload (small file sizes and short lifetimes). It is also a good idea to use asynchronous I/O processes or threads to replace blocking I/O routines in order to avoid the synchronous overhead in mail server designs.

10. MAIL SERVERS AND BENCH MARKS

10.1 The State-of-the-art Mail Servers

There are several different e-mail systems at present.

POP Server

It is one of the most traditional Internet mail servers that offer only store and forward mail services, such as Post Office Protocol server [10], in which messages are kept on the server only until the user first accesses and downloads them.

Server-Only Mail System

The system simply keeps all messages on the server. This architecture is adopted by most web-based mail servers such as Hotmail, Yahoo!, and other traditional enterprise mail servers like Microsoft Exchange and Lotus Mail.

Client-Side Caching Mail System

The system combines a client-side message cache with a permanent message repository on the server, such as Internet Message Access Protocol (IMAP) servers [9].In addition to the mail server services for end users, there is also another kind of mail service that stores and forwards e-mails between mail servers. Simple Mail Transfer Protocol (SMTP) [8] is a typical example

10.2 The Mail Server in Experiments

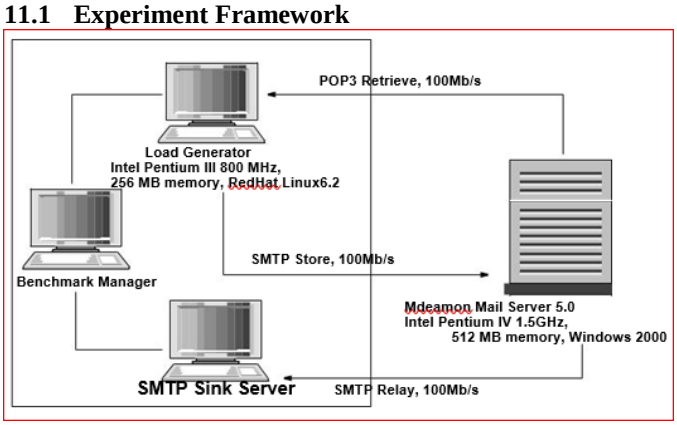
There are many different mail server systems on the market. We chose one of the most popular commercial products, MDaemon 5.0.1 from Alt-N Technologies Inc., as the mail server under test. MDaemon provides both ISP and enterprise e-mail services. It is a very representative and versatile mail server which supports POP3, SMTP and IMAP. We ran MDaemon on top of the Windows 2000 Operating Systems.

10.3 Mail Server Benchmarks

Currently there are only a few standard mail server bench-marks. In this paper we chose SPECmail2001, one of the leading benchmark software developed by SPEC or-ganization. It is a standardized mail server benchmark designed to measure a system's ability to act as a mail server servicing e-mail requests, based on the Internet standard protocols SMTP and POP3. This benchmark characterizes the throughput and response time of a mail server system under test with realistic network connections, disk storage, and client workloads. The benchmark focuses on ISP mail servers rather than the enterprise class of mail servers, with an overall user count in the range of approximately 100 to 1,000,000 users. The SPECmail2001 goal is to enable objective comparisons among different mail server systems. Because SPECmail2001 only supports POP3 and SMTP, this paper addresses the evaluation with these two protocols.

11. METHODOLOGY

11.1 Experiment Framework



We set up the experiment framework in a switched Fast Ethernet (100 Mbps) Local Area Network. Three machines from the same cluster computer systems are used in experiments. The isolated network could prevent the disturbance from the external network traffic. One Dell 4300 PC, with a Pentium IV (1.5 GHz) CPU and 512 MB SDRAM main memory running Windows 2000, was chosen as the mail server under test. The other two PCs with Pentium III (800 MHz) CPUs and 256 MB SDRAM memory functioned as the local clients and remote clients. One of them worked as the remote mail server (or called sink server) at the same time. The architecture configuration is shown in Figure 1. The MDaemon is running all the time during experiments. Two load machines run separate load generator processes to simulate the communication with the mail server. The benchmark management process, which synchronizes two load generators based on preconfigured parameters, is running on any machine in the cluster systems before experiments start. Typically, there are several operations shown as follows:

POP session (Uplink)

Clients retrieve e-mails from the mail server.

SMTP Store (Downlink)

The simulated internal users and external users send e-mails to the ISP mail server. These requests are generated by the load generators. SPECmail2001 assumes 90% of the mails are sent from users originated from external users (i.e. from remote mail servers).

SMTP Forward

The sink server receives e-mails from the MDaemon mail server. SPECmail2001 assumes that 90% of the messages sent are going to external users (i.e.are forwarded to remote mail servers). The number of users mail servers support is the best term to evaluate the scalability. The SPECmail2001 benchmark workload and transaction workloads are initialized by this number. We varied the user number from 200 to 10,000 and investigated the Network and I/O performance in every benchmark run. SPECmail2001 provides an statistical results

of network performance for study. Since Windows 2000 does not provide a good tool to measure the file system and I/O performance, we used a commercial software called Filemon/EE to collect file system traces generated the mail server. We can evaluate the I/O performance of the mail server by analyzing traces.

11.2 FILEMON/EE

Filemon/EE (Enterprise Edition) is commercial software that monitors file system activity in Windows 2000 [18]. It runs as a log process working in the background. Every file system event is recorded in a log buffer and to be written to the disk later. The filemon program has little impact on the experiment system because it works adaptively in a low-memory state. In the experiments, we carefully select the memory capacity for both the mail server and load machines because a small memory will affect the normal operations of the benchmark and the filemon. If the system runs out of the memory, the benchmark results and the collected traces will be incorrect because of the data loss or network congestion. By checking the system memory status and log records in experiments, we tuned the memory capacity to the proper values for both the mail server and load machines so that the benchmark experiments could run well.

11.4 VALIDATE EXPERIMENTS AND TRACES

In order to guarantee the correctness of experiments, we also carefully validate benchmark results and file system traces using the following methods.

11.4.1 Experiments

SPECmail2001 provides several performance metrics of quality of services (QoS) that we can adopt as the confidence of experiments. The benchmark characterizes the mail server performance by giving three data points and the degree of adherence to QoS requirements at these points. A qualified mail server (which means a correct benchmark experiment) should handle the 80% load mark, work well under the 100% load mark, but fail to satisfy the 120% load mark.

Quality of Service

There are two aspects to QoS: one is the interactive response time of the mail server at each protocol step, as directly or indirectly observed by the user, and the one is the delivery time that it takes to make a message available to a (local) recipient. The SPECmail2001 QoS standard is meant to support high-quality mail service, rather than promote barely acceptable service levels. For all requests other than SMTP-data and POP-retrieval, SPEC-mail2001 requires that 95% of all response times must be under 5 seconds.

Delivery Time

Delivery time is defined as the time it takes for a message sent via SMTP to become available to a user attempting retrieval via POP. By adding this QoS requirement, the mail server is forced to actually deliver the mail to the target mailboxes in a reasonable time, rather than just spooling it, and processing it later (e.g. after the benchmark is over). SPECmail2001 requires that 95% of all messages to local users get delivered to the target mailbox within 60 seconds.

Remote Delivery

Remote delivery is only checked for count. Namely, 95% of all messages to the remote mail server are received by the remote server within the test time frame. In this paper, only experiments that satisfy the above requirements are accepted for analysis.

11.4.2 File System Traces

NTFS is the disk storage subsystem for the mail store in this paper. We used Filemon/EE to collect file system traces. In order to collect the accurate system events in experiments, we performed pre-processing and post-processing on these traces. The mail storage partition was separated from the disk partition of file system traces. The benchmark uses a verify program to clean up mail-boxes of all users before the real benchmark runs. After we collected the traces, we also filtered out all unrelated events from Filemon/EE and Windows 2000 system. Every benchmark ran for about 5 hours. We removed additional records with timestamps after the benchmark finished. The records with errors from the failed requests, such as network connection failures or file creation errors (less than 4%), were also deleted.

12 PERFORMANCE EVALUATION AND ANALYSIS

In this section, we analyze the network and I/O performance based on benchmark results and file system traces.

12.1 Network Performance

Network is obviously an important factor in ISP mail servers. We analyze the network traffic not only between the ISP mail server and users, but also between the local mail server and remote mail servers.

12.1.1 Definitions

SPECmail2001 supports two kinds of network transaction protocols: POP and SMTP. IMAP is not currently considered in this version.

SMTP Sessions

An SMTP session consists of several phases: connecting to the server; receiving the synchronous messages; transmitting sender, recipients and contents of a message; and disconnecting from the server. There are two kinds of SMTP sessions: an SMTP session simulating a message sent by a local user and an SMTP session simulating incoming traffic from another mail server.

POP Sessions

A POP session has the following phases: establishes a connection, receives the synchronous messages, authenticates as a local user, checks a mailbox, downloads and deletes all messages found, and disconnects.

POP Retry

POP Retry is a special POP session. POP users often have set up their e-mail clients so that it performs periodic mailbox checks after a defined interval time. From the point of the mail server, this generates a load on mailboxes which have recently been checked and which in all likeliness are empty. Overall, the vast majority of POP sessions are logged as ”empty”, i.e. no messages are downloaded at all.

12.1.2 Different Number of Users

ISP mail servers perform variously when servicing different number of users. We varied the user number and collected the results in every benchmark run. Table 2 shows the details. There are two different POP sessions: those download mails from the server and those do not. The latter type is the common case: mail clients contact the server to check for new mails but do not find any. The two different POP sessions are

listed separately in Table 2. We can see that the network traffic (the number of attempts) becomes heavier when the number of users increase. The average transaction latencies also increases. Because the concurrent maximum number of threads in SMTP (or POP) outbound (or inbound) sessions in MDaemon is restricted, the mail server would most likely meet with a network congestion problem if the workload is too heavy.
SPECmail2001 only provides some breakdown results of SMTP and POP sessions, including average, minimum and maximum values. We modified a few source codes (less than 100 lines) of SPECmail2001 and collected latencies for every networking request.

12.1.3 The Breakdown of SMTP sessions

We collected the times spent on different phases of the SMTP store session. Figure 2 shows those latencies as well as their percentages of the total overhead for these phases. The "SMTP Data" phase, in which the email body text is transferred to the server, is the most time-consuming part. Our speculation is that e-mails involve a more complicated protocol (with more round trips) than the simple web services. Hence SMTP connections are kept around longer than simple data transfers. One example is that, each e-mail message may get sent to one or more recipients. SPECmail2001 configures 86% messages sent to single recipient while others sent to up to 20 recipients. The "SMTP-sender" and "SMTP-receiver" would negotiate with several recipients if the messages get rejected. These complicated operations result in more overhead. We suggest that the initial network overhead of the mail server can be reduced by using the grouping and reorganizing techniques. Since the e-mails are not "real time", we can schedule the sending sequences of different e-mail messages. Messages from the same user or sent to the same remote server can be packed up together into a "super message". When we send many small messages to another server, a single super message is sent only once rather than in many separate times. This saves many un-necessary initial overhead. We also found a few error messages like "the SMTP Sink Server is busy, try again later" under the peak load. Since all outbound SMTP e-mails are directed to the same sink server (simulating a remote server) in a short period, the sink server easily becomes busy. This phenomena is quite common in real-world e-mail services. Outbound SMTP e-mails from an ISP server will go to other mail servers rather than individuals. This group of e-mails will easily make a recipient server overloaded. Here we give one possible solution: introducing a load balancing policy for all mail servers in a wide-area network and reordering the sending sequence of these e-mails that go to the same server. For example, since the e-mail can be delayed in a reasonable time[2], we could evenly send e-mails to the different remote mail servers in a round robin rule if the outbound traffic is too heavy.

12.1.4 The Breakdown of POP sessions

Figures 3 and 4 show the breakdown results for POP sessions. For POP sessions with downloads, the "POP Retrieve" phase takes the longest time (about 30%) to transfer email text to users. "POP Delete", the next most expensive phase, takes about 20–23% of time. This is because deleting files are very expensive I/O operations on regular file systems. Other phases are relatively short. For POP sessions without downloads, the overheads of all phases are similar.

12.1.5 Prioritize the Mail Server Services

From Figure 2 and Figure 3, it is seen that the mail server will be quite busy when servicing many users simultaneously. The SMTP and POP network latencies are much longer under 10,000 user's workloads than those of fewer users. In order to keep providing high performance, the mail server could improve its QoS by prioritizing different requests. When the server has a lot of requests in the wait queue,it can process the SMTP store requests first, then SMTP relay and last POP. The new incoming e-mails are processed at first because they cannot be delayed due to the message integrity. Outbound e-mails to the remote server can be processed in next step. Since the e-mails are already stored on the mail server, POP requests can be handled a little bit later. This strategy could provide a high QoS for mail servers.

12.2 I/O Storage Subsystems

Mail servers are I/O intensive applications. Incoming e-mails result in frequent file creations. Various e-mail retrieval requests generate many file reads and file deletions.

13. CONCLUSIONS

By using the SPEC-mail2001 benchmark to test a commercial mail service– MDaemon 5.0.1, we explore the both network and I/O performance of mail servers with user populations between 200 to 10,000. File system traces are also collected for analysis. From the benchmark results, trace-driven simulations and offline trace analysis; we arrive at several important conclusions for the ISP mail servers and give the technical suggestions correspondingly. We observed that, in SMTP and POP sessions, the initial network connection setup usually takes the very long time. I/O latencies contribute up to 40–55% of the data transfer time of e-mail requests, especially when the user population is high. We also found that mail servers have better file sys-tem temporal and spatial locality than web servers. Most files are short-lived. Because of the skewed file access patterns, a reasonably-sized file system cache can capture most file accessed.

14. SUMMARY

SQL Server 2012 is a significant product release that helps customers continues to build and support mission critical environments, now with more confidence and efficiency out of the box. New tools and enhancements enable breakthrough insight across all levels of the organization while cloud-ready technologies for application symmetry across server, private and public cloud helps customers stay agile for the future. SQL Server delivers a new standard in enabling mission-critical operations — with true enterprise-class availability, performance and security features built into the solution. Integrated high-availability solutions enable faster failover and more reliable backups — and they are easier to configure, maintain and monitor, which helps organizations reduce the total cost of ownership (TCO). SQL Server also delivers mission-critical performance and scale, with predictable performance across server activities including complex queries, data integration and analysis. Because SQL Server is designed to security standards, it has minimal total surface area and database software that is inherently more secure. Enhanced security, combined with built-in, easy-to-use tools and controlled data access, helps organizations meet strict compliance policies. SQL Server supports complex data types and nontraditional data sources, and it handles them with the same attention — so organizations experience seamless support for a variety of platforms and heterogeneous environments. Finally, SQL Server delivers mission-critical capabilities at low TCO with full enterprise capabilities that

are built into the solution, not provided as costly add-ons. Ultimately, organizations can rely on a comprehensive, integrated solution that helps to contain costs and manage compliance requirements while meeting the demands of the evolving digital world.

14. REFERENCES

[1]. Jun Wang and Yiming Hu, "A performance study of email Services" Science, vol. 294, Dec. 2001, pp. 2127-2130, doi:10.1126/science.1065467.

[2]. M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[3]. Zend Technology white paper

[4]. MySQL White Paper