Terminal Server Sizing Guide

Abstract
Terminal Server as a form of server-based computing has established itself within a wide range of the IT infrastructure. This paper deals with the question as to how many users can be served with adequate performance by a PRIMERGY configured as a terminal server. The aim is to find the right PRIMERGY model for the performance level required. After a general explanation about measuring terminal servers, the simulation tool used to size the PRIMERGY systems (T4US) is presented. The components that influence the performance of a terminal server system are also discussed in detail and this is followed by a classification of older and current PRIMERGY systems with regard to their performance as terminal servers. Since sizing always includes a customer-specific aspect, a general overview of useful analysis tools and bottleneck analyses is provided. And finally, some measuring methods of other manufacturers are outlined.

Table of contents
| Introduction | 2 |
| PRIMERGY | 2 |
| Server-based Computing | 2 |
| Windows Terminal Server | 2 |
| Solutions | 3 |
| Trends | 3 |
| Sizing | 4 |
| Methods | 4 |
| The parameter “User” | 5 |
| User Simulation | 5 |
| Interpretation | 6 |
| Benchmark Description | 7 |
| Simulation tool: T4US | 7 |
| Load Profile | 7 |
| Load Profile V1 | 8 |
| Load Profile V2 | 8 |
| Measuring Method | 9 |
| Measuring Environment | 10 |
| Resource Requirements | 12 |
| Operating System | 12 |
| IA64 | 12 |
| x86 | 12 |
| Limitations | 13 |
| Effects | 14 |
| Computing Performance | 15 |
| Processor Type | 15 |
| Clock Frequency | 16 |
| Memory Connection | 17 |
| Caches | 18 |
| Hyper-Threading | 19 |
| Number of Cores | 20 |
| Main Memory | 21 |
| Disk Subsystem | 24 |
| Network | 25 |
| Infrastructure | 26 |
| Overload behavior | 28 |
| Number of Processes | 28 |
| PRIMERGY Scaling | 29 |
| Analysis Tools | 31 |
| Task Manager | 31 |
| Sysinternals Process Explorer | 31 |
| Performance Monitor | 31 |
| Windows Performance Toolkit | 32 |
| Citrix Resource Manager | 34 |
| Microsoft Operations Manager 2005 (MOM) | 35 |
| Bottleneck Analysis | 36 |
| Computing Performance/System | 37 |
| Main Memory | 38 |
| Operating System Resources | 39 |
| Disk Subsystem | 40 |
| Network | 41 |
| Terminal Server | 42 |
| Applications | 43 |
| Performance Monitor Configuration Script | 44 |
| Measurement Tools | 45 |
| Literature | 46 |
| Contact | 46 |
Introduction

PRIMERGY

Since 1995 “PRIMERGY” has been the trade name for the very successful industry standard server family from Fujitsu Technology Solutions. It is a product line that has been developed and produced by Fujitsu Technology Solutions with systems for small work groups and solutions for large-scale companies. In addition to economical single-socket entry-level systems, the PRIMERGY models also include high-performance dual-socket systems as rack and tower models right through to multi-socket systems and blade servers. Top of the performance range Intel processors are used as the heart of the systems. More detailed information about the current systems is available on the Internet site of Fujitsu Technology Solutions.

Server-based Computing

Windows Terminal Server

Terminal server is a form of server-based computing on the basis of Microsoft Windows Server operating systems.

Classic server-based computing is a system architecture in which Microsoft Windows client applications are fully installed and performed on the server. Not only their management is effected from there, but also their maintenance, administration, and support take place directly on the server. Merely the user interface, that is, display, mouse, and keyboard information, is transferred between client and server. In this way, the user can directly access Windows applications from virtually any clients, even non-Windows-based ones, through such a terminal server—without first having to transfer the respective applications to the client, to start them there or even having them held in local mass storage. If a client is solely used in this server-based scenario, it has considerably less requirements as regards memory and disk configuration than a traditional client, that is, it is therefore also referred to as a thin client.

The advantages of server-based computing are reduced costs due to better server load and improved administration as a result of centralizing the applications.

In principle, a terminal server can be used for all kinds of applications. Where until now small computers or terminals were used for simple data entry and request processes, modern applications can be integrated into an existing environment with the terminal server.

A terminal server farm is a pooling of terminal servers that have a joint administration unit, known as a “data store”. These are administered together. The allocation of users to servers and applications can be done statically or dynamically according to the terminal server load (Load Balanced Server Farm).
Solutions
Since 2000, the concept of server-based computing under the name “Terminal Services” has very much been a part of all server products of the Microsoft Windows 2000 Server and Windows Server 2003 product line. The terminal service is even available in the client operation system from Windows XP Professional to a limited extent under the name “Remote Desktop”. In this way, the remote system can be accessed from any Windows client, with the applications running fully on the remote system. The underlying protocol is called Remote Desktop Protocol (RDP). Even if the terminal services have no longer been called “Terminal Services” since the current operating system Windows Server 2008, but were renamed “Remote Desktop Services” (RDS), this paper continues – for reasons of simplicity – to use the name “Terminal Services”. The Remote Desktop Services under Windows Server 2008 have been extended to include functions that affect the administration of terminal server applications (RemoteApp), web access (Remote Desktop Web Access) to the former as well as secure access to the terminal services via the Internet (Remote Desktop Gateway).

The major player in the terminal server environment is the US-American software company Citrix. Its enhanced product “Citrix XenApp” (formerly known as “Citrix Presentation Server”, and prior to that “Citrix Metaframe”) is used to virtually provide and manage applications, especially in large-scale server environments. Citrix XenApp is based on Microsoft Windows Terminal Services. It offers enhancements and additional functions for the central management and monitoring of large server farms, as well as support in largely heterogeneous IT environments. In contrast to Microsoft Terminal Server, Citrix uses its self-developed, efficient ICA (Independent Computing Architecture) protocol to transfer data between client and server.

Trends
In addition to the concept of server-based computing, the concept of virtualization has established itself in various forms over the course of the last few years.

Similar to terminal server solutions, the task of server virtualization is to increase the load of physical servers and to reduce operating, maintenance and administration costs.

By application virtualization we mean the approach of isolating applications from their environment so that conflicts with other programs or the operating system can be avoided. The terminal server solution from Citrix, “Citrix XenApp” implements application virtualization through the two functions of Application Streaming and Application Isolation. Application Streaming enables you to bring applications to the client device and to let them run there in a protected and isolated environment. The application virtualization solution from Microsoft is located in a separate product of Windows Terminal Server “Microsoft Application Virtualization” (also known as App-V, formerly SoftGrid).

One of the greatest barriers to the introduction of a terminal server solution is the fact that the same workplace interface and the same applications are made available to all users. Complete individualization, to which the user is accustomed in a traditional workplace, is not possible. This is avoided with desktop virtualization, which is a combination of the two concepts virtualization and traditional server-based computing, and finds itself in the concept of the term first introduced by VMware “Virtual Desktop Infrastructure” (VDI).

In a virtual desktop environment the desktops of individual users are “hosted” in each case in a virtual machine of a server. Thus, every user has a separate desktop system. This virtual workplace can be accessed e.g. via protocols like RDP or ICA. Even a traditional terminal server environment has the option of making a desktop available to a user in a session, but the option of personalization, as would be possible with a traditional PC workplace, is no longer applicable. A discussion as to which concept – VDI or Terminal Server – offers more advantages as regards costs and functionality in which environment and thus establishes itself, would go beyond the scope of this paper, but remains an issue of great interest.
Sizing

Methods

Scaling, the process of adapting the system to the performance required, makes a distinction between two methods:

- **Scale-up** denotes the use of increasingly larger server systems.
- **Scale-out** denotes the use of many smaller systems that share the work.

With **scale-up**, the performance of a terminal server is increased by using high-performance hardware, that is, computing performance and main memory in particular. This is an adequate scaling method when it is a matter of serving a manageable number of users. The maximum performance of a server system or even software architecture places a limit on this scaling process. Memory limitations, which can result in the inability to fully utilize the computing performance of modern processors, arise in a 32-bit Windows operating system in particular. This is discussed in more detail in the section **Operating System**.

**Scale-out** combines a great many servers to form a group. You can distinguish between three versions:

1. **Just a Bunch of Servers**

   "Just a bunch of servers" is a loose collection of servers, here terminal servers, to which dedicated user groups or applications are assigned. However, no exchange of information and no load sharing take place between the terminal servers. The advantage of this architecture is its very simple expandability. It is a disadvantage that server performance could remain unused, because no automatic load sharing takes place. The administrative expenditure is rather high because each system has to be administered separately. This version of scale-out is nevertheless used in practice in smaller configurations.

2. **Server-Farm**

   A terminal server farm is a pooling of terminal servers that have a joint administration unit. The allocation of users to servers and applications is for the most part done statically. The advantage compared with the "just a bunch of servers" version is simplified administration of the servers. This version of scale-out is very frequently used in practice.

3. **Load-balanced Server-Farm**

   In the case of a load-balanced server farm, the individual terminal servers are pooled to form a logical unit. If a session is initiated by a client, this session is then delegated to the server currently with the minimum load by a load balancer in accordance with certain mechanisms. In addition to the better terminal server load, load balancing also has a certain redundancy. Current terminal server software solutions, both from Microsoft and Citrix, offer comprehensive load balancing options.

Below we deal with a scale-up scenario, in other words the sizing of an **individual terminal server**.
The parameter “User”

Before implementing an application server, the same question always arises: Which is the right hardware for the task set? It goes without saying in this respect that the aim is to have an optimal as possible system that is neither too small for requirements nor totally over dimensioned (for cost reasons). In other words, the question is how to find a well-dimensioned system?

The one parameter that is mostly only to hand is the number of users who are to work with the system. The most frequently asked question is: "What hardware or which PRIMERGY system is required for a terminal server that is to serve \( n \) users?". At best, you would expect a handy table as an answer from which, on account of the number of users in one column, you can read the ideal PRIMERGY system directly from the second column. Unfortunately, such a table does not exist, even if many a competitor suggests this to the customer in colorful Web pages. The answer to this seemingly simple question is actually considerably more complex because it includes one great unknown—the user.

Even if a great many perhaps wish it to be so, the user is not a standardized and predictable component, but an individual with a varying work speed and method of working. The speed of working e.g. determines the frequency with which entries are made and thus has a direct influence on the load of the terminal server system. Add to this the various tasks that lead to the different requirements made of a computer system. A user whose task consists of making queries of a warehousing system will generate a different load on a computer system than a user whose task is to design a graphic advertising brochure. The resource consumption of applications, which increases with time as a result of their further development, e.g. in 3D graphics or multi-media/video applications, should not be neglected here.

The following table shows a classification of users into groups according to their user behavior and load profile.

<table>
<thead>
<tr>
<th>Heavy</th>
<th>Knowledge Worker</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Uses several applications simultaneously.</td>
</tr>
<tr>
<td></td>
<td>Enters data at moderate speed.</td>
</tr>
<tr>
<td></td>
<td>Performs more complex operations.</td>
</tr>
<tr>
<td>Medium</td>
<td>Process Worker</td>
</tr>
<tr>
<td></td>
<td>Works intensively with only one application at a time.</td>
</tr>
<tr>
<td></td>
<td>Enters data quickly.</td>
</tr>
<tr>
<td></td>
<td>Works continuously.</td>
</tr>
<tr>
<td>Light</td>
<td>Data Entry Worker</td>
</tr>
<tr>
<td></td>
<td>Works with only one application at a time.</td>
</tr>
<tr>
<td></td>
<td>Enters only few data.</td>
</tr>
<tr>
<td></td>
<td>Has extended breaks between data entry operations.</td>
</tr>
</tbody>
</table>

As is yet to be explained in the section Load Profile, the behavior of a medium user was assumed in the measurements for this document.

User Simulation

Performance measurements do not usually use real users, but simulate the users with the aid of computers, so-called load simulators, and special software. During the measurements, one physical load generator mostly simulates a large number of logical users so that, depending on the load generator, dozens or even hundreds of users can be simulated. The following figure shows a typical simulation environment.
The controller is the master control panel managing and monitoring the simulation. The controller is connected to the load generators through a simulation network. Each load generator can simulate a large number of users. The load generators access the system under test (SUT) by means of a secondary network that also includes an infrastructure server. The infrastructure server provides the required services to the SUT, but the server itself is not measured.

The various user groups discussed above are served by the load simulators mostly according to various load profiles, also called scripts, in the terminal server environment.

For the user simulation, a distinction is made between the terms “load generator”, “client”, and “user”. In the subsequent sections, the term “load generator” refers to the hardware. A “client” is the terminal server client, one or several versions of which are running on the load generator. A simulated “user” works within a terminal server session.

**Interpretation**

Other than with various application benchmarks, where there is a benchmark or an appropriate regulation for the implementation of the application from its manufacturer or an independent body, such as with the Microsoft Exchange Server, SAP R/3, SPECweb, or TPC-C, there has to date been no standardized and accepted benchmark for terminal servers. Although various products of different manufacturers exist for user simulation, there is no standardized tool that can be used to measure and compare both Microsoft and Citrix Terminal Server.

Results of such performance measurements of various manufacturers or benchmark laboratories cannot of course be compared with each other under such conditions. Only measurements performed in the same environment and with a comparable load profile can be meaningfully compared.

Furthermore, it must be observed that the terminal server sizing measurements are evaluations in a simplified, idealized, and standardized environment with the aim of creating comparable conditions for all systems. Additional components or programs are not installed, and the terminal server is loaded up to its performance limits. In reality, additional software will have been installed, such as a virus scanner, or other add-ons of the terminal server software are run. Terminal servers in productive operation are not taken to their limit as regards performance.

Based on this premise, the following can therefore be said:

Although the “number of users per server” is our unit of measurement, results of performance measurements within the same environment should only be compared with each other. **Primarily they should also only be seen relatively, thus, for example, “system A is twice as powerful as system B” or “doubling the main memory results in an x% increase in efficiency”. Therefore, as already mentioned in section The parameter “User”, a user is difficult to quantify and our synthetic user need not correlate with a real user in all cases.**
Benchmark Description

Simulation tool: T4US

Fujitsu Technology Solutions has developed its own load simulator T4US that can simulate any user profile independent of the terminal server used and that does not have an impact on the system under test. The T4US Record tool records user input as keyboard and mouse activities in real time as well as display outputs and stores it in a T4US Script. T4US Scripts are the load profiles used during the measurement.

The T4US load simulator has three components.

**T4US Control** centrally controls and monitors the entire simulation process and evaluates measurement data during the measurement. Several instances of **T4US Playback** run on the load generator. Each T4US Playback "feeds" keyboard and mouse inputs in real time to a terminal server client on the basis of T4US Scripts recorded with T4US Record, and monitors the display content of the terminal server client. Thus, the response time of the terminal server is determined by means of high-resolution timers. A **T4US Agent** runs on every load generator. The T4US Agent is responsible for handling communication with the controller, controls and monitors the instances of T4US Playback and transfers the measured response times to the controller.

During the measurement the number of users working with Terminal Server is continuously increased. The Terminal Server response times are monitored by the T4US controller and compared with stored reference values which were determined from a previous reference measurement with only few users. If the response time of the application has deteriorated to such a degree that it no longer complies with the predefined rules, the measurement is terminated and the **number of users** is the **result of this measurement**.

Load Profile

As already explained in the section The parameter “User”, the load of a terminal server very much depends on the user. In a simulation user behavior is reflected in the load profile. The load profiles described here simulate the behavior of a “medium user”, who works intensively with one application at a time and continuously enters a large amount of data quickly.

The majority of the measurements for the “Sizing Guide” were performed with **Load Profile V1**. However, it turned out that due to improved performance on the part of the systems to be measured (new processor generations) the benchmark with this load profile reached a level, at which the measured number of users was achieved through the number of logoff / logon processes, in other words through the resulting restrictions in the operating system, and no longer through the system’s processor performance. This meant that the benchmark reached its limits without putting the processor under full load. Therefore, it was no longer possible to measure improved processor performance and thus also system performance in this way.
This is why a new Load Profile V2 has been defined. It goes without saying that the maximum number of users achieved per system with load profile V2 cannot be directly compared with the user numbers of the load profile V1. Therefore, in section PRIMERGY Scaling the measurement results are specified in relation to each other and no longer as absolute numbers.

Load Profile V1

In load profile V1 Microsoft Word is used as an application, and the user enters an illustrated text at an average rate of 230 strokes per minute.

In addition, the load profile included the following features:

- Every user works under an own user account.
- The user’s first log-in and the first start of the application are not included in the measuring interval. Nevertheless, the user logs off after the first work session at the terminal server and logs in again to start a new session. Because the individual users start one after another with a delay, individual log-ins and log-offs take place continuously over the entire duration of the simulation.
- Every terminal server client (user) starts the application from its desktop; the application is started and ended for every script run.
- Every user has his own directory that stores the pictures to be used in the text. This approach ensures that not all users load the same picture files at the same time, thus preventing the pictures from being loaded into the server file cache after a short time. Every user writes a new document with a unique name for each script run. Once created successfully, the document is stored in a user-specific directory on the hard disk of the terminal server.
- The average input speed is approximately 4 characters or cursor movements per second. However, input does not take place during the entire run because various periods of reflection of differing length have been interspersed, which is almost equivalent to natural work behavior.
- The display resolution is 1024x768 pixels; the color depth is 16 bits.
- One script run takes approximately 16 minutes, including the waiting times.

Load Profile V2

The new load profile V2 is characterized by the user to be simulated working with various Microsoft Office applications one after another. In addition to preparing a Microsoft Word document, a PowerPoint presentation is also created and calculations are performed on a newly created Excel worksheet. In comparison with the load profile V1, the number of log-ins and log-offs is reduced. On average only every sixth user logs on to and off from the terminal server on a cyclical basis. Likewise, every sixth user prints on average one Word document. Additional CPU load is achieved through the zipping and unzipping of files in the memory. The typing speed of the simulated user is between 330 and 440 strokes per minute.

The other general conditions apply as with load profile V1:

- Every user works under an own user account.
- Every terminal server client (user) starts the application from its desktop; the application is started and ended for every script run.
- Every user has its own directory that stores the pictures to be used in the text. This approach ensures that not all users load the same picture files at the same time, thus preventing the pictures from being loaded into the server file cache after a short time. Every user writes a new document with a unique name for each script run. Once created successfully, the document is stored in a user-specific directory on the hard disk of the terminal server.
- The display resolution is 1024x768 pixels; the color depth is 16 bits.
Measuring Method

In addition to a simulation tool and a load profile that is as realistic as possible, a performance measurement includes a regulatory framework that serves as a basis for carrying out and analyzing the individual measurements. The rules and regulations by which measurements are performed for this sizing guide are described below.

For the measurement with a variable number of users, the number of users working with Terminal Server is continuously increased according to a predefined rule until Terminal Server is overloaded. The overload of the server is defined by comparing measured response times and previously set reference times. All in all, 90% of the measuring values must be within a defined range, 10% freak values will be tolerated. The number of users thus determined is the result of the measurement and is called “score”.

The reference times are determined beforehand from the average response times at low load (only a few simulated users). They are largely limited by the waiting times in the scripts and only differ minimally from system to system.

The diagram illustrates the way in which the T4US controller works when analyzing the measuring values. The horizontal line at a response time of 1000 ms is the set limit that may not be exceeded by 90% of the measuring values. Some freak values are permitted, and they will be compensated by subsequent values within the permitted range. However, if too many freak values are determined, the terminal server is assumed to be overloaded and the measurement will be terminated at this point. The score in this case is “76 users”.

The diagram also shows how the measuring values would deteriorate even more if more users were added. To the right of the vertical line the response times for all users increase considerably.

Even if there are numerous articles from Citrix and Microsoft about the optimization of operating system and terminal server settings, we dispensed with these fully in our measurements. The reason for this is that many of these settings only make sense in certain environments. Used in another environment they often bring about the opposite effect. Since this series of measurements examined various PRIMERGY systems with different system components, it would not have been possible to compare the results with each other.

The only settings that are changed to subject all PRIMERGYs to the same test conditions are the following ones:

- The page file of the operating system was set to a fixed size in accordance with the main memory used to avoid fragmentation and to ensure the same conditions for all servers under test.
- With Citrix, the restriction to 100 users per server—also preset by the integrated load balancing (even if the terminal server farm consist of only one terminal server) —had to be lifted.
Measuring Environment

We studied all the current PRIMERGY models suited to be employed as typical terminal servers in the measuring environment described below:

Controller (T4US Control):
- The controller was equipped with Windows Server 2003 Standard Edition.

Load generators:
- 20 - 40 load generators (PRIMERGY dual server)
- The load simulators ran under Windows Server 2003 Standard Edition SP1. The measurements with load profile V2 were carried out with SP2.

Clients:
- The Citrix terminal server client (program neighborhood with 32-bit ICA client) was used to access the terminal server through the ICA protocol (either version 7.00.17534 of “Citrix MetaFrame XP Presentation Server Feature Release 3” or version 9.00.32649 of “Citrix Presentation Server 4.0”).
- Microsoft’s RDP client (“Remote Desktop”) enables access to a terminal server using the RDP protocol. Windows Server 2003 Standard Edition includes version 5.2.3790.1830 of the RDP client, which supports the RDP protocol V5.2. The measurements with load profile V2 were carried out with RDP protocol V6.0.6000.16459.

Network:
- The connection of the load simulators to the SUT network was established by means of a 100-Mbit Ethernet network where the terminal server was connected through a gigabit uplink. TCP/IP was used as the network protocol.

Terminal server (system under test):
- The PRIMERGY servers measured (system under test) were in each case equipped with Windows Server 2003 Enterprise Edition. The terminal services were activated in the application server mode. The measurements with load profile V2 were carried out under Windows Server 2008.
- Citrix MetaFrame Enterprise Edition including Service Pack 3 and Feature Release 3 or Citrix Presentation Server was installed for the tests measuring a Citrix Terminal Server. The terminal server farm included one terminal server. Data store was implemented as a Microsoft access database and maintained on the local system disk of the system measured.
- The users’ files to be read and written during the measurement were also maintained locally on the terminal server.
- By default, the user profiles were stored on the terminal server.
Infrastructure server:

- The infrastructure server itself was not measured but provided the required services to the system under test. The server was dimensioned sufficiently so that it would not represent a bottleneck.
- The simulated users' accounts were created on the Active Directory Domain Controller. Log-in was always effected to the Active Directory.
- The Active Directory system was simultaneously used as a DNS server and as a Terminal Server Licensing Service.
Resource Requirements

Below we want to discuss and, in the light of the measurement results, substantiate which components have which influence on the performance of a terminal server system. In addition to aspects of the operating system, performance-relevant factors of a server system, such as computing performance, main memory, disk subsystem and network, are discussed here. Then the environment, in which a terminal server is run, is looked at and reference is made to infrastructure components that influence performance behavior. And finally, the behavior of a terminal server in monitored overload situations is explained.

Operating System

Microsoft Windows Server operating systems are available up to Windows Server 2008 in two versions, the 32-bit and the 64-bit version. Even if future operating systems – Windows Server 2008 R2 and higher – are only available in the 64-bit version, the differences and therefore the possible losses in performance with terminal server are also to be discussed here.

Prerequisite for a 64-bit operating system is a 64-bit hardware platform. Here we currently distinguish between Intel Itanium (IA64) and x64.

IA64

Processors of the IA64 platform are not code-compatible with 32-bit applications (x86). The consequence of this is that x86 applications are only emulated on Itanium systems, but which means an appropriate loss in performance through the emulation layers. Therefore, running a terminal server on IA64 architecture does not make sense, because all the 32-bit applications (e.g. Microsoft Office 2007) would as a result of the emulation run very slowly. At present this is why there are no Citrix terminal server solutions for Itanium-based systems and also why the terminal server feature cannot be configured under Windows Server 2008.

x64

Architectures that are fully compatible with the x86 architecture and only offer extensions for 64-bit are indicated by x64. These include the AMD Opteron with AMD64 architecture and the Intel Pentium and Xeon processors of the latest generation with EM64T (now Intel 64) architecture.

The advantage of the x64 platform is that 32-bit applications, such as Microsoft Office, can be run directly under the x64 operating system. The 64-bit Windows provides the WoW64 ("Windows on Windows64") subsystem to enable smooth operation of the 32-bit applications. Memory, registry and file system accesses are isolated in WoW64. However, it is imperative for many system-related applications and all applications that contain driver components to be specially adapted for x64 systems, because they cannot run in 32-bit mode. Microsoft SQL Server or Microsoft Exchange Server and the terminal server products from Citrix are examples of applications that have to be specially adapted for 64-bit. In general, 16-bit applications for DOS or Windows can no longer be run; this is particularly a problem for applications with a 16-bit installer.

The table below gives you an overview of the relevant system platforms, the Windows products and the requirements they make of device drivers and applications.

<table>
<thead>
<tr>
<th>Server platform</th>
<th>x86</th>
<th>x64</th>
<th>x64</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating system</td>
<td>32-bit</td>
<td>32-bit</td>
<td>64-bit</td>
</tr>
<tr>
<td>Device drivers</td>
<td>32-bit</td>
<td>32-bit</td>
<td>64-bit</td>
</tr>
<tr>
<td>Applications</td>
<td>32-bit</td>
<td>32-bit</td>
<td>32-bit and 64-bit</td>
</tr>
</tbody>
</table>
Limitations

The basic difference between 32-bit and 64-bit architecture lies in the size of addressable memory. If you use 32-bit architecture, it is possible through the word length of the processor to address a maximum of $2^{32}$ bytes = 4 GB, and up to $2^{64}$ bytes = 16 exabytes (EB) with 64-bit architecture. The table below gives you an overview of the limitations of the various 32-bit and 64-bit Windows products that make sense to be used for Terminal Server:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Standard Edition</td>
<td>Enterprise Edition</td>
</tr>
<tr>
<td>Number of processors</td>
<td>1 – 4</td>
<td>1 – 8</td>
</tr>
<tr>
<td>Max. RAM</td>
<td>4 GB</td>
<td>64 GB (with PAE)</td>
</tr>
<tr>
<td></td>
<td>32 GB</td>
<td>2 TB</td>
</tr>
<tr>
<td>Total Virtual Address Space</td>
<td>4 GB</td>
<td>16 TB</td>
</tr>
<tr>
<td>Virtual Address Space per 32-bit process</td>
<td>2 GB</td>
<td>2 GB</td>
</tr>
<tr>
<td></td>
<td>3GB if booted with /3GB switch</td>
<td>4GB if compiled with /LARGEADDRESSAWARE</td>
</tr>
<tr>
<td>Virtual Address Space per 64-bit process</td>
<td>-</td>
<td>8 TB</td>
</tr>
<tr>
<td>Paged Pool</td>
<td>470 MB / WS 2008 dynamic</td>
<td>128 GB</td>
</tr>
<tr>
<td>Non-Paged Pool</td>
<td>256 MB / WS 2008 dynamic</td>
<td>128 GB</td>
</tr>
<tr>
<td>System Page Table Entries</td>
<td>ca. 900 MB / WS 2008 dynamic</td>
<td>128 GB</td>
</tr>
<tr>
<td>System Cache</td>
<td>1 GB / WS 2008 dynamic</td>
<td>1 TB</td>
</tr>
</tbody>
</table>

In the 32-bit Windows the 4-GB virtual address space is as standard split into 2 GB for the operating system and 2 GB for the applications. The 2 GB used by the operating system contains all the data structures and information of the kernel. Here three special data areas are worth mentioning: The paged pool area, the system page table entries (PTE) area, and the system file cache. Paged pool area memory is requested by kernel-mode components, whereas the PTE area is used for kernel stack allocations. System file cache is used to store memory images of opened files. These areas share a memory area and under Windows Server 2003 the memory limit between them is defined during system start. If the limit of the operating system's memory is reached in one area during a run, it cannot be compensated by the other area. The consequence is that no more new users can be logged on or unexpected errors occur. In 32-bit Windows Server 2008 the kernel address space is administered dynamically and the memory areas for paged pool, system cache, etc. can grow or shrink according to the workload. However, the areas are limited through the currently available virtual address space of the kernel.

PAE (Physical Address Extension) is a technical extension for x86-compatible CPUs (from Intel Pentium and AMD Athlon), which enables memory of up to $2^{36}$ bytes = 64 GB to be addressed. This is possible because these processors have a 36-bit wide address bus. Special extensions in the “paging” unit of the processor ensure that 36-bit physical addresses are generated. To enable PAE to be used support must be provided by the operating system.

Of the theoretical value of 16 EB of addressable memory 64-bit Windows uses 16 TB as virtual address space. Windows divides this gigantic address space (at least from today’s point of view) into various areas so that an address space of 8 TB is available for the kernel and for each 64-bit process. For 32-bit applications running in the so-called compatibility mode, an addressing space of 4 GB is available for each application. This still exceeds the addressing space of an application under a pure 32-bit operating system in which it is only 3 GB.
Effects

If adequate hardware resources are available, i.e. neither the CPU nor the main memory is the cause for the limitations, then the number of users that can be managed on a 64-bit system is considerably higher than on a 32-bit system. The reason for this is to be found in the operating system architecture, or to put it more precisely, in the kernel resources. The 32-bit operating system has an address space of 2 GB available to store its data, including kernel tables and system cache. With an adequate computing performance the number of users reaches a magnitude that is too high for a 32-bit operating system. The kernel tables are fully occupied, which in turn results in paging activities, although main memory is still free. Moreover, the system cache is also overloaded and cannot be enlarged any further, which results in a poor hit rate in the cache. Consequently, the disk accesses also increase dramatically. As soon as accesses are made to the hard disk instead of the main memory, this becomes immediately noticeable in a deterioration in the response times. If a terminal server would be further overloaded beyond this bottleneck situation, the applications would encounter errors or users would no longer be able to login at all with the terminal server.

The scenario in which a 32-bit operating system or a 64-bit operating system offers advantages can be best explained on the basis of a series of measurements. The same PRIMERGY system with adequate computing performance in each case was equipped with 4, 8, 16 and 32 GB RAM and the maximum number of light users was determined. In the measurements with 4 GB main memory the limiting factor - both under the 32-bit operating system and the 64-bit operating system - is the main memory, which results in paging activities. The measurements under 64-bit even give a slightly worse result. This can be explained by the fact that 64-bit applications usually need more main memory than the 32-bit versions, because all address pointers are twice as wide with 64-bit (see section Main Memory).

If the memory is enlarged to 8GB, more users can be handled and this memory configuration is the optimal configuration for the 32-bit system. A further memory upgrade to 16 GB and 32 GB respectively is not advantageous with these measurements under the 32-bit operating system. As already explained above, the limited kernel resources result in the support of no further users.

In contrast, an increasing number of users is achieved through a larger memory configuration with the 64-bit operating system, because no limitations occur as a result of operating system resources.

In conclusion it can therefore be seen that a 32-bit operating system has a limiting affect with terminal server if either the main memory required for the necessary number of users (>64 GB) is not supported, or if internal kernel bottlenecks occur on account of the many users. Here the memory usage of an individual user depends of course on the application. If these two limitations do not exist, a 32-bit operating system can as a result of lower memory usage score points over the 64-bit operating system.
Computing Performance

The computing performance of a system depends on the processor features and the number of processors. By processor features we mean parameters such as clock frequency, cache, Hyper-Threading, memory connection or number of cores. Here processor features must always be seen with regard to the respective process architecture.

Processor Type

The following overview shows processors of various generations that are in use in older and current PRIMERGY systems for terminal server. The performance data are valid in each case for one CPU, as determined with the terminal server simulation tool “T4US” with the load profile V1 and V2 respectively. All the values are shown standardized to the results of the best Xeon X5570 CPU.

As far as computing performance is concerned, all systems are equipped with sufficient main memory. Hyper-Threading, if available, was enabled, because for terminal server applications, this feature results in a reduced load on the system and a larger number of users can work with Terminal Server. It is not possible to draw conclusions about the performance of a system with a higher number of CPUs because the performance gain is not linear (see section Number of Cores).
Clock Frequency

The processor lines of PRIMERGY servers are available in various rating classes, which differ among other things in clock frequency. Clock frequency in itself says nothing about processor performance. Other factors, such as architecture, instruction set, caches, memory connection, also influence processor performance. You should therefore not make the mistake of always connecting greater performance with higher clock frequency. This becomes evident in the Intel Xeon 55xx processors, which in comparison with the predecessor, Xeon 54xx, have a lower clock frequency, but which prove to be much more powerful. The advantage of the lower clock frequency lies in lower electrical power consumption.

As a result of introducing Turbo Boost technology in the Intel Xeon 55xx series the key indicator of frequency is also no longer so clear-cut. With this technology and depending on the application, the processor is namely automatically overclocked when it works below certain thermal and electrical thresholds. Put simply this means that if the processor is not working to capacity and the Thermal Design Power (TDP) is not exceeded, the processor is put to higher frequency.

If you compare processors of a family that only differ in clock frequency whereas the other parameters, such as architecture, caches, memory connection and number of cores, are identical, greater performance can also be demonstrated with a higher clock frequency. However, the increase in frequency is not reflected on a 1:1 basis in the relative increase in performance. This is explained by the constant speed for memory and I/O accesses.

The results of the PRIMERGY RX300 S4 1-CPU measurements of the Xeon 54xx series are shown as examples in the diagram opposite. The results were achieved with the terminal server benchmark and the load profile V1 under Microsoft Terminal Server and are presented standardized to the smallest processor. All processors have a front-side bus with 1333 MHz and a 2 × 6 MB L2-Cache.

(Load profile V1, Microsoft Office 2003, Microsoft Terminal Services)
Memory Connection

Memory connection in the PRIMERGY systems differs depending on the processor architecture.

In PRIMERGY systems that are equipped with AMD Opteron processors the memory controller is integrated in the CPU and the processors are connected to each other by a HyperTransport link.

With Intel processors up to the Xeon 5400 series communication between the processor and the so-called Northbridge is handled via the front-side bus (FSB). The Northbridge is an integral part of the chipset, which is in turn connected with other components, such as the main memory or the PCI bus. The FSB is operated with a specific clock rate, which influences system performance. However, since the processors are generally not only different in the clock frequency of the FSB, a direct comparison of various FSBs is hardly possible.

Under laboratory conditions, however, it was possible to measure the influence of the FSB rate. The diagram shows the improvement in performance through increasing the FSB clock frequency from 667 MHz to 800 MHz, which corresponds to an increase of about 20%. The number of Terminal Server users increases in all three configurations measured. A faster FSB relieves the load on the processor, the “% Processor Time” is somewhat lower at 800 MHz FSB and similarly, so is the “Processor Queue Length”. As a result of this CPU reserve more users can work with the Terminal Server before the response times deteriorate and are no longer within the bounds of what is permissible. The increase in the number of users, however, is smaller than the increase in the FSB clock frequency. Terminal Server as an application not only makes demands on processor performance, but the interrelation of all resources, such as processor, memory, network and disk, makes a contribution toward the overall performance.

From the PRIMERGY generation with processors of the Xeon 5500 (Nehalem EP) series a change was made in the system architecture, because FSB technology has with regard to its complexity, e.g. the number of pins needed in the chipset per FSB, reached its limits: Intel Quick Path Interconnect (QPI) instead of Front-Side Bus (FSB). QPI connects processors to each other, as well as processors and the chipset responsible for I/O via unidirectional, serial links. In order to link the main memory the processors of the Xeon 5500 series are equipped with memory controllers. In other words, every processor directly controls a group of assigned memory modules. The performance of Quick Path Interconnect is specified in giga transfers per second. The current processors of the Xeon 5500 series achieved with a QPI of up to 6.4 GT/s (is equivalent to up to 25.6 GB/s) a much higher bandwidth than the most powerful processor of the Xeon 5400 series could provide with the aid of front-side-bus architecture.
Caches

Generally speaking, a cache is a faster temporary memory accelerating data access by buffering data. Caches of Intel CPUs are cascaded on several levels. A difference is made between level 1 caches, level 2 caches (also called second-level caches—SLCs), and level 3 caches (third-level caches—TLCs). In most cases, only the cache of the last level is mentioned when talking about performance data of CPUs. The cache has been designed to save the processor from having to wait for data from the slower main memory. The greater the cache, the less memory accesses are necessary. This saving in time results in turn in a greater computing performance.

Comparative measurements with the terminal server benchmark showed that doubling the cache resulted in a decrease in the CPU load. Two processors with a 1-MB cache and 2-MB cache, respectively, were used under otherwise identical conditions. As shown in the diagram, the load on the terminal server system processor is lower when a larger cache is used. As a benchmark result this resulted in a higher number of users.

Since more data have to be processed in the 64-bit operating system on account of the twice as large address width, the size of the CPU cache has more influence here.

The same PRIMERGY system was equipped alternatively with Xeon processors with a 1-MB or a 2-MB SLC. The diagram shows that doubling the cache results in a higher performance for both platforms. The 64-bit system profits most from the double size cache; the performance increase is 25%. The 32-bit system gains up to 15% more performance with the bigger cache. These measurement results show that the 64-bit system needs a larger cache because of its address overhead.

A general recommendation can be deduced from these results: Processor versions with a large cache should by all means be used for systems with a 64-bit operating system. This is more important than a slightly higher clock frequency.

However, you should not make the mistake of comparing cache sizes of processors with a different architecture. The caches of the processors are attuned to the processor architecture. Thus, for example, the processors of the Xeon 5500 (Nehalem) series with 256 kB per core SLC and at most 8 MB TLC for all four cores nominally have less cache at present than processors of the Xeon 5400 (Harpertown) series with 2 × 6 MB SLC. Nevertheless, the overall processor performance is far higher.
Hyper-Threading

With Hyper-Threading processors, some resources on the chip are doubled so that the CPUs are now capable of executing two threads in parallel. Thus, two virtual or logical CPUs are simulated. To the operating system, a CPU with Hyper-Threading appears to be two CPUs and will be controlled as such. This increases speed if the operating systems and applications are suited for higher speeds. Windows is designed as an operating system capable of Hyper-Threading, and particularly within terminal server environments, numerous individual users simultaneously work with quite a large number of mostly small applications so that Hyper-Threading can be expected to lead to a performance increase.

Most Intel processors of the current Xeon 5500 series (Nehalem) support Hyper-Threading. As a basic principle, AMD processors do not have Hyper-Threading.

Measuring has shown that the increase in performance due to Hyper-Threading is the largest for systems with low to medium load. The performance gain is lower for systems working at their load limit. Moreover, the performance gain on a monoprocessor system is higher than that on a multiprocessor system.

A measurement sequence was run on a PRIMERGY RX200 S1 with two processors; one set of simulations each was carried out with enabled and disabled Hyper-Threading. The same number of users (101) was simulated in both cases. For terminal server applications, Hyper-Threading resulted in a reduced load on the system. The CPU load could be lowered by 31% as shown in the diagram. When the test was run on the PRIMERGY RX200 S1 and 101 users without Hyper-Threading, a CPU bottleneck was indeed detectable. The response times of Terminal Server were no longer within the specified time slice. However, the system complied with the specified response times again when the relief of the CPU load was achieved by Hyper-Threading.

Another series of measurements determined the absolute number of users for systems with and without Hyper-Threading. The diagram shows the measuring results. The PRIMERGY system used for the test was a PRIMERGY RX300 S2 with one or two processors with different clock frequencies. For this comparison, all processors were equipped with a 1-MB SLC. Hyper-Threading was alternatively enabled (“HT on”) or disabled (“HT off”). It can be seen that a larger number of users can work with Terminal Server when Hyper-Threading was enabled. The performance gain due to Hyper-Threading for a slower monoprocessor system is much higher than for a fast dual system.
Number of Cores

A common method used to increase the performance of a processor is through increasing the clock frequency. However, for frequencies of about 4 GHz this is accompanied by waste heat that can no longer be sensibly managed. One opportunity of further development here is to use multi-core processors. In contrast to “Hyper-Threading”, where only parts of a processor are redundant on the chip to be able to perform threads in parallel, on a multi-core processor all processor resources are replicated. Multi-core processors also have the advantage that the costs for the use of a single chip with several resources are frequently less than with several individual chips. Whereas single-core processors dominated until 2005, the current processors of the Intel Xeon series are designed as 4-fold or even 6-fold cores. Future processor generations will see the number of cores per chip multiply still further.

The purely theoretical increase in performance is at most 100% (compared with an individual core) for every additional core. In practice, however, the increase in performance depends greatly on the degree of parallelization of the performed program and the operating system in use.

On top of that, the computing performance of a system can be increased by using several processors. The use of several identical physical processors is also called “symmetric multiprocessing” (SMP). Thus, scaling with an increasing number of processors is only linear in the ideal case of an application that can be parallelized optimally. However, the more accesses made to shared resources, such as main memory, hard disks or network - thus necessitating coordination between the processors - the more the scaling curve levels out. In an extreme case, with a very large number of processors and with a large proportion of the processors coordinated to each other it is even possible for “reverse state” of the scaling to occur. As early as 1967 Gene Amdahl observed the acceleration of programs through parallel execution in a mathematical model and found that the increase in speed is above all limited by the sequential part of the program (“Amdahl’s law”).

Current multiprocessor systems counter this by providing the processors with large caches or by forming groups of processors and assigning separate storage and I/O components to the latter, which is suitable for operating systems that have been adapted for optimal performance, such as Windows Server 2003 Enterprise and Datacenter Edition with “non-uniform memory access” (NUMA) support.

As already explained in section Hyper-Threading numerous individual users work in terminal server environments with a great many, mostly smaller applications in parallel, which is a good prerequisite for using many cores. However, the number of cores to which a terminal server application is well scaled is also specific to the application and cannot be generalized.

As depicted in the diagram opposite, terminal server measurements with the T4US load profile V2 on a PRIMERGY RX300 S5 with one and two Xeon Quad-Core processors and with enabled Hyper-Threading showed very good scaling of factor 1.7.

(Load profile V2, Microsoft Office 2003, Microsoft Terminal Services)
Main Memory

The main memory has the greatest influence on the performance of the terminal server. This is particularly reflected in the response time. As and when required, Windows acquires further virtual memory by relocating (paging) data currently not needed from the main memory (RAM) to the page file on the hard disk. However, since disk accesses are about a thousand times slower than memory accesses, this results directly in a breakdown in performance and a rapid increase in response times.

With terminal server, the memory requirements increase in proportion with the number of users, independent of the PRIMERGY model and likewise for 32-bit and 64-bit operating systems, as both diagrams illustrate.

When the occupied memory, the “committed” memory and the “Working Set” is shown as a graph, a linear development can be observed that rises with the increasing number of users. While “Available MBytes” indicates the free physical main memory currently available, “Committed Bytes” specifies how much virtual main memory was assigned to the running applications. The “Working Set” of an application is the memory already occupied by this application.

It should be mentioned, however, that 64-bit operating systems and 64-bit applications generally require more main memory than the 32-bit versions because all the address pointers of 64-bit systems are twice as wide. This can in extreme cases mean that the memory required by 64-bit is twice as large when compared with 32-bit.

As shown in the diagram, the same user who started the desktop and is working with Microsoft Word 2003, uses approximately 60% more main memory compared with the 32-bit system. In both cases, the application run by the terminal server user is Microsoft Word, which at present only exists as a 32-bit version.

However, the fact that memory requirements grow linear according to the following formula is universally valid:

\[ \text{Memory} = \text{Memory}_{CS} + \#\text{Cnct} \cdot \text{Memory}_{App} \]

In this connection, note that the memory requirements per user greatly depend upon the application used and must in each case be determined on a customer-specific basis. If the memory requirements are known for an application for a user, it is easy to calculate the overall memory requirements.
Only the maximum possible memory configuration of the system and the support of the operating system have the effect of an upper limit (see section Limitations).

The maximum memory configuration of a system depends on the model.

The x86-based PRIMERGY servers have different architectures. The older models (from Intel Pentium Pro processor (1995) to Intel Xeon 5400 (Harpertown)) comply with the principle of symmetric multiprocessing (SMP). The connection of the processors to the main memory is achieved by means of a front-side bus. Even in the configuration with only one processor the entire memory can be used.

Intel Xeon processors from the Xeon 5500 series are subject to a different architecture. These processors are equipped with memory controllers for the connection to the main memory - in other words, every processor directly controls a group of memory modules that has been assigned to it. The processor can simultaneously provide memory contents to the neighboring processor via the QPI link and request such contents itself (see section Memory Connection). As a result of the direct connection between the processor and memory an increase in memory performance is plausible, but with the difference in performance between a local and a remote request, which justifies the classification of this architecture as NUMA. NUMA is taken into account by the operating system when allocating the physical memory and when scheduling processes. If possible, the total amount of main memory should be symmetrically distributed over both processors. If the system only has one processor, only the memory banks assigned to this processor can be used.

More performance-relevant information about efficient memory configurations of the Xeon 5500 based PRIMERGY systems is available in the document “Memory Performance of Xeon 5500 (Nehalem EP) based PRIMERGY servers” [L4].

PRIMERGY models with AMD Opteron processors also have direct assignment of the memory to the processors. Every processor is directly connected to “its” memory, which results in fast access. Conversely, it can be deduced from this system architecture that a PRIMERGY, which is equipped with only one AMD Opteron processor, can also only use half the memory banks because the second CPU-internal memory controller is missing to access the memory.

When calculating the main memory for the terminal server, two particular features should not be disregarded:

- “Desktop” or “Published Application”?

  In terminal server environments it is not necessary to make the entire desktop with all its applications available to the user, that is, access to it can be limited. It is possible to set up Microsoft Terminal Server in such a way that only one particular application is started instead of the entire desktop.

  With Citrix Presentation Server, it is possible to make individual applications directly available to the users (“Published Application”); starting the application on the desktop is no longer applicable. In this respect, a savings of approximately 5 to 10 MB of main memory is achieved per user, since the Explorer process does not also have to be started. Even if it is perhaps not a matter of winning this small amount of main memory in a certain environment; one advantage of this configuration that cannot be neglected is the fact that the user can only let the applications planned for him run on the server. In other words, the user’s actions can be better forecast and also be limited.

  The memory required by a user starting an application from the desktop was compared with the memory required by a user starting the same application without the desktop. The Microsoft Word application from Microsoft Office 2003 was started once through the desktop and once directly through the RDP client. As can be clearly seen in the diagram opposite, the increase in memory used is less without starting the desktop.

(Load profile V1, Microsoft Office 2003, Microsoft Terminal Services)
“Logoff” or “Disconnect”?  
It is a difference whether the user ends the connection to the terminal server by logging off or whether he only interrupts the connection with a “Disconnect”. In the latter case, the application continues to run on the terminal server and does not release its resources. The user can continue his work at this juncture. When the used main memory of the server is analyzed, the connections of the “Disconnect” status also count. Some applications still require CPU resources even after the connection has been disconnected. “Disconnected” sessions are support by both Microsoft as well as Citrix.
Disk Subsystem

If we assume that a server is used as a dedicated terminal server and not as a file server or database server at the same time, then no great demands will be made on the disk subsystem. In the main, it only has to accommodate the operating system, the paging area, and the applications available to the terminal clients. Disk accesses in this connection are low. The operating system and the applications are only accessed when they are loaded into the memory for the first time. In principle, the paging area does not play a role either because the system has to be configured in such a way that it does not begin paging heavily. Otherwise, the efficiency of the system is by all means substantially impaired.

In a customer configuration, the user data and user profiles are typically placed on appropriate disk subsystems or external file servers and not on the local hard disks of a terminal server.

If a terminal server is configured with modern server hardware with multi-core architecture and, where applicable, a 64-bit operating system, more users can work with it and thus the disk subsystem must also grow to be able to meet these requirements. As a result of more user sessions, more accesses to the local page file and to the disk subsystem take place and with the 64-bit operating system even larger amounts of data are also written in the direction of the page file. For these reasons the disk subsystem must be sized accordingly. Information about monitoring the disk subsystem is available in section Bottleneck Analysis.

To achieve maximum throughput, all available controller and disk caches (including the write caches) should be activated. Hard-disk write caches make a considerable contribution toward increasing performance and it is recommended - also in productive use - to make use of this functionality, which is available on all hard disks. In this regard, it is advisable to use a UPS to protect against power failures and the data loss that these entail.


If the terminal server system is also simultaneously used as a file or database server, it goes without saying that additional criteria, as are typical for file or database servers, are then applicable for the disk subsystem. Such a constellation, however, is not advisable, unless the system is only used in a very restricted workgroup environment. Otherwise, dedicated systems should be installed for the individual tasks, such as terminal server, file server, database server, or application server. A server system can only be optimally tailored for its task area in this way.

Depending on the number of hard disks in the server system, the following recommendations apply:

Two hard disks, configuration for security: If only two hard disks are available and security is an issue, mirroring should then be set up for security reasons through RAID 1. The operating system, paging area, and applications are then to be found there. Mirroring can either be implemented by means of the onboard RAID 1 functionality offered by almost all PRIMERGY servers as standard or as a software solution with Windows features. Alternatively, it is also possible to use a RAID controller.

Two hard disks, configuration for performance: For a better performance, the page file should not be configured on a mirrored disk. If only two hard disks are available, the operating system and the applications should be hosted on the first disk while the page file should be stored on the second disk. Because the disk of the operating system is not mirrored, be sure to place no user data on this disk and take backups at regular intervals.

Three or more hard disks: If a total of at least three hard disks is available, two hard disks should be mirrored by using RAID 1 and the operating system, and applications should be accommodated there. The paging area is placed on the third dedicated hard disk because configuring the page file on a mirrored disk may have substantial performance implications. Since these data are all of a temporary nature, a safeguard by means of RAID is of course unnecessary.
Network

Although the topic network has an important role to play in the terminal server environment, it is not dealt with in depth in this paper, as it is a topic in its own right. An individual terminal server and not a terminal server farm should also be the focus of attention here. Besides, network topologies, into which the terminal server is to be integrated, frequently exist in practice.

When considering an individual terminal server you should take into account that with regard to the network connection it provides the necessary bandwidths for all users. Which bandwidths are needed for the terminal server depends very much on the user. Users, who e.g. do more text processing in an Office environment, need less bandwidth than users of 2D or 3D applications. And the transfer protocol used also plays a role. Here the ICA protocol developed by Citrix has proved to be more efficient than RDP from Microsoft.

Current PRIMERGY servers with their Gigabit Ethernet connection provide a good prerequisite for operating a terminal server at high performance levels.

If the task scenario calls for the applications that run on the terminal server to access large quantities of data, databases, or even host applications, it is then advisable to equip the terminal server with a further network card for dedicated access to these server services so as to separate the data transfer in this three-tier environment between server/server and client/server communications.

In a terminal server environment the following network traffic must be taken into account:

- **Terminal Server and Active Directory**
  Information from the Active Directory was mainly required when individual users logged on to the domain. In real configurations, this kind of traffic is frequently handled through a network segment that is isolated from the client networks.

- **Clients and Terminal Server**
  Keyboard and mouse activities were transferred to the terminal server, and changes in the display contents were sent to the client.
Infrastructure

In our discussions, we have always studied the terminal server as an isolated entity. The simulation environment included other components cooperating with the terminal server. These components were, however, always constant and designed to ensure that they could not represent a bottleneck. In reality, however, this is not always the case. In this section, we will discuss which other components of the infrastructure influence user perceptions within a terminal server environment, thus possibly reflecting a negative overall impression.

Clients

In addition to the server resources and the network, the thin client also forms a part of the overall environment of server-based computing. In other words, as far as the client is concerned, the question as to the influence of its performance on the overall configuration is also justified. Since the actual application runs on the server in classic server-based computing, the CPU performance of the client is only needed for the operation of the network and for image processing, which with regard to its demand on computing power cannot be ignored. It was found that the entire time required to execute an application may, depending on the client system used, altogether vary. Under realistic operating conditions, however, these differences should at best be reflected in a subjective performance impression of the user and have no direct influence on the performance of the terminal server system.

The “thinness” of the thin client depends primarily on the demands the application used makes of the graphics, such as resolution, color depth, and complexity (text, graphics) as well as, if need be, on additional requirements, which exceed pure server-based computing and are made of applications running locally on the server.

In addition to the stationary (thin) clients, there is demand for mobile devices, such as notebooks or PDAs, to be connected to the terminal server. Usually, these devices are then no longer connected to cable-bound networks but through radio LANs (W-LAN) or mobile radio networks (such as general packet radio services—GPRS or universal mobile telecommunication system—UMTS).

This does not represent a problem for the resource-saving ICA protocol. The ICA client also exists for many current PDAs and the ICA protocol is as a result of its design also particularly suited for slow connections. For users who work with terminal server applications permanently and exclusively, such a device is undoubtedly not the ideal client, but someone who has to work when mobile and only occasionally needs to access a terminal server profits from the flexibility and functionality of this solution.

Active Directory

Under normal circumstances, terminal server users authenticate in a domain, that is, the terminal server verifies the user credentials entered against the Active Directory. Apart from very small workgroup environments, Active Directory and Terminal Server should always run on different systems, and no users should be managed on the terminal server system itself. Although Active Directory must fulfill the same requirements as in an environment without a terminal server, neither Active Directory nor the network between Active Directory and the terminal server should represent the bottleneck.

User Profiles

A user profile serves to store the individual user settings. Even when terminal server users log on to a domain, in an Active Directory environment their user profiles would be stored on the terminal server by default. With a load-balanced terminal server farm in particular, the user profiles would have to be stored centrally on a server within the network to ensure that the user will always find the same settings, independent of which terminal server his session was run on. This functionality has already been made available for so-called “roaming users” who log on to different workstations. When using terminal servers, it should be taken into account that different user profiles may have to be managed. This is the case when, for example, the local operating system of the workstation differs from the operating system on the terminal server and/or when different applications are available. For this reason, a terminal server user profile can be set up in addition to the user profile to be loaded locally. The mandatory user profile is a specific variant of the server-based user profile, and it cannot be modified by the user. Server-based user profiles should in general be kept as small as possible.
DNS

DNS is also used in the terminal server environment and serves to implement the name resolution of connections. In a load-balancing environment in particular this method is used to link a virtual name with a virtual or real IP address so that a terminal server farm presents itself as a single server to the user. This in turn results in the request that DNS must always be available to enable a user to establish a connection to the terminal server. Although DNS in general does not represent a bottleneck, this service should be designed in a fail-safe and redundant manner.

Terminal Services Licensing Server

When a user logs on to Terminal Server, the server will try to find a licensing service server and request a valid license for the access through Terminal Server from the licensing server. In major configurations, this licensing server will be a separate system.

Back-End Server

Particularly in load-balanced terminal server farms, user files will not be stored on the local hard disks of the terminal server systems but rather on file servers or NAS systems. It can be assumed that in larger environments other services such as e-mail, databases, and so on will also be required, which means that servers for applications such as Exchange, SQL, or SAP R/3 will be found together with the terminal server. While only a small network bandwidth is required for the connection of clients to terminal server, this does not apply to the connection of the terminal server with the back-end server. For the latter, sufficient network and processor capacity should be made available. For information on the individual server types, refer to separate performance studies and sizing guides, as these aspects would go far beyond the framework of this paper. It is not recommended to host back-end services on a Terminal Server system.
Overload behavior

For all PRIMERGY servers, but in particular for the more powerful servers, it is possible to observe a behavior where a system with a seemingly normal load becomes overloaded when some more users are added. When the number of users, the CPU load of the system and the response times are put into relation to each other, you can see how the response times of the terminal server behave with increasing usage. The number of users was continuously increased during the initial 15 minutes of a measurement lasting 25 minutes. Each user first logged in, then worked with Microsoft Word, and finally logged off after approximately 16 minutes to start all over again.

This means that the user logons (blue graph 1) were spread over a relatively small time slice, a behavior that is very close to reality if all users start their work at roughly the same time. The CPU usage of the terminal server showed a steady increase (red graph 2) up to almost 100%. Two significant measuring results could be observed: the time a user needed to log on to the terminal server (magenta graph 3) and the time required to add a picture under Microsoft Word (green graph 4). The logon to Terminal Server not only includes this process itself, but a user's desktop must also be started. This represents a higher load for Terminal Server than inserting a picture under Microsoft Word. Actions that by their nature represent a higher load for Terminal Server are slowed down to a greater extent under high-load conditions than with less demanding activities. Three phases can be observed in the terminal server usage:

- The CPU load of Terminal Server is below 70%. The response times of the server are extended only to a small degree, which will not become apparent to the user.
- The CPU load of Terminal Server is between 70% and 90%. Actions placing a higher load on the server are slowed down, but the response times are still within tolerable limits for the user. Actions placing a lesser load on the server show, on average, the same response times but fluctuations in the response times are possible.
- When the CPU load of Terminal Server exceeds 90%, the server is clearly overloaded. Depending on user actions, the server responds with a considerable delay, in particular actions such as a logon require more time. But even more simple actions are slowed down considerably. The user will no longer tolerate the response time behavior of the server.

Number of Processes

Even if it sounds like a paradox—it is possible to have constellations in which there can be performance bottlenecks although sufficient CPU and memory resources are available. Even disk I/O or networks do not represent a bottleneck in this situation, but virtually the system architecture. This situation is especially provoked by a large number of users, who induce many processes with a low, even load. Here the processor queue plays a role that cannot be ignored. Therefore, in dependence of the processor performance and the number of processors, there is a point at which the system can no longer serve any further processes and thus clients. In principle, the system in this connection is only concerned with administering the processes. Or put technically: In a multitasking operating system, each process is given a time slice that it cannot release again quickly enough. This situation could only be remedied by smaller time slices and thus greater turn-around times. In other load situations, however, this would have a negative effect on the basic load that the operating system generates. The number of processes in which this situation occurs depends, in addition to the available CPU performance and number, on the applications used so that no general formula can be given for this. This effect mostly does not come to light for power users since other resources, such as memory and computing performance, are the dominating restrictive factors here.
PRIMERGY Scaling

The efficiency of the terminal server is determined by CPU performance and main memory. It is possible to assess the memory configuration quite easily on the basis of the formula:

\[
\text{Memory} = \text{Memory}_{\text{CS}} + \#_{\text{Cert}} \times \text{Memory}_{\text{App}}
\]

If various applications are used at the same time, then it is necessary to form the sum of the memory requirements of all the simultaneously used applications.

The number of users for a specified amount of memory can be calculated with the formula

\[
\#_{\text{Client}} = \frac{\text{Memory} - \text{Memory}_{\text{CS}}}{\text{Memory}_{\text{App}}}
\]

Unfortunately, there is no simple formula for CPU performance. For a medium or heavy user, the CPU performance generally represents the restrictive factor. For a class of light users, the maximum number of users is more restricted by other system resources, such as the number of processes and threads.

The variety of processors, with which each PRIMERGY model can be equipped, gives cause to suspect that there is not a certain number of users who can be served by a PRIMERGY model, but that every PRIMERGY model covers a certain bandwidth. There is also no exact demarcation where the performance of one model ends and that of the next, more powerful one begins. On the contrary, there are overlaps between the systems. The following results that were gained in our series of measurements are therefore only in a position to give an impression of the performance range. The following diagram shows a comparison of the PRIMERGY systems among each other.
Since the systems were measured with different load profiles, no absolute user numbers are specified. On the contrary, the relative performance of the systems is to be shown under terminal server applications. This presentation takes the PRIMERGY RX300 S3 as the reference system and sets the number of users measured on the system as 100%. All the other systems are specified in relation to the reference system with the different load profiles being taken into account accordingly.

If you are planning a terminal server or a terminal server farm, you should take the time to precisely analyze the user behavior beforehand.

- Which applications usually have to be provided by the terminal server?
- Which user uses the application when and how often?
- How intensively is the application used?
- What screen resolution and color depth does the client use?
- Which network and protocol are used?
- What response times are expected?

For larger configurations, a pilot phase under real conditions should not be omitted.
Analysis Tools

In the event of performance problems or also prior to system migration it is helpful to analyze the terminal server environment. Depending on the required degree of detail, various more or less complex methods are suited to provide an overview of the current situation or to record the system load. Below is a brief presentation of the tools used for system analysis. A detailed description of the tools would go beyond the scope of this paper.

Task Manager

The most well-known tool is the Windows Task Manager “taskmgr.exe”, which is to be found in all Windows Server operating systems, and also in Windows XP. The task manager can be used to provide a swift overview of the current system usage, moreover ongoing applications or processes and the network usage can be shown. As an administrator it is also possible to end processes or change their priorities. To select click with the right mouse key on a space in the taskbar and then click on “Task Manager”. The program itself provides online help; other information is to be found at Microsoft in the Internet. On terminal server systems it is advisable to select the setting “Show processes from all users” under “Processes”. In the “Performance” tab, the setting “Show Kernel Times” under “View” can also be a help in monitoring the CPU usage of the kernel separately.

Sysinternals Process Explorer

More detailed information is offered by the freeware program “Process Explorer”, which is provided free-of-charge by the two Windows specialists Mark Russinovich and Bryce Cogswell under www.sysinternals.com. It offers a considerably larger functionality than the Windows task manager. The tree view of the processes, which enables a simple assignment of processes to users especially with terminal server systems, and the extended “System Information” display are particularly worth mentioning. The “Process Explorer” does not need to be installed, but it can be integrated instead of the task manager in Windows. This program provides online help, but the Sysinternals web page also supplies useful information.

Performance Monitor

The above tools are less suited to document and save the course of resource usage. To this end, a system monitor, also known as the Performance Monitor, exists in all current Windows versions, with which it is possible to observe so-called performance counters over a longer period of time. Performance counters are grouped in an object-specific manner, some of them also exist in various instances when an object is available several times over. For example, there is a performance counter, “%Processor Time”, for the object “Processor”, with one instance per CPU existing for a multi-processor system. Not all performance counters are already available in Windows, and many applications, such as Citrix Presentation Server, come with their own performance counters, which integrate in the operating system and can be queried via the performance monitor. The performance data can either be monitored on screen or, better, written in a file and analyzed offline. Not only can performance counters of the local system be evaluated but also of remote servers, which necessitate corresponding access rights. The program, which can be invoked under “permon.exe”, is described in Windows help or in Microsoft articles in the Internet, and there is also an explanation of each individual performance counter under “Explain”.

Please note that the performance monitor is also a Windows application that needs computing time. It is possible with an extreme server overload for the performance monitor itself not to be able to determine and show any performance data; in this case the appropriate values are then 0 or blank.
Windows Performance Toolkit

In its Windows Performance Tools (WPT) Kit Microsoft provides an official toolkit for free use. These tools are suited for the measurement and analysis of system and application behavior as well as resource consumption on Windows Vista, Windows Server 2008 and subsequent Windows operating systems. They are based on Event Tracing for Windows (ETW) and therefore only cause a very small overhead. Through ETW events of the kernel or of applications, such as context switches, interrupts, thread creation and many more, can be recorded in trace files, evaluated at a later date and presented as clearly structured graphs or tables – thus making it possible to analyze Windows system behavior in a very detailed way.

At present the toolkit contains three programs that can be run from the command line:

- Xperf Run event tracing and evaluate
- Xperfview View the performance data
- Xbootmgr Create a trace of booting

Basic procedure consists of four steps:

1) Event tracing is started using “xperf”; parameters can be used to control precisely what is to be traced.
2) Scenario analysis is performed.
3) Event tracing is stopped using “xperf”; all the data required for the analysis are written in the trace file.
4) The trace file is evaluated using “xperf” or also viewed using “xperfview”; can also be done on another machine.

Example of an Xperfview graph for CPU sampling:
Example of an Xperfview graph for disk IO sampling:

The Windows Performance Tool Kit enables a far more in-depth analysis of system behavior than the Performance Monitor. However, due to the fact that the recording of the events for a longer period of time or on a system subject to a high load through numerous processes results in very large trace files, this tool is more suited for the analysis of a “snapshot”, whereas e.g. the Performance Monitor can be used to analyze system behavior over a longer period of time, e.g. for a day or overnight.

Windows Performance Analyzer can only be used to a limited extent on Windows XP and Windows Server 2003. Although event traces can also be written, all the analyses that contain trace decoding – this also includes graphical presentation using “Xperfview” – have to be performed on a Windows Vista and Windows Server 2008 system.
Citrix Resource Manager

The Resource Manager from Citrix is perhaps also used in larger terminal server configurations with Citrix Presentation Server Enterprise Edition. It offers a user-friendlier view of the performance counters relevant to the Citrix Presentation Server and also threshold monitoring with administrator notification. The thresholds are already preset to certain values, but can be adapted when required. Since these thresholds are set in the Citrix Resource Manager software, it is questionable whether they can do justice to the variety of currently available performance categories of server systems without being individually adapted.

On a Citrix Presentation Server 4.0 Terminal Server under Windows Server 2003 Enterprise Edition 32-bit with 16GB main memory and a Gigabit LAN the following values are set by default:

<table>
<thead>
<tr>
<th>Object</th>
<th>Counter</th>
<th>Warning</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Citrix MetaFrame Presentation Server</td>
<td>Data Store Connection Failure</td>
<td>6</td>
<td>60</td>
</tr>
<tr>
<td>Logical Disk</td>
<td>% Disk Time</td>
<td>400</td>
<td>600</td>
</tr>
<tr>
<td>Logical Disk</td>
<td>% Free Space</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>Memory</td>
<td>Available Bytes</td>
<td>858924851</td>
<td>343569940</td>
</tr>
<tr>
<td>Memory</td>
<td>Pages/sec</td>
<td>3000</td>
<td>5000</td>
</tr>
<tr>
<td>Network Interface</td>
<td>Bytes Total/sec</td>
<td>3000000</td>
<td>4000000</td>
</tr>
<tr>
<td>Paging File</td>
<td>% Usage</td>
<td>40</td>
<td>50</td>
</tr>
<tr>
<td>Processor</td>
<td>% Interrupt Time</td>
<td>0.3</td>
<td>0.4</td>
</tr>
<tr>
<td>Processor</td>
<td>% Processor Time</td>
<td>80</td>
<td>90</td>
</tr>
<tr>
<td>System</td>
<td>Context Switches/sec</td>
<td>120000</td>
<td>14000</td>
</tr>
<tr>
<td>Terminal Services</td>
<td>Active Sessions</td>
<td>50</td>
<td>100</td>
</tr>
<tr>
<td>Terminal Services</td>
<td>Inactive Sessions</td>
<td>10</td>
<td>20</td>
</tr>
</tbody>
</table>

However, if the thresholds are adapted to the regular operations (baseline) of a Citrix system, then any overstepping triggers a warning or an alarm in good time and this helps the administrator detect a bottleneck early. For the Citrix Resource Manager, reference is made to the attached documentation from Citrix.
Microsoft Operations Manager 2005 (MOM)

In Microsoft Operations Manager (MOM) 2005 Microsoft makes a powerful software product available, with which events and the system performance of various server groups can be monitored within the company network. MOM offers proactive notifications in the event of alerts and errors as well as creates reports and trend analyses.

In doing so, MOM makes use of existing Windows resources, such as the event log, reads performance counters and monitors services, and this in a convenient way for whole groups of servers. The results are stored in an SQL database.

Additional management packs for MOM are offered by Microsoft and other vendors for their products and store further system or application-specific rules and reports in the MOM database. In this way, MOM can not only monitor terminal servers, but a variety of different server types at the same time.

For Terminal Services it is possible to add the “Microsoft Terminal Server MOM 2005 Management Pack”, which includes terminal-server-specific rules and reports for the following components

- Terminal Services
- Terminal Services Licensing Server
- Session Directory Server

and offers performance monitoring.

For example, an alert is given when more than 520 terminal server sessions are active or the processor load of a session is over 80% for a period of 15 minutes. An error is displayed if more than 20 terminal server sessions are inactive or the cache hit rate is less than 99%. These predefined rules must be adapted to customer-specific circumstances.

It is also possible to observe active and inactive sessions for certain periods of time so that statements about user behavior can be obtained.

A description of the features of MOM would go beyond the scope of this document. Detailed descriptions are available from Microsoft in the Internet about MOM and about the individual management packs.
Bottleneck Analysis

In order to be able to forecast how an existing terminal server environment behaves or be able to evaluate the load of a terminal server in a pilot installation, it is necessary to perform a bottleneck analysis yourself or have one performed by our Professional Service.

Recording or monitoring performance values is less of a problem than drawing the right conclusions from the measurement values. Before we discuss some important performance counters for various performance-relevant server objects below, we will briefly present some typical behavior patterns of performance counters. If you analyze a bottleneck on a server system, it is not easy to draw conclusions from a snapshot of the bottleneck situation. It is beneficial if you have experience of how the server system behaves under various loads and if you have prepared a baseline as a comparison.

While you can put some measured values, such as network throughput, in relation to the theoretically achievable maximum value, e.g. in a 100Mbit network connection, this is not possible with other measured values, such as the number of context switches per second.

The following typical curves exist for synthetic load:

**Linear:**
- The performance values increase in the same ratio to the load. No bottleneck.

**Logarithmic:**
- If the load increases, the performance counters do not continue to increase, but reach saturation. This can be an indication of a bottleneck.

**Constant:**
- The performance counters do not depend on the load, but are constant. It is possible with a different load for different stages of a step function to be formed.

**Exponential:**
- In the lower load range the curve rises relatively gently, while it grows disproportionately with an increasing load. In an extreme case, it is possible here for one more user to be the last straw that breaks the camel’s back.

Below, individual performance counters, grouped according to system components, are described and discussed. Subsequently we have added a sample script, which after slight adaptation can be directly used for a terminal server system.
Computing Performance/System

The most important performance counters are:

- \Processor(*)\% Processor Time or \Processor(_Total)\% Processor Time
- \System\Processor Queue Length

“% Processor Time” is the total computing time used. On every Windows system a “System Idle Process” runs, which can also be seen using the Task Manager or Process Explorer. All CPU time, which this idle process does not receive, is “% Processor Time”. In “% Processor Time” there are several instances (one per logical CPU) which Windows knows. Our sample configuration below uses the object “Processor(*)” to record all existing processors and their total. The usage of the individual processors does not always need to be equally high, particularly in the low-load range, and this is often the case with Hyper-Threading. And when working normally with terminal server applications, an overall 100% CPU load is seldom achieved. The more logical processors work in parallel, the lower the average value of CPU load can be — even if the terminal server is already overloaded.

“Processor Queue Length” denotes the number of threads waiting to be executed. This counter is only available once per system, i.e. not on a CPU-specific basis. Therefore this value should not permanently exceed the number of processors.

Normally, “% Processor Time” and “Processor Queue Length” increase together when the server is put under load. The diagram opposite shows typical curves of processor time and “Processor Queue Length”, while the load on the terminal server is increasingly growing. % CPU times as of 80% together with high peaks in the PQL curve, which then also permanently remain at a higher level, are critical.

However, there can also be situations, in which a variety of small processes are waiting to be run and the system is already overloaded due to process management, although the necessary CPU resources were still available.

Additional performance counters:

- \System\Context Switches/sec
- \System\System Calls/sec
- \System\Processes
- \Processor(*)\Interrupts/sec or \Processor(_Total)\Interrupts/sec

“System Calls/sec” counts the number of system calls per second. “Context Switches/sec” counts the number of context switches per second, i.e. switches between threads or between user programs and kernel activities. “Processes” is the number of currently running processes. “Interrupts/sec” is a counter, which exists per processor instance and as an average of all processors. Interrupts are mostly events that are initiated by hardware, such as network cards, keyboard, mouse, system clock, etc., and which the processor has to handle. It should be noted that Windows in the x64 edition has considerably higher values for interrupts/sec than with the 32-bit Windows. However, this difference is solely due to the counting of the interrupts, because 64-bit Windows also documents the interrupt forwarding between the processors, which is not counted under 32-bit Windows.

Normally these counters increase together with the needed computing performance. It is critical when these curves reach saturation, which means that the system cannot process any more of these events. Saturation can best be assessed over a longer period of time in connection with the other performance counters. The number of users increases more steeply than the recorded performance counter.

A CPU bottleneck can also be caused by faulty applications, see “Applications”.
Main Memory

The most important performance counters with regard to main memory usage are:

- `\Memory\Available MBytes`
- `\Memory\Pages Input/sec`
- `\Process(_Total)\Working Set`

"Available MBytes" is the currently free physical main memory in MBs. "Pages Input/sec" shows the number of pages read from disk per second. The "Working Set" of an application is the memory it has already used.

The performance record opposite was created on a PRIME server, which was intentionally driven into a main memory bottleneck. One feature of the Windows operating system can be seen here: If sufficient main memory is available, more memory is occupied. "Tidying up" and memory swapping outside the "Working Set" is only done when the free memory falls below a certain threshold value. This means that it is not possible to implicitly read the actual memory requirements of a system that is operated outside the memory limit range. However, in the bottleneck situation that was provoked here, excessive paging activities become evident. The memory bottleneck is already looming when "Available Mbytes" falls below a threshold and Windows therefore decreases the "Working Set". At the same time the paging activities ("Pages Input/sec") increase. If it now happens that more memory is needed, the "Working Set" is then further reduced step by step and the paging activities increase notably. The performance counter "Paging File\%Usage" also increases accordingly.

The following performance counters can also give an indication as to the usage of the main memory:

- `\Memory\Committed Bytes`
- `\Memory\Pages Output/sec`
- `\Memory\Pages Faults/sec`
- `\Paging File(\??\C:\pagefile.sys)\% Usage`

"Committed Bytes" specifies how much virtual main memory was assigned to running applications and reserved in page file. This can be used to recognize whether the system is getting into a bottleneck with the virtual address space.

"Pages Output/sec" shows the number of pages output per second, through which physical main memory is released by writing the data on the hard disk. Even if sufficient main memory is available, a moderate output takes place. A memory bottleneck is only given when this value increases dramatically. Particular attention must be paid to the fact that this value harmonizes with the capacity of the hard disk on which the page file is to be found.

"Pages Faults/sec" is an average value of page faults per second, but includes both page input from the physical main memory (soft fault) and accesses to the hard disk (hard fault).

"Paging File\%Usage" of the page file is its occupation expressed as a percentage. It is normal for the page file to be slightly occupied – even if sufficient RAM is available. However, if this increases significantly, a main memory bottleneck could be to hand – this is substantiated by the other performance counters of the paging activities.

A main memory bottleneck goes hand in hand with an increase in the disk counters on the hard disk that houses the page file (see section Bottleneck Analysis - Disk Subsystem).
Operating System Resources

The most important performance counter with which you can analyze the health of the operating system is:

- Cache\Copy Read Hits %

“Copy Read Hits %” is the percentage of successful read operations from the system cache. If data are found in the system cache, hard disk access is superfluous which in turn results in an enormous gain in performance. According to Microsoft this percentage should always be above 95%. However, with this performance counter a deterioration of a few percent can already be perceived through a delayed reaction of the Terminal Server. Ideally “Copy Read Hits %” is above 99% during steady-state server operation. In comparison with the 64-bit operating system a 32-bit operating system also has - as a result of the limited kernel address space - a smaller system cache.

Since operating system bottlenecks in particular are analyzed in the terminal server environment, the following additional performance counters must by all means also be taken into consideration:

- Memory\Free System Page Table Entries
- Memory\Pool Nonpaged Bytes
- Memory\Pool Paged Bytes

“Free System Page Table Entries” shows the number of free entries in the system page table. A bottleneck in this operating system resource is reached when only a few 1000 entries are still available. “Pool Nonpaged Bytes” is the number of bytes that is currently used for the non-paged pool. In the non-paged pool the operating system saves all the data that always have to be kept in the main memory and may not be written into the page file. “Pool Paged Bytes” is the number of bytes that are currently used for the paged pool. System memory areas, which can be written into the page file, are to be found in the paged pool. On Windows Server 2003 32-bit operating systems these memory areas are limited and cannot be extended, so with a great many users working on the terminal server, these limits can be reached. A larger memory configuration does not result in larger kernel structures; on the contrary even more kernel memory is occupied by the administration and addressing of these data areas.

The diagram opposite shows such a situation, in which a bottleneck of the operating system resources of a terminal server was provoked by continuously increasing the user load. The performance counter “Working Set” is also depicted in order to illustrate this. Four phases of server load can be seen. In the first phase the terminal server is high-performance, the hit rate in the system cache is almost 100% 1, sufficient free PTEs 2 are still to hand and the working set 3 as well as the paged pool 4 and the non-paged pool 5 can be increased with the number of users. In the second phase the hit rate of the system cache decreases. The paged pool experiences saturation. Performance will slowly decrease. In the third phase the efficiency of the system caches is considerably degraded, whereas paged pool resources can still be set free through paging activities. However, the terminal server performance will no longer be satisfactory due to the lack of capacity in the system cache together with increased disk activities. If the system is overloaded beyond this phase, saturation of the paged pool can then be seen. It is also no longer possible to increase the “Working Set” and the free PTEs have decreased to a minimum value. In addition to a poor terminal server performance, part of the users will also no longer be able to logon or applications can no longer be started or handled.

However, the system has, in contrast to the case of a main memory bottleneck shown in the section “Main Memory”, still sufficient free physical memory. In this case the system still had 26 GB free of 32 GB physical main memory at the end of the measurement.
Disk Subsystem

The most important performance counter to assess disk usage is:

- `\PhysicalDisk(*)\Avg. Disk Queue Length`
- `\LogicalDisk(*)\Avg. Disk Queue Length`

The performance counters for the "PhysicalDisk" record all activities on a physical disk, whereas the "LogicalDisk" object divides these activities according to the individual partitions, e.g. C: or D:.

If a RAID controller, SAN or iSCSI is used, the "PhysicalDisk" counter does not, as the name suggests, refer to the physical disk but to the RAID array or the LUN.

The counter "Avg. Disk Queue Length" is the average of the disk queue for read and write operations, but separate counters can also be activated for read and write operations. This performance counter is a significant indicator for the load of the disk subsystem and should not be permanently greater than 1 per net available hard disk. In a RAID 1 configuration consisting of two hard disks it should therefore not permanently exceed 1, in a RAID 1+0 with 6 hard disks it should not be permanently higher than 3.

The sample diagram shows the "Avg. Disk Queue Length" for a configuration with one hard disk, for which the value should be below 1. Whereas the hard disk is not put under a considerable load during the first period of this recording, the disk subsystem is clearly overloaded at the end of the measurement.

The performance of the disk subsystem, on which the page file is to be found, should always be regarded together with the Main Memory.

If main memory runs out, Windows relocates parts of the memory to the page file on the hard disk and re-imports these later, which results in distinctly increased disk activities on the hard disk where the page file lies.

Additional performance counters of the `\PhysicalDisk(*)` and `\LogicalDisk(*)` objects, which can give a further indication as to disk activity:

- `Avg. Disk Bytes/Read` Average read request size
- `Avg. Disk Bytes/Write` Average write request size
- `Avg. Disk sec/Read` Average value of the time required in seconds per read request
- `Avg. Disk sec/Write` Average value of the time required in seconds per write request
- `Disk Read Bytes/sec` Total number of bytes read per second
- `Disk Reads/sec` Total number of read operations per second
- `Disk Write Bytes/sec` Total number of bytes written per second
- `Disk Writes/sec` Total number of write operations per second

The frequently used counter "% Disk Time" is not to be recommended, as in the present Windows versions it is equivalent to precisely one hundred times the "Avg. Disk Queue Length".
The pattern with which the hard disks are accessed can be read from the above mentioned performance counters, i.e. the read and written block sizes and the latency times ("Avg. Disk sec") when reading and writing. In this case, these measured values should be evaluated as additional information if "Avg. Disk Queue Length" points toward a hard disk bottleneck. The number of read and write operations per second can be used in connection with the average request size used and the RAID level used in order to draw conclusions as to the usage of the disk subsystem. A separate white paper "Performance Report – Modular RAID for PRIMERGY" [L5] is available for the sizing of disk subsystems. 

It should also be noted that a disk subsystem, which is connected via iSCSI, appears - in addition to the performance counters of the hard disk objects - among the performance counters of the network card, which are described below.

**Network**

A first view of network load is provided by the Task Manager. The “Networking” page gives you a quick outline of the “Network Utilization” of the individual network cards and their “Link Speed”.

In a performance monitor recording, the following performance counters should also be taken into consideration:

- Network Interface(*)\Bytes Sent/sec
- Network Interface(*)\Bytes Received/sec

"Bytes Sent/sec" and “Bytes Received/sec” count in bytes - from the server’s viewpoint - the data sent and received.

In addition, it also makes sense to consider the performance counters

- Network Interface(*)\Packets Received/sec
- Network Interface(*)\Packets Sent/sec

“Packets Sent/sec” and “Packets Received/sec” are the numbers of data packets sent and received from the viewpoint of the server. The average packet size can be determined from the ratio of the data bytes and the number of packets.

The larger the network packets, the higher the possible network utilization.

The protocols RDP or ICA, which a terminal server uses between client and server, do not contain very large data packets. Therefore, 100% utilization of a 100 MBit network will hardly be observed. The ICA protocol of the Citrix Presentation Server is in this regard still somewhat more economical than the RDP protocol of the Microsoft Terminal Services.

However, in most application scenarios, in addition to client/server communication, other network traffic also occurs to back-end servers, to the Internet, or, when using a network-based disk subsystem such as NAS or iSCSI, data transfer to the disk subsystem. It is highly advisable to use dedicated network interfaces for these different data flows.

The various network interfaces can also be selected individually by their name instead of the asterisk (*) in the performance monitor. If (*), as in the example above, is specified as the network card name, then all the available network cards in the system are recorded. This also includes the loopback interface, on which there is not usually very much network traffic.

It should be noted that that the performance monitor or the task manager can only analyze the usage of the network from a terminal server viewpoint. Data traffic from other servers and clients on the network cannot be recorded. However, it is often not the server-side network itself that is too small in size, but the problems can for example be based on faulty or incorrectly configured components or arise as a result of varied problems in the connection of these components. At this point, a network specialist should be consulted.
Terminal Server

The Microsoft Terminal Server itself does not provide many performance counters. However, under the object "Terminal Services Sessions", Terminal Server provides performance counters per session for a detailed analysis of individual sessions.

The number of terminal server connections should also always be recorded in order to be able to put the other performance counters in relation to the number of connected users:

- \Terminal Services\Active Sessions

It is a difference whether the user ends the connection to the terminal server by logging off or whether he only interrupts the connection with a "Disconnect". In the latter case, the application continues to run on the terminal server and does not release its resources. The user can continue his work at this juncture. Some applications still require processor resources even after the connection has been disconnected. This is why the number of inactive sessions

- \Terminal Services\Inactive Sessions

should also be included in the performance protocol.

These two parameters are likewise valid for Microsoft Terminal Services and for Citrix Presentation Server.

Citrix itself also supplies performance objects with various performance counters. The following performance objects are installed:

- "Citrix CPU Utilization Mgmt User"
  Counter for the feature "CPU Utilization Management"
- "Citrix IMA Networking"
  Counter for the number of connections and network load
- "Citrix MetaFrame Presentation Server"
  Counter among other things for data store, local host cache, licensing
- "ICA Session" (for individual sessions or in total "_Server Total")
  Counter specific to ICA session

However, these are more suited for special evaluations and less for a general performance analysis of a terminal server.
Applications

A faulty application can put a load on a CPU from time to time or permanently, and on a terminal server in particular this can have fatal consequences for the overall system performance, because this application can also be started several times. If the application has a thread, which e.g. runs into an endless loop or "actively waits", this then occupies at most "100% / #CPUs" processor time. On a system with two logical processors 50%, on a system with four logical CPUs 25%. The more logical CPUs the system has, the less such a program attracts attention. In the event of CPU bottlenecks a check should always be made whether such applications exist which permanently need precisely this percentage of CPU performance. Short-term load peaks of an application are on the other hand completely normal. A first glance at the applications is made possible by Task Manager or Process Explorer. The following screenshot shows a synthetic load simulator on a PRIMERGY system with eight logical processors, which simulates an endless loop. Process Explorer indicates a CPU load of 12.50%, i.e. this one application puts a full load on a processor core.

Such behavior can often be found in desktop applications that are written for a single user and are therefore only conditionally suited for a terminal server environment. If Citrix Presentation Server 4.0 is used, then the computing performance of such applications can be restricted. This restriction is only applicable if too few CPU resources on the whole are available.

The memory requirements of an application should also be checked. Our synthetic load simulator has occupied approximately 1.5GB of virtual memory, but the working set is only 5.3MB. Although this application has reserved a great amount of memory, it is not using it. The working set only increases and the memory available in the system ("Available Mbytes") only decreases when the application works on the memory.

The performance monitor can be used to also show numerous performance counters for individual running applications instead of for the overall system. For this purpose, the object "Process" is used and then as an instance the process that is to be monitored.
Performance Monitor Configuration Script

At this point we would like to present a sample configuration with the most significant performance counters, which can if required be extended. Normally, the selection of performance counters is put together with the performance monitor program and these are then saved as a "Counter Log" as a *.htm file. This HTML file can also be imported to another environment, but a number of the parameters have – where applicable – to be set specific to the system, and the performance counters must be renumbered when performance counters are added or removed.

With remote monitoring, the name of the server must be placed in the format “\<server>” before the performance counter, e.g. for the server “obelix”: “\ibelix\Terminal Services\Active Sessions”. In addition to an individual server it is also possible to monitor several servers at the same time with remote monitoring.

Alternatively, the configuration of the performance counters can also be written in a text file and loaded using the command “logman”. To this end, see the help of the “logman” command by using “logman?”

Opposite is a complete configuration script. It can be set up as a HTML file, e.g. with the name “perf.htm”, best of all using a text editor like notepad, and then imported in the performance monitor using “New Log Settings From ...”. Especially important parameters are marked in color, whereas the bold drive letter has, where applicable, to be adapted to the memory location of the page file.

```xml
<DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HEAD>
  <META NAME="GENERATOR" CONTENT="Microsoft System Monitor">
</HEAD>
<BODY>
  <OBJECT ID="DiSystemMonitor1"> WIDTH="100%" HEIGHT="100%"
  CLASSID="CLSID:C4D2DBE0-D1DD-11CE-940F-008029004347">
    <PARAM NAME="sampleIntervalValue" VALUE="15">
    <PARAM NAME="SampleIntervalUnitType" VALUE="1">
    <PARAM NAME="UpdateInterval" VALUE="15">
    <PARAM NAME="Counter00001.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00002.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00003.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00004.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00005.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00006.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00007.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00008.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00009.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00010.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00011.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00012.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00013.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00014.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00015.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00016.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00017.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00018.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00019.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00020.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00021.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00022.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00023.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00024.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00025.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00026.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00027.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00028.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00029.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00030.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00031.Path" VALUE=""></PARAM>
    <PARAM NAME="Counter00032.Path" VALUE=""></PARAM>
    <PARAM NAME="EOFCommandFile" VALUE="">
    <PARAM NAME="RestartMode" VALUE="0">
    <PARAM NAME="LogFileType" VALUE="2">
    <PARAM NAME="LogFileFolder" VALUE="C:\">
    <PARAM NAME="LogFileBaseName" VALUE="PerfLogs">
    <PARAM NAME="LogFileAutoFormat" VALUE="1">
    <PARAM NAME="LogFileType" VALUE="2">
    <PARAM NAME="StartMode" VALUE="0">
    <PARAM NAME="StopMode" VALUE="0">
    <PARAM NAME="EOFCommandFile" VALUE="">
    <PARAM NAME="RestartMode" VALUE="0">
    <PARAM NAME="LogFileType" VALUE="2">
    <PARAM NAME="LogFileFolder" VALUE="C:\">
    <PARAM NAME="LogFileBaseName" VALUE="PerfLogs">
    <PARAM NAME="LogFileAutoFormat" VALUE="1">
    <PARAM NAME="LogFileType" VALUE="2">
    <PARAM NAME="StartMode" VALUE="0">
    <PARAM NAME="StopMode" VALUE="0">
</OBJECT>
</BODY>
</HTML>
```
Measurement Tools

Since there is no standard benchmark for terminal server, a series of various proprietary measurement tools and methods exist for the scaling of terminal servers. Apart from Fujitsu Technology Solutions, Microsoft and Citrix also use their own software, and the well-known Microsoft-Press author and terminal server expert Dr. Bernhard Tritsch has also carried out analyses with a separate load simulator. The company LoginConsultants, which was founded in 2002, provides a free tool on its Internet site for the measuring of server-based computing sessions.

The table below characterizes the various tools.

• Only runs under Microsoft Terminal Services (RDP protocol)  
• 20 users maximum with one load generator  
• Knowledge Worker works with a remote desktop and various applications (Microsoft Excel, Outlook, Internet Explorer, Word)  
• Keyboard input, 35 words per minute  
• Record of performance counters with the Windows Performance Monitor |
|---|---|
| Citrix CSTK / Citrix ICAMark 3.0 | • CSTK is a load simulator offered by Citrix (http://www.citrix.com/cdn).  
• Citrix ICAMark 3.0 is an internal Citrix tool based on CSTK.  
• Only for Citrix Presentation Server (ICA protocol.)  
• Generates load but does not measure response times for individual actions—only the total running time for a complete script can be measured.  
• Light User: Microsoft Excel, around 20 words per minute  
• Heavy User: Microsoft Excel, Microsoft Access, Microsoft PowerPoint (sequential) around 40 words per minute  
• Record of performance counters with the Windows Performance Monitor |
| Fujitsu Technology Solutions T4US | • Self-development from Fujitsu Technology Solutions  
• Both for Microsoft Terminal Services and for Citrix Presentation Server  
• Up to 40 users per load generator  
• Monitoring of response times for all users for various actions, comparison with reference measurement  
• Medium-load profile: Login + Desktop + Office applications + Logoff  
  Keyboard and mouse input; reading and writing data  
• Record of performance counters with the Windows Performance Monitor |
| visionapp Remote Desktop + vRD Load Edition | • Self-development from visionapp (http://www.visionapp.com)  
• Measurements with Microsoft Terminal Services (RDP protocol)  
• Up to 40 users per load generator  
• Desktop + applications:  
  Notepad: Text file, 8.4 KB  
  Microsoft Word: Word document, 3.1MB  
  Acrobat Reader 7: PDF file, 3.8MB  
• Apart from the user logon, no input is effected during the test, no data is created by the applications  
• Measurement of the logon times as well as the number of started connections and applications  
• Record of performance counters with the Windows Performance Monitor |
| Login Consultants Login Virtual Session Indexer (VSI) | • Self-development from LoginConsultants (http://www.loginconsultants.com)  
• VSI: free version with a (medium) load profile, cannot be modified.  
• VSIpro: subject-to-license customizable version, various load profiles  
• Measurements performed on the SUT, therefore protocol-independent  
• Medium-load profile: Login, Outlook, Internet Explorer, Word, Excel, PowerPoint, PDFWriter actions  
• Continuous starting of user sessions and measuring of response times  
• VSI result: the maximum number of virtual sessions based on response times |
Literature

[L1] General information about Fujitsu Technology Solutions products
http://www.ts.fujitsu.com

[L2] General information about the PRIMERGY product family
http://www.primergy.com

[L3] PRIMERGY Benchmarks • Performance Reports and Sizing Guides

http://docs.ts.fujitsu.com/dl.aspx?id=1a55e793-4231-490d-8941-42c89f5670ad

[L5] Performance Report - Modular RAID for PRIMERGY
http://docs.ts.fujitsu.com/dl.aspx?id=8f6d5779-2405-4cdd-8268-1f948ba050e6

http://www.microsoft.com/terminalserver

http://www.microsoft.com/windowsserver2003/techinfo/overview/tsscaling.mspx

[L8] Remote Desktop Services Windows Server 2008 R2

[L9] Citrix
http://www.citrix.com

[L10] Sysinternals Tools
www/sysinternals.com


Contact

PRIMERGY Hardware
PRIMERGY Product Marketing
mailto:Primergy-PM@ts.fujitsu.com

PRIMERGY Performance and Benchmarks
PRIMERGY Performance and Benchmarks
mailto:primergy.benchmark@ts.fujitsu.com