



Technical Whitepaper

The Big Iron Test

Scalability of Large Terminal Server Platforms

Dr. Bernhard Tritsch
Chief System Architect
visionapp GmbH, Frankfurt/Main



Copyright © visionapp GmbH, 2004. All rights reserved.

The information, concepts, and ideas contained in this document are the property of visionapp GmbH. No part of this document may be duplicated, photocopied, reproduced, translated, transferred to an electronic medium, or put in machine-readable form without the prior written permission of visionapp GmbH.

The information and data contained in this document are subject to change without notice.

All brand names and product names used in this document are trademarks of their respective holders and are recognised as such.

visionapp GmbH
Theodor-Heuss-Allee 110
60486 Frankfurt a.M.
Germany
Phone: +49-69-79583-0
Fax: +49-69-79583-599
E-mail: info@visionapp.de
Web: <http://www.visionapp.de>



1 Contents

1	Contents	3
2	Introduction	4
3	Test Setup	5
3.1	Test Environment	5
3.2	Testing Tools and Scripts.....	6
3.2.1	Creating the User Accounts.....	6
3.2.2	Installing the Applications and Providing Documents.....	8
3.2.3	Creating the User Sessions	8
3.2.4	Creating the User Profiles	9
3.3	Testing Methodology	10
4	Analysis of the Results	12
4.1	Windows Server 2003 with HyperThreading turned on.....	12
4.2	Windows Server 2003 with HyperThreading turned off	16
4.3	Windows 2000 Server with HyperThreading turned off	19
5	Conclusion	20
6	Appendix	21
6.1	System Information	21
6.2	Performance Counters Analysed	23

2 Introduction

Microsoft Windows Terminal Server allows users to run Windows-based applications on a remote Windows 2000 or Windows 2003-based server. This white paper focuses on large server platforms (8 CPUs, 8 GB RAM) and contains scalability test results for both Windows 2000 Server and Windows Server 2003 Terminal Servers in application server mode.

Note: For the purposes of this document, Terminal Server specialists from visionapp GmbH worked in cooperation with Fujitsu Siemens Computers server specialists to perform the tests and data collection in Fujitsu Siemens Computers labs. The tests were performed using Windows Server 2003, Enterprise Edition and Windows 2000 Server, Enterprise Edition.

In a server-based computing environment, all application execution and data processing occurs on the server. Therefore it is useful and desirable for server manufacturers to test the scalability and capacity of their servers to determine how many client sessions a server can typically support under a variety of different scenarios. This test focuses on large server platforms with eight processors and eight megabytes of memory.

Please note that the results and analysis contained here should not be interpreted in isolation. The applications used in the test executed in Terminal Server sessions on specific hardware and were intended to consume a reproducible amount of server resources. The tests assume a rather static quality, with users only logging onto the system, starting three applications with corresponding documents, and subsequently staying inactive for the rest of their session. This helps to create reproducible results, but your results may vary.

3 Test Setup

Server specialists from visionapp GmbH and Fujitsu Siemens Computer performed the tests at the Fujitsu Siemens Computers Laboratories in Augsburg, Germany. They used specific test scripts and tools. A tool for creating the test user accounts, Console User Manager, was provided by Microsoft Consulting Services. The test laboratory used Fujitsu Siemens Computers servers.

3.1 Test Environment

The Terminal Services testing laboratory provided a simple setup using two types of servers:

- 1 Fujitsu Siemens Computers Primergy T850 Enterprise Server; 8 x Intel XEON 1.6 GHz CPUs, 1 megabyte second-level cache, 8 gigabytes of memory
- 20 Fujitsu Siemens Computers BX300 Blade Servers used as client; 2 x Intel 966 MHz CPUs, 2 gigabytes of memory

The Primergy T850 Enterprise Server was booted with the /PAE boot.ini switch in order to be able to access the complete memory. Windows 2000 Advanced Server, Windows 2000 Datacenter Server, Windows Server 2003, Enterprise Edition, and Windows Server 2003, Datacenter Edition, can use /PAE to take advantage of physical memory beyond 4 gigabytes.

Other switches should not be used. Using the /3GB (for Windows 2000) or the /4GT (for Windows 2003) boot.ini switches is inappropriate for Terminal Server environments because those switches change the partition between the application memory space and kernel memory space. These switches give each application three gigabytes of memory, which in turn only leaves 1 gigabyte for the kernel - a disaster in Terminal Server environments. I doubt that an individual application will consume 3 gigabytes of memory on a Terminal Server very often. On large Terminal Servers, it only matters if an application is able to access physical memory space beyond four gigabytes - and this is exactly what the /PAE boot.ini switch provides.

Other components of the testing laboratory included a Gigabit Ethernet infrastructure.

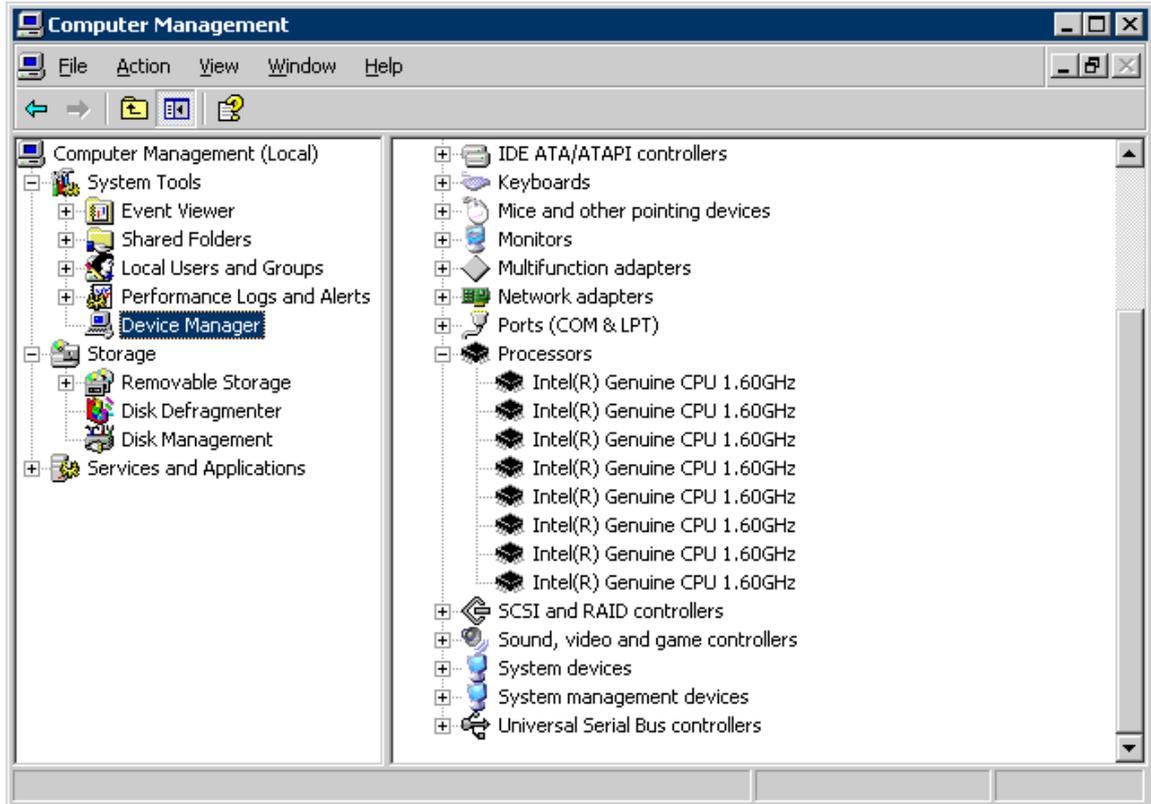


Figure 3-1: Screenshot of the Fujitsu-Siemens Primergy T850 Enterprise Server Device Manager showing eight physical CPUs

3.2 Testing Tools and Scripts

To assist with testing, visionapp GmbH and Fujitsu Siemens Computers developed the testing scripts used on the client computers.

3.2.1 Creating the User Accounts

Before the start of the test, 150 user accounts had to be created. This was accomplished using two scripts and a command line.

```
Echo Create 150 user accounts >>User150.log
time /T >> user150.log
Date /T >> user150.log
call nuser.cmd FSC1 "FSC1 User" x
call nuser.cmd FSC2 "FSC2 User" x
call nuser.cmd FSC3 "FSC3 User" x
...
call nuser.cmd FSC148 "FSC148 User" x
call nuser.cmd FSC149 "FSC149 User" x
call nuser.cmd FSC150 "FSC150 User" x
Echo Finished the creation of 150 user accounts >>User150.log
time /T >> user150.log
```

```
Date /T >> user150.log
```

Listing 3.1: Script User150.cmd, which creates 150 user accounts by using the secondary script Nuser.cmd.

The secondary script, Nuser.cmd, was used to create the individual user accounts, add them to the Remote Desktop Users group and set some required user's attributes.

```
call net user %1 /ADD
call net group "Remote Desktop Users" %1 /add
call cusrmgr.exe -u %1 -f %2 -P %3 -s PasswordNeverExpires
```

Listing 3.2: Script Nuser.cmd.

Changing the user's attributes requires the usage of a command line tool that was provided by Microsoft Consulting Services. In the following, the options of the Console User Manager are displayed.

```
CUsrMgr Ver 1.0 Jan98 by G.Zanzen (c) MCS Central Europe
Sets a random password to a user
usage: -u UserName [-m \\MachineName] \\ default LocalMachine
Resetting Password Function
  -p Set to a random password
  -P xxx Sets password to xxx
User Functions
  -r xxx Renames user to xxx
  -d xxx deletes user xxx
Group Functions
  -rlg xxx yyy Renames local group xxx to yyy
  -rgg xxx yyy Renames global group xxx to yyy
  -alg xxx Add user (-u UserName) to local group xxx
  -agg xxx Add user (-u UserName) to global group xxx
  -dlg xxx deletes user (-u UserName) from local group xxx
  -dgg xxx deletes user (-u UserName) from global group xxx
SetProperties Functions
  -c xxx sets Comment to xxx
  -f xxx sets Full Name to xxx
  -U xxx sets UserProfile to xxx
  -n xxx sets LogonScript to xxx
  -h xxx sets HomeDir to xxx

  -H x sets HomeDirDrive to x

+s xxxx sets property xxxx
-s xxxx resets property xxxx
where xxxx can be any of the following properties:
  MustChangePassword
  CanNotChangePassword
  PasswordNeverExpires
  AccountDisabled
  AccountLockout
  RASUser
returns 0 on success
```

Listing 3.3: CUsrMgr by MCS Central Europe.

3.2.2 Installing the Applications and Providing Documents

The following applications were installed on the Terminal Servers.

- Microsoft Office 2000. For the test installation, the termsrvr.mst transformation file from the Office 2000 Resource Kit was used.
- Adobe Acrobat Reader 6

During the test three applications were launched using the following documents.

- Notepad.exe – Terminalservices.txt: ANSI text file, 3,140 bytes
- Acrobat Reader – PAG3.pdf: Adobe Acrobat document, 8,399,585 bytes
- Microsoft Word – FSCDemo.doc: Microsoft Word document, 2,388,992 bytes

During the tests, the applications showed specific memory footprints, all related to the documents loaded. Winword.exe used between 13 and 18 megabytes, Notepad.exe used approx. 3 megabytes, and Acrobat Reader used almost 30 megabytes.

3.2.3 Creating the User Sessions

The following script was applied to create user-specific RDP files. These RDP files were used to launch the individual user sessions.

```
@echo off
if "%1"==" " goto error
set rdpusername=fsc
echo screen mode id:i:1 > %rdpusername%%1.rdp
echo desktopwidth:i:800 >> %rdpusername%%1.rdp
echo desktopheight:i:600 >> %rdpusername%%1.rdp
echo session bpp:i:16 >> %rdpusername%%1.rdp
echo winposstr:s:2,3,0,0,800,600 >> %rdpusername%%1.rdp
echo full address:s:p-t850-a >> %rdpusername%%1.rdp
echo compression:i:1 >> %rdpusername%%1.rdp
echo keyboardhook:i:2 >> %rdpusername%%1.rdp
echo audiomode:i:0 >> %rdpusername%%1.rdp
echo redirectdrives:i:0 >> %rdpusername%%1.rdp
echo redirectprinters:i:0 >> %rdpusername%%1.rdp
echo redirectcomports:i:0 >> %rdpusername%%1.rdp
echo redirectsmartcards:i:0 >> %rdpusername%%1.rdp
echo displayconnectionbar:i:1 >> %rdpusername%%1.rdp
echo autoreconnection enabled:i:1 >> %rdpusername%%1.rdp
echo username:s:fsc%1>> %rdpusername%%1.rdp
echo domain:s:p-t850-a >> %rdpusername%%1.rdp
echo alternate shell:s: >> %rdpusername%%1.rdp
echo shell working directory:s: >> %rdpusername%%1.rdp
echo disable wallpaper:i:1 >> %rdpusername%%1.rdp
echo disable full window drag:i:1 >> %rdpusername%%1.rdp
echo disable menu anims:i:1 >> %rdpusername%%1.rdp
echo disable themes:i:1 >> %rdpusername%%1.rdp
echo disable cursor setting:i:0 >> %rdpusername%%1.rdp
echo bitmapcachepersistenable:i:1 >> %rdpusername%%1.rdp

start /B "%programfiles%\remote desktop\mstsc.exe" %rdpusername%%1.rdp
goto success

:error
echo Error: Parameter is missing
goto end
```

```

:success
echo Success
goto end

:end

```

Listing 3.4: The script Cr.cmd to create a new session

Executing the script Cr.cmd with the parameter "1" created the following RDP file:

```

screen mode id:i:1
desktopwidth:i:800
desktopheight:i:600
session bpp:i:16
winposstr:s:2,3,0,0,800,600
full address:s:p-t850-a
compression:i:1
keyboardhook:i:2
audiomode:i:0
redirectdrives:i:0
redirectprinters:i:0
redirectcomports:i:0
redirectsmartcards:i:0
displayconnectionbar:i:1
autoreconnection enabled:i:1
username:s:fsc1
domain:s:p-t850-a
alternate shell:s:
shell working directory:s:
disable wallpaper:i:1
disable full window drag:i:1
disable menu anims:i:1
disable themes:i:1
disable cursor setting:i:0
bitmapcachepersistenable:i:1

```

Listing 3.5: Resulting RDP configuration for user "fsc1"

3.2.4 Creating the User Profiles

It is not possible to open the same physical document file from multiple user sessions. Consequently, it was necessary to provide individual document files for all users. For this purpose, individual user profiles were used, which allowed the required documents to be copied to each user's desktop and to launch its associated application.

Creating the user profiles automatically without using a domain controller was the next challenge. To accomplish this, we created the folder %Systemroot% \System32 \Repl \Imports \scripts on the Terminal Server. We then copied the login file Logon.cmd into this folder and created a share with the name netlogon. This was the standard login share. The login script looked as follows:

```

@echo off
if not exist "%userprofile%\Desktop\Terminalservices.txt"
copy c:\Documents\Terminalservices.txt "%userprofile%\Desktop"
if not exist "%userprofile%\Desktop\PAG3.pdf"
copy c:\Documents\PAG3.pdf "%userprofile%\Desktop"
if not exist "%userprofile%\Desktop\FSCDemo.doc"
copy c:\Documents\FSCDemo.doc "%userprofile%\Desktop"
echo Document files copied
sleep 10
start /B notepad.exe "%userprofile%\Desktop\Terminalservices.txt"

```

```
echo Tried to launch Notepad with Terminalservices.txt (4 KB)
sleep 10
start /B "C:\Program Files\Adobe\Acrobat 6.0\Reader\AcroRd32.exe"
"%userprofile%\Desktop\PAG3.pdf"
echo Tried to launch Acrobat Reader with PAG3.pdf (8.2 MB)
sleep 10
start /B "C:\Program Files\Microsoft Office\Office\winword.exe"
"%userprofile%\Desktop\FSCDemo.doc"
echo Tried to launch Microsoft word with FSCDemo.doc (2.3 MB)
pause
```

Listing 3.6: The login script Logon.cmd

3.3 Testing Methodology

For the first two tests Windows Server 2003, Enterprise Edition (one of the Windows Server 2003 family of operating systems) was installed. For the third test, Windows 2000 Server, Enterprise Edition was installed. For both operating systems Office 2000 and Adobe Acrobat Reader 6 were installed.

For each test scenario, the test engineer started groups of ten client sessions on the client machines with an interval of some seconds between each session. To do so, the test engineer needed to create the session by running the script Cr.cmd adding an individual number between 1 and 150 as an attribute for each user. After the initialisation of a session the logon script checked if the documents required for the test scenario were available on the desktop of the user. If not, they were copied there from a standard source. Finally, every ten seconds one of the applications with a corresponding document was started.

- Notepad with Terminalservices.txt
- Adobe Acrobat Reader 6 with PAG3.pdf
- Microsoft Word 2000 with FSCDemo.doc

After the group of ten client sessions was started, a short stabilisation period was observed in which no additional sessions were started. After the stabilisation period, the test engineer started the next user sessions on another client machine; this prevented any interference between each group of ten client sessions.

Each user session showed a memory footprint of approx. 60 megabytes when all applications were launched.

During the test the following performance counters were monitored and analyzed:

- Memory: Available MBytes
- Memory: Free System Page Table Entries
- Memory: Page Faults/sec
- Memory: Pages/sec

- Memory: Pool Nonpaged Bytes
- Memory: Pool Paged Bytes
- Physical Disk: Avg. Disk Queue Length
- Processor: % Interrupt Time
- Processor: % Processor Time
- Processor: Interrupts/sec
- Server: Pool Nonpaged Bytes
- Server: Pool Paged Bytes
- System: Context Switches/sec
- System: Processes
- System: Processor Queue Length
- Terminal Services: Active Sessions

The analyses in the following pages show the results of some of these performance counters.

4 Analysis of the Results

4.1 Windows Server 2003 with HyperThreading turned on

The first test covered Windows Server 2003 with HyperThreading turned on by the appropriate BIOS settings. During one hour and 10 minutes 100 users were logged in using the testing methodology described above. The Terminal Server system on a hardware platform with eight physical CPUs and HyperThreading turned on showed increasing symptoms of heavy load. After 100 users were logged in, no more user logins were performed for 15 minutes. Still the CPU load showed a permanent load of 20% for all CPUs.

When 20 additional users were logged in, the used physical memory was beyond four gigabytes of memory. Moreover, the system showed substantial saturation effects. If a new application was started in the administrator's session, it took more than 60 seconds until the initial application window was displayed.

The following image shows the number of active Terminal Server sessions over time. Each unit on the x axis represents 5 seconds (1,000 units = 5,000 seconds = 83 minutes = 1 hour and 23 minutes).

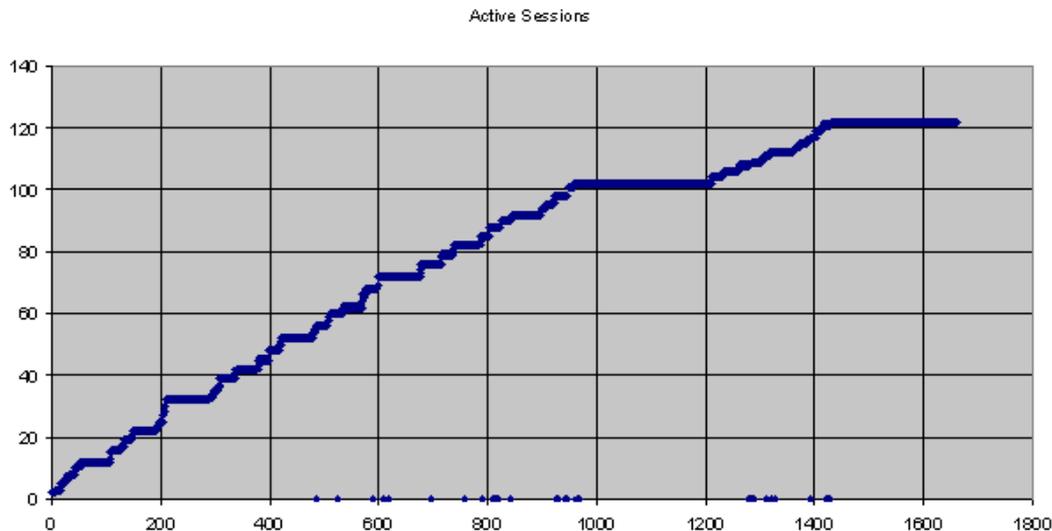


Figure 4-2: Active Terminal Server sessions during the first test

During the test, the amount of Committed Bytes increased from a couple of hundred megabytes to six gigabytes. Committed Bytes is the amount of committed virtual memory, in bytes. Committed memory is the physical memory which has space reserved on the disk paging file(s).

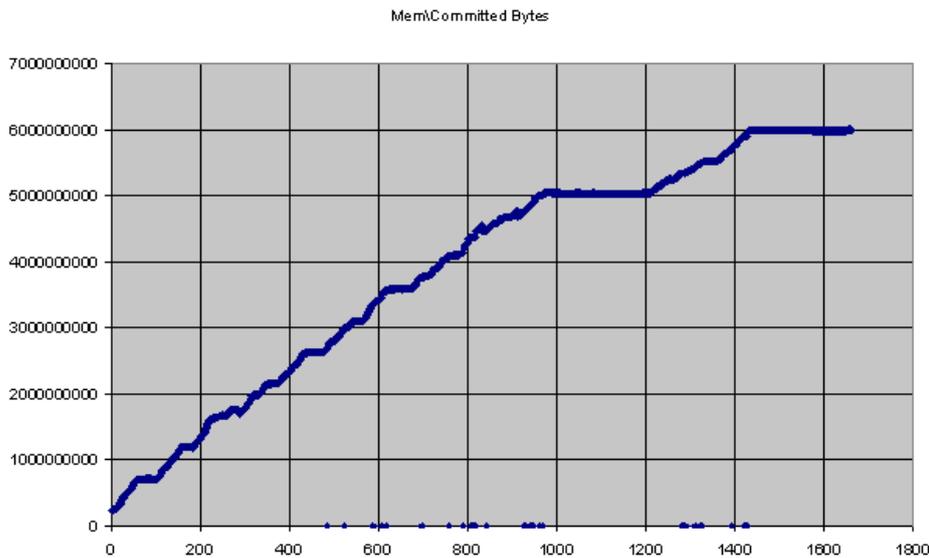


Figure 4-3: Committed Bytes during the first test

Within the same period, the amount of available physical memory dropped from almost eight gigabytes to approximately three and a half gigabytes. Available Bytes is the amount of physical memory immediately available for allocation to a process or for system use.

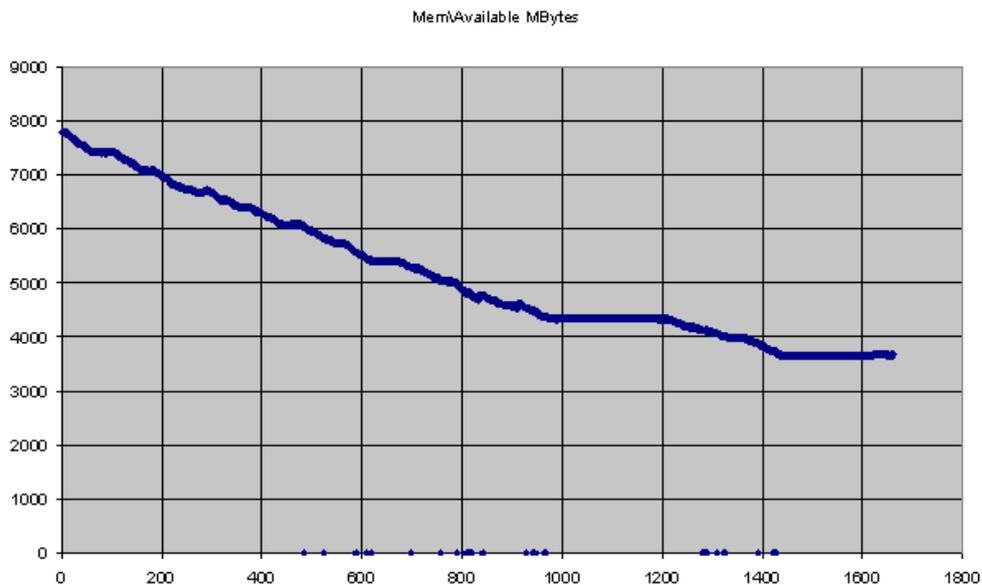


Figure 4-4: Available megabytes physical of physical memory.

Between 1,000 and 1,200 and between 1,400 and 1,650 on the x axis, no new user logged in. The users of the active sessions did not interact with their applications.



However, average CPU load stayed at a rather high level of 20% during this time. Please keep in mind that we are talking about logical 16 CPUs (= eight Hyperthreaded physical CPUs) during this test.

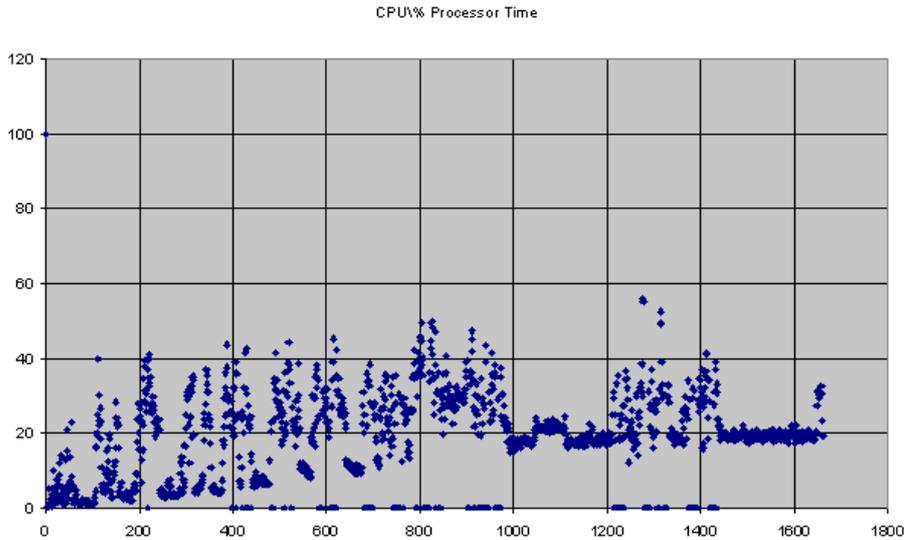


Figure 4-5: Average processor time in %.

Another exceptional result was the increase of the Processor Queue Length during the test. It did not stabilise even during the periods of inactivity. Therefore, we have a strong indication that the processors were consuming a lot of resources while they tried to synchronise their tasks. This seemed to be directly related to the number of processes running on the Terminal Server platform.

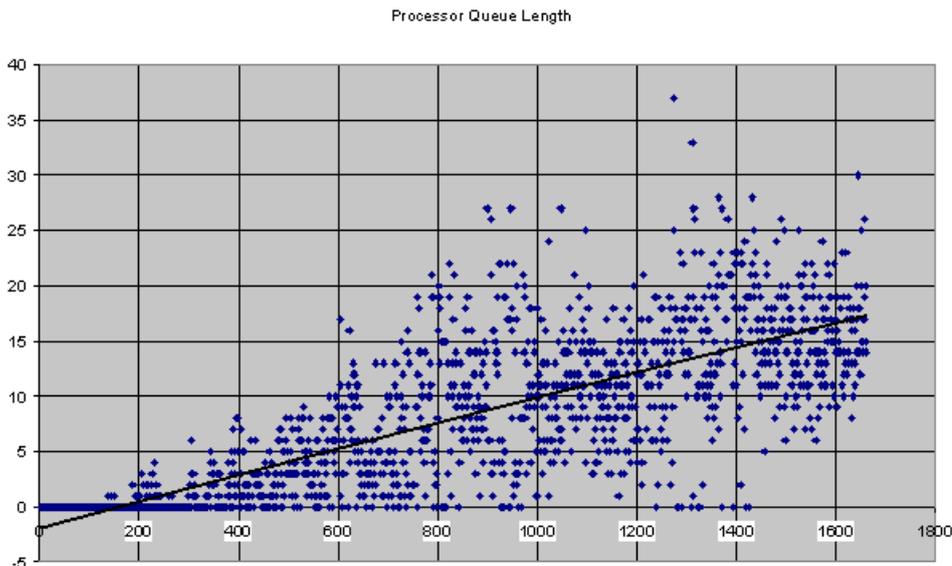


Figure 4-6: Processor Queue Length

The last figure regarding this test shows the Task Manager in the final test phase.

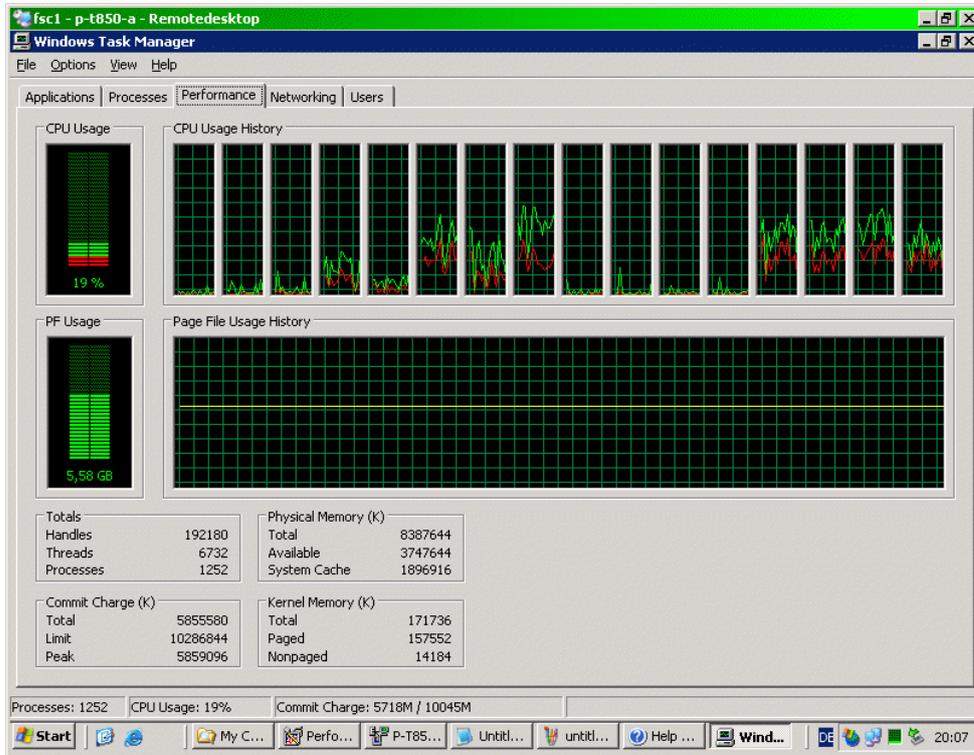


Figure 4-7: Task Manager in the final phase of the test with 120 test users and two administrators logged in

4.2 Windows Server 2003 with HyperThreading turned off

The second test was done on the same hardware platform with eight physical CPUs, but with HyperThreading turned off. This time, it only took just over one hour until 100 users were logged in. Again the Terminal Server system showed increasing symptoms of heavy load, but not as significant as during the first test. It took about 40 more minutes until an additional 30 users were fully logged in.

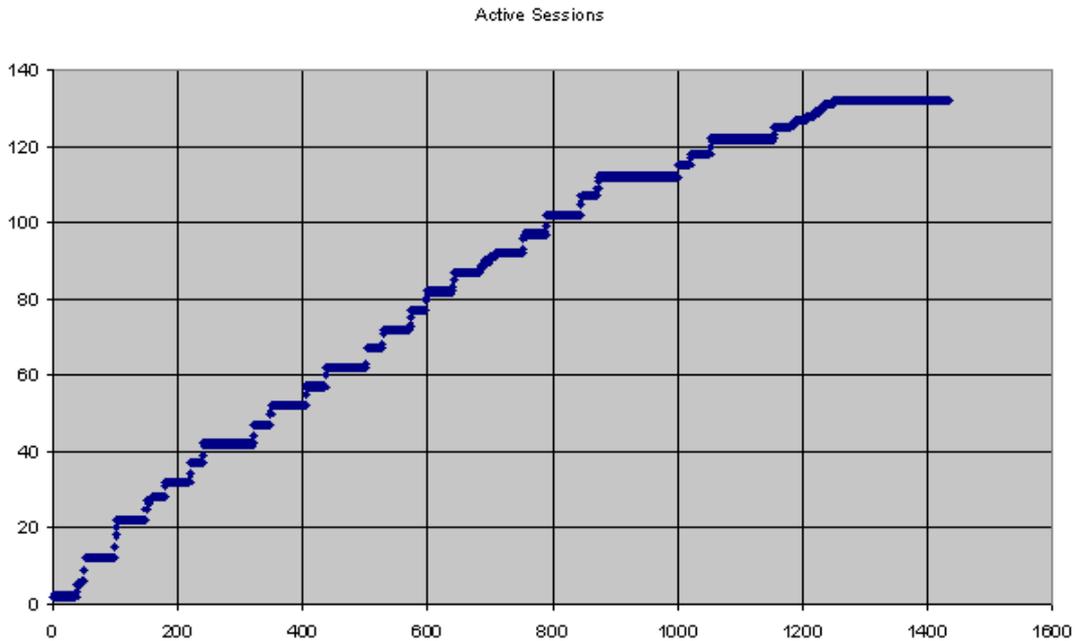


Figure 4-8: Active Terminal Server sessions during the second test. Each unit at the x-axis represents 5 seconds (1,000 units = 5,000 seconds = 83 minutes = 1 hour and 23 minutes)

However, the CPU load still showed a permanent load of more than 20% for all CPUs at the end of the test. Memory consumption was at about the same level as during the previous test and the Processor Queue Length was even higher; it came to an average of 45 by the end of this test.

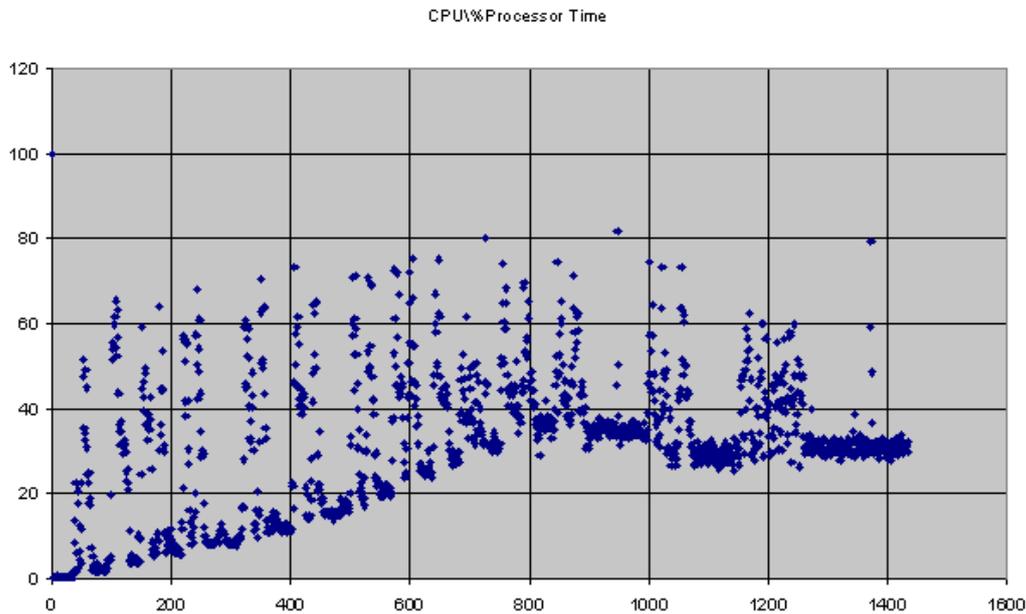


Figure 4-9: Average processor time in %

With 130 users logged in, this high-end machine was getting very slow. The test users were not interacting with the applications at any time during the test; the applications were simply occupying memory. System-internal administrative tasks thus accounted for the full processor load. System reaction time to user actions increased noticeably from the time previously measured. Simple dialog and window changes took more than 10 seconds.

At this point we decided to take a closer look at the monitored values of the Context Switches per second. This performance counter measures the combined rate at which all processors on the computer are switched from one thread to another. Context Switches occur when a running thread voluntarily relinquishes the processor, is preempted by a higher priority ready thread, or switches between user-mode and privileged (kernel) mode to use an Executive or subsystem service. It is the sum of Thread\Context Switches per second for all threads running on all processors in the computer and is measured in numbers of switches. There are Context Switch counters on the system and thread objects. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

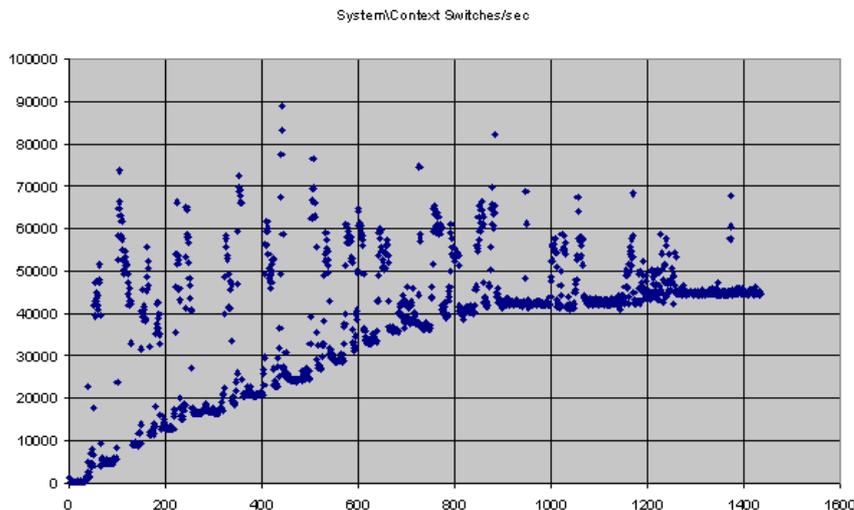


Figure 4-10: Context Switches per Second

The result was a major surprise: more than 40,000 Context Switches per second in average by the end of the test. The CPUs were switching from one thread to another and synchronising their activities.

Immediately prior to the end of the test, the Windows Task Manager showed that the load was not balanced equally between the eight CPUs. It was obvious that adding more CPUs could make the problem even worse.

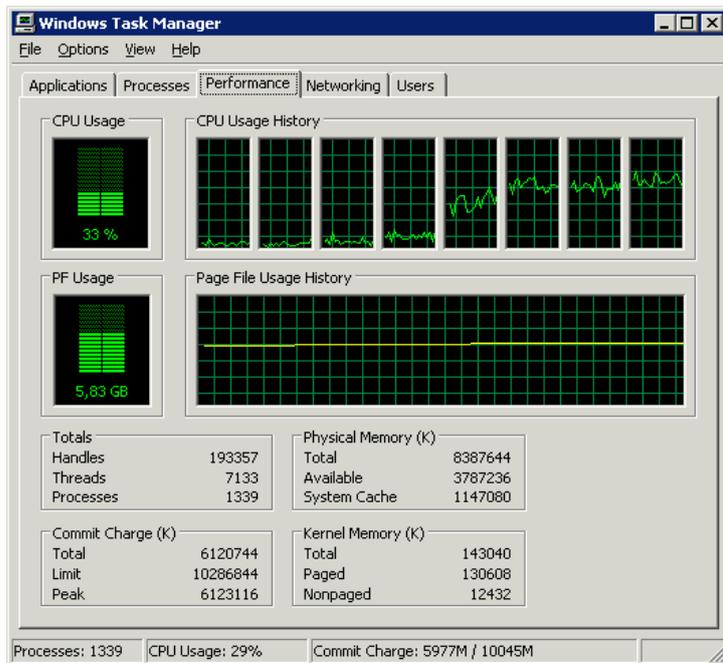


Figure 4-11: Task Manager with 130 users logged in, but none of them working interactively

4.3 Windows 2000 Server with HyperThreading turned off

The last test covered Windows 2000 Server. It illustrated the drastic internal system differences between Windows Server 2003 and Windows 2000 Server. The same hardware platform was used as in the previous tests, but the system was already in saturation after 45 test users were logged in.

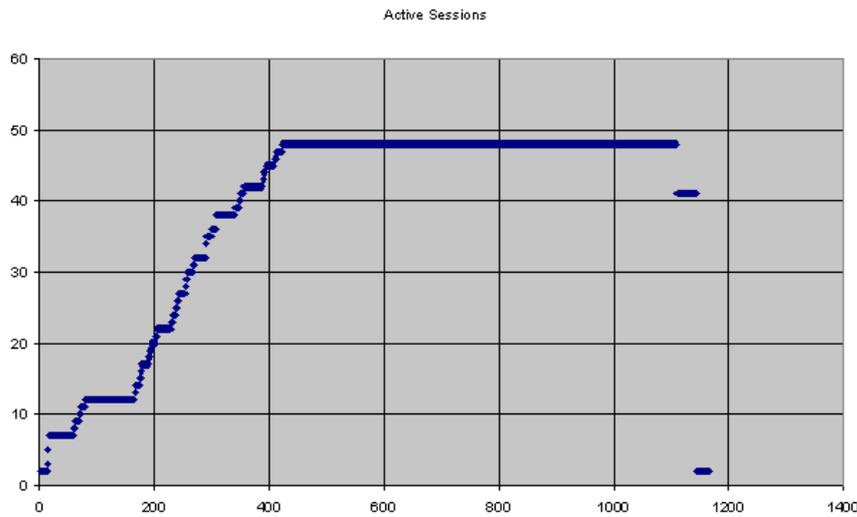


Figure 4-12: Active Sessions during the third test

A very good indicator for the performance deterioration seemed to be the Processor Queue Length. It was growing to an absolutely incredible value of 350. The system was almost at a complete stop and never recovered before the test users were logged out again.

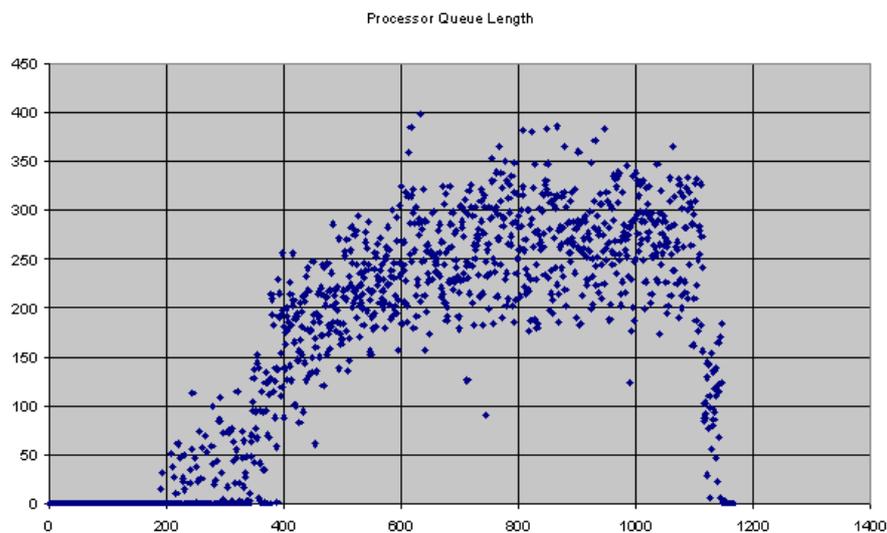


Figure 4-13: Processor Queue Length during the third test

5 Conclusion

The initial purpose of the tests was to investigate the behaviour of large server hardware platforms during massive utilisation through terminal services user sessions. The tests showed that using main memory above the 4 GB barriers posed no problems. However, a limiting factor was the massive context switching of the processors, which occurred after initialising more than 100 user sessions even if the users didn't interact with the applications they had launched.

The third test was a significant indication that Windows 2000 Server may not be ready for high-end hardware platforms where Terminal Services are involved. Windows Server 2003 changed this substantially. However, due to the results from tests one and two, we still cannot recommend using hardware platforms with more than four physical CPUs in a Terminal Server infrastructure. Consequently we recommend the usage of two-way or four-way platforms with up to 4 gigabytes of RAM and an appropriate load balancing solution for use in terminal server scenarios.

6 Appendix

6.1 System Information

System Information written by SystemInfo on Windows Server 2003 with HyperThreading turned on:

```

Created using SystemInfo
120 users logged in via RDP session
Host Name: P-T850-A
OS Name: Microsoft(R) windows(R) Server 2003, Enterprise Edition
OS version: 5.2.3790 Build 3790
OS Manufacturer: Microsoft Corporation
OS Configuration: Standalone Server
OS Build Type: Multiprocessor Free
Registered Owner: fsc
Registered Organization: fsc
Product ID: 69713-OEM-4211904-04011
Original Install Date: 01.08.2003, 15:05:43
System Up Time: 0 Days, 20 Hours, 27 Minutes, 6 Seconds
System Manufacturer: FUJITSU SIEMENS
System Model: PRIMERGY T850
System Type: X86-based PC
Processor(s): 16 Processor(s) Installed.
[01]: x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1599 Mhz
[02]: x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1600 Mhz
[03]: x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1599 Mhz
[04]: x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1599 Mhz
[05]: x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1599 Mhz
[06]: x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1599 Mhz
[07]: x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1599 Mhz
[08]: x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1599 Mhz
[09]: x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1599 Mhz
[10]: x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1599 Mhz
[11]: x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1599 Mhz
[12]: x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1599 Mhz
[13]: x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1599 Mhz
[14]: x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1599 Mhz
[15]: x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1599 Mhz
[16]: x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1599 Mhz

BIOS Version: IBM - 1000
Windows Directory: C:\WINDOWS
System Directory: C:\WINDOWS\system32
Boot Device: \Device\HarddiskVolume2
System Locale: de;German (Germany)
Input Locale: de;German (Germany)
Time Zone: (GMT+01:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna
Total Physical Memory: 8.191 MB
Available Physical Memory: 3.675 MB
Page File: Max Size: 18.237 MB
Page File: Available: 7.874 MB
Page File: In Use: 10.363 MB
Page File Location(s): C:\pagefile.sys
Domain: MYUSERGROUP
Logon Server: \\P-T850-A
Hotfix(s): 1 Hotfix(s) Installed.
[01]: Q147222

Network Card(s): 3 NIC(s) Installed.
[01]: Intel(R) PRO/1000 XT Server Adapter
Connection Name: Local Area Connection 3
DHCP Enabled: Yes
DHCP Server: 10.0.1.23
IP address(es)
[01]: 10.100.0.10
[02]: Intel(R) PRO/1000 XT Server Adapter
Connection Name: Local Area Connection 2
DHCP Enabled: Yes
DHCP Server: 10.0.1.23
IP address(es)
[01]: 10.100.0.9

```

```
[03]: Broadcom NetXtreme Gigabit Ethernet
      Connection Name: Local Area Connection
      DHCP Enabled:    No
      IP address(es)
      [01]: 10.0.1.49
```

System Information written on Windows 2000 Server with HyperThreading turned off:

```
System Information report written at: 06.08.2003 18:00:00
[System Information]
```

[Following are sub-categories of this main category]

[System Summary]

Item	Value
OS Name	Microsoft Windows 2000 Advanced Server
Version	5.0.2195 Service Pack 4 Build 2195
OS Manufacturer	Microsoft Corporation
System Name	P-T850-A
System Manufacturer	FUJITSU SIEMENS
System Model	PRIMERGY T850
System Type	X86-based PC
Processor	x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1599 Mhz
Processor	x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1599 Mhz
Processor	x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1599 Mhz
Processor	x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1599 Mhz
Processor	x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1599 Mhz
Processor	x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1600 Mhz
Processor	x86 Family 15 Model 1 Stepping 1 GenuineIntel ~1600 Mhz
BIOS Version	IBM BIOS Ver 1.08
Windows Directory	C:\WINNT
System Directory	C:\WINNT\system32
Boot Device	\Device\Harddisk0\Partition2
Locale	Germany
User Name	P-T850-A\Administrator
Time Zone	W. Europe Daylight Time
Total Physical Memory	7.863.352 KB
Available Physical Memory	7.511.024 KB
Total Virtual Memory	17.621.612 KB
Available Virtual Memory	17.091.632 KB
Page File Space	9.758.260 KB
Page File	C:\pagefile.sys

6.2 Performance Counters Analysed

Terminal Services\Active Sessions: Total number of Terminal Services sessions.

Memory\Committed Bytes: Committed Bytes is the amount of committed virtual memory, in bytes. Committed memory is the physical memory that has space reserved on the disk paging file(s). There can be one or more paging files on each physical drive. This counter displays the last observed value only; it is not an average.

Memory\Available MBytes: Available MBytes is the amount of physical memory, in megabytes, immediately available for allocation to a process or for system use. It is equal to the sum of memory assigned to the standby (cached), free and zero page lists.

Processor\% Processor Time: % Processor Time is the percentage of elapsed time that the processor spends to execute a non-Idle thread. It is calculated by measuring the duration of the idle thread is active in the sample interval, and subtracting that time from interval duration. (Each processor has an idle thread that consumes cycles when no other threads are ready to run). This counter is the primary indicator of processor activity, and displays the average percentage of busy time observed during the sample interval. It is calculated by monitoring the time that the service is inactive, and subtracting that value from 100.

System\Processor Queue Length: Processor Queue Length is the number of threads in the processor queue. Unlike the disk counters, this counter shows ready threads only, not threads that are running. There is a single queue for processor time even on computers with multiple processors. Therefore, if a computer has multiple processors, you need to divide this value by the number of processors servicing the workload. A sustained processor queue of less than 10 threads per processor is normally acceptable, dependent of the workload.

System\Context Switches/sec: Context Switches/sec is the combined rate at which all processors on the computer are switched from one thread to another. Context switches occur when a running thread voluntarily relinquishes the processor, is pre-empted by a higher priority ready thread, or switches between user-mode and privileged (kernel) mode to use an Executive or subsystem service. It is the sum of Thread\Context Switches/sec for all threads running on all processors in the computer and is measured in numbers of switches. There are context switch counters on the System and Thread objects. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.