Dr Tritsch
remote desktop services and desktop virtualization

Microsoft Most Valuable Professional

Technical Whitepaper

Scalability of Office Suites on Terminal Servers

Comparative test between OpenOffice.org 2.1, Microsoft Office 2003, and Microsoft Office 2007

Version 1.1, April 2007

Dr. Bernhard Tritsch

Web: http://drtritsch.com

Twitter: @drtritsch

1 Contents

2 Introduction

The goal of this test was to obtain a reproducible and independent assessment for the implementation of OpenOffice 2.1 on terminal servers in comparison with Microsoft Office 2003 and Microsoft Office 2007. The functionality and user-friendliness of one or another Office Suite did not take the foreground of this assessment. Much more importantly, the results of the test methodology applied should provide an indication concerning the number of users who are simultaneously logged on to and can work with the respective Office Suite on terminal server platforms.

2.1 Summary of the Test Results

The results show that roughly 30 to 40% more users can sign on to the platform with Microsoft Office 2003 than with OpenOffice.org 2.1 where the hardware is completely identical and the terminal server on Windows Server 2003 SP1 is configured the same way. Even for terminal servers with Microsoft Office 2007, one can expect that the system resources can be used roughly 20% more efficiently than is possible on a system with OpenOffice.org.

For the 32-bit Windows server with OpenOffice.org 2.1, the hard drive is drawn upon increasingly when there are more than 50 users due to the main memory used during the testing procedure; this increase in turn leads to longer waiting times on the system for individual users. A similar pattern emerges for the 32-bit server with Microsoft Office 2003, but only with more than 100 users, while the same may be said for Microsoft Office 2007 with 80 users or more.

One of the main reasons behind the differences exhibited in the performance of the different Office Suites has to do with the specific memory working set used by the associated applications. The different ways in which page files are used play a significant role in these differences. During the memory management of the operating system for both versions of Microsoft Office, data is often paged too early, whereas this occurs much later with OpenOffice. There are clear differences between the architecture of the run-time environments in Microsoft Office with its standard Windows applications and those environments in OpenOffice with container applications serving as watchdogs for all components.

Servers with the 64-bit variants of Windows Server 2003 SP1 were used as a reference for identical tests on OpenOffice.org 2.1, Microsoft Office 2003, and Office 2007. Observations of the use of memory and process resources could be verified here.

3 Test Procedure and Measuring Methods

3.1 Systematic

The performance of load simulations under standardized conditions is often designated as “benchmarking“; however, no generally accepted standardized test procedure exists for applications and terminal servers. For this reason, the term “load test” should be used for the analyses referenced here.

The analyses and results of this load test should not be interpreted in isolation. The selection of the applications activated in terminal server sessions during the test ensured that the load generated on the server resources would be reproducible. In addition, the test possesses a static quality because users only logged on to the system, started three applications and their accompanying documentation, remaining inactive for the rest of the test. This feature allows for the generation of reproducible results, on the one hand, but on the other hand the results can vary depending on the applications activated as well on the documents used.

Nevertheless, this type of test provides clear results that emerge from the comparison of multiple pre-defined scenarios. The absolute numbers, such as the maximum number of user sessions, are therefore less relevant; the relative relationships between two scenarios are much more relevant than the numbers taken individually. The result of this test should above all provide greater insight as to which scenario requires proportionately more or less system resources in comparison with other scenarios.

3.2 Tools Implemented

Unfortunately, there is no standardized and easily reproduced procedure that can be used to test the performance of applications on terminal servers. Manufacturers such as Microsoft and Citrix do provide tools for this purpose, although they have not successfully established any generally recognized system for comparison.

Only the tools that were easiest to use were implemented for the test described here. This facilitates the reproducibility of the reference scenarios and results.

- vRD Load Edition, a tool for the timed control of log-on sequences by test users on a terminal server and for the representation of all related sessions.
- Scripts on the basis of the Cmd.exe command line interpreters including the Sleep.exe tool from the Windows Server 2003 Resource Kit.
- Scripts on the basis of the Windows Script Host, which can be executed from the Windows command line with Cscript.exe.
- Msinfo32.exe to determine the exact hardware configuration of the server.
- The Pslist.exe Information Lister from Sysinternals to list all started processes at the end of a test sequence.

4 Performing the Test

4.1 Infrastructure Used

The test described here was performed in various instances from February to April, 2007 at the Microsoft GmbH location in Unterschleissheim.

Multiple Hewlett Packard ProLiant BL20p G3 servers, each with two dual core processors (2.8 GHz) and 4 or 8 GB RAM, were made available all tests. These servers were operated in slightly different configurations:

- OpenOffice.org 2.1, Microsoft Office 2003, and Microsoft Office 2007 under Microsoft Windows Server 2003, Enterprise Edition, Service Pack 1, 32-bit version, English, 4 GB RAM
- OpenOffice.org 2.1, Microsoft Office 2003, and Microsoft Office 2007 under Microsoft Windows Server 2003, Enterprise Edition, Service Pack 1, x64 version, English, 8 GB RAM

The initial size of the page file was pre-configured to 2 GB on all servers.

Multiple PCs running under Windows Server 2003 or Windows XP, which were connected via a separate network to the servers to be tested, were used to generate the load. The measurement results are, however, independent of the absolute performance capacity of the hardware platforms for load generation. No performance limits were reached during the test either in terms of storage or processor resources.

4.2 Preparations

4.2.1 Installation of the Basic Platform

All servers are outfitted with a pre-installed English-language operating system. The terminal services were activated in the mode of an application server (terminal server). Office 2003 or 2007, respectively, was already pre-installed on the servers for the test on Microsoft Office. Special attention was paid to ensure that the installation mode was configured during the installation and that the common transform files for terminal servers from the corresponding Microsoft Office Resource Kits were used. Additional optimizations were not performed on the operating system or Microsoft Office.

120 local user accounts were set up with the following script. These accounts were used in the context of the test for interactive logons. All user accounts must be added to the “Users” and “Remote Desktop Users” local groups.

```
nMaxUsers = 121
strPassword = "OO2007_!"
```

```
Set objComputer = CreateObject("Shell.LocalMachine")
strComputer = objComputer.MachineName
Wscript.Echo "Computer name: " & strComputer
Wscript.Echo "Multiple users enabled: " & objComputer.IsMultipleUsersEnabled
Wscript.Echo "Remote connections enabled: " & objComputer.IsRemoteConnectionsEnabled

nCounter = 1

Do While nCounter < nMaxUsers
  strUser = "v" & nCounter

  Set colAccounts = GetObject("WinNT://" & strComputer & "")

  Set objUser = colAccounts.Create("user", strUser)
  objUser.SetPassword strPassword
  objUser.SetInfo

  wscript.Echo "User account " + strUser + " created"

  Set objNewUser = GetObject("WinNT://" & strComputer & "/" & strUser & ",user")

  Set objGroup = GetObject("WinNT://" & strComputer & "/Remote Desktop Users,group")
  objGroup.Add(objNewUser.ADsPath)
  wscript.Echo "Added User account " + strUser + " to Remote Desktop Users group"

  Set objGroup = GetObject("WinNT://" & strComputer & "/Users,group")
  objGroup.Add(objNewUser.ADsPath)
  wscript.Echo "Added User account " + strUser + " to Users group"

  nCounter = nCounter + 1
Loop
```

Bginfo, a Microsoft Sysinternals tool, was pre-installed on all servers. The automatic start of Bginfo was removed from the start-up sub-item of the start menu for all users in order to avoid negative side-effects on the test system.

During the test, no screen savers should be activated during the user sessions. For this reason, the screen savers for all user sessions were deactivated according to local policy. The accompanying configuration information may be found at Local Computer Policy | User Configuration | Administrative Templates | Control Panel | Display | Screen Saver.

4.2.2 Installation and Configuration of the Applications

Transform files were used to produce installations of Microsoft Office 2003 and Microsoft Office 2007 which were specially optimized for terminal servers. Transform files are a component of the Resource Kits for Office 2003 and Office 2007.

In order to stop the Configuration Wizard from opening during the first start up of an application on Microsoft Office 2003, Word, Excel, and PowerPoint will be opened for the first time in an administrator session, which was previously activated in the installation mode. In this manner, all associated configurations access the shadow-key section of the registry and will be passed on to all other user sessions. Automatically transferring the registry entries in the shadow-key section to a new user's profile prevents the Configuration Wizard from starting the first time that an application is opened in Office 2003. There is a corresponding administrative template for Microsoft Office 2007 that achieves this same result.

The installation of OpenOffice.org 2.1 on the server was quickly and easily configured in installation mode with an MSI file after unpacking the installation package on the local file system. The space occupied on the hard drive amounted to about 250 MB. There were color errors on the RDP 6 Client where the color depth was limited to 16 bit, but these errors were easily corrected by selecting a greater color depth.

The user-specific information from OpenOffice is stored in a folder hierarchy in the user profile instead of in the registry. The configuration files are based on XML, so they can also be modified as needed with a simple text editor. Upon the first start-up of an OpenOffice application by a user, a series of basic settings must be made with a configuration wizard. This can be effectively avoided during a test if the entire directory hierarchy of the XML configuration files is copied to the user profile before starting OpenOffice for the first time. The hierarchy of a user's XML configuration files is available as source material for the configuration, thus pre-emptively answering all of the configuration wizard's queries.

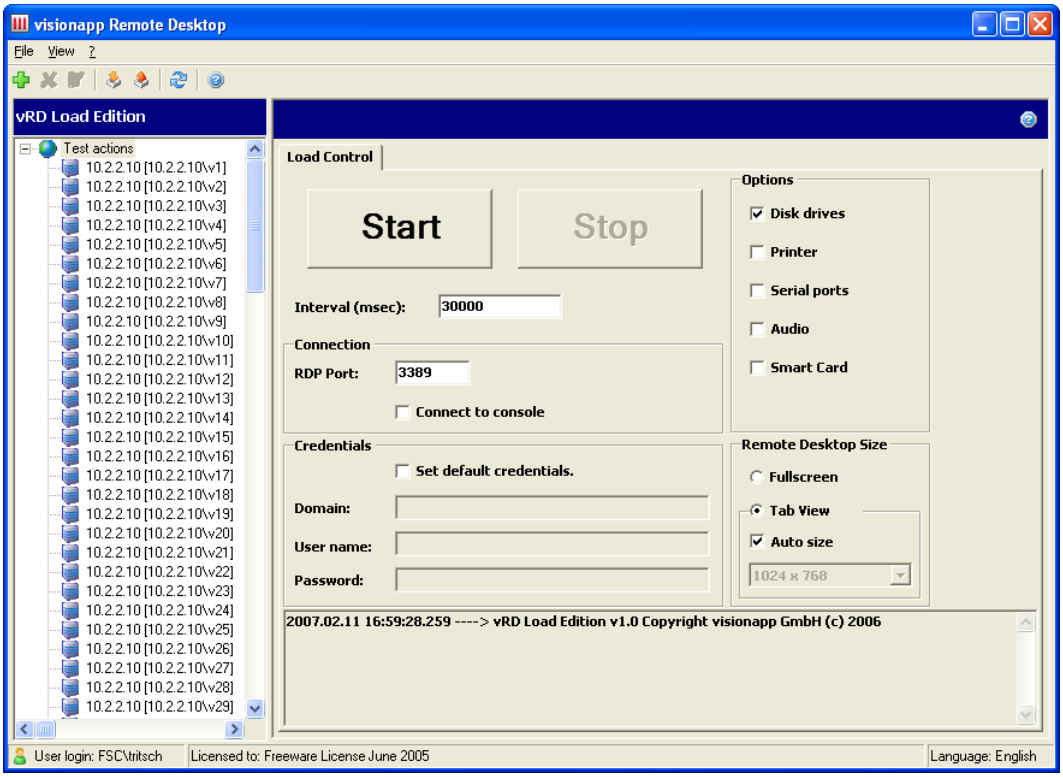
During the installation of OpenOffice, the activation of the accompanying runtime environment is anchored in the start-up folder. This accelerates the start-up of the actual OpenOffice applications. However, this early start of the runtime environment for each new session does not permit pre-configuration by means of copying the XML configuration files. Consequently, the option to activate the runtime environment was removed from the start-up folder for the purposes of the test. The runtime environment only started when the OpenOffice application was opened for the first time.

4.3 Measurements

4.3.1 Clients – Load Generation

A server is allocated a functional PC with the vRD tool to generate loads. The vRD was set up in such a way that a new user can log on to the server via RDP every 30 seconds. It was therefore possible for 120 users to sign on to the server within an hour.

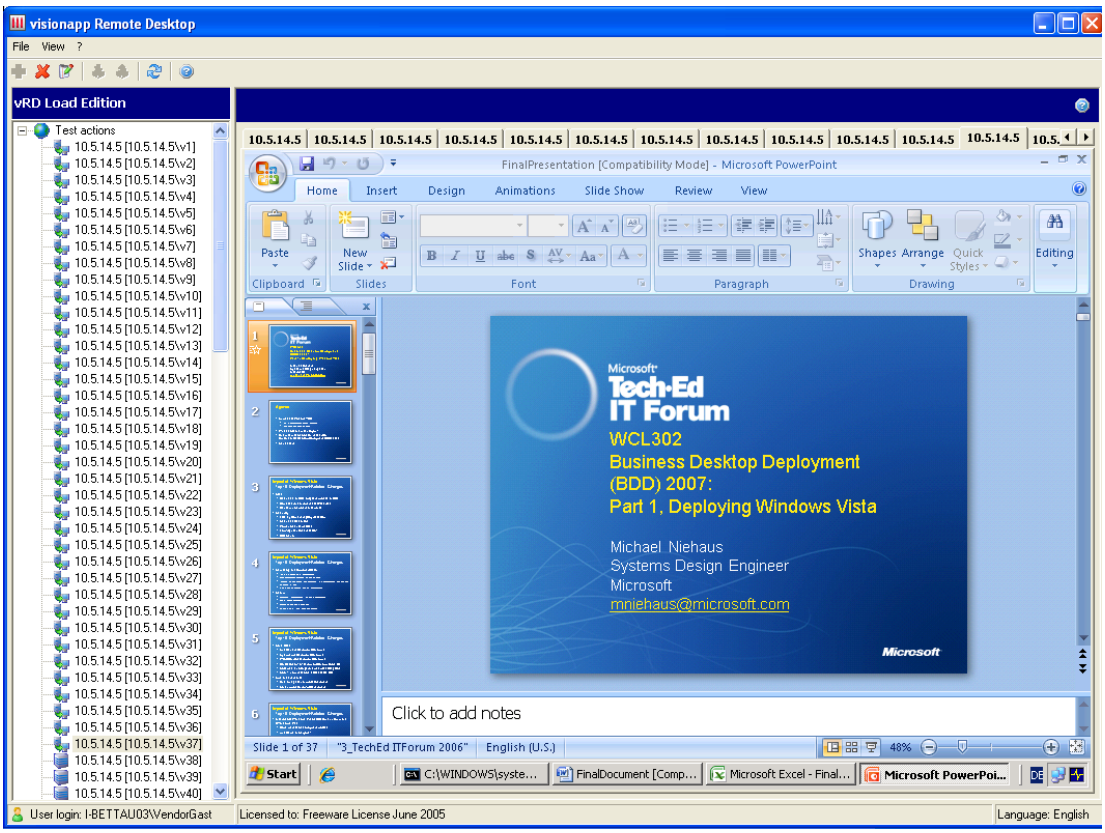
The vRD configuration can be seen in the following image.



A configuration file containing the log on information of 120 users was stored on each client and loaded into vRD for the test's execution. A sample configuration file for three users can be seen in the following:

```
<?xml version="1.0" encoding="utf-16"?>
<vRDLoadConfigurationFile>
  <Testcase ServerName="10.2.2.10">
    <Domain>10.2.2.10</Domain>
    <User>v1</User>
    <Password>OO2007_!</Password>
  </Testcase>
  <Testcase ServerName="10.2.2.10">
    <Domain>10.2.2.10</Domain>
    <User>v11</User>
    <Password>OO2007_!</Password>
  </Testcase>
  <Testcase ServerName="10.2.2.10">
    <Domain>10.2.2.10</Domain>
    <User>v21</User>
    <Password>OO2007_!</Password>
  </Testcase>
</vRDLoadConfigurationFile>
```

The test was be started by pressing the corresponding button. It was possible to view each new user session in a separate window in vRD at any time.



4.3.2 Servers

Three different documents were opened in the corresponding Office application for each user session during the test. After the users logged on it was necessary to copy the required documents directly to the desktop, if they were not already there. The documents are then opened with the appropriate application. These actions may be executed with a script that is stored in the startup folder.

The log-on script on the terminal server with Microsoft Office 2003 can be seen in the following:

```
@echo off
if not exist "%userprofile%\Desktop\FinalDocument.doc" copy
c:\Install\Documents\FinalDocument.doc "%userprofile%\Desktop"
if not exist "%userprofile%\Desktop\FinalSpreadsheet.xls" copy
c:\Install\Documents\FinalSpreadsheet.xls "%userprofile%\Desktop"
if not exist "%userprofile%\Desktop\FinalPresentation.ppt" copy
c:\Install\Documents\FinalPresentation.ppt "%userprofile%\Desktop"
echo Document files copied

rem Specific extensions here

sleep 20
start /B "C:\Program Files\Microsoft Office\Office11\WinWord.exe"
"%userprofile%\Desktop\FinalDocument.doc"
echo Launching Word - Documents
sleep 20
start /B "C:\Program Files\Microsoft Office\Office11\Excel.exe"
"%userprofile%\Desktop\FinalSpreadsheet.xls"
echo Launching Excel - Spreadsheets
sleep 20
start /B "C:\Program Files\Microsoft Office\Office11\Powerpnt.exe"
"%userprofile%\Desktop\FinalPresentation.ppt"
echo Launching PowerPoint – Presentations
```

Note: This script was used with only slight modification on the servers with Microsoft Office 2007 or on the 64-bit platforms with Microsoft Office. Changes generally related to the installation path of the individual office applications. Due to the static nature of the test methodology presented here, a special adaptation was necessary for Office 2007. A registry setting was required to stop Office 2007 applications from blinking on the menu list after the initial start. Due to the blinking, which also caused unnecessary RDP communication data streams, related system activities were observed, which prevented objective load measurements. Thus, for every terminal server session and before the start of the first accompanying application, Office 2007 had to be brought into a state as if the user had already clicked on the new menu list. This was accomplished by setting the registry entry OfficeMenuDiscovered under HKCU \SOFTWARE \Microsoft \Office \12.0 \Common \General to the value DWORD:1 in the log-on script at the position marked “specific extensions here“ (see appendix).

The documents that were then opened in the log-on script with the related applications had the following characteristics

- FinalDocument.doc opened with Microsoft Word: 3.3 MB
- FinalSpreadsheet.xls opened with Microsoft Excel: 1.2 MB, no embedded macros
- FinalPresentation.ppt opened with Microsoft PowerPoint: 3.4 MB

The following log-on script appeared on the terminal server OpenOffice.org 2.1:

```
@echo off
if exist "%HOMEPATH%\Application Data\OpenOffice.org2\user\config\standard.soc" goto :JUMP

cd "%HOMEPATH%\Application Data"
mkdir OpenOffice.org2
xcopy C:\Install\OOO "%HOMEDRIVE%%HOMEPATH%\Application Data" /s /e
echo OO Profile created

if not exist "%userprofile%\Desktop\FinalSpreadsheet.ods" copy
c:\Install\OOO\Documents\FinalSpreadsheet.ods "%userprofile%\Desktop"
if not exist "%userprofile%\Desktop\FinalDocument.odt" copy
c:\Install\OOO\Documents\FinalDocument.odt "%userprofile%\Desktop"
if not exist "%userprofile%\Desktop\FinalPresentation.odp" copy
c:\Install\OOO\Documents\FinalPresentation.odp "%userprofile%\Desktop"
echo Document files copied

:JUMP
sleep 20
"C:\Program Files\OpenOffice.org 2.1\program\swriter.exe"
"%userprofile%\Desktop\FinalDocument.odt"
echo Launching OO Writer - Documents
sleep 20
"C:\Program Files\OpenOffice.org 2.1\program\scalc.exe"
"%userprofile%\Desktop\FinalSpreadsheet.ods"
echo Launching OO Calc - Spreadsheets
sleep 20
"C:\Program Files\OpenOffice.org 2.1\program\simpress.exe"
"%userprofile%\Desktop\FinalPresentation.odp"
echo Launching OO Impress – Presentations
```

In order to guarantee the possibility of drawing comparisons between the test results, the exact same documents were used for both Microsoft Office and OpenOffice. The documents used for Microsoft Office were opened with the OpenOffice import filter and saved in the corresponding native OpenOffice formats. These converted documents were then used in the test.

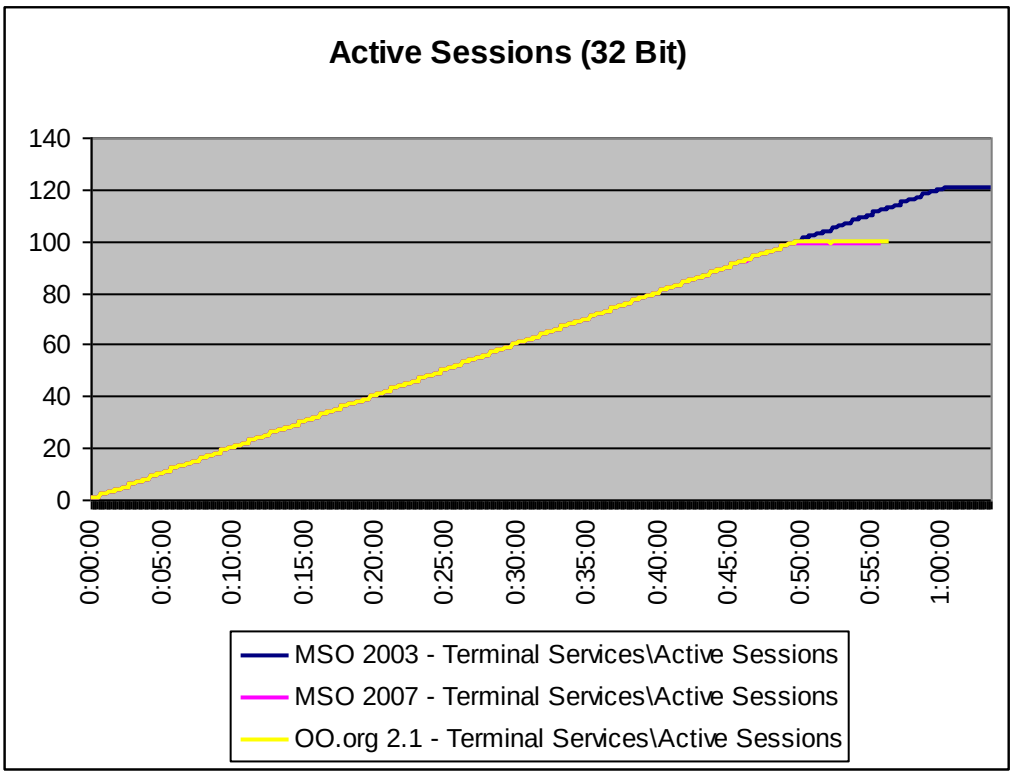
- FinalDocument.odt opened with OpenOffice Writer: 1.4 MB
- FinalSpreadsheet.ods opened with OpenOffice Calc: 290 KB, no embedded macros
- FinalPresentation.odp opened with OpenOffice Impress: 2.5 MB

It could be observed at this point that the document formats for OpenOffice generally required less disk space than those for Microsoft Office.

5 Results

The log-on sequences ran synchronously on all servers, a fact which was guaranteed by the vRD tool on the load generators. The results were therefore easily comparable.

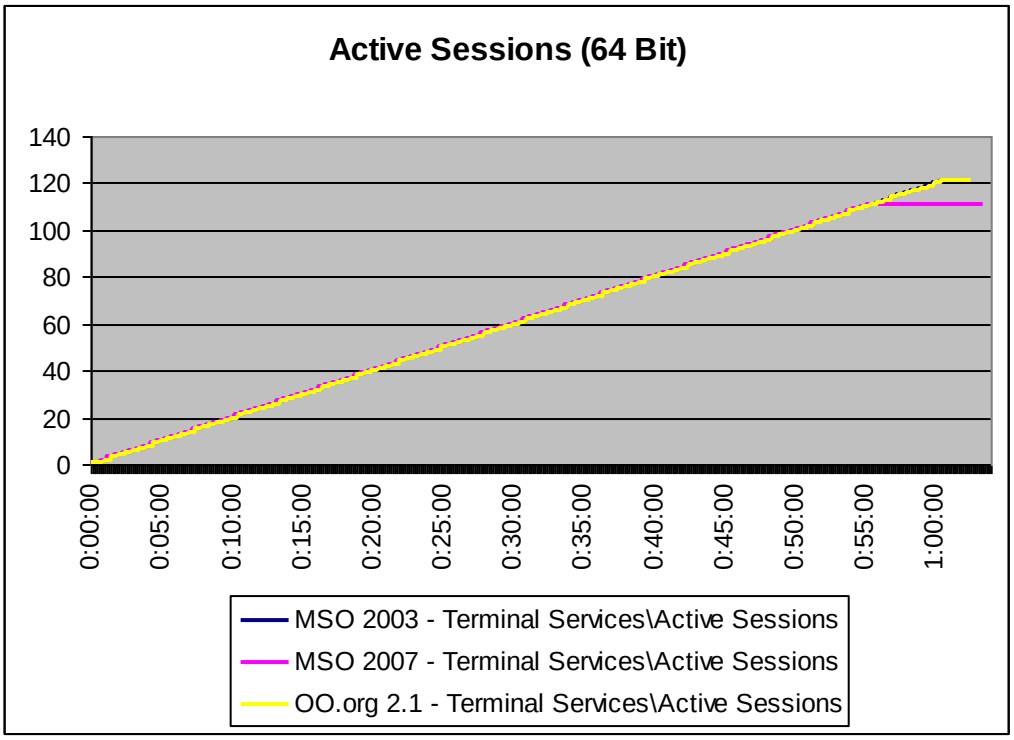
On the 32-bit server with Microsoft Office 2003, it was possible for all 120 users to log on and to start Office applications including Word, Excel, and PowerPoint with the accompanying documentation. This resulted in a total of 1,238 processes, whereby 120 instances each of Word, Excel, and PowerPoint contributed a total of 360 processes.



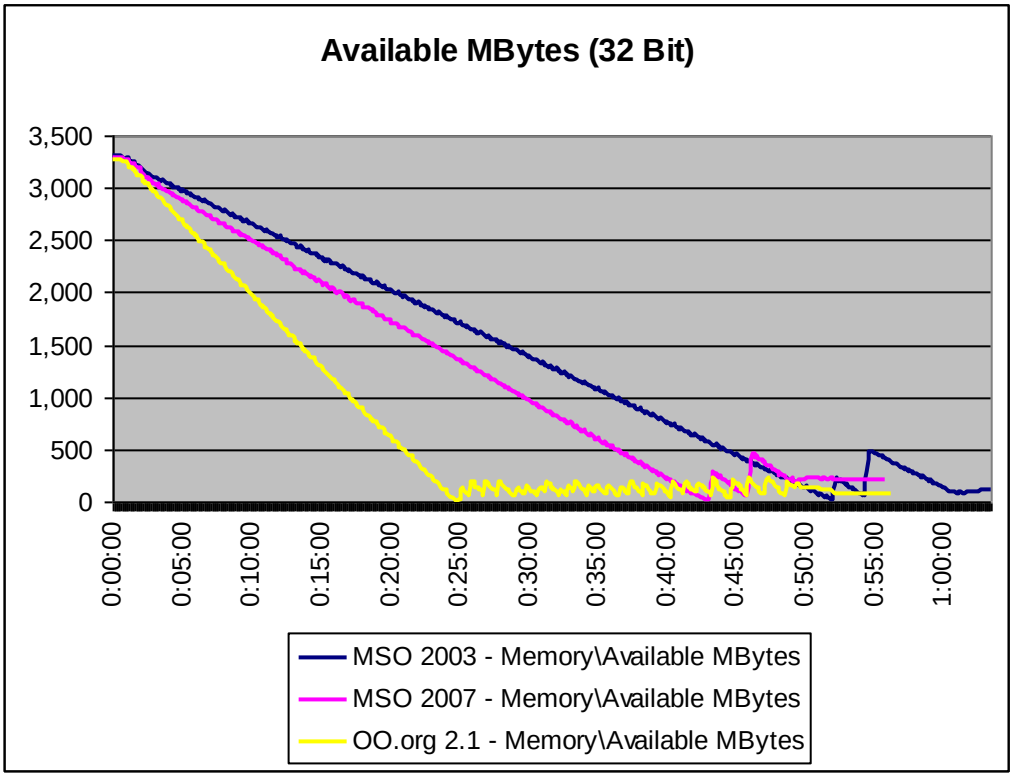
However, on the 32-bit server with Microsoft Office 2007 only 100 users could log on. After that point, the system was saturated and would not permit any further user logons. 97 Word, 97 Excel, and 95 PowerPoint processes were started. At the end of the test, a maximum of 916 processes and 7,256 threads were running on the system.

Similarly, only 100 users could log on to the 32-bit server with OpenOffice.org 2.1 before the system became saturated and stopped permitting log on's. A total of 928 processes on OpenOffice.org 2.1 with Writer, Calc, and Impress could be observed. All OpenOffice applications were activated within the Soffice.bin runtime environment and thus did not form their own processes. The Soffice.exe start process is additionally necessary per user session. 99 instances of both processes, Soffice.exe and Soffice.bin, were observed on the server with OpenOffice.

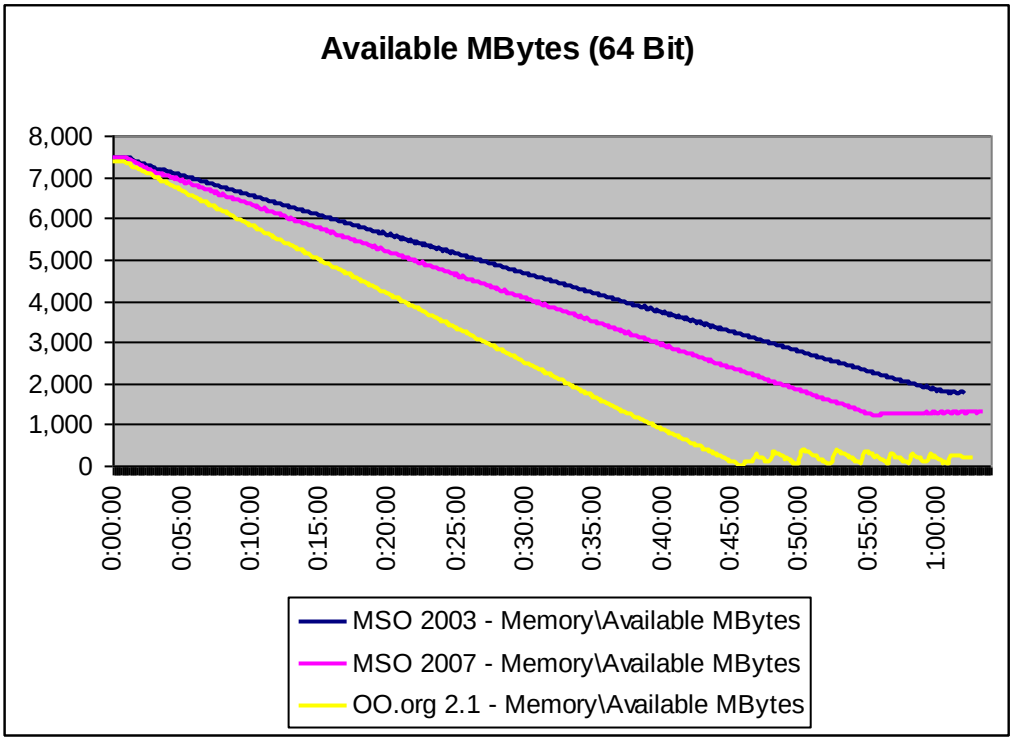
On the 64-bit servers, all 120 users were able to log on with Office 2003 as well as with OpenOffice.org, whereas roughly 115 users were able to log on with Office 2007. The remaining observations relating to log-on mechanisms can be transferred from the 32-bit world to the 64-bit environment.



The differences in saturation behaviour exhibited by the 32-bit servers on the respective office suites are even more pronounced when considering the available memory on the servers over the length of the test period. At the beginning of the 32-bit test runs, roughly 3.3 GB free memory was available. Roughly 100 users could log on to the server with Microsoft Office 2003 and 80 users on Microsoft Office 2007 before the memory started running low, so giving rise to obvious paging activity; the same happened on the server with OpenOffice.org 2.1, but as early as with 50 users.



For the 64-bit servers that had more than double the amount of physical memory with 8 GB, this behaviour was only observed later. The available memory for OpenOffice.org ran low with 90 users, with the same occurring with 110 users on Microsoft Office 2007. The memory with Office 2007 was sufficient for no bottlenecks to appear during the entire test run.



For the 32-bit systems, it was necessary to take a closer look at memory use. The fundamental memory requirements of the applications on the 32-bit servers with Office 2003 were as follows:

- Microsoft Word: The typical working set was 20 MB, the maximum value ran to up to about 26 MB.
- Microsoft Excel: The typical working set amounted to about 6.7 MB, while the maximum value was around 12 MB.
- Microsoft PowerPoint: The typical working set was 1.8 to 2.2 MB, the maximum value was 22 MB.

When the three applications are started in a user session with the corresponding documents, initial memory requirements are 60 MB; this reduces to 30 MB after a relatively short period of time. The first “trimming” of the working set occurs before memory limits are reached.

On the 64-bit system, the Office 2003 applications used the following memory resources: Microsoft Word on average 26-27 MB (27,3 MB initial maximum value), Microsoft Excel on average 13,6 MB (13,6 MB initial maximum value), and Microsoft PowerPoint on average 1.8-2.9 MB (23 MB initial maximum value). The memory requirements per user session amounted to nearly 65 MB at the beginning and dropped to 45 MB with time. The operating-system memory management did not have to start the massive paging at any time.

For the 32-bit server with Microsoft Office 2007, the following memory usage by the applications was observed:

- Microsoft Office Word: The typical working set was 22 MB, the maximum value amounted to roughly 28 MB.
- Microsoft Office Excel: The typical working set was 16 MB, the maximum value amounted to roughly 22 MB lag.
- Microsoft Office PowerPoint: The typical working set was 4 MB, the maximum value amounted to roughly 38 MB.

Initially, the memory requirement of the three Office 2007 applications amounted to just over 80 MB, although this requirement dropped to 40 MB per user session in a short time.

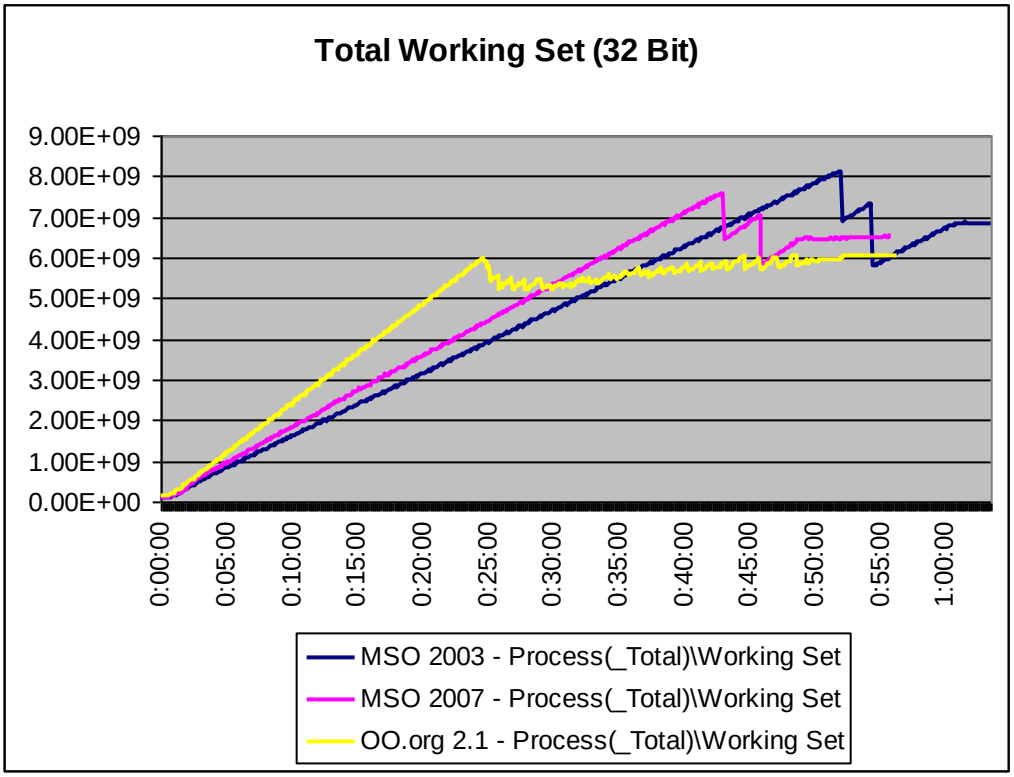
For the 64-bit system, the Office 2007 applications used the following memory resources: Microsoft Word on average 26 MB, which also represented an initial maximum value, Microsoft Excel on average 23,8 MB (24 MB initial maximum value), and Microsoft PowerPoint on average 3,8-7,0 MB (45 MB initial maximum value). The initial memory requirement therefore amounted to nearly 95 MB per user session and was then reduced to 55 MB. In addition, the Splwow64 process contributed an extra 5 MB for each user session whereby the runtime environment supported 32-bit processes on the 64-bit platforms. The 64-bit platform with 8-GB RAM did not reach its limits with this memory requirement even after 100 users had signed on.

For OpenOffice.org, the situation at the processor level was a bit different, especially because the individual applications were not directly distinguishable by their memory footprint. All applications were started in the OpenOffice runtime environment Soffice.bin. The following values resulted on the 32-bit platform:

- Soffice.bin: The typical working set amounted to 30 MB, although the maximum value reached 84 MB.
- Soffice.exe: The typical working set was only 200 kB and the accompanying maximum value was 2 MB.

The maximum value of the three OpenOffice applications along with their accompanying documents therefore amounted to nearly 90 MB per user session. Obviously an operating system has much more difficultly to manage the working sets for application components which are encapsulated in a runtime environment.

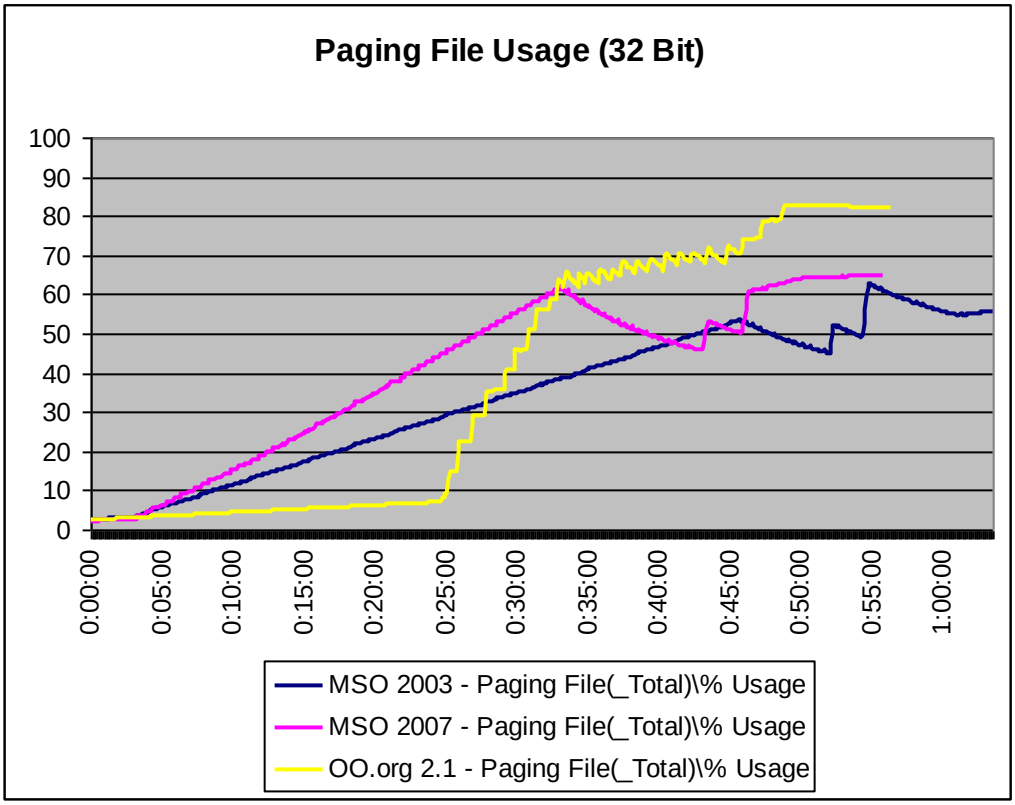
With the 64-bit variants of the Windows Server 2003, the typical working set of Soffice.bin was 78 MB; the maximum value amounted to 87 MB. Soffice.exe merely added a working set of 200 kB and a maximum value of just 3 MB.



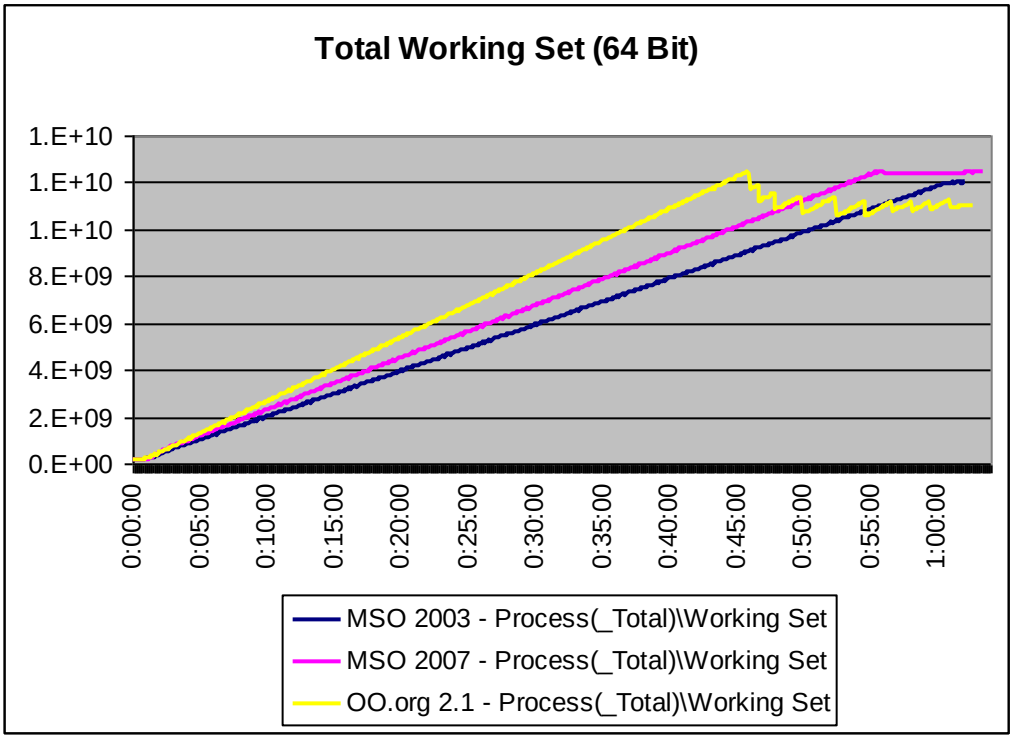
The graph quite clearly indicates that the Windows operating system can manage its memory more efficiently on the 32-bit server with Office 2003 and Office 2007. “Trimming” of the working set when critical limits are reached, evidenced by the characteristic sawtooth pattern, demonstrates the optimization of memory as used by each individual process. The optimization of memory size results in freed memory, which can be used by new processes. The typical quantity of memory that can be freed on servers with Microsoft Office amounts to about 1 GB per sawtooth.

On the 32-bit server with OpenOffice, the amount of working sets for all processes rises faster than with Microsoft Office; the saturation point is reached with just 50 users. The memory must be massively paged to the hard drive, which leads to additional system activities. For the 32-bit server with Microsoft Office 2003, similar behavior can be observed with more than 100 user sessions, and for Office 2007, similar behavior can be observed with more than 80 user sessions. When the number of 110 users is reached—which is no problem for the system on Office 2003--the 32-bit server with OpenOffice, and even with Office 2007, has to allocate all of its resources and cannot permit any additional user log on’s.

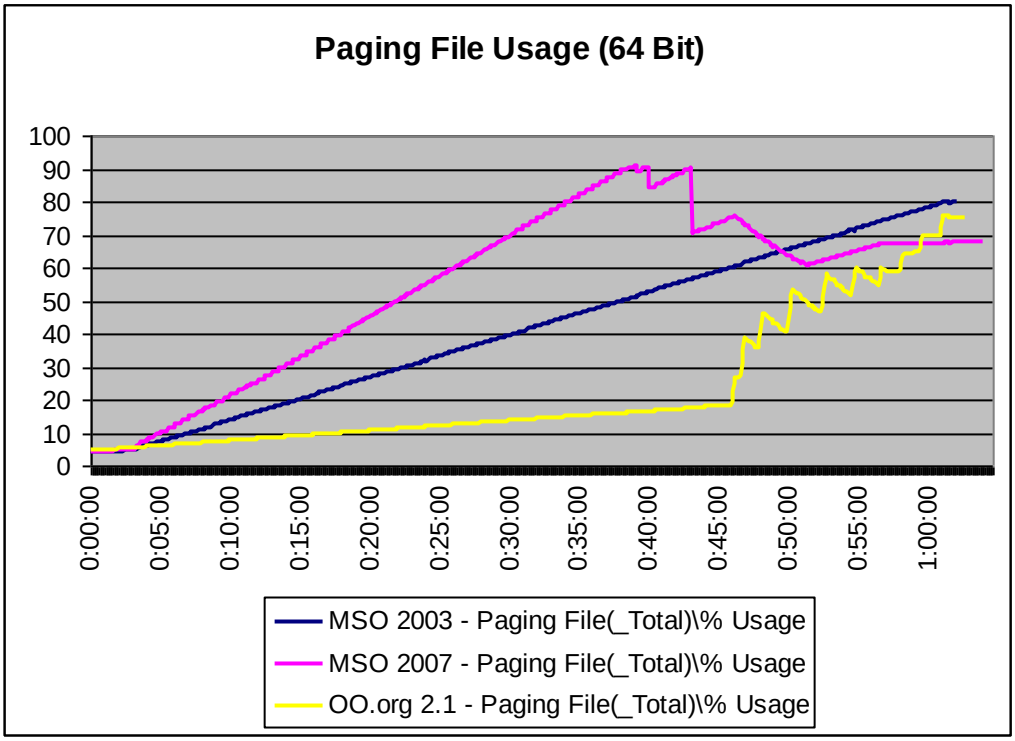
This system can also be observed in the use of page files. While the server with OpenOffice hardly uses page files at all initially, Microsoft Office permanently pages data to this file.



The evaluation of the 64-bit systems also displays a clear working set increase over all processes. Due to the higher amount of physical memory, none of the typical sawtooth patterns can be viewed for the massive trimming of the working set by memory management. Only with OpenOffice.org does the system attempt to optimize memory usage in smaller intervals.

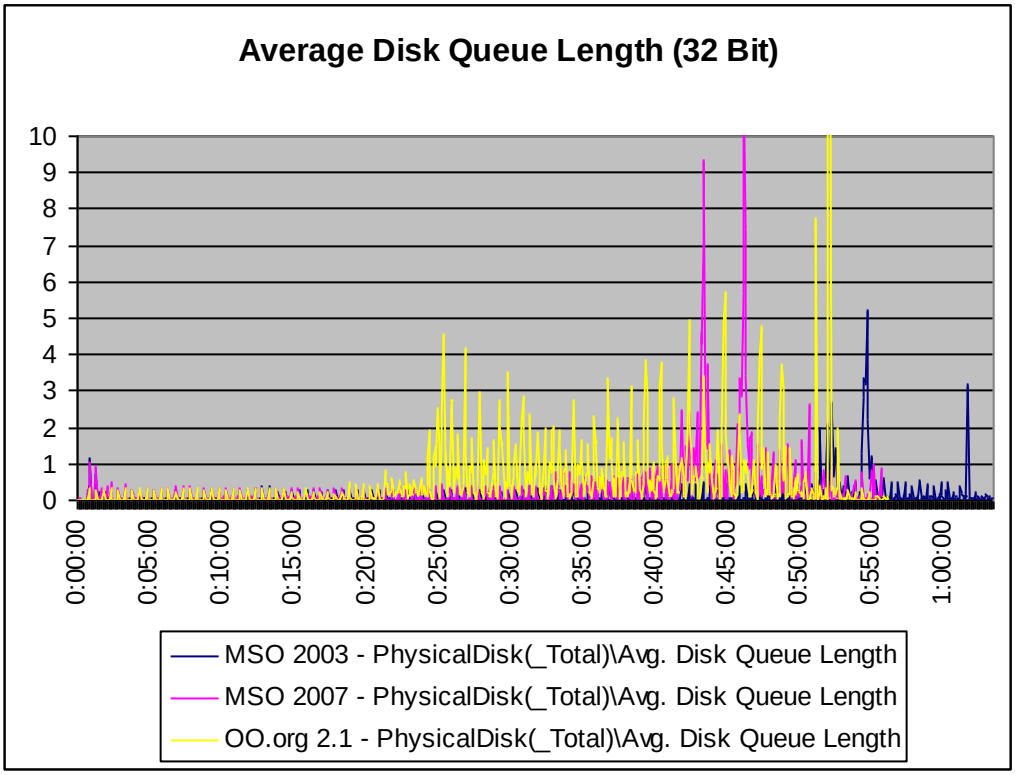


Access to the page file follows a similar pattern to the 32-bit systems. The memory management of the server with OpenOffice first begins to actively use the page file when saturation limits are reached. The server with Microsoft Office on the other hand stores process data from the very beginning, thus raising the requirements for the size of the page file. This in turn can trigger saturation effects.

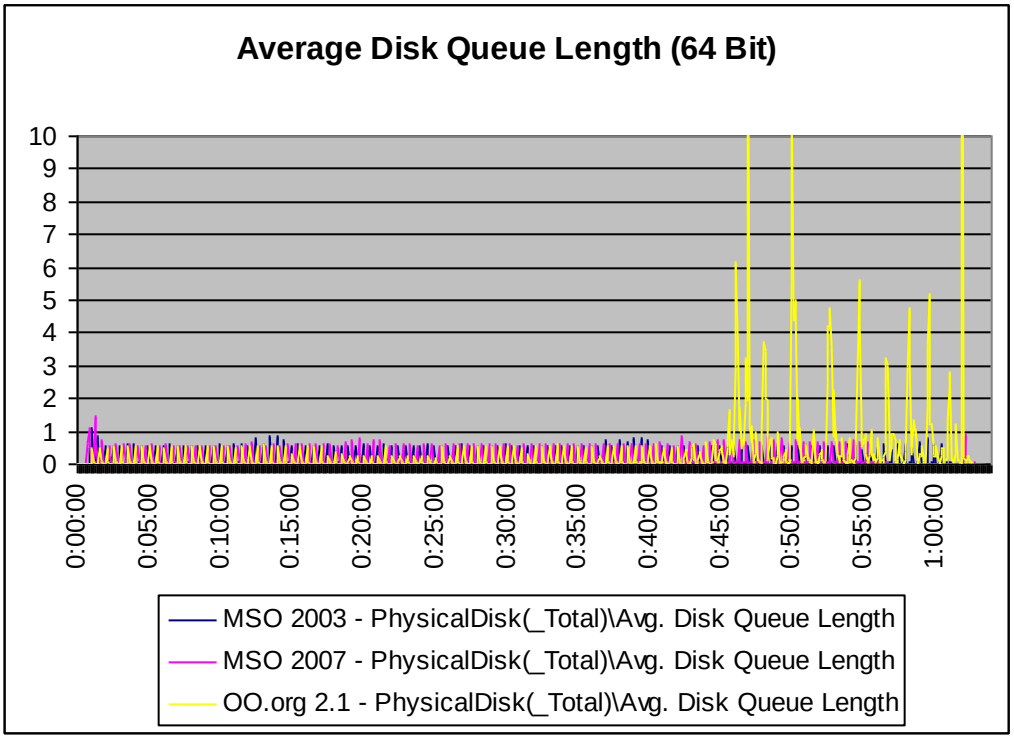


Upon closer inspection of hard disk activity, the different resource requirements are confirmed. As seen above, increased access to the hard drive, which causes longer waiting times, begins on the 32-bit server with OpenOffice.org 2.1 with 50 or more users due to the draw on free memory. For the 32-bit server with Microsoft Office 2007 and Office 2003, similar patterns first develop with roughly 80 and 100 users, respectively.

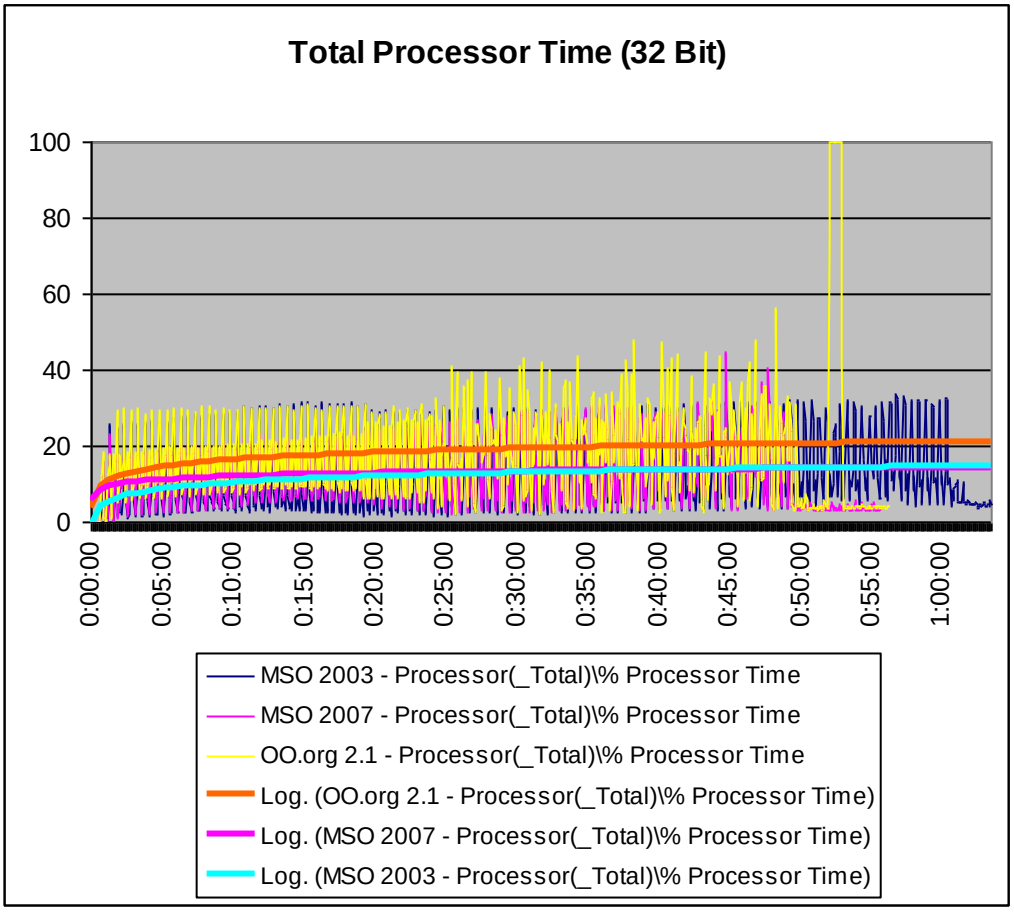
The onset of such concentrated memory paging often leads to a clearly noticeable decline to the system reaction within the individual user sessions on a terminal server. This behavior can be observed in tests as well. The log-on times for the sessions will become longer after the fiftieth user has logged on to the system with OpenOffice.org. For users, interaction with the system will become markedly slower as of this moment.



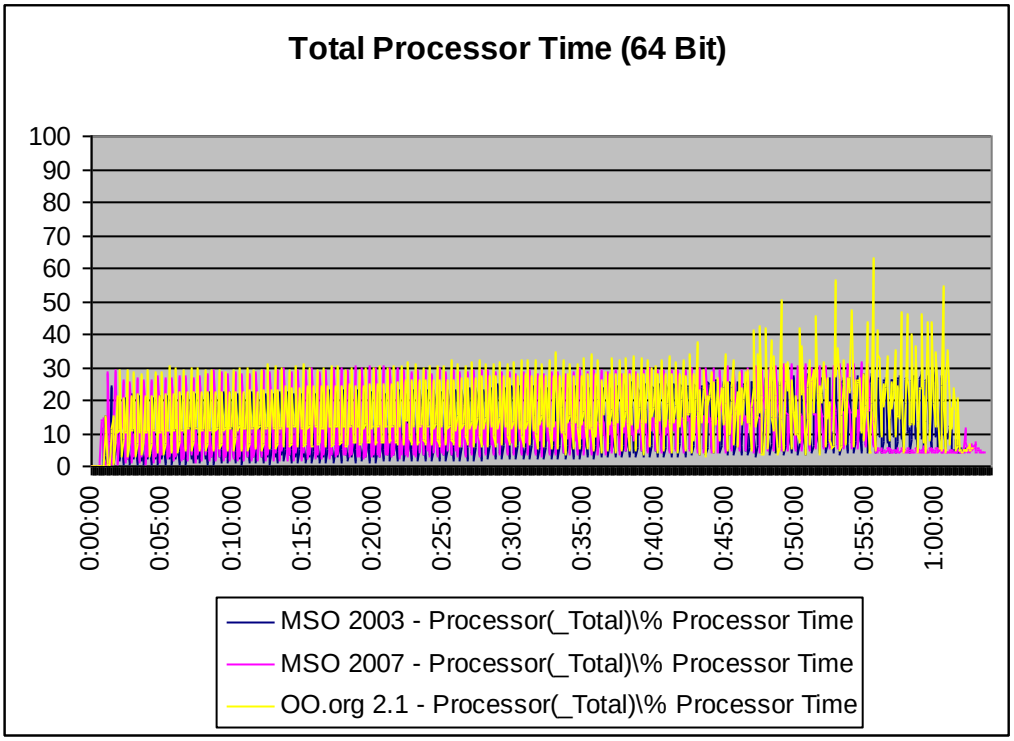
Marked increases to queue lengths will only be observed on OpenOffice.org because Microsoft Office's permanent memory optimization of the individual related processes does not necessitate sudden, massive paging actions on behalf of memory management.



Along with memory load, differences in processor loads are also apparent. On average, the 32-bit server with OpenOffice.org 2.1 displays a higher load than the comparable 32-bit server with Office 2003 or Office 2007. OpenOffice.org generates a mean processor load of 20%, whereas the figure for servers with Microsoft Office is around 10%. Large spikes to CPU load are also obvious once the physical memory is full, requiring massive paging activities.



The 64-bit variant of Windows Server 2003 appears similar to the 32-bit platform in terms of processor load on the server with various office suites.



The results show that roughly 30 to 40% more users can sign on to the platform with Microsoft Office 2003 than with OpenOffice.org 2.1 with identical hardware and an identical configuration of the terminal server on Windows Server 2003 SP1. Even for terminal servers with Microsoft Office 2007, one can expect that the system resources will be used somewhat more efficiently than is possible on a system with OpenOffice.org. This applies both for the 32-bit and the 64-bit version of Windows Server 2003 when used as the primary platform.

6 Appendix

6.1 Configuration of the Servers

Necessary modifications to the pre-installed basic system:

Bginfo	Removal of Bginfo.exe from Documents and Settings\All Users\Start Menu\Programs\Startup
Screen saver	Deactivate per local policy (in Gpedit.msc) under Local Computer Policy | User Configuration | Administrative Templates | Control Panel | Display | Screen Saver
Required tools	The following tools and files must be located on the server on the C:\: • Pslist.exe • Sleep.exe • TS Test.htm
Log-on script	Copy the log-on script on \Documents and Settings \All Users \Start Menu \Programs \Startup\
Documents	Copy the documents on C:\Install \Documents\

6.2 Detailed Measurement Results

Log-on times for individual users per system, where MSO stands for Microsoft Office and OOO stands for OpenOffice.org.

	MSO 2003 32 Bit	MSO 2007 32 Bit	MSO 2007 64 Bit	OOO 2.1 32 Bit	OOO 2.1 64 Bit
User 1	1.3s	1.3s	1.3s	1.3s	1.2s
User 10	1.3s	1.3s	1.2s	1.3s	1.2s
User 50	1.3s	1.4s	1.3s	2,5s	1.3s
User 80	1.4s	1.5s	1.3s	1.6s	1.3s
User 90	1.4s	1.6s	1.3s	1.8s	1.5s
User 100	1.4s	--	1.4s	--	2,1s
User 110	1.4s	--	1.4s	--	1.4s
User 120	1.4s	--	--	--	1.6s

Selected measured values after the completion of individual test runs.

	MSO 2003 32 Bit	MSO 2003 64 Bit	MSO 2007 32 Bit	MSO 2007 64 Bit	OOO 2.1 32 Bit	OOO 2.1 64 Bit
Processes (max.)	1,238	1,358	916	1,245	928	1,238
Threads (max.)	7,310	7,381	6,256	7,689	5,847	7,315
Text processing	120	120	97	113	99	120
Spreadsheet	120	120	97	112	99	120
Presentation	120	120	95	113	99	120

6.3 Adaptation of Microsoft Office 2007

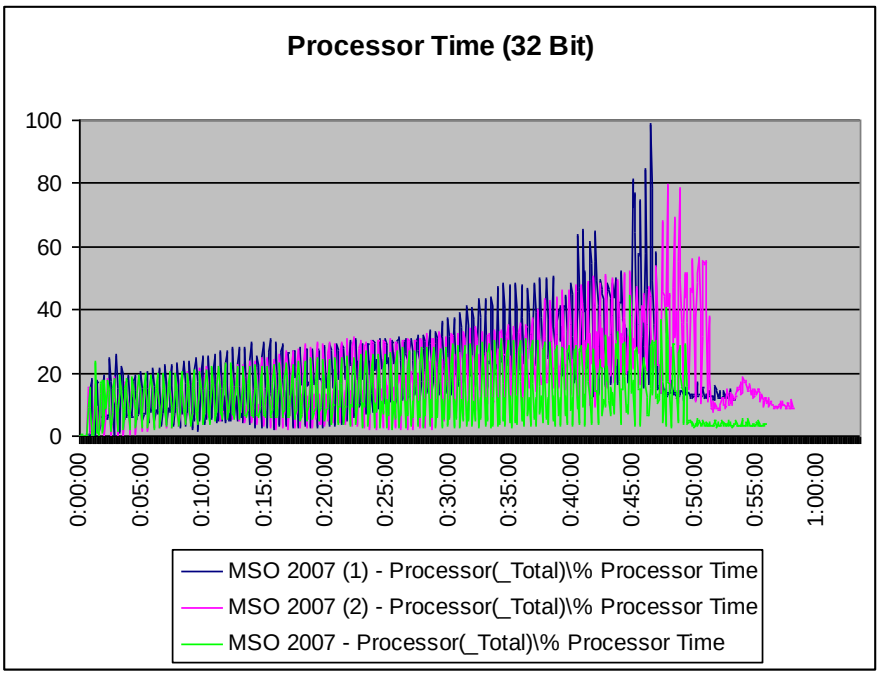
Microsoft Office Ultimate 2007, installed at the beginning of the test, required increasing possessor resources as the number of user sessions increased, whereas Office 2003 and OpenOffice.org 2.1 did not. This difference was clear even on the 64-bit platform, whereby at the end of the test with Office Ultimate 2007, the fully occupied processor resources did not allow any more users to log on the system.

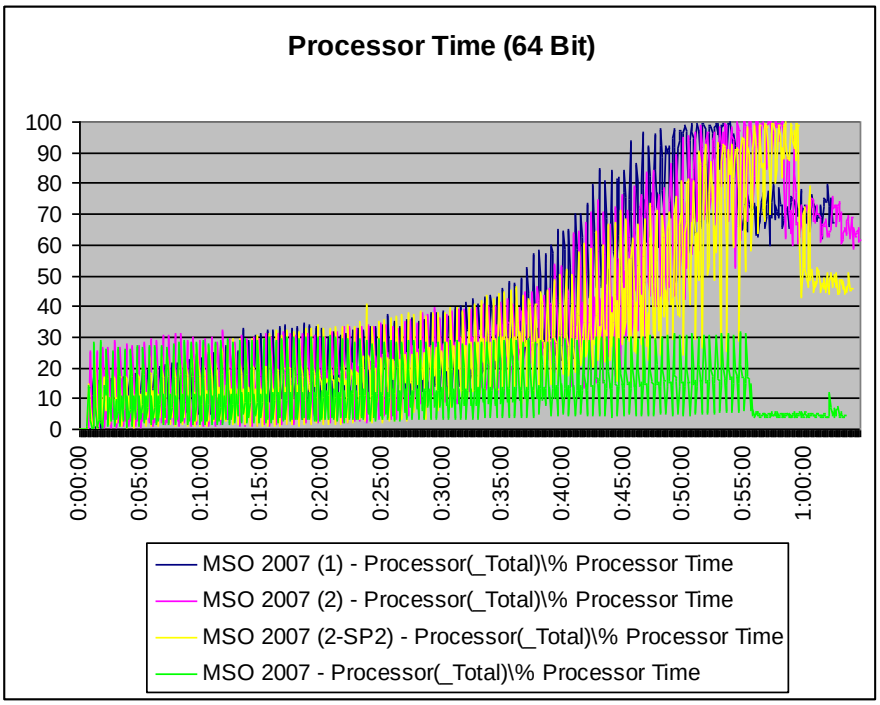
The new process GrooveMonit.exe, which was loaded for each user session, contributed to the memory load. Directly after its activation, it occupied 5.7 MB, later occupying a typical 2.6 MB. This occupied an additional 250 MB of memory on the total system.

In order to guarantee comparisons with other office suites (Microsoft Office 2003 and OpenOffice.org 2.1), the actual test Office Professional Plus 2007 (without Groove 2007) was implemented. It was additionally necessary to ensure that the menu for the office applications no longer blinked by setting the registry entry OfficeMenuDiscovered under HKCU \SOFTWARE \Microsoft \Office \12.0 \Common \General to the value DWORD:1.

The following images display the differences for processor loads during the respective test runs:

- MSO 2007 (1): Microsoft Office Ultimate 2007.
- MSO 2007 (2): Microsoft Office Professional Plus 2007 without setting the registry entry OfficeMenuDiscovered to DWORD: 1.
- MSO 2007 (2-SP2): Microsoft Office Professional Plus 2007 without setting the registry entry OfficeMenuDiscovered to DWORD: 1. use of Windows Server 2003 SP2.
- MSO 2007: Microsoft Office Professional Plus 2007 setting the registry entry OfficeMenuDiscovered to DWORD: 1.





The variant MSO 2007 was used for all processor tests.

6.4 Step-by-Step Description of a Test Run

6.4.1 Setting up the Load Generators

1. Install vRD Load Edition (vRDLoad) by running vRDLoadSetup.msi.
2. Modify settings of .vrd XML files to be compliant to your test environment: Domain (IP address of test server), username and password.
3. Copy the required .vrd files to the clients.
4. Run vRD on the Controller and log in to all clients.
5. Run vRD Load Edition on the clients and import the appropriate .vrd file.
6. Change settings in vRD Load Edition, such as the interval between automated user log ins.

6.4.2 Test Sequence

1. Run MSinfo32.exe, store data into a file and copy the output file to a save place. IMPORTANT: This is a required test result!
2. Set up the performance log. To do so select Performance in the Administrative Tools menu. In the Performance Monitor right-click Performance Logs and Alerts | Counter Logs and select New Log Settings From. In the Open dialog select TS Test.htm file as a template for the required performance counters. Now start the log.
3. Start the test sequence in vRD Load Edition.
4. Now the test sequence is running. On the server, you should do nothing interactively except watching Task Manager. You can interrupt the test sequence manually by clicking the Stop button in all instances of vRD Load Edition. When the test is over all Start buttons in vRD Load Edition switch back to active.
5. Stop the performance log and copy the log file to a save place. IMPORTANT: This is a required test result!
6. Run “pslist -m > C:\<testname>_after.txt” on the server and copy the output file to a save place. IMPORTANT: This is a required test result!
7. Save the log file vRDLoad.log of vRD Load Edition from the Program Files\visionapp\visionapp Remote Desktop. Then rename the old log file in the program folder. This allows vRD Load Edition to create a new log file during the next test sequence.

6.4.3 Cleaning up

1. Use Terminal Services Manager to remove all user sessions from the test server.
2. If Terminal Services Manager is not able to start due to memory limitations, use Task Manager to force one user log off after the other.
3. If you want to use the server for another test, you have to remove all user profiles. This can be done by running System from the Control Panel. In the Advanced tab select Settings of the User Profiles. Delete all user profiles beginning with “v” and followed by a number. NOTE: Use shortcuts (Alt+D and Alt+Y) instead of mouse clicks.