

# **Microsoft Office 2003 Application Scalability Analysis**

**By Citrix Consulting**

**Citrix Systems, Inc.**

---

## Notice

The information in this publication is subject to change without notice.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. CITRIX SYSTEMS, INC. ("CITRIX"), SHALL NOT BE LIABLE FOR TECHNICAL OR EDITORIAL ERRORS OR OMISSIONS CONTAINED HEREIN, NOR FOR DIRECT, INCIDENTAL, CONSEQUENTIAL OR ANY OTHER DAMAGES RESULTING FROM THE FURNISHING, PERFORMANCE, OR USE OF THIS PUBLICATION, EVEN IF CITRIX HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES IN ADVANCE.

This publication contains information protected by copyright. Except for internal distribution, no part of this publication may be photocopied or reproduced in any form without prior written consent from Citrix.

The exclusive warranty for Citrix products, if any, is stated in the product documentation accompanying such products. Citrix does not warrant products other than its' own.

Product names mentioned herein may be trademarks and/or registered trademarks of their respective companies.

**Copyright © 2003 Citrix Systems, Inc., 851 West Cypress Creek Road, Ft. Lauderdale, Florida 33309-2009 U.S.A. All rights reserved.**

Version History		
Dan Simmons	July 2003	Version 1.0

# Table of Contents

<b>INTRODUCTION .....</b>	<b>1</b>
KEY FINDINGS .....	1
DOCUMENT OVERVIEW .....	2
<b>EXECUTIVE SUMMARY .....</b>	<b>3</b>
OBJECTIVE .....	3
RESULTS SUMMARY .....	3
APPROACH .....	4
COMMENTS .....	4
<b>SCALABILITY TESTING METHODS.....</b>	<b>6</b>
SCRIPTED TEST.....	6
REAL USER TEST .....	6
REAL USERS WITH TASK LIST .....	6
COMBINATION .....	7
SCALABILITY TEST METHODS SUMMARY .....	8
SCALABILITY TESTING METHOD.....	8
<b>LOADRUNNER AND CITRIX TESTING PROCESS.....</b>	<b>9</b>
PLANNING THE TEST .....	9
CREATING vUSER SCRIPTS .....	9
BENCHMARKING .....	9
CREATING THE SCENARIO .....	10
RUNNING THE SCENARIO.....	10
MONITORING THE SCENARIO.....	10
ANALYZING TEST RESULTS.....	10
<b>PERFORMANCE METRICS THRESHOLDS .....</b>	<b>11</b>
USER EXPERIENCE .....	12
<b>CUSTOM SCRIPTS.....</b>	<b>13</b>
REQUIREMENTS AND ASSUMPTIONS .....	13
SCRIPTS .....	15
<i>Standard Word User Script Narrative .....</i>	<i>16</i>

<i>Power Office User Script Narrative</i> .....	17
<b>ENVIRONMENT SPECIFICATIONS</b> .....	<b>19</b>
METAFRAME XP ARCHITECTURE OVERVIEW .....	19
<i>LoadRunner Controller and vUser Generators</i> .....	20
<i>MetaFrame XP Presentation Server</i> .....	20
<i>SQL 2000 Database sever</i> .....	20
METAFRAME XP HARDWARE SPECIFICATIONS .....	21
SUPPORTING HARDWARE SPECIFICATIONS .....	22
SOFTWARE AND PLATFORM SPECIFICATIONS .....	23
CITRIX® METAFRAME XP™ PRESENTATION SERVER CONFIGURATION .....	24
<i>ICA Connection Configuration</i> .....	24
<b>SCALABILITY TEST CASES</b> .....	<b>25</b>
CASE 1: NO-LOAD TEST .....	25
CASE 2: BASELINE TESTS .....	25
CASE 3: LOAD TESTS .....	25
<b>SCALABILITY RESULTS AND ANALYSIS</b> .....	<b>26</b>
STANDARD WORD USER RESULTS .....	26
ANALYSIS .....	27
POWER OFFICE USER RESULTS .....	28
ANALYSIS .....	29
<b>APPENDIX A: LOADRUNNER</b> .....	<b>30</b>
VUSER SCRIPT GENERATION .....	30
TIPS FOR WORKING WITH CITRIX VUSER SCRIPTS .....	31
<i>Recording Tips</i> .....	31
<i>Replay Tips</i> .....	31
<b>APPENDIX B: PERFORMANCE METRICS</b> .....	<b>32</b>
PERCENT PROCESSOR TIME .....	32
PROCESSOR QUEUE LENGTH .....	32
AVAILABLE AND COMMITTED MEMORY .....	32
PAGE FAULTS PER SECOND .....	32
PAGE READS PER SECOND .....	32

FREE SYSTEM PAGE TABLE ENTRIES.....	33
PERCENT USAGE PAGE FILE.....	33
PERCENT DISK TIME.....	33
CURRENT DISK QUEUE LENGTH.....	34

## Introduction

Citrix Systems, Inc. has been working to bring our customers the services and software necessary to operate efficiently in the challenging IT environment of today. To ensure accuracy and timely support, Citrix sought to evaluate application performance and functionality of the Microsoft Office 2003 application suite delivered through a Citrix® MetaFrame XP™ Presentation Server, Feature Release 3 server farm.

Citrix consulting conducted an application analysis project focusing on the following key objectives:

- Test the application functionality of Microsoft Office 2003 while hosted on a Windows 2003 Enterprise server running Citrix® MetaFrame XP™ Presentation Server, Feature Release 3
- Determine resource consumption factors that will affect the number of users that can be hosted in a MetaFrame XP Presentation Server environment on a single server

The test project was conducted using the following tools and technologies:

- Citrix® MetaFrame XP™ Presentation Server with Feature Release 3
- Microsoft Office 2003 (version 11)
- Windows 2003 Enterprise Server with Terminal Services
- Mercury Interactive's LoadRunner 7.51
- Citrix 's ICA Client integration with Mercury Interactive's LoadRunner
- Dell's PowerEdge™ 1655MC Blade servers

## Key Findings

**Primary Scalability Factors.** Memory and CPU utilization were the primary constraints observed throughout the test process, with CPU Utilization representing the primary factor limiting scalability.

**CPU.** The primary factor affecting scalability observed during the scalability analysis was CPU utilization. Thus, CPU utilization was the primary area of focus in determining the maximum number of users that could be logged on while still maintaining a level of operability within the Microsoft Office 2003 suite. For the purpose of this analysis, a sustained CPU utilization rate of 80% was deemed the maximum load as per Microsoft best practice recommendations on sustained CPU utilization rates.

**Memory.** Memory consumption was the secondary limiting factor after CPU utilization. Baseline tests indicated that the average memory consumption per user was approximately 34MB. Memory consumption scaled linearly and was not a factor even when the maximum level of users was reached based off CPU utilization thresholds. Thus, in server class systems with more processing power and the same amount of RAM, memory may become the limiting scalability factor. Refer to the [MetaFrame XP Hardware Specifications](#) section for specific details on the Dell PowerEdge 1655MC Blade servers used in this analysis.

**User Limits.** Maximum observed server load on the system was 81 for "Standard Word" users, and 57 for "Power Office" users. For the purposes of this test, the more conservative figures of 60 to 65 users reflect tests with minimal transaction errors and good application response times. These are reported as the expected production load limit.

**Configuration.** No special configurations were required or applied to the Office 2003 suite. All test user accounts ran using a default Microsoft 2003 office setup with standard Windows local profiles. The MetaFrame XP Presentation Server was configured and then modified to conform to best practice recommendations listed in the [MetaFrame XP Advanced Concepts Guide](#) Optimizations section.

**Subjective Testing.** The primary focus of the testing was to evaluate application response and usability. To ensure functional validity in the testing, the test team performed logins and limited manual application execution to evaluate application response and usability. Throughout the entire range of testing, the performance for Standard Word users remained acceptable, without any noted failures or issues in application delivery or processing up until the number of running users exceeded 70. Once the system exceeded this level of Standard Word user activity, slow logon times were observed and initial application execution delays were apparent. The performance for Office Power users remained acceptable, without any noted failures or issues in application delivery or processing up until the number of running users exceeded 50. Once the system exceeded this level of Power Office user activity, slow logon times were observed and initial application execution delays were apparent.

**Test Procedures.** The project team relied entirely on Mercury Interactive's LoadRunner integration with the Citrix ICA client. This test approach ensures that no process is executed against the server that would artificially reduce the overall load figures. Additionally, since the script execution relied on mouse clicks, typing, and other UI based activities, and the script represented an extremely close approximation for a production user executing the Office 2003 suite of applications via Citrix MetaFrame XP Presentation Server with Feature Release 3.

## Document Overview

This document is broken down into the following major sections:

- [Scalability Testing Methods](#) – Discusses the differences between the types of scalability testing methods such as scripted versus real user tests.
- [LoadRunner and Citrix Testing Process](#) – Discusses the testing methodology used to perform the scalability tests.
- [Benchmarking](#) – Describes the Performance Monitor benchmarks used to set thresholds of acceptable performance.
- [Performance Metrics](#) – Details the meaning and significance of each of the Performance Monitor counters used to analyze the servers.
- [Performance Thresholds](#) – Identifies the common thresholds for the performance metrics.
- [Custom Scripts](#) – Discusses the requirements, assumptions, and the process flows behind the custom scripts used in the scalability testing.
- [Environment Specifications](#) – Details the MetaFrame XP architecture and configuration, server hardware, software and platform specifications, and all other configurations required to setup the MetaFrame XP testing environment.
- [Scalability Test Cases](#) – Discusses the cases that were used during the scalability testing efforts.
- [Scalability Results and Analysis](#) – Presents and analyzes the meaning of the results obtained from the testing effort.
- [Scalability Test Summary](#) – Presents conclusions from the results and analysis data and provides future direction and caveats. Additionally, provides a summary format of the test results / procedures that can be reviewed independent of the rest of the document.
- [Appendices](#) – The appendices provide supplemental information for the content provided in the main body of the document.

## Executive Summary

### Objective

The purpose of this analysis was to determine the number of users a server running Citrix MetaFrame XP Presentation Server can support when running the Office 2003 Application Suite. For this analysis a Dell PowerEdge 1655MC Blade dual 1.4GHz processor server was used. Server resources and usability are the primary factors in determining this total number of concurrent users. As server resources become increasingly scarce, latency increases reducing the usability of the applications.

### Results Summary

# Of Sessions	Mem: Available GB	Mem: Page Faults/sec	Mem: Pages/sec	Pagefile: % Usage	Disk: % Disk Time	Disk: Queue Length	Proc: % Proc Time	Usability
<b>Baseline (0 users)</b>	1.63	112.8	0.004	0.933	3.16	0.06	2.39	Excellent
<b>71 Standard Users</b>	0.42	1323.8	4.1	9.7	76.7	0.32	76.92	Acceptable
<b>51 Power Users</b>	0.54	4903.54	26.15	14.80	96.09	0.34	85.91	Acceptable

The recommended number of concurrent user sessions for Microsoft Office 2003 running on a Windows 2003 Enterprise server with MetaFrame XP Presentation Server with Feature Release 3 is approximately **60-65** users on a Dell PowerEdge 1655MC Blade dual processor 1.4GHz server with 2 GB RAM. This value was determined based on the extensive testing and resulting data captured during the tests detailed in this document. The tests were performed repeatedly, to confirm initial results and ensure the reliability of the data. Further, the project team applied both objective and subjective analysis methods to complete the evaluation. The objective analysis/counters are detailed in the section [Performance Metrics](#). The subjective analysis relied upon response time to mouse movements & keystrokes, as well as time to paint the screen. The subjective analysis was relative to a stand-alone workstation running local to the MetaFrame XP servers with no outside factors to impact performance. An explanation of the subjective factors is detailed in the section [User Experience](#).

Because application usage patterns will be a variable in each environment, it is strongly recommended that each MetaFrame XP environment perform a level of detailed scalability analysis to accurately determine the supported number of concurrent Office 2003 user sessions. The figures represented here were determined based on the specific hardware and test procedures specified in this document.



## Approach

The testing approach applied by Citrix consulting was structured with two primary scenarios for evaluation:

- **Baseline testing scenario.** A Mercury Interactive LoadRunner scenario designed to evaluate the expected server resource consumption of a single user and a baseline test with no users. This test was evaluated from system logon through logoff.
- **Steady-state scenario.** A Mercury Interactive LoadRunner scenario designed to place an increasing number of users on the system performing a series of functional tasks concurrently. This was accomplished by loading two users every thirty seconds, then after all the designated number users have logged onto the system one additional user would logon manually to perform usability testing.

Baseline figures are appropriate for early system and environment planning, as well as providing good data that can be used in calculating budgets and initial Total Cost of Ownership (TCO) figures. The additional data gathered during the steady-state testing represent conservative figures, since it is expected that an actual production server would not be used in such a rapid, consistent fashion. To further ensure accuracy and relevancy in individual environments, Citrix consulting strongly recommends that all scalability testing follow a standard process that includes:

- Evaluation of the application and/or client components in a stand-alone environment. Single session and/or single workstation.
- Creation of a baseline scripted test scenario to evaluate multiple concurrent sessions and their impact on system performance.
- Evaluation of the user environment and generate more complex and customized scripted test scenarios to better evaluate the application's performance in a production simulation environment.
- Deployment of a proof-of-concept to a small user subgroup, monitor their testing, and use of the application.
- Finally, when deploying the application, deployment first into a pilot environment. Evaluate and monitor the performance in the pilot environment, then migrate the application into full production. Utilize load manager for MetaFrame XP to ensure optimal application and server performance by limiting the concurrent sessions/applications on the MetaFrame XP servers.

## Comments

It is important to note that the scalability testing was performed to determine the server sizing, which is testing that measures the ability for a single MetaFrame XP server to support a given user load. Following the Citrix consulting methodology, server scalability is performed during a Proof of Concept, or a Pre-Production Pilot. It is important to note that server scalability differs from system scalability testing in that system scalability measures the ability for the overall infrastructure (of which MetaFrame XP is one component) to support a given user load versus the scalability of an individual server to support a given user load. The overall infrastructure includes components such as the Network (LAN/WAN/Internet), MetaFrame XP Servers, Application Servers, Database Servers, Authentication Servers, File Servers and Print Servers. System scalability is undertaken for a production environment design and pilot.

The scalability tests for this project were geared towards determining the maximum number of Office 2003 application users that a two-processor server can support given defined hardware specifications and constraints. Please refer to the [Environment Specifications section](#) in this document for the server specifications of the equipment used.

The tests showed that some server resources were not fully utilized even with the maximum user loads. However, this does not mean they should be overlooked or further stressed in a production environment. They help maintain availability on each server in order to provide redundancy within a MetaFrame XP server farm. For example, if one or more servers go off-line, the remaining servers should be able to handle the additional connections. Furthermore, continually running at, or near, capacity will increase the likelihood of a server crash.

The scripted scalability tests used in this analysis are helpful in obtaining rough estimates for server sizing. The data and analysis in this document should serve as a guide, and not definitive numbers, for further MetaFrame XP scalability

testing of the Office 2003 Application Suite. The results obtained from further testing may not be the same as the results from this testing effort. Each environment is unique and the number of concurrent user sessions could vary significantly from one environment to the next. Also, each end-user utilizes the applications differently making it difficult for scripted scalability tests to accurately mimic all user activity. Therefore, the true scalability of the environment, design, and configuration should be determined by conducting a full pilot with "live" users.

## Scalability Testing Methods

In a scalability test of Microsoft Office 2003 running on a Windows 2003 Enterprise server with MetaFrame XP Presentation Server with Feature Release 3, decisions regarding the test methods must be made in order to standardize a valid test of the environment.

Four primary types of scalability testing are appropriate for the MetaFrame XP Presentation Server environment:

- **Scripted Test:** Automated execution of scripts that mimic a user's actions without any user intervention.
- **Real Users Test:** Actual users enter the system and run through their daily tasks without any specified order.
- **Real Users with Tasks List:** Actual users enter the system and run through a set of pre-defined tasks.
- **Combination:** A combination of two or more of the aforementioned testing methods.

The following section discusses each method in more detail and compares the advantages and disadvantages of each method. Citrix chose to leverage the scripted approach, as the scripted test allowed the system to be quickly loaded to failure in a very controlled fashion.

### Scripted Test

A standard set of scripts can be written to control the actions of test users. Test user scripts are created and configured similar to typical MS Office users. These scripts would be developed to simulate a desired set of predefined actions (workflow). The workflows created should be based on the user's role and applications they use during a typical user session. Each workflow may contain sub-workflows that dictate the multiple paths users take to complete these daily tasks. These sub-workflows will be the basis for scripts that are generated. Script execution would be randomized across users to ensure that steps taken while working in an application are not repeated for users involved in the test.

For the test process detailed in this document, the functional flows for these scripts had been developed by the project team, which focused these scripts on basic operation within each application.

### Real User Test

The second method for scalability testing is to have users log into the system and perform tasks of a typical workday. The results obtained from this method are geared toward a real-life scenario. The caveat to using this method is that more variables are present in the test. This makes it more difficult to run the same exact test while increasing user load, making system configuration changes, or repeating the test. Additionally, the project team does not have access to a large enough user base to perform these tests. Therefore, it would be difficult to obtain actual users to enter the system and perform daily tasks.

When running this type of test, most client environments would benefit from monitoring their systems and capturing the performance counters and data in a database format. Resource manager for MetaFrame XP is designed to accomplish this, and these figures can provide significant value and accuracy, if a large enough population sample of data is captured.

### Real Users with Task List

The next method for scalability testing is a combination of scripted tests and real user testing. Real user testing with task lists includes having real users access the system, while executing a written set of tasks in a random order. Developing customer specific tasks for scalability testing will best represent the different types of users that will access the system on a daily basis. Each user will be accessing the system at different speeds, reflecting a realistic production environment. However, these users will be following a common set of tasks that will help with standardizing the scalability tests when they need to be re-run with additional users.

This type of test is resource intensive and can be difficult to coordinate. Most corporate environments cannot provide multiple resources for this type of application testing and evaluation. Developing customer specific tasks for scalability testing will best represent the different types of users that will access the system on a daily basis. Each user will be accessing the system at different speeds, reflecting a realistic production environment. However, these users will be following a common set of tasks that will help with standardizing the scalability tests when they need to be re-run with additional users.

## Combination

The final method for scalability testing is a combination of a custom script and real users accessing the test environment. For example client computers emulating users each could be used in conjunction with real users performing more complex Office 2003 operations. This would allow the administrators to load the system to specific level, and then evaluate the subjective and objective results of the users' interaction with the MetaFrame XP servers.

## Scalability Test Methods Summary

The table below, Table 1 - Scalability Test Methods Summary, summarizes the advantages and disadvantages of each scalability test method described above.

Testing Method	Advantages:	Disadvantages:
<b>Scripted Test:</b>	<ul style="list-style-type: none"> <li>No variables. Completely controlled</li> <li>Identical tests can be repeated as many times as needed</li> <li>No user time required to do test</li> <li>Tests can be re-run as environment grows</li> </ul>	<ul style="list-style-type: none"> <li>Takes significant time and tools to create test scripts</li> <li>No "user skill levels" incorporated into test</li> </ul>
<b>Real Users Test:</b>	<ul style="list-style-type: none"> <li>Real life test</li> <li>Allows for different user types and skill levels</li> </ul>	<ul style="list-style-type: none"> <li>Impossible to have two identical tests</li> <li>User's time is needed to perform test</li> <li>Need users from each application's user base</li> </ul>
<b>Real Users with Task List Test:</b>	<ul style="list-style-type: none"> <li>Can be as controlled as needed</li> <li>Test can be repeated with high degree of similarity between previous tests</li> <li>Allows for different user types and skill levels</li> </ul>	<ul style="list-style-type: none"> <li>User's time is needed to perform test</li> <li>The project team will have to create a task list for users customized to their role. This can be very complex and time consuming</li> </ul>
<b>Combination</b>	<ul style="list-style-type: none"> <li>Can emulate most user activities with custom scripts and live users can test actions that were not scripted and determine the acceptable latency.</li> </ul>	<ul style="list-style-type: none"> <li>Multiple users' time is needed to perform test</li> </ul>

**Table 1 - Scalability Test Methods Summary**

## Scalability Testing Method

Based on the project requirements, Citrix selected the scripted test method. This ensured identical, controlled tests that could be repeated for multiple tests. The scripts developed for testing were carefully formulated by Citrix to accurately simulate normal user load on the MetaFrame XP servers. Usability tests were also performed where a real user accessed the system during various load levels to determine if the applications responded at an acceptable rate.

The testing tool that was used to develop the Office 2003 scripts was Mercury Interactive's LoadRunner with the Citrix ICA client integration. For more information regarding Mercury Interactive's LoadRunner and the partnership between Citrix and Mercury please refer to their web site:

[http://www.svca.mercuryinteractive.com/alliances/alliance\\_directory/index/citrix.html](http://www.svca.mercuryinteractive.com/alliances/alliance_directory/index/citrix.html).

## LoadRunner and Citrix Testing Process

As with all testing strategies, a clearly defined testing process helps to ensure accurate and repeatable results. The following section is an overview of a six-step process for testing applications under load.

### Planning the Test

Successful testing requires development of a thorough test plan. A clearly defined test plan ensures that the LoadRunner scenarios that are developed will accomplish the load testing objectives. Load test planning involves:

- Analyzing the application to determine hardware and software components, the system configuration, and typical usage patterns.
- Defining testing objectives (e.g., maximum user load, application upgrade compatibilities, and identifying bottlenecks).
- Planning the LoadRunner implementation. This involves defining the scope of performance measurements, defining vUsers workflows, selecting vUsers, and choosing test hardware.
- Define the launch sequence of the test users.
- Monitor the server with no activity before test execution. This will allow the project team to acquire a proper benchmark (see the [Benchmarking](#) section for additional information).
- After the benchmark information has been gathered in a log file, start adding groups of users by spawning test sessions from the LoadRunner controller. The LoadRunner scenarios can be configured to control the speed and timing of user script execution. For these tests, the project team chose to initiate script activity immediately once each user logged onto the system.
- Additional users should be added to the test until it is complete. Throughout the duration of the scripted test, a user should manually logon to system to measure user experience (see the [User Experience](#) section for more information) to validate the performance of the system.
- For the initial scalability and performance tests, performance graphs should be monitored. User load increases should be stopped when the system is reaching critical thresholds or the scripts fail to respond. These thresholds are explained later in the [Performance Metrics Thresholds](#) section of this document.
- After investigating the results of the initial scalability and performance tests, some performance enhancements or configuration changes can be made to the MetaFrame XP server such as the addition of more RAM or faster processors. After these changes, the test should be re-run in the identical fashion and results should be compared to the previous test to see if any performance gains were detected. If performance loses are detected or the modifications result in no increase in performance, the changes and enhancements should be removed.

### Creating vUser Scripts

The vUsers emulate simulated production users interacting with the system under test conditions. The vUser scripts contain the actions that each virtual user will perform during a scenario execution. These scripts should emulate what real users typically do with the system in a production environment. Creating a detailed functional flow of user activities, breaking the flow down into manageable transactions, and then creating the actual test script can best accomplish this task.

### Benchmarking

To get a better result set from the scalability and performance tests for the MetaFrame XP environment, some performance logging should be completed before users start accessing and testing the server. Doing so will help to determine the system resources needed to run the operating system and testing tools. This information will allow a much better representation of what resources the application requires.

## **Creating the Scenario**

A scenario describes the events that occur in a testing session. A scenario includes defining the client machines that vUsers will run on, scripts that vUsers will execute, and a specified number of vUsers or vUser groups that run the scenario. Perhaps most importantly, the LoadRunner scenario can be configured to control the rate at which users are introduced into the actual test. This scheduling feature is extremely powerful, and when combined with the other LoadRunner Controller Scenario features, and provides the test team with an invaluable tool.

## **Running the Scenario**

User load is emulated by instructing multiple vUsers to perform tasks simultaneously. Before executing a scenario, configuration and scheduling is defined. This determines how all the load generators and vUsers behave when the scenario is run.

## **Monitoring the Scenario**

While the scenario is being run, a monitoring tool such as Performance Monitor or resource manager for MetaFrame XP should be leveraged to monitor all components of the MetaFrame XP servers. Real-time effects of the user load can be observed using the resource manager components in the management console for MetaFrame XP.

## **Analyzing Test Results**

Throughout execution of a scenario, the monitoring tool records the performance of the system under test at different load levels. This information is later organized into a readable format for archiving, analysis, and reporting.

## Performance Metrics Thresholds

Below is a table of standard thresholds when monitoring system performance. These are very general guidelines and counter threshold values and they should be used as a basis to help identify points at which performance bottlenecks could be developing.

	General Problem Value	General Cause	Possible Other Factors
<b>Percent Processor Time</b>	>= 70 – 80% Consistently	Processor Bottleneck	Disk Bottleneck, Memory Paging / Faults, and / or Application type / coding
<b>Processor Queue Length</b>	> 2	Processor Bottleneck	Disk Bottleneck and/or Memory Paging / Faults
<b>Memory – Available Bytes</b>	< 30% of RAM 4 GB (4 Proc) 2 GB (2 Proc)	Memory Bottleneck	
<b>Memory – Committed Bytes</b>	85% of Page File	Memory Bottleneck	
<b>Memory – Page Reads/sec</b>	> 5	Memory Bottleneck	Application type / coding
<b>Memory – Free System PTEs</b>	Varies*	Large number of users / applications	Multi-processor systems
<b>Page File – Percent Usage</b>	85%	Memory Bottleneck	Disk Activity
<b>Physical Disk - % Disk Time</b>	40%	Disk Bottleneck	Memory
<b>Physical Disk – Disk Queue Length</b>	>= 2 Consistently	Disk Bottleneck	Memory

**Table 2 – Performance Metrics Thresholds**

**Note:** The number of Free System PTEs varies depending on the memory configuration for the server. Typically, PTEs only cause bottlenecks on servers with a high number (4, 6, or 8) of processors.



## User Experience

The user experience is a key value used to establish thresholds for the scalability testing, as application responsiveness exposes the net effect of all load stress on the system. Unlike the other data, the user experience is a subjective measure that cannot be found in resource manager or Performance Manager. The user experience is determined by executing manual sessions during the test to validate the “user experience”. The user experience is based on subjective analysis and is divided into four levels or categories detailed in the following table. The category into which a user experience measurement falls is determined by the time required to complete the logon process, keystroke-to-screen update time, the time required to enumerate local client drives through Windows Explorer, and the performance of the applications.

	<b>Application Responsiveness</b>
<b>Excellent</b>	Equivalent or better than local PC performance.
<b>Acceptable</b>	Screen updates are fluid and there is minimal effect on user’s workflow.
<b>Poor</b>	Screen updates are noticeable and latency is increased. However, the user is still able to function.
<b>Failure</b>	The session becomes frozen or disconnected. Therefore, the user cannot continue his/her tasks.

**Table 3 – User Experience Measurements**

## Custom Scripts

This section details the custom scripts developed for the Microsoft Office 2003 application scalability initiative. Citrix has defined two types of virtual user groups based on the usage patterns of two types of users that may run the Office 2003 application suite:

- **Standard Word Users** – Standard word users run a single published application of Microsoft Word 2003. A standard user will do a simple task and then stop. The functionality that a standard user implements in an application will be less complex than a power user.
- **Power Office Users** – Power office users are defined as users that will be running a MetaFrame XP published desktop to access multiple applications at one time. Power users will generate functionality that causes interaction between two applications, such as copying an object from Excel into Word.

Scripts were created for each user type to simulate typical tasks performed during the course of a day as defined by Citrix. The following sections detail the requirements, assumptions and the process flow of each script.

## Requirements and Assumptions

The purpose of this section is to identify the functional requirements and assumptions used to develop and execute the Microsoft Office 2003 application scalability tests.

The following table lists the requirements identified by Citrix during information gathering session held during the initial phase of the project.

Requirement	Requirement Description
1	The scripts will simulate Standard Users and Power Users.
2	All user actions are automated.
3	Simulation should include at a minimum the normal task of opening a new file, entering data both as text and graphics, and saving the file for the most commonly used office applications these being defined by Citrix as Word, Excel, and PowerPoint.
4	New feature functionality may be tested were applicable.

**Table 4 - Custom Scripts Requirements**

The following table lists the assumptions identified by Citrix that were made during the information gathering sessions held during the initial phase of the project.

<b>Assumption</b>	<b>Assumption Description</b>
1	The scripts reflect the typical usage pattern and resource utilization for both user types defined as standard and power users.
2	The requirements provided to the Citrix project team reflects what users typically do within Microsoft Office.
3	User variance is based on the tasks that they execute not the speed that they use the application.
4	The network configuration for the testing environment is adequate and will not adversely affect the outcome of the testing.
5	The scripts do not simulate printing or file copying.
6	The user accounts are configured correctly.
7	The MetaFrame XP servers are optimally configured for the environment

**Table 5 - Custom Scripts Assumptions**

## Scripts

The objective of the scripts is to simulate the usage pattern of two typical Microsoft Office 2003 user types: A standard Word user running a published instance of Microsoft Word, and a power Microsoft Office user running a published desktop and running multiples instances of Word, Excel, and PowerPoint. The scripts have been broken down into three separate components the vUser\_init, the actions, and the vUser\_end. The diagram and workflow below shows the actions taking place for a vUser running the standard Word user script and is followed by the power Office user script diagram and workflow.

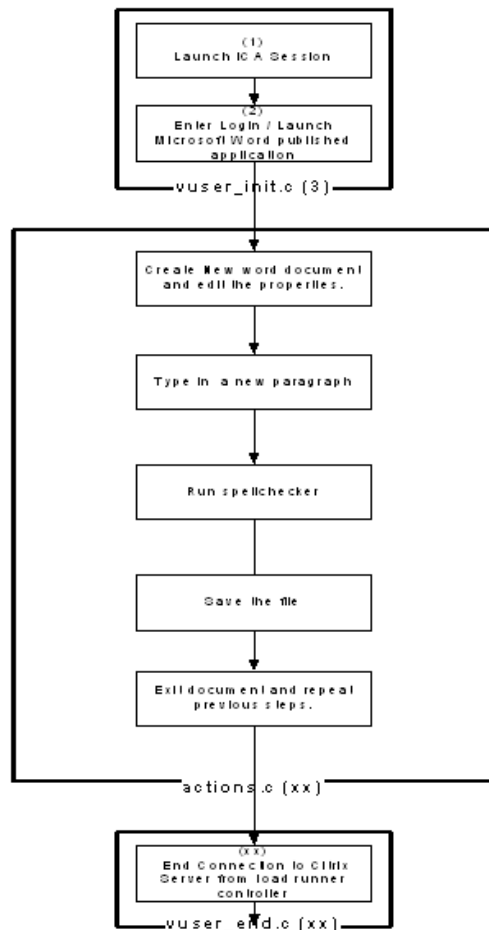


Figure 1 - Standard Word User Workflow

## Standard Word User Script Narrative

- The virtual user launches an ICA session for the Word published application.
- User creates a new file, choosing a blank presentation.
- User saves file as a unique file name.
- User types "http://www.citrix.com/support" (this link is not launched).
- User types a paragraph, being sure to misspell the word "replace".
- User launches spell checker and fixes misspelled word.
- User saves file again.
- User closes file.

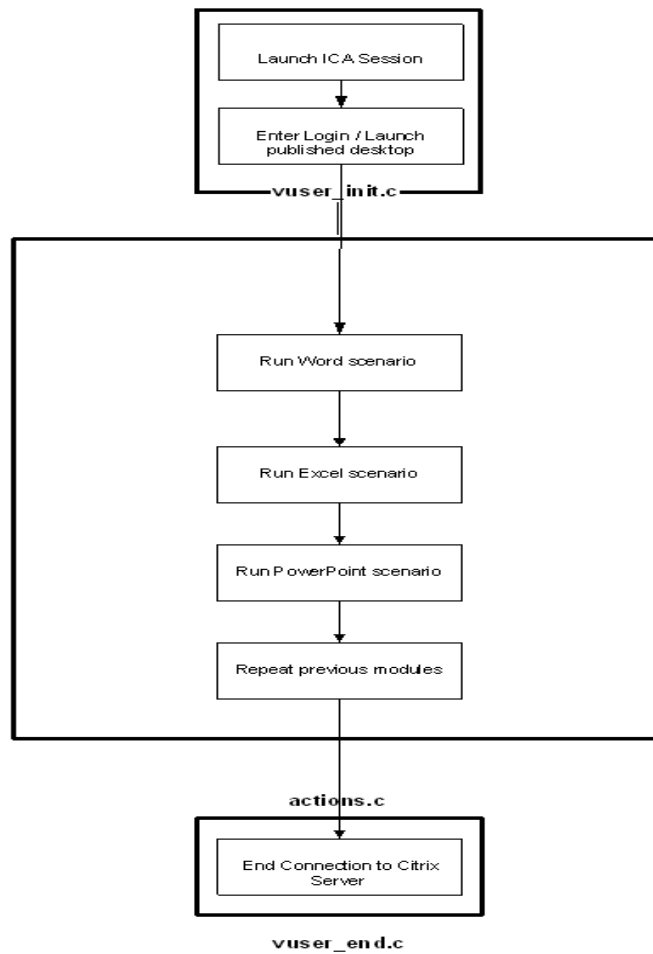


Figure 2 - Power Office User Workflow

## Power Office User Script Narrative

The virtual user launches an ICA session for the published desktop of the MetaFrame XP server.

### Word Scenario

- User starts Microsoft Word.
- User creates a new file, saves it as a unique file name.
- User types "http://www.citrix.com/support".
- User types a paragraph, being sure to misspell the word "replace".
- User launches spell checker and fixes the misspelled word.
- User saves file.
- User clicks URL, launching Internet Explorer.
- User clicks links in IE.
- User changes address in IE to "www.google.com".
- User closes IE.
- User closes Word.

### Excel Scenario

- User starts Microsoft Excel.
- User saves document as a unique name
- User types in the following data:

Time	Speed	Distance
1	=rand()*100	=rand()*100
2	=rand()*100	=rand()*100
3	=rand()*100	=rand()*100
4	=rand()*100	=rand()*100
5	=rand()*100	=rand()*100
6	=rand()*100	=rand()*100
7	=rand()*100	=rand()*100
8	=rand()*100	=rand()*100
9	=rand()*100	=rand()*100
10	=rand()*100	=rand()*100

- Highlight the data, and create a chart.
- Choose all default settings when creating the graphic.
- Save the file.
- Highlight the chart and make a copy of it in the clipboard.
- Exit Excel.

### **PowerPoint Scenario**

- User starts Microsoft PowerPoint.
- User created a new default presentation in PowerPoint and saves the file as a unique name.
- User clicks in the Title section of the presentation and adds the following title "Power User Test".
- User clicks in the bottom section and types, "This is only a test."
- User inserts a new slide, choosing the blank slide template from the menu on the right.
- User pastes the graphic copied from Excel into the PowerPoint document.
- User inserts a new slide, using the default template.
- User clicks on the title box and adds the following text, "Results".
- User clicks on the add text box and types the following, "Poor".
- User hits the Enter key and types, "Fair".
- User hits the Enter key and types, "Good".
- User hits the Enter key and types, "Excellent".
- User hits the Enter key and types, "Terrific".
- User saves the file and exits PowerPoint.

All actions repeat as specified in the scenario setup.

## Environment Specifications

This section describes the test environment and configuration of the scalability tests.

### MetaFrame XP Architecture Overview

This section provides an overview of the interaction between the components in the MetaFrame XP Scalability testing environment. The Hardware system components are identified in Figure 3 – MetaFrame XP Scalability Test Architecture Overview and described in the ensuing dialogue.

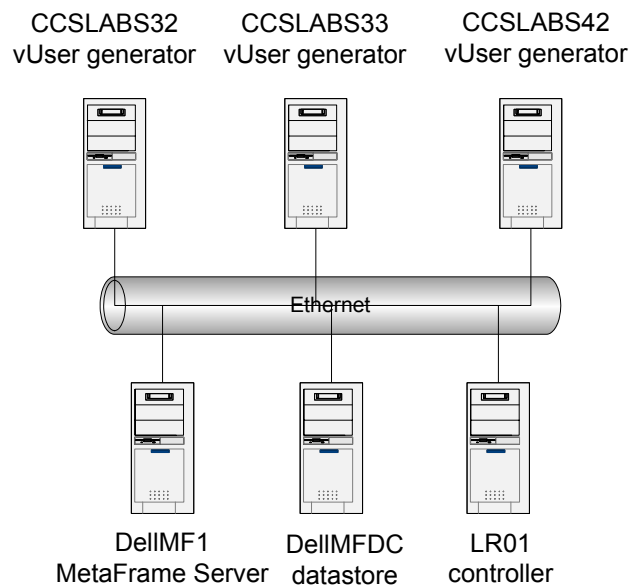


Figure 3 – MetaFrame XP Scalability Test Architecture Overview



## **LoadRunner Controller and vUser Generators**

Mercury LoadRunner enables testing and monitoring of MetaFrame XP based systems before and after deployment to ensure high performance, scalability, and availability. LoadRunner works by emulating large numbers of ICA client connections from just a handful of machines. Sessions are initiated from the LoadRunner Controller and dedicated vUser generators. The workflow scripts are executed on the controller and from clients' setup as load generators, simulating real life scenarios and generating realistic network and resource loads. Since scripts controlling user sessions are executed on the controller and client machines, no additional overhead is placed on the MetaFrame XP servers that might otherwise skew results.

The virtual users are created using a recording technology that captures the ICA traffic between the client and server into a high level, easy to read, maintainable test script. These scripts can be easily modified to represent real users with their own sets of data and replay speeds. By licensing key Citrix technology, LoadRunner's virtual users generate exactly the same traffic as the ICA client and "fool" the system into believing it is under the load of real users.

## **MetaFrame XP Presentation Server**

Citrix® MetaFrame XP™ Presentation Server is a server-based computing solution that enables robust, easily managed, and cost-effective delivery of applications to a great number of client devices. The ICA protocol supports nearly all types of hardware, operating platforms, network connections, and network protocols in general business use. Thus, organizations can deliver a common set of applications to different types of client devices and to users in separate locations with better performance than alternative technologies.

MetaFrame XP™ Presentation Servers are organized at the highest level into server farms. A Citrix server farm is a group of Citrix servers managed as a single entity with Citrix's Independent Management Architecture (IMA) components. Servers in such a farm share connectivity and a single IMA-based data store. MetaFrame XP servers use the data store to centralize configuration information for a server farm. The data store maintains information about the servers, applications, and Citrix administrators in the server farm.

Servers in a MetaFrame XP server farm are also further organized into zones. A zone is a logical grouping of MetaFrame XP servers. This grouping is intended to enhance the performance of the MetaFrame XP environment by allowing geographically related servers to be grouped together. Each zone within a server farm has one server that is designated as a Zone Data Collector. Data collectors store information about the servers and published applications in the farm from each server within its zone and all other Zone Data Collectors. This information is in turn used to dynamically load balance users to the least-busy server when connecting to a MetaFrame XP server farm.

## **SQL 2000 Database sever**

In this testing environment, the MetaFrame XP data store, the database volume, and the resource manager summary database resided on a SQL 2000 server. The data store information can be found in the data store database, while the resource manager metric information can be found in the resource manager database.

## MetaFrame XP Hardware Specifications

The following table outlines the hardware specifications for all the servers in the farm that were tested during the MetaFrame XP Office 2003 scalability testing initiative.

Server	DELLMF1	DELLMFDC
Purpose	MetaFrame XP FR3 hosting Office2003 suite	MetaFrame XP Zone Data Collector
Vendor	Dell	Compaq
Server Model	PowerEdge 1655MC Blade	DL360 G2
Processor Type	1.4 GHz	1.4GHz
Number of Processors	2	2
RAM	2,048 MB	2,048 MB
Partition Size c:	30 GB	30 GB
Partition Size d:	N/A	N/A
Partition Size e:	N/A	N/A
RAID Level	RAID 0	RAID 0
NIC Vendor	Broadcom	Compaq NC7780
NIC Speed	GB Ethernet set to 100Mb Full Duplex	GB Ethernet set to 100Mb Full Duplex

**Table 6 – MetaFrame Hardware Specifications**

## Supporting Hardware Specifications

Server	LR01	CCSLABS32	CCSLABS33	CCSLAB42
<b>Purpose</b>	LoadRunner Controller	LoadRunner vUser generator	LoadRunner vUser generator	LoadRunner vUser generator
<b>Vendor</b>	Compaq	Compaq	Compaq	Compaq
<b>Server Model</b>	DL360 G2	DL360 G2	DL360 G2	DL360 G2
<b>Processor Type</b>	1.4GHz	1.4GHz	1.4GHz	1.4GHz
<b>Number of Processors</b>	2	2	2	2
<b>RAM</b>	2,048 MB	2,048 MB	2,048 MB	2,048 MB
<b>Partition Size c:</b>	30 GB	30 GB	30 GB	30 GB
<b>Partition Size d:</b>	N/A	N/A	N/A	N/A
<b>Partition Size e:</b>	N/A	N/A	N/A	N/A
<b>Partition Size f:</b>	RAID 0	RAID 0	RAID 0	RAID 0
<b>NIC Vendor</b>	Compaq NC7780	Compaq NC7780	Compaq NC7780	Compaq NC7780
<b>NIC Speed</b>	GB Ethernet set to 100Mb Full Duplex	GB Ethernet set to 100Mb Full Duplex	GB Ethernet set to 100Mb Full Duplex	GB Ethernet set to 100Mb Full Duplex

**Table 7 – Support Hardware Specifications**

## Software and Platform Specifications

The following table outlines the software configuration for each component used during the scalability testing effort.

Component	Software															
<b>LoadRunner Controller/vUser Generators</b>	<ul style="list-style-type: none"> <li>• Windows 2000 Advanced Server with SP2</li> <li>• LoadRunner 7.51</li> <li>• Citrix ICA Client Integration with LoadRunner</li> <li>• Post-SP2 hotfixes:               <table style="margin-left: 20px; border: none;"> <tr> <td>✓ Q147222</td> <td>✓ Q296185</td> <td>✓ Q300972</td> </tr> <tr> <td>✓ Q276471</td> <td>✓ Q298012</td> <td>✓ Q301625</td> </tr> <tr> <td>✓ Q285156</td> <td>✓ Q299553</td> <td>✓ Q302755</td> </tr> <tr> <td>✓ Q285851</td> <td>✓ Q299687</td> <td>✓ Q303984</td> </tr> <tr> <td>✓ Q292435</td> <td>✓ Q299796</td> <td></td> </tr> </table> </li> </ul>	✓ Q147222	✓ Q296185	✓ Q300972	✓ Q276471	✓ Q298012	✓ Q301625	✓ Q285156	✓ Q299553	✓ Q302755	✓ Q285851	✓ Q299687	✓ Q303984	✓ Q292435	✓ Q299796	
✓ Q147222	✓ Q296185	✓ Q300972														
✓ Q276471	✓ Q298012	✓ Q301625														
✓ Q285156	✓ Q299553	✓ Q302755														
✓ Q285851	✓ Q299687	✓ Q303984														
✓ Q292435	✓ Q299796															
<b>MetaFrame XP Presentation Server FR3</b>	<ul style="list-style-type: none"> <li>• Windows 2003 Enterprise Server</li> <li>• MetaFrame XP Presentation Server with Feature Release 3</li> <li>• Office 2003</li> </ul>															
<b>SQL 2000 Database Server</b>	<ul style="list-style-type: none"> <li>• Windows 2000 Advanced Server with SP2</li> <li>• SQL 2000</li> </ul>															

**Table 8 - Environment Software and Platform Specifications**

## Citrix® MetaFrame XP™ Presentation Server Configuration

This section details the steps required for staging the MetaFrame XP servers in preparation for the scalability test. Because of the limited size needed for this particular test environment, a single MetaFrame XP server with a SQL 2000 database server in a single farm with a single zone was created. The Citrix XML service was configured to use port 80.

### ICA Connection Configuration

In order to ensure consistent test results the following ICA connection configuration settings were configured and validated on the MetaFrame XP servers.

#### Advanced Settings

- Timeout Setting for Connection = Inherit User Configuration
- Timeout Setting for Disconnection = Inherit User Configuration
- Timeout Setting for Idle = Inherit User Configuration
- Required Encryption = Basic
- Auto Logon = Inherit User Configuration
- Prompt for Password = No
- Initial Program = Inherit Client/User Configuration
- Only Run Published Apps = Not Selected
- User Profile Overrides = Not Selected
- On a broken or timed out connection = Reset
- Reconnect sessions disconnected = Inherit User Configuration
- Shadowing = Inherit User Configuration

#### ICA Client Settings

- Client Audio Quality = Medium (default setting)

#### Client Settings

- Connect Client Drives at Logon = Not selected / Disabled
- Connect Client Printers at Logon = Not selected / Disabled
- Default to Main Client Printer = Selected / Disabled
- Disable Client Drive Mapping = Selected
- Disable Windows Client Printer Mapping = Selected
- Disable Client LPT Port Mapping = Selected
- Disable Client COM Port Mapping = Selected
- Disable Client Clipboard Mapping = Not Selected
- Disable Client Audio Mapping = Selected

## Scalability Test Cases

The scalability tests have been designed to determine the number of Microsoft Office 2003 users that can be supported in a MetaFrame XP environment for a specific hardware configuration. The tests aim to answer the following question:

***How many users can be supported on a MetaFrame XP server in steady state<sup>1</sup> on a two-processor machine and what is the bottleneck, which caused the environment to stop scaling.***

The following sections discuss the test cases that were defined and executed.

### Case 1: No-Load Test

A no-load test was executed in order to determine the resource consumption of Performance Monitor and other system processes. Performance Monitor was used to gather data on the MetaFrame XP server; therefore, it was necessary to factor in the processing power and memory usage of this process in the test analysis. This testing also give a baseline for the resource consumption of Windows 2003 processes unrelated to the MetaFrame XP environment.

### Case 2: Baseline Tests

During the initial phase of the scalability testing initiative, a baseline test was completed to determine the baseline for a single user executing the common applications of the Office 2003 suite on a MetaFrame XP server. These baseline figures were then used to project the number of users that can be supported when executing each of the primary applications to be tested in Office 2003 for power Office users, and Microsoft Word for standard Word users.

### Case 3: Load Tests

The object of this test case was to determine the number of users that can be supported on a MetaFrame XP server given a predetermined hardware configuration. The test case was designed to measure system performance during execution of the standard word and power Office user scripts running each independently. This load was intended to simulate a production server running Office 2003 during a standard business day. In order to meet the requirements of the project, one test cycle was defined for each version of the test cases one for standard Word users and one for power Office users.

---

<sup>1</sup> Steady State is referred to as the state at which all users are actively accessing that application and no longer logging onto the system.

## Scalability Results and Analysis

The following sections detail the results obtained from the Office 2003 scalability testing performed on Dell blade 1655 dual 1.4GHz processor machines. The results are broken down by test type. Test types include a no-load system baseline, standard and power user baseline tests, standard and power user load tests, and usability testing.

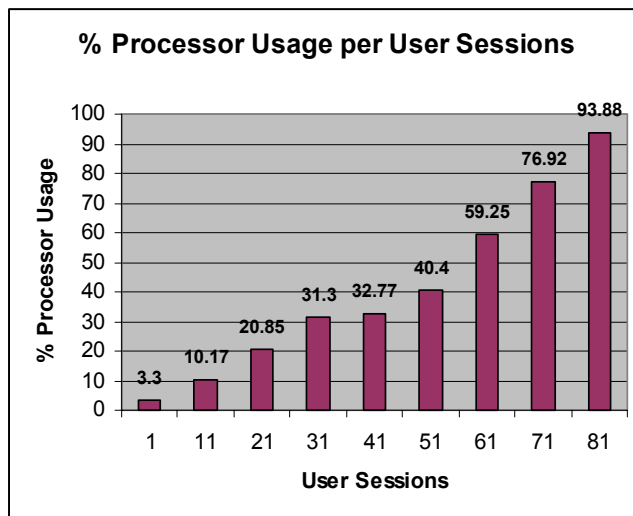
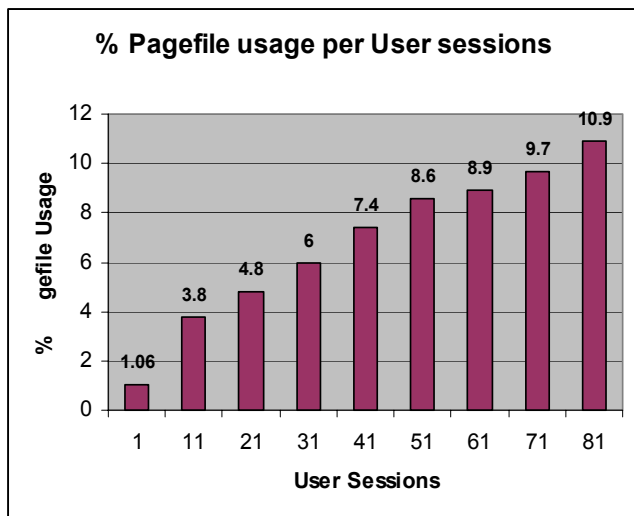
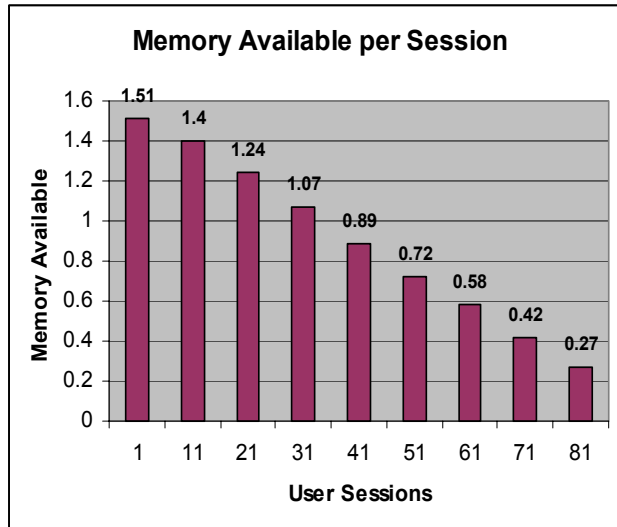
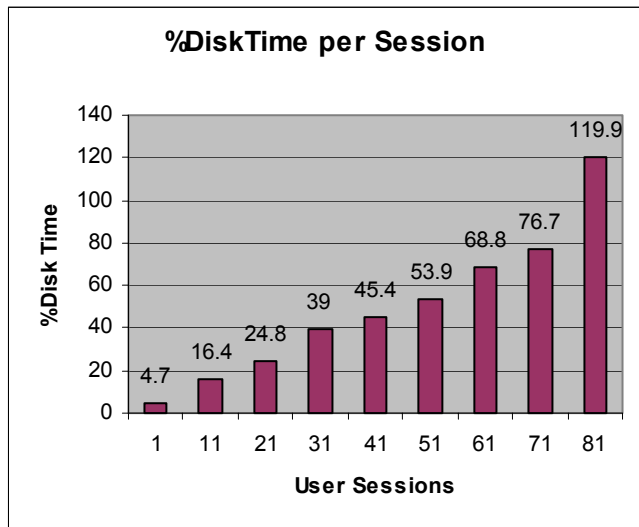
### Standard Word User Results

The following table contains the values for each counter that was averaged over steady states in each session interval for standard Word user testing. Each counter was measured every five seconds during the execution of the tests. The chart separates the data into different user session intervals.

# Of Sessions	Mem: Available GB	Mem: Page Faults/sec	Mem: Pages/sec	Pagefile: % Usage	Disk: % Disk Time	Disk: Queue Length	Proc: % Proc Time	Usability
Baseline	1.63	112.8	0.004	0.933	3.16	0.06	2.39	Excellent
1	1.51	131.3	0.05	1.06	4.7	0.03	3.30	Excellent
11	1.40	459.4	0.3	3.8	16.4	0.10	10.17	Excellent
21	1.24	604.6	0.5	4.8	24.8	0.63	20.85	Excellent
31	1.07	791.8	0.8	6.0	39.0	0.49	31.30	Excellent
41	0.89	1151.1	2.0	7.4	45.4	0.20	32.77	Excellent
51	0.72	1101.1	1.5	8.6	53.9	0.15	40.40	Excellent
61	0.58	1426.6	3.4	8.9	68.8	4.47	59.25	Excellent
71	0.42	1323.8	4.1	9.7	76.7	0.32	76.92	Acceptable
81	0.27	1866.8	14.3	10.9	119.9	0.65	93.88	Poor

**Table 9 – Standard Word User Maximum Concurrent Load Test Results**

The following graphically depicts the results obtained during testing for the primary counter and objects for memory, disk and processor for Standard Word users.



## Analysis

Citrix recommends that the results from the 80+ user observation be considered beyond the maximum level, as CPU usage would consistently remain over 80% for extended periods of time. The user experience factor was “Excellent” through 51 user sessions and performance was “Acceptable” up to 71 concurrent user sessions. However upon exceeding 71 user sessions, extended periods of CPU utilization over 80% would occur. Based on this, the recommended range of 70-75 “Standard Word” users is made, since extended periods of Total Processor Time over 80% can lead to system instability.

The user session interval between 71 and 81 standard word users should be the focal point of future scalability testing. It was between these intervals that larger processor spikes were observed.



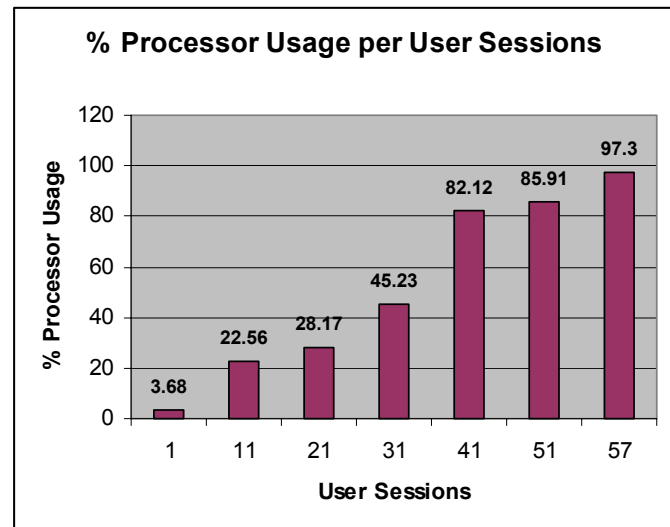
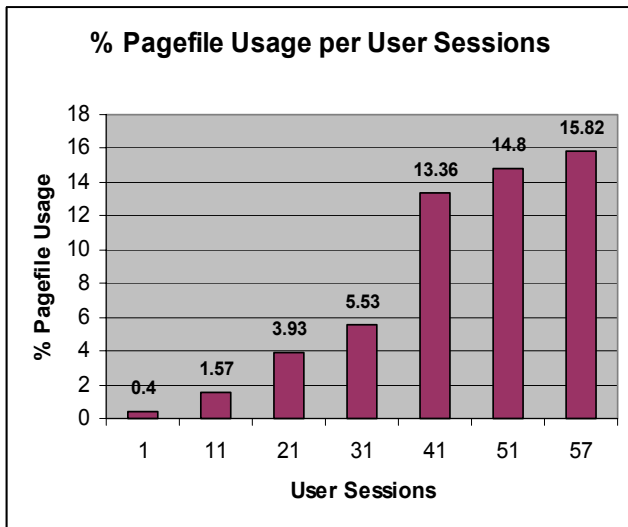
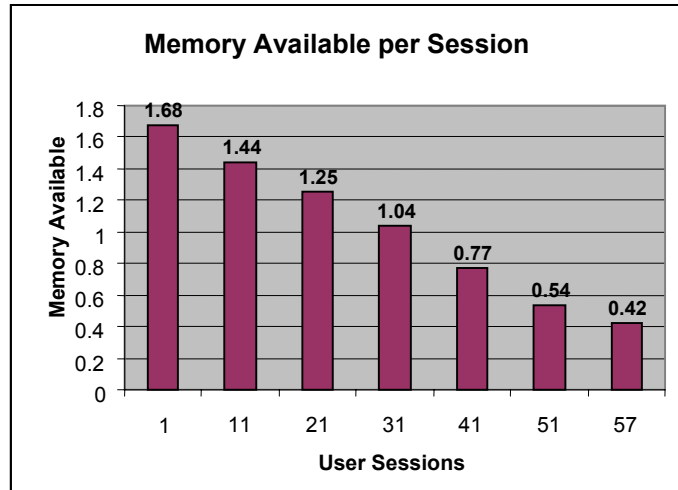
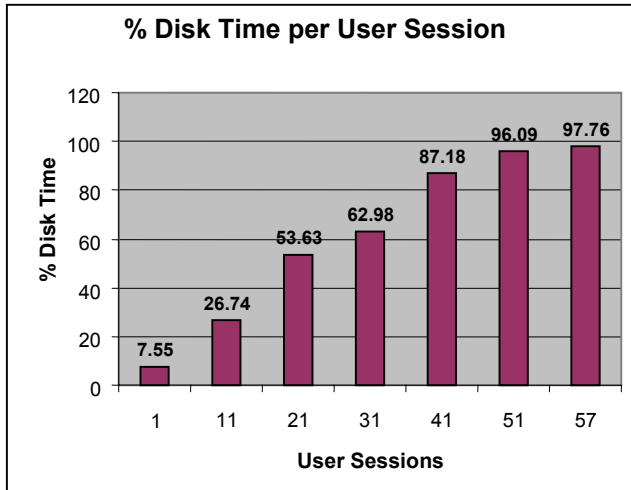
## Power Office User Results

The following table contains the values for each counter that were averaged over steady states in each session interval. Each counter was measured every five seconds during the execution of the tests. The chart separates the data into different user session intervals

# Of Sessions	Mem: Available GB	Mem: Page Faults/sec	Mem: Pages/sec	Pagefile: % Usage	Disk: % Disk Time	Disk: Queue Length	Proc: % Proc Time	Usability
Baseline	1.63	112.8	0.004	0.933	3.16	0.06	2.39	Excellent
1	1.68	301.02	5.29	0.40	7.55	0.07	3.68	Excellent
11	1.44	966.39	5.16	1.57	26.74	0.17	22.56	Excellent
21	1.25	2539.08	17.52	3.93	53.63	0.23	28.17	Excellent
31	1.04	3560.46	25.93	5.53	62.98	0.36	45.23	Excellent
41	0.77	5506.85	20.63	13.36	87.18	1.19	82.12	Acceptable
51	0.54	4903.54	26.15	14.80	96.09	0.34	85.91	Acceptable
57	0.42	4193.02	13.90	15.82	97.76	0.36	97.30	Poor

**Table 10 – Power Office User Maximum Concurrent Load Test Results**

The following graphically represents the results obtained during testing for the primary counter and objects for memory, disk, and processor for Power Office users:



## Analysis

Citrix recommends that the results from the 57+ user observation be considered beyond the maximum level, as CPU would consistently remain over 80% for extended periods of time. The user experience factor was "Excellent" through 40 user sessions and performance was "acceptable" up to 51 concurrent user sessions. However upon exceeding 51 user sessions, extended periods of CPU utilization over 85% would occur. Based on this, the recommended range of 45-50 "Power Office" users is made, since extended periods of Total Processor Time over 80% can lead to system instability. The user session interval between 41 and 51 Power Office users should be the focal point of future scalability testing. It was between these intervals that larger processor spikes were observed.

## Appendix A: LoadRunner

### vUser Script Generation

Citrix vUser scripts emulate the Citrix ICA protocol communication between a Citrix client and server. LoadRunner's VuGen enables easy creation of automated testing scripts by recording all activity during manual interactions with a MetaFrame application and creating a corresponding vUser script. The functions within this vUser script emulate the analog movements of the mouse and keyboard in an ICA session. In addition, these functions allow synchronization during the replay of the scripts used in a load test by waiting for bitmap changes, comparing bitmaps, or waiting for specific windows to open.

To provide an interface to the ICA Client APIs, the Citrix ICA Client integration must be installed on the LoadRunner Controller. To install the Citrix ICA Client integration:

1. Extract Citrix\_Headless\_Client.zip to the LoadRunner Controller.
2. Run setup.exe to install the Citrix ICA Client Integration with LoadRunner
3. Run "%Program Files%\Citrix\ICA Client\wfica32.exe /setup" to register the necessary objects in the Citrix ICA Client Integration.

Additionally, LoadRunner 7.51 must be updated to include functions that enable emulation of the ICA protocol:

1. Unzip LR\_For\_Citrix\_ICA.zip to the c:\mercury interactive\LoadRunner (assuming this is where you installed LoadRunner) directory. Overwrite any files if prompted. These files are available for download from the Mercury interactive Web site.
2. Execute c:\mercury interactive\LoadRunner\bin\citrix.bat. This will register all the necessary DLLs with the system.

One of the Citrix support files located in the LoadRunner\bin directory is called CitrixClientImpl.dll. If this file has a time stamp earlier than 09/02/2002, it must be replaced with the version below to support wildcard names for dialog boxes.

The high level steps for recording a vUser script using VuGen are as follows:

- Record the actions using VuGen.** Invoke VuGen and create a new vUser script, specifying Citrix as the type. Choose the application to record and set the recording options. Record typical operations in the Citrix session.
- Enhance the vUser script.** Enhance the vUser script by inserting transactions, rendezvous points, and control-flow structures into the script.
- Define parameters.** Define any parameters for the fixed-values recorded into the vUser script. By substituting fixed-values with parameters, the same business process can be repeated many times using different values.
- Configure the run-time settings.** The run-time settings control the vUser behavior during script execution. These settings include the pacing, logging, think time, and connection information.
- Run the vUser script from VuGen.** Save and run the vUser script from VuGen to verify that it runs correctly.

## Tips for Working with Citrix vUser Scripts

### Recording Tips

- When recording a session, make sure to perform the complete business process, starting with the connection and ending with the cleanup. End your session at a point from where you could start the entire process from the beginning.
- Configure the Citrix server to completely close a session. Open the Citrix Connection Configuration dialog box. Choose Start > Programs > Citrix > MetaFrame XP > Citrix Connection Configuration. Double-click on the ica-tcp connection name. The Edit Connection dialog box appears. Click on the advanced button. In the bottom section of the dialog box, clear the inherit user Configuration check box adjacent to the "On a broken or timed-out connection list box". Change the entry for this list box to reset.
- Record the connection process into the vUser\_init section, and the closing process in the vUser\_end section. This will prevent you from performing iterations on the connection process.
- Display settings of 1024 x 768 are recommended on the recording machine. This will allow the Citrix window, which is 800 x 600 to be displayed properly.
- When opening expanded menu options, click explicitly on each option do not depend on the expanding menu. For example, Start > Programs > Microsoft Word, be sure to click on the word Programs.
- If having trouble with window or dialog box names not being consistent, edit the script to use wildcards (\*). For example, use ctrx\_set\_window ("Spelling and Grammar \*").
- If you minimize the ICA Client during a recording or playback and subsequently stop the recording or playback, the ICA Client will remember its previous state when you start the next recording or playback. As a result, the ICA Client will appear in the taskbar, but no GUI for the connection will be displayed unless you right-click and select Maximize. Avoid minimizing the ICA Client unnecessarily during recordings and playbacks.

### Replay Tips

- To prevent overloading by multiple vUsers while connecting, set an initialization quota or use a ramp up schedule from the Controller's scheduler.
- For best results, do not disable think time in the run-time settings. Think time is especially relevant after ctrx\_set\_window and before ctrx\_type functions, which require time to stabilize.

### Debugging Tips

- You can view additional replay information in the extended log. You enable extended logging from the Run-Time settings (F4 Shortcut key) Log tab. You can view this information in the Execution Log tab, or the output.txt file in the script's directory. When running a LoadRunner scenario each vUser generates its own execution log located in  
C:\mercury interactive\LoadRunner\scripts\ScriptName\res\log\scriptname\_vUserID.log.
- When an error occurs, VuGen saves a snapshot of the screen to the script's output directory. You can view the bitmap to try to determine why the error occurred.

## Appendix B: Performance Metrics

The following sections provide summary information about the metrics, both objective and subjective, used to quantify performance during the scalability testing. Additionally, a standard table of performance monitoring thresholds for these metrics is provided in the subsequent section.

### Percent Processor Time

Percent Processor Time is the percentage of time the processor is busy handling non-idle tasks. When observing processor utilization, quick and sudden spikes are not a huge concern. Simple tasks, such as logging in, will cause the processor utilization to spike. Administrators should look for an upward trend in the total utilization percentage. On an idle MetaFrame XP server, typical processor utilization will hover around 0-10%. As more users connect to the server and begin to work, this percentage should slowly creep upwards on the performance monitor scale. Once the 'Percent Processor Time' reaches a sustained value between 70 and 80% or more, users might begin to notice performance degradation in the system.

### Processor Queue Length

Processor Queue Length is the number of threads that are waiting to be processed by the processor. Processor queue lengths are minimal, typically less than two. If the queue length is consistently above two, the processor is typically considered to be overloaded.

### Available and Committed Memory

Available memory indicates how much RAM is available for system processes, whereas committed memory is how much of the paging file has been reserved for future memory paging in case it is needed. By doing a proper benchmarking test of the system, one will get a better representation of how much memory, available and committed, is being used just by the operating system. By then monitoring the system with one user, and then subsequent numbers of users, an administrator should be better able to extrapolate how much memory a new user's session will utilize. By having a good estimate of this, one would be better equipped to identify how much memory a MetaFrame XP server will require to serve a certain number of concurrent users.

### Page Faults per Second

Page Faults/sec is the overall rate faulted pages are handled by the processor. It is measured in numbers of pages faulted per second. A page fault occurs when a process requires code or data that is not in its working set (its space in physical memory). This counter includes both hard faults (those that require disk access) and soft faults (where the faulted page is found elsewhere in physical memory). Most processors can handle large numbers of soft faults without consequence. However, hard faults can cause significant delays. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.

### Page Reads per Second

Page Reads/sec is the number of times the disk was read to resolve hard page faults. (Hard page faults occur when a process requires code or data that is not in its working set or elsewhere in physical memory, and must be retrieved from disk). This counter was designed as a primary indicator of the kinds of faults that cause system-wide delays. It includes reads to satisfy faults in the file system cache (usually requested by applications) and in non-cached mapped memory files. This counter determines the number of read operations, without regard to the numbers of pages retrieved by each operation.

## Free System Page Table Entries

Free System Page Table Entries is the number of page table entries not being used by the system. This counter displays the last observed value only; it is not an average. Windows 2000 can directly address up to  $2^{32}$  bytes or 4 gigabytes (GB) of memory address space, regardless of how much physical Random Access Memory (RAM) is installed. By default, 2 GB of this address space is allocated to each process, and 2 GB is allocated to the kernel. While separate 2 GB regions of address space are used for each process in the computer, most of the 2 GB kernel area is global and remains the same no matter which user-mode process is currently active.

The 2 GB of kernel area contains all system data structures and information. Therefore, the 2 GB kernel address space area can impose a limit on the number of system data structures and the amount of kernel information that can be stored on a computer, regardless of how much RAM is installed.

The two types of data that share a portion of this 2 GB address area are Paged Pool allocations, and kernel stack allocations. Paged Pool allocations are memory allocations made by kernel-mode components. Kernel stack allocations are stacks created in the kernel for each thread to use for making system calls. Paged pool allocations are made in the Paged Pool area and kernel stack allocations are made in the System Page Table Entry (PTE) area.

While these different allocations share the same area, the partition between them is fixed at startup. If the operating system runs out of space in one of those areas, the other area cannot donate space to it, and programs may begin to encounter unexpected errors. Therefore, when you encounter a Windows 2000-based computer that is experiencing unexpected errors or an inability to accept new logins, and the computer does not have some other resource limitation such as Central Processing Unit (CPU) or disk bottlenecks, it is highly likely that the Paged Pool or System PTE areas are dwindling. Because, by default, the System PTE area is sized to be as large as possible on a computer with Terminal Services enabled, the limitation will usually be due to insufficient Paged Pool address space. Fortunately on some computers, the System PTE area can be configured to be smaller, which can alleviate these symptoms and allow more users access to the computer.

The number of Free System PTEs varies depending on the memory configuration for the server. Typically, PTEs only cause bottlenecks on servers with a high number of processors (i.e., 4 or 8 processors).

## Percent Usage Page File

By monitoring paging file percent usage, a systems administrator should be capable of establishing certain baselines for system usage. The goal for a MetaFrame XP server is to under-utilize the paging file and over-utilize the memory, because using memory will have a significant speed improvement over a paging file. Hence, the usage percentage should never reach 100%. If it does creep up to a level close to this, chances are that the system is having a severe memory shortage. As the percent usage of the paging file approaches one hundred percent, the responsiveness of the MetaFrame XP server will be severely hindered by the slower speeds of the disks that house the paging file.

With the large amounts of RAM available on a MetaFrame XP server, if scaled correctly, the paging file should rarely approach 100% usage. If it does, the available memory is probably approaching zero. A possible solution to this problem is adding more RAM.

## Percent Disk Time

Percent Disk Time is the amount of time that the disk subsystem is busy trying to fulfill the requests to read or write data to or from the disks. This value is the sum of the percent disk time for all the disk drives in the system, and can therefore exceed 100%. For example: if there are 3 disks in the system and the utilization of each disk is 60%, 50% and 0%, the percent disk time would respond with 110%. However, the actual percent disk time is  $110/3$  or 37% busy.

## Current Disk Queue Length

Current Disk Queue Length is the current number of disk requests to do read and write processes. Typically, one should not see any queue lengths when examining the disk subsystem. It is very likely one will see the queue length hit one or two. However, when a sustained value of greater than two is consistently observed, the disk subsystem is quite possibly the bottleneck.



851 West Cypress Creek Road Fort Lauderdale, FL 33309

954-267-3000

<http://www.citrix.com>



Copyright © 2003 Citrix Systems, Inc. All rights reserved. Citrix, WinFrame and ICA are registered trademarks, and MultiWin and MetaFrame are trademarks of Citrix Systems, Inc. All other products and services are trademarks or service marks of their respective companies. Technical specifications and availability are subject to change without prior notice.