

MetaFrame XP Oracle 11i 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 U.S.A. All rights reserved.

Version History		
1.0	Celimaris Vega, Citrix Consulting	12/20/2002

TABLE OF CONTENTS

INTRODUCTION	1
KEY FINDINGS	1
DOCUMENT OVERVIEW	2
EXECUTIVE SUMMARY	3
OBJECTIVE	3
APPROACH	3
TEST SUMMARY	3
RESULTS AND COMMENTS	4
SCALABILITY TESTING METHODS	5
SCRIPTED TEST	5
REAL USER TEST	5
REAL USERS WITH TASKS LIST	5
COMBINATION	5
SCALABILITY TEST METHODS SUMMARY	6
ORACLE 11i SCALABILITY TESTING METHOD	6
LOADRUNNER AND CITRIX TESTING PROCESS	7
LOADRUNNER TERMINOLOGY	7
PLANNING THE TEST	7
CREATING vUser SCRIPTS	8
CREATING THE SCENARIO	8
RUNNING THE SCENARIO	8
MONITORING THE SCENARIO	8
ANALYZING TEST RESULTS	8
BENCHMARKING	9
PERFORMANCE METRICS THRESHOLDS	10
USER EXPERIENCE	10
CUSTOM ORACLE 11i SCRIPTS	12
REQUIREMENTS AND ASSUMPTIONS	12
ORACLE 11i - AR CUSTOMER & INVOICE TO GL SCRIPT	13

<i>Oracle Script Narrative – Initialization</i>	14
<i>Oracle Script Narrative – Actions</i>	14
<i>Oracle Script Narrative – Termination</i>	15
ENVIRONMENT ARCHITECTURE OVERVIEW	16
LOADRUNNER CONTROLLER.....	16
METAFRAME XP PRESENTATION SERVER.....	17
ORACLE BACK-END	17
HARDWARE SPECIFICATIONS	18
SOFTWARE AND PLATFORM SPECIFICATIONS.....	19
METAFRAME XP SERVER CONFIGURATION.....	20
INTERNET EXPLORER 6.0 CONFIGURATION	21
ORACLE 11i SCALABILITY TEST CASES	22
CASE 1: ORACLE 11i AR CUSTOMER & INVOICE TO GL BASELINE TESTS.....	22
CASE 2: ORACLE 11i AR CUSTOMER & INVOICE TO GL SUPPORTED USERS	22
<i>Steady State User Tests</i>	22
SCALABILITY RESULTS AND ANALYSIS	24
BASELINE TEST RESULTS.....	24
ORACLE 11i AR CUSTOMER & INVOICE TO GL PERFORMANCE RESULTS.....	26
SCALABILITY TEST CONCLUSIONS	30
APPENDIX A – LOADRUNNER SCRIPT GENERATION	31
APPENDIX B – PERFORMANCE METRICS	32
PERCENT PROCESSOR TIME.....	32
AVAILABLE AND COMMITTED MEMORY	32
PAGE READS PER SECOND.....	32
FREE SYSTEM PAGE TABLE ENTRIES.....	32
PERCENT USAGE PAGE FILE.....	33
PERCENT DISK TIME	33
CURRENT DISK QUEUE LENGTH.....	33
BYTES TOTAL PER SECOND	33
USER EXPERIENCE	34

Introduction

Oracle 11i E-Business Suite is a complete set of business applications or modules, which enable companies to efficiently manage and integrate processes. These processes include every aspect of the business; customer relationship management, finance, human resources manufacturing, supply chain management, asset management and project management, among others. Oracle 11i provides business automation across an enterprise because all applications that comprise the suite are engineered to work together.

Citrix Consulting conducted a server scalability and bandwidth testing project for Oracle 11i, specifically for the Accounts Receivable module. All testing was completed using Citrix MetaFrame XP Presentation Server, Enterprise Edition software. The result of this provided valuable information required to compare the implementation of Oracle 11i through a Server Based computing model vs. a standard client/server model. The testing provided details on server resource utilization and sizing estimates for an implementation of Oracle 11i through a MetaFrame XP server farm.

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

- MetaFrame XP with Feature Release 2
- Oracle eBusiness Suite version 11.5.7
- Oracle RDBMS version 8.1.7.3.10
- Oracle Jinitiator 1.1.8.19
- Microsoft Internet Explorer 6
- Microsoft Performance Monitor as the primary server monitoring and data capture tool
- Windows 2000 Server with Terminal Services
- Mercury Interactive's LoadRunner 7.51
- Citrix ICA Client integration with Mercury Interactive's LoadRunner
- A functional process flow focusing on the Oracle 11.5.7 Accounts Receivable module functionality prepared by CPG Solutions
- Oracle 11i Test Scripts created by Citrix consulting for Mercury LoadRunner

Key Findings^{*}

- **Primary Scalability Factors.** CPU and Memory were the primary server resource constraints identified through the evaluation process. Contributing to load limitations, memory utilization represented the secondary factor.
- **CPU.** Average CPU utilization was found to be an average of 2.4% per user throughout all test scenarios. The CPU utilization fluctuated depending on the functions being performed in the script. Some functions required low CPU while others generated significant spikes in CPU utilization that were magnified as more users were added. As seen in the data analysis, it is those spikes in utilization that limited the number of concurrent users/sessions to 20-25 users on the two-processor system, with CPU utilization hovering between 63 and 80%. Peak utilization was reached at 45 users, which caused the CPU to spike to 100% utilization for over eight minutes, after which over 10% of the user transactions started to fail.

* The scripted scalability tests used in this analysis are helpful in obtaining rough estimates when sizing the server. The data and analysis in this document should serve as a guide, and not definitive numbers, for further MetaFrame XP scalability testing for Oracle 11i. 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 Oracle 11i application 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.

- **Memory.** Memory was found to average at 54Mb per user throughout all test scenarios. Testing indicated a useful maximum production load between 25 and 30 users. Upon 45 concurrent users, the utilization reached slightly more than 2 GB of memory committed for the application.
- **User Limits.** Maximum observed server load was 45 users on the two-processor system. A limit of 35 concurrent users was observed without affecting user experience. For the purposes of this test, the more conservative figure of 25 users reflects tests with minimal transaction errors. These are reported as the expected **conservative** production load limit.
- **Internet Explorer Configuration.** The configuration of the MetaFrame XP servers was absolutely minimal, as all testing was effectively performed through a published Internet Explorer application. When launched, the browser immediately pointed to the portal login page for the Oracle 11.5.7 application. There is only one critical caveat: the browser / user profile must be set to allow the download and execution of ActiveX controls. This will permit the downloading of the Java applet when the user initially opens a browser and connects to a particular Oracle module.
- **Subjective Testing Results.** The primary focus of the testing was to evaluate objective factors such as memory and utilization. To ensure functional validity in the testing, the test team performed logins and limited manual application execution as a subjective test to evaluate application response and usability. Throughout the entire range of testing, the performance remained acceptable, with the exception of the downloading of the Java applet during high user load times. It was noted that the downloading process took over three minutes during the test cases with over 40 concurrent users. Aside from this, no other failures or issues in application delivery or processing were noted.
- **Test Procedures.** The project team relied heavily on Mercury Interactive's 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; the script represented an extremely close approximation for a production user executing the Oracle 11i Accounts Receivable application via MetaFrame XP. Furthermore, LoadRunner Scenarios were designed in a way that ensured all test users would be executing part of the script at the same time. For more information, please refer to the [Creating the Scenario section](#).

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.
- [Performance Metrics Thresholds](#) – Identifies the common thresholds for the performance metrics.
- [Custom Oracle 11i Scripts](#) – Discusses the requirements, assumptions and the process flows behind the custom scripts used in the scalability testing.
- [Environment Architecture Overview](#) – Details the MetaFrame XP architecture and configuration, server hardware, software and platform specifications, and all other configurations required to set up the MetaFrame XP testing environment.
- [Oracle 11i Scalability Test Cases](#) – Discusses the cases that were tested during the scalability testing efforts.
- [Scalability Results and Analysis](#) – Presents and analyzes the meaning of the results obtained from the testing effort.
- [Scalability Test Conclusions](#) – 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.
- [Appendixes](#) – The appendixes provide supplemental information for the content provided in the main body of the document.

Executive Summary

Objective

The purpose of this test process was to determine overall resource consumption on a MetaFrame XP server when running the Oracle 11i EBusiness Application Suite using Microsoft Internet Explorer 6.x. These resources would be the primary factors in determining the total number of concurrent users that could be supported in a production environment. Because of the significant functionality available within the Oracle 11i application, the most appropriate controlled test was to implement the Accounts Receivables module of such application.

Approach

The focus of this testing was on the Oracle 11i Accounts Receivable module. The scalability testing applied by Citrix consulting was intentionally 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. To further ensure validity, a second user was run concurrently to provide correlating factors.
- **Steady state scenario.** A Mercury Interactive LoadRunner scenario designed to place an increasing number of users on the system performing effectively the same functional tasks concurrently. This was accomplished by loading one user every 10 seconds after all users have logged into the system.

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, condensed fashion. To further ensure accuracy and relevancy in individual client environments, Citrix consulting strongly recommends that all scalability testing follow a standard process:

- Evaluate the application and/or client components in a stand-alone environment. Single session and/or single workstation.
- Create a baseline scripted test scenario to evaluate multiple concurrent sessions and their impact on system performance.
- Evaluate the user environment and generate more complex and customized scripted test scenarios to better evaluate the application's performance in a production simulation environment.
- Deploy a proof-of-concept to a small user subgroup and monitor their testing and use of the application.
- Finally, when deploying the application, deploy 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.

Test Summary

- **CPU.** One of the primary system resource constraints during testing was overall CPU utilization. The threshold was reached between 20-25 concurrent sessions.
- **Memory.** The second bottleneck was Memory. At 2GB RAM memory became a concern upon reaching 25-30 users.
- **Reduced Think Time.** Reducing the think time of the script increases the probability that the application and the script would become un-synchronized resulting in errors.

- **Users.** The test scripts and process used to determine when the server was 'loaded' were designed to provide conservative figures. Actual testing with a group of production users may indicate the ability to add additional users, depending on the actual mix of user activities.

Results and Comments

The number of users, intensity of their work and applications that they are using are the most significant factors in affecting the performance of a server. This testing effort and all performance data gathered was based on a set of requirements and assumptions that was determined to accurately simulate user activity for the Oracle 11i Accounts Receivable module of the Oracle 11i EBusiness Application Suite. The following table outlines the results and conclusions of the Oracle 11i scalability testing effort. Note that Maximum Concurrent Users value is the maximum number of users Citrix consulting was able to login to the server and run the scripts successfully.

Hardware	System Bottleneck	Maximum Observed Concurrent Sessions	Maximum Concurrent Sessions Without Degradation	Recommended Concurrent Sessions
Two-Processor 1.4 GHZ with 2GB of RAM	% Processor Time % Available Bytes	45	35	25

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. Per Citrix consulting methodology, Server Scalability is performed during a Proof of Concept, or a Pre-Production Pilot. Server scalability is different from system scalability testing which measures the ability for the overall infrastructure (of which MetaFrame XP is one component) 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.

Therefore, the Oracle 11i application scalability tests are server scalability tests and were geared towards determining the maximum number of Oracle 11i Accounts Receivable users that two-processor servers can support given defined hardware constraints. Please refer to the [Hardware 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 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. It is also important to note that this scripted scalability analysis focused primarily on the determining of a conservative baseline figure for the amount of Oracle 11i Account Receivable users that can be successfully supported on a MetaFrame XP server given defined hardware constraints.

The scripted scalability tests used in this analysis are helpful in obtaining rough estimates when sizing the server. The data and analysis in this document should serve as a guide, and not definitive numbers, for further MetaFrame XP scalability testing for Oracle 11i. 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 Oracle 11i application 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/performance test of Windows 2000 server with MetaFrame XP, decisions regarding the test methods must be made in order to standardize valid testing of the environment.

There are four primary types of scalability testing that are appropriate to the MetaFrame XP 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 sections discuss each method in more detail and compare the advantages and disadvantages of each method. The approach used for Oracle 11i testing is discussed once each method has been defined.

Scripted Test

A scripted test that includes a standard set of scripts could be written to control the actions of test users. Test user scripts are created and configured similar to typical Oracle 11i users. These scripts 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 configured to stagger events in order to ensure a realistic use of the application, therefore achieving more accurate results.

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 towards 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, a large enough user base must be obtained in order to perform these tests, a difficult task given the decline in productivity that could result.

Real Users with Tasks List

Real User Testing with Task Lists includes having real users access the system, while executing a written set of tasks. 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 in 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, 30 users can be simulated from one client computer in conjunction with several Real Users performing searches and more complex customer transactions.

Scalability Test Methods Summary

The following table summarizes the advantages and disadvantages of each scalability test method described above.

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

Oracle 11i Scalability Testing Method

Based on the project requirements, Citrix consulting chose to leverage the **Combination Approach**, of Scripted Test and Real Users with Task List to ensure identical, controlled tests that can be repeated by other Citrix consultants for different application configurations and/or server specifications at a later date. Also, including real user testing would enable the project team to establish subjective performance assessments. The scripts developed for testing have been carefully formulated by Citrix consulting and are aimed at accurately simulating normal user load on the MetaFrame XP servers. To ensure that figures obtained from the test were conservative, and to avoid overestimating user load, the Citrix consulting project team focused on a standard user scenario. For more information, please refer to the section entitled [Creating the Scenario](#).

The primary testing tool that was used to develop the custom Oracle 11i scripts was Mercury Interactive’s LoadRunner and the Citrix ICA client integration. For more information regarding Mercury Interactive’s LoadRunner and the partnership between Citrix and Mercury Interactive please refer to Mercury Interactive’s web site. (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.

LoadRunner Terminology

The following section defines Mercury Interactive's LoadRunner terms that will be used frequently throughout the following sections.

- vUser – Virtual user that runs through the recorded script and simulates real user actions in the system.
- Scenario – Testing session events. A scenario includes defining the client machines that vUsers will run on, scripts that will be executed, and a specified number of vUsers or vUser groups that run the scenario. Scenarios are created using the LoadRunner Controller.
- vInit – Portion of the script that contains initialization procedures. This section of the script is only executed once for every vUser in a scenario.
- Actions – Portion of the script that perform the significant events to be executed in the applications. vUsers iterate through this portion as many times as pre-configured at the Scenario level.
- vEnd – Portion of the script that contains all actions required to logoff and exit the application. This section of the script is only executed once for every vUser in a scenario.

The following sections are an overview of the six-step process for testing applications under load using Mercury Interactive's LoadRunner.

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 vUser workflows, selecting vUsers, and choosing test hardware.
- Define the launch sequence of the test users.
- Monitor the server with no activity prior to test execution. This will allow the project team to acquire proper benchmark (see the [Benchmarking](#) section for additional information).
- After that 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.
- 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 the system measuring user experience to validate the performance of the system. For more information refer to the section entitled [User Experience](#) 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 this document, see the [Performance Metrics Thresholds](#) section.
- After investigating the results of the initial scalability and performance tests, some performance tweaks 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 losses are detected or the changes result in no increase in performance, the changes should be removed.

Creating vUser Scripts

Before creating a script it is important to determine what type of user the vUsers will emulate. Users can be classified at various levels of server usage. For example, users can be light, medium, or power users. A light user typically uses only one application for their daily tasks, probably logs on once in the morning and remains logged on until they go home, and performs minimal processing within the application including data searches and minimal data entry. On the other extreme, a power user typically requires access to two or more applications while at the same time accessing email, performing data analysis (which required more server resources than a search or data entry), running reports, creating graphs of these reports and possibly even creating presentations. A medium user would fall within these two extremes. For example, they might require access to one to two applications at the same time mostly entering data, performing searches and periodically running or printing a report. For this scalability test, a medium user has been selected.

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 will accomplish load simulation. The simulation becomes even more realistic when “think” (wait) time is added between actions. For instance the script can be configured to wait 2 seconds between the time the vUser clicks the Find button and the time the Find window is displayed. Think time configuration is estimated based on acceptable user response times for the type of user being simulated.

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, 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 executed, 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.

Throughout execution of a scenario, the Performance Monitor tool was used to record the performance of the system under test at different load levels.

Analyzing Test Results

The data gathered using Performance Monitor was gathered in CSV format and organized into meaningful graphs using Excel. These graphs provided a good context for analyzing, reporting and archiving results.

Benchmarking

To get a better result set from the scalability and performance tests for the MetaFrame XP server 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 Oracle 11i application requires.

Performance Metrics Thresholds

Performance Monitor counters were used to gather information about system resource utilization during the tests. For a brief description of each of the selected Performance Monitor counters reference [Appendix B – Performance Metrics](#).

Below is a table of standard thresholds that should be captured when monitoring system performance. These are very general statements, and they are used as a basis to help identify points at which performance bottlenecks could occur.

	General Problem Value	General Cause	Other Possible Factors
Percent Processor Time	>= 70 – 80% Consistently	Processor Bottleneck	Disk Bottleneck, Memory Paging / Faults, and / or Application type / coding
Memory – Available Bytes	< 30% of RAM	Memory Bottleneck	-
Memory – Committed Bytes	> 85% of Page File	Memory Bottleneck	-
Memory – Page Reads/sec	> 5	Memory Bottleneck	Application type / coding
Memory – Free System PTEs	Varies*	Large number of users / applications	Multi-processor systems
Page File – Percent Usage	85%	Memory Bottleneck	Disk Activity
Physical Disk - % Disk Time	40%	Disk Bottleneck	Memory
Current Disk Queue Length	>= 2 Consistently	Disk Bottleneck	Memory
Server – Bytes Total/Sec	Approaches Max for network	Network Bottleneck	Application type / coding
User Experience	'Acceptable'	-	-

User Experience

In addition to the Performance Monitor metrics mentioned above, user experience was calculated. 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

* 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. Also, thread context switches should be observed along with other counters, such as CPU utilization. If an upward trend on CPU utilization can be correlated with an increased number of context switching, inefficient code running on the server may be the culprit.

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.

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

Custom Oracle 11i Scripts

This section details the custom scripts developed for the Oracle 11i application scalability initiative. Scripts were created to simulate typical tasks performed daily by Oracle 11i users as defined by CPG Solutions consultants. The functional script selected for scalability testing was the AR Customer & Invoice to GL functional process flow, as it was one of the most robust. 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 Oracle 11i Financials application (specifically the Accounts Receivable module) scalability tests. The table below lists the requirements that must be met by the Oracle 11i LoadRunner Scripts.

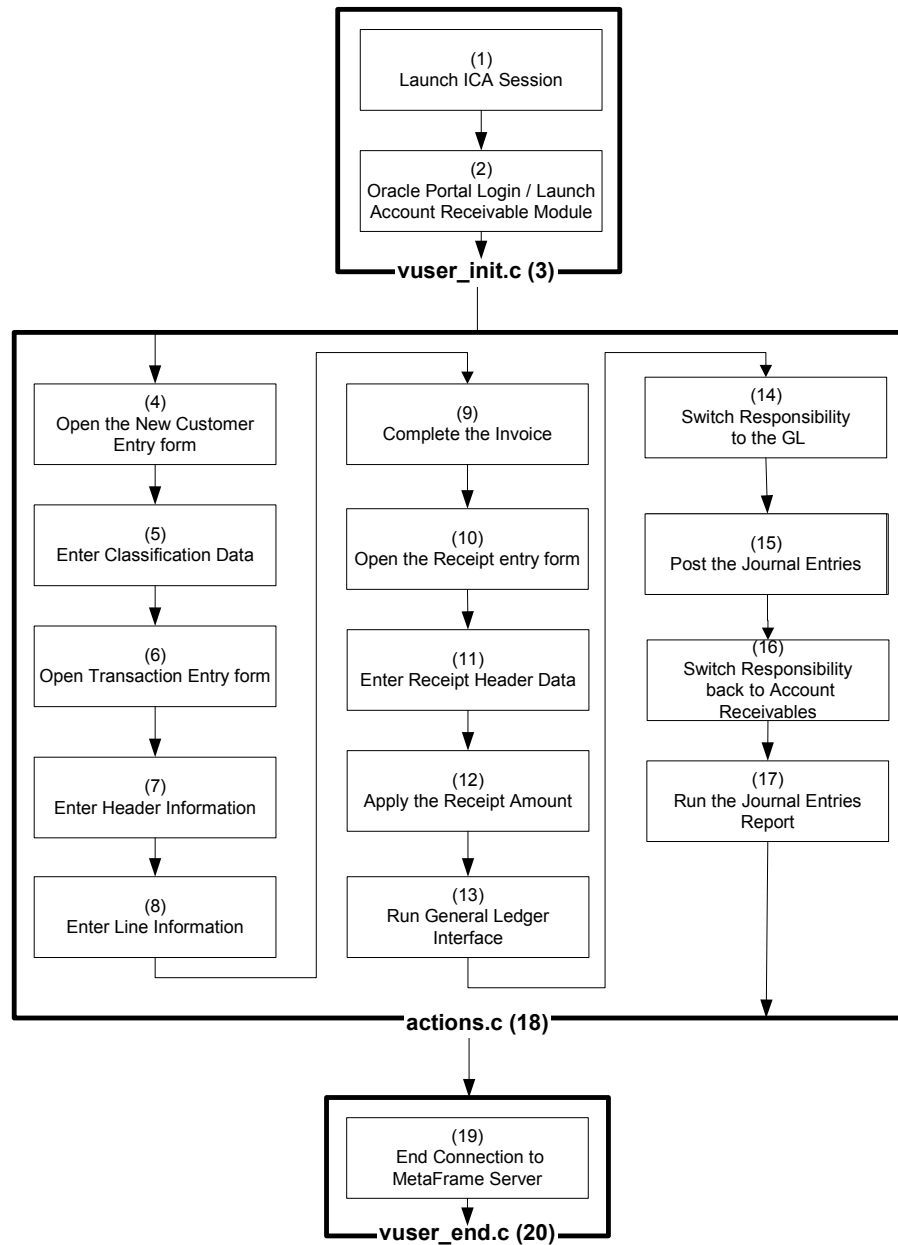
1. The AR Customer & Invoice to GL scripts simulates Standard Users.
2. All user actions are automated.
3. Oracle 11i simulation includes the normal task of adding a database entry, posting entry to the General Ledger (GL) and running a Summary Report.
4. Authentication process is imitated.
5. Initialization procedure is excluded from load analysis.

The following lists the assumptions identified by Citrix consulting during the initial phase of the engagement.

1. The AR Customer & Invoice to GL script reflects the typical usage pattern and resource utilization of the rest of Oracle 11i modules.
2. User variance is based on the speed that they use the application and not the tasks that they execute.
3. The network configuration for the testing environment is adequate and will not be a bottleneck.
4. The scripts do not simulate printing or file copying.
5. The user accounts are configured correctly.
6. The Oracle 11i and MetaFrame XP servers are optimally configured for the environment.

Oracle 11i - AR Customer & Invoice to GL Script

The objective of this script is to simulate the usage pattern of a typical Oracle 11i user. The script has been broken down into three separate components the vUser_init, the Actions, and the vUser_end. Each component of the script has been broken down further into transactions to modularize the script and help track the progress of the script during the testing cycles. This script is depicted in the diagram below (each number in the diagram corresponds to a transaction).



A detailed description of each transaction is included below:

Oracle Script Narrative – Initialization

1. The virtual user launches an ICA session using a LoadRunner generated user ID ranging from "lbus01" to "lbus99". The virtual user launches the published Oracle JInitiator application and authenticates to Windows using "lbus01" as the user ID and a hard-coded password.
2. Once the Oracle JInitiator application starts, the login window is displayed. The vUser_init component uses the generated ID's to login to the Oracle application. After Oracle authentication the Oracle 11i portal window is displayed. The virtual user clicks on Receivables, Vision Operations (USA).
3. The vUser_init component of the script is complete and initializes the action component of the script.

Note: When analyzing the results, the team ignored the performance data during the initialization phase of all the scenarios. It was determined that in a production environment these actions would probably be performed once per day for each user; therefore, aggregating that load may skew the results and would not provide an accurate representation of standard Oracle JInitiator application load.

Oracle Script Narrative – Actions

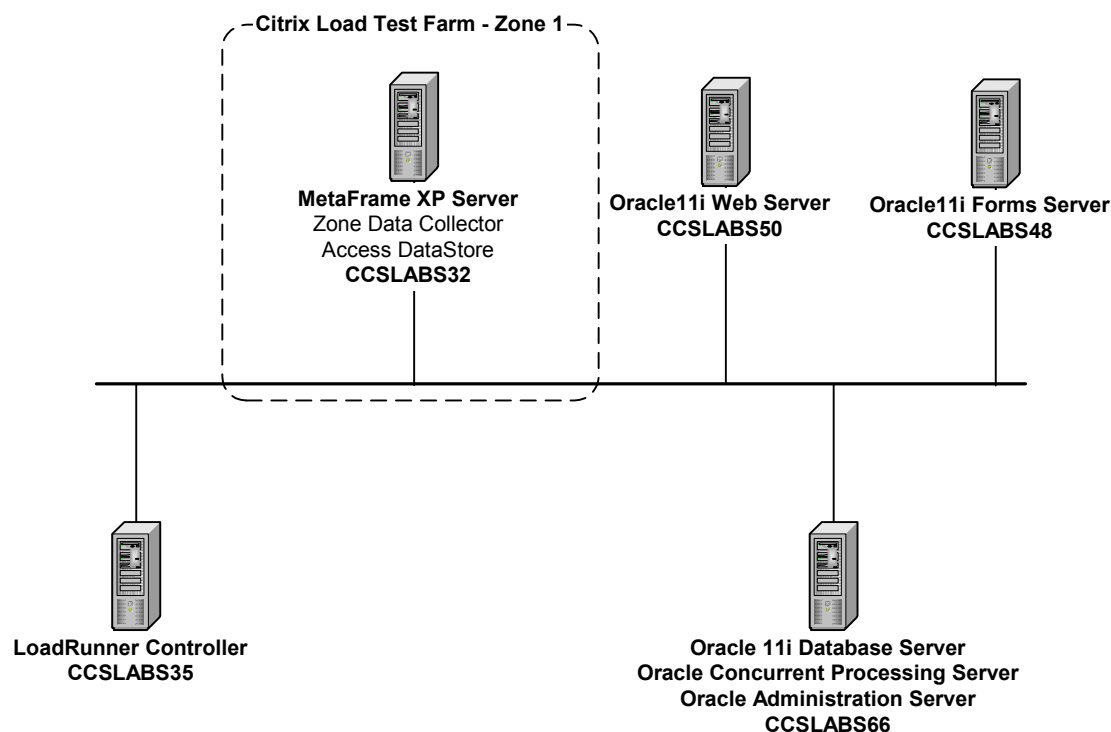
1. The virtual user navigates the functions menu to Customer and selects Standard. Then the virtual user enters a unique customer name. LoadRunner generates the customer name by appending a time stamp (in millisecond) to ensure unique contact name per virtual user and per script iteration. The virtual user generates a new customer by clicking the "New" button.
2. The new Customer window is displayed and the virtual user classifies the customer as a High Technology customer. The virtual user will also assign them an address for both billing and shipping.
3. The virtual user exits the Customer function and navigates to the Transactions function.
4. The virtual user classifies the transaction as a "Manual" one. Then he assigns the newly created Customer to that transaction.
5. The virtual user clicks the line item button to enter transaction detail information. After entering the line item information, the virtual user saves the transaction. Then, the virtual user closes the window and copies the transaction number that was automatically generated to the clipboard.
6. The virtual user clicks on the Complete button to complete the invoice.
7. The virtual user navigates to the Receipts function.
8. The virtual user enters a receipt number. LoadRunner generates the receipt number by using a random number of any 3 digits. The virtual user enters the receipt amount and payment method and pastes the transaction number from the clipboard to associate the receipt to the previously created transaction.
9. The virtual user clicks the Applications button and applies the amount of the receipt to the invoice.
10. The virtual user navigates to the General Ledger function and enters a start date to run the GL and clicks the Submit button. LoadRunner generates the date based on the current date (in the following format "DD-MMM-YYYY"). After the virtual user submits the GL request the virtual user refreshes the request list until it has been completed.
11. The virtual user switches responsibilities to General Ledger, Vision Operations (USA).
12. After the switch, the virtual user navigates to Journals/Post and posts the previous receivables batch.
13. When the batch is posted, the virtual user switches responsibilities back to Receivables, Vision Operations (USA).
14. The virtual user submits a new request for "Journal Entries Report". The virtual user searches for the report request and refreshes the data until the request is completed. After the request is completed, the virtual user views the output (report). The virtual user closes the report and the Requests window.
15. The action component of the script completes and the vUser_end component of the script initializes.

Oracle Script Narrative – Termination

16. The virtual user closes the Oracle JInitiator application and logs out of the system.
17. The vUser_end component is complete and the script is finished processing.

Environment Architecture Overview

This section describes the test environment and configuration of the hardware and software used to conduct the Oracle 11i AR Customer & Invoice to GL scalability tests. It also provides an overview of the interaction between the components in the MetaFrame XP scalability testing environment. The components are identified and described in the ensuing diagram and described further in the following sections.



LoadRunner Controller

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 workflow scripts are executed on client machines, simulating real life scenarios and generating realistic network and resource loads. Since scripts controlling user sessions are executed on the 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 the world's most widely deployed presentation server for centrally managing heterogeneous applications and delivering their functionality as a service to workers, wherever they may be and to a variety of client devices. Because the ICA protocol, which enables this application access, supports all types of hardware, operating platforms, network connections, and network protocols, it lets organizations 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 servers are organized at the highest level into server farms. A MetaFrame XP server farm is a group of MetaFrame XP servers managed as a single entity with Citrix's Independent Management Architecture (IMA) components. Servers in such a farm share some form of physical connection and a single IMA-based data store. MetaFrame XP uses the data store to centralize configuration information for a server farm in one location. The data store maintains information about the servers, applications, and MetaFrame XP 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 intended to enhance the performance of MetaFrame XP 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. A zone data collector stores 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.

For testing purposes, the MetaFrame XP environment was composed of one server with a local Microsoft Access database acting as the data store. This server was also acting as the zone data collector.

Oracle Back-End

Following is a description for each of the Oracle back-end components.

- **Oracle Web Server** - The Oracle Web Server provides connectivity for the Oracle back-end through a web interface called JInitiator. For this test, the Oracle system is accessed with a web client via Internet Explorer 6 hosted on the MetaFrame XP server. This server runs Oracle 9iAS Apache Web Server.
- **Oracle Forms Server** - The Oracle 11i application server that generates Oracle forms.
- **Oracle Database Server** - The Oracle Database Server runs the back-end database that supports the Oracle 11i application and contains all company-specific data as well as metadata that supports the application. It runs Oracle RDBMS 8.1.7.3.10.
- **Oracle Concurrent Processing Server** - The Oracle Concurrent Processing Server provides the batch-processing component that generates reports as well as runs data processing transactions, both scheduled and immediate.
- **Oracle Administration Server** - The Oracle Administration Server provides access to the administrative settings for Oracle 11i.

In order to split the major processing components, the Oracle environment was broken down into three pieces. One server acting as a web server, a second server acting as a forms server and a third and more powerful server acting as a database, concurrent processing and administration server.

Hardware Specifications

The following table outlines the hardware specification for the MetaFrame XP servers that were tested during the MetaFrame XP scalability testing initiative.

Server	CCSLABS32
Purpose	MetaFrame XP server
Vendor	Compaq
Server Model	Proliant DL360 G2
Processor Type	1.4 GHz
Number of Processors	2
RAM	2 GB
Partition Size c:	16.9 GB
NIC Vendor	Compaq NC7780 Gigabit Server Adapter
NIC Speed	Auto Detect

The following table outlines the hardware specifications and functionality of the remainder of servers in the environment.

Server	CCSLABS35	CCSLABS50	CCSLABS48	CCSLABS66
Purpose	LoadRunner Controller	Oracle Web Server	Oracle Forms Server	Oracle Database
Vendor	Compaq	Compaq	Compaq	Compaq
Server Model	Proliant DL360 G2	Proliant DL360 G2	Proliant DL360 G2	Proliant DL580
Processor Type	1.4 GHz	1.4 GHz	1.4 GHz	700 MHz
Number of Processors	2	2	2	4
RAM	2 GB	2 GB	2 GB	2 GB
Partition Size c:	16.9 GB	4 GB	4 GB	50 GB
Partition Size d:	-	12.9 GB	12.9 GB	-
NIC Vendor	Compaq NC7780 Gigabit Server Adapter	Compaq NC7780 Gigabit Server Adapter	Compaq NC7780 Gigabit Server Adapter	Compaq NC3134 Fast Ethernet NIC
NIC Speed	Auto Detect	Auto Detect	Auto Detect	Auto Detect

Software and Platform Specifications

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

Component	Software															
LoadRunner Controller	<ul style="list-style-type: none"> ▪ Windows 2000 Server with SP2 ▪ LoadRunner 7.51 ▪ Citrix ICA Client Integration with LoadRunner v 6.21.031 															
MetaFrame XP Server	<ul style="list-style-type: none"> ▪ Windows 2000 Server with SP2 ▪ MetaFrame XP Presentation Server, Enterprise Edition with Feature Release 2 ▪ Internet Explorer 6.0.2600 ▪ Jinitiator 1.1.8.19 ▪ Post-SP2 hotfixes: <table style="margin-left: 20px; border: none;"> <tr> <td>✓ Q147222</td> <td>✓ Q313450</td> <td>✓ Q319733</td> </tr> <tr> <td>✓ Q295688</td> <td>✓ Q313582</td> <td>✓ Q320176</td> </tr> <tr> <td>✓ Q299956</td> <td>✓ Q313829</td> <td>✓ Q320206</td> </tr> <tr> <td>✓ Q300845</td> <td>✓ Q314147</td> <td>✓ Q321599</td> </tr> <tr> <td>✓ Q311967</td> <td>✓ Q318138</td> <td>✓ SP2 SRP1</td> </tr> </table> 	✓ Q147222	✓ Q313450	✓ Q319733	✓ Q295688	✓ Q313582	✓ Q320176	✓ Q299956	✓ Q313829	✓ Q320206	✓ Q300845	✓ Q314147	✓ Q321599	✓ Q311967	✓ Q318138	✓ SP2 SRP1
✓ Q147222	✓ Q313450	✓ Q319733														
✓ Q295688	✓ Q313582	✓ Q320176														
✓ Q299956	✓ Q313829	✓ Q320206														
✓ Q300845	✓ Q314147	✓ Q321599														
✓ Q311967	✓ Q318138	✓ SP2 SRP1														
Oracle Servers	<ul style="list-style-type: none"> ▪ Windows 2000 Server SP2 ▪ Oracle RDBMS 8.1.7.3.10 (<i>database server only</i>) ▪ TAR 1.11.2a (<i>database server only</i>) ▪ GZIP 1.2.4 (<i>database server only</i>) ▪ Oracle 9iAS Apache Web Server (<i>web server only</i>) ▪ Microsoft Visual Studio 6.0 SP4 ▪ MKS Toolkit for Developers version 8.1 ▪ GNU Make 3.79.1 ▪ Sun Java Development Kit (JDK) 1.3.1_04 ▪ Adobe Acrobat Reader ▪ WinZip 															

MetaFrame XP Server Configuration

This section details the steps required to stage 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 farm with one MetaFrame XP server in a single zone was created. The Citrix XML service was configured to use port 80.

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	=	No Timeout
▪ Timeout Setting for Disconnection	=	No Timeout
▪ Timeout Setting for Idle	=	No Timeout
▪ Required Encryption	=	Basic
▪ AutoLogon	=	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
------------------------	---	--------

Client Settings

▪ Connect Client Drives at Logon	=	Inherit User Configuration
▪ Connect Client Printers at Logon	=	Inherit User Configuration
▪ Default to Main Client Printer	=	Inherit User Configuration
▪ 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

Internet Explorer 6.0 Configuration

Internet Explorer 6.0 is the web browser that is used to access the Oracle JInitiator application. Custom configurations need to be validated before the test can be executed. An administrator completed the following configurations before the scalability tests were executed.

- Added the Oracle 11i portal to the trusted sites list.
- Enabled the following items from the security settings for the trusted sites to include:
 - Download Signed ActiveX controls
 - Download unsigned ActiveX controls
 - Initialize and Script ActiveX controls not marked as safe
- **Note:** The remaining items retained the default values for the Low security settings.
- Navigated to the Oracle 11i Portal and installed JInitiator accepting default settings.
- Published JInitiator in the MetaFrame XP server.

Oracle 11i Scalability Test Cases

The scalability test cases were designed to answer the following challenges:

- How many steady state users running Oracle JInitiator can be supported by MetaFrame XP on a production Compaq DL360 G2 server at a maximum 70% CPU utilization?
- What components limit the scalability of a production Compaq DL360 G2 server when running Oracle JInitiator?

Note: For further information of the resources included with a Compaq DL360 G2 server, reference the [Hardware Specifications](#) section of this document.

The following sections discuss the test cases that were defined and executed in order to answer the above questions.

Case 1: Oracle 11i AR Customer & Invoice to GL Baseline Tests

During the initial phase of the scalability testing initiative a baseline test was completed to determine the baseline for a single user when executing the Oracle 11i AR Customer & Invoice to GL script on a MetaFrame XP server running Microsoft Internet Explorer 6 through a published application accessing Oracle 11i. An additional user was added to the test to validate the initial baseline figures. The following baselines were obtained:

- No users baseline, server running for 15 minutes
- 1 Oracle 11i AR Customer & Invoice to GL vUser running for one script iteration (approximately 8 minutes)
- 2 Oracle 11i AR Customer & Invoice to GL vUsers running for one script iteration (approximately 8 minutes each)

The above baselines were obtained in order to be able to better determine the correlation between the number of users in the system and the server resource utilization. Therefore, allowing the team to make projections and estimate server thresholds.

Case 2: Oracle 11i AR Customer & Invoice to GL Supported Users

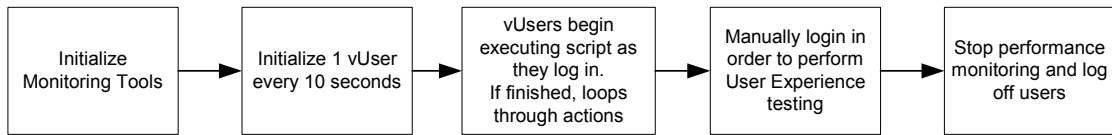
The object of this test case is to determine the number of users that can be supported on a MetaFrame XP server given a predetermined hardware configuration. The test case was designed to isolate system performance to iterations of just the AR Customer & Invoice to GL script and load the system to simulate a production server.

Steady State User Tests

This test case is designed to obtain steady state numbers. A server is considered at steady state when all the users have logged in and are working normally. To ensure steady state conditions, vUsers were setup to iterate through the script's actions, thus guaranteeing all vUsers would be executing different parts of the script once all of them are logged into the server. Scenarios were created in the following way:

- Scenario is set up to initialize all vUsers before running
- 1 vUser logs in every ten seconds
- vUsers are set-up to iterate through their actions when finished
- Steady state scenarios were created for 5, 10, 15, 20, 25, 30, 35, 38, 42 and 45 vUsers
- Scenario ends after 20 minutes for 5 thru 20 vUsers, 25 minutes for 25 and 30 vUsers and 30 minutes for 25 vUsers and above

The process flow is depicted in the following diagram:



Scalability Results and Analysis

The following sections analyze the server resource data and attempt to determine the bottlenecks that limited the number of users that a Compaq Proliant DL360 G2 two-processor server can support. All conclusions will be drawn and discussed in a later section titled [Scalability Test Conclusions](#).

Baseline Test Results

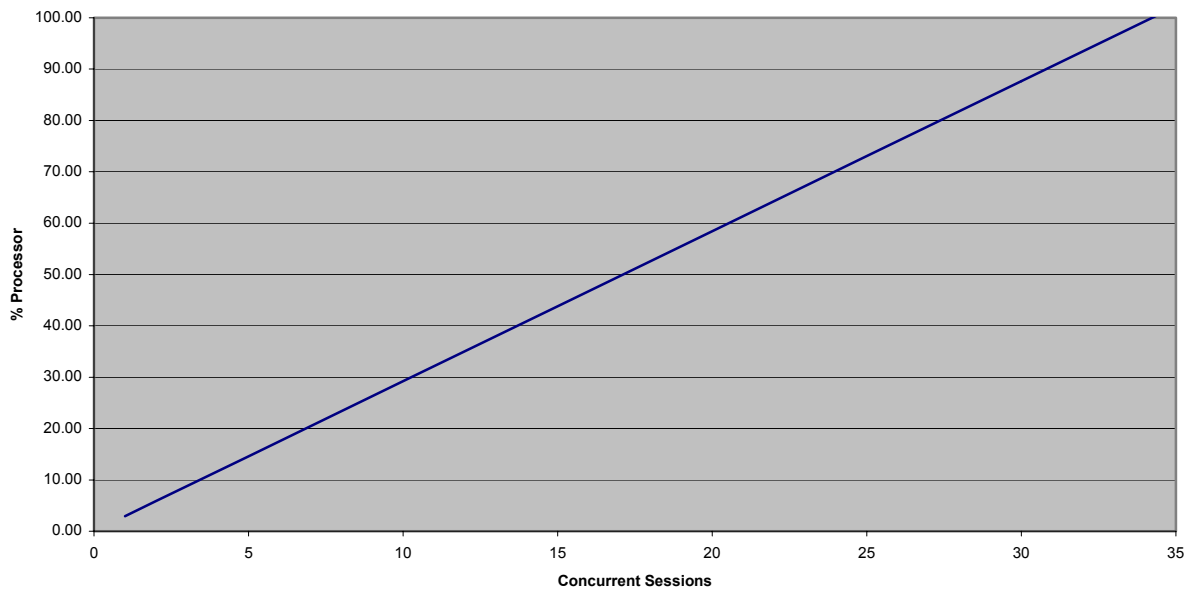
This section contains the results obtained when performing the baseline testing with the Oracle JInitiator application on a Compaq Proliant DL360 G2 two-processor server with 2GB of RAM. The following table shows averages for the counters recorded every 5 seconds in Performance Monitor.

Total Sessions	0	1	2
Memory Available MB	1,622.27	1,581.76	1,536.72
Memory Committed MB	272.41	320.36	367.43
Memory Free System Page Table Entries	152,624.00	152,628.00	152,228.00
Memory Page Reads/sec	0.44	0.00	0.01
Paging File (_Total)%Usage	0.37	0.44	0.47
Physical Disk % Disk Time	1.74	1.28	1.23
Physical Disk Current Disk Queue Length	0.03	0.02	0.38
Processor(_Total) % Processor Time	1.86	6.49	7.70
User Experience	Excellent	Excellent	Excellent

A hypothesis was established based on the data gathered in the baseline test. The hypothesis states that:

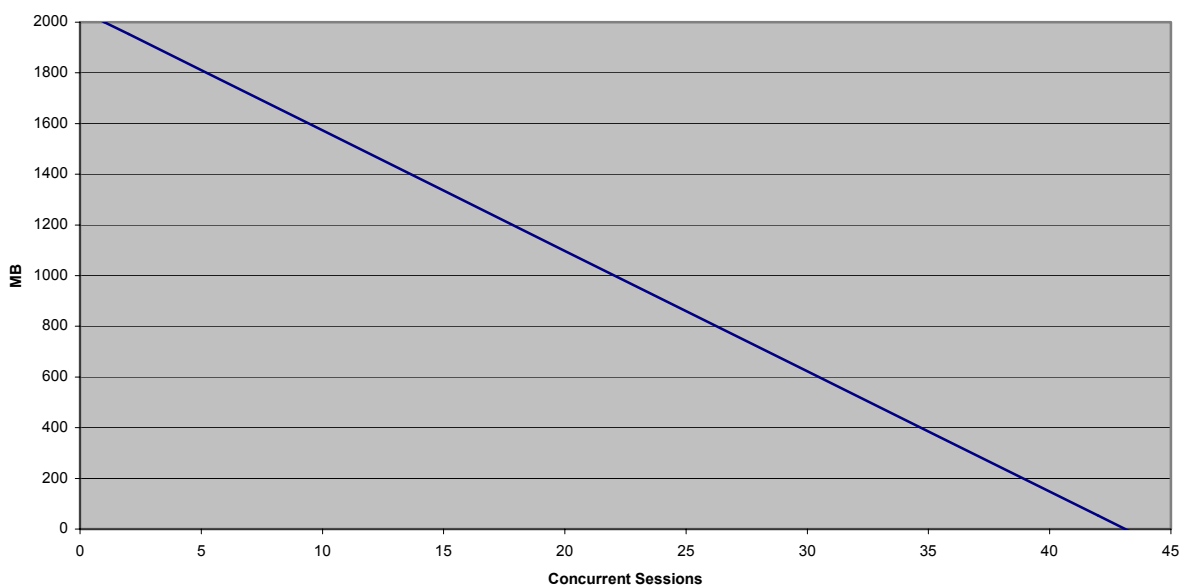
- CPU will be the primary system constraint. An average of 2.92% CPU will be consumed per session; therefore 20-25 concurrent sessions would make the server reach its CPU threshold. The following graph displays the projected CPU utilization based on the results from the baseline tests.

Projected CPU Utilization



- Memory will be the secondary system constraint. An average of 48MB will be consumed per session; therefore 30-35 concurrent sessions would make the server reach the recommended Available RAM threshold. The following graph displays the projected RAM utilization based on the results from the baseline tests.

Projected Available RAM



Oracle 11i AR Customer & Invoice to GL Performance Results

This section contains the results obtained when load testing with the Oracle JInitiator application on a Compaq Proliant DL360 G2 two-processor server with 2GB of RAM. The scalability testing was performed to determine the maximum number of users that the server could support while the users executed typical activities as defined by the automated script.

The following table contains the values for each counter that were averaged over steady state in each session interval. Each counter was measured every 5 seconds during the test's execution. The chart separates the data into different user session intervals.

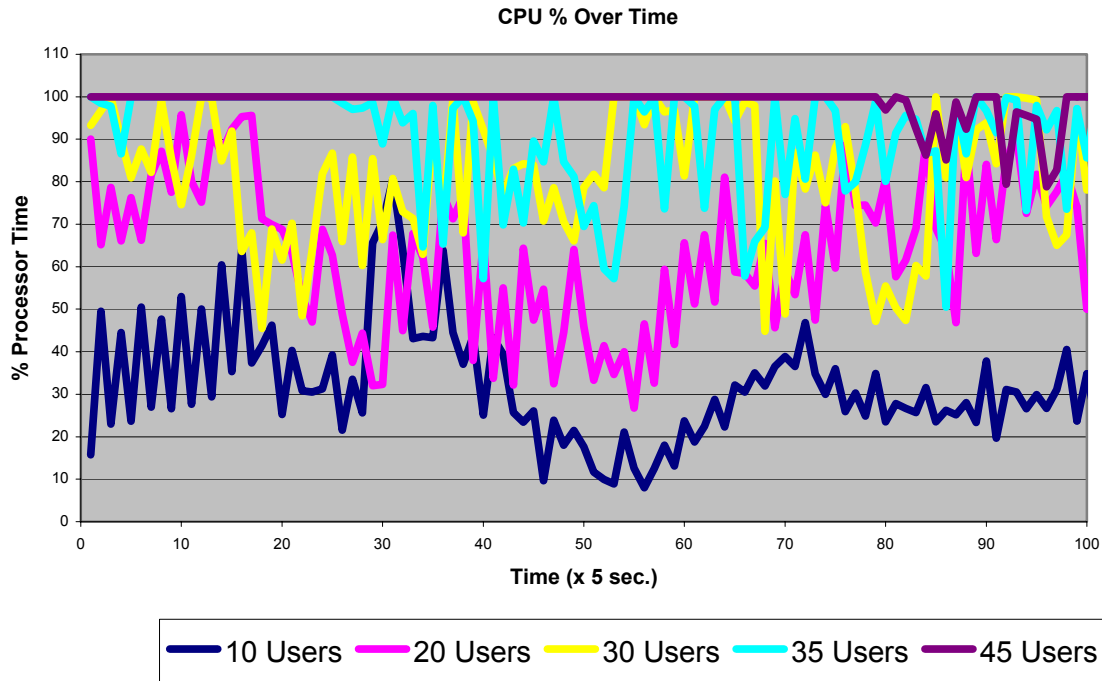
Total Sessions	5	10	15	20	25	30	35	38	42	45
Memory Available MB	1,411.52	1,206.96	1,024.89	841.60	654.67	516.63	477.73	337.41	244.54	187.94
Memory Committed MB	543.82	795.41	1,041.13	1,278.60	1,531.30	1,780.44	2,000.86	2,183.70	2,397.61	2,497.76
Memory Free System Page Table Entries	149,596.64	144,990.59	140,742.62	136,576.82	132,610.19	128,730.93	124,663.48	122,410.12	119,662.56	117,810.07

Total Sessions	5	10	15	20	25	30	35	38	42	45
Memory Page Reads/sec	0.43	0.02	0.02	0.03	0.03	0.03	0.03	0.04	0.95	2.35
Paging File (_Total)%Usage	1.90	2.15	2.42	2.64	2.90	5.60	12.04	11.90	16.20	16.43
Physical Disk % Disk Time	2.39	2.38	2.54	2.58	3.92	4.47	5.95	6.55	5.55	12.22
Physical Disk Current Disk Queue Length	0.29	0.01	0.08	0.01	0.00	0.04	1.85	1.02	1.00	1.72
Processor(_Total) % Processor Time	18.75	33.14	49.99	63.49	80.26	82.26	89.97	93.69	97.49	98.45
User Experience	Excellent	Excellent	Excellent	Excellent	Excellent	Excellent	Excellent	Acceptable	Acceptable	Acceptable

The user experience factor was “Excellent” and performance was consistent up to 35 concurrent sessions. At that point user experience started to degrade becoming “Acceptable” for 38, 42 and 45 concurrent sessions. The test was stopped at 45 users due to over 10% of the scripts failing. It is worth mentioning that some of the script failures can be attributed to un-synchronization of the script and the application. The script was recorded with reasonable ‘think times’ for a fast-paced medium user. For more information of user classification reference the section entitled [Creating vUser Scripts](#). The testing team assumed that although more ‘think time’ might solve some of the script synchronization problems it would be at the cost of user experience.

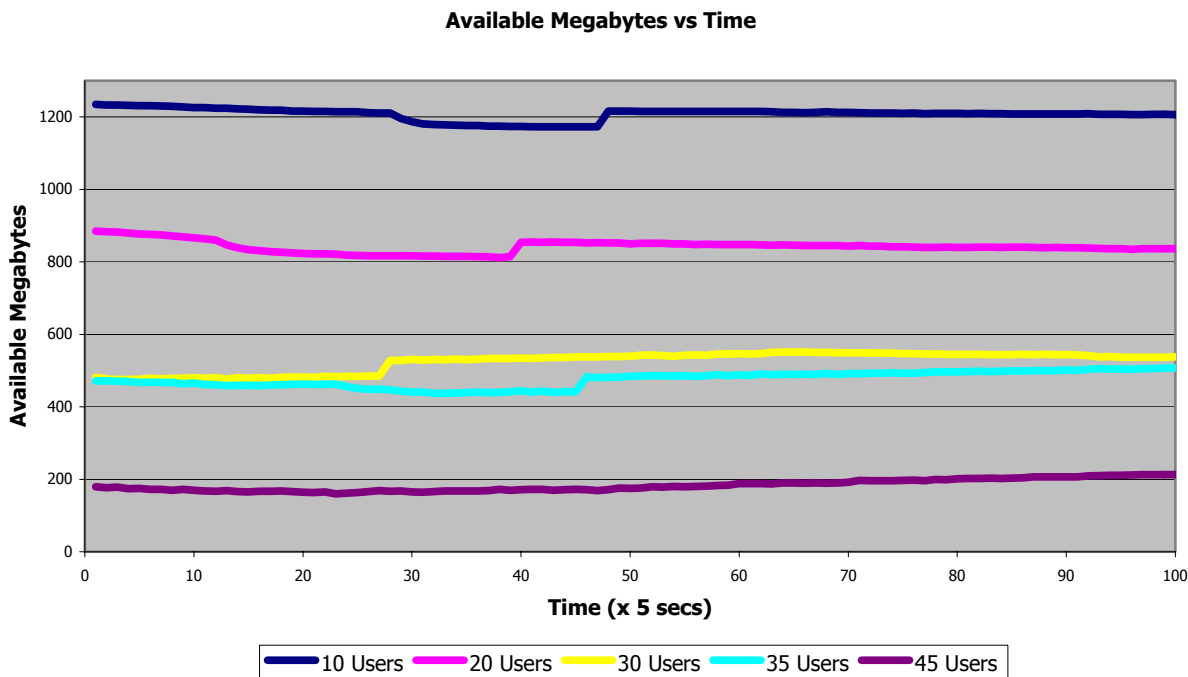
When analyzing the results, two limiting factors were discovered in system resource utilization: CPU and Memory. These correlate to the hypothesis established during the baseline analysis. Following are two graphs that will further detail the consumption of both factors in order to analyze results and identify correlations between resource utilization and transactions executed in the script.

The following graph displays the Total % Processor Time over the duration of the test under different user loads.



As it can be seen from the graph above, a direct relationship can be identified between number of concurrent sessions and CPU utilization. There are significant peaks during the first 50 seconds, which correlate to the first transactions: Adding New Customer and Enter Customer Classification Data. The graph also displays a consistent CPU utilization of over 70% when the system is loaded with over 30 concurrent sessions. The actual utilization figures based on the steady state testing indicated approximately of 2.4% CPU per user/session.

The next graph depicts memory utilization under different user loads.



As it is expected, as more users are loaded into the system, the less memory is available. The actual utilization figures based on the testing indicated approximately 54MB of RAM per user/session.

As can be seen in the graph, memory became a bottleneck when the system was loaded with more than 30 concurrent users since the average available bytes was less than 600MB which is 30% of the total RAM (2GB).

When analyzing the results, the team ignored the performance data during the initialization phase of all the scenarios. It was determined that in a production environment these actions would probably be performed once per day for each user; therefore, aggregating that load may skew the results and would not provide an accurate representation of standard Oracle JInitiator application load. It is important to note that there are three events that occur during the initialization phase: login to the MetaFrame XP server, login to the Oracle application and entry to the Receivables module, which downloads the Java applet. Although the testing indicates that the average resources required for the initialization phase are not significantly higher than during the actions phase, several considerable spikes were noticed. These spikes tend to occur predominantly in the following counters: Processor Time, Page Reads per Second and in Percent Disk Time. It is important to account for those spikes during the MetaFrame XP farm planning phase to ensure appropriate responsiveness for all users at all times.

Scalability Test Conclusions

The purpose of this scalability test process was to determine overall resource consumption on a MetaFrame XP server when running the Oracle 11i application using Microsoft Internet Explorer version 6.x. These resources would be the primary factors in determining the total number of concurrent users that could be supported in a production environment.

The following table summarizes the findings and conclusions based on objective (Performance Monitor counters) and subjective (User Experience) measures.

Hardware	System Bottleneck	Maximum Observed Concurrent Sessions	Maximum Concurrent Sessions Without Degradation	Recommended Concurrent Sessions
Two-Processor 1.4 GHZ with 2 GB of RAM	% Processor Time % Available Bytes	45	35	25

The following bullets summarize the scalability test findings for usability.

- **Maximum observed concurrent sessions.** 45 concurrent sessions although performance degradation was noticed during user experience testing.
- **Maximum sessions without degradation.** 35 concurrent sessions was the maximum observed without performance degradation.
- **Recommended concurrent sessions.** Typically, Citrix consulting recommends sizing servers for a production environment at 60 to 70% of the maximum values. This helps to ensure appropriate responsiveness for all users at all times. This will also help to minimize logon effects or power usage effects of a few users on many other users. Therefore, the recommended user load is 25 sessions per server, which will also tie in with the CPU and Available Bytes, observed thresholds.

User adaptability to longer wait times could influence the User Experience thresholds identified above. However, it is imperative to monitor objective thresholds to ensure good server performance.

The following bullets summarize the observed objective thresholds that limited the number of concurrent sessions on the Compaq DL360 G2 two-processor 1.4 GHZ server with 2 GB of RAM.

- **CPU.** One of the primary system resource constraints during testing was overall CPU utilization. The threshold was reached between 20-25 concurrent sessions.
- **Memory.** The second bottleneck was Memory. At 2GB RAM memory became a concern upon reaching 25-30 users.

Note: The scripted scalability tests used in this analysis are helpful in obtaining rough estimates when sizing the server. The data and analysis in this document should serve as a guide, and not definitive numbers, for further MetaFrame XP scalability testing for Oracle 11i. 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 Oracle 11i application 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.

Appendix A – LoadRunner Script Generation

LoadRunner's VuGen enables easy creation of automated testing scripts by recording all activity during manual interactions with a MetaFrame XP 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.

Citrix vUser scripts emulate the Citrix ICA protocol communication between a Citrix ICA client and MetaFrame XP server. 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:

- Extract Citrix_Headless_Client.zip to the LoadRunner Controller.
- Run setup.exe to install the Citrix ICA Client Integration with LoadRunner
- Run “%Program Files%\Citrix\ICA Client\wfica32.exe /setup” to register the necessary objects in the Citrix ICA Client Integration
- Download any patches that may be needed from <http://support.mercuryinteractive.com>

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

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.

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 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 paging file 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.

Bytes Total per Second

The number of bytes the server has sent to and received from the network. This value provides an overall indication of how busy the server is. Bytes Total/Sec is a good measure to determine if the Network Interface is the bottleneck. If this number approaches its maximum (100Mbs), then the network is likely to be the cause of user latency in the system.

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.

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



851 W. 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.