

Performance Test Summary Report

<Department Name>

<Project Name>

<Version Number>

Document history

Date	Version	Author	Reviewed by	Approved by	Description

Table of contents

1	Introduction	4
1.1	Engagement	4
1.2	Objective of the report	4
1.3	Intended audience	4
2	Scope of Performance Testing	5
3	Test Approach and Methodology	6
4	Test Environment and Configuration	9
4.1	Test Environment	9
4.2	Load Generator System Configuration	9
4.3	Performance Monitoring	9
4.4	Performance Test Tool Details	11
4.5	Test Configuration for Baseline Test	11
5	Test Execution Summary	12
5.1	Execution Summary	12
5.2	Test Result Analysis	12
5.3	Test Analysis - Graphical	14

1 Introduction

1.1 Engagement

<Details of the Engagement>

1.2 Objective of the report

<Define for what purpose this document has been built for>

1.3 Intended audience

2 Scope of Performance Testing

<The following items are in the scope of work of the performance test:>

3 Test Approach and Methodology

The following sections describe in detail the devised strategy to execute the performance test activity of <portal Name>.

Approach

The following types of performance tests were planned in order to exercise the system fully to ensure that the required performance SLAs are met. These tests are also designed with the aim of exposing the maximum number of performance bottlenecks.

Based on the identified test scenarios following kinds of tests were created.

1. Best Case Scenario

This is to test with a single user over a period of time to find out the best possible time for the application to be performance tested. Application would be loaded with a single user activity and corresponding performance metrics would be gathered against it. This is primarily to ensure that the scripts will execute end to end without any issues.

2. Baseline Test

Each application workflow identified in the performance critical list will be executed with a peak load in the full configuration of the system in isolation with no other concurrent load with the aim of getting the baseline performance metrics.

During this test execution, application performance would be monitored to gather the critical metrics like CPU Utilization, Memory Utilization, Network/ Bandwidth usage etc.

3. Spike Test

This Test will be executed on the environment to ensure that the <Application Name> IT System is capable enough of handling sudden load in exceptional conditions. This would demonstrate system's ability to deal with large number of requests in short span of time, to which the current system architecture can sustain itself. This would be achieved by applying a step user load on a stable running condition.

During this test execution, application performance would be monitored to gather the critical metrics like CPU Utilization, Memory Utilization, Network/ Bandwidth usage etc.

4. Stress Test

If the Baseline test achieves satisfactory performance, a separate Test will be executed on the environment to ensure that the <Application Name> IT System is capable enough of handling excess load in exceptional conditions. This would establish the load limit, to which the current system architecture can sustain itself. This was achieved by increasing the User Load by around 100%, 200% and 400% of its expected usage and Transactional Load.

Methodology

The methodology of the performance analysis activates are mentioned below:



1. Requirement Gathering :-

The performance test activities will be limited to the business transactions as defined in the subsequent sections in this document. The test strategy assumes that there is no batch activity therein during normal business operation.

Performance testing scenarios that are modelled on the real life transactions and user load would be carried out on the system to investigate the system response, resource utilization and other system parameters.

During performance testing, PwC performance testing team will simulate only those transactions from the source system that cause server interaction or place load on <Application Name> Application, Web and DB servers. Any wait period for transactions posting within the application would be handled through appropriate wait time in the scripts. Any other external systems/portals which have a direct/indirect reference/interaction with <Application Name> Application, will not be considered for Performance test.

Identification of scenarios is based on questionnaire response by <Team Name>.

2. Script preparation :-

As part of the performance testing all scripts were developed in dry run mode for the identified features and executed by applying concurrent user load. The features to be tested have been logically grouped to be covered through five Scripts that were developed followed by load requirement analysis for the modules. Following table captures the featured covered by each scripts.

Scripts	Features covered
Bid Doc Create	Procurement Document Creation
Bid Submission	Bid Submission

3. Performance Test Execution :-

Response times of each transaction along with all other necessary parameters pertaining to performance testing were measured. Execution was basically performed in following ways:

- Best Case Testing.
- Baseline Testing.
- Spike Testing.
- Stress Testing.

4. Server monitoring during Performance test Execution: -

Server monitoring was done through adding a plug-in at Apache JMeter tool and executing a batch script in servers. Windows PerfMon tool was also used to capture those parameters. We got the various server statistics or parameters like CPU Usage, Memory Usage, Disks Input/output, Networks Input/output, Server Locks from Server Performance monitoring activity.

5. Report Preparation :-

After executing the Performance tests we have created an executive summary report which included performance test methodology, performance test approach, business workflows that were used for performance testing, all the conducted Performance testing results, Client and server side details and recommendations.

4 Test Environment and Configuration

Test Environment will be used for performance test execution. As there is a difference between configuration of Test Environment and actual production environment, there is a scope of extrapolation of test result. Hence the identification of bottleneck and recommendations will be made at high level.

4.1 Test Environment

The test environment is the designated performance testing environment for <Application Name> Portal performance testing. The following table lists the hardware and software configuration of the <Application Name> Portal Stage and Production environment.

Application Component	Host Name	Operating System	CPU Type	# of CPU	RAM	Software
Proxy Server	PROX YSRV.lo caldom ain	RHEL Version 5.5	Intel(R) Xeon(R) CPU E5620 @ 2.40GHz, 4 Core(s), 4 Logical Processor(s)	8	32GB	Apache Tomcat
Application, DB and CMS Server	- APPSRV.I ocald omain	RHEL Version 5.5	Intel(R) Xeon(R) CPU E5620 @ 2.40GHz, 4 Core(s), 4 Logical Processor(s)	8	64GB	JBoss, Alfresco, MySQL

Test environment was connected with Load generator through Checkpoint VPN network.

4.2 Load Generator System Configuration

Test was conducted from a machine of following configuration –

Processor – Intel(R) Core(TM)2 i5-3340M CPU @ 2.70GHz

RAM details – 8 GB

Hard Disk Details – 256 GB SSD

Operating System – Windows 7 professional 64 bit

Test Database Details – MySQL and SQLYog

System Type – 64 bit

4.3 Performance Monitoring

During each test execution, pre-defined performance metrics will be monitored and analyzed to ensure compliance to performance SLAs and to detect bottlenecks in the various application components. The performance metrics that would be monitored fall under one of the following three broad categories:

- Response and Elapsed Time
- Throughput / Hits
- Resource Utilization (Infrastructure)
- Response and Elapsed Time

End user response time for <Application Name> Portal will be monitored and measured for all the

end user transactions using performance testing tool.

- **Throughput/Hits**

Throughput is defined as the number of departures past a given point per unit time. Throughput metrics pertaining to the rate of requests served from the <Application Name> Portal server would be monitored over the test period.

- **Resource Utilization (Infrastructure)**

The following resource utilization metrics will be monitored and measured during all the performance tests on all the servers that are part of the <Application Name> IT System Infrastructure to ensure that there are no capacity bottlenecks.

- Processor Utilization
- Memory Utilization
- Disk I/O statistics (Rates, Wait times) would be monitored for server. This eventually contributes to a large extent for bad application performance

Tests were carried out over the Nepal VPN assuming that the Network bandwidth is good enough to support the defined Load.

4.4 Performance Test Tool Details

Apache JMeter 2.9 was used to conduct performance testing. The **Apache JMeter** desktop application is open source software, a 100% pure Java application designed to load test functional behavior and measure performance. It was originally designed for testing Web Applications but has since expanded to other test functions. Apache JMeter can be used to test performance both on static and dynamic resources (files, Servlets, Perl scripts, Java Objects, Data Bases and Queries, FTP Servers and more). It can be used to simulate a heavy load on a server, network or object to test its strength or to analyze overall performance under different load types. It can be used to make a graphical analysis of performance or to test server/script/object behavior under heavy concurrent load.

4.5 Test Configuration for Baseline Test

Scenario configuration details for baseline load testing consisting of all the test scripts. Script parameter details are noted in the table below:

Sl No	Test Script	Total Vusers during Test	Transaction rate(per Hour per Vuser)	Think Time (in seconds)	Total Ramp up Time (in mins)
1	Procurement Document Creation	50	30	30	20
2	Bid Submission	75	50	30	20

Detailed Load calculation for each module and each test has been mentioned in Annexure 1.

5 Test Execution Summary

5.1 Execution Summary

The test is conducted to observe the system behavior for a load of 100 Virtual Users.

Test Duration:	1 hours
Maximum Running Vusers:	100
Average Response Time (second):	113.732
Average Hits per Second:	28
Average Throughput (KB/second):	50.140
Total Transactions Tested:	2111

5.2 Test Result Analysis

- 45% of the total transactions showed an average transaction response time above 120 seconds. The transactions are

Transaction Name
_ BidSubmission _10_PriceSchedule
_ BidSubmission _04_FinancialSituationForm
_ BidSubmission _03_SubmissionForm
_ BidSubmission _14_GenerateReport
_ BidCreate _07_BidDataSheet
_ BidCreate _09_SOR
_ BidCreate _10_SCC
_ BidCreate _12_AddBOC
_ BidCreate _13_GenerateReport
_01_LaunchPortal
_02_Login
_ BidCreate _05_UploadAbbr
_ BidCreate _06_InvitationForBids

- Average transaction response times of 21% of the total transactions were between 60-120 seconds. The key transactions are

Transaction Name
_ BidSubmission _07_PendingLitigation
_ BidSubmission _08_SpecificExperienceForm
_ BidSubmission _14_TechSpec
_ BidSubmission _11_Declaration
_ BidSubmission _13_Documents
_ BidSubmission _15_SubmitBid.

Refer to Annexure 1 for details

- 10% of the total transactions showed average transaction response time in between 30-60 seconds. The key transactions are

Transaction Name
_ BidCreate _08_E&Q
_ BidCreate _11_AdditionalDocs
_ BidCreate _03_NavigateToPage

- Rest 24% of the total transactions had an average response time in between 3-30 seconds. . The key transactions are

Transaction Name
_ BidCreate _03_BidSchedule
_ BidCreate _04_CoverPage
_ BidCreate _14_CreateBid
_ BidSubmission _05_AnnualTurnoverForm
_ BidSubmission _06_FinancialResourceForm
_ BidSubmission _09_ManufacturerAuthLetter
_ BidSubmission _12_Payment

The detailed response time for 100 User's Test is attached in Annexure 1.

5.2.1 Performance Test Server Analysis

Test was conducted over a period of 60 minutes with a load of 100 concurrent virtual users. The CPU Utilization snapshot of the Performance Test Servers was also collected during the test. On analyzing the CPU Utilization snapshots, following can be interpreted:

The server was showing an average CPU utilization of 17.313% under load.

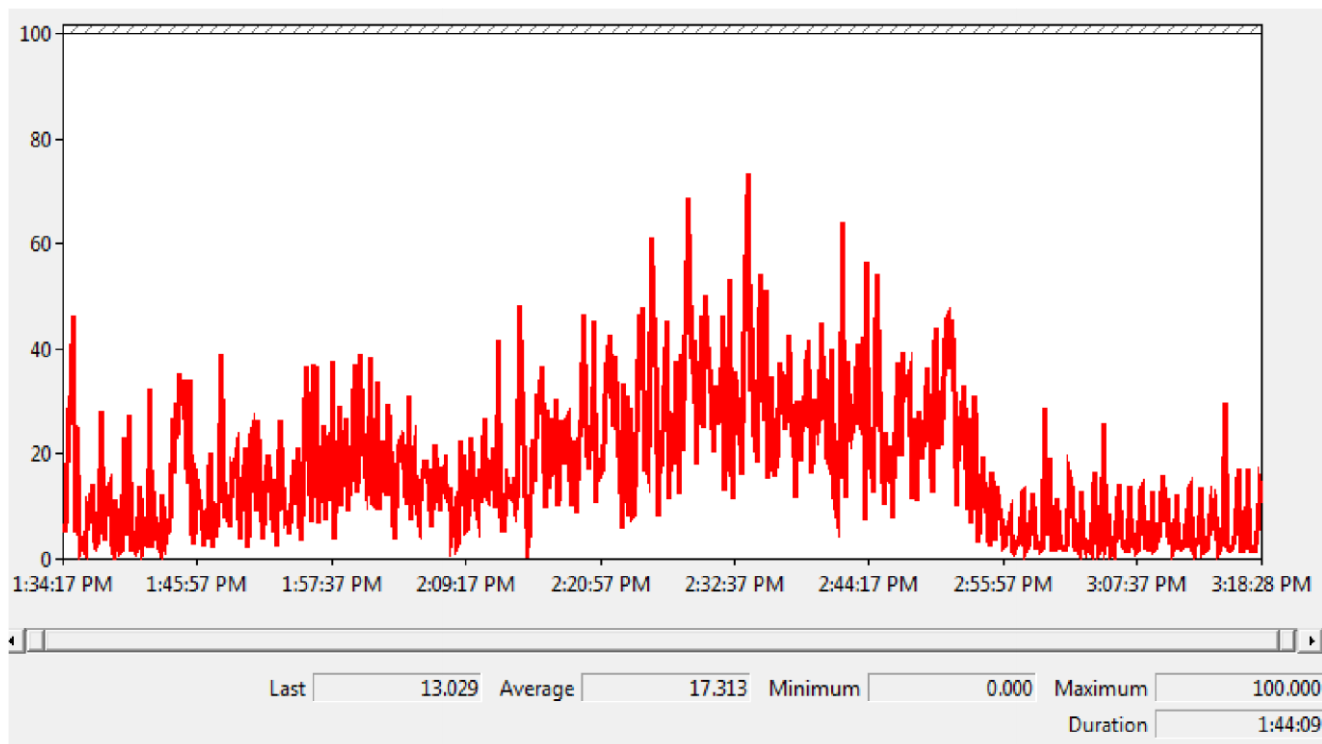
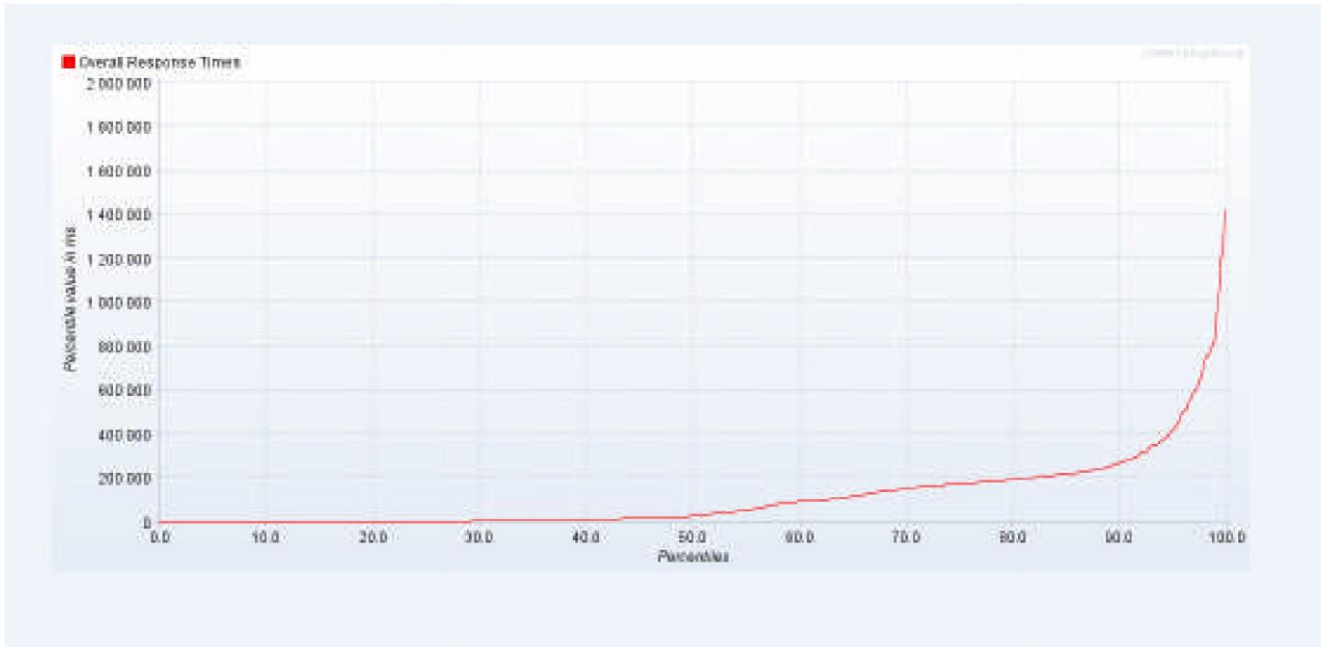
Maximum CPU utilization reached around 50% line frequently during the test.

The CPU Utilization and other system resources snapshot for 100 User's Test is attached in the Annexure 1.

5.3 Test Analysis - Graphical

Graphical analysis of the Server Performance was analyzed during the Test execution. The details are listed below:

5.3.1 Response Analysis and CPU Utilization of Servers



5.4 Glossary of terms

Term	Definition
Think Time	Wait time between two transactions/steps
Parameterization	Process to send various parameters from client side
Correlations	Process to capture dynamic value generated by server side
Iterations	Indicates the number of execution a flow
Transaction	Specific Steps/operations/process of the Applications that goes back to the server
Test Script	A collection of relevant transactions/steps recorded and enhanced in Load testing tool
Vuser	Virtual User i.e. simulated user created through Load testing tool
Throughput	The amount of data received from server for requests processing in specified duration
Hits per Second	The number of requests sent to server per second
Ramp up Time	The time taken to create particular user load
Transaction Rate	Number of transactions occurring for per user in a hour

Annexure

Annexure 1 - Performance Test Report (Mandatory)

Performance Test report based on any standard tool should be attached in this section.