# Performance Testing of Uhuroo.com using OpenSTA

#### <u>Team Members</u> Prathab K - 06329902 Radha A - 06329024 Sabil Ali - 06005101

## Introduction

The primary problems that web application after deployment are not system crashes or incorrect system responses, but rather system performance degradation or problems handling required system throughput. Performance testing addresses these issues well before the application is deployed. Performance testing is performed from one perspective, to determine how fast some aspect of a system performs under a particular workload. The application is evaluated from a user's perspective, and is typically assessed in terms of throughput, stimulus response time, or some combination of the two. Alternatively, performance testing is done for the uhroo.com web application. Uhuroo is a web-based software provides an integrated environment for the information collaboration, organization, sharing and management of the content. Uhuroo software provides user to collect, store, organise, share, print the information available online (Please refer the link <u>www.uhuroo.com</u> for details and functionalities provided). For performance testing, the OpenSTA (Open System Testing Architecture), a distributed software architecture for developing, executing and analyzing the results of tests, tool is used.

This document will provide a modelling of performance testing for the uhuroo.com using OpenSTA tool and discuss the performance testing results. The document is organized as follows. In next section scope of the testing is discussed and followed by modelling of performance testing i.e. characterizing the workloads, creating operational profiles. Finally the results obtained using OpenSTA are analyzed.

#### **Performance Testing – Scope**

The scope of this performance testing of uhuroo.com application includes:

- measuring the stimulus response time with respect to user under various workloads operated with different operation profiles
- measuring the error rate as the workload increases

There is no evaluation of the server is done (since access to the server is restricted).

## OpenSTA

OpenSTA (Open System Testing Architecture) is a distributed software architecture for developing, executing and analyzing the results of Tests.

A Test may include Scripts or Collectors or both. Scripts define the operations performed by Virtual Users. Collectors define sets of SNMP, NT Performance data or other data to be retrieved during all or part of a Test-run. They can provide useful information about system activity and the Results can be analyzed alongside those from other OpenSTA Modules.

The OpenSTA Architecture provides generic facilities that may be used by other OpenSTA Modules:

- OpenSTA is designed to create and run HTTP/S load Tests to help assess the performance of Web Application Environments (WAE).
- Tests can be developed to simulate realistic Web scenarios by creating and adding Scripts to a Test to reproduce the activity of hundreds to thousands of users.
- Resource utilization information and response times from WAEs under test can be monitored and collected during a Test-run and then displayed.

Please refer the following link <u>http://www.opensta.org/docs</u> for more details regarding the OpenSTA tool.

### Modelling the performance testing for uhuroo.com

This section describes how the performance testing for uhuroo.com is modelled and tested. Usually the performance testing is frequently not performed against a specification, i.e. no one will have expressed what the maximum acceptable response time for a given population of users should be. Performance testing is frequently used as part of the process of performance profile tuning. The idea is to identify the threshold limit of the application and discovering the areas where the performance improvement is required. The performance testing is conducted by creating workload, a representative of how the system will be used in the field, and then run the system on those benchmarks. The system's behavior on the benchmarks in considered indicative of how the system will behave in the field. In this case study, the workload is characterized with different user operations with respect to the applications. Using OpenSTA tool, the workload is simulated with varying number of virtual users.

#### **Operational profiles for uhuroo.com**

An operational profile is a probability distribution describing the frequency with which selected important operations are exercised to test the applications.

Following operations are considered as widely performed with respect to uhuroo (There is no survey conducted in identifying the widely used operations, this is based on the idea of the normal user interaction with application in the perspective of user needs.):

- Reading scribbles
- Posting/writing scribbles
- Editing scribbles
- Searching scribbles
- Meta-operations (like applying tags to scribbles)
- Organizing the scribbles (printing scribbles)
- Navigation on the application (like exploring other groups, public scribbles, etc.)

Below are the operational profiles used is conduct the performance testing of uhuroo.

Profile Name : Profile-1					
Profile Description	Workload Type	Workload percentage			
	Reading scribbles	40%			
Involves reading scribbles and application navigation	Application navigation (accessing help, exploring groups and public scribbles, etc.)	60%			
Profile Name : Profile-2					
Profile Description	Workload Type	Workload percentage			
Involves scribbles	Reading scribbles	30%			
and write scribbles	Posting scribbles	30%			
	Editing scribbles	40%			

Profile Name : Profile -3					
Profile Description	Workload Type	Workload percentage			
Involves more search	Reading scribbles	30%			
and navigational operations	Searching scribbles	50%			
	Navigational operations	20%			
Profile Name : Profile -4					
Profile Description	Workload Type	Workload percentage			
	Editing user settings	20%			
	Meta operations on scribbles	40%			
Involves miscellaneous operations	Organizing scribbles (print basket)	20%			
	Navigational operations	10%			
	Logout and re-login operations	10%			
Profile Name : Profile -5					
Profile Description	Workload Type	Workload percentage			
	Reading scribbles	20%			
T 1 4 C41 11	Write/Editing scribbles	20%			
used operations	Searching scribbles	20%			
	Meta operations on scribbles	20%			
	Navigational operations	20%			

The workload type represents the type of operations performed and the workload percentage represents the amount of particular operation is performed (in terms of number of users).

#### **Environmental details**

Client Environment:

- P4 with Single Processor
- Processor speed 3.06 GHZ
- Memory 512 MB

#### Server Environment:

- P4 with Single Processor
- 140 160 GB hard disk (7200 RPM) capacity
- Hyper threading is enabled
- Processor speed 3.06 GHZ
- System Bus Speed 533 MHZ
- System Memory Speed 533 MHZ
- L2 Cache RAM 1MB
- Memory 1GB

- SDRAM frequency 533 MHz
- Single Memory Mode

*Network Bandwidth* : 1Mbps (The testing is carried out in the local server)

## Performance testing results for uhuroo.com and its analysis

The performance testing for uhuroo.com is done using OpenSTA testing tool. OpenSTA is designed to create and run HTTP/S load Tests to help assess the performance of Web Application Environments (WAE). Open STA provides a variety of data collection about HTTP/S Load and display options to assist in the analysis of Test results. In this study, the following HTTP graphs are taken for the analysis:

- HTTP Active Users v Elapsed Time Graph
- HTTP Response Time v Elapsed Time Graph
- HTTP Errors v Elapsed Time Graph

The HTTP data list also analyzed for the type responses and response of specific URLs. Below are the graphs and analysis of the selected operation profiles. **Profile-1** 

Operations	Read Private Scribble	Accessing Help	Read Public Scribble	Navigation
Total virtual users	120	60	60	60
Interval between batches	1s	1s	1s	1s
Number of virtual users per batch	1	1	1	1
Batch Ramp up time	1s	1s	1s	1s
Iteration(s)	1	2	1	1







3000







Fig: HTTP information graphs with respect to time

	response time at (at second)	with users	with response time of (in seconds)
Server's first HTTP peek response	199	300	110
Server's first reach of 10s response duration	97	210	10

Within server's first HTTP peek response	
HTTP Success response (200)	25%
HTTP not modified response (304)	71%
HTTP error response (400,500)	4%
Average read scribble response (approx)	2s

## **Profile-2**

Operations	Read	Create	Edit
Operations	Scribbles	Scribble	Scribble
Total virtual users	90	90	120
Interval between batches	1s	1s	1s
Number of virtual users per batch	1	1	1
Batch Ramp up time	1s	1s	1s
Iteration(s)	1	1	1







HTTP Active Users vs Elapsed Time



HTTP Errors vs Elapsed Time

Fig: HTTP information graphs with respect to time

	response time at	with users	with response time of
	(at second)		(in seconds)
Server's first HTTP peek response	88	250	55
Server's first reach of 10s response duration	81	230	10

Within server's first HTTP peek response	
HTTP Success response (200)	21%
HTTP not modified response (304)	75%
HTTP error response (400,500)	4%
Average create scribble response (approx)	2s
Average edit scribble response (approx)	2s
Average read scribble response (approx)	2s

# **Profile-3**

Operations	Read Scribbles	Search Scribbles	Navigation
Total virtual users	90	150	60
Interval between batches	1s	1s	1s
Number of virtual users per batch	1	1	1
Batch Ramp up time	1s	1s	1s
Iteration(s)	1	1	1





HTTP Response Time vs Elapsed

Time



HTTP Active Users vs Elapsed Time

HTTP Errors vs Elapsed Time

Fig: HTTP information graphs with respect to time

	response time at	with users	with response time of
	(at second)		(in seconds)
Server's first HTTP peek response	134	270	120
Server's first reach of 10s response duration	84	210	10

Within server's first HTTP peek response	
HTTP Success response (200)	23%
HTTP not modified response (304)	73%
HTTP error response (400,500)	4%
Average search query response (approx)	28
Average read scribble response (approx)	4s

## **Profile-4**

Onemations	Edit	Meta	Organize	Navigation	Re-Login
Operations	Settings	Operations	Scribbles		
Total virtual users	40	80	40	20	20
Interval between batches	1s	1s	1s	1s	1s
Number of virtual users per batch	1	1	1	1	1
Batch Ramp up time	1s	1s	1s	1s	1s
Iteration(s)	1	1	1	1	1





Time



HTTP Active Users vs Elapsed Time

HTTP Errors vs Elapsed Time

Fig: HTTP information graphs with respect to time

	response time at	with users	with response time of
	(at second)		(in seconds)
Server's first HTTP peek response	154	200	12
Server's first reach of 10s response duration	150	200	10

Within server's first HTTP peek response	
HTTP Success response (200)	21%
HTTP not modified response (304)	74%
HTTP error response (400,500)	5%
Average login response (approx)	2s

#### **Profile-5**

Operations	Read	Edit	Search	Meta	Navigation
	Scribble	Scribble	Scribbles	Operations	
Total virtual users	40	40	40	40	40
Interval between batches	1s	1s	1s	1s	1s
Number of virtual users per batch	1	1	1	1	1
Batch Ramp up time	1s	1s	1s	1s	1s
Iteration(s)	1	1	1	1	1







HTTP Active Users vs Elapsed Time HTTP Response Time vs Elapsed Time HTTP Errors vs Elapsed Time

Fig: HTTP information graphs with respect to time

	response time at	with users	with response time of
	(at second)		(in seconds)
Server's first HTTP peek response	72	200	10
Server's first reach of 10s response duration	72	200	10

Within server's first HTTP peek response	
HTTP Success response (200)	23%
HTTP not modified response (304)	72%
HTTP error response (400,500)	5%

## Conclusion

This performance testing study is conducted to analyze the response time and error rates. By carrying out the testing under different operational profiles, we can conclude that the system (server) can provide an acceptable response time with load of 200 users. Under varying load (within 200 users), the system provides a good response time. The stability of the system goes off, when there are more than 300 users and lot of timeout operations occurs. Since it is tested in local network environment, the response time is good, but there will be delays in the remote access scenario. System provides good amount of caching with respect to retrieval of images and scripts (java scripts), but caching with respect to the data and results (like search results, frequently accessed scribbles) are quite low. The error rate is also quite low under limited load (200 users). The caching can be improved (with respect to user) to handle more requests.