Performance Testing - Uhuroo.com

Team Members
Prathab K - 06329902
Radha A - 06329024
Sabil Ali - 06005101

Performance Testing

- Testing that is performed, from one perspective, to determine how fast some aspect of a system performs under a particular workload
  - expected load in terms of concurrent users or HTTP connections
  - acceptable response time

Performance Testing – Uhuroo.com

- Scope
  - 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
  - No evaluation of server is done

OpenSTA (Open System Testing Architecture)

- OpenSTA (Open System Testing Architecture) is a distributed software architecture for developing, executing and analyzing the results of Tests
- 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
- Resource utilization information and response times from WAEs under test can be monitored and collected

Performance Testing – Steps Involved in OpenSTA

- Create Scripts
  - Create scripts by recording the actions and modify according the requirement
- *Example of script request & response*

```
PRIMARY POST URI "http://10.147.10.10/uweb/readScribble.do HTTP/1.0" ON 6 &
    HEADER DEFAULT_HEADERS &
    ,WITH {"Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, " &
            "application/x-shockwave-flash, application/vnd.ms-excel, application/vnd.ms-powerpoint, " &
            "application/msword, */*", &
        "Referer: http://10.147.10.10/uweb/login.do", &
        "Accept-Language: en-us", &
        "Content-Type: application/x-www-form-urlencoded", &
        "Connection: Keep-Alive", &
        "Content-Length: 197", &
        "Pragma: no-cache", &
        "Cookie: "+cookie_2_0}   &
    ,BODY "user=4&view2=&groupid=&userid=&field=&hgrpid=&htagid=&hgrp=&htag=&hmode=&model=&page=&view=23&" &
        "param=&doc=&scribbleid=33&mode=e&merge=&seltag=&selgroup=&seltopic=&seladdress=&selfieldsl=&sel" &
        "fields2="

WAIT 110

    GET URI "http://10.147.10.10/uweb/auxiliary/includes/style.inc.css HTTP/1.0" ON 7 &
        HEADER DEFAULT_HEADERS   &
        ,WITH {"Accept: */*", &
            "Referer: http://10.147.10.10/uweb/readScribble.do", &
            "Accept-Language: en-us", &
            "Cookie: "+cookie_2_0, &
            "If-None-Match: W/~<22>12587-1171962083000~<22>", &
            "Connection: Keep-Alive"}
```

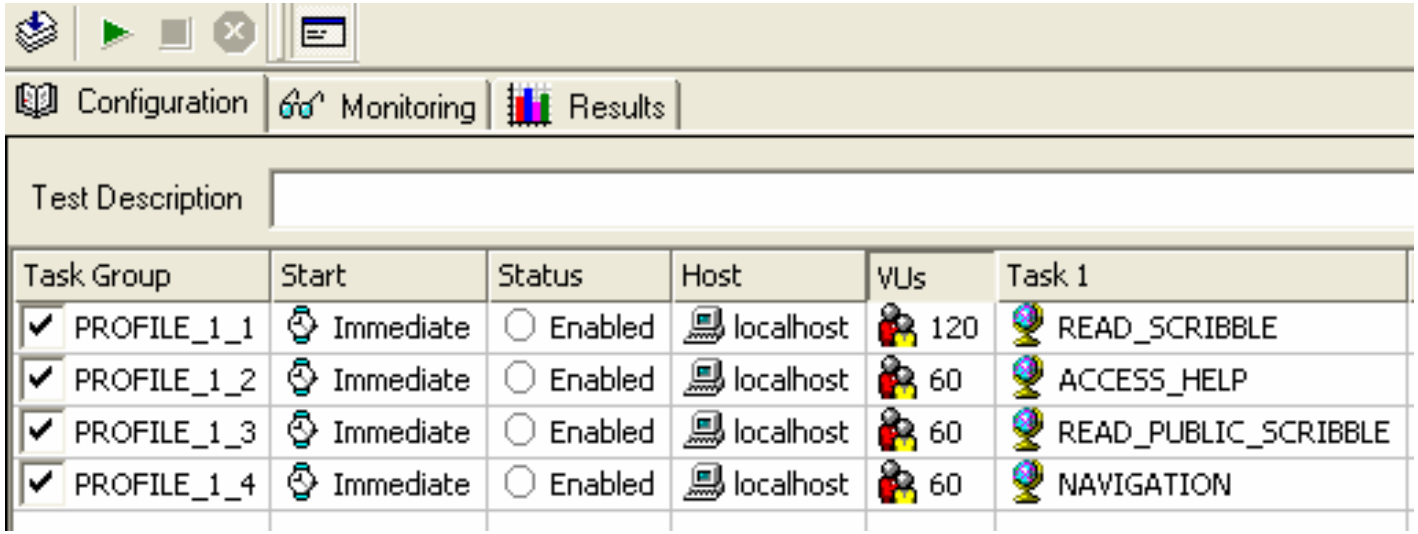
Performance Testing – Steps Involved in OpenSTA (contd.)

- Create operation profiles
  - Operation profile is a probability distribution describing the frequency with which selected important operations are exercised to test the applications
- ***Sample Profile***

Profile Name : Profile -4		
Profile Description	***Workload Type***	***Workload percentage***
Involves miscellaneous operations	Editing user settings	20%
	Meta operations on scribbles	40%
	Organizing scribbles (print basket)	20%
	Navigational operations	10%
	Logout and re-login operations	10%

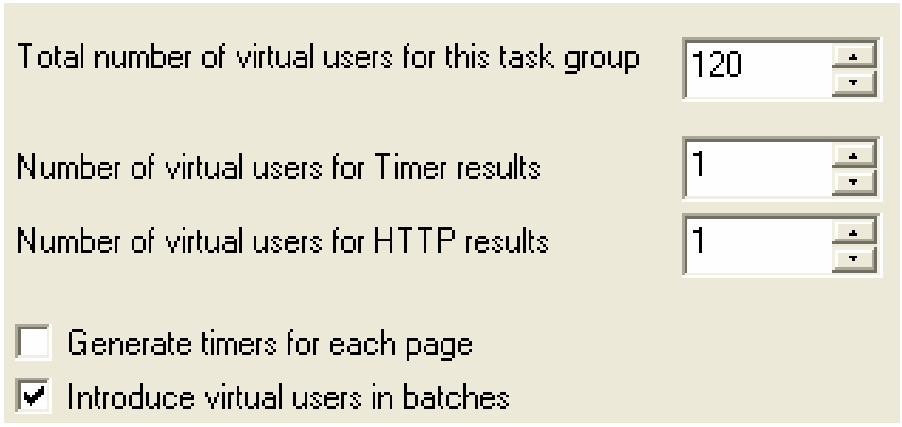
Performance Testing – Steps Involved in OpenSTA (contd.)

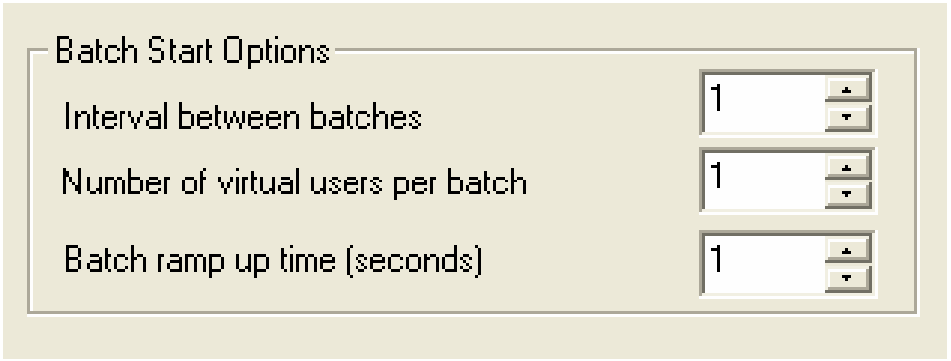
- Create Task groups
  - Task group is a collect of scripts (for a operational profile) that to be executed for testing.
- ***Sample Task Group***



Performance Testing – Steps Involved in OpenSTA (contd.)

- Set the load settings
  - Load settings involves the number of virtual users, number of iterations and load varying settings.
  - ***Eg: Load settings***





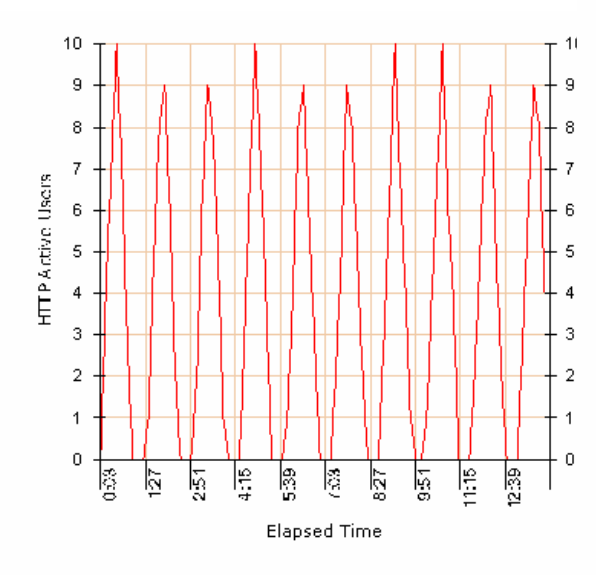
Performance Testing – Steps Involved in OpenSTA (contd.)

- Run the tasks and analysis the results
- Test Summary Snapshots
  - Information about average time connection, HTTP requests, etc...
- HTTP Data List
  - Provides details on HTTP requests and response time
- HTTP Data Graph[1)]
  - Various graph on HTTP data against time and user data

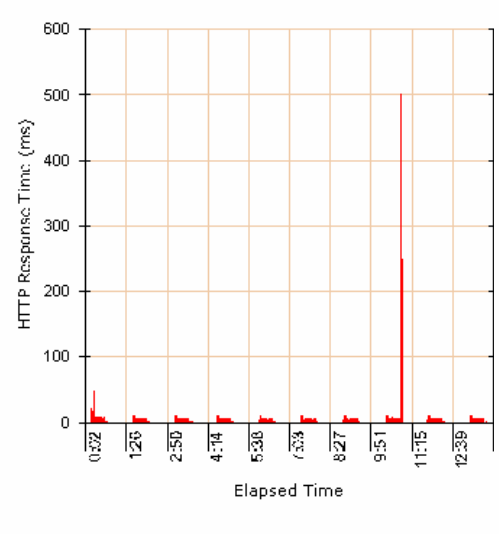
1) *http://www.opensta.org/docs/ug/os-resul.htm#48027*

HTTP Graphs – Under varying load

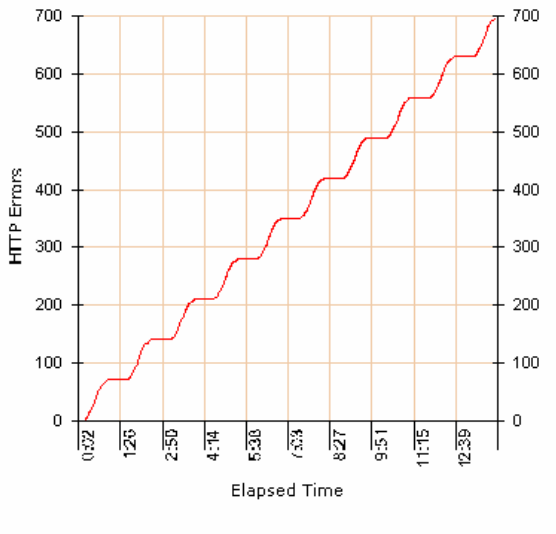
Total virtual users	100
Interval between batches	60s
Number of virtual users per batch	10
Batch Ramp up time	30s



HTTP Active Users *vs* Elapsed Time



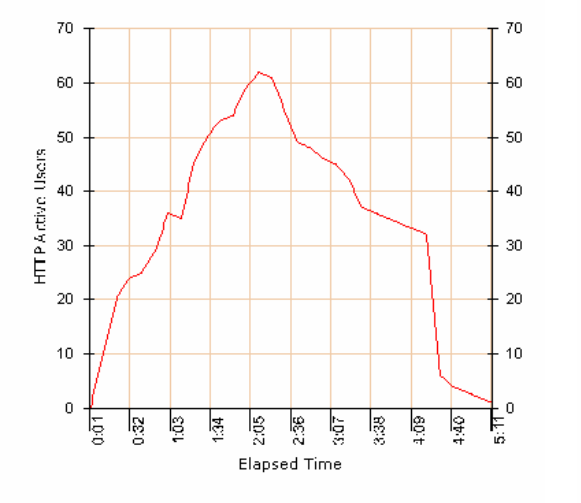
HTTP Response Time *vs* Elapsed Time



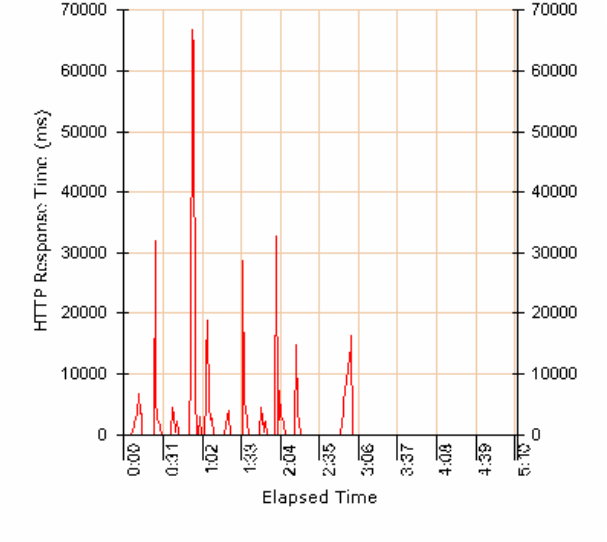
HTTP Errors *vs* Elapsed Time

HTTP Graphs – Under constant increasing load

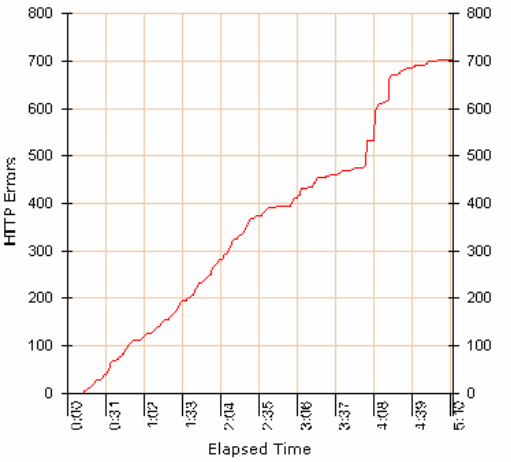
Total virtual users	100
Interval between batches	10s
Number of virtual users per batch	10
Batch Ramp up time	5s



HTTP Active Users *vs* Elapsed Time



HTTP Response Time *vs* Elapsed Time

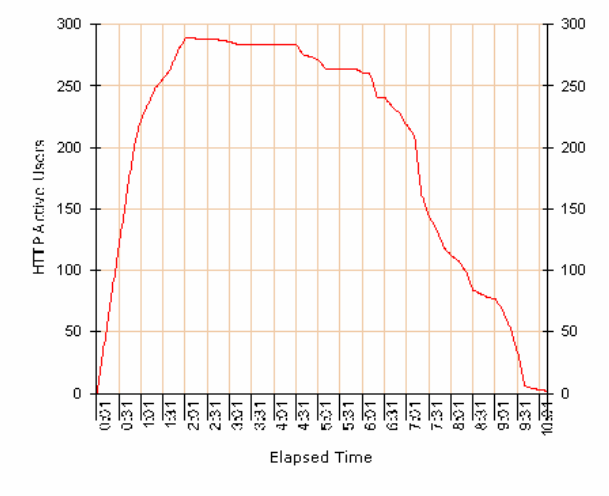


HTTP Errors *vs* Elapsed Time

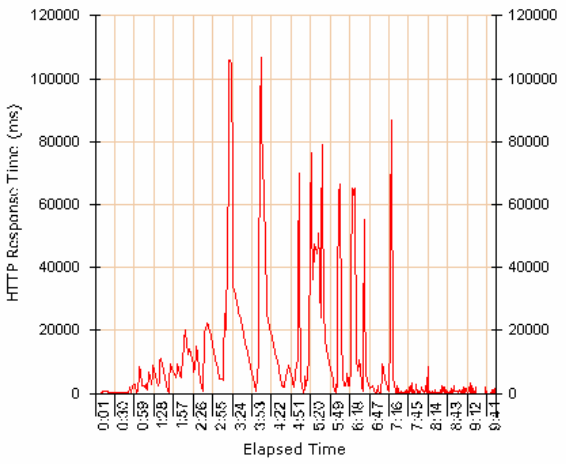
Performance Testing – Result Analysis

Profile-1

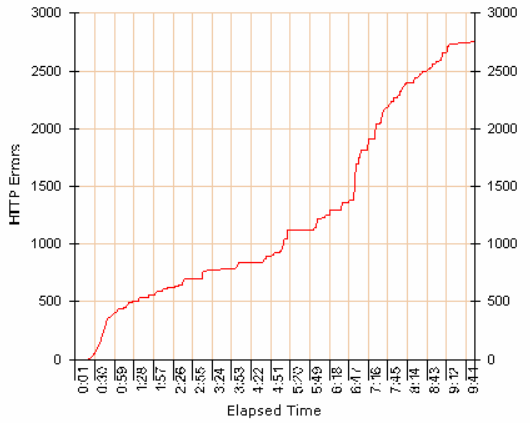
Operations	Read Private Scribble	Accessing Help	Read Public Scribble	Navigation
Total virtual users	120	60	60	60
Interval between batches	1s	1s	1s	1s
Number of virtual users per batch	1	1	1	1
Batch Ramp up time	1s	1s	1s	1s
Iteration(s)	1	2	1	1



HTTP Active Users *vs* Elapsed Time

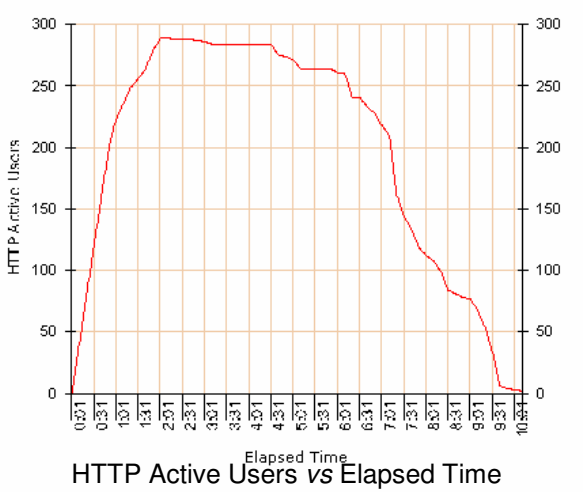


HTTP Response Time *vs* Elapsed Time

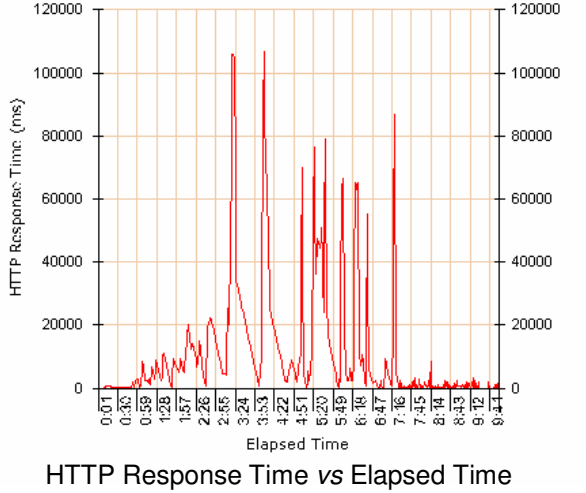


HTTP Errors *vs* Elapsed Time

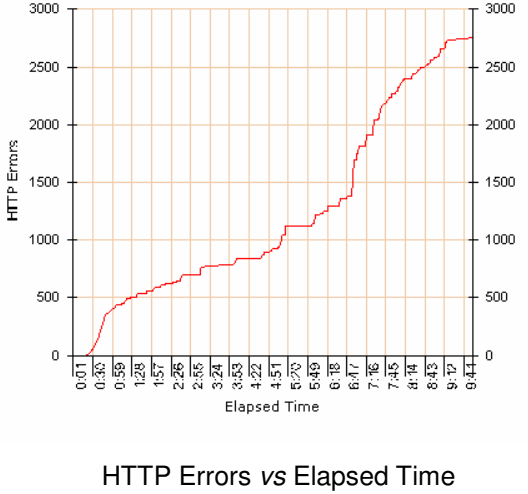
Performance Testing – Result Analysis (contd.)



HTTP Active Users *vs* Elapsed Time



HTTP Response Time *vs* Elapsed Time



HTTP Errors *vs* Elapsed 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

Conclusion

- System (Server) 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).