Performance Test

MEDEVANTAGE

Performance Test Plan

VERSION 1.1

Revision History

Document/Dept. Owner: Performance Engineering			
Created By:			
Revised By	**Nature of Changes**	**Date Revised**	**New Version #**
Compumatrice Team	Initial Performance Test	06/06/2020	1.1

Contents

1. Executive Summary

The purpose of this document is to describe the Performance test plan for Medevantage

This document contains the following information:

- Establishment of performance test goals
- List of items that are in scope and out of scope
- List of Assumptions and Dependencies
- Detailed description of test phases and test cycles
- Detailed description of testing tools
- List of test environments and hardware specifications
- Detailed steps of the business scenario
- List of metrics to be measured and the expected criteria
- List the interfaces to be included in the performance test
- Test data details
- Test Schedule
- Explanation of the differences between load test, duration test and stress test

Application Performance Testing will focus on the elements within the application team's control, primarily the application code that is written and maintained by the application team. Response and capacity criteria will be evaluated within the performance lab environment.

2. Objective

The main objectives for performance testing are:

- Baseline application performance under current load and usage.
- Identify the **application breakage point**
  - Unresponsive - App characteristics: Failed requests or timeouts, Unacceptable Page load times, User inability to interact with app, Application errors.
  - System/Hardware Characteristics: CPU pegged above defined break point, IO thrashing. and excessive swapping, Memory errors.
- Monitor the Application and Web Server Resource utilization.
- To verify that the systems under test meet the business performance requirements.
- To provide project stakeholders with information on how much the load the system under test can handle.

3. Scope

Items in Scope

Table 1 - Functions in Scope

Feature	Comment	Operation
CRM Login	CRM Login Page	Read/Write
Dashboard	Summary of Patient and providers data	Read
Create Patient	Details of patient's data	Read/Write
Search Patient	Details of patient's data	Read/Write
Schedule Appointment	Provider's Specialty and location data to search	Read/Write

Test cases for scenarios listed in scope.

- **Baseline Tests** – Baseline test is application performance under current typical usage conditions. 200 to 300 users are our baseline.
- **Load Test** - To verify application behavior under normal and peak load conditions
- **Capacity Test** - To determine how many users and/or transactions a given system will support and still meet performance goals
- **Endurance Test** –To determine application behavior of the expected amount of load over 8 hours
- **Stress Test** - To determine or validate the application's behavior when it is pushed beyond normal or peak load conditions

Items Out of Scope

- End user desktop simulation
- Functional testing
- Simulating user traffic from different Locations
- Email communication
- Other infrequently used functions not listed in Table 1 above

4. Assumptions and Dependencies

There are assumptions made for this performance testing effort:

- **Environment**
  - ✓ Test Environment is available for testing with latest code.
- **Development Team Responsibilities**
  - ✓ Setting up users and required data to complete the performance test scripts
  - ✓ For any Database restores/refresh to the Test Environment (masked data)
  - ✓ Notifying the performance team with the package deployments and changes included

5. Performance Requirements

The following metrics will be recorded and reported in the Performance Test Results Summary.

Requirement	Success Criteria
IT Requirements	
% CPU Utilization	Average % of CPU utilized during the test Acceptable: <= 80%
% Server Memory Utilization	Average % of Memory utilized during the test Acceptable: <= 80%
Project Requirements	
Function Response Time – CRM Login	Acceptable: <= 5 seconds
Function Response Time – Dashboard	Acceptable: <= 10 seconds
Function Response Time -- Create Patient	Acceptable: <= 10 seconds
Function Response Time -- Search Patient	Acceptable: <= 10 seconds
Function Response Time--Schedule Appointment	Acceptable: <= 10 seconds
Total Concurrent Users	Acceptable: 200-300

6. Test Cases and Workload Model

The test cases and result listed below are for the Book Appointment API.

6.1 Business Scenario/Test Cases

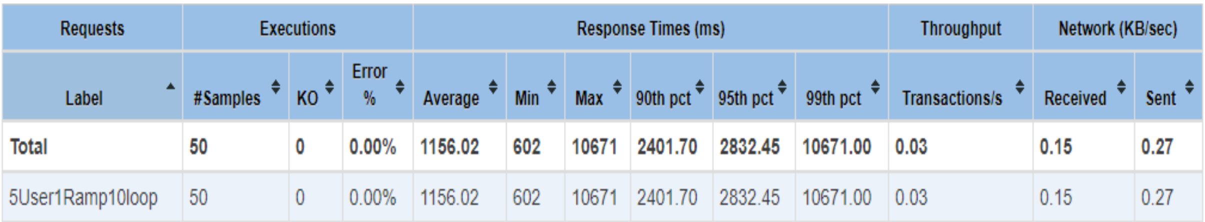
This below test shows the API and system performance with 5 users, here all these 5 users book total 50 appointments where each user is booking 10 appointments in this 30 mins test. To achieve real time scenario, added Think Time (constant timer) into Jmeter test suit which will add 3 min delay before book each appointment.

API: http://172.18.0.76:8083/v1/appointments [Method: POST]

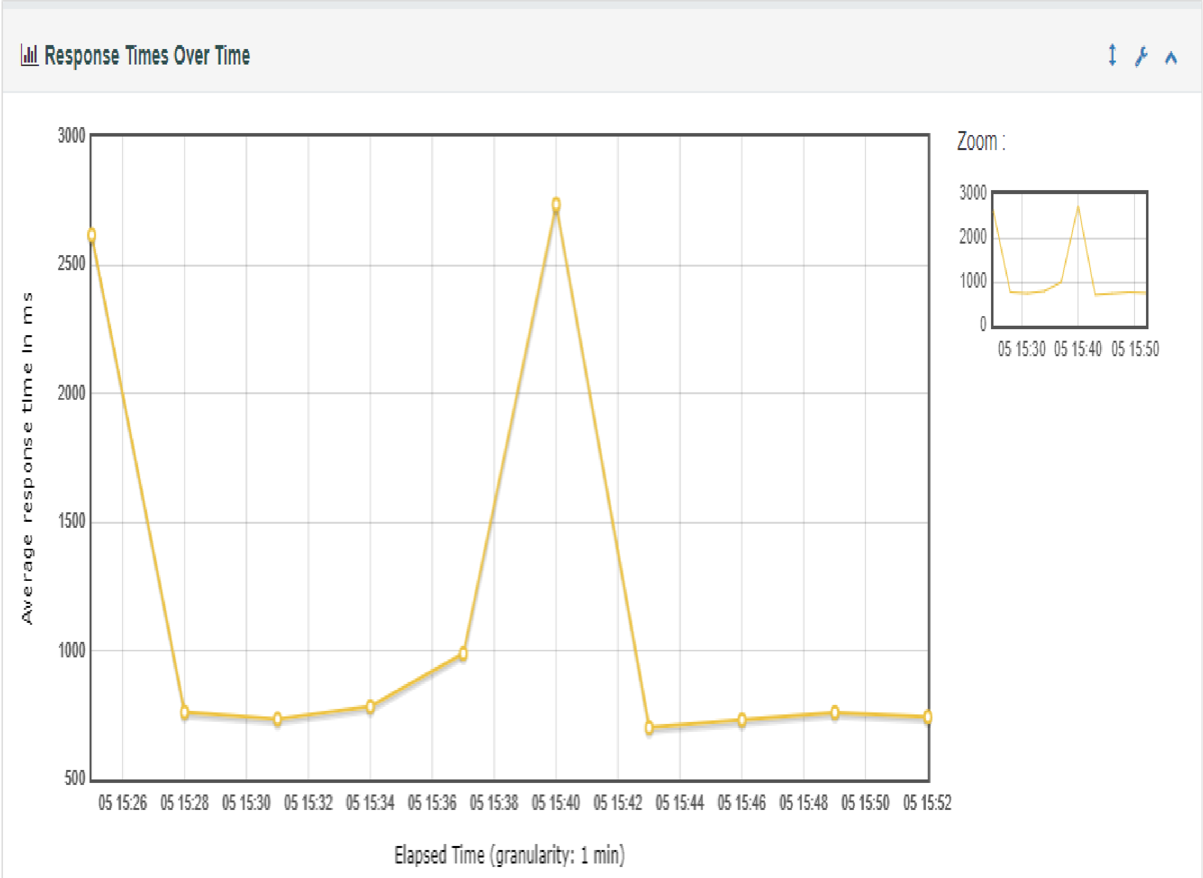
Request Data: Importing appointments data using .csv file.

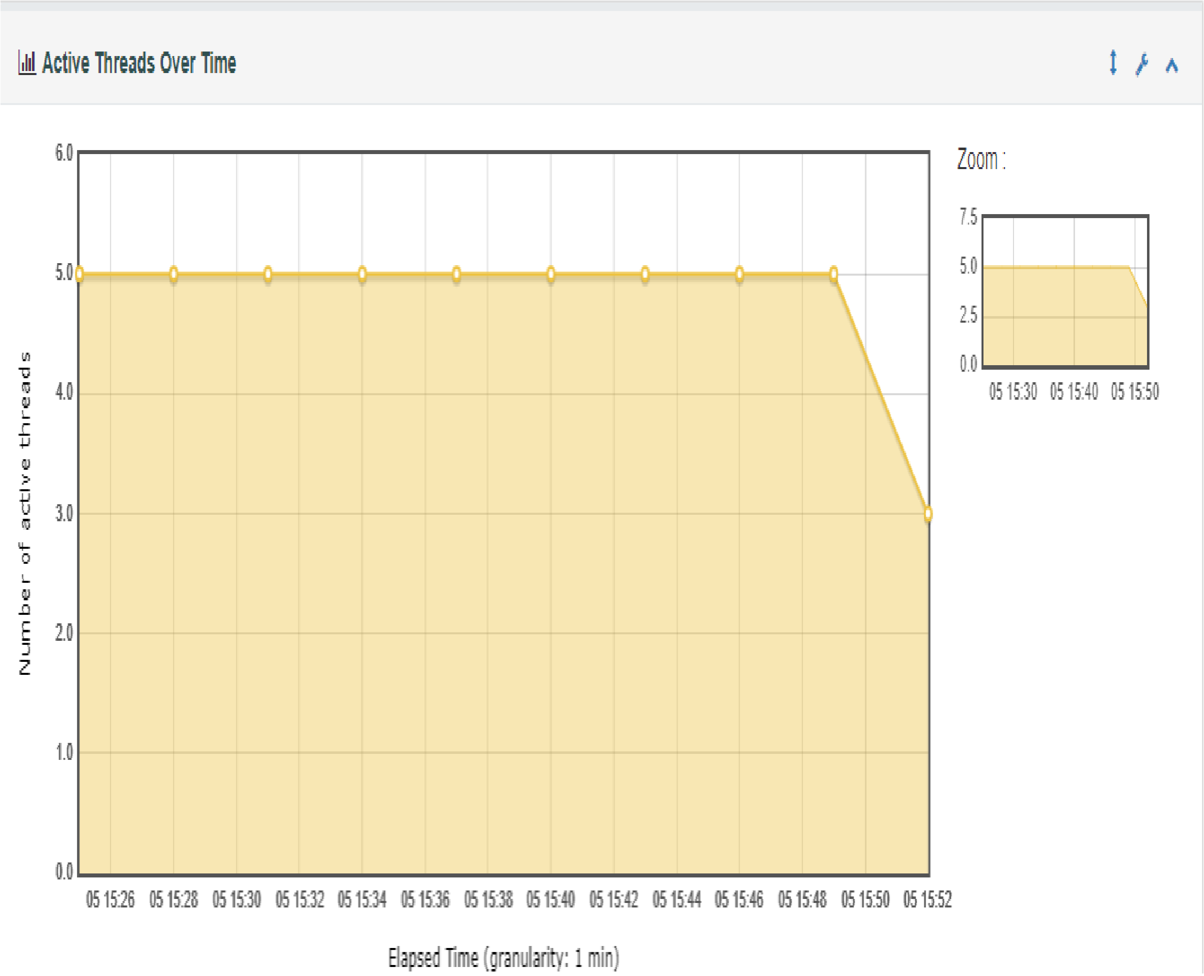
Response Data: It is verifying using "View result tree" listener of Jmeter.

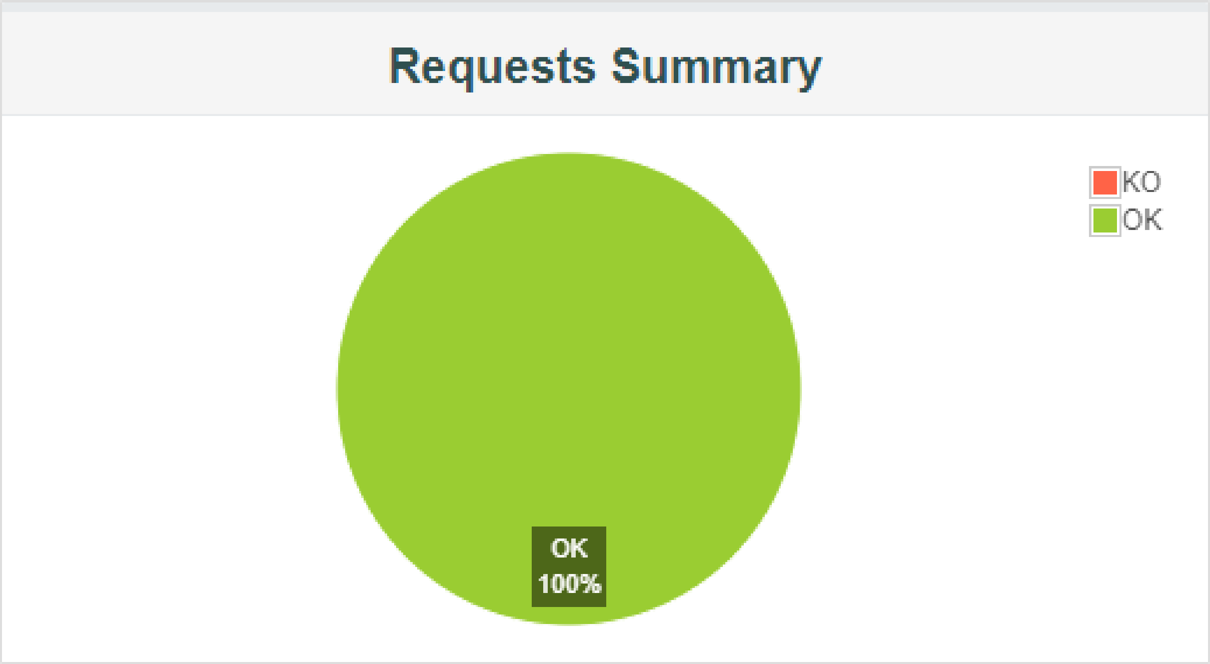
Performance Test: 5 users ramped up with 1 sec and sustain for 30 min with 3 min think time.



Requests	Executions			Response Times (ms)						Throughput	Network (KB/sec)	
Label	#Samples	KO	Error %	Average	Min	Max	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
Total	**50**	**0**	**0.00%**	**1156.02**	**602**	**10671**	**2401.70**	**2832.45**	**10671.00**	**0.03**	**0.15**	**0.27**
5User1Ramp10loop	50	0	0.00%	1156.02	602	10671	2401.70	2832.45	10671.00	0.03	0.15	0.27




Active Threads Over Time
Number of active threads
6.0
5.0
4.0
3.0
2.0
1.0
0.0
05 15:26
05 15:28
05 15:30
05 15:32
05 15:34
05 15:36
05 15:38
05 15:40
05 15:42
05 15:44
05 15:46
05 15:48
05 15:50
05 15:52
Elapsed Time (granularity: 1 min)
Zoom :
7.5
5.0
2.5
0.0
05 15:30
05 15:40
05 15:50


Requests Summary
KO
OK
OK
100%

6.2 Business Scenario/Test Cases

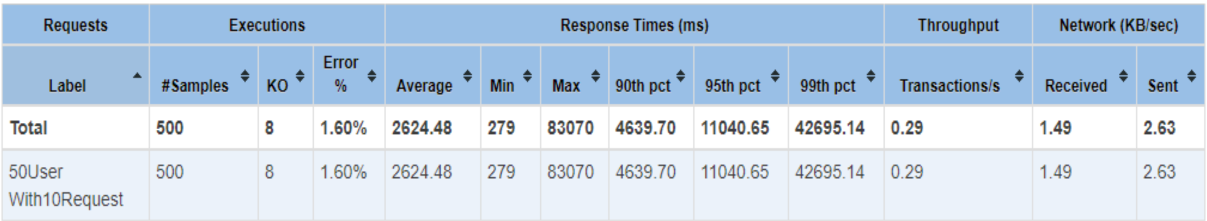
This below test shows the API and system performance with 50 users, here all these 5 users book total 500 appointments where each user is booking 10 appointments in this 30 mins test. To achieve real time scenario, added Think Time (constant timer) into Jmeter test suit which will add 3 min delay before book each appointment.

API: http://172.18.0.76:8083/v1/appointments [Method: POST]

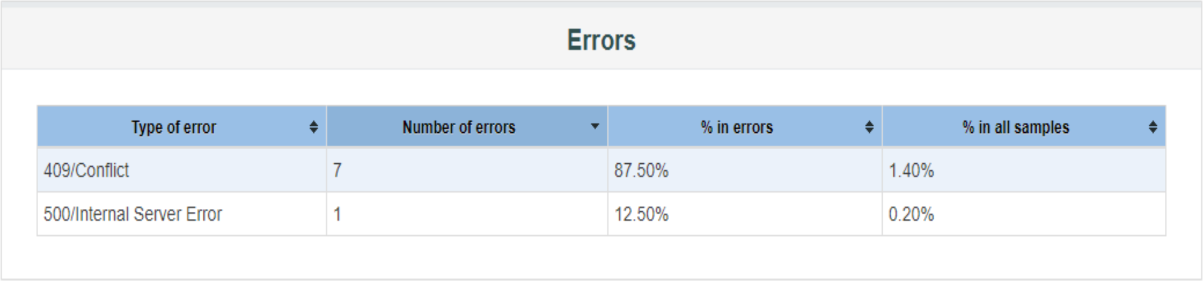
Request Data: Importing appointments data using .csv file into Jmeter

Response Data: It is verifying using “View result tree” listener of Jmeter.

Performance Test: 50 users ramped up with 1 sec and sustain for 30 min with 3 min think time.

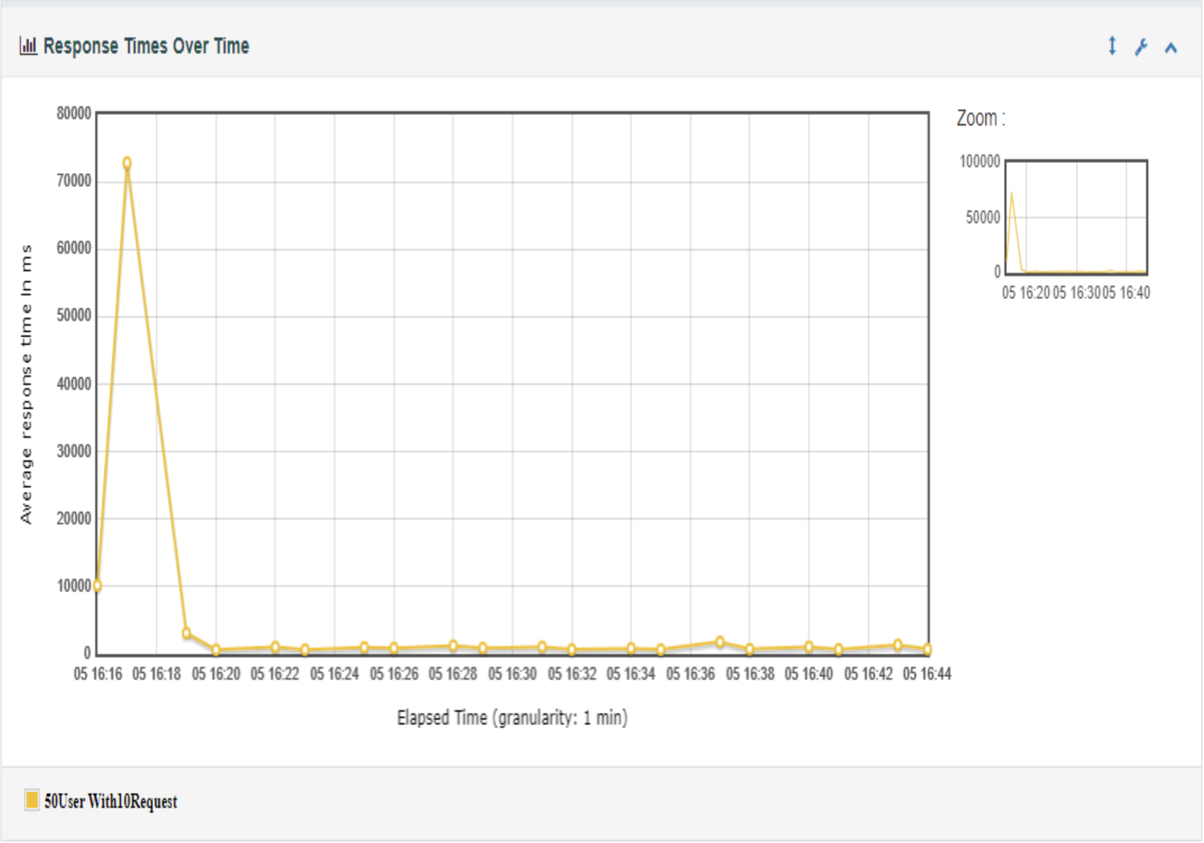


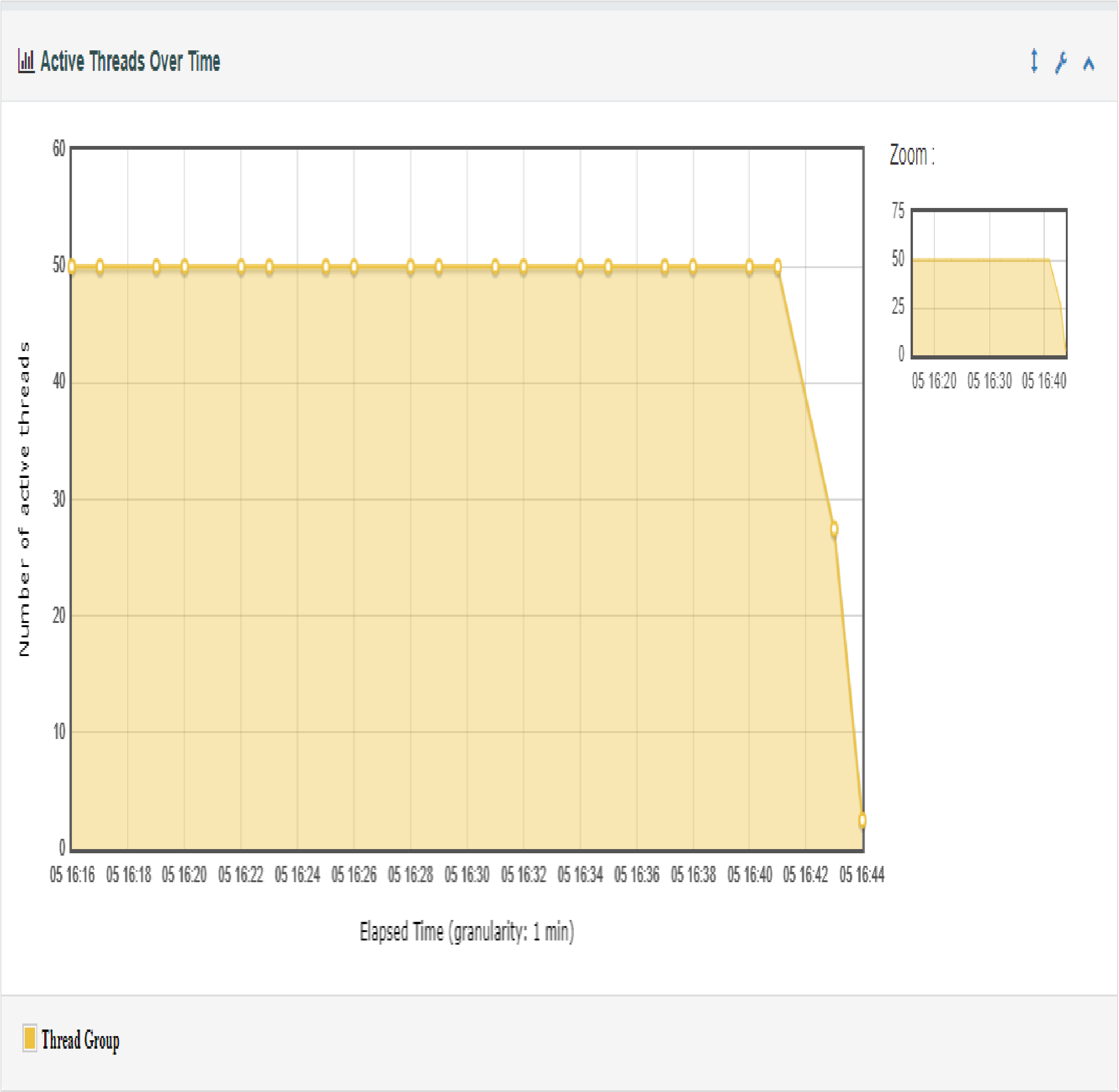
Requests	Executions			Response Times (ms)						Throughput	Network (KB/sec)	
Label	#Samples	KO	Error %	Average	Min	Max	90th pct	95th pct	99th pct	Transactions/s	Received	Sent
Total	**500**	**8**	**1.60%**	**2624.48**	**279**	**83070**	**4639.70**	**11040.65**	**42695.14**	**0.29**	**1.49**	**2.63**
50User With10Request	500	8	1.60%	2624.48	279	83070	4639.70	11040.65	42695.14	0.29	1.49	2.63

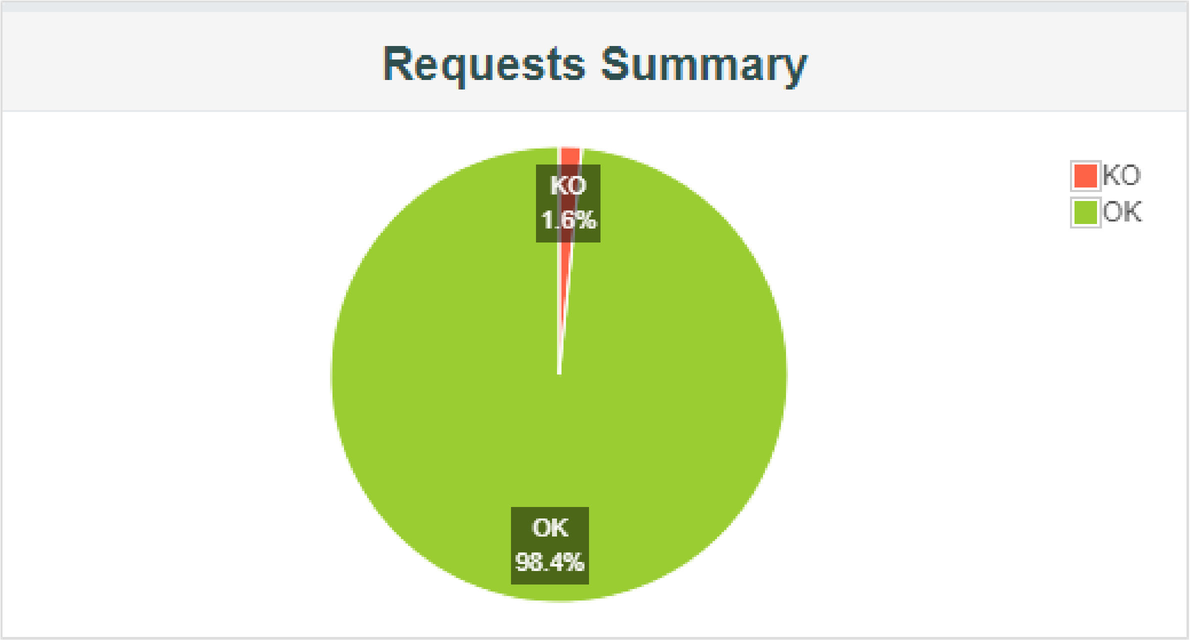


Errors

Type of error	Number of errors	% in errors	% in all samples
409/Conflict	7	87.50%	1.40%
500/Internal Server Error	1	12.50%	0.20%




Active Threads Over Time
Number of active threads
60
50
40
30
20
10
0
05 16:16
05 16:18
05 16:20
05 16:22
05 16:24
05 16:26
05 16:28
05 16:30
05 16:32
05 16:34
05 16:36
05 16:38
05 16:40
05 16:42
05 16:44
Elapsed Time (granularity: 1 min)
Zoom :
75
50
25
0
05 16:20
05 16:30
05 16:40
Thread Group


Requests Summary
KO
1.6%
OK
98.4%
KO
OK

7. Test Environment

Test Environment Specifications

The chart below provides detailed information about test systems to be used for all performance test execution. This information is vital in understanding and interpreting the test results, and is a crucial input into capacity modelling exercises.

Testing Environment

Server	Hardware Specifications	Software Specifications	FQDN	Data Center
Azure Server	EX: m4. xlarge (AWS Instance)	EX: Apache/PHP	N/A	AWS
DB	EX: m4.2xlarge (AWS RDS Instance)	EX: MySQL	N/A	AWS

8. Production Environment

Production Environment Specifications

The chart below provides detailed information about the Production environment configuration. This information is vital in validating test results, and is a crucial input into capacity modelling exercises

Testing Environment

Server	Hardware Specifications	Software Specifications	FQDN	Data Center
Prod Server	EX: m4. xlarge (AWS Instance)	EX: Apache/PHP	N/A	AWS
DB	EX: m4.2xlarge (AWS RDS Instance)	EX: MySQL	N/A	AWS

9. Test and Monitoring Tools

- **Jmeter**
- **Postman**

Appendix A – Test Phases

Test Phases

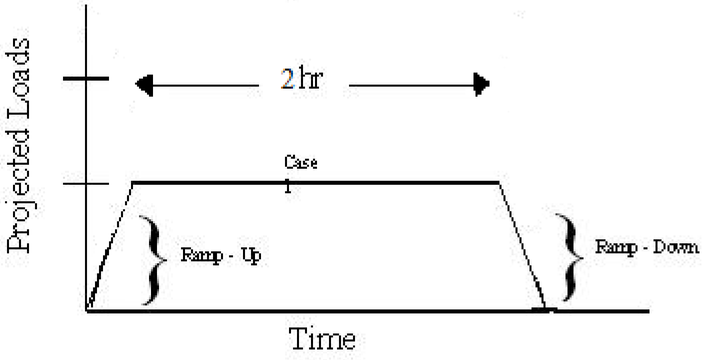
The Performance test will be broken down into five phases: Planning, Preparation, Execution, Tuning, and Analysis.

1. The **Planning phase** of Performance test develops the specific approach and plan that will be used to organize and execute the Performance test
2. The **Preparation phase** of Performance test prepares test data, Performance test scripts, and the test environment; assesses requirements for the execution phase (and for Capacity Modelling if required).
3. The **Execution phase** of Performance test executes the performance test scripts created in the preparation phase. Performance test execution will produce results based on stress, load and duration tests.
4. The **Tuning phase** of the Performance test identifies and resolves performance issues within the application components prior to performance evaluation to ensure optimal overall application performance
5. The **Analysis phase** of Performance test consolidates and presents the test results generated during test execution. The results will be provided to the application team for analysis.

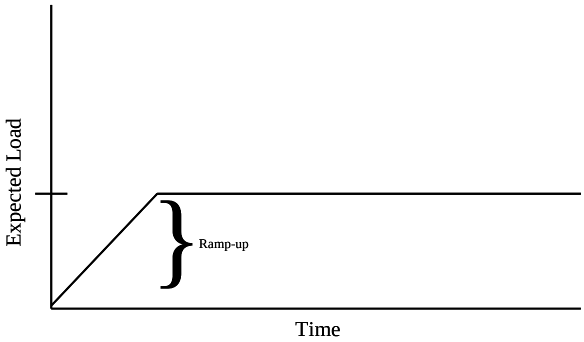
Appendix B – Test Types and Descriptions

For the purposes of this project, Load, Duration and Stress testing will be defined as:

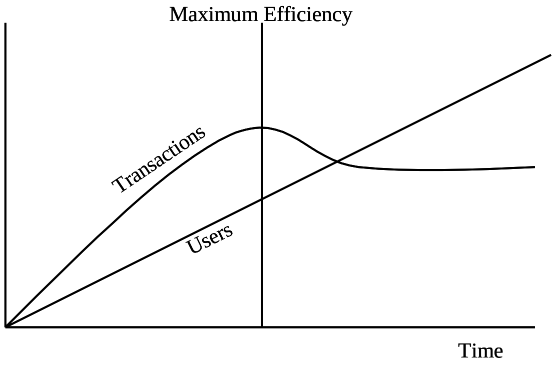
- **Load Test** (Normal, Break/Fix and Emergency Release Tests) with 2 hours – Assess application performance results (as opposed to environmental performance such as network latency, backend application response time, etc.) based on the current and projected user volumes. Load tests will identify whether an application can support current and projected loads as defined by the performance test criteria. Load tests require environments that are as close to production as feasible to produce realistic response time and transaction results. Virtual users will be configured to execute transactions based on the current/proposed volumes for the application.



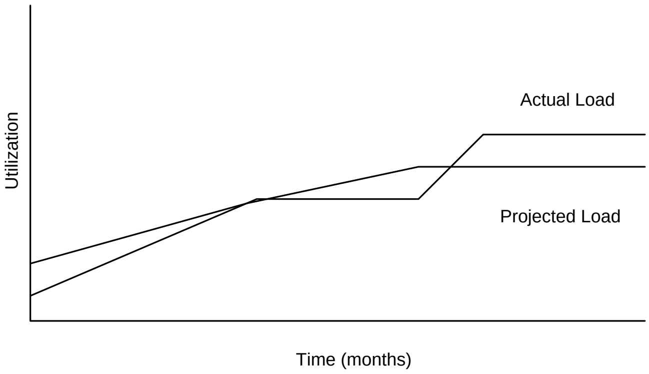
- **Duration Test** (Normal and Emergency Release tests) – Test with a constant load over a period between 5 to 8 hours. Duration tests will be conducted to determine if an application's performance degrades over an extended period of time. This type of testing may indicate if there are issues, such as memory leaks, that could occur after prolonged period of the testing.



- **Stress Test** (Normal tests) – Incrementally increase the application activity to identify the maximum efficiency of the application within a controlled environment and identify the break point of the application and server. The application maximum efficiency is the point at which the application reaches its peak ability to process the transactions within a given period of time by increasing the user load (e.g 2x, 3x, 4x… etc). It is used to understand how the application satisfies scalability and stability criteria. It will also identify areas for application and architecture remediation. Stress tests could be performed in an environment that is similar to production.



- **Capacity Test** (Normal Tests) – A load test run at a higher transaction and/or user rate based on business projections of expected or potential user load. This test is meant to identify potential future system bottlenecks early so they can be corrected as early as possible.



Appendix C – Definitions

- **Vuser – Virtual User:** LoadRunner replaces human users with *virtual users* or *Vusers*. Vusers emulate the actions of human users by performing typical business processes. Each action that a Vuser performs submits input to the client/server system. By increasing the number of Vusers, you increase the load on the system. While a workstation accommodates only a single human user, many Vusers can run concurrently on a single workstation.

- **Ramp-up period (in seconds):** It is the amount of time it will take JMeter to add all test users (threads) to test execution. In other words, how long it will take to JMeter to start execution of all the users.
  Ex: If we have no. of thread(users) 5 and ramp up period is 10 secs then jmeter call 5 users within 10 sec. i.e. Jmeter will give 2 sec to every user to call.

- **Think Time:** In real world user is not move quickly from one page to another page. Every user is spent some time in sec on any page then it will be move to another page. Here, Delay timers are needed to add for getting accurate load testing results.

- **Aggregate Report:**
  Median: This time shows half request take more time than this time and half of the request take less time from this time.
  90% Line: This time shows 90% request less than this time.
  95% Line: This time shows 95% request less than this time.
  99% Line: This time shows 99% request less than this time. This time is normally equal to Max Time column value.
  Throughput: The throughput is the number of requests per unit of time (seconds, minutes, hours) that are sent to your server during the test.
  Error %: This denotes the error percentage in samples during run. This error can be of 404(file not found), 500 (Internal Server Error) or may be exception or any kind of error during test.