BugRaptors
All Ways of Quality Always

CASE STUDY ON LOAD AND PERFORMANCE TESTING

ABOUT BUGRAPTORS

BugRaptors a CMMi level5 company is preeminent independent software testing organization, led by committed, young, and dynamic professionals with extensive expertise and experience of independent testing and QA consulting services. We maintain a distinguished level of testing services which blends top quality with cost effective solutions. The proficiency we have gained in software testing and QA services over the years by implementing latest tools and technologies ensures that client's product is launched with highest degree of quality.

THE CLIENT

The client is a leading application development and application service provider (ASP) in the domain of banking and finance. Application is a finance tracking application in which client wants to manage all his customer's financial activities and updates. Client developed a mobile application for android and iOS platforms. App has several user roles involved in the system and system is with web and mobile interfaces for particular type of users.

The application was about to track and manage financial information of its customers. In which bank provides loan facility to the customers and then the agents track and update the information of customers who need the loan, need to submit installment and what is the status of current or previous installment of finance etc. Web interfaces was accessible by admin and other super users and mobile app was being used by agents to update and manage customer information.

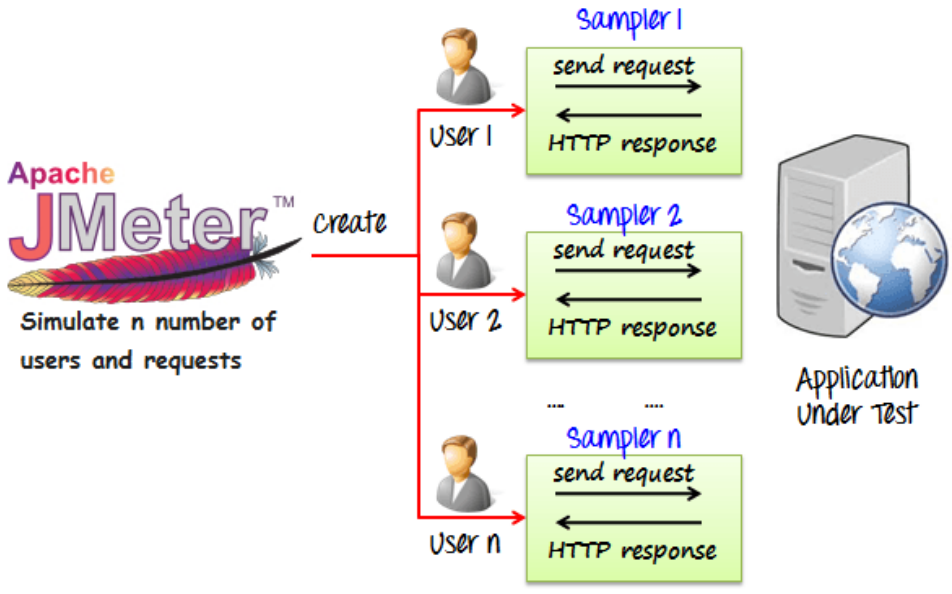
CHALLENGES & REQUIREMENTS

Client's focus was on the mobile application, required an extensive performance testing platform to stress application at the Web Service/ API layer in which we to determine the performance level of the Web Service/ APIs that were used in the application in development phase. Client had the large number of users of the application and expecting more numbers of users in coming days. So, client wanted us to check and let him know that currently how many concurrent and serial users can access the system and run the application without any performance issue or delay.

Being an ASP, client required us to provide performance level of the application ASAP so that client can optimize the APIs (Web Services/ APIs) or upgrade their infrastructure, so that more upcoming user can be accommodated without any issue and slowness of the system. Client had also asked to pin point particular API that is week and not performing up to the point with a comparison between all APIs. Although client gave us an idea of potential number of users and time but still there was a challenge for us to fine the bottlenecks of the system as quick as possible, so that the system can be shifted to upper version of infrastructure or Web Service/ APIs can be optimized.

BUGRAPTORS APPROACH & SOLUTION

After having a discussion on the requirement and functional process of application we had a discussion with our technical experts on load and performance of the application. Then BugRaptors defined a QA strategy compatible to the testing requirements of the client's application. Our first step was to decide on the right tool that would replace Load Runner. Selection of the right tool ensures saving not only valuable time but also money. In order to shortlist the right tool we decided to user jMeter for load and performance of the application that provide flexibility to simulate n number of users and requests.



To arrive at an optimal solution, we decided to migrate the performance testing activities for the application to an open source tool without impacting the quality of the results and analysis provided to the

client i.t. jMeter. And as per defined strategy, BugRaptors had taken following actions to achieve required goal to provide results to client on time.

1. Defined Test Strategy:

As per client requirements we developed a strategy in which we have decided to use an open source tool i.e. jMeter and defined a plan of execution. First we prioritized the APIs to perform load test and as per priority list worked on the APIs one by one. After all individual APIs execution we had a task in hand to get a comparison then we decided to go for parallel execution to get a clear a comparison between the APIs. And deliver the results to client as per priority list parallel and then combined.

2. Identified stable and realistic test data:

It was required to identify the correct test data to perform the load test, this is first thing that we need to work on any application for load and performance testing. So, firstly we identified the test data and also created the same as we need to perform the request in bulk on APIs. We created users in bulk and created other required data that is similar to production environment of the client.

3. Devised a monitoring strategy:

For every task, monitoring is the crucial part in which we have to manage and monitor all the statistics that further defines end results. We had developed a monitoring strategy to keep an eye on ongoing execution of the test. During the monitoring we kept all the APIs and infrastructure statistics that gave us clear picture that how the application and server are behaving before-load, on-load and after-load.

4. Developed test scripts:

In jMeter we already have various preloaded listeners and various types of samplers to get desired monitoring result and create a request. But sometimes we need custom and user defined request. So in this execution we worked on the test scripts that were used in jMeter.

5. Analyzed server performance:

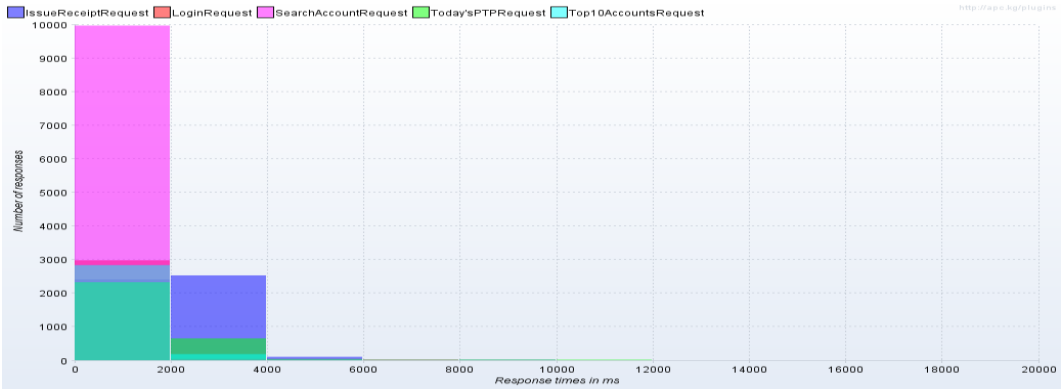
As client wanted to upgrade his infrastructure as per given results of performance test if required, so it was required to monitor server resources at the time of execution. During the execution we monitored all the server resource like CPU, Memory, Disk I/O and Network, to monitor server

resources we used jMeter's "Standard-1.3.1" set plugin that was configured on client server to communicate with our jMeter client.

BENEFITS TO CLIENT:

By executing a well-planned QA process and strategy, BugRaptors helped the client attain the following benefits:

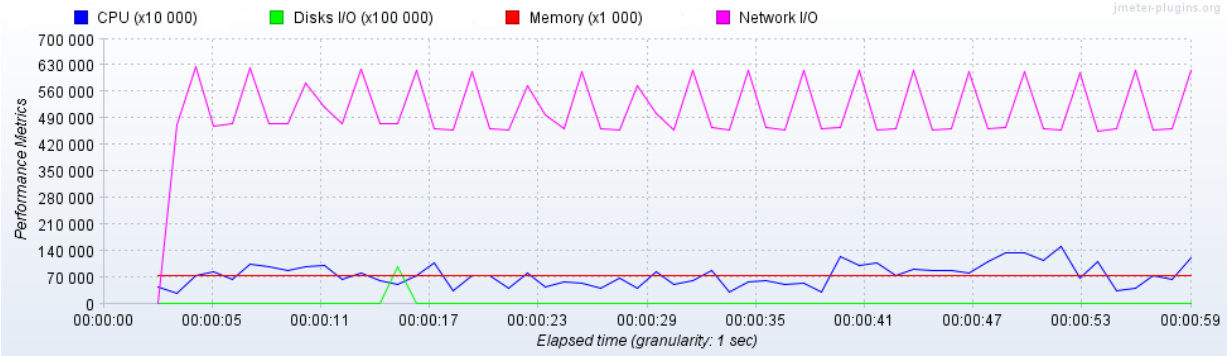
Response time improved 70%: As per results client tuned the APIs and queries to come over 70% improvements in response times.



Reduced risk and overall time for testing life cycle: As the performance of the application was increased and now application was quick responsive this increased overall time of testing and reduced risk of project delays caused by performance issues.

Mat the performance benchmark: As client was expecting more users in coming days, so as per first report client enhanced the overall system and was able to mat the performance benchmark as per expected number of users in coming days.

Identified production server requirement based on performance data: Client analyzed the performance data and shared report then client was clear about the current performance of the server and accordingly client had identified the production server/ infrastructure requirements and was able to determine the monitoring thresholds.


CPU (x10 000)
Disks I/O (x100 000)
Memory (x1 000)
Network I/O
jmeter-plugins.org
Performance Metrics
700 000
630 000
560 000
490 000
420 000
350 000
280 000
210 000
140 000
70 000
0
00:00:00
00:00:05
00:00:11
00:00:17
00:00:23
00:00:29
00:00:35
00:00:41
00:00:47
00:00:53
00:00:59
Elapsed time (granularity: 1 sec)