Mindtree

Performance Test Report For

Execution of 3 months Slot availability

Batch Job

Date: 19th June 2019

Author: Anand Babaleshwar

Summary

This report presents the observations and findings of the 3 months slot availability batch job and load test conducted for a load of 150 users, which will make booking appointments after executing 3 months Slot availability batch jobs running for a duration of 1 hour.

The objective of this load test was to observe and record the behavior of the application when users are booking appointments after executing slot availability for 3 months.

Below are the scenario details:

Sprint/Report Name	Booking appointments after executing slot availability batch job for 3 months
Run Date	19-June-2019
Period	05:19 AM to 06:21 AM (UTC)
Number of concurrent users	150
Ramp up	4 min
Run Duration	60 minutes
Ramp down	2 min

Batch Job execution details:

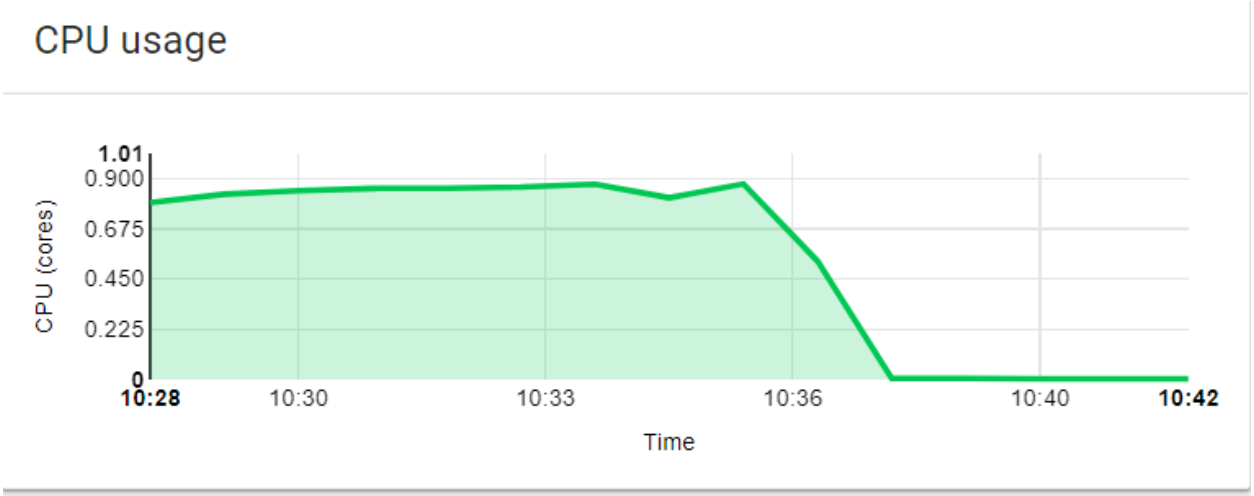
Executed slot availability batch job for 3 months and verified in the DB as well after creation of slots for 3 months, below are the details:

instance_id	Create_Time	Start_Time	End_Time	Execution Time	Status	Slots available (DB)	Days (DB)
1104	2019-06-19 04:50:53.262	2019-06-19 04:50:53.287	2019-06-19 05:06:34.991	00:15:42	Completed	35559	90

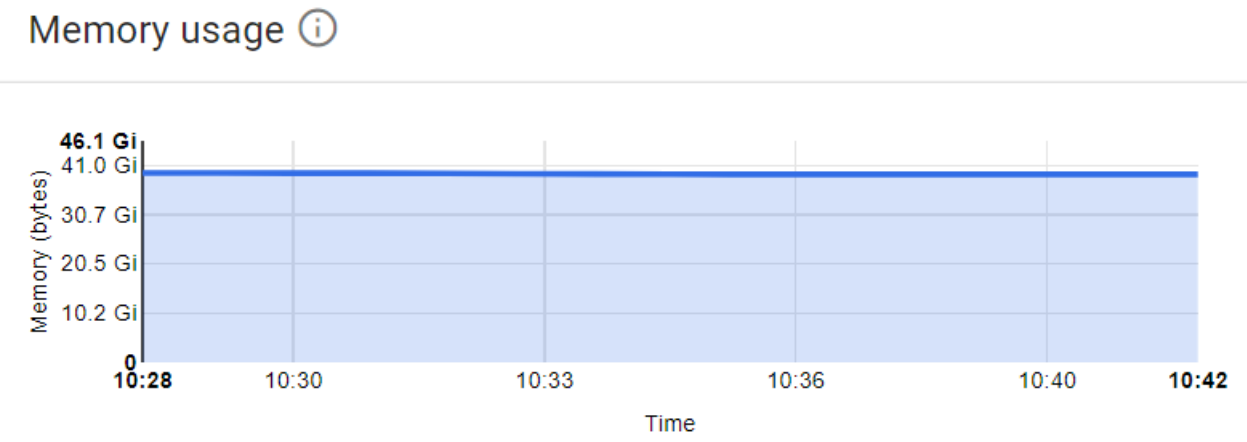
Slot availability batch job took **00:15:42** sec for executing 3 months. Batch job status is completed but its created 3 months slots. Verified the slots available in DB.

We have taken the CPU and memory utilization graphs of the slot availability batch job (pods) , They are below:

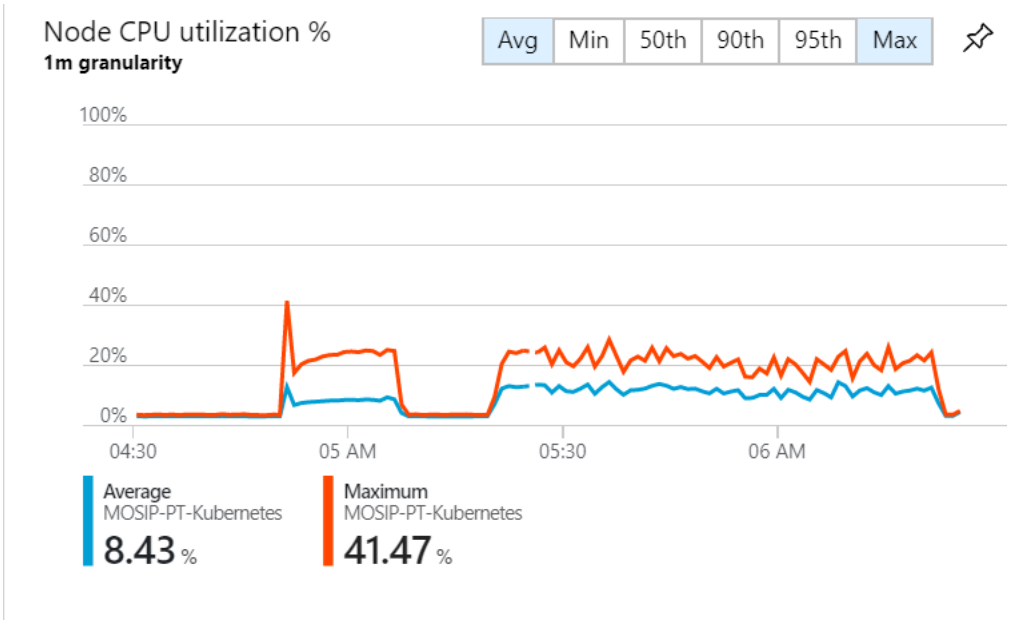
CPU Utilization:

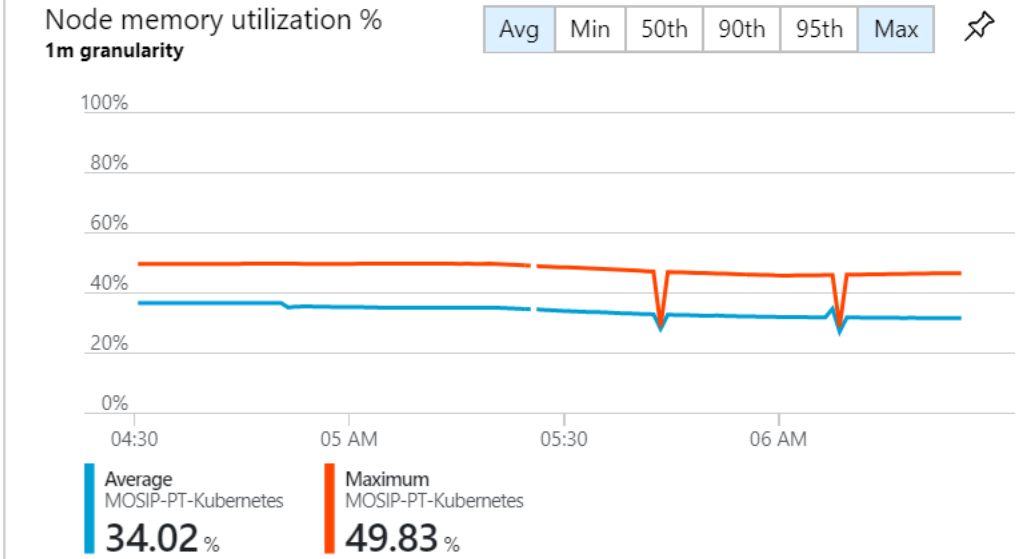


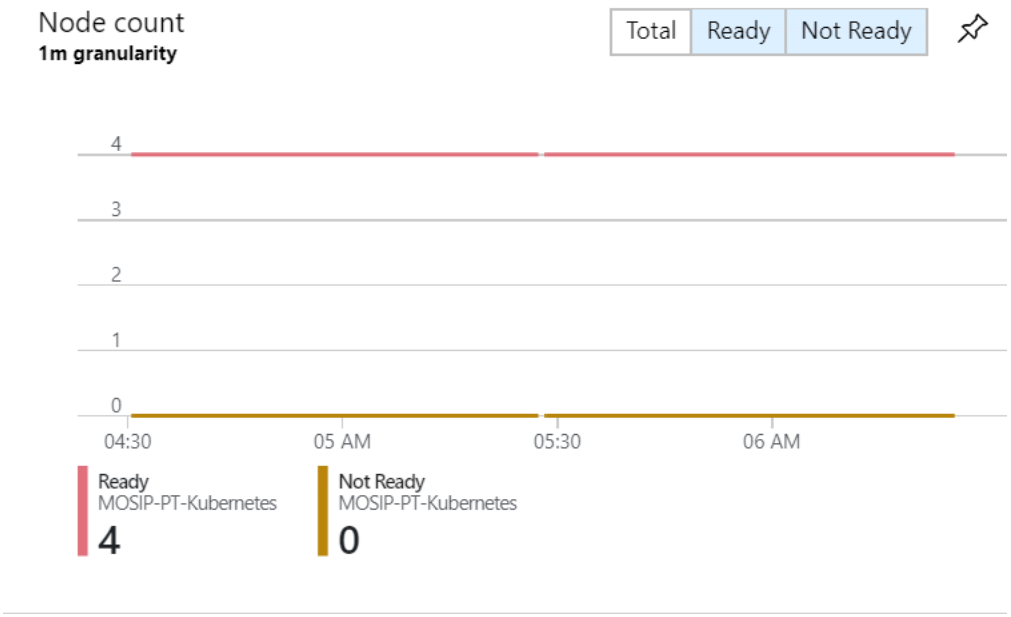
Memory utilization:



CPU and Memory utilization from Kubernates cluster:







After verifying the slots available in DB for 3 months and In-order to create volume in the database executed the booking appointment full flow scenario script the

Details are mentioned below:

The transaction response times observed were as below:

Label	# Samples	Average (msec)	90% Line (msec)	Min (msec)	Max (msec)	Error %	Throughput (Sec)
mosip_preReg_homepage	2296	66583	131368	397	241276	0.00%	0.61267
mosip_preReg_send_otp	2294	386	415	18	1748	0.09%	0.61398
mosip_preReg_validate_otp	2292	187	196	158	1840	0.00%	0.61355
mosip_preReg_submit_demographics	2292	1025	1510	488	9315	0.00%	0.61348
mosip_preReg_upload_poi_document	2290	484	563	385	1195	0.87%	0.6138
mosip_preReg_upload_poa_document	2284	419	493	339	2504	0.00%	0.61326
mosip_open_preview_pageupload_poi_document	2284	15	19	9	225	0.00%	0.61333
mosip_preReg_open_regCeneter_selection_page	2281	142	155	119	364	0.00%	0.61348
mosip_preReg_search_registration_center	2280	65	73	53	266	0.00%	0.61391
mosip_preReg_open_book_appointment_page	2278	578	646	498	1258	0.00%	0.61397
mosip_preReg_book_appointment	2273	330	398	270	1536	0.00%	0.61359
mosip_preReg_logout	2209	67777	135276	600	238413	0.00%	0.59733

Performance Test Execution Details

We have executed the booking appointment user flow scenario script, which has transactions mentioned in above table.

Most of the transactions average response times are beyond SLA of **3** seconds, They are listed below:

1. Mosip_preReg_homepage - 66.583 sec
2. Mosip_preReg_logout- 67.777 sec

The error rate for all transactions is less than 1%.

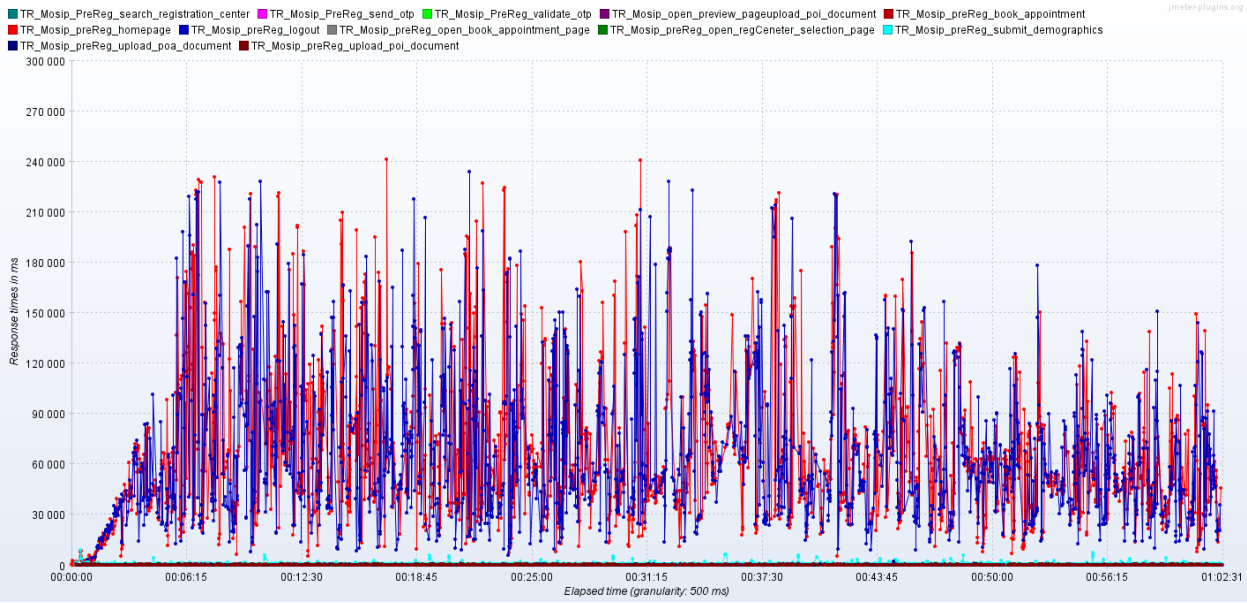
Test Environment

	Common proxy server (NGINX)	(Kubernets cluster) apache Tomcat 8.5.31	DB Postgress SQL 10.2
Number Of nodes	1	4	1
RAM	4 GB	112 GB	16GB
PROCESSOR	2 cores	16 core	4 cores

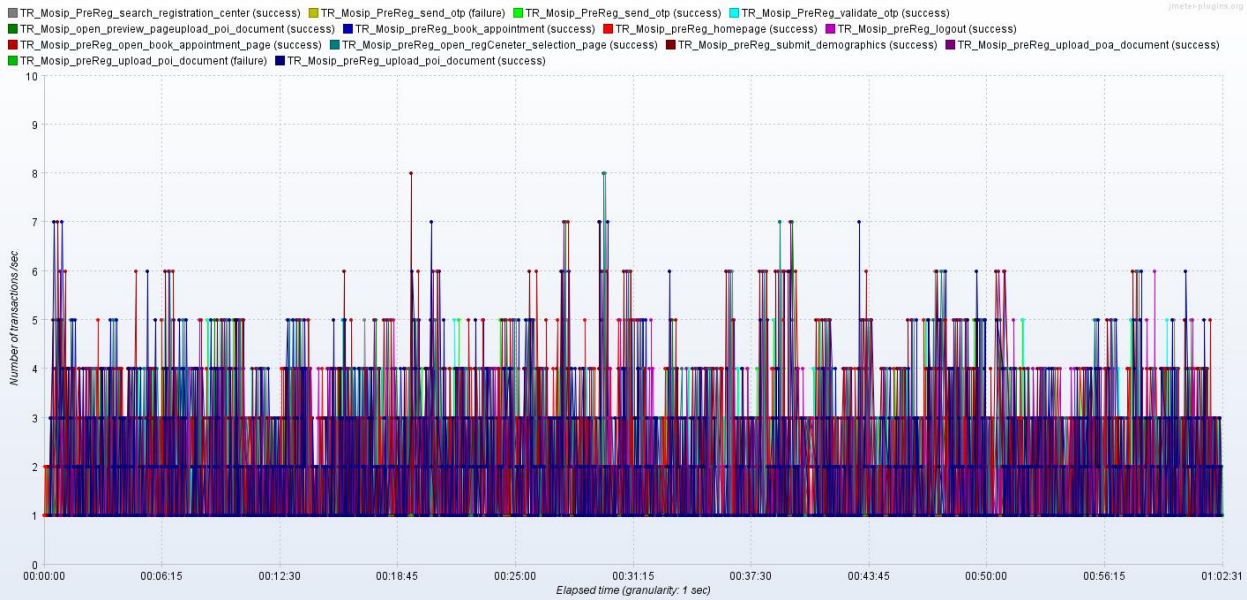
Response Time Graph

All the transactions average response times are within SLA of 3 seconds except below:

1. Mosip_preReg_homepage - 66.583 sec
2. Mosip_preReg_logout- 67.777 sec



Transactions per second:



Conclusion and Next Steps:

The Execution for 3 months slots availability is completed, the high response times observed during 150 users load test which was executed for 1 hour steady period and similar issue is covered in the ticket https://mosipid.atlassian.net/browse/MOS-25581 and We will proceed slot availability batch for future periods like for 4 months ,5 Months and 6 Months and we will repeat the same performance test and observe the performance of transactions and execution time for batch jobs