Mindtree

Performance Test Report

For

Execution of

Kernel PRID generator API – 200 users

Date: 7 May 2020

Author: Anand Babaleshwar

Summary

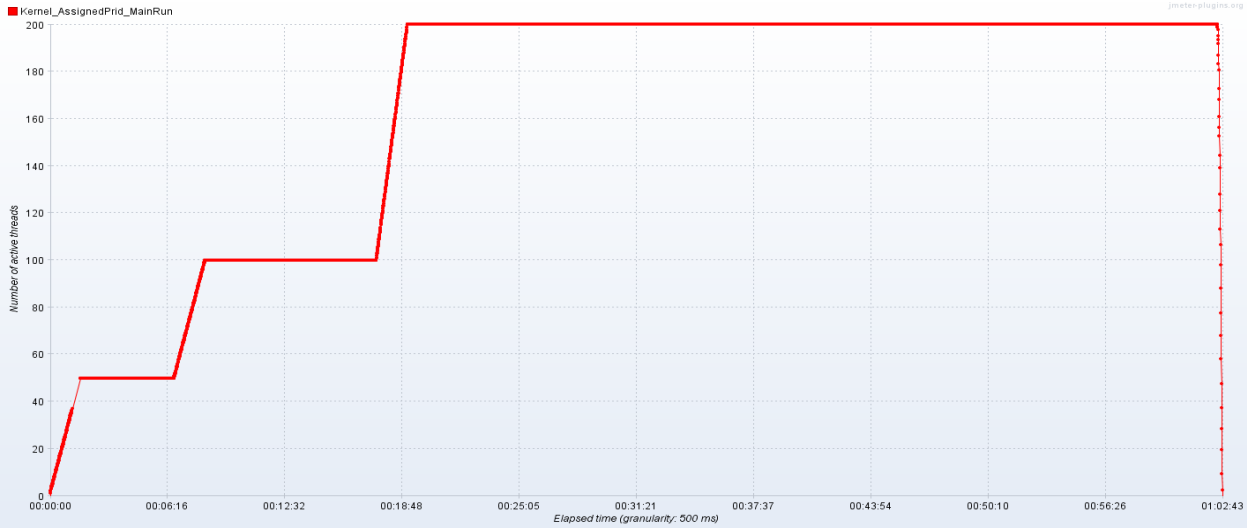
This report presents the observations and findings of the load test conducted for a load of 200 concurrent users on kernel prid generator service

The objective of this load test was to observe and record the behavior of the application when users are calling kernel prid generator service API

Below are the scenario details:

Script/Report Name	Kernel prid generator
Run Date	07-May-2019
Period	07:14 to 08:17 AM (UTC)
Number of concurrent users	0 to 50 to 200
Ramp up	Refer pic of **Concurrent users Ramp up pattern shown below**
Run Duration	1.1 hours
Ramp down	1 min

Concurrent users Ramp up pattern:



The transaction response times observed were as below:

Label	# Samples	Average (ms)	90% Line (ms)	Min (ms)	Max (ms)	Error %	Throughput (sec)
TR_kernel_assign_prid	51898	10887	12885	17	30030	3.53%	13.79136

Performance Test Execution Details

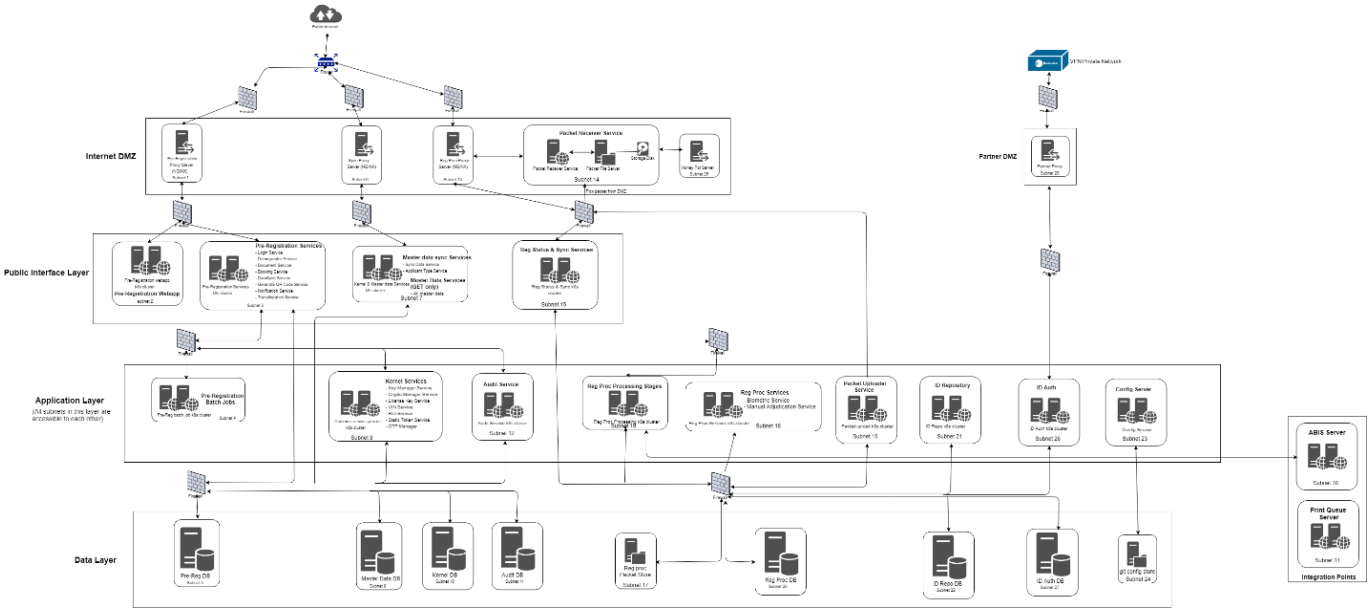
Both APIs transactions average response times were more than 3sec mentioned below:

1. TR_kernel_assign_prid -10.887 sec

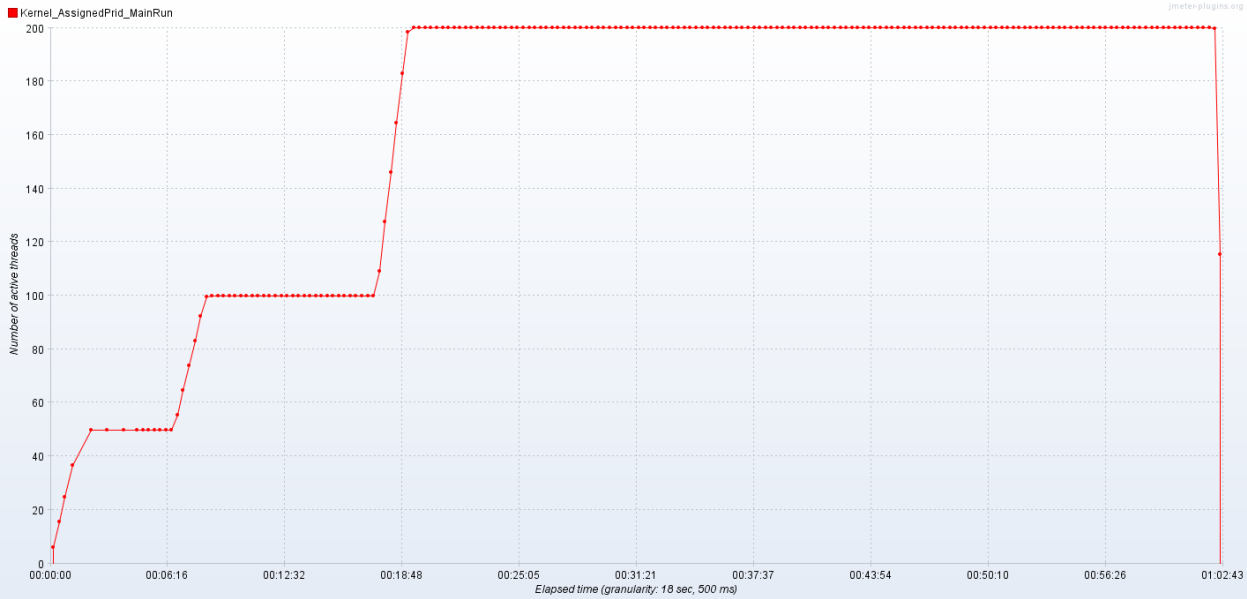
The error rate for below transactions are more than 1%:

Transactions	Error %
TR_kernel_assign_prid	3.53%

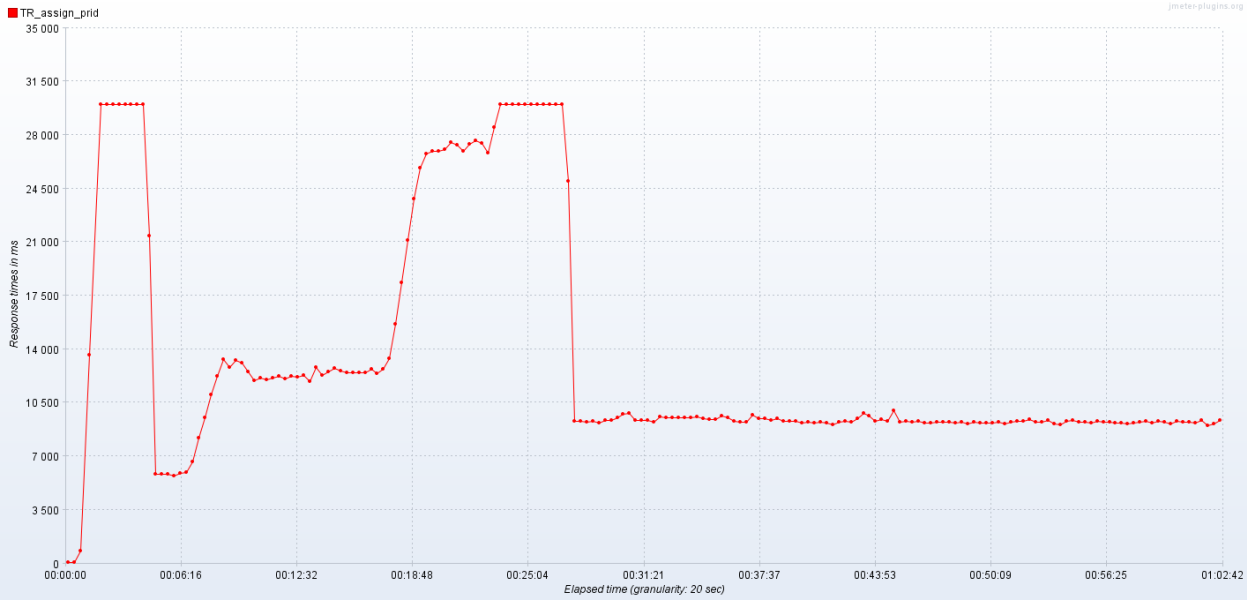
Test Environment : we are using scale-down version of below Architecture



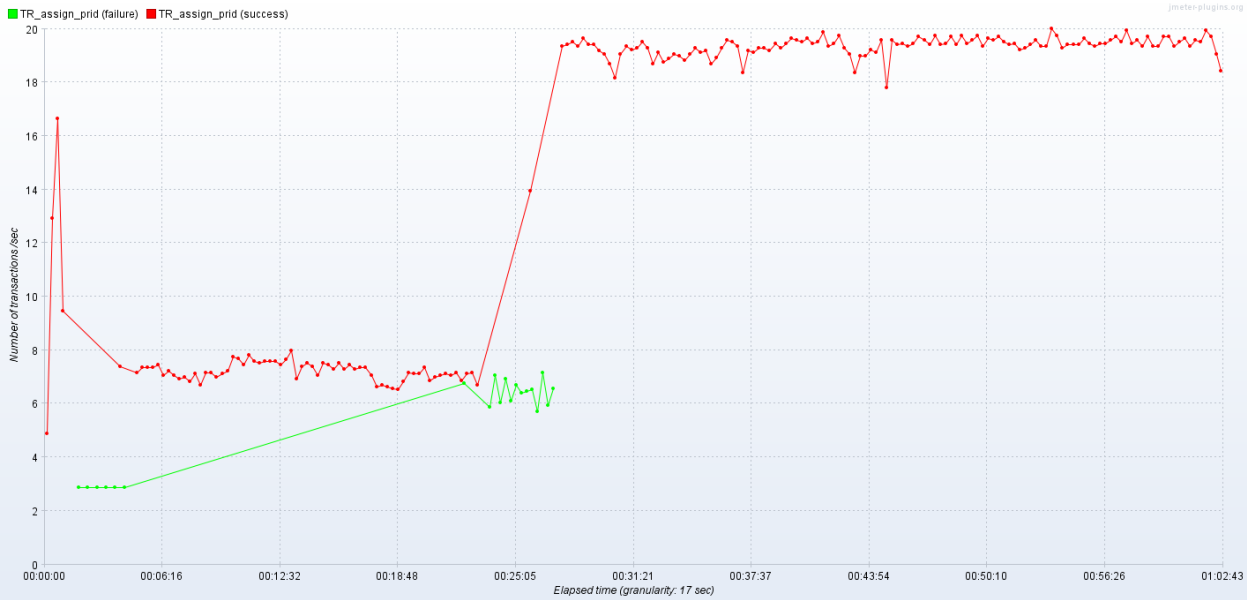
Active threads over Time:



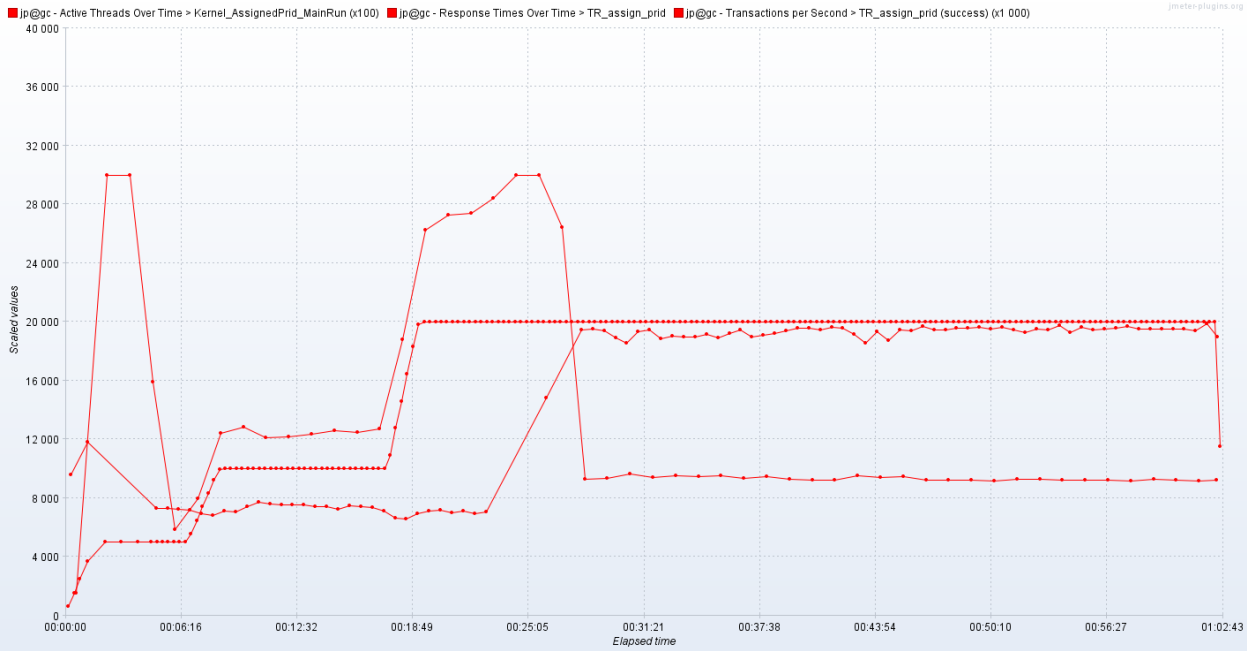
Response Time Graph



Transactions per second: (success)



Active threads vs response times over time:



JMeter graph:

Kernel cluster node 0 monitoring:

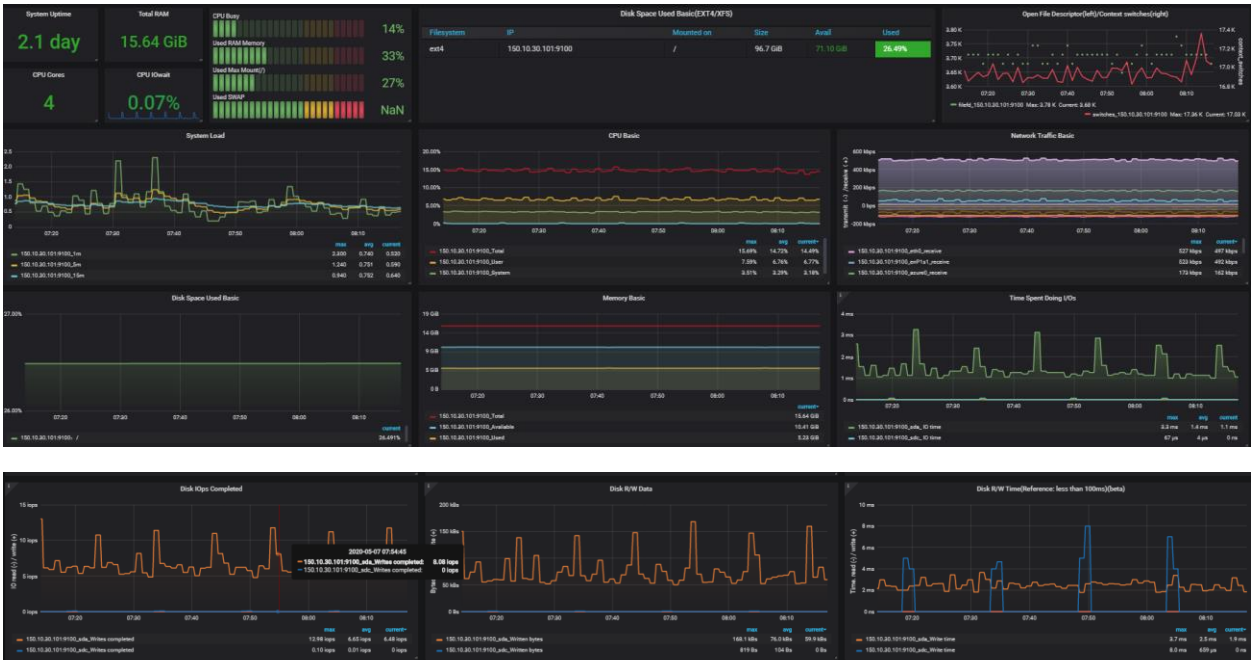
Observations:

- Max Total CPU basic is 21.94% and Avg is 08.70%
- Max used memory is 2.83 Gib out of 15.64 Gib
- Max avg system load (1m)is 0.92 and avg (1m) is 0.34

Kernel cluster node 1 monitoring:

Observations:

- Max Total CPU basic is 15.69 % and Avg is 14.72%
- Max used memory is 5.23 Gib out of 15.64 Gib
- Max avg system load (1m)is 2.30 and avg (1m) is 0.74

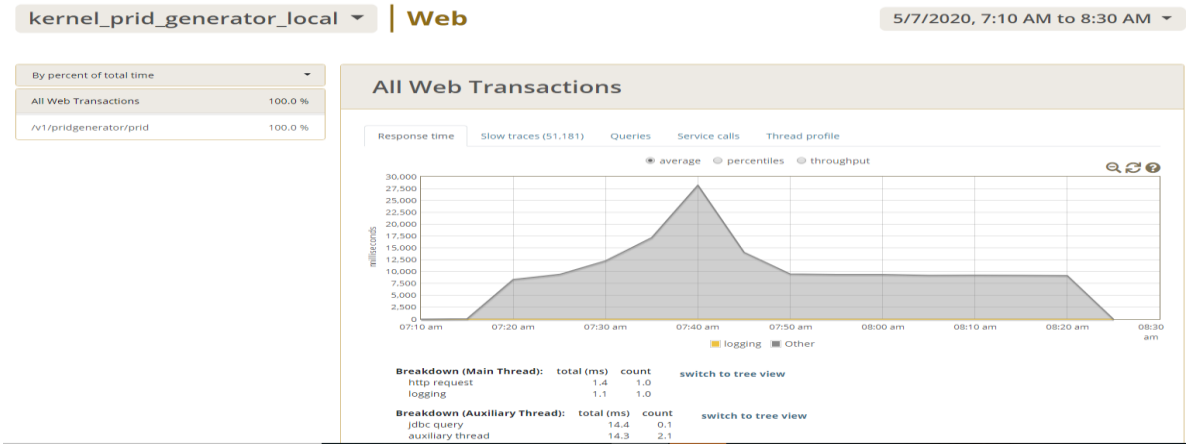


Kernel DB node monitoring: (2 core8 Gib)

Observations:

- Max Total CPU basic is 98.60% and Avg is 48.11%
- Max used memory is 2.75 Gib out of 7.78 Gib
- Max avg system load (1m)is 5.84 and avg (1m) is 1.79

Glow Root Graphs:



GET /v1/pridgenerator/prid

ASYNC
kernel_prid_generator_local
Transaction type: Web
Transaction name: /v1/pridgenerator/prid
Start: 2020-05-07 7:16:03.579 am (+00:00)
Duration: 48,578.8 milliseconds

Breakdown (Main Thread):	total (ms)	count	switch to tree view
http request	0.51	1	
logging	0.37	1	

Breakdown (Auxiliary Threads):	total (ms)	count	switch to tree view
auxiliary thread	0.081	3	

JVM Thread Stats (Main Thread)
CPU time: 0.0 milliseconds
Blocked time: 0.0 milliseconds
Waited time: 0.0 milliseconds
Allocated memory: 12.9 KB

JVM Thread Stats (Auxiliary Threads)
CPU time: 0.0 milliseconds
Blocked time: 0.0 milliseconds
Waited time: 0.0 milliseconds
Allocated memory: 888 bytes

Trace entries (1)

```
log info: i.m.k.p.router.PridFetcherRouter - publishing event to CHECKPOOL
```

Conclusion and Next Steps:

When concurrent users reached 50 we have observed high response times as well intermittent errors for kernel prid generator API calls ,Error messages are as below:

"2020-05-07 08:15:41,293 [get-prid-206] ERROR [i.m.k.p.s.i.PridServiceImpl].fetchPrid.54 : KER-PRID-001 --> PRID not available for allocation

2020-05-07 08:15:41,296 [get-prid-206] ERROR [i.m.k.p.r.PridFetcherRouter].setError.118 : KER-PRID-005 --> KER-PRID-001 --> PRID not available for allocation "

We will follow up with kernel dev team on issues -https://mosip.atlassian.net/browse/MOSIP-1191 observed today and we will rerun once fix is available