



BITS Pilani
Pilani Campus

Analyzing NFR aspect of transactions, different lock over normal Read/Write Operations in different data stores

Saurabh Araiya

2017HT12225

Motivation



- Today's industry is ruled by data, and how we handle data. There are battles over data and whole companies built around it
- We have a plethora of databases to look into these days
- Having an automated test bench would help in quick decision making

Test Bench Overview

- Exposes REST Based API's for quick execution
- Can generate the load in a concurrent manner
- Easy to plugin any data store based on use case
- Open Sourced, so readily available for use of community
- Follows industry practices like ORM, metrics, conf etc

Test Environment Details

- Primary test and dev environment was Macbook running OS X. Specifications:
 - 2.2 Ghz Intel Core i7
 - 16GB 1600MHz DDR3 Ram
 - Standard SSD Storage

Sample Request

```
Curl -I -X POST -H "content-type:application/json" -d '{"threads":5, "load": 1, "timeInMillis": 1000, "transactional" : true, "durable" : true, "requestType":"WRITE"}' 'http://localhost:1730/test/gen-load/rmq'
```

Request Param Description



- Threads: Specify Load generator thread count
- TimeInMilis: Time duration for which we are running
- Durable: Valid of durable queue in RMQ
- Transactional: This would use transactional channel in RMQ and versioned table in MySQL
- requestType: READ/WRITE

Assumptions

- Assumed very basic operation, this can be a first level flag at best. May not give complete picture
- Single server mode

Demo

Technologies Used

- Java
- Dropwizard
- Hibernate
- Yammer
- MySQL
- Elasticsearch
- RabbitMQ

Thank You