ENGINEERING BLOG

DISTRIBUTED JMETER

admin July 24, 2019 Leave a comment Edit

Jmeter is a load test tool written in java. Laptops with 16gb ram can place 150 concurrent requests but tend to slow down when the count increases. In order to simulate more number of requests, this particular setup is made.

JMETER –> INFLUXDB –> GRAFANA

Jmeter offers the capability of running servers in master-slave model. But, in this configuration, master tries to maintain the log statistics of all the slave machines. Thus when a test for thousand concurrent requests is made, master ends up maintaining thread for receiving logs from each of the slave machine.

In order to avoid this, the particular configuration is adopted where we simulate the test by running different jmeter systems. Logs from them are collected in influxdb and displayed in grafana dashboard.

```
SETUP PARTICULARS :

1. EC2 machine with influxdb natively installed and Grafana running as docker c

2. ECS cluster for running jmeter

3. S3 bucket which stores respective jmeter files

4. Jmeter configuration properties which mentions the name of the jmx file to l

5. Jenkins job to push jmx files to S3 bucket
```

```
PROCEDURE TO RUN A NEW TEST

1. Check out the code from "svn://192.168.10.7/iconcept/branches/coreInfra/Loa

2. "TestFiles" folder contains the following sub folders "Data" (Place the dat

3. Before pushing it to S3, change the respective data directory file path in

4. Jenkins has a job called SvnS3Sync which synchronises jmx files and data fr

5. "jmeter" image runs Apache Jmeter in headless mode.

6. Before spinning up the load test process, the script fetches all the files

7. All the headless jmeter servers part of the "jmeter" cluster spin up almost
So, 1000 concurrent request can be simulated by running jmeter headlessly in 1

8. All the machines send their logs to InfluxDB configured with ip as part of

9. Grafana dashboard configured with this datasource shows us the statistics f
```

Leave a comment

Comment

Post Comment