Cloud Computing Applications and Services
Benchmarking

2023

JMeter

The main goal of this guide is to better understand performance evaluation while resorting to benchmarking tools. Also, it has the goal of enriching benchmarking analysis through the observation of monitoring metrics. The following component will be used to evaluate Swap's performance:

- JMeter: `https://jmeter.apache.org`

Tasks

VMs Setup

1. Deploy the setup used in *Guide5: Monitoring*. You should use the VMs (*server1* and *server2*) from the previous class.
2. Start the ELK monitoring components (Elasticsearch, Kibana and Metricbeat) and the Swap application.
   (a) at *server1* run:
   ```
   docker start elasticsearch
   docker start kibana
   ```
   (b) at *server2* run:
   ```
   docker start swapdb
   docker start swapapp
   docker start metricbeat
   ```
3. Access Kibana's (`http://<SERVER1_IP>:80`) and Swap's (`http://<SERVER2_IP>:80`) webpages to ensure they are correctly running.

JMeter Installation

1. Install Java at your computer.
   `https://www.java.com/en/`
2. Download JMeter Binaries (Binaries - version 5.5)
   `https://dlcdn.apache.org//jmeter/binaries/apache-jmeter-5.5.tgz`

JMeter Configuration

1. Run JMeter in GUI mode (`./bin/jmeter`) at your computer.
2. Now let us create the testing workload:
   - Add a *Thread Group* to your test plan.
     At the group we will define the behavior of benchmark clients.

- Add *Constant Timer* to the *Thread Group*.
  The timer allows defining the delay between requests of a given client (*Thread Delay*).
  Set this value to 300 ms.
- Add an *HTTP Request Sampler* to the *Thread Group*.
  The sampler defines the requests being done by clients.
  Use an HTTP GET request type to the Swap login page (*Path:* /login).
  The IP address should be the one exposed by your Swap setup.
- Add *Summary Report* and *Graph Results Listeners* to the *HTTP Request Sampler*.
  These Listeners provide visual representations for the results being observed.
  Note: Listener results can be exported to file.

Swap Testing

1. At the *Thread Group* set the number of threads to 1 and the loop count to infinite (*i.e.,* 1 benchmark client doing requests infinitely).
   Run the experiment and observe Kibana (Dashboard: *[Metricbeat System] Host overview ECS*) and JMeter's *Summary Report* and *Graph Results Listeners* to obtain different metrics.
2. Stop the test and change the number of threads to 10.
   Re-run the experiment and observe the results at JMeter and Kibana.
3. Stop the test and change the number of threads to 100.
   Re-run the experiment and observe the results at JMeter and Kibana.
4. Export the template to file and run the experiment with the Non GUI mode.
   `./bin/jmeter -n -t  [jmx file] -l [results file]`

   Important: always run experiments for collecting measurements with the Non GUI mode!

Extra

1. Explore JMeter browser recording capabilities
   `https://jmeter.apache.org/usermanual/jmeter_proxy_step_by_step.html`

Learning Outcomes

Recognize the goals and considerations that must be taken when doing a performance evaluation of real systems. Apply the JMeter benchmark to evaluate a web-based service. Complement benchmarking analysis with monitoring.