Multi-tenancy version of BigDataBench

Gang Lu

Institute of Computing Technology, Chinese Academy of Sciences

BigDataBench Tutorial
MICRO 2014 Cambridge, UK
Multi-tenancy software

- **Software perspective**
  - Multi-tenancy refers to a principle in software architecture where a single instance of the software runs on a server serving multiple client-organizations (tenants).
  - With a multi-tenancy architecture, a software application is designed to *virtually partition its data and configuration*, and each client organization works with a *customized* virtual application.
Multi-tenancy infrastructure

Application owners

Tenant

Tenant

Tenant

Tenant

Internet

Infrastructure resources

General characteristics:
1. Resource pooling and broad network access
2. On-demand and elastic resource provision
3. Metered resources
Multi-tenancy workloads

Application owners

Tenant

Tenant

Tenant

Tenant

Internet

Infrastructure resources

Cloud infrastructures

Big Data Workloads

Hadoop

MySQL

Apache HBase

MPI

Spark

Spark Streaming

Storm

MongoDB

Hive

 tenants workloads

infrastructure resources

Internet

application owners

cloud infrastructures
Problem of single-tenancy benchmarks

- Focus on a single run of workload
- Scenarios are not realistic (simple and synthetic)!
  - Does not match the typical operating conditions of real systems, in which mixes of different percentages of tenants and workloads share the same computing infrastructure
- We need
  - Emulate real-world datacenter cluster with different amounts of tenants and various workload types and consequently various benchmarking scenarios.
Multi-tenancy version of BigDataBench

Mining real-world Workload traces (Google, Facebook, Sogou)

Profiling workloads from BigDataBench

Workload matching using Machine learning techniques

Parametric workload generation tool

Benchmarking scenarios

Mixed workloads in public clouds

Data analytical workloads in private clouds
Example: The first two steps

- An example of match Hadoop workloads

Mining Facebook workload trace (Exact resource usage information: CPU, memory, I/O)

Profiling Hadoop workloads from BigDataBench (Also collect resource usage information)

Workload matching using k-means clustering

Matching result: replaying basis

<table>
<thead>
<tr>
<th>Job type</th>
<th>Input size (GB)</th>
<th>Starting Time (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bayes</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>Sort</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>K-means</td>
<td>0.5</td>
<td>25</td>
</tr>
<tr>
<td>Bayes</td>
<td>5</td>
<td>30</td>
</tr>
<tr>
<td>Sort</td>
<td>1</td>
<td>40</td>
</tr>
</tbody>
</table>
Example: The last two steps

- An example of mixing search engine and Hadoop workloads

1. Hadoop jobs with matching replaying basis
2. Nutch searching with matching replaying basis
3. Decide how many tenants to emulate and their workload types
4. Parametric workload generation tool
5. Benchmarking scenarios
   - Mixed workloads in public clouds
What can you do with it?

- We consider two dimensions of the benchmarking scenarios
  - From tenants’ perspectives
  - From workloads’ perspectives
You can specify the tenants

- The number of tenants
  - **Scalability Benchmark:** How many tenants are able run in parallel?

- The priorities of tenants
  - **Fairness Benchmark:** How fair is the system, i.e., are the available resources equally available to all tenants? If tenants have different priorities?

- Time line
  - How the number and priorities of tenants change over time?
You can specify the workloads

- **Data characteristics**
  - Data type, source
  - Input/output data volumes, distributions

- **Computation semantics**
  - Source code
  - Big data software stacks

- **Job arrival patterns**
  - Arrival rate
  - Arrival sequence
You can specify the interference

- Each individual tenant:
  - Different types of workloads
  - How they interference each other at different resource dimensionalities?

- Multiple tenants:
  - How well are tenants isolated from one another with respect to performance?
  - How do individual tenants influence other tenants' performance?
Current status

- Multi-tenancy V1.0 releases:
  - Emulate workloads based on real-world workload traces
  - Support mixes of both online service and offline batch workloads

<table>
<thead>
<tr>
<th>Workloads</th>
<th>Software stack</th>
<th>Workload trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hadoop</td>
<td>Hadoop 1.0.2</td>
<td>Facebook (<a href="https://github.com/SWIMProjectUCB/">https://github.com/SWIMProjectUCB/</a> SWIM/wiki)</td>
</tr>
<tr>
<td>Shark</td>
<td>Shark 0.8.0</td>
<td>Google data center (<a href="https://code.google.com/p/">https://code.google.com/p/</a> googleclusteringdata/)</td>
</tr>
</tbody>
</table>
Any Questions