Functional Workload Model

Francois Raab, InfoSizing,
Yanpei Chen, Cloudera
# Successful Benchmarks

By number of published results (approx.)

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Results</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPECcpu (2006 + 2000 + 95)</td>
<td>10,800</td>
<td>Component</td>
</tr>
<tr>
<td>SPECjbb (2005 + 2000)</td>
<td>1,050</td>
<td>End-to-End</td>
</tr>
<tr>
<td>TPC-C</td>
<td>760</td>
<td>End-to-End</td>
</tr>
<tr>
<td>SPEC SFS</td>
<td>730</td>
<td>End-to-End</td>
</tr>
<tr>
<td>SPECweb (2009 + 2005 + 99 + 96)</td>
<td>700</td>
<td>End-to-End</td>
</tr>
<tr>
<td>TPC-D/H</td>
<td>650</td>
<td>End-to-End</td>
</tr>
</tbody>
</table>
End-to-End Definition Flow

- Identify use cases (real-life applications)
- Group use cases by Application Domain
- Select an “interesting” application domain
- Build Functional Workload Model
  - Refine into Functions of Abstraction
  - Define Schema, Population, Scheduling
- Complete benchmark definition
  - Execution rules (do’s and don’ts)
  - Constraints (must have’s)
  - Metrics (absolute, composite)
Benchmark Pyramid

Application Domain
- Use-Case X
- Use-Case Y
- Use-Case Z

Functional Workload Model
- Functions of Abstraction
- Schema
- Population
- Scheduling

Systems (Logical) View
- Web Server
- TP Monitor
- App Server
- MapReduce
- Relational/SQL
- NoSQL DB

Physical View
- SMP Server
- Blades
- Cluster
- Cloud
- CPU
- Network
- Memory
- Storage

End-to-End Benchmark
Component Benchmark
Proposal: Use Cases Library

• Inventory of specific use cases (deployment)
  • Summary description of context
  • Workload characteristics
    • Data set (4 V’s)
    • Basic operations
    • Execution environment
  • Classification in Application Domains
    • Commonalities
    • Popularity
    • Adoption rate