Component vs. End-to-End Benchmarks

Meikel Poess
Oracle Corporation
Typical Elements of Today’s Industry Standard Benchmarks

• **Scope**
  – Who can play in the benchmark

• **Workload**
  – Data
  – Operations on data

• **Execution Rules**
  – What operations to run when and on what data

• **Measurement Methodology**
  – What and How to report measurements

• **Metric(s)**
  – How to combine measurements into one or multiple numbers
  – Where to publish the metric and how to compare with others

• **Referee**
  – How to guarantee correctness of measurements (a.k.a. AUDIT)
Component Benchmarks

• Limited in their scope
  – Measure performance of small number of system components (hardware or software)
  – Extreme case, only one system component
    • CPU, disk, Graphic Card
  – Example: SPEC CPU2006 benchmark
    • Benchmark to test performance of a system's processor, memory subsystem and compiler
    • Benchmark publication 6 pages
    • Single system, no disk (except for OS disk)

Component Benchmarks

• Easier to develop than an End-to-End benchmark
  – Scope is limited
• Easier to implement for benchmark sponsors than End-to-End benchmark
  – For components that expose standardized API’s a benchmark can be developed to run as-is
• Easier to run
• Less expensive to run
• Does not express a customer environment
• Does not expose interaction problems of components
End-to-End Benchmark

• Broad in scope:
  – Measure performance of entire systems
  – Extreme case, entire lifecycle of a system
    • E.g. TPC-C benchmark
    • Benchmark publication 741 pages (http://c970058.r58.cf2.rackcdn.com/fdr/tpcc/Bull-Escala-PL6460-080615-FDR.pdf)
End-to-End Benchmark

• More difficult to develop than component benchmark
  – Wide scope, which might include many components and vendors

• More difficult to implement than component benchmark
  – Benchmark tends to be specification driven, i.e. vendors need to develop a Benchmark KIT
  – Expresses real customer environment
  – Exposes component interaction problems
Specification vs. KIT based

**Specification based**
- Written in English prosa
- Needs to cover all possible cases
- Leaves room for interpretation
- Requires vendors to built Benchmark KIT, which is an interpretation of the spec
- Interpretation leads to problems

**KIT based**
- Written to be executed as-is
- Very specific for an limited number of systems
- Leaves no room for interpretation
- Vendors can run Benchmarks without further investments
Questions to the Community

• Shall we develop a Big Data Benchmark as a Component Benchmark or as an End-to-End benchmark?

• Is it possible to develop a benchmark as a set of component benchmarks that could be combined into an end-to-end benchmark?

• Shall we strictly adhere to a KIT based benchmark or a specification based benchmark?