Benchmarking Abstractions
-or-
The Problem of Defining the Problem

Len Wyatt
Principal Program Manager
Microsoft
Benchmarks for Consumers

Big Data Customers → Implementers → Technically sophisticated
Find component benchmarks useful

Consumers → Seeking off-the-shelf solutions
Find “end-to-end” benchmarks useful

Benchmark must be relevant → Different abstraction?
Super-tuned?
Super hardware?
Not relevant

Jim Gray’s criteria for a useful benchmark:
relevance, portability, scalability and simplicity

Abstraction and Relevance

• Sample benchmark problem:
  – Create an aggregate over large data set
  – A simple, scalable problem
• If using HDFS, most efficient to use M/R job
• Consumer is likely to use HiveQL for that
  – Customer favors time to solution over efficiency
  – May not see an M/R implementation as relevant
• Flexibility of implementation vs. relevance
  – The more general abstraction sacrifices relevance
  – Doesn’t capture time to solution
Abstraction and Portability

• Solve the relevancy problem by defining the benchmark using the user’s level of abstraction
  – HiveQL is a good language for doing an aggregation
  – Define the benchmark in terms of HiveQL

• Not portable!
  – Pig Latin is a good language for doing an aggregation
  – Cannot run the benchmark using Pig Latin

• The more specific definition makes the benchmark less portable
  – In the rapidly evolving Big Data ecosystem, more specific benchmarks will have a limited lifespan