



In-Production Benchmarks for Distributed Large-Scale Data Processing

Jerry Zhao, Google Inc.

WBDB2012

One Challenge in Big Data Processing



- To understand how **sensitive** BDP applications are to the **dynamics** in the **shared** infrastructure.
 - Shared infrastructure
 - Resource isolation v.s. oversubscription
 - Evolving system components
 - Dynamic execution environment
 - Variability due to budget and priority
 - Variability due to distributed nature of BDP
 - Diversified applications
 - Different performance requirements
 - Root causes hidden by BDP frameworks

One Proposal for BDP Benchmarking



- To "Embed" benchmarks in production jobs
 - Configurable to explore parameter space
 - also to quantify the running environment
 - Executed probabilistically in production
 - to expose correlation with specific user/app
 - Triggered to target interesting scenarios
 - by the hints from internal states of DP framework
 - Monitored and analyzed over time
 - to capture (unforeseen) infrastructure changes
- Complemented by other strategies

Example: Simple Sort MicroBenchmark



- Sorting xGB input k/v pairs, writing the result to GFS.
 - #threads, buf sizes for different stages
 - encoding/compression option used
 - job priority, allocated CPU/IO budgets
- Built within application binaries
 - the same compiler optimization
- Support required from infrastructure
 - local load on CPU/DiskIO/NetIO
- Support required from MapReduce
 - Schedule/execute BM on the same machine when slow tasks detected
- Data collectively analyzed along all dimensions
 - Users, clusters, or platforms