Leo Giakoumakis, Microsoft SQL Server

Grand Challenges in Testing Data-intensive Computing Systems

DBTest Workshop 2010

Terminology

- *Testing* is:
  - Ensuring that the system is built as designed
  - Ensuring customer requirements are met
  - Finding bugs
  - Quantifying, tuning and deciding engineering tradeoffs
  - Providing *insurance*
  - Providing information: learning

- *Data management systems:*
  - Broad term: includes any of SQL Server, Stream Insight, SAP, Big Table, Hadoop, etc.

Data management & Testing

- Data management field
  - More than 30 years of research and development
  - Built on strong foundations
  - Enormous body of published work by academia and industry
  - Today anyone can build a RDBMS, *the "text-book" has been written!*
    - MySQL, Postgress, and other code bases in the academia are available
- The field of testing data management systems
  - Test engineering is an immature discipline
  - Very little information is shared
  - The state of the art is yet to be defined
- *It's time to start writing the text-book on testing data management systems!*

What makes testing challenging

- Size and Cost
  - Testing at realistic scale and size is hard
  - Testing the size and richness of the programming surface
  - The cost of testing an increment of the feature-set is often a function of the entire feature-set
  - Test code bases are becoming large, hard to manage
  - Test *immortal test case* problem!

- Understanding coverage
  - Knowing when you have done enough testing is hard
  - Code coverage is not enough; you need state, workload, scenario coverage

- Complexity
  - Testing is multidisciplinary: language compiler, optimization, operating system, etc.
  - Complexity increases with appliances, distributed systems, the Cloud

Areas in need of solutions 1/2

1) Workloads/benchmarks
   - Standard benchmarks are: performance oriented, simple, linear
   - We need large, mixed workloads with un-steady states and built-in failures
   - Methodology and metrics for workload characterization
   - Important: this is what most researchers rely on to evaluate their ideas

3) Test architecture and test reuse
   - What are the abstract primitives that would apply to testing most systems?
     - A standard blueprint for testing data management systems?
     - A standard suite of test methodologies?
   - The industry is moving towards "multiple engines"

4) Large data
   - Data expansion and data reduction

Areas in need of solutions 2/2

1) Query optimization quality
   - Defining metrics for QO quality is hard
   - Also, measuring quality over time and over code changes
   - How can you compare optimizers of two competitive systems?
   - Proving query results correctness for complex queries over large data

2) The Cloud
   - Distributed data processing/storage at a large scale
   - Failure: from unlikely to certain
   - Failure conditions: harder to reproduce
   - Testing needs to takes place “in production” too
   - SLA testing

Thank you!