Reverse Engineering Query Execution Engine Design Decisions

Helga Gudmundsdottir, Parmita Mehta, Magdalena Balazinska, and Dan R. K. Ports

-- Work in Progress --
Same analytics query on two different systems
Different analytics query on same two systems

Query

Spark

Myria
What are these systems really doing?

- Writing intermediate data to disk? Vs in-memory?
- CPU-intensive operations?
- Shuffling data?
- Inefficient memory management?
- Any synchronization barriers? Data skew?
- Able to use entire cluster?
- Were there failures?
- ...

...
Want to understand the underlying design decisions of query execution engines

Requires:
- Deep expertise in each DBMS
- Detailed experimentation
- Knowledge of code base
- Instrumentation & profiling
Is there a way to do this automatically?
80% workers utilized
Three data shuffling ops
No synch. barriers
Periodic data materialization
Column-store storage

Do this automatically, for any engine
How do design decisions manifest in low-level system metrics?

- Can we learn their signatures?

Can system metrics provide insights into how design decisions manifest during query execution?

- Provide rich information about query execution and system behavior
Automatically distill low-level metrics into design decisions

- 80% workers utilized
- Three data shuffling ops
- No synch. barriers
- Periodic data materialization
- Column-store storage

Do this automatically, for any engine
Initial Approach

Sysstat gathers system metrics

Learn signatures of basic operations for each system

Simple queries: Select, Join, Aggregate

Use TPC-H benchmark data at multiple scale factors
Can we distinguish between systems?
Can we distinguish between basic operators?

JOIN followed by a PROJECT
Questions to the audience

- Would you use such a system?
- What design decisions are worth inferring?
- Other applications of this approach?
SLA for Multi-tenant Analytics as a Service
Analytics as a Service

Dedicated resources can provide performance guarantees

Over-provisioning can provide predictable response times

But, what if we want analytics cheaper?
SLA for Multi-tenant Analytics Service

Can the information about system load be captured and displayed to customers utilizing the system?

How to expose contention that is interpretable by the user?
- Slow down factor for the service?
- How long before query will complete?
SLA for Multi-tenant Analytics Service

What does the user want?

• How should we think about SLAs in Analytics as a service with contention?

• How should we expose this information to end users?

• What scheduling algorithms make sense for Analytics as a service?