



ACM SIGMOD 2010 Programming Contest Distributed Query Engine

Task Description & Preliminary Results

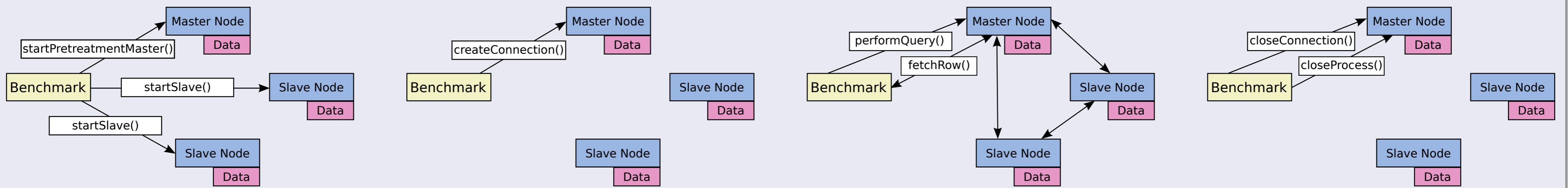
Clément Genzmer, Pierre Senellart



Data distributed over 8 identical nodes, simple SQL queries, the **fastest** distributed query engine wins!

API to implement

Initialization phase → Connection phase → Query phase → Closing phase



Workloads

- 8 distinct (**secret**) workloads
- Selection on value equality, range queries, projections, natural joins
- Up to 150,000 queries per workload
- Up to 1,000,000 tuples per node
- 5 to 10 min allocated per workload

Participants:

- 29 teams
- 23 different institutions
- 13 different countries

5 Finalists: (in lexicographic order)

- bugboys KAUST
- cardinality Stanford U.
- dbis TU Kaiserslautern
- insa INSA Lyon
- spbu Saint-Petersburg U.

Implementation of a distributed query engine over relational data

Aurei Teodorescu, Brice Arnould, Vlad Georgescu, Horia Iancu

National Institute of Applied Sciences, Lyon, France, Department of Computer Science

Our implementation is based on a combination of primitives:

- Select elements based on a primary key or an indexed field
- Get/Read one file, line by line
- Insert lines into a hashmap

2 types of optimizations

	High level	Low Level
Reorder tables	Change the master node	<p>2X: Transform table names and fields into indexes</p> <p>Up to 6X: Use two one-way sockets instead of one two-way socket</p>
Optimize memory consumption and reduce CPU usage	Avoid communication between nodes	
Compute once	Preprocess table information	<p>2X: Pre-hash string values to avoid string comparisons</p>
Memorize join results for later use	Compute min/max values of tables to avoid duplicate computations	
Testing techniques		
Test generation	Code profiling	Network communication monitoring

Supported by

