criteo

First Impressions of the Presto CBO

Håkan Jakobsson
Criteo

What is Criteo?

- Ad-tech company headquartered in Paris, France
  - Offices around the world
  - ~2,800 employees
  - ~$2.3 billion in revenue
- Demand Side Platform for digital marketing
  - Bids on ad space on behalf of advertisers
  - Heavy use of ML in bidding process
- Heavy use of Hadoop and other open-source technologies
  - Clusters with ~300 PB disk for HDFS
  - Some tables ingest around 10 billion rows per hour
  - Hive, Spark, Kafka, Flink, MySQL, Druid, etc., as well has home-grown technologies
  - Also, commercial products like Vertica and Tableau
- Recently introduced Presto for fast SQL access to data on HDFS
  - Currently ~200 nodes (16 physical cores, 256 GB memory per node)
  - Waiting for another 300 nodes
  - First use case: interactive queries by analysts

Behind Criteo's Interest in the CBO

- Presto is proving highly useful; need to understand all the features
- Expanding Presto use cases to areas where the CBO may matter more
  - Not just used by sophisticated early adopters
- Cost-based optimization has numerous advantages (and some trade-offs)
- Personal interest
  - Experience with optimizer development at Oracle
  - Including when Oracle's CBO was an immature feature

Why Cost-Based Optimization (General Benefits)

- There are often multiple possible execution plans for a query
- What typically determines which one will be used without a CBO?
  - The syntax of query (e.g., order of the tables in query)
  - Parameter settings (e.g., join_distribution_type)
- What typically determines which one will be used with a CBO?
  - Estimates of the costs of the various alternative execution plans
- Who makes the determination without a CBO?
  - End user who types in SQL (possibly an Analyst)
  - A developer of an application containing canned SQL statements
  - A query tool that generates SQL
- Who makes the determination with a CBO?
  - The database engine (but typically requires statistics management)

Basis for CBO Cost Estimates

- Table statistics
  - Number of rows, files, size of data, etc.
- Column statistics
  - Min and max value, number of distinct values, number of nulls, etc.
- Cost model
- Optimization goal (may be implicit in the cost model)
  - What to optimize: resource utilization, response time, etc.
- Possibly, some parameter settings
- Cost estimates can potentially be used by a proactive Resource Manager

Potential CBO trade-offs

- No query optimizer is going to be 100 perfect
- But something is better than nothing
- Cost-based query optimization requires some effort
  - Statistics management, for instance
- But a CBO might be the difference between a query working or failing
- Key point: Have the right expectations

So What About Presto?

- Criteo's use case for Presto is very much with aligned with its design criteria
- A replacement for Hive for SQL queries accessing data on HDFS clusters
- First target: interactive queries from business analysts
  - ETL jobs may follow
- Pretty much universally liked
  - Though one guy in NYC complained about queries being too fast (no time for ping-pong)

Motivating Example

```
SELECT * FROM t1, t2
WHERE t1.c1=t2.c2
UNION ALL
SELECT * FROM t3, t4
WHERE t3.c3=t4.c4
```

- What about distribution methods?
  - The two joins may benefit from using different distribution methods
  - Deciding the distribution method based on a parameter setting may not be optimal

Presto Hash Joins

- May be the primary use case for Presto as SQL on Hadoop
- Hash joins have a number of fundamental rules of thumb
  - Like “it’s usually better to build a hash table on a smaller data set”
- Two major ingredients for low-hanging fruit
  - Join order: Build side versus probe side
  - Distribution: Do you broadcast a dimension table to all the nodes in a cluster?
- The Presto CBO can make informed decisions on these issues

The CBO -- How Good?

- First of all, everything is workload dependent
- Bears repeating: everything is workload dependent
- Criteo used a variety of queries in testing the CBO
  - Standard benchmarks, internal queries, atomics, etc.
- Some queries will have no benefits
  - Queries without joins won't benefit much from join order optimizations
  - Some queries could deteriorate (most likely a very small percentage)
- Straightforward relational queries can have significant benefits
  - A factor of 2-3 performance benefit could have significant economical implications
  - Some queries could see order-of-magnitude improvement (but likely not the norm)
  - Nested data is a major issue for relational query optimization
  - CROSS JOIN UNNEST

CBO Limitations

- Statistics management
  - Statistics need to be current
  - Dependent on connectors
  - Nested data an issue
- Functions
  - WHERE f(col) = 5
- Correlation
  - WHERE make = ‘FORD’ and model = ‘MUSTANG’

Potential Enhancements

- More connectors
- Hints
- Statistics management
- Plan stability
- Dynamic sampling

Q & A

Questions?