- Assignment 2 grading in progress
- Midterm review session on Nov 2 at 5pm at 1221 CS
- Course Project Proposal feedback
WIDE AREA ANALYTICS
- Support analytics queries (including joins)
- Minimize wide-area network usage
- Resources within single DC are plentiful
- Primary metric: Bandwidth cost not latency
APPROACH

1. Join order selection
   - Choice of join algorithm
   - Order in which they are executed

2. Task assignment

3. Manage data replication
ARCHITECTURE

Queries → Geode Command Layer → Optiq++, ILP → Workload Optimizer → Suggestions

Measurements

Results

Proxy layer

Hive

End-user facing DB
SUB QUERY DELTAS

Cache intermediate results in sub-queries

What does this help?
- Repeated queries (issued every hour etc.)
- Shared sub-queries (across data-scientists ?)

What does this not help with?
- Computation still happens within DC
- Extra storage for cache (how do you expire this ?)
QUERY OPTIMIZER: CALCITE++

Apache Calcite: centralize SQL query planner
Input: SQL parse tree. Output: Optimized parse tree
Similar to Catalyst, but includes Cost-based optimization

Calcite++
Estimate distributed join cost
Use rough statistics based on table size etc.
Important to pick right plan not estimate accurate cost!
PSEUDO DISTRIBUTED EXECUTION

Original

Pseudo Distributed
PSEUDO DISTRIBUTED EXECUTION

Key idea: Use stats from repeated executions

Advantages
- Precise estimation
- General across operators

Disadvantages?
- Latency overhead
- Huge space of executions
SITE SELECTION, DATA REPLICATION

Integer linear program formulation

Objective: Minimize replicationCost + executionCost

Constraints
- Disaster recovery
- Regulatory constraints

Solution
- Assignment of which task runs on which DC
- Which partition is replicated to which DC
ILP doesn’t scale for large workloads

Greedy heuristic
  Greedily pick datacenter for task based on copying cost
  Plugin values, run ILP for replication strategy

Limitations
  No consideration of inter-query caching
  Requires knowing “core workload” ahead of time
EXTENSIONS

Can layer additional optimizations on top
Further reduce amount of data transmitted

Examples
- Top-k queries 9x reduction in big-data benchmark
- Approximate percentile 170x reduction for TPC-CH
SUMMARY

New area of wide-area big data analytics

Combine query optimization + network awareness

Main contributions
  - Optimize data replication, task placement
  - Intelligent caching to reuse sub-queries