ADMINISTRIVIA

- Assignment 2 grades
- Midterm coming up Tuesday!
- AEFIS feedback form
SQL IN BIG DATA SYSTEMS

- Scale: How do we handle large datasets, clusters?

- Wide-area: How do we handle queries across datacenters?
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
SELECT pageURL, ...
FROM PagInfo
WHERE pageCategory = 'Entertainment'

Q: SELECT sourceIP, sum(adRevenue), avg(pageRank)
   FROM ClickLog cl JOIN PagInfo pi ON cl.destURL = pi.pageURL
   WHERE pi.pageCategory = 'Entertainment'
   GROUP BY sourceIP
   HAVING sum(adRevenue) >= 100

Sizes of the table
Join How do we do it
Broadcast
Filter before Broadcast
Do a local join
Aggregation, return results
APPRAOCH

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

Results

Measurements

Proxy layer
  - Hive
  - End-user facing DB
OPTIMIZER SETUP

Datacenters / topology
slowly evolving

Workload properties
Produced by other process

Data birth

Sovereignty
Restriction on data movement

Fixed Queries
Does data change?
Small number of queries
Queries evolve very slowly

Diagram:
- Optiq++ produces Annotated DAGs
- DAGs undergo site selection and data replication
- Sovereignty, fault-tolerance, and scheduled DAGs
- Replication policy

Optiq++
DAG₁
Annotated
DAG₁

Optiq++
DAGₖ
Annotated
DAGₖ

site selection + data replication

scheduled DAGs

q₁
qₖ

pseudo-distrib measurement

pseudo-distrib measurement
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 ?)
Apache Calcite: centralized SQL query planner
Input: SQL parse tree. Output: Optimized parse tree
Similar to Catalyst, but includes cost-based optimization

Calcite++
Estimate distributed join cost
Important to pick right plan not estimate accurate cost!
Select join strategy e.g. Broadcast
PSEUDO DISTRIBUTED EXECUTION

Original

SELECT pageURL, ...
FROM PagelInfo
WHERE pageCategory = 'Entertainment'

Pseudo Distributed

SELECT sourceIP, ...
FROM ClickLog cl
JOIN PagelInfo pi ON ...

SELECT sourceIP, ...
FROM ClickLog cl
JOIN PagelInfo pi ON ...
HAVING sum(adRevenue) > 100

Actually running in 1 DC

Simulating effect of multiple DC
PSEUDO DISTRIBUTED EXECUTION

Key idea: Use stats from repeated executions

Advantages
- Precise estimation
- General across operators

Disadvantages?
- Overhead of fetching
- Latency fluctuations in network, data
- Space of executions is large
SITE SELECTION, DATA REPLICATION

Integer linear program formulation

Objective: Minimize $\text{replicationCost} + \text{executionCost}$

Constraints
- Disaster recovery
- Regulatory constraints

Solution
- Assignment of which task runs on which DC
- Which partition is replicated to which DC
SITE SELECTION, DATA REPLICATION

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

- Solutions may not be optimal
- Fixed workload

Cost, model, caching etc.
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
DISCUSSION

https://forms.gle/Qr142WN1LVNyVAfLA
If the orders table was distributed across three geographic locations: US, Europe and Asia, how can the query can be executed by using Geode.

```
SELECT order.id, item.name, item.price, order.count
FROM item
JOIN order
WHERE item.id = order.itemid and item.price < 1400 and order.count > 2 - 1
```

If the orders table was distributed across three geographic locations: US, Europe and Asia, how can the query can be executed by using Geode.

- Depends on sizes of tables.
- Try filter first on item price, order count
- Do either centralized join or broadcast items & join.