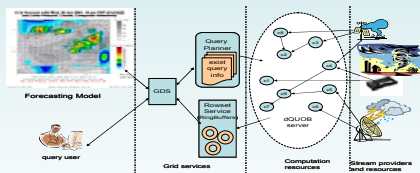


Distributed Query Planner in Calder System

Ying Liu, Beth Plale, Nithya Vijayakumar
Computer Science Department, Indiana University

Problem and solution



Problem

- Motivating application is Calder which is shown in above figure
- Stream resources highly **distributed** and asynchronous
- Scientific application needs streams in **real-time**
- Cost metrics in general incapable of distinguishing different plans in this domain.
- Hundreds **existing queries** become reusable resources for system. New techniques to utilize these resources are needed.

Solution

- Extending cost metric space
- Efficient query distribution and assignment algorithm
- Inventing novel query re-using techniques for new application.

Re-Usable query

Types of re-usable queries and the re-using techniques for each.

Identical Query

- Hash query's normalized text string to search the identical query
- Statistic data shows more than 60% queries are generated by machines.

Parameterized Query

- Each query group has a query model.
- During the execution, we use parameter table to match the different queries.

Parameterized Query Model	
SELECT	e.Name
FROM	employee e
WHERE	e.age=?

Contained Query

- For query pair (A,B), we define $B \Rightarrow A$ if $Result(B) \subseteq Result(A)$
- Redirect Query A's output as Query B's input.

QueryA	QueryB
SELECT e.Name FROM employee e WHERE e.age>30	SELECT e.Name FROM employee e WHERE e.age>40

Structure-sharable Query

- Query structure is defined as the query tree just with the operation node.
- We use graph-based tree matching algorithm to locate re-usable queries.

Extended cost metric space

Rate-based cost model

- Computation cost is computed based on the input stream's input rate instead of the cardinality
- Minimum computation cost is the goal to choose the query plan

Extended cost metrics

- Query deployment cost
- Query compiling, optimization and set-up time
- Query re-using cost
- Re-usable query searching and tuning time
- Query execution cost
- Network transmission cost

Cost estimation

- Use statistic data to estimate the unit processing cost for different objects

Structure-sharable query

Motivation

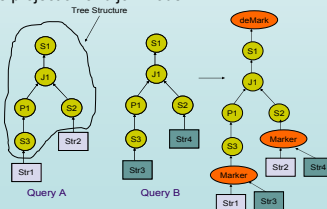
- Hundreds of sensors generate streams in the **same format**
- Streams are highly asynchronous
- Query Structure: query tree just with operation nodes

Optimal Compatible Sub-Plan Set

- Based on tree-matching algorithm
- Find optimal coverage with the existing query set

Re-usable query tuning

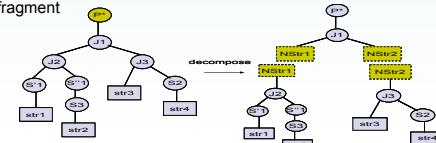
- Reserved attribute: queryID
- Add "Marker" and "deMark" node
- Tune projection and join node



Distributed query plan generation

Query decomposition

- Optimized central query can be decomposed and assigned to distributed processing site for better performance
- Heuristic-based greedy algorithm for query decomposition
- Use heuristics to optimize the central plan
- Depth-first traversal for query decomposition
- Greedy put two input parts of a join operator within the same fragment



Query fragments assignment

- Fragment assignment is based on re-usable query availability, computation site workload, network bandwidth and input stream availability
- Finding re-usable query has priority
- Stream availability is related with bandwidth cost

$$score = (1-d) * w_j + \frac{b}{max_b} * w_b$$

Summary

Conclusion and Contribution

- Data-driven scientific application brings new challenges to traditional query cost model
- Novel metric space considers deployment cost and network bandwidth cost
- Multi-query optimization techniques are evaluated in new application domain
- Novel structure-sharable query and its searching, tuning algorithm

Future Work

- Work on more effective greedy-based query distribution algorithm
- Experimentally analyze how reuse threshold varies under different working scenarios.



Distributed
Data Everywhere
(DDE) Lab
Indiana University