MOIRA

A Goal-Oriented Incremental Machine Learning Approach to Dynamic Resource Cost Estimation in Distributed Stream Processing Systems

Daniele Foroni, C. Axenie, S. Bortoli, M. Al Hajj Hassan, R. Acker, R. Tudoran, G. Brasche, Y. Velegrakis
Context

Spark Streaming

Flink

Presentation board
Context

# Events

Time
Context

# Events

Time

Daniele Foroni - BIRTE 2018
Only 1 Fixed Parameter and NOT query specific!
Why should we consider the goal?
MOIRA

- Goal-centric approach
- Goal specified for a single query
- Rescheduling for improving the goal performance
Query Plan

Source 1
- Map
- Filter
- Join
- Source 2
- Filter
- Join
- Sink
Query Plan Optimization

- Source 1
- Map
- Reduce
- Join
- Source 2
- Filter
- Join
- Sink
Query Plan Optimization

Source 1

Map

Reduce

Chain

Source 2

Join

Filter

Parallelize

Filter

Join

Sink
Goal = \langle \text{cost}, \text{latency}, \text{throughput} \rangle

cost + latency + throughput = 100\%
Cost-Optimizer (2)

rate Operator 1
rate Operator 2

window 1
window 2

Operator 1
Operator 2

$\alpha$

$\text{rate } 1$
$\text{rate } 2$
$\text{window } 1$
$\text{window } 2$

$\alpha$ parallelism

$\text{Cost}_\alpha$

$w_\alpha$ Output window
$cpu_{w_\alpha}$ CPU usage of the window
$size_{w_\alpha}$ RAM usage of the window
$r_\alpha$ Output Rate
Is it possible to extend the chain?

Yes! Let’s check if

1. is chainable

2. supports the parallelism of \textit{chain}

3. no back-pressure

4. \textit{Cost} complies the \textit{Goal} \langle cost, latency, throughput \rangle
Moira Architecture: Before

Query -> Flink

Flink

Goal
Moira Architecture: Static Analysis

Query
Stream SQL

Goal

Cost Optimizer

Flink

Flink
Static

Goal

Daniele Foroni - BIRTE 2018
Monitoring System

Apache Flink listener through JMX

- CPU
- Network
- Data Rate
- RAM
- Back-pressure
- Window size

Goal

Flink
Static
Dynamic
Feature Extractor

Metrics from the monitoring system

Builds the operator features (input rate/window size)

Invokes the cost estimator

Applies the rescheduling policy

Goal

Flink
Static
Dynamic

Monitoring System
MOIRA Architecture: Dynamic Analysis

Query

Goal

Cost Optimizer

Feature Extractor

Monitoring System

Static

Dynamic

Goal

Flink

Daniele Foroni - BIRTE 2018
Latency evaluation

![Graph showing latency evaluation with three lines representing different query plans: base (avg 1196.49 ms), static (avg 1119.26 ms), and dynamic (avg 965.22 ms). The graph illustrates the time (min) on the x-axis and latency (ms) on the y-axis. The graph shows a comparison of the new query plan and rescheduling, with a noted 7% improvement and a 20% improvement.]
Throughput evaluation

New Query Plan & Rescheduling

- **base (avg 253.04)**
- **static (avg 253.12)**
- **dynamic (avg 261.32)**

4 %
Open Challenges!

Machine Learning approaches to find relationships among variables…

…then, entropy analysis to simplify the problem!
Conclusion

- Goal-centric approach for each singular query

- **MOIRA**: framework for dynamic rescheduling

- Static and dynamic analysis

- Performance improvements
Thank you!

Questions?
References


Experiments

- TPC-H queries
- Adapted for streaming benchmark
- 4k events/second
- Running time: 120 minutes
TPC-H Queries

• Query Example

SELECT o_st, l_orderkey, o_shippriority,
    SUM(l_extendedprice) AS revenue
FROM orders, lineitem
WHERE l_orderkey = o_orderkey
    AND o_orderstatus = 'F'
    AND YEAR(o_orderdate) > 1993
    AND o_orderpriority LIKE '5%'
    AND o_proctime BETWEEN l_proctime - INTERVAL '2' HOUR
    AND l_proctime + INTERVAL '2' HOUR
GROUP BY o_st, l_orderkey, o_shippriority
TPC-H Queries

• Query Example

`SELECT o_st, l_orderkey, o_orderdate, o_shippriority, 
   SUM(l_extendedprice*(1-l_discount)) AS revenue, 
FROM customer, orders, lineitem 
WHERE c_mktsegment = 'AUTOMOBILE' 
   AND c_custkey = o_custkey 
   AND l_orderkey = o_orderkey 
   AND o_orderdate < date '1995-03-12' 
   AND l_shipdate > date '1995-03-12' 
   AND o_proctime BETWEEN l_proctime - INTERVAL '2' HOUR 
   AND l_proctime + INTERVAL '2' HOUR 
   AND o_proctime BETWEEN c_proctime - INTERVAL '2' HOUR 
   AND c_proctime + INTERVAL '2' HOUR 
GROUP BY o_st, l_orderkey, o_orderdate, o_shippriority`
TPC-H Queries

• Query Example

```
SELECT c_custkey, c_name, c_address, n_name, c_acctbal,
       SUM(l_extendedprice * (1 - l_discount)) AS revenue
FROM customer, orders, lineitem, nation
WHERE c_custkey = o_custkey
  AND l_orderkey = o_orderkey
  AND YEAR(o_orderdate) > 1990
  AND l_returnflag = 'R'
  AND c_nationkey = n_nationkey
  AND o_proctime BETWEEN l_proctime - INTERVAL '2' HOUR
  AND l_proctime + INTERVAL '2' HOUR
  AND o_proctime BETWEEN c_proctime - INTERVAL '2' HOUR
  AND c_proctime + INTERVAL '2' HOUR
  AND o_proctime BETWEEN n_proctime - INTERVAL '2' HOUR
  AND n_proctime + INTERVAL '2' HOUR
GROUP BY c_custkey, c_name, c_acctbal, n_name, c_address
```