Encapsulation of Parallelism in the Volcano Query Processing System

Huawei Wang
Overview

- Architecture
- Bracket Model
- Operator Model
- Pros & Cons
- Comparison
Typical Query Engine Architecture

Query
(SQL, SQL/XML, or XQuery!)

Parser

Global Query Semantics

Query ReWrite Transform

Plan OPTimization

ThreadedCodeGen

Plan Execution

Query Explain

Query Graph Model

Compile time

Run-time

Executable Plan
Similar Systems

- System R
- Starburst
Bracket Model

Problem:

• Extensibility
• Large overhead
Operator Model

Single process:
• Operator mapping to iterator (Can be applied to big data system)
• Use stream as abstractions for input between operators

Multiple process:
• Introduce exchange operator
  • Vertical parallelism
• Horizontal parallelism
Operator Model

![Diagram showing operator model with actions and states]

- open-filter()
- next-filter()
- close-filter()
- print()
- open-file-scan()
- next-file-scan()
- close-file-scan()
- predicate()
Vertical Parallelism

- Inter-process communication
  - Shared Memory
  - Semaphore

```plaintext
PRINT

JOIN

XCHG

XCHG

JOIN

XCHG

FS

FS

next_exchange

open_exchange

XCHG

FS

close_exchange
```
Horizontal Parallelism

- Bushy parallelism
- Intra-operator parallelism

Producer (sort A)  join  Consumer (sort B)
Horizontal Parallelism

How to partition data?

Producer

Queue 1: Consumer 1
Queue 2: Consumer 2
Queue 3: Consumer 3
Horizontal Parallelism

• Centralized scheme
• Propagation tree scheme
• Primed process
Pros & Cons

- More generalized
  - Algorithm Level
  - System Level
- Easy Implementation
- Heavy weight creating process
Comparison

- Spark can choose whether to persist RDD
- Volcano only let intermediate results exist in buffer
- Volcano is only a query execution engine with 2 key meta operators.