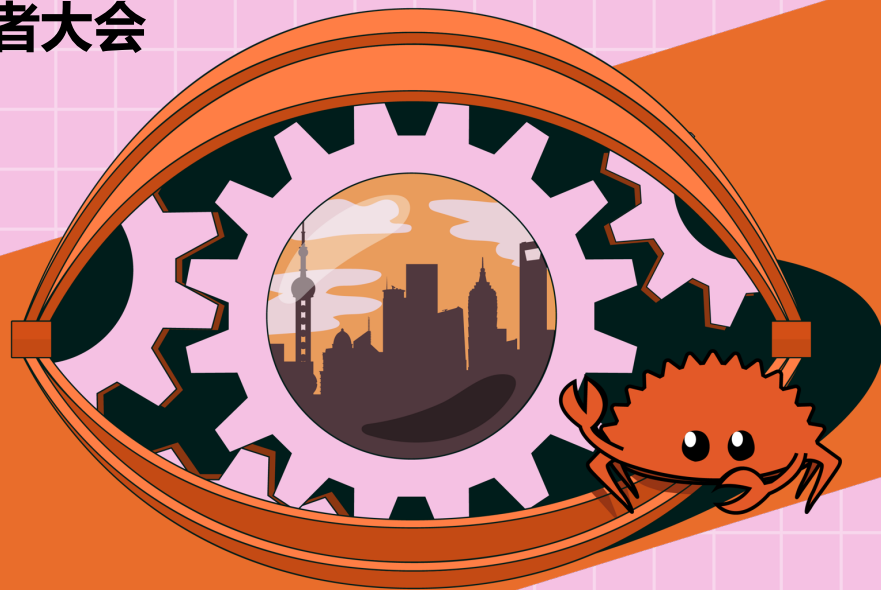


RUST CHINA CONE 2023

第三届中国Rust开发者大会



6.17-6.18 @Shanghai

Apache Ballista Introduction

钟阳红 (John Zhong)

Software Engineer @ eBay

nju_yaho@apache.org



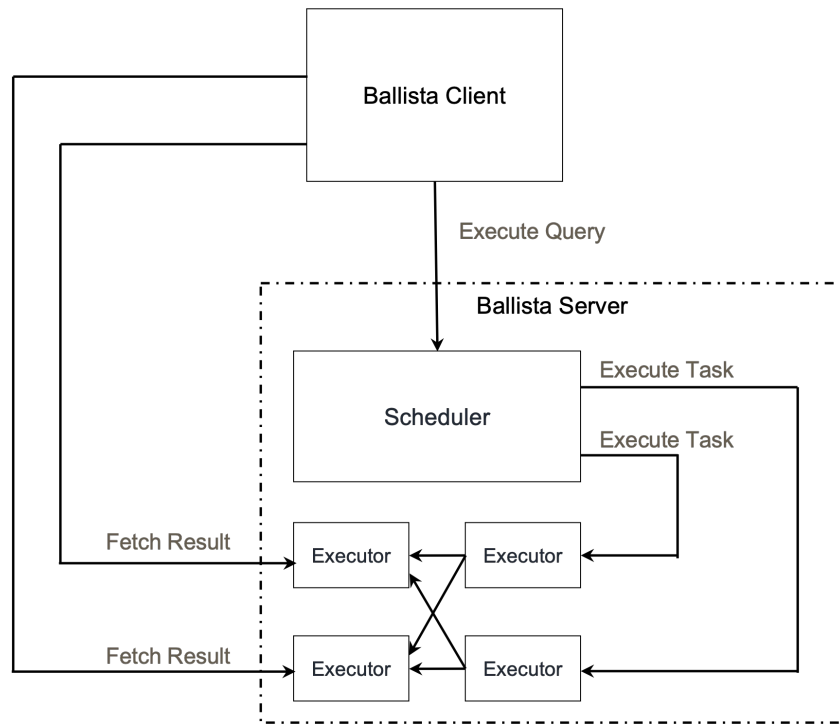
Agenda

- Overview
- Cluster Setup
- SQL Execution
- Data Cache
- Future

Overview

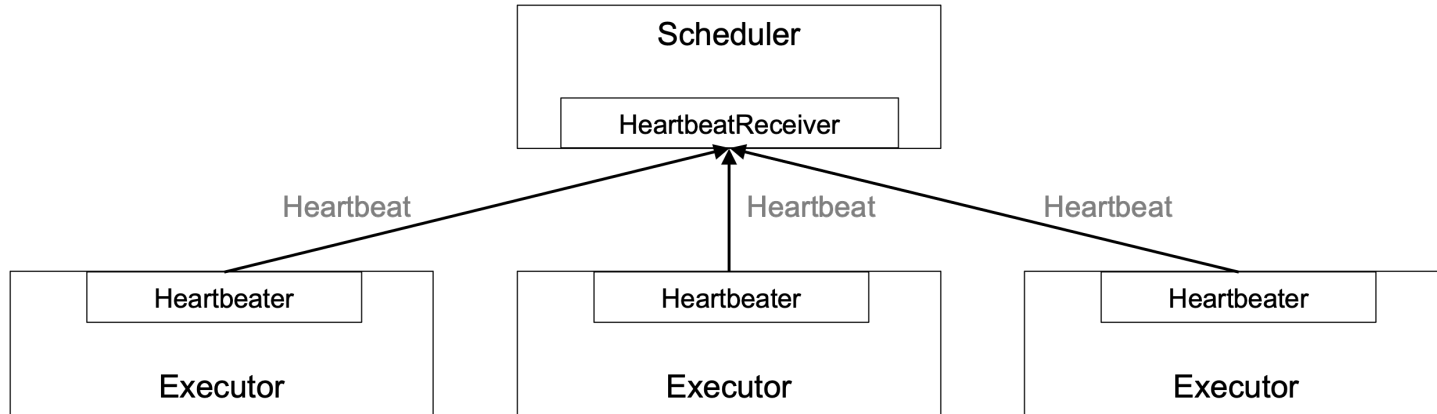
Apache Ballista is a distributed SQL query engine powered by the Rust implementation of Apache Arrow and DataFusion. It's mainly for interactive queries of low latency.

- Support DAG and fault tolerance
- Support data exchange
- Support different kinds of object stores, like HDFS, S3, Azure, etc
- Support data cache and cache aware task scheduling



Cluster Setup

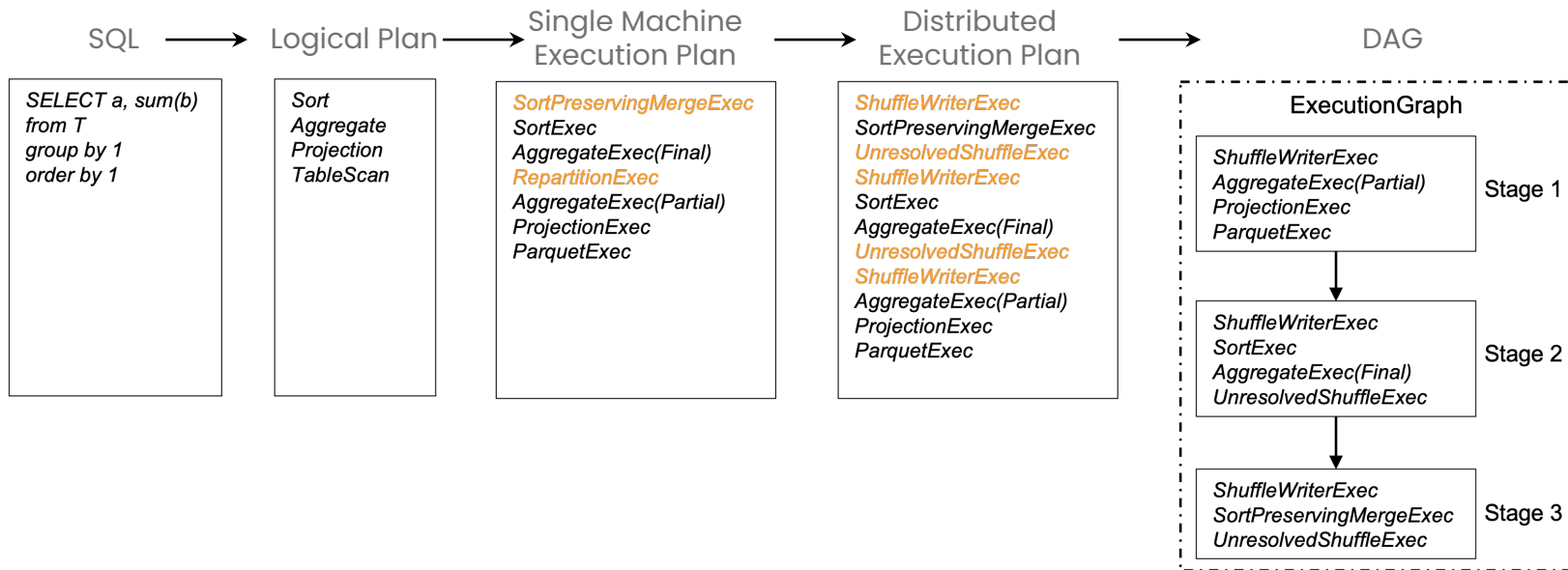
The cluster consists of one scheduler and a number of executors. Both of scheduler and executor can be deployed on K8S. Executors can be added to the cluster flexibly by registering to the cluster scheduler.



SQL Execution

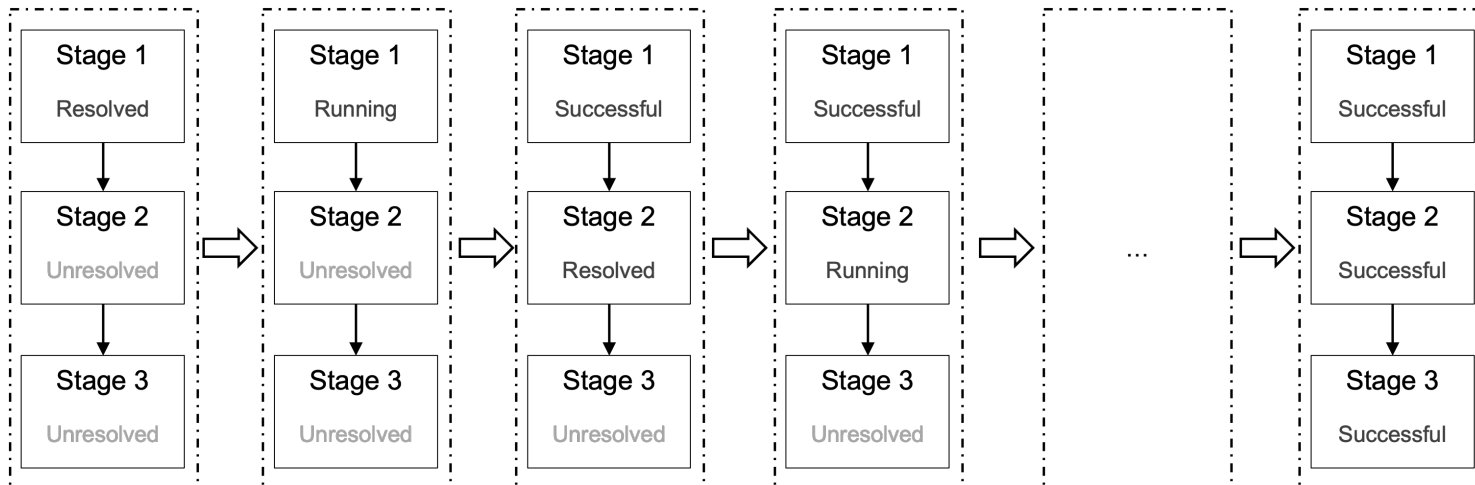
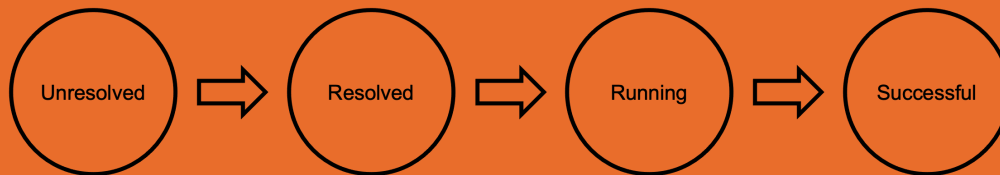
- SQL -> DAG (Directed Acyclic Graph)
- DAG State Machine
- Task Assignment
- Event Loop based Processing

SQL Execution — DAG Generation



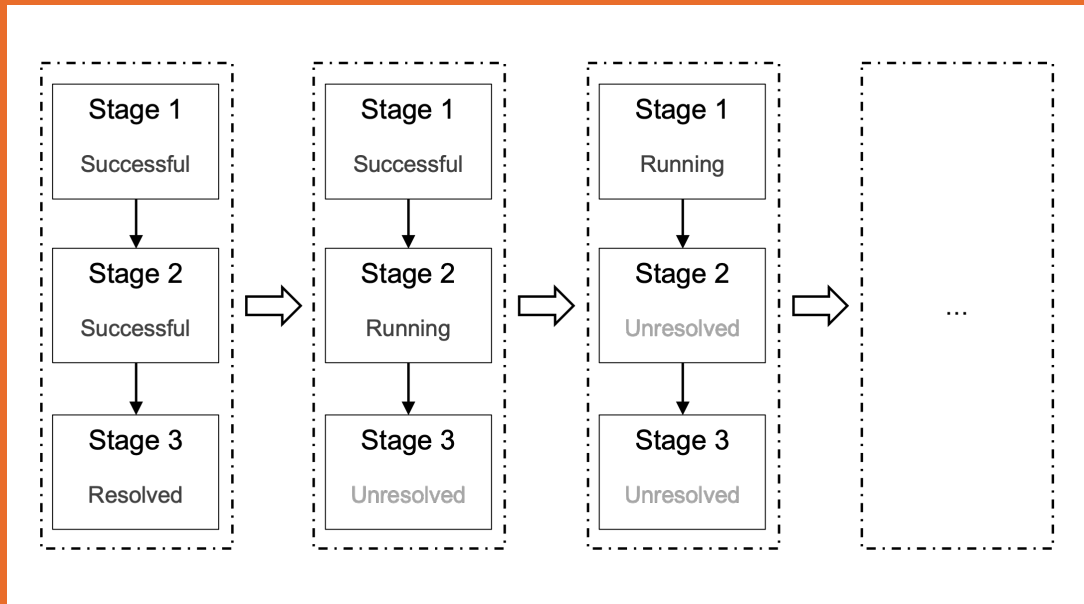
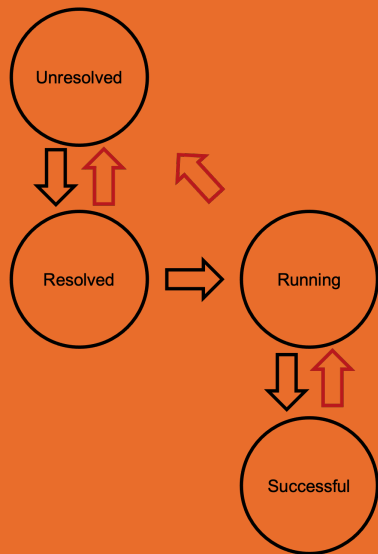
SQL Execution — DAG State Machine

Normal Stage State Machine



SQL Execution — Fault Tolerance

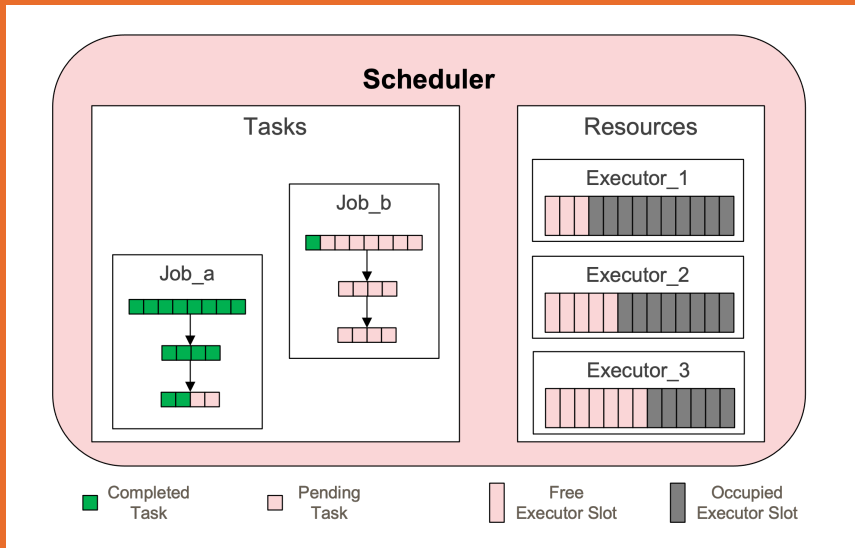
Stage State Machine
for Executor Lost



SQL Execution — Task Assignment

Task: each execution stage for a number of data partitions.
one task for each data partition.

Executor slot: each executor has a number of slots for task execution.



One round task assignment will bind pending tasks with available executor slots as many as possible.

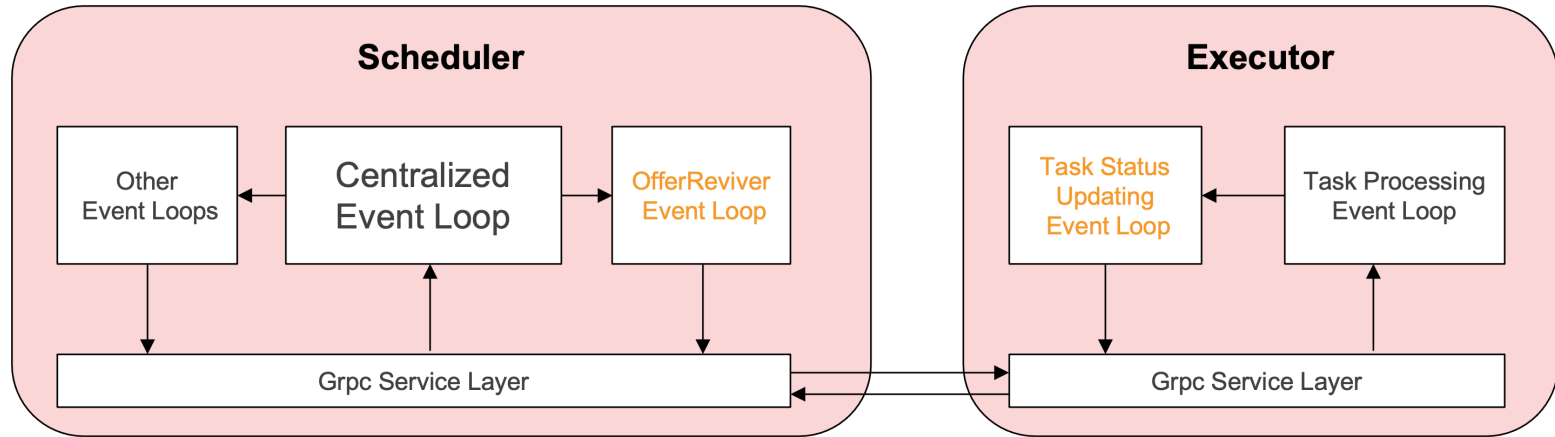
Two assignment policies:

Policy	Result of One Round
Round-robin	Job_a: 1 slot from executor_3 1 slot from executor_2 Job_b: 2 slots from executor_3 2 slots from executor_2 3 slots from executor_1
Bias	Job_a: 2 slots from executor_3 Job_b: 5 slots from executor_3 2 slots from executor_2

SQL Execution — Event Loop based Processing

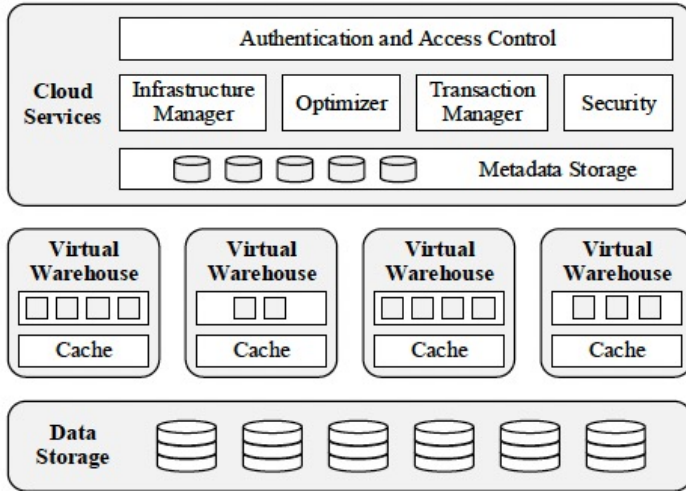
Advantages:

- Decoupled
- Efficient processing for batch events

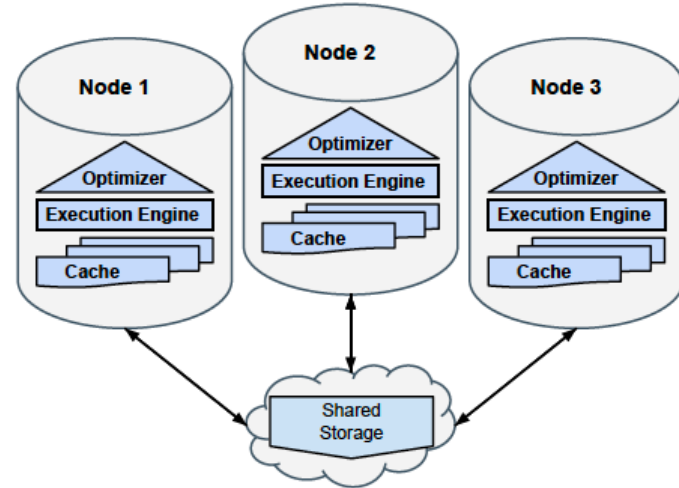


Data Cache

Data cache is a very common feature for the cloud data warehouses for accelerating the access to the data source.



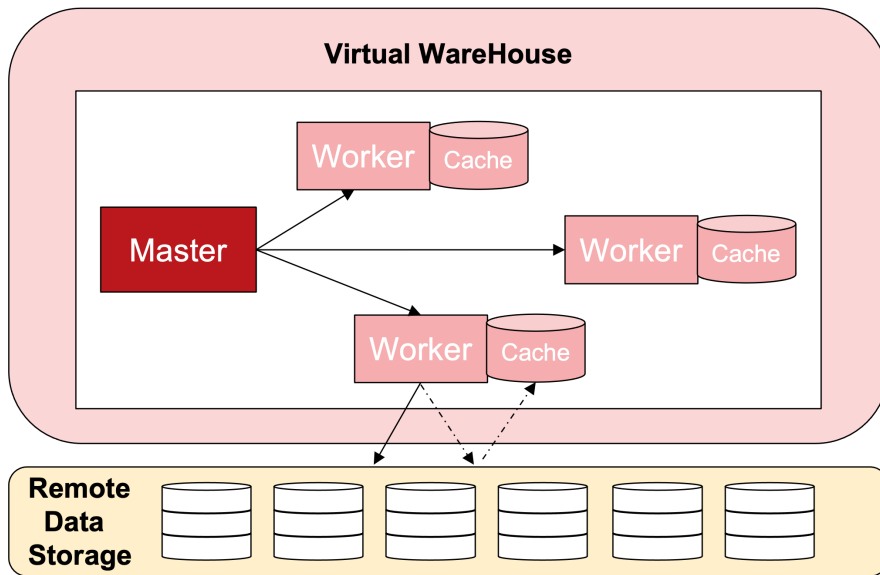
Snowflake – Multi-Cluster Shared Data Architecture



Vertica – Eon Architecture

Data Cache

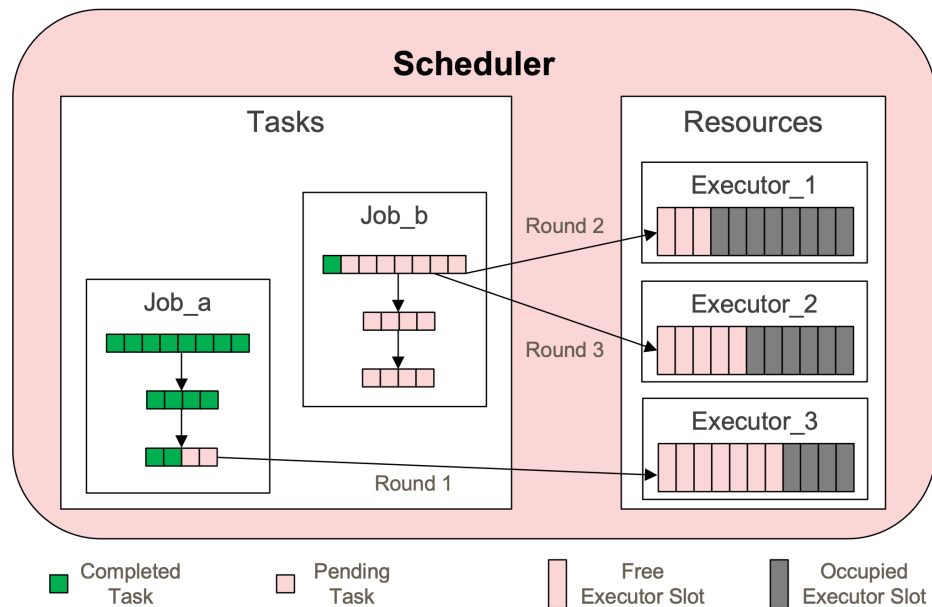
- Consistent hashing-based assignment (Snowflake)
- LRU based retirement
- Cache aware scheduling
- Consistent hashing tolerance-based work stealing
- Currently it's file-level



Data Cache

Three rounds cache aware task Scheduling:

- Assign non-map stage tasks (without scanning files) in a round robin way
- Assign map stage tasks (scanning files) based on the consistent hashing policy on the hash value of the file name and the executor topology
- Assign tasks with scanning files based on the consistent hashing policy on the hash value of the file name and the executor topology with N tolerance



Future

- Scheduler HA
- Shuffle Improvement
 - Self-adjustable shuffle partition number
 - Sort-based shuffle writer for pull-based shuffling
 - Push-based shuffling

Reference

- Eon Mode: Bringing the Vertica Columnar Database to the Cloud
https://www.vertica.com/wp-content/uploads/2018/05/Vertica_EON_SIGMOD_Paper.pdf
- The Snowflake Elastic Data Warehouse
<https://event.cwi.nl/lsde/papers/p215-dageville-snowflake.pdf>
- Apache Arrow
<https://arrow.apache.org/>
- Apache Arrow DataFusion
<https://github.com/apache/arrow-datafusion>
- Apache Arrow Ballista
<https://github.com/apache/arrow-ballista>

Thank you !

