Overview of Apache Flink

Zichen Liu, Haopeng Zhang,
Zirui Tan, Yuting Wang

Overview

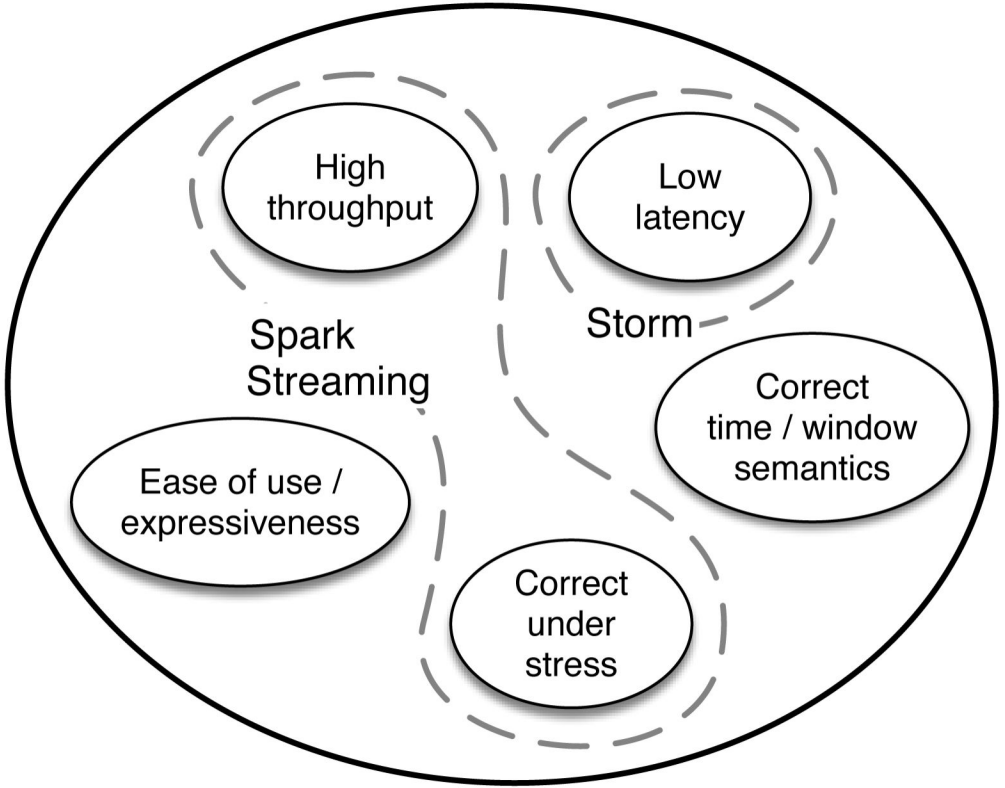
- Architecture
  - Flink Core & Components
  - Flink Runtime
  - Job client, manager
- Features
  - Streaming & Batching
  - Stateful Processing
  - Fault Tolerance
  - Memory Management
  - Layered API
  - Exactly once Processing
- Use Cases
  - Data-driven, Data Analytics, Data Pipeline

What is Apache Flink ?

- Core: a real streaming dataflow engine that provides data distribution, communication and fault tolerance for distributed computations over data streams
- Access the data in-time
- Flink provides
  - Dataset API - for bounded streams
  - Datastream API - for unbounded streams
- Flink embraces the stream as abstraction to implement its dataflow.

Flink Capacities

Flink combines many of the desired traits needed to efficiently process data from continuous events.



Flink core

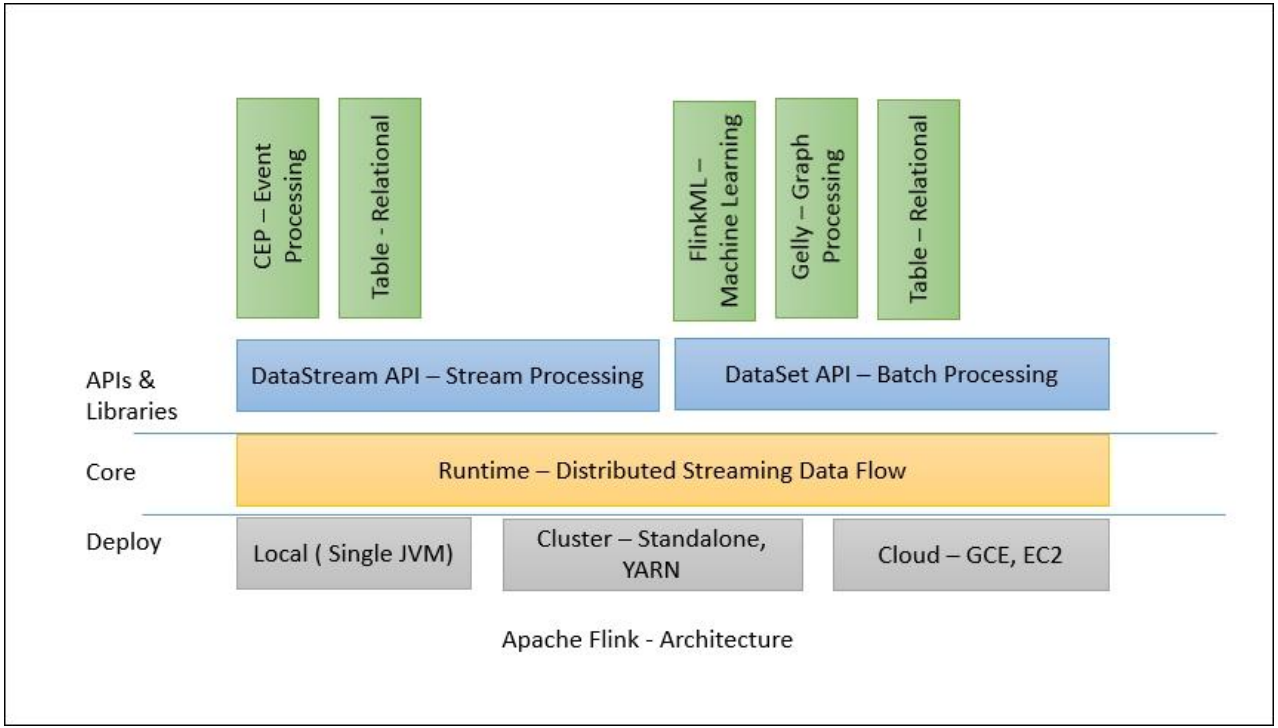
- Flink coere
  - Job Client
  - Job Manager
  - Task Manager
  - The main process of runtime

Flink Components

API layer

Runtime Layer

Deploy Layer



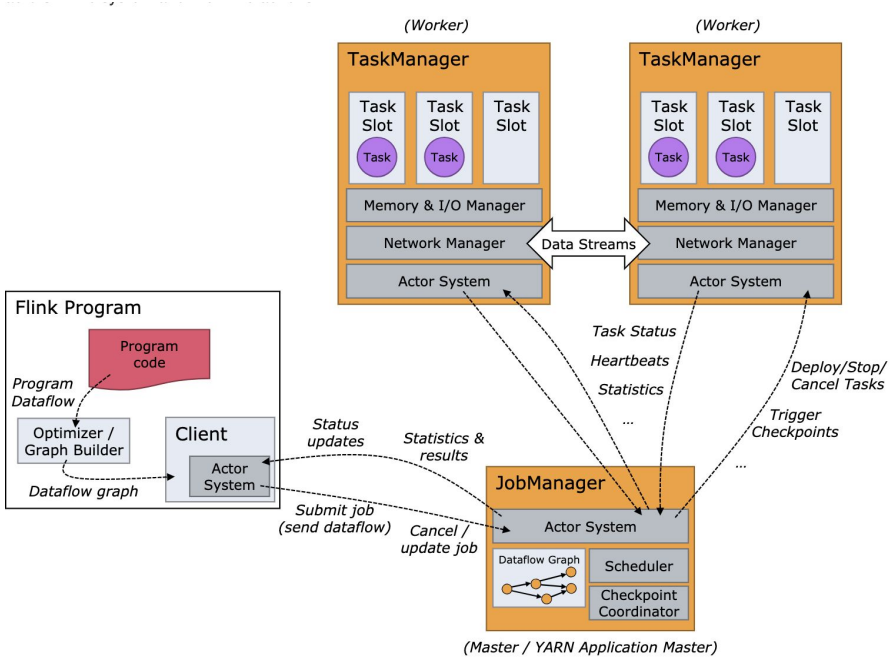
Flink Runtime

Runtime main components.

Job client

Job Manager

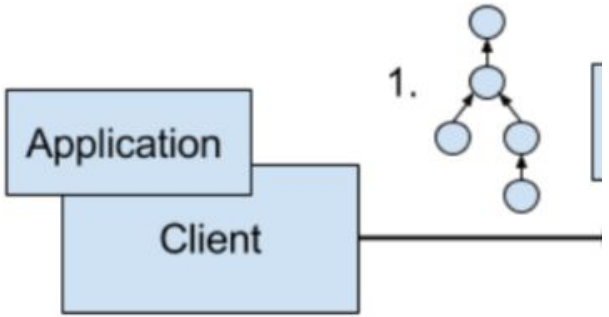
Task Manager



Job client

The job client’s responsibility

Submitting the data flow(job flow) to the Job Manager for further execution. Once the execution is completed, the job client provides the results back to the user.



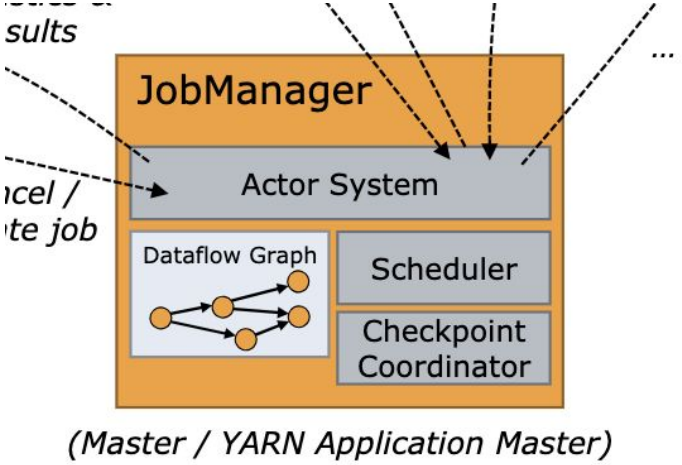
Job manager

Job manager's responsibility:

Coordinate and manage the execution of the program. .Their main responsibilities include scheduling tasks, managing checkpoints, failure recovery.

It will convert the job graph into a execution graph here

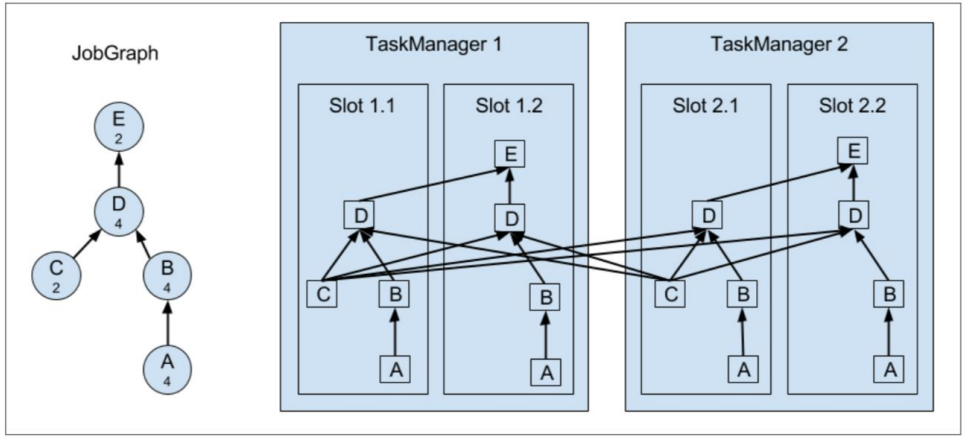
And ship them to task manager.



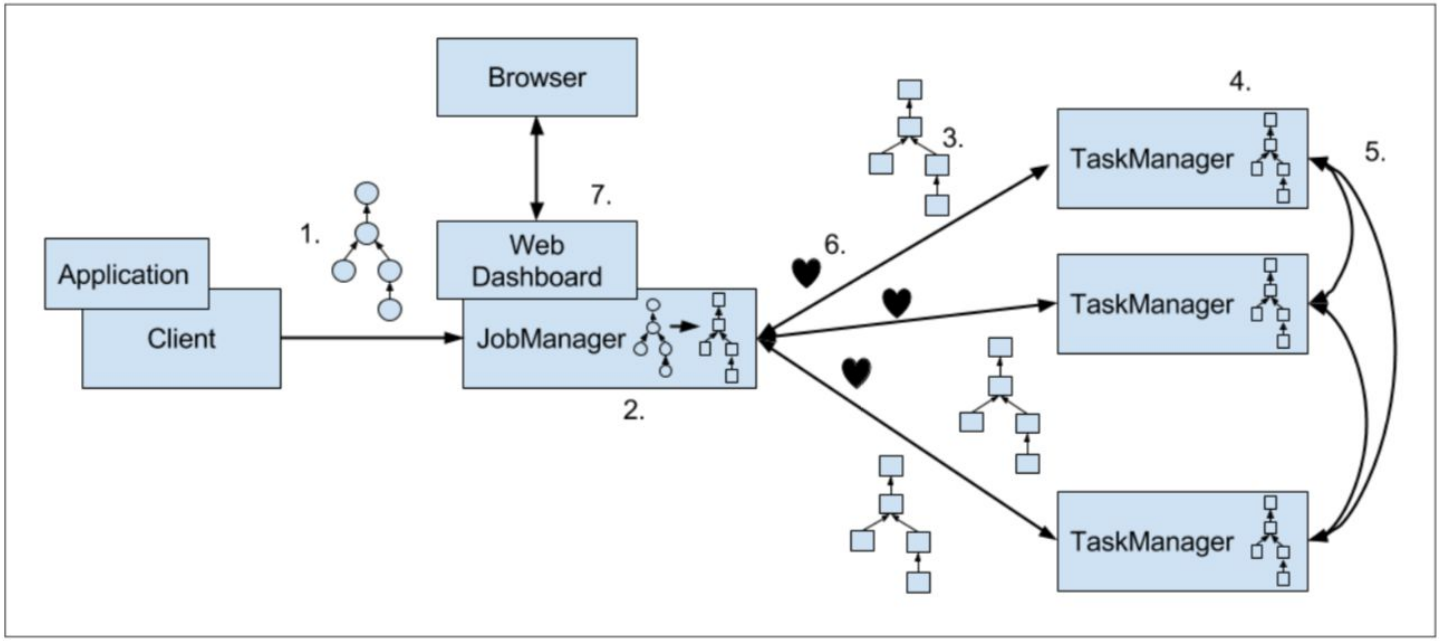
Task manager

Task managers are worker nodes that execute the tasks in one or more threads in JVM. Parallelism of task execution is determined by the task slots available on each Task Manager.

A TaskManager provides a certain number of processing Slots to control the number of subtasks that can be executed concurrently. A processing slot is able to execute one slice of an application. During execution, it will handle the data transfer.



The process of runtime

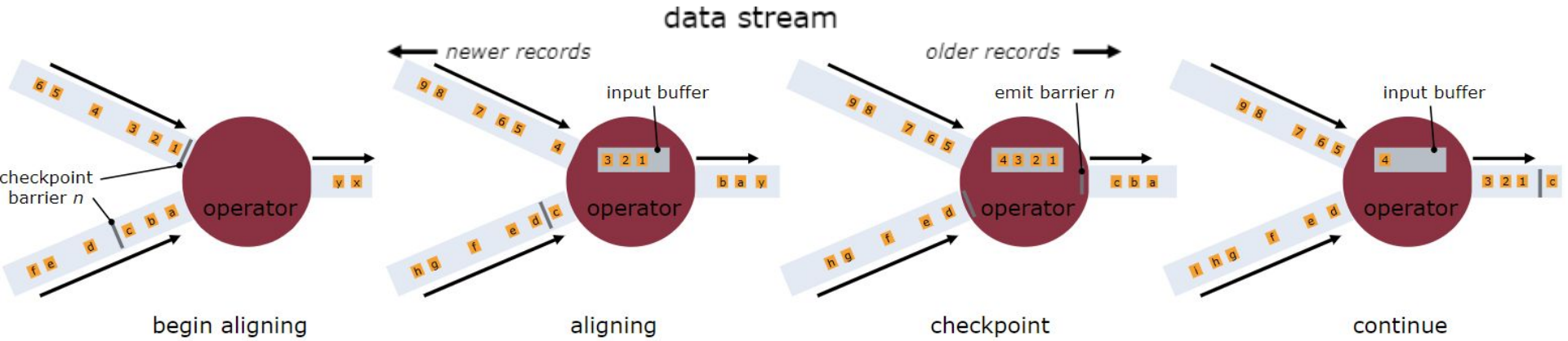


The job client will ship the job graph for the job manager.

JobManager interface.

Features

- ❏ Streaming & Batching
- ❏ Memory Management
- ❏ Layered API
- ❏ Stateful Processing
- ❏ Fault Tolerance
- ❏ Exactly once processing



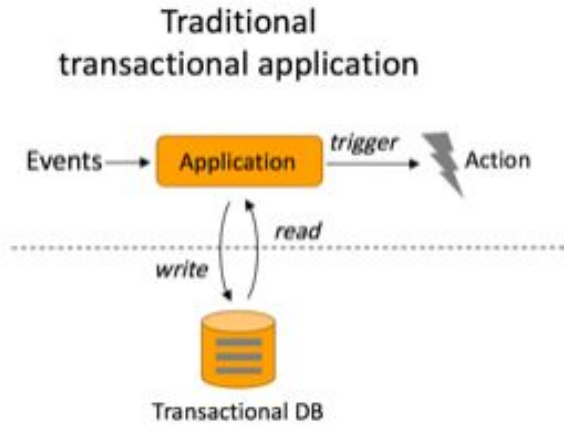
Use Cases

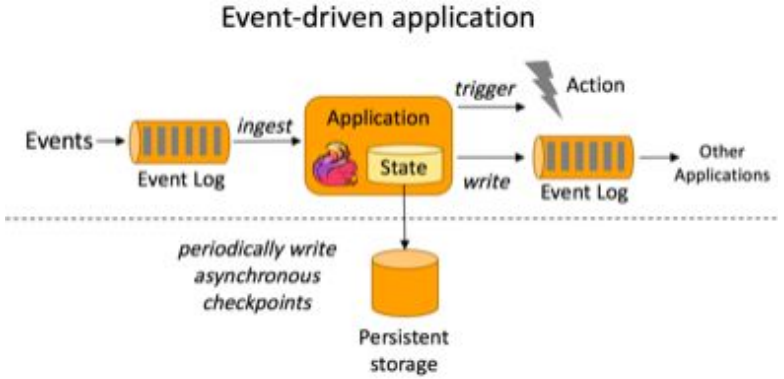
- Event-driven Applications

1. Based on stateful stream processing applications
2. Access data locally

   => Better latency, better throughput
3. Responsible for its own data

   => Changes to the data requires less coordination





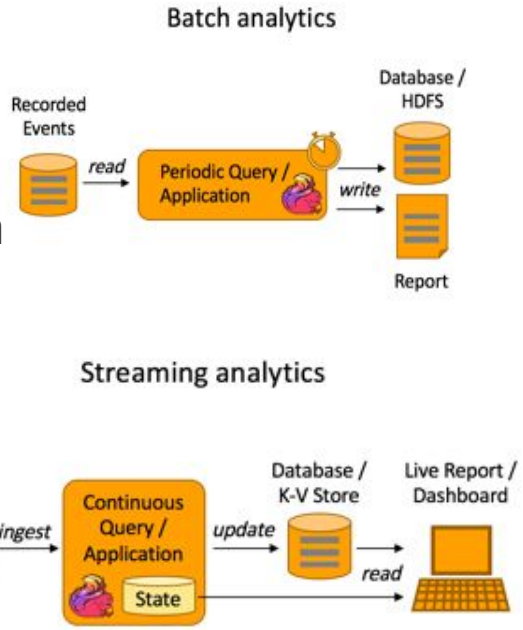
❑ Data Analytics Applications

1. Data Analytics performed in a real-time fashion
2. Elimination of periodic import and query execution

   => Much lower latency

3. Runs on a sophisticated stream processor

   => Simpler application architecture



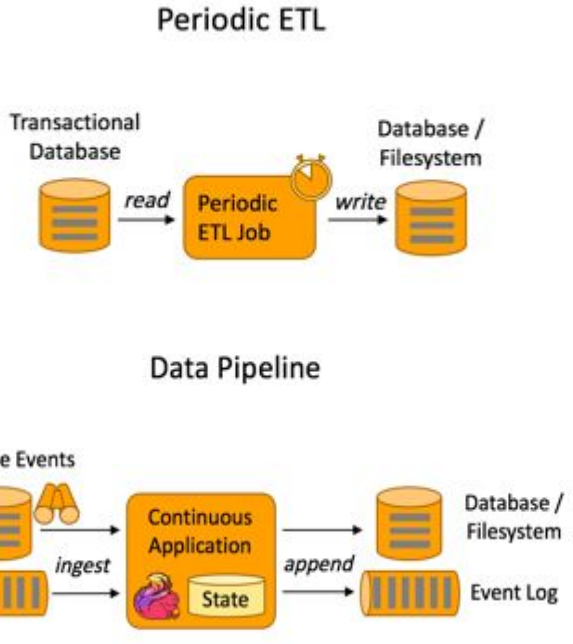
- Data Pipeline Applications

1. Lower latency compared with ETL
2. Continuous streaming mode

   =>Read records from sources Continuously

   =>Move data with low latency to their destination
3. Continuously consume and emit data

   => Employed for more use cases



References

[1].https://www.oreilly.com/library/view/introduction-to-apache/9781491977132/ch01.html

[2].https://www.cnblogs.com/feiyudemeng/p/8998772.html

[3].https://hackernoon.com/what-makes-apache-flink-the-best-choice-for-streaming-applications-fc377858a53

[4].https://flink.apache.org/zh/flink-architecture.html