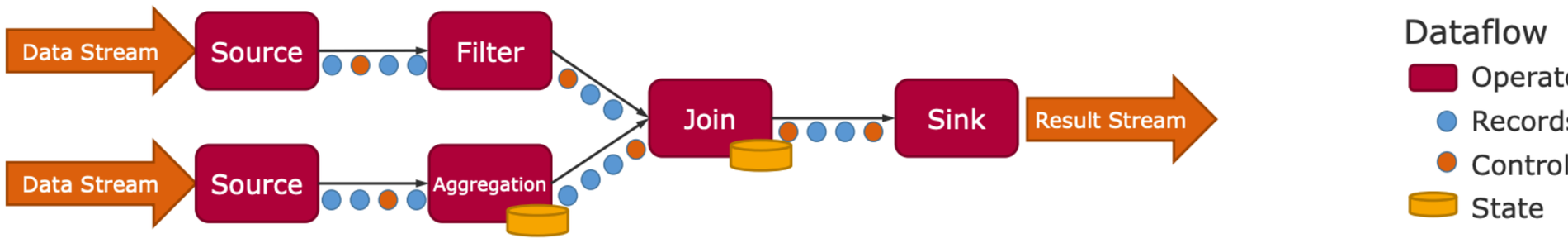
Stream Processing: Continuous Optimization

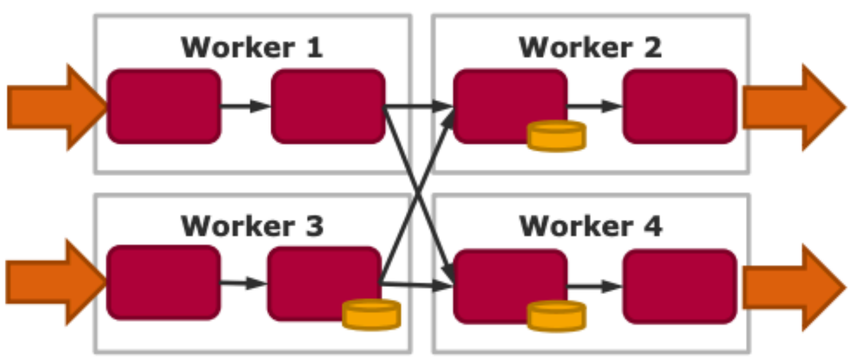
Stream Processing

Streaming: a type of data processing engine that is designed with infinite data sets in mind

Why use stream processing? On one data stream multiple queries can be done. They deliver continuous results. Streaming is a push model, so the data production and procession is controlled by the source. There are different ways to decide when the data is produced / processed data should be put out:



Current Solution:



With stream processing, you have a long-time operator that you work on continuously. With multiple workers and a pipeline, some parts are data-parallel and others are test-parallel. So the different workers can perform different calculations or the same tasks on different parts of the data. Pipeline parallelism occurs because different workers have different parts of the pipeline.

Problem:

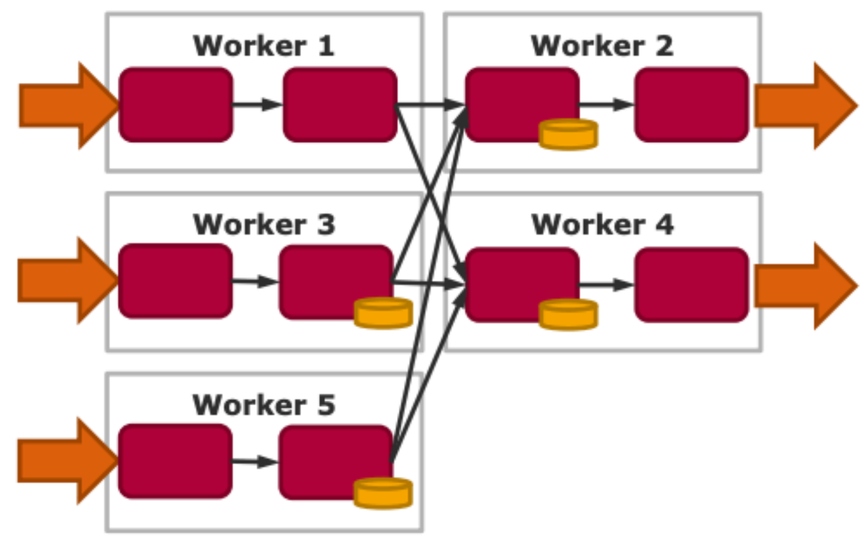
Partitioning is decided at the time of deployment by the choice or selection of the system. If the system is inefficient, for example, if there is an uneven distribution of workers' calculations, you must stop the system. You will have to repartition the data and correct the code, which will cause downtime.

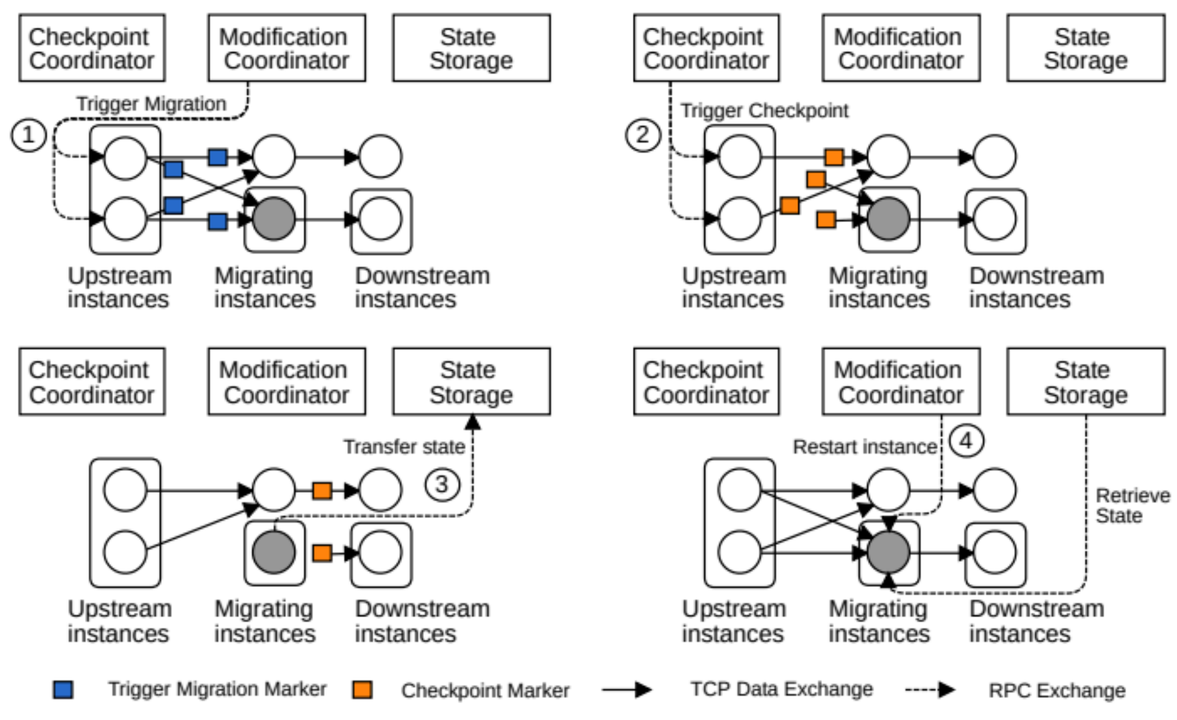
Current state: stop and restart

***Better** solution: Continuous Optimization*

The partitioning can be fixed by reconfiguration at **runtime**. For example with the use of **watermarks** jobs can be duplicated when they reach a certain level of computation. So data will be rerouted in a system and updates are getting handled.

Continuous Optimization:





Modification Protocol:

- Migrating operator instances
- Introducing new operator instances
- Changing an operator's UDF (user-defined functions)

Up and down-scaling operator instances enable:

Goal:

Dynamically adaption of Stream Processing Engines of incoming workload without halting the whole job

Solution:

Further research and implementation of the elastic approach through cloud architectures

Quentin Kuth

IT-Systems Engineering
Hasso Plattner Institute, Potsdam, Germany

E-Mail: quentin.Kuth@student.hpi.uni-potsdam.de

HPI Hasso Plattner Institut