**Stream processing with streaming dataflows**

Graphical user interface, text, application

Description automatically generated

Let’s say we have a code snippet of a Flink program as shown in the image above.

Step 1: The jar file of this program is distributed amongst all nodes inside the Flink cluster.

Diagram

Description automatically generated

Step 2: Flink converts all the user-defined transformations into a **streaming dataflow**, which is a data structure that encloses the order of operations applied to the input data stream. This dataflow can be represented as a directed graph that always starts with source(s) and ends at sink(s).

Diagram, schematic

Description automatically generated

Step 3: Flink converts the dataflow into an execution graph. This graph not only depicts how the abstract streaming dataflow is mapped to the system’s available resources, it also shows the level of parallelism of the job’s execution process.

**References:**

https://nightlies.apache.org/flink/flink-docs-release-1.17/