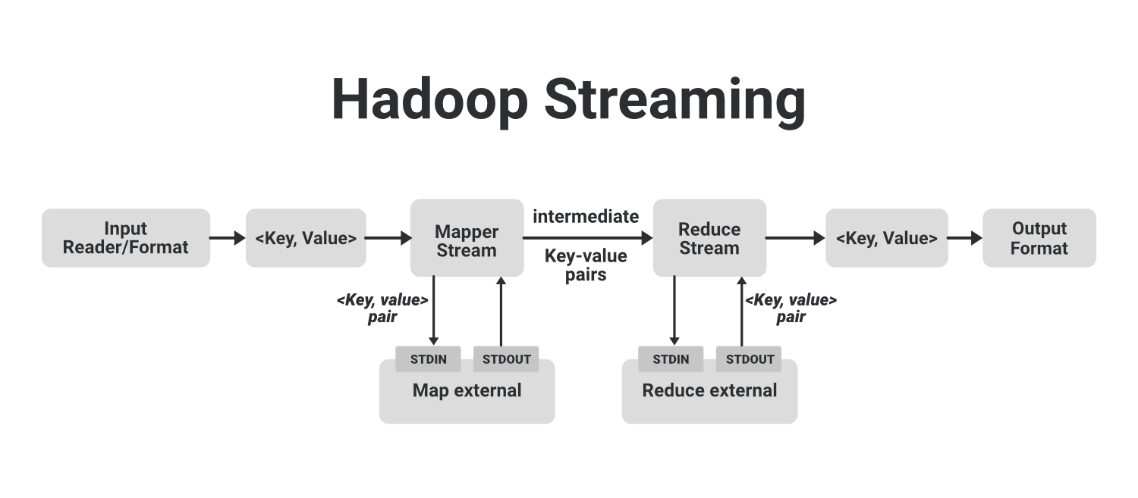
**Hadoop Streaming:**

* Hadoop Streaming is a utility included with Hadoop to write MapReduce programs using **any scripting or programming language** that supports reading from standard input (stdin) and writing to standard output (stdout).
* It's ideal for users who prefer working with high-level scripting languages like **Python**, **Ruby**, or **Perl** instead of Java.

**How it Works?**

* Hadoop Streaming runs custom **mapper** and **reducer** scripts or commands as part of the Hadoop MapReduce process.
* Hadoop handles the split of data into input splits, feeds them to your custom mapper and reducer via stdin, and collects the results via stdout.
* The **key-value pairs** generated by the mapper are shuffled and sorted automatically by Hadoop before being passed to the reducer.

**Architecture:**

**Components:**

 **Mapper**:

* A script or program that processes input data line-by-line.
* Reads from stdin and outputs key-value pairs to stdout, separated by a tab (\t).

 **Reducer**:

* A script or program that receives sorted key-value pairs.
* Processes these and outputs the final aggregated result.

 **Input/Output**:

* The input files are divided into splits (blocks).
* Each split is processed in parallel by different mapper tasks.
* The output of reducers is written to HDFS.

**Steps to Use Hadoop Streaming:**

* **Prepare Your Scripts**:

Write a mapper script and a reducer script in your preferred language.

* **Run the Job**:

Use Hadoop’s command-line tool to submit your scripts as a streaming job.

* **View Output**:

The output will be stored in the specified HDFS directory.

**Advantages of Hadoop Streaming**

1. **Language Independence**:
   * Write programs in any language, avoiding Java.
2. **Ease of Use**:
   * Simple and fast for prototyping or smaller tasks.
3. **Integration**:
   * Ideal for leveraging existing scripts or programs.

**Disadvantages**

1. **Performance**:
   * Slower than native Java or C++ implementations due to script interpretation.
2. **Debugging**:
   * Error handling and debugging can be less straightforward compared to Java.

**Hadoop Pipes:**

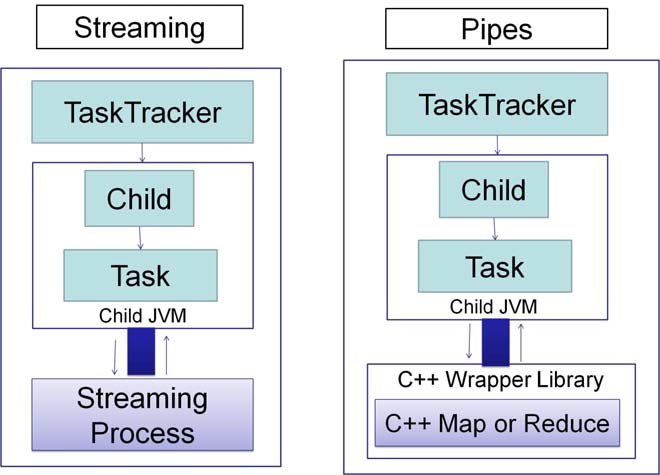
* Hadoop Pipes is a framework that enables writing MapReduce programs in **C++**. It provides a high-performance alternative to Hadoop Streaming and integrates natively with the Hadoop ecosystem.

**How it Works?**

 Hadoop Pipes uses a **C++ API** that communicates with Hadoop's Java processes using the **Hadoop Pipes Protocol** (based on sockets).

 The Pipes framework allows users to write:

* **Mapper**
* **Reducer**
* **Combiner**
* **Partitioner**

**Architecture:**

**Components:**

 **Mapper**:Written in C++, the mapper transforms input into key-value pairs.

 **Reducer**:

Aggregates or processes the key-value pairs output by the mapper.

 **Factory Methods**:

C++ programs must implement factory methods to create and initialize the mapper, reducer, and other components.

 **Hadoop Pipes Protocol**:

A lightweight binary protocol used to exchange data between Hadoop and the C++ program.

### ****Steps to Use Hadoop Pipes****

1. **Write the Code**:
   * Implement the mapper, reducer, and any other required components using the Hadoop Pipes API.
2. **Compile the Code**:
   * Use a C++ compiler to compile the code into a shared library or executable.
3. **Run the Job**:
   * Submit the job using Hadoop with the Pipes program as the mapper and reducer.

### ****Advantages of Hadoop Pipes****

1. **Performance**:
   * Faster execution due to compiled C++ code.
2. **Resource Efficiency**:
   * Less resource-intensive than interpreted scripts.
3. **Flexibility**:
   * Offers fine-grained control of MapReduce processes.

### ****Disadvantages****

1. **Complexity**:
   * Requires knowledge of C++ and the Hadoop Pipes API.
2. **Portability**:
   * C++ programs must be compiled for the target system.