

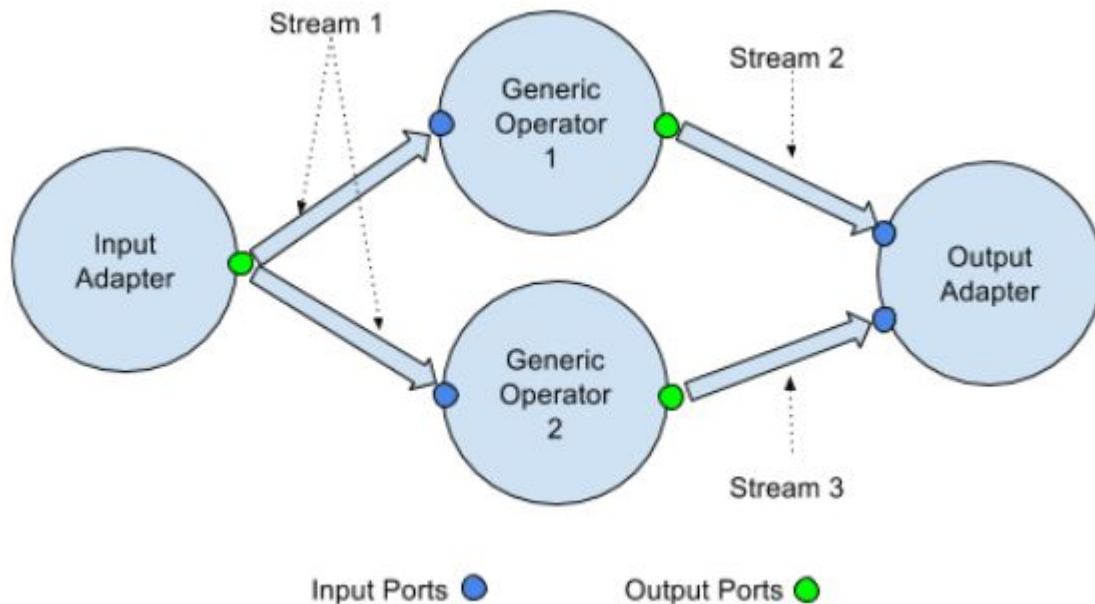


Apex Hands-on Lab - Into the code!

Getting started with your first Apex Application!

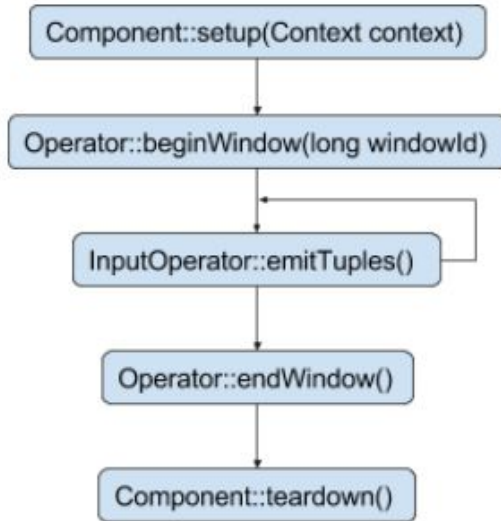
Akshay Gore, Bhupesh Chawda
DataTorrent

Operators

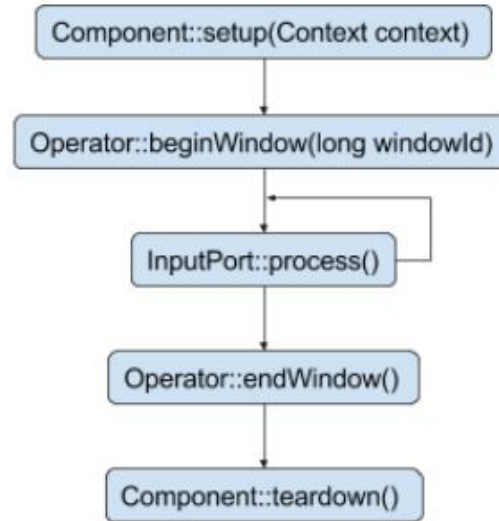


- Input Adaptor Vs Generic Operators ?
- What are streams?
- What are ports?

Apex Operator Lifecycle



Flow for Input Adapters



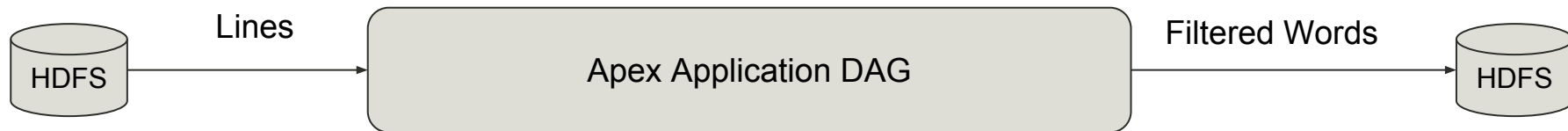
Flow for Generic Operators
and Output Adapters

Apex Streaming Application

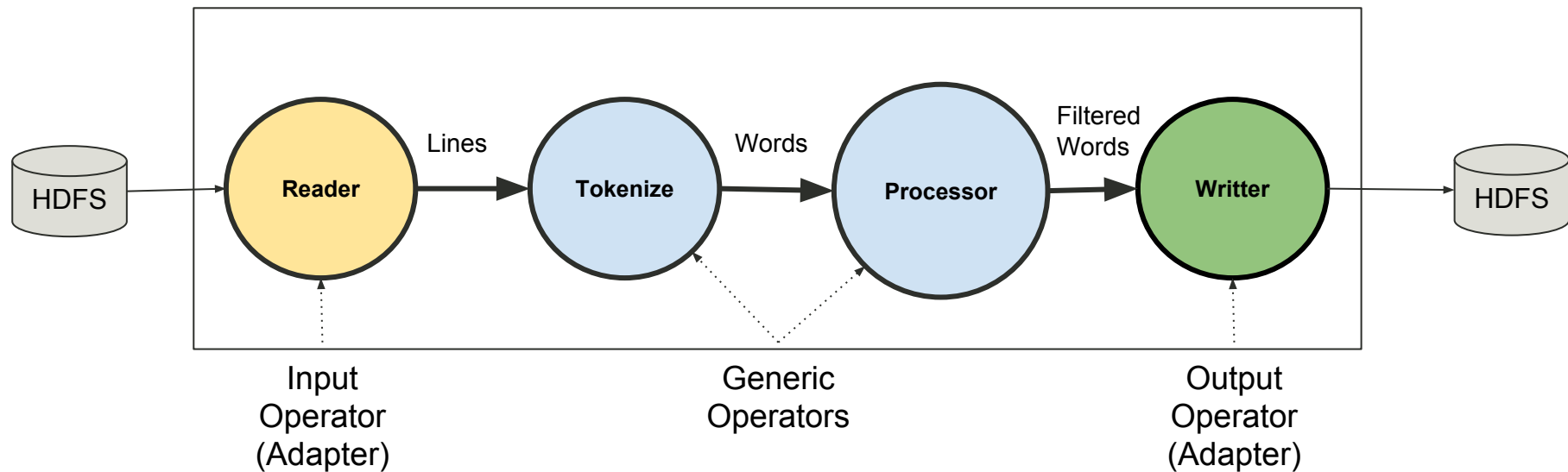
```
public class Application implements StreamingApplication
{
    populateDAG(DAG dag, Configuration conf)
    {
        // Add Operators to dag - dag.addOperator(args)
        // Add Streams between operators - dag.addStream(args)
        // Additional config + Hints to YARN - Optional
    }
}
```

Apex Application - FilterWords

- Problem statement - Filter words in the file
 - Read a file located on HDFS
 - Split each line into words, check if it is not one of the forbidden words and write it down to HDFS



FilterWords Application DAG



Prerequisites

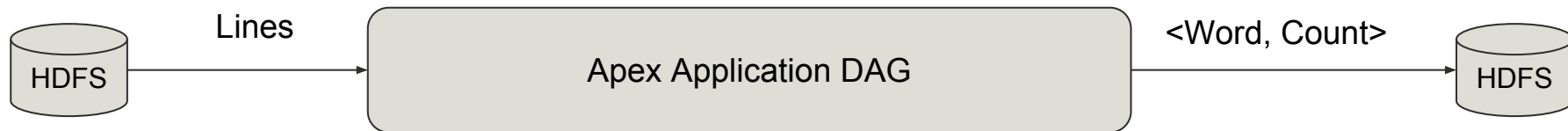
- JAVA 1.7 or above
- Maven 3.0 or above
- Apache Apex projects:
 - Apache Apex Core: core platform, engine
 - Apache Apex Malhar: operators library
- Hadoop cluster in running state
- Your favourite IDE - Eclipse / vi

Demo time!

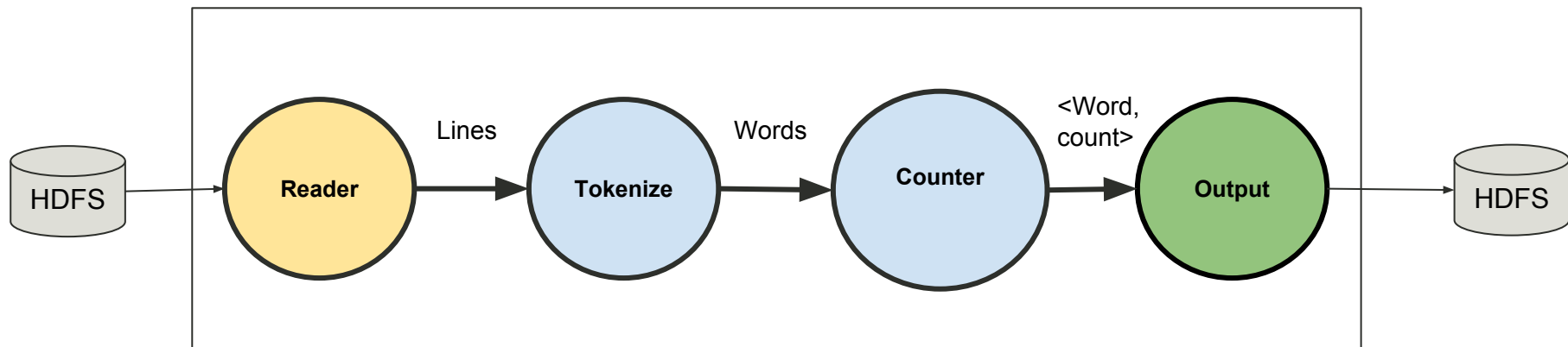
- Apex application structure
- Application code walk through
- How to execute the application
- Assignment

Assignment - WordCount

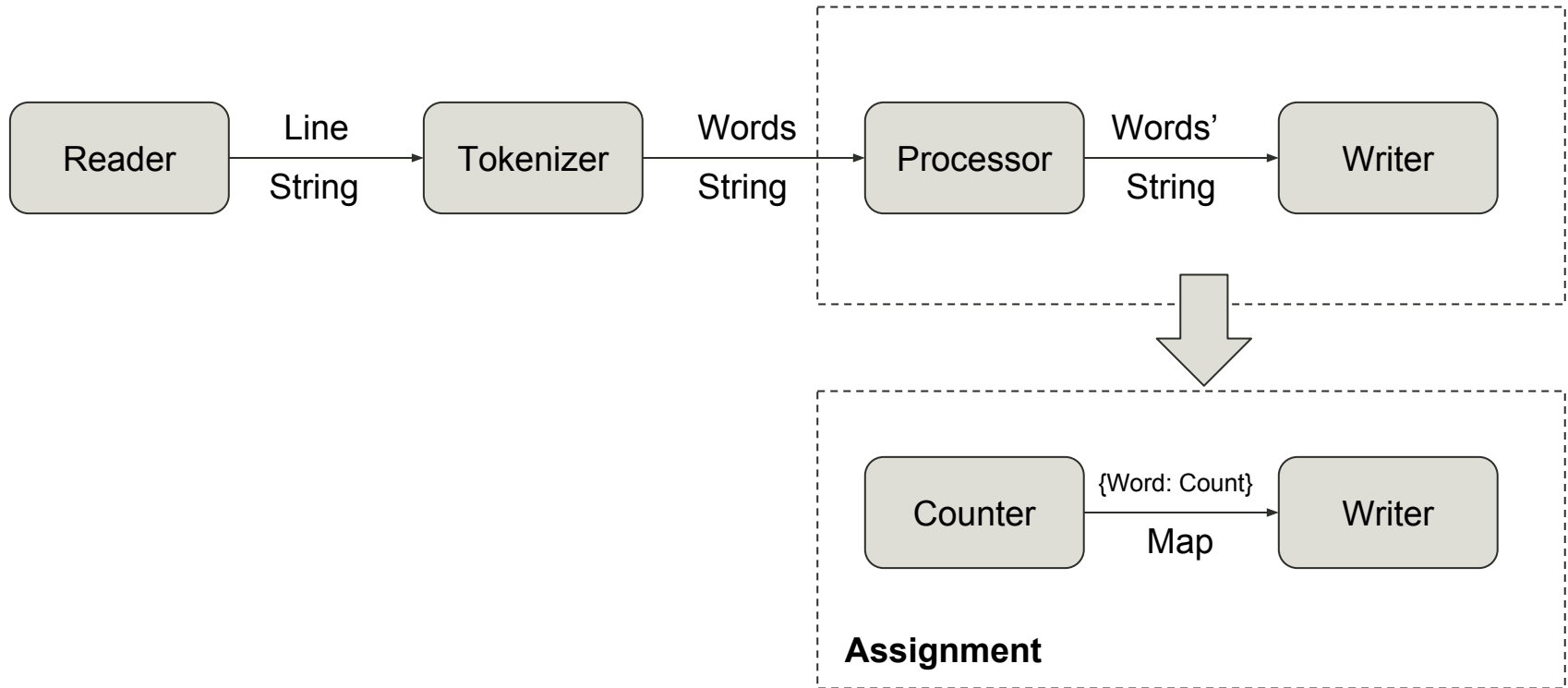
- Problem statement - Count occurrences of words in a file
 - Read a file located on HDFS
 - Emit count at the end of the every window and writes into HDFS



Assignment - Word Count Application DAG



Assignment - What you need to do



Assignment - Hints

- Create copy of Processor.java. Name it Counter.java
- Modify Counter.java as follows:
 - Define a data structure which can hold counts for words
 - Process method of input port must count the occurrences
 - Clear the counts in beginWindow() call
 - Emit the counts in endWindow() call

Solution - Changes to Counter.java

- Need to define a data structure which can hold counts for words

```
private HashMap<String, Integer> counts = new HashMap<>();
```

- Process method of input port must count the occurrences

```
if(counts.containsKey(refinedWord)) {  
    counts.put(refinedWord, counts.get(refinedWord) + 1);  
} else {  
    counts.put(refinedWord, 1);  
}
```

- Clear the counts in beginWindow call

```
counts.clear();
```

- Emit the counts in endWindow call

```
output.emit(counts.toString());
```

- Run Application Test

Assignment - Are we done yet?

- Change the DAG
 - Replace Processor operator with the newly created operator - Counter

Assignment - Slight change

- We are emitting a Map. However it is still a string.
 - Change type of output port of Counter to type Map
 - Change type of input port of Writer to Map
 - Make appropriate changes to Writer to read a Map and write in a format such that each line belongs to a single word.

Assignment - Final change

- Change the code such that each count is the overall count, not just for each window?

Summary - Recap

- Writing Apache Apex *operators*
- Chaining the operators into an Apache Apex *application*
- Executing the application on the Apache Apex *platform*

Where to go from here?

Apache Apex Documentation - <http://apex.incubator.apache.org/docs.html>

Apache Apex Core Git - <https://github.com/apache/incubator-apex-core>

Apache Apex Malhar Git - <https://github.com/apache/incubator-apex-malhar>

Join Users Mailing List - users-subscribe@apex.incubator.apache.org

Join Dev Mailing List - dev-subscribe@apex.incubator.apache.org

Send queries to Users Mailing List - users@apex.incubator.apache.org

Send queries to Dev Mailing List - dev@apex.incubator.apache.org

Thank You