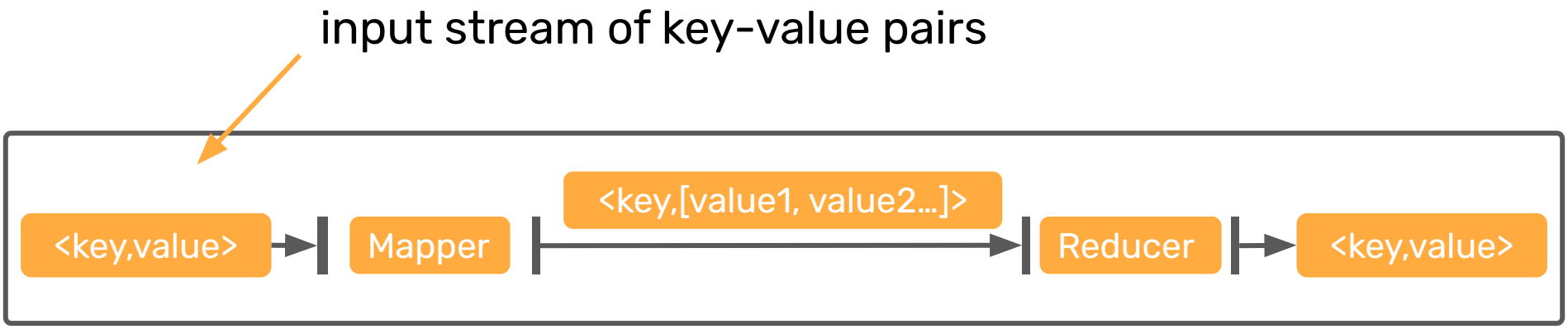
# MapReduce Streaming, решение задачи Line Count

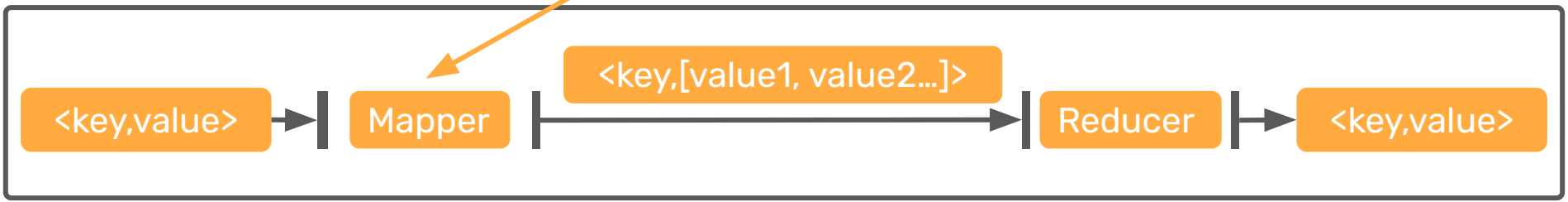
**Драль Алексей**, study@bigdatateam.org
CEO at BigData Team, https://bigdatateam.org
https://www.facebook.com/bigdatateam

# MapReduce Streaming

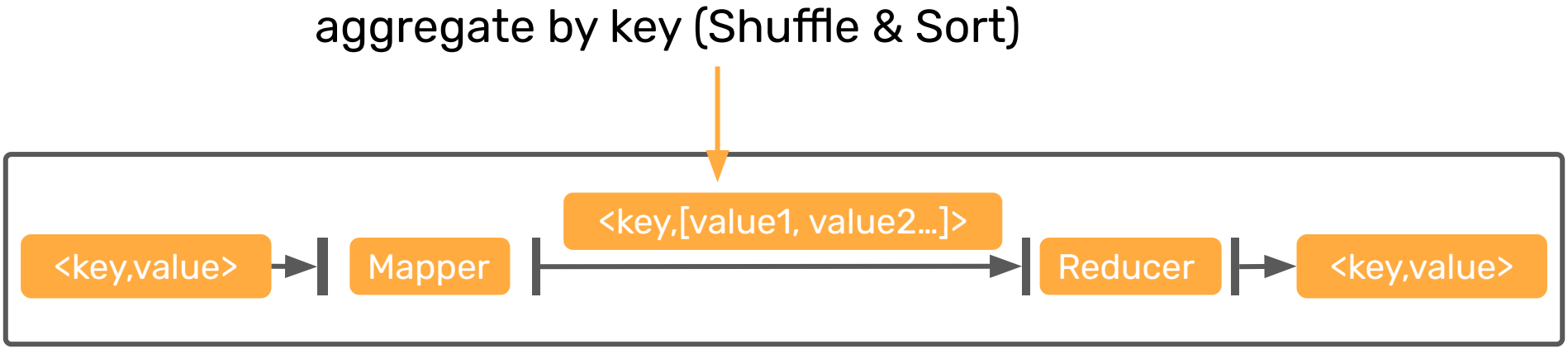


# MapReduce Streaming

map: (k_in, v_in) --> [(k_interm, v_interm), …]

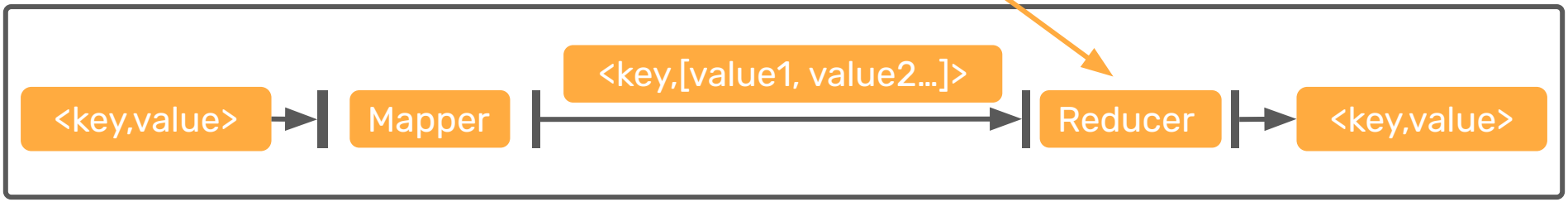


# MapReduce Streaming



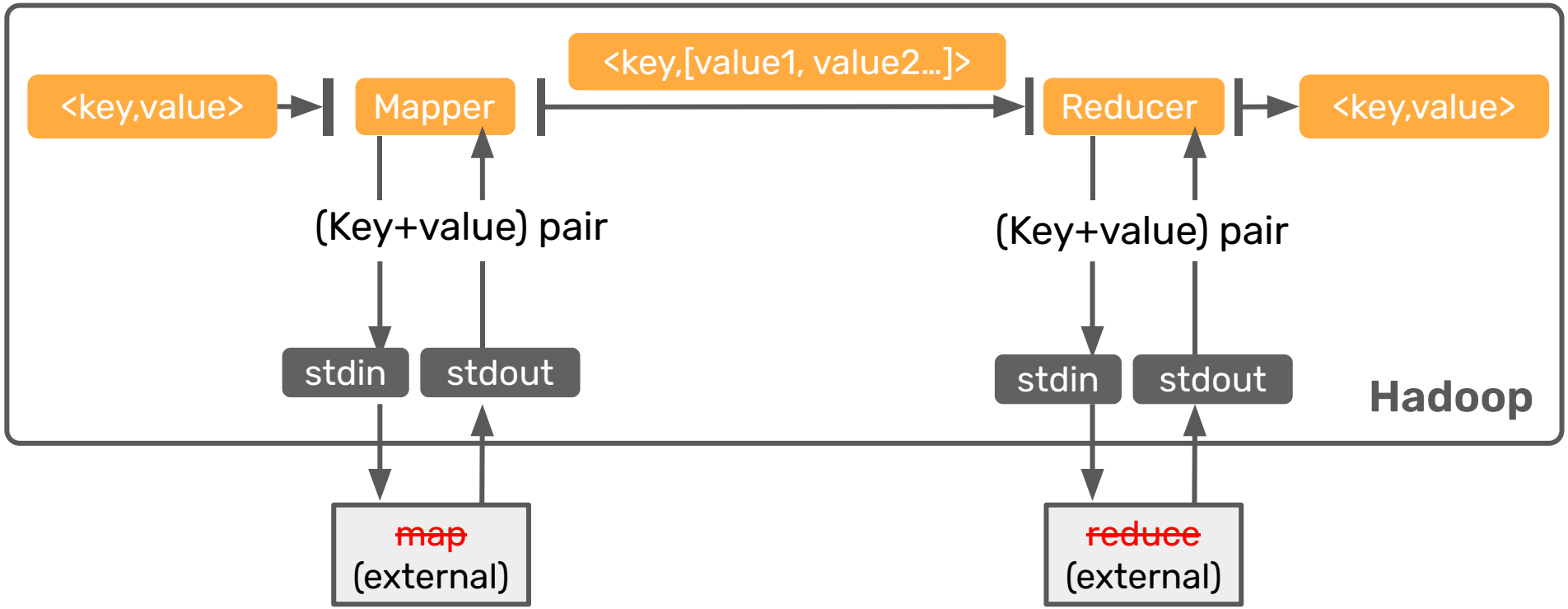
# MapReduce Streaming

reduce: (k_interm, [(v_interm, ...)] ) --> [(k_out, v_out), ...]



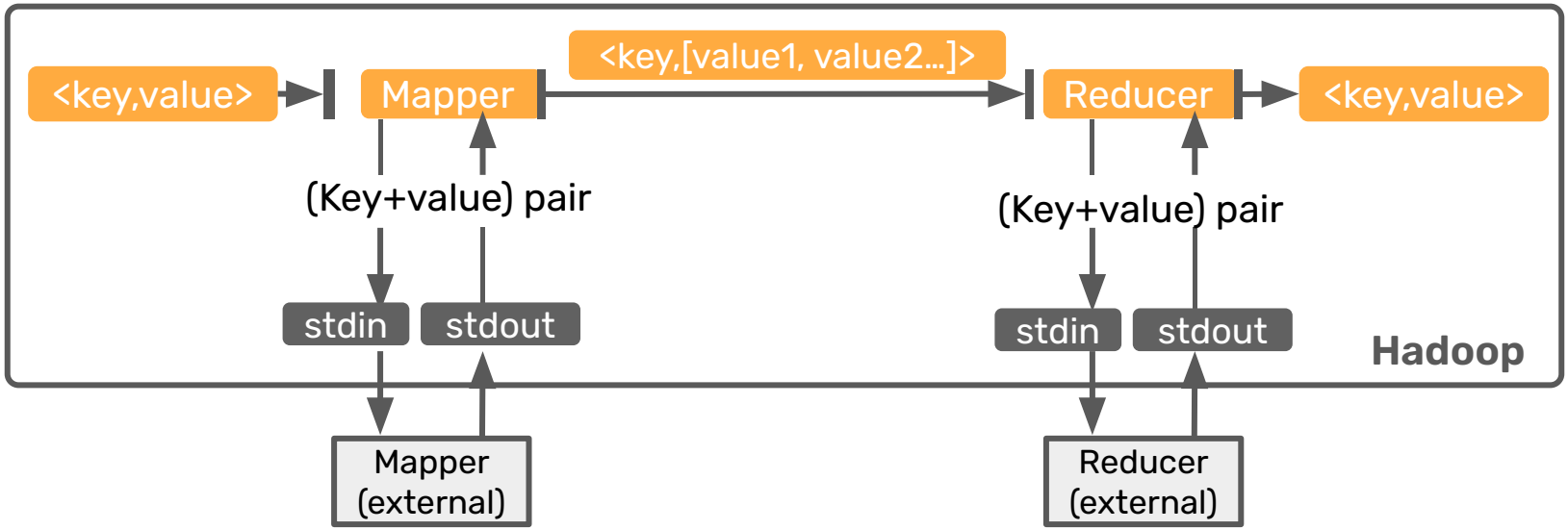
BIGDATA TEAM

# MapReduce Streaming



BIGDATA TEAM

# MapReduce Streaming




Mapper:

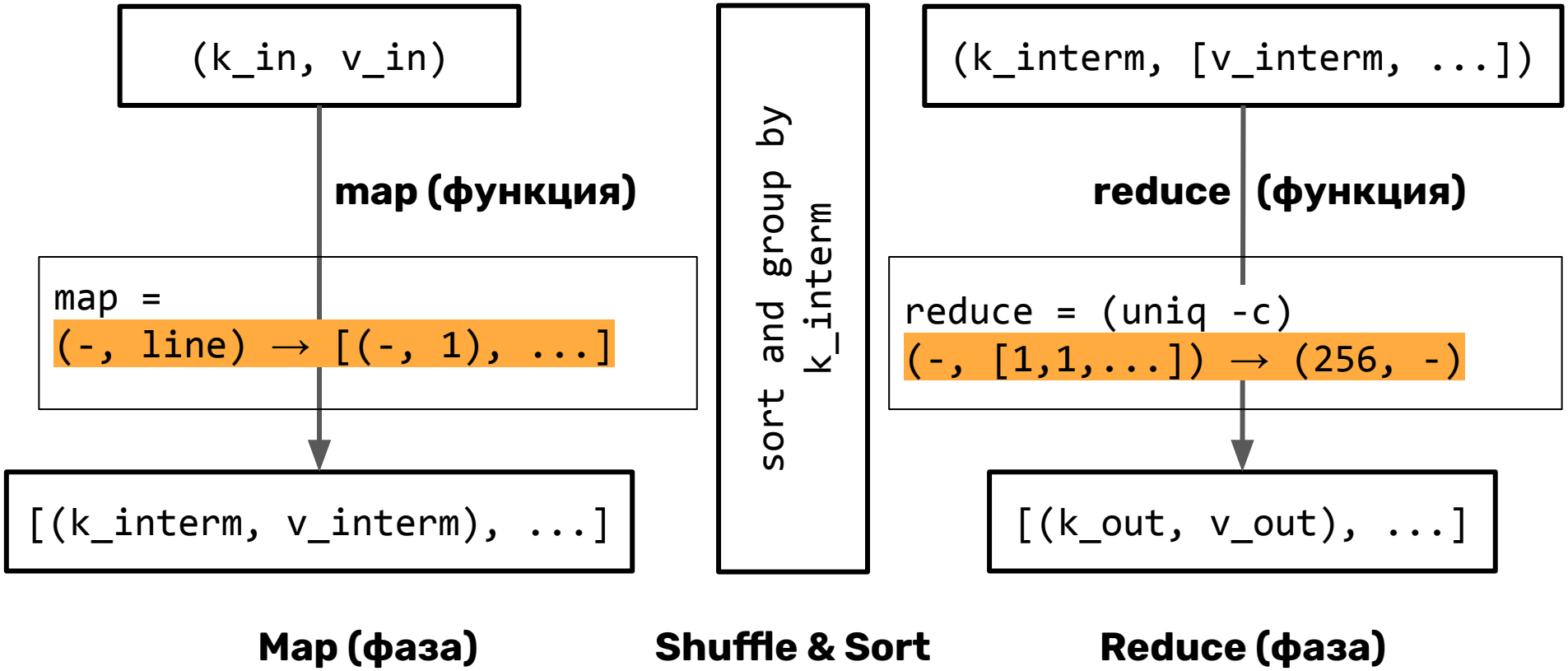
- Как данные читаем (input format)
- Как данные обрабатываем
- Как данные выводим (output format)

Тоже, что и Mapper, плюс:

- Как агрегируем по ключам отсортированные данные

BIGDATA TEAM

# Line Count



BIGDATA TEAM

# Постановка задачи Line Count



```
<article_id> <tab> <article_content>
```

run.sh

```
HADOOP_STREAMING_JAR=/path/to/hadoop-streaming.jar


yarn jar $HADOOP_STREAMING_JAR
```

run.sh

```
HADOOP_STREAMING_JAR=/path/to/hadoop-streaming.jar


yarn jar $HADOOP_STREAMING_JAR \


    -input /data/wiki/en_articles_part
```

run.sh

```
HADOOP_STREAMING_JAR=/path/to/hadoop-streaming.jar
OUT_DIR=my_hdfs_output

yarn jar $HADOOP_STREAMING_JAR \



    -input /data/wiki/en_articles_part \
    -output $OUT_DIR
```

run.sh

```
HADOOP_STREAMING_JAR=/path/to/hadoop-streaming.jar
OUT_DIR=my_hdfs_output

yarn jar $HADOOP_STREAMING_JAR \
    -mapper "wc -l" \

    -input /data/wiki/en_articles_part \
    -output $OUT_DIR
```

run.sh

```
HADOOP_STREAMING_JAR=/path/to/hadoop-streaming.jar
OUT_DIR=my_hdfs_output

yarn jar $HADOOP_STREAMING_JAR \
    -mapper "wc -l" \
    -numReduceTasks 0 \
    -input /data/wiki/en_articles_part \
    -output $OUT_DIR
```

```
$ hdfs dfs -ls my_hdfs_output
```

```
Found 3 items
-rw-r--r--   3 aadral hdfs    0 2021-02-15 18:38 my_hdfs_output/_SUCCESS
-rw-r--r--   3 aadral hdfs    6 2021-02-15 18:38 my_hdfs_output/part-00000
-rw-r--r--   3 aadral hdfs    5 2021-02-15 18:38 my_hdfs_output/part-00001
```

```
$ hdfs dfs -text my_hdfs_output/*
```

```
3624
476
```

```
$ hdfs dfs -ls my_hdfs_output
```

```
Found 3 items
-rw-r--r--   3 aadral hdfs    0 2021-02-15 18:38 my_hdfs_output/_SUCCESS
-rw-r--r--   3 aadral hdfs    6 2021-02-15 18:38 my_hdfs_output/part-00000
-rw-r--r--   3 aadral hdfs    5 2021-02-15 18:38 my_hdfs_output/part-00001
```

```
$ hdfs dfs -text my_hdfs_output/*
```

```
3624
476
```

```
$ hdfs dfs -ls my_hdfs_output
```

```
Found 3 items
-rw-r--r--   3 aadral hdfs    0 2021-02-15 18:38 my_hdfs_output/_SUCCESS
-rw-r--r--   3 aadral hdfs    6 2021-02-15 18:38 my_hdfs_output/part-00000
-rw-r--r--   3 aadral hdfs    5 2021-02-15 18:38 my_hdfs_output/part-00001
```

```
$ hdfs dfs -text my_hdfs_output/*
```

```
3624
476
```

```
$ hdfs dfs -text my_hdfs_output/*
```

```
3624
476
```

```
$ hdfs dfs -ls -h /data/wiki/en_articles_part
```

```
Found 1 items
-rw-r--r--   3 hdfs hdfs      73.3 M 2020-03-12 21:03
/data/wiki/en_articles_part/articles-part
```

```
$ ./run.sh
```

```
...
ERROR streaming.StreamJob: Error Launching job : Output directory
hdfs://brain-master.bigdatateam.org:8020/user/aadral/my_hdfs_outpu
t already exists
Streaming Command Failed!
```

run.sh

```
HADOOP_STREAMING_JAR=/path/to/hadoop-streaming.jar
OUT_DIR=my_hdfs_output

hdfs dfs -rm -r $OUT_DIR

yarn jar $HADOOP_STREAMING_JAR \
    -mapper "wc -l" \

    -numReduceTasks 0 \
    -input /data/wiki/en_articles_part \
    -output $OUT_DIR
```

run.sh

```
HADOOP_STREAMING_JAR=/path/to/hadoop-streaming.jar
OUT_DIR=my_hdfs_output

hdfs dfs -rm -r $OUT_DIR

yarn jar $HADOOP_STREAMING_JAR \
    -mapper "wc -l" \

    -numReduceTasks 1 \
    -input /data/wiki/en_articles_part \
    -output $OUT_DIR
```

run.sh

```
HADOOP_STREAMING_JAR=/path/to/hadoop-streaming.jar
OUT_DIR=my_hdfs_output

hdfs dfs -rm -r $OUT_DIR

yarn jar $HADOOP_STREAMING_JAR \
    -mapper "wc -l" \
    -reducer "awk '{line_count += \$1} END { print line_count }'" \
    -numReduceTasks 1 \
    -input /data/wiki/en_articles_part \
    -output $OUT_DIR
```

```
$ hdfs dfs -ls my_hdfs_output
```

```
Found 2 items
-rw-r--r--   3 aadral hdfs    0 2021-02-17 11:22 my_hdfs_output/_SUCCESS
-rw-r--r--   3 aadral hdfs    6 2021-02-17 11:22 my_hdfs_output/part-00000
```

```
$ hdfs dfs -text my_hdfs_output/*
```

```
4100
```

reducer.sh

```
#!/usr/bin/env bash
awk '{line_count += $1} END { print line_count }'
```

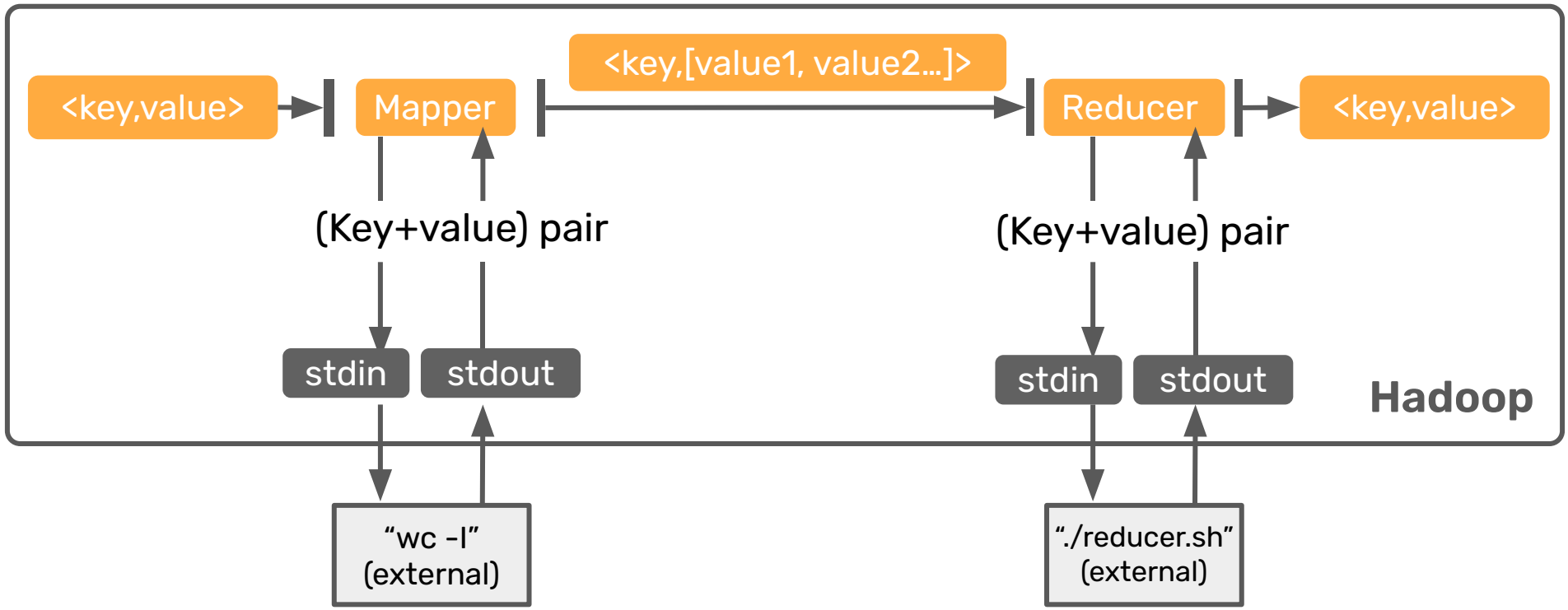
run.sh

```
HADOOP_STREAMING_JAR=/path/to/hadoop-streaming.jar
OUT_DIR=my_hdfs_output

hdfs dfs -rm -r $OUT_DIR

yarn jar $HADOOP_STREAMING_JAR \
    -files reducer.sh \
    -mapper "wc -l" \
    -reducer "./reducer.sh" \
    -numReduceTasks 1 \
    -input /data/wiki/en_articles_part \
    -output $OUT_DIR
```

# На заметку

```
$ chmod a+x some_script.[py,sh,...]
```

# Line Count


<key,value>
Mapper
<key,[value1, value2…]>
Reducer
<key,value>
(Key+value) pair
(Key+value) pair
stdin
stdout
stdin
stdout
Hadoop
“wc -l”
(external)
“./reducer.sh”
(external)

# Резюме

Теперь вы можете:

# Резюме

Теперь вы можете:

- Перечислить зоны ответственности Java и Python разработчиков MapReduce приложений

# Резюме

Теперь вы можете:

- Перечислить зоны ответственности Java и Python разработчиков MapReduce приложений
- Использовать -files, -mapper, -reducer, -numReduceTasks, -input, -output

# Резюме

Теперь вы можете:

- Перечислить зоны ответственности Java и Python разработчиков MapReduce приложений
- Использовать -files, -mapper, -reducer, -numReduceTasks, -input, -output