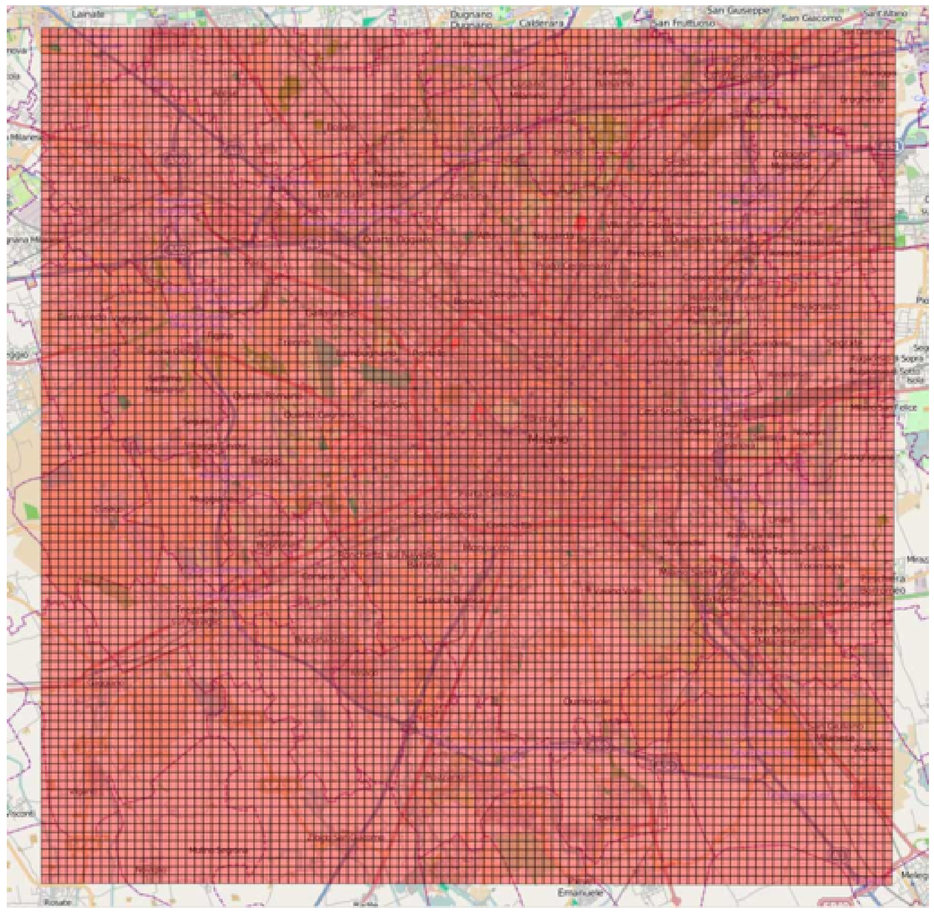
Yandex

Telecommunications Analytics

Tabular Data, KeyFieldSelection

Telecommunications — SMS, Call, Internet — MI

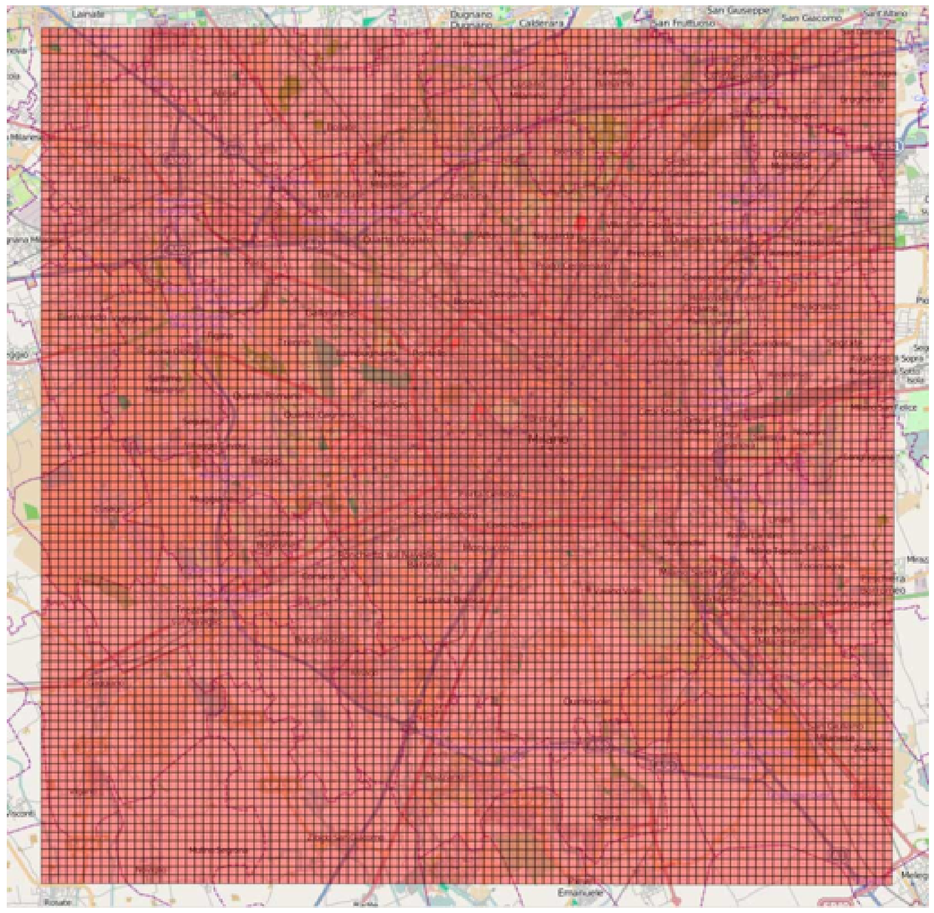


Milano Grid

- Square ID
- Time Interval
- Country Code
- SMS-in Activity
- SMS-out Activity
- Call-in Activity
- Call-out Activity
- Internet Traffic Activity

Schema

Telecommunications — SMS, Call, Internet — MI



Milano Grid

- Square ID
- Time Interval
- Country Code
- SMS-in Activity
- SMS-out Activity
- Call-in Activity
- Call-out Activity
- Internet Traffic Activity

Schema

```
value_column_index = os.environ["column_index"]

geojson = json.load(open("milano-grid.geojson"))
grid = load_grid(geojson)
for line in sys.stdin:
    square_id, aggregate = line.split("\t", 1)
    square_id = int(square_id)
    stat = aggregate.split("\t")[value_column_index]
    if stat:
       print(grid[square_id], stat, sep="\t")
```

yandex — bash

```
yarn jar $HADOOP_STREAMING_JAR \
        -D mapreduce.fieldsel.map.output.key.value.fields.spec=0:3,5- \
        -mapper org.apache.hadoop.mapred.lib.FieldSelectionMapReduce \
        -numReduceTasks 0 \
        -input sms-call-internet-mi-2013-11-01.txt \
        -output telecom-joins
```

yandex — bash

```
yarn jar $HADOOP_STREAMING_JAR \
        -D mapreduce.fieldsel.map.output.key.value.fields.spec=0:3,5- \
        -mapper org.apache.hadoop.mapred.lib.FieldSelectionMapReduce \
        -numReduceTasks 0 \
        -input sms-call-internet-mi-2013-11-01.txt \
        -output telecom-joins
```

key index: 0 (Square ID)

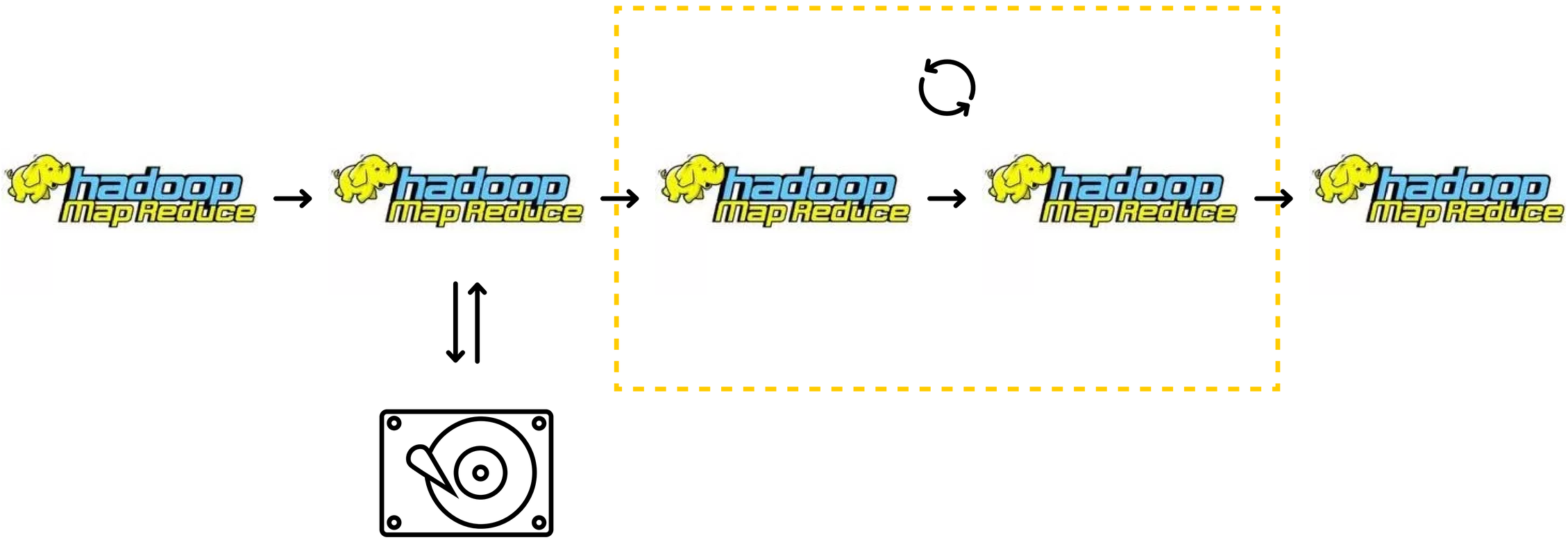
yandex — bash

```
yarn jar $HADOOP_STREAMING_JAR \
        -D mapreduce.fieldsel.map.output.key.value.fields.spec=0:3,5- \
        -mapper org.apache.hadoop.mapred.lib.FieldSelectionMapReduce \
        -numReduceTasks 0 \
        -input sms-call-internet-mi-2013-11-01.txt \
        -output telecom-joins
```

value index: 3,5-

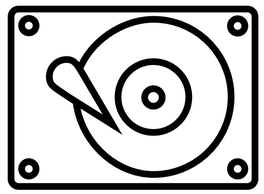
```
$ head sms-call-internet-mi-2013-11-01.txt
11383260400000    0     0.08136262351125882
1  1383260400000  39   0.1418642547024292 0.15678700503902460.1609379369170182
0.05227484852857320 11.028366381681026
1  1383261000000  0  0.1365878227582310 0.02730046487718618
1  1383261000000  33  0.02613742426428660 2
1  1383261000000  39   0.2784520774606602 0.1199257201417413 0.1887771729145041
0.1336374720398320 11.100963451409388
...

$ head telecom-joins/part-00000
1  0.08136262351125882
1  0.1418642547024292 0.1609379369170182 0.05227484852857320 11.028366381681026
1  0.1365878227582310 0.02730046487718618
1  0.02613742426428660 2
1  0.2784520774606602 0.18877717291450410.1336374720398320 11.100963451409388
...
```

Job Chaining



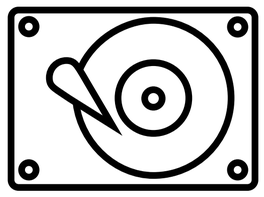
Job Chaining

FieldSelection **MapSide Join**



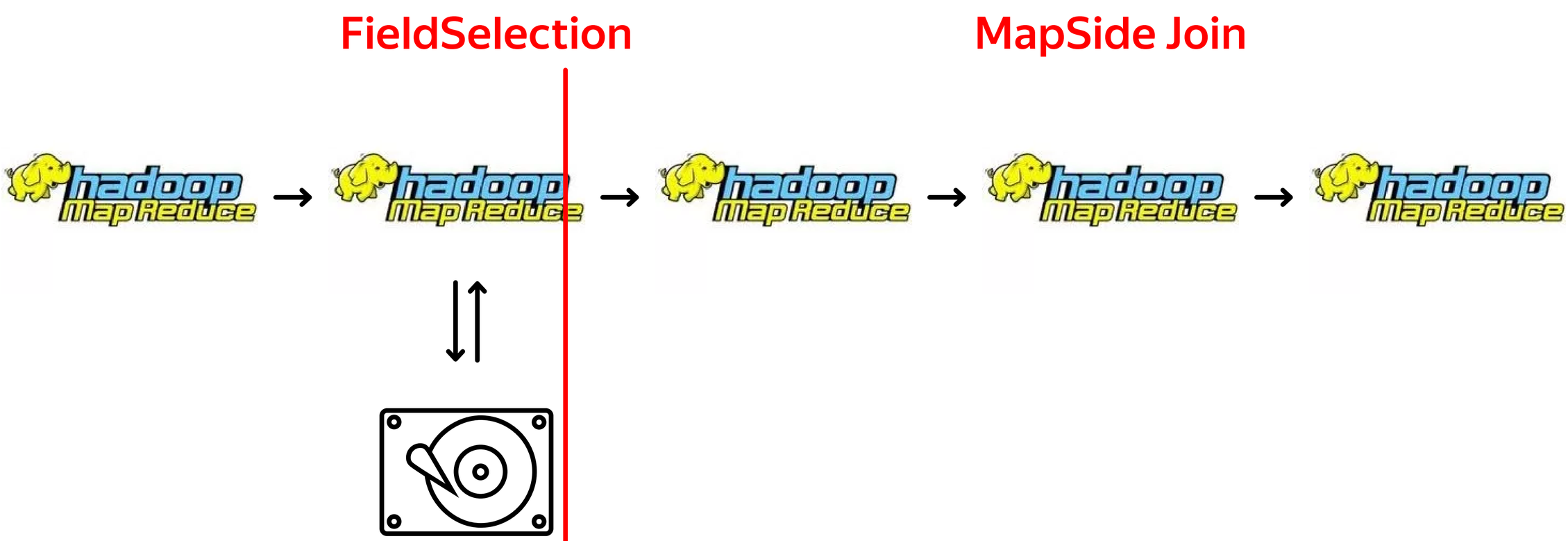
Job Chaining

FieldSelection **MapSide Join**



HADOOPY

MrJob




```
yarn jar …FieldSelectionMapReduce…
application_return_code=$?

if [ $application_return_code != "0" ]
then
    echo "FieldSelection phase was NOT successful"
    exit $application_return_code
fi
```

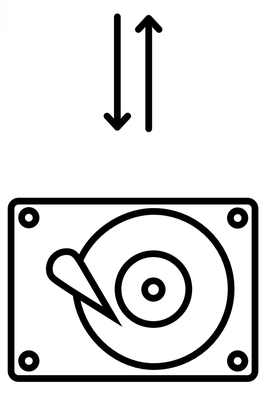
FieldSelection

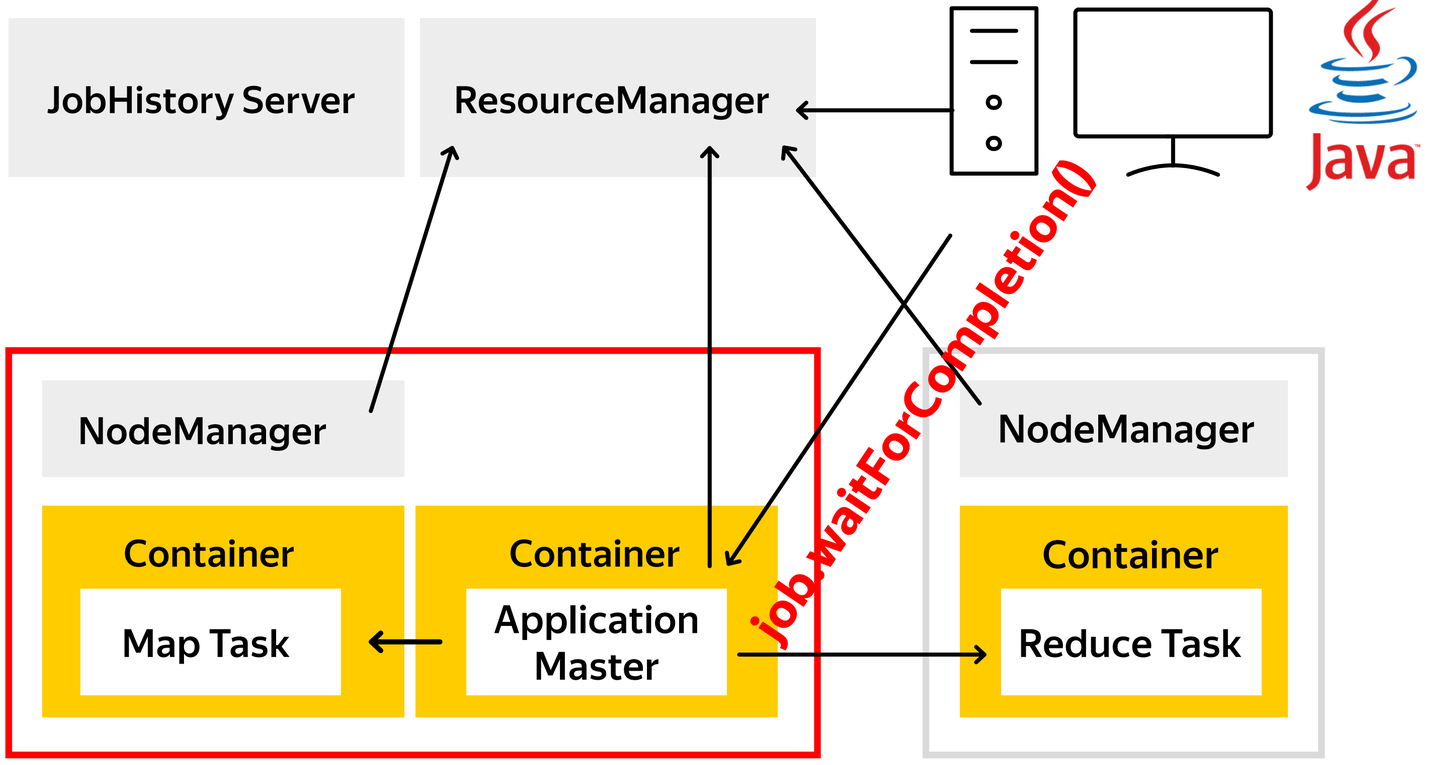
MapSide Join

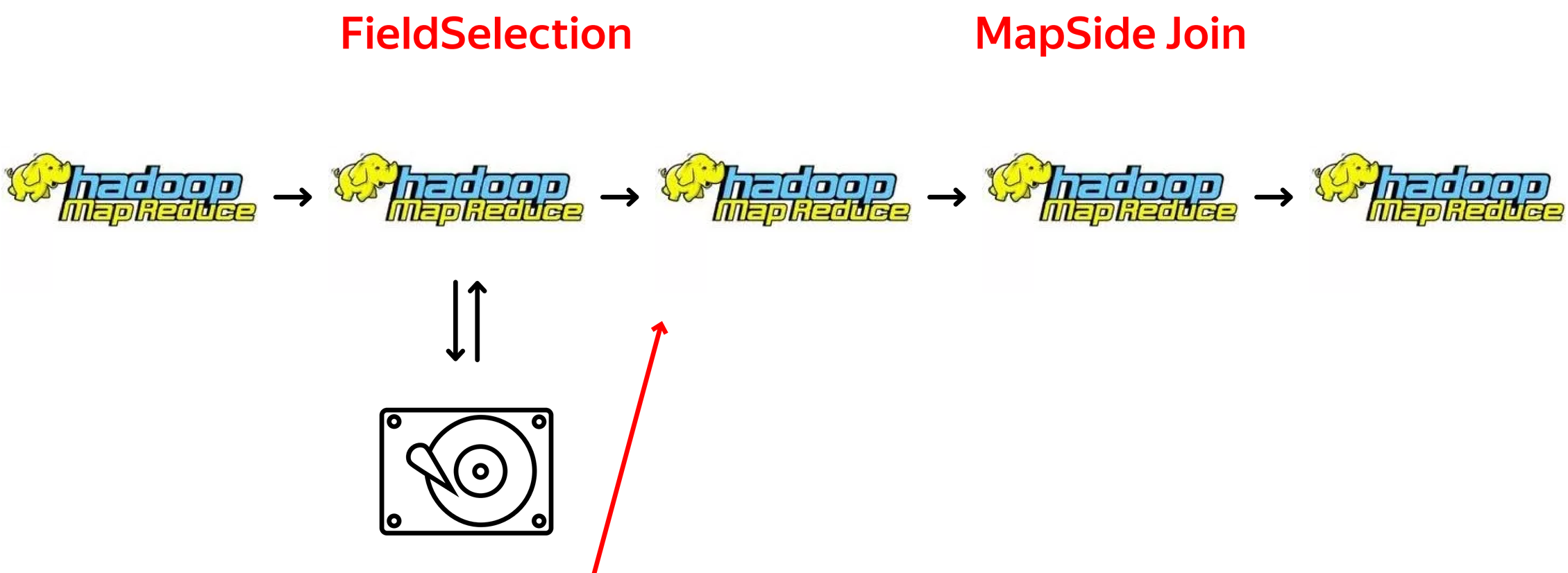
```
yarn jar …FieldSelectionMapReduce…
application_return_code=$?

if [ $application_return_code != "0" ]
then
    echo "FieldSelection phase was NOT successful"
    exit $application_return_code
fi
```

FieldSelection

MapSide Join








```
$ yarn application -list
          Application-Id    Application-Name    Application-Type        User
Queue           State         Final-State       Progress                Tracking-URL

application_1491639197451_0559  select age from (select a.Age as age,...
1000(Stage-1)     MAPREDUCE     s201701  root.users.s201701             RUNNING
UNDEFINED          90.74% http://virtual-node1.atp-fivt.org:45272
…
application_1491639197451_0565  streamjob7453411855907589438.jar
MAPREDUCE      adral    root.users.adral             RUNNING           UNDEFINED
5%  http://virtual-node3.atp-fivt.org:46751
…
```

$ yarn application -status **application_1491639197451_0565**

Application Report :
Application-Id : application_1491639197451_0565
Application-Name : streamjob7453411855907589438.jar
Application-Type : MAPREDUCE
User : adral
Queue : root.users.adral
Start-Time : 1492272089050
Finish-Time : 0
Progress : 41%
State : RUNNING
Final-State : UNDEFINED
Tracking-URL : http://virtual-node3.atp-fivt.org:46751
RPC Port : 42533
AM Host : virtual-node3.atp-fivt.org
Aggregate Resource Allocation : 244872 MB-seconds, 80 vcore-seconds
Log Aggregation Status : NOT_START
Diagnostics:

$ yarn application -status **application_1491639197451_0565**

Application Report :
Application-Id : application_1491639197451_0565
Application-Name : streamjob7453411855907589438.jar
Application-Type : MAPREDUCE
User : adral
Queue : root.users.adral
Start-Time : 1492272089050
Finish-Time : 1492272181541
Progress : 100%
State : FINISHED
Final-State : SUCCEEDED
Tracking-URL : http://virtual-master.atp-fivt.org:19888/jobhistory/job/job_1491639197451_0565
RPC Port : 42533
AM Host : virtual-node3.atp-fivt.org
Aggregate Resource Allocation : 544105 MB-seconds, 176 vcore-seconds
Log Aggregation Status : SUCCEEDED
Diagnostics:

$ hdfs dfs -ls telecom-joins/field-selection

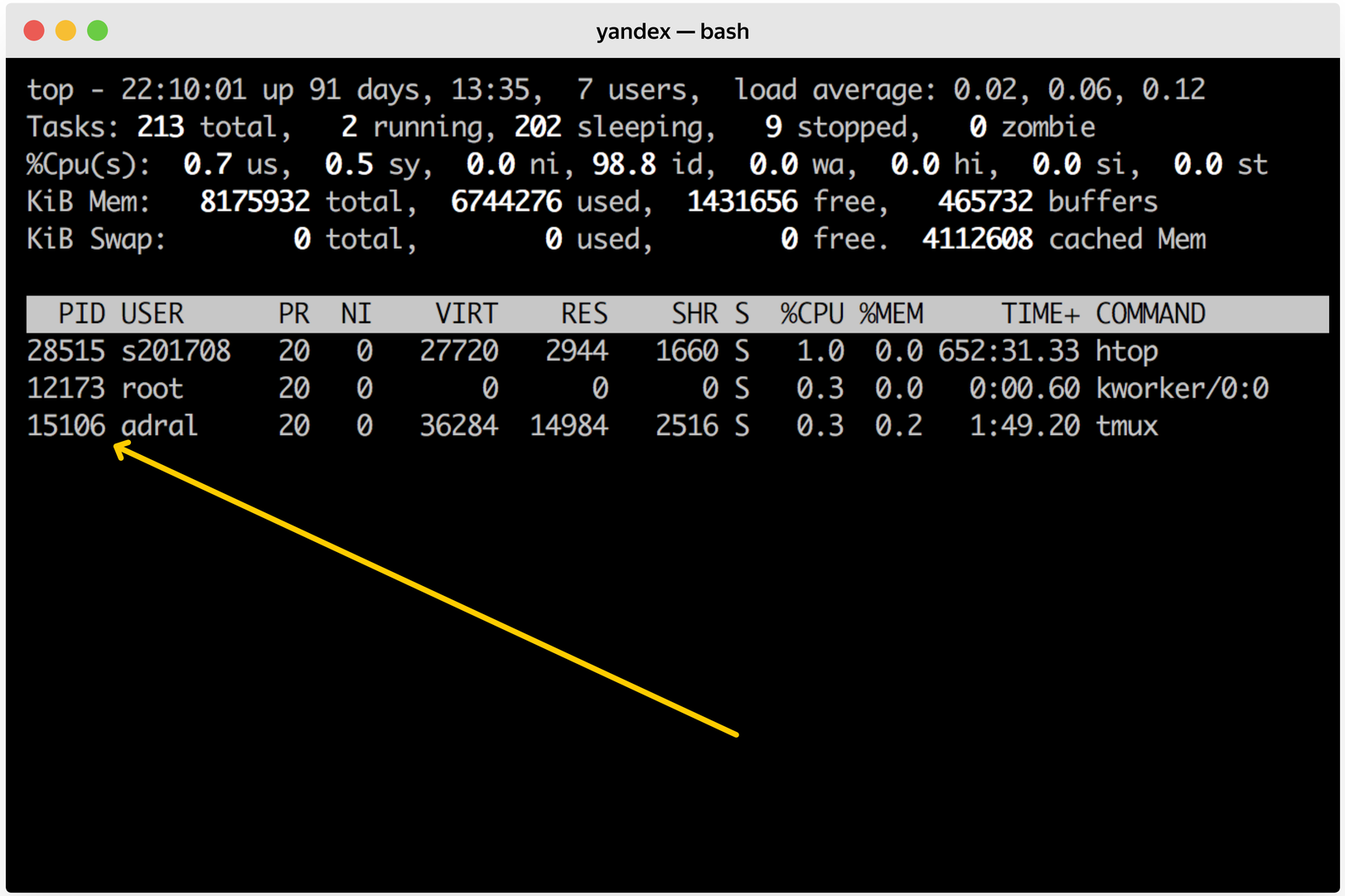
Found 11 items
-rw-r--r-- 3 ... 0 2017-04-15 **18:30** telecom-joins/field-selection/**_SUCCESS**
-rw-r--r-- 3 ... 8516829 2017-04-15 **18:28** telecom-joins/field-selection/part-00000
-rw-r--r-- 3 ... 8555052 2017-04-15 18:29 telecom-joins/field-selection/part-00001
-rw-r--r-- 3 ... 8380546 2017-04-15 18:29 telecom-joins/field-selection/part-00002
-rw-r--r-- 3 ... 8546802 2017-04-15 18:29 telecom-joins/field-selection/part-00003
-rw-r--r-- 3 ... 8293343 2017-04-15 18:29 telecom-joins/field-selection/part-00004
-rw-r--r-- 3 ... 8348375 2017-04-15 18:29 telecom-joins/field-selection/part-00005
...

$ hdfs dfs -ls telecom-joins/field-selection

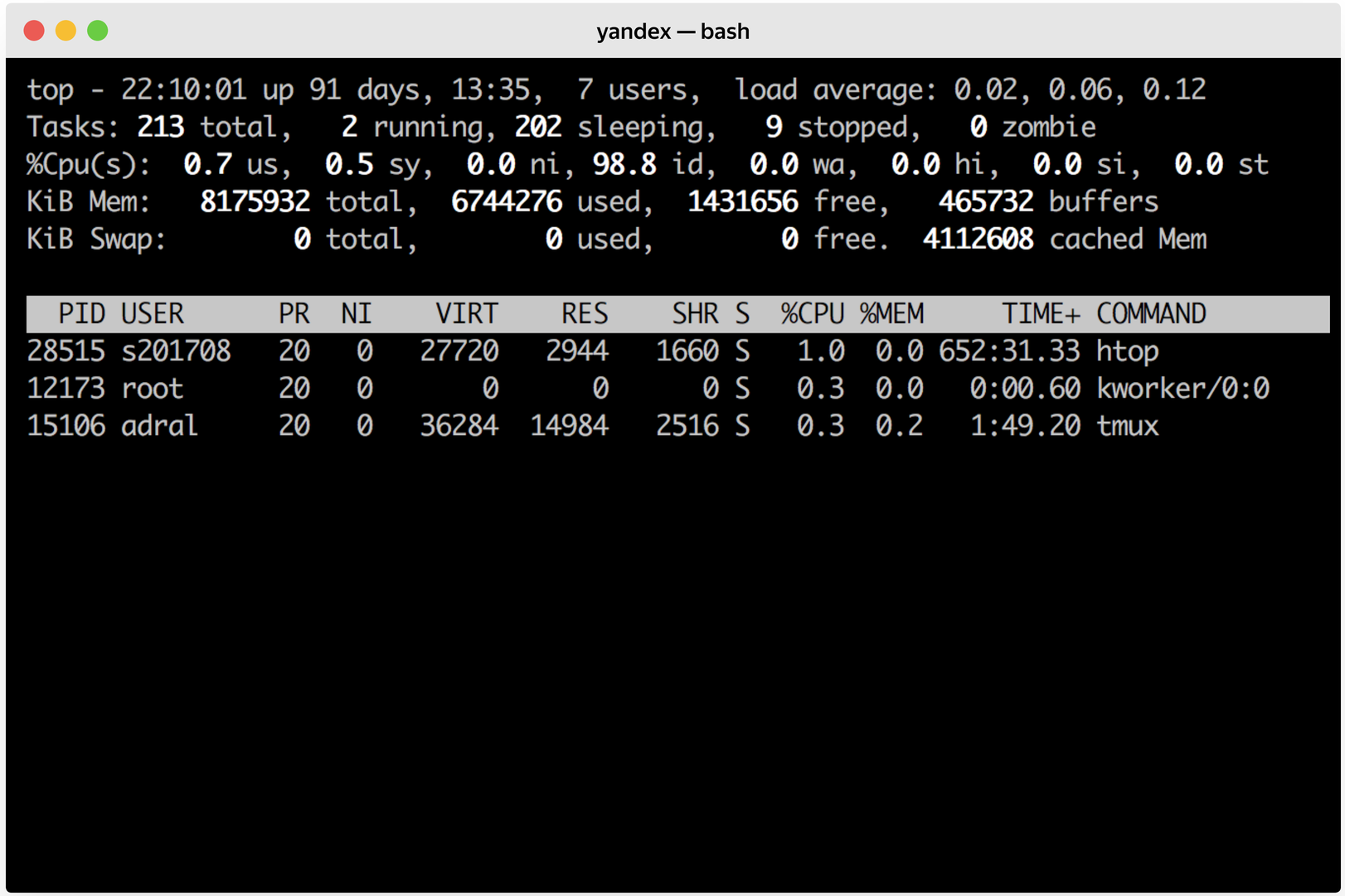
```
Found 11 items
-rw-r--r--   3 ...       0 2017-04-15 18:30 telecom-joins/field-selection/_SUCCESS
-rw-r--r--   3 ...  8516829 2017-04-15 18:28 telecom-joins/field-selection/part-00000
-rw-r--r--   3 ...  8555052 2017-04-15 18:29 telecom-joins/field-selection/part-00001
-rw-r--r--   3 ...  8380546 2017-04-15 18:29 telecom-joins/field-selection/part-00002
-rw-r--r--   3 ...  8546802 2017-04-15 18:29 telecom-joins/field-selection/part-00003
-rw-r--r--   3 ...  8293343 2017-04-15 18:29 telecom-joins/field-selection/part-00004
-rw-r--r--   3 ...  8348375 2017-04-15 18:29 telecom-joins/field-selection/part-00005
...
```

hdfs dfs -test -e telecom-joins/field-selection/_SUCCESS

$ top


yandex — bash
top - 22:10:01 up 91 days, 13:35, 7 users, load average: 0.02, 0.06, 0.12
Tasks: 213 total, 2 running, 202 sleeping, 9 stopped, 0 zombie
%Cpu(s): 0.7 us, 0.5 sy, 0.0 ni, 98.8 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem: 8175932 total, 6744276 used, 1431656 free, 465732 buffers
KiB Swap: 0 total, 0 used, 0 free. 4112608 cached Mem
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
28515 s201708 20 0 27720 2944 1660 S 1.0 0.0 652:31.33 htop
12173 root 20 0 0 0 0 S 0.3 0.0 0:00.60 kworker/0:0
15106 adral 20 0 36284 14984 2516 S 0.3 0.2 1:49.20 tmux

$ top




```
PIDFILE=job_chain_with_success.pid
[ -e $PIDFILE ] && kill -0 `cat $PIDFILE` && echo "already working (pid=`cat $PIDFILE`)" && exit 0
echo $$ > $PIDFILE
```

HDFS client

HDFS client

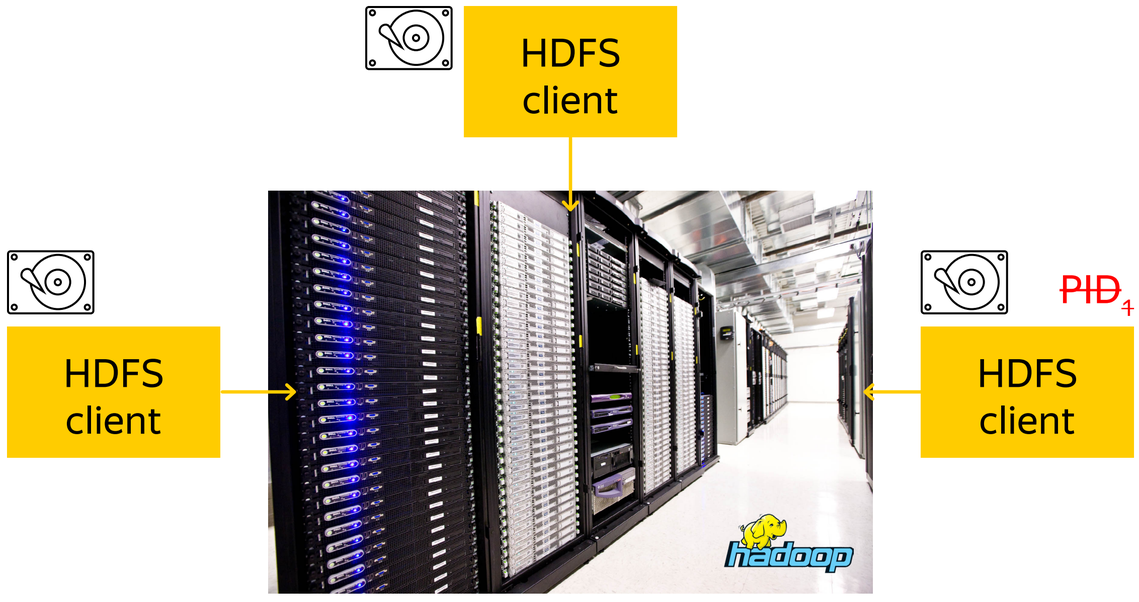
HDFS client

HDFS client

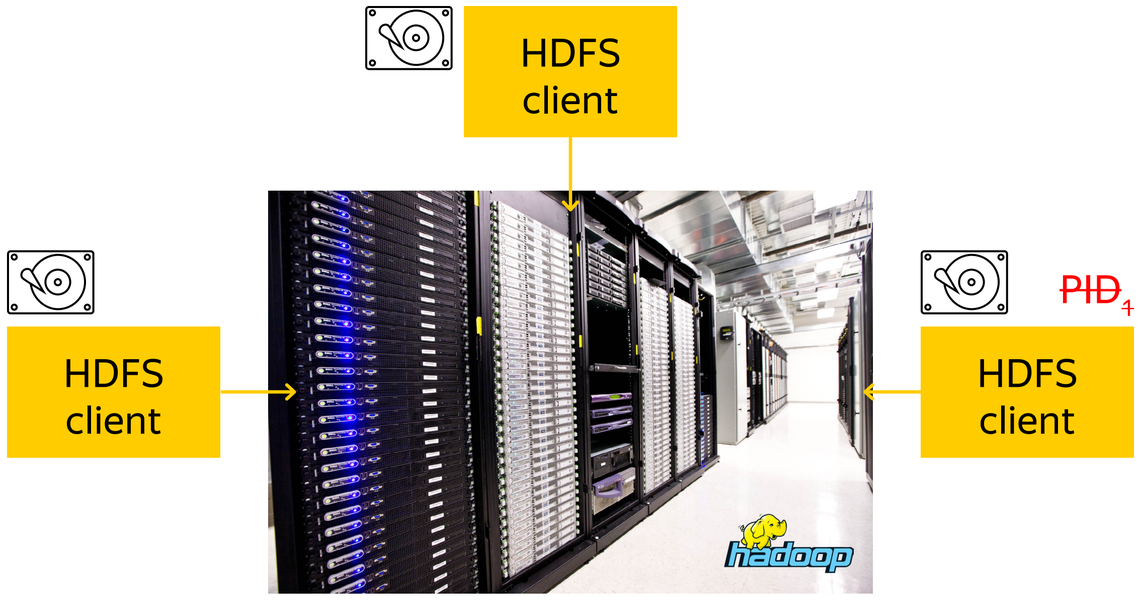
HDFS client

HDFS client

~~PID 1~~

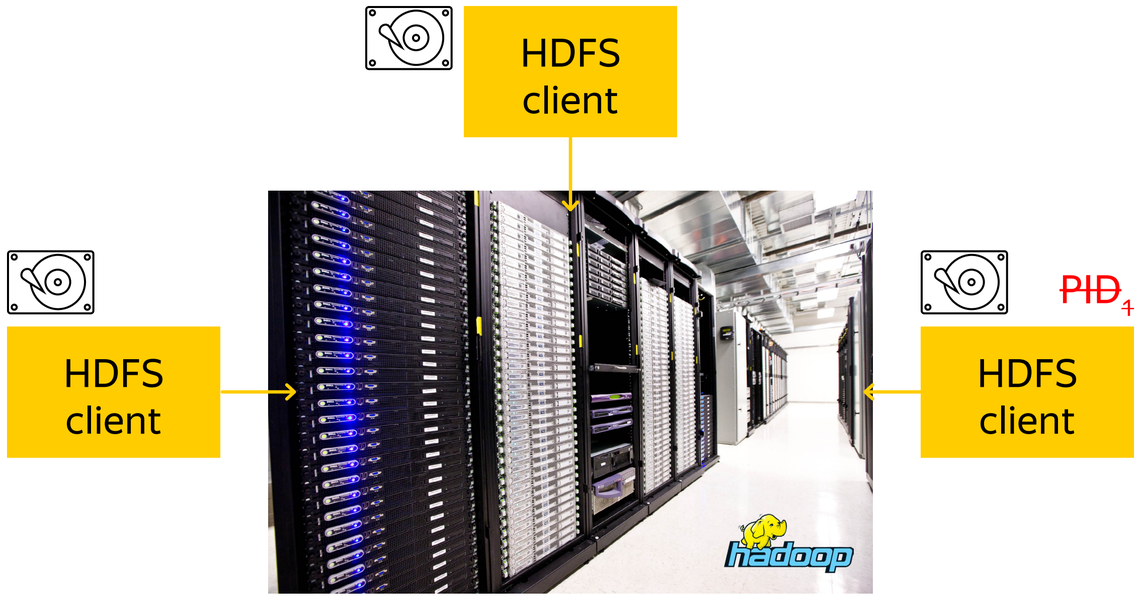


- HDFS
- Hadoop MapReduce
- ...
- Locking Service



example:
hdfs dfs -put /lock/file

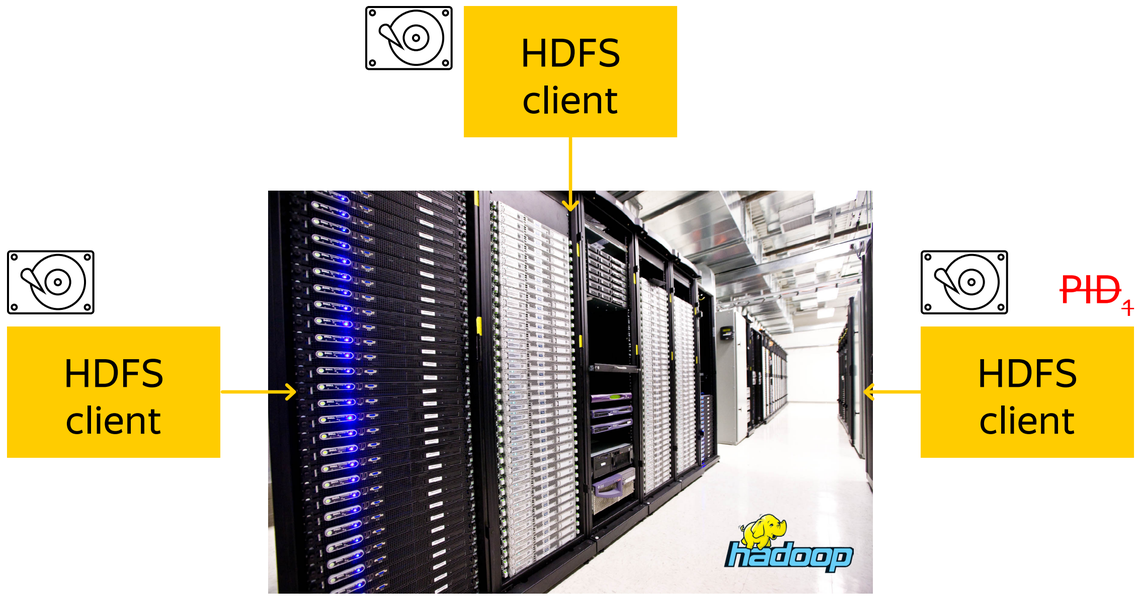
- HDFS
- Hadoop MapReduce
- ...
- Locking Service



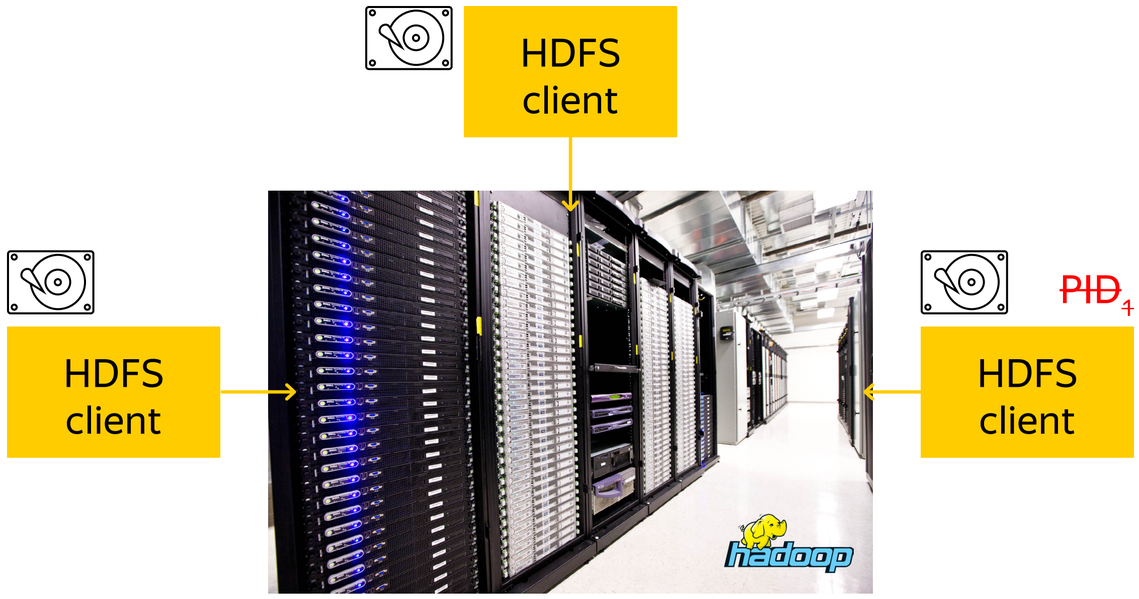
example:
hdfs dfs -put /lock/file

http://stackoverflow.com/a/11205414

- HDFS
- Hadoop MapReduce
- ...
- Locking Service

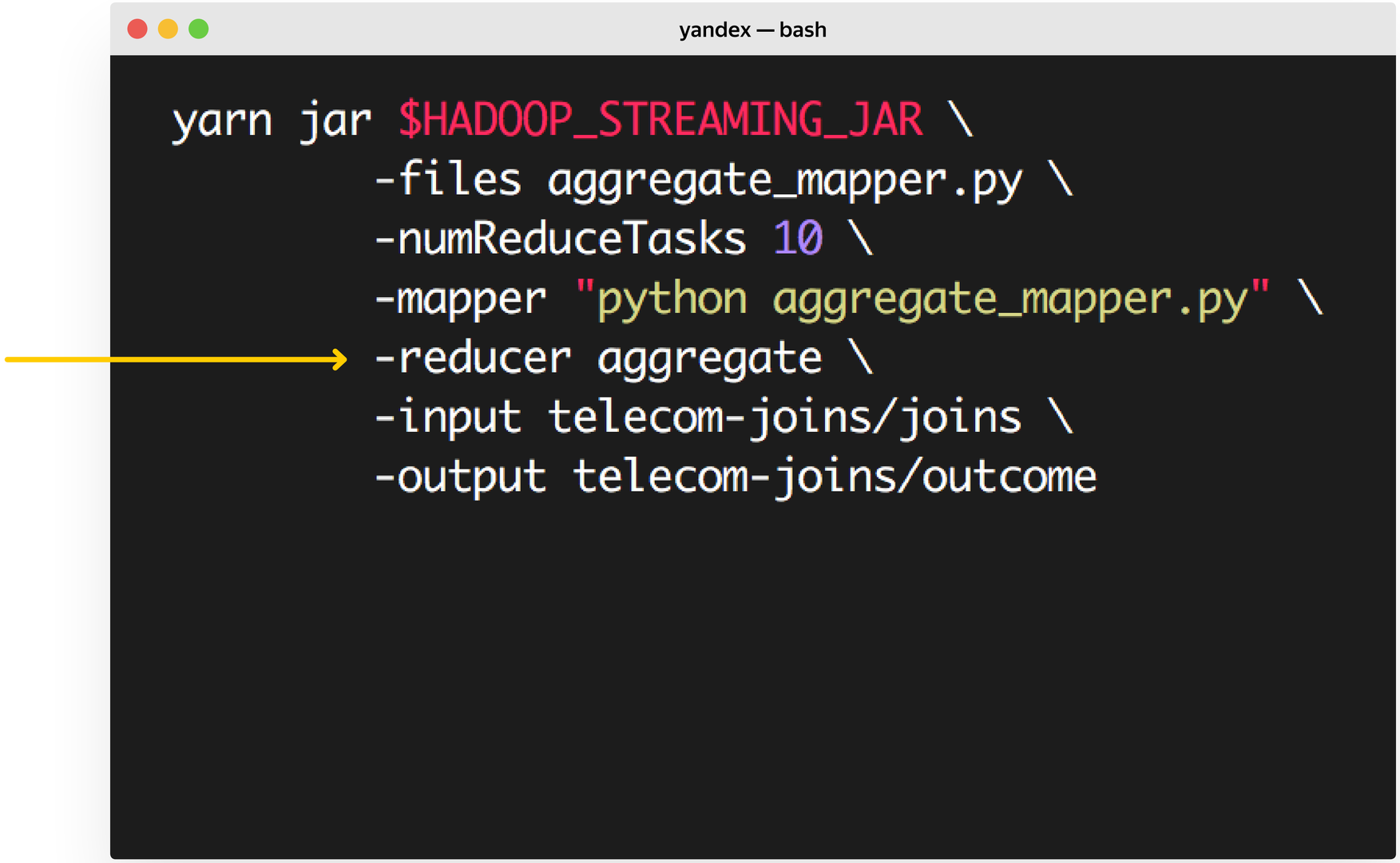


- HDFS
- Hadoop MapReduce
- ...
- Locking Service



- HDFS
- Hadoop MapReduce
- …
- Locking Service

…


yandex — bash
yarn jar $HADOOP_STREAMING_JAR \
-files aggregate_mapper.py \
-numReduceTasks 10 \
-mapper "python aggregate_mapper.py" \
-reducer aggregate \
-input telecom-joins/joins \
-output telecom-joins/outcome

yandex — bash

```
yarn jar $HADOOP_STREAMING_JAR \
        -files aggregate_mapper.py \
        -numReduceTasks 10 \
        -mapper "python aggregate_mapper.py" \
        -reducer aggregate \
        -input telecom-joins/joins \
        -output telecom-joins/outcome
```

```
from __future__ import print_function
import sys


if __name__ == "__main__":
    for line in sys.stdin:
        key, value = line.rstrip("\n").split("\t", 1)
        print("DoubleValueSum:{}".format(key), value, sep="\t")
```

yandex — bash

```
yarn jar $HADOOP_STREAMING_JAR \
        -files aggregate_mapper.py \
        -numReduceTasks 10 \
        -mapper "python aggregate_mapper.py" \
        -reducer aggregate \
        -input telecom-joins/joins \
        -output telecom-joins/outcome
```

```
from __future__ import print_function
import sys


if __name__ == "__main__":
    for line in sys.stdin:
        key, value = line.rstrip("\n").split("\t", 1)
        print("DoubleValueSum:{}".format(key), value, sep="\t")
```

South 1638476.7866161505
North 2967910.544348439

Summary

Summary

- you can **process** tabular data with FieldSelectionMapReduce

Summary

- you can **process** tabular data with FieldSelectionMapReduce
- you can **break down** solution into multiple MapReduce jobs and chain them

Summary

- you can **process** tabular data with FieldSelectionMapReduce
- you can **break down** solution into multiple MapReduce jobs and chain them
- you can **validate** MapReduce job successfulness with _SUCCESS file

Summary

- you can **process** tabular data with FieldSelectionMapReduce
- you can **break down** solution into multiple MapReduce jobs and chain them
- you can **validate** MapReduce job successfulness with _SUCCESS file
- you can **use** aggregate streaming package to count simple statistics

BigDATAteam