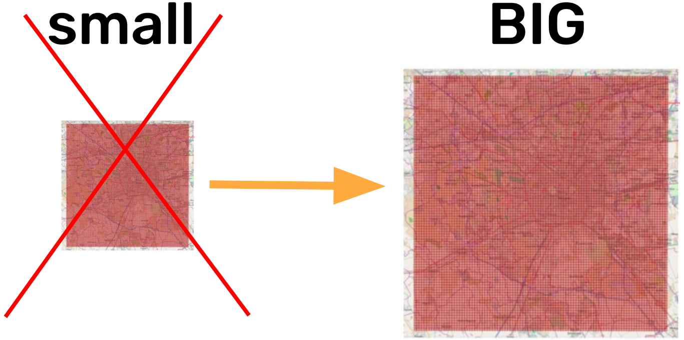
# Reduce-Side Join

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

# Увеличение покрытия

## BIG

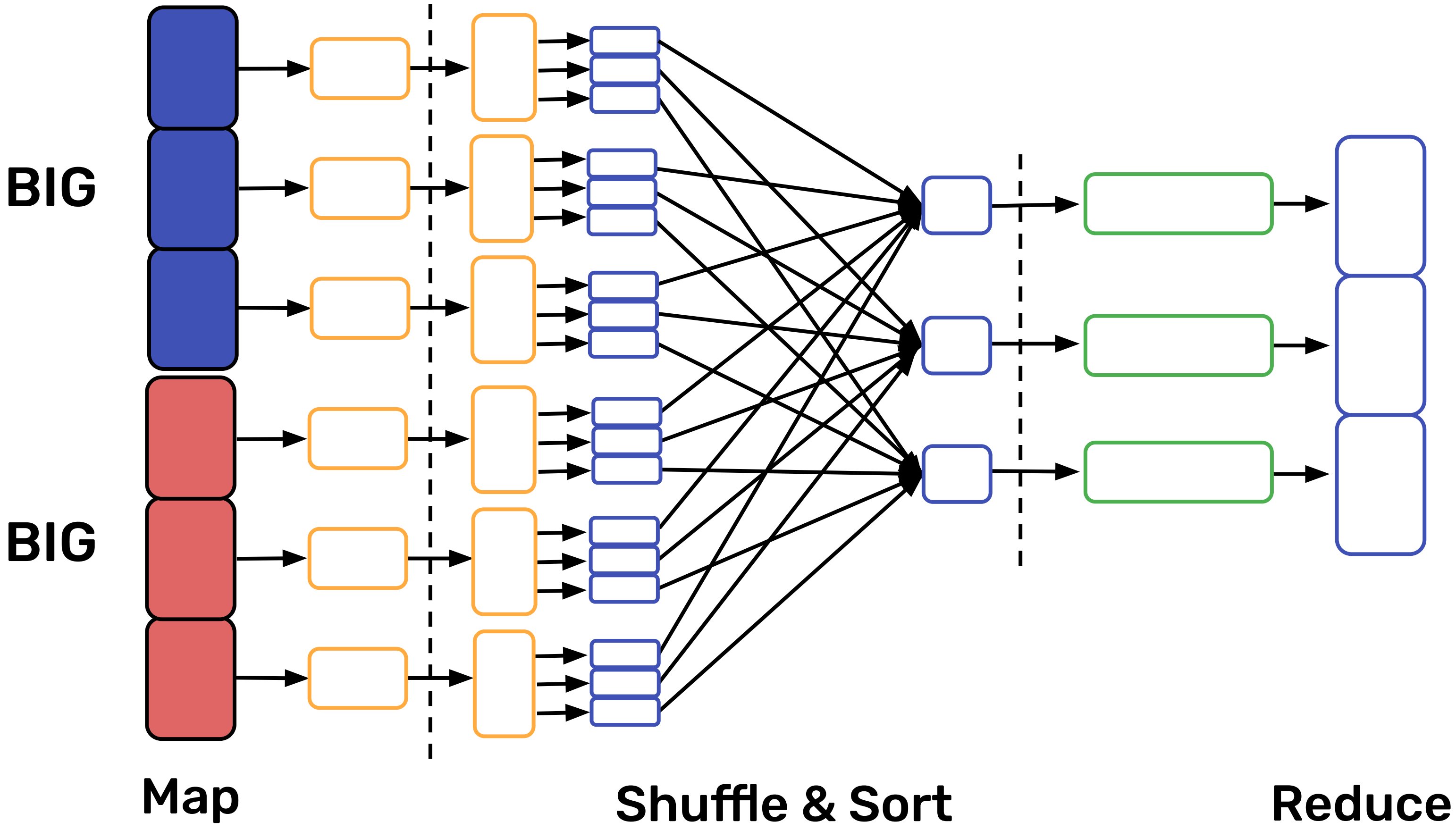
- Square ID
- Time Interval
- Country Code
- SMS-in Activity
- SMS-out Activity
- Call-in Activity
- Call-out Activity
- Internet Traffic Activity



```
1  1383260400000 0 0.08136262351125882
1  1383260400000 39 0.14186425470242922
0.1567870050390246 0.16093793691701822
0.052274848528573205 11.028366381681026
1  1383261000000 0 0.13658782275823106
0.02730046487718618
1  13832610000000 33
0.026137424264286602
```

```
{‘type’: ‘Polygon’, ‘coordinates’:
[[[9.0114910478323, 45.35880131440966],
[9.014491488013135, 45.35880097314403],
[9.0144909480813, 45.35668565341486],
[9.011490619692509,
45.356685994655464], [9.0114910478323,
45.35880131440966]]]}
...
```

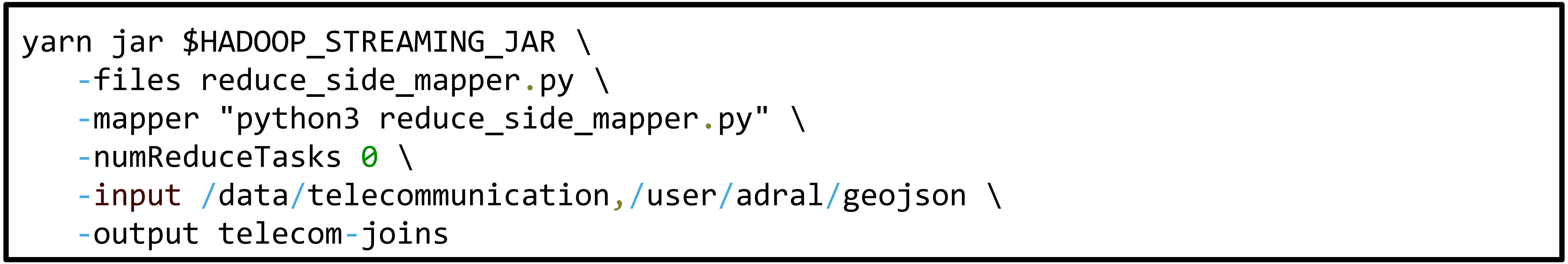
# Reduce- Side Join



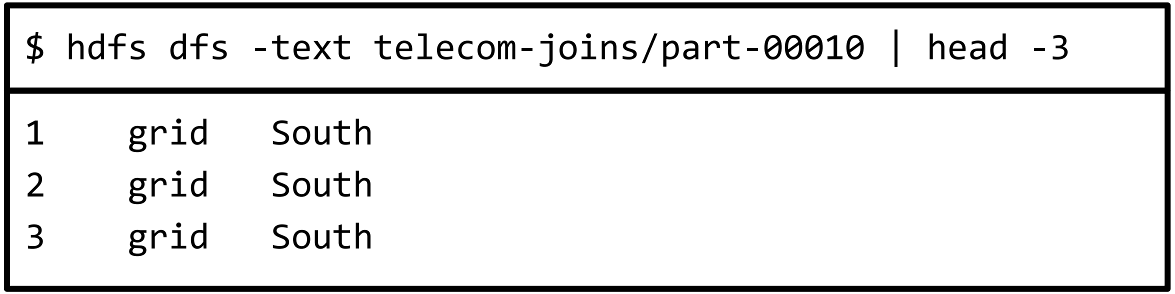
# Tagging

```
if "geojson" in os.environ["mapreduce_map_input_file"]:
    geojson = json.load(sys.stdin)
    grid = load_grid(geojson)
    for grid_id, ce11_type in grid.items():
              print(grid_id, "grid", ce11_type, sep="\t")
else
    for line in sys.stdin:
      grid_id, aggregate = line.split("\t", 1)
      grid_id = int(grid_id)
      time_interval, country, sms_in, sms_out, call_in, call_out, internet = aggregate.split("/t")
      if sms_in:
          sms_in = float(sms_in)
          print(grid_id, "logs", sms_in, sep="\t"))
```

# Пример запуска

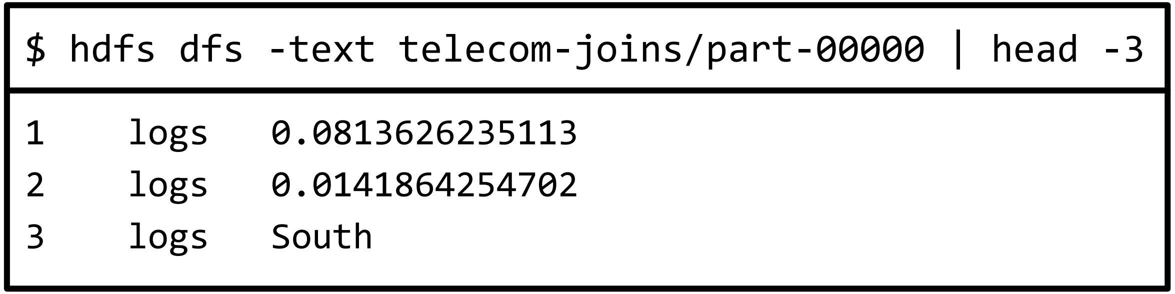


```
yarn jar $HADOOP_STREAMING_JAR \
    -files reduce_side_mapper.py \
    -mapper "python3 reduce_side_mapper.py" \
    -numReduceTasks 0 \
    -input /data/telecommunication,/user/adral/geojson \
    -output telecom-joins
```



```
$ hdfs dfs -text telecom-joins/part-00010 | head -3

1     grid    South
2     grid    South
3     grid    South
```



```
$ hdfs dfs -text telecom-joins/part-00000 | head -3

1     logs    0.0813626235113
2     logs    0.0141864254702
3     logs    South
```

# Пример запуска

```
yarn jar $HADOOP_STREAMING_JAR \
    -files reduce_side_mapper.py \
    -mapper "python3 reduce_side_mapper.py" \
    -numReduceTasks 0 \
    -input /data/telecommunication,/user/adral/geojson \
    -output telecom-joins
```

```
$ hdfs dfs -text telecom-joins/part-00010 | head -3
```

```
1     grid    South
2     grid    South
3     grid    South
```

string

```
$ hdfs dfs -text telecom-joins/part-00000 | head -3
```

```
1     logs    0.0813626235113
2     logs    0.0141864254702
3     logs    South
```

numeric

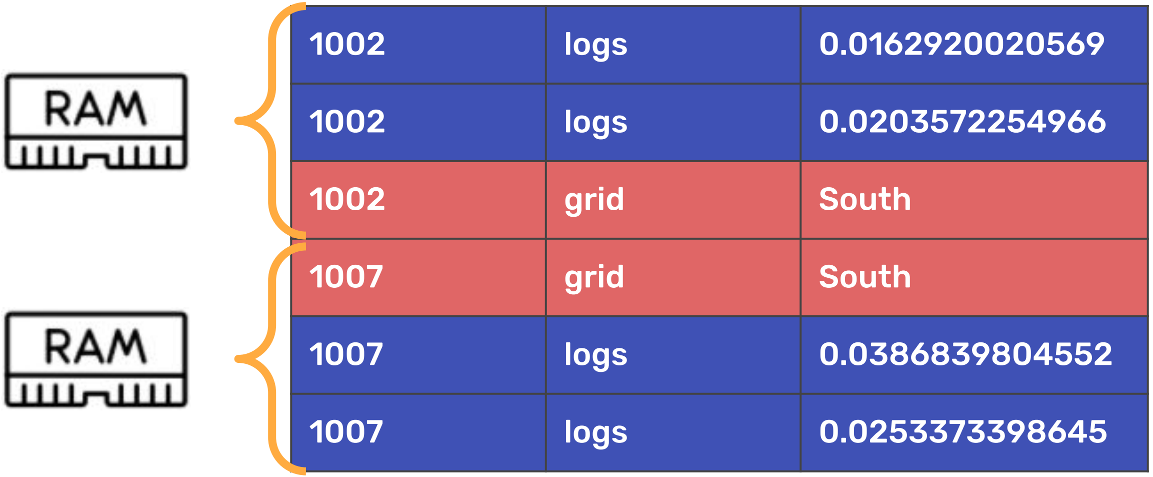
# Добавляем Identity Reducer

```
yarn jar $HADOOP_STREAMING_JAR \
    -D mapreduce.partition.keypartitioner.options="-k1,1" \
    -files reduce_side_mapper_slice.py \
    -mapper "python3 reduce_side_mapper.py" \
    -numReduceTasks 5 \
    -input /data/telecommunication,/user/adral/geojson \
    -output telecom-joins \
    -partitioner.org.apache.hadoop.mapred.lib.KeyFieldBasedPartitioner
```

| | | |
|---|---|---|
| 1002 | logs | 0.0162920020569 |
| 1002 | logs | 0.0203572254966 |
| 1002 | grid | South |
| 1007 | grid | South |
| 1007 | logs | 0.0386839804552 |
| 1007 | logs | 0.0253373398645 |

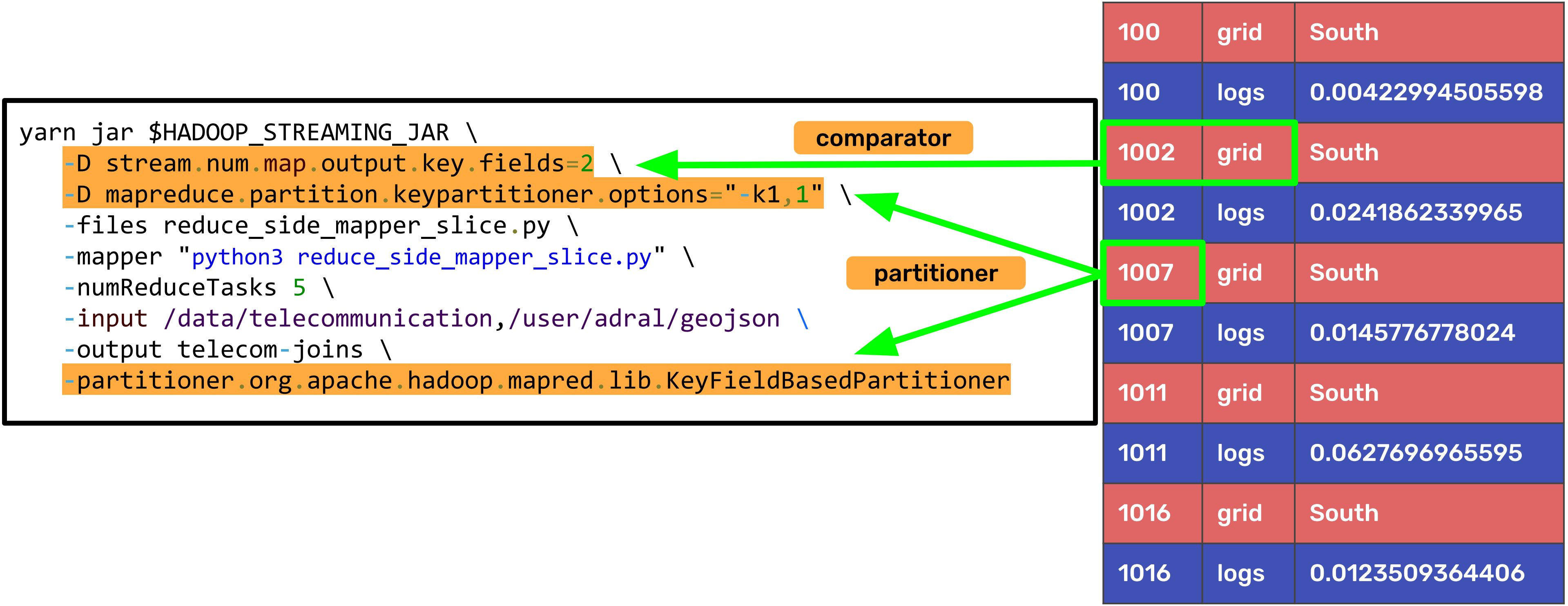
# Добавляем Identity Reducer

```
yarn jar $HADOOP_STREAMING_JAR \
    -D mapreduce.partition.keypartitioner.options="-k1,1" \
    -files reduce_side_mapper_slice.py \
    -mapper "python3 reduce_side_mapper.py" \
    -numReduceTasks 5 \
    -input /data/telecommunication,/user/adral/geojson \
    -output telecom-joins \
    -partitioner.org.apache.hadoop.mapred.lib.KeyFieldBasedPartitioner
```



| | | |
|---|---|---|
| 1002 | logs | 0.0162920020569 |
| 1002 | logs | 0.0203572254966 |
| 1002 | grid | South |
| 1007 | grid | South |
| 1007 | logs | 0.0386839804552 |
| 1007 | logs | 0.0253373398645 |

# Secondary Sort (via Comparator)



```
yarn jar $HADOOP_STREAMING_JAR \
    -D stream.num.map.output.key.fields=2 \
    -D mapreduce.partition.keypartitioner.options="-k1,1" \
    -files reduce_side_mapper_slice.py \
    -mapper "python3 reduce_side_mapper_slice.py" \
    -numReduceTasks 5 \
    -input /data/telecommunication,/user/adral/geojson \
    -output telecom-joins \
    -partitioner org.apache.hadoop.mapred.lib.KeyFieldBasedPartitioner
```

| | | |
|---|---|---|
| 100 | grid | South |
| 100 | logs | 0.00422994505598 |
| 1002 | grid | South |
| 1002 | logs | 0.0241862339965 |
| 1007 | grid | South |
| 1007 | logs | 0.0145776778024 |
| 1011 | grid | South |
| 1011 | logs | 0.0627696965595 |
| 1016 | grid | South |
| 1016 | logs | 0.0123509364406 |

# Secondary Sort: Comparator Flags

```
yarn jar $HADOOP_STREAMING_JAR \
        -D mapreduce.job.output.key.comparator.class=org.apache.hadoop.mapreduce.lib.partition.KeyFieldBasedComparator \
        -D mapreduce.partition.keycomparator.options="-k1,2r" \
        -D mapreduce.partition.keypartitioner.options="-k1,1" \
        -D stream.num.map.output.key.fields=2 \
        -files reduce_side_mapper_slice.py \
        -mapper "python3 reduce_side_mapper_slice.py" \
        -numRediceTasks 0 \
        -input /data/telecommunication,/user/adral/geojson \
        -output telecom-joins \
        -partitioner org.apache.hadoop.mapred.lib.KeyFieldBasedPartitioner
```

| | | |
|---|---|---|
| 9996 | logs | 0.0149333295147 |
| 9996 | grid | North |
| 9991 | logs | 0.330465627227 |
| 9991 | grid | North |
| 9987 | logs | 0.0296826530265 |
| 9987 | grid | North |
| 9982 | logs | 0.262932749854 |
| 9982 | grid | North |
| 998 | logs | 0.0881801546604 |
| 998 | grid | South |