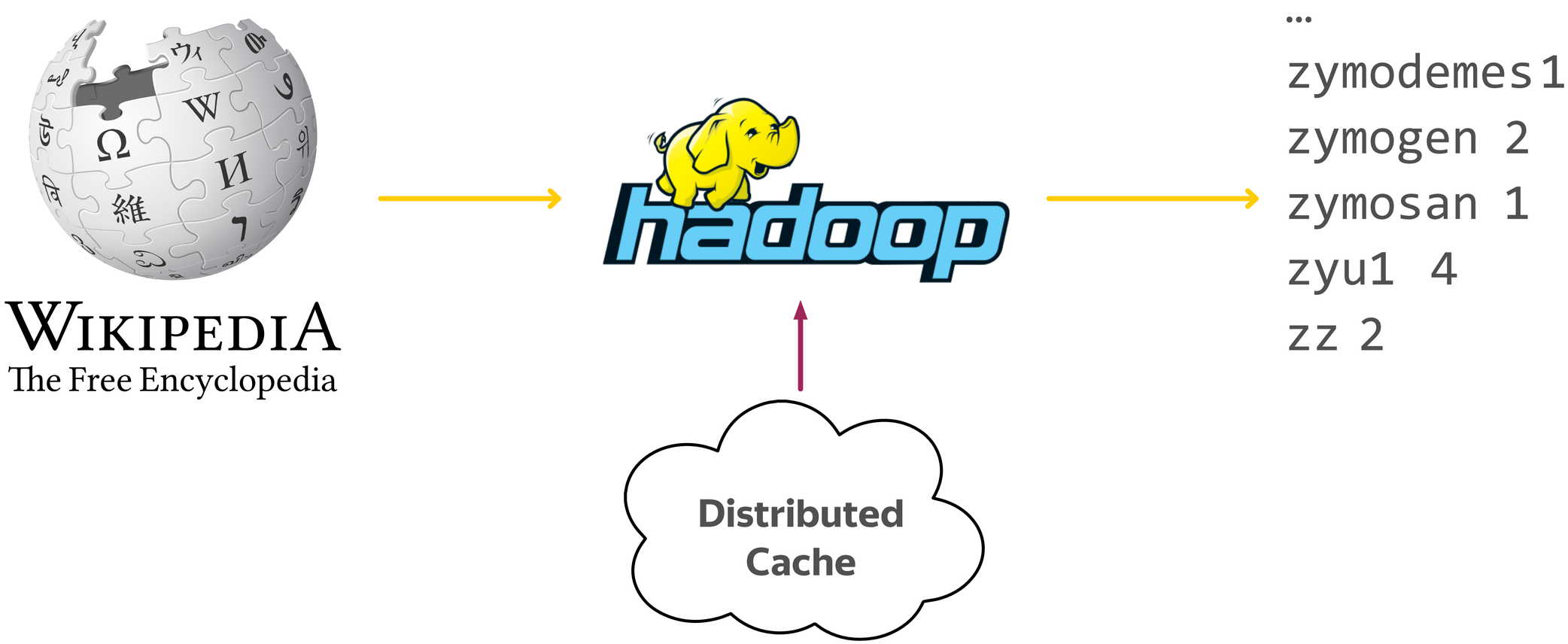
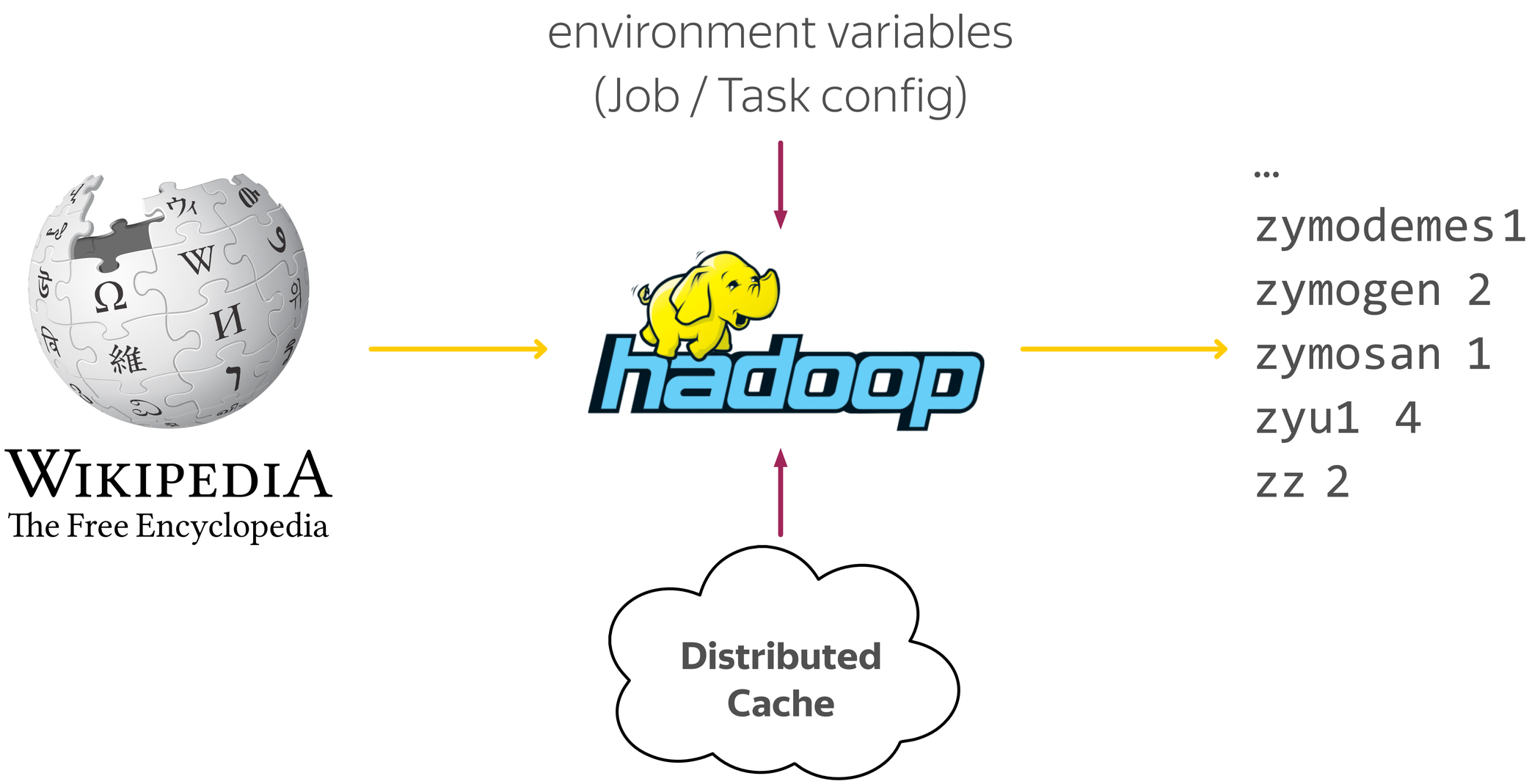
Yandex

MapReduce

Environment, Counters


WIKIPEDIA
The Free Encyclopedia
hadoop
Distributed Cache
...
zymodemes 1
zymogen 2
zymosan 1
zyu1 4
zz 2


environment variables
(Job / Task config)
WIKIPEDIA
The Free Encyclopedia
hadoop
Distributed
Cache
...
zymodemes 1
zymogen 2
zymosan 1
zyu1 4
zz 2

environment variables
(Job / Task config)

Mapper (Python): mapper.py

```
from __future__ import print_function
import re
import sys

if os.environ["mapred_task_is_map"] == "true":
   print("input_file:{}, start:{}, size:{}".format(
     os.environ["mapreduce_map_input_file"],
     os.environ["mapreduce_map_input_start"],
     os.environ["mapreduce_map_input_length"],
  ))

for line in sys.stdin:
  pass
```

environment variables
(Job / Task config)

Mapper (Python): mapper.py

```
from __future__ import print_function
import re
import sys

if os.environ["mapred_task_is_map"] == "true":
   print("input_file:{}, start:{}, size:{}".format(
     os.environ["mapreduce_map_input_file"],
     os.environ["mapreduce_map_input_start"],
     os.environ["mapreduce_map_input_length"],
  ))

for line in sys.stdin:
  pass
```

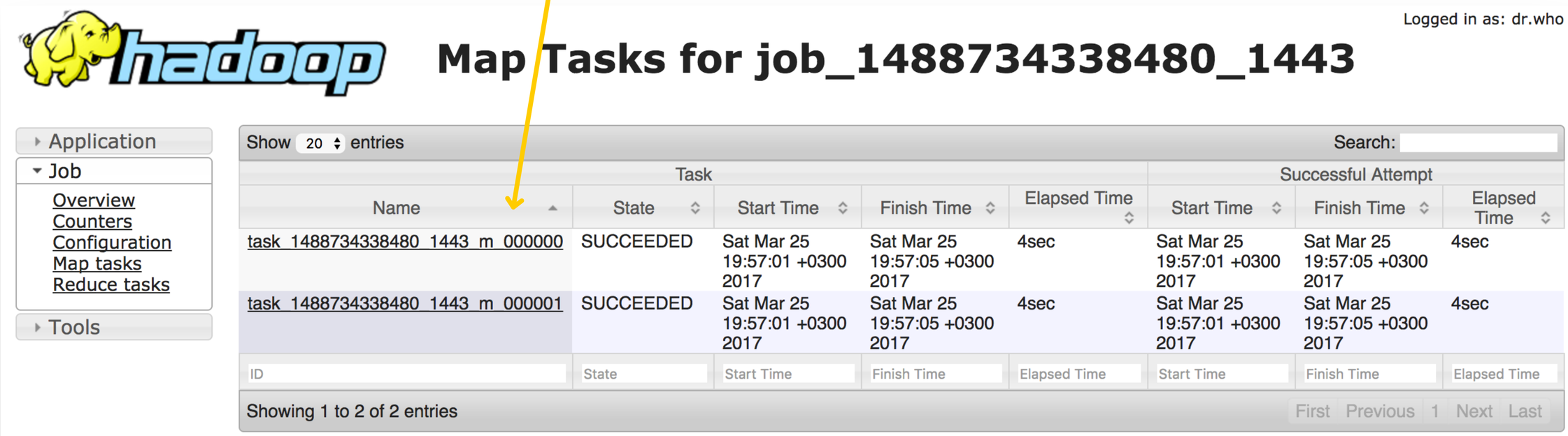
Mapper (Python): mapper.py

```
os.environ["mapreduce_task_id"]
os.environ["mapreduce_task_partition"]
```

Mapper (Python): mapper.py

```
os.environ["mapreduce_task_id"]
os.environ["mapreduce_task_partition"]
```

task_1488734338480_1443_m_000001
task_1488734338480_1443_r_000008



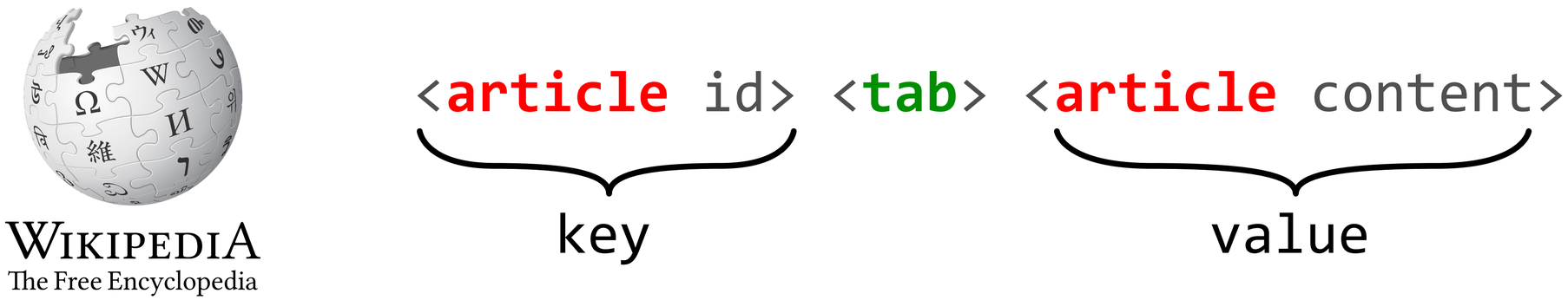
Mapper (Python): mapper.py

```
os.environ["mapreduce_task_id"]
os.environ["mapreduce_task_partition"]
```

1 ← task_1488734338480_1443_m_000001

8 ← task_1488734338480_1443_r_000008

```python
from __future__ import print_function
import re
import sys

CHARS_IN_LINE = 9

if os.environ["mapred_task_is_map"] == "true":
   split_input_start = int(
        os.environ["mapreduce_map_input_start"]
   )//CHARS_IN_LINE

for split_line_index, line in enumerate(sys.stdin):
   line_number = split_line_index + split_input_start
   if (line_number < 10):
      print(line_number, line, sep='\t')
```



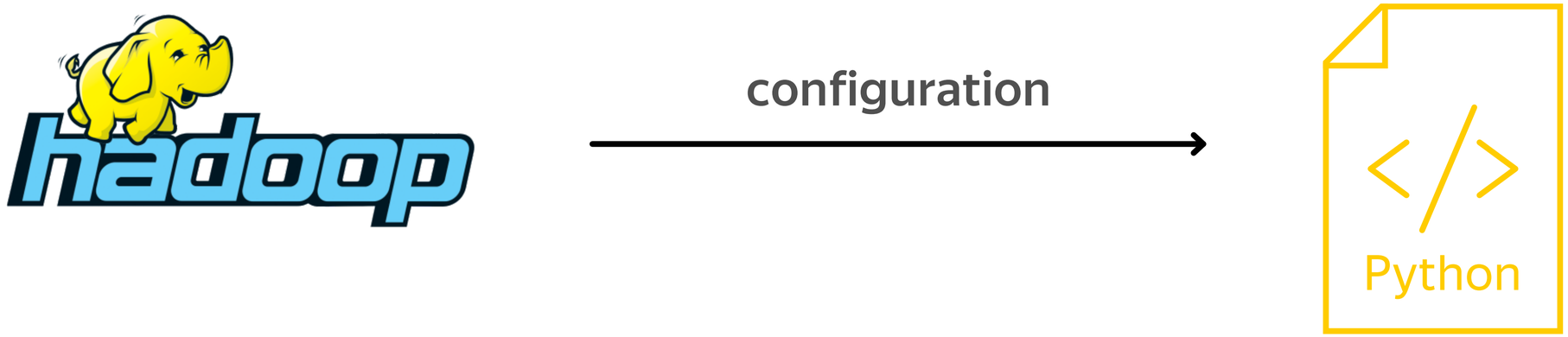
Mapper (Python): wc_mapper.py

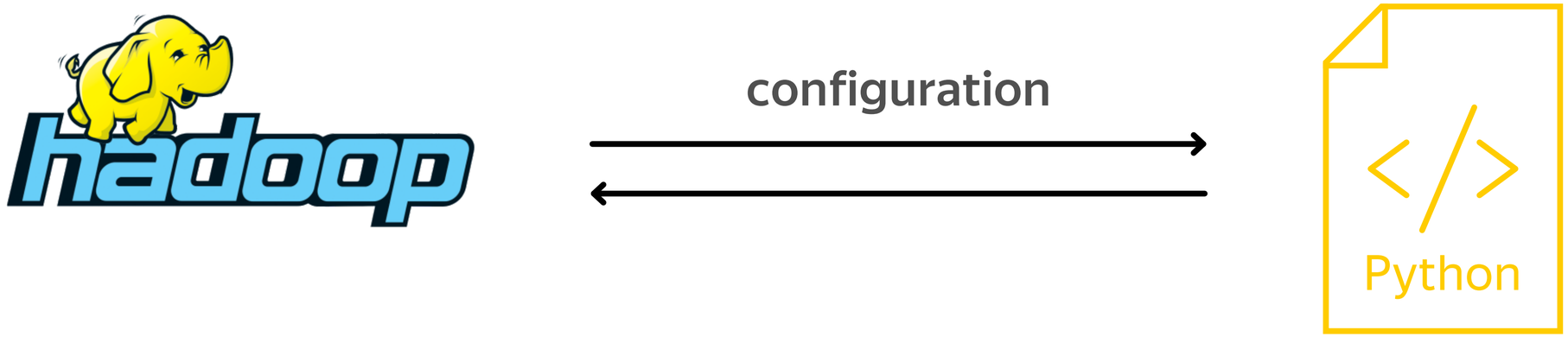
```
from __future__ import print_function
import os
import re
import sys

pattern = re.compile(os.environ["word_pattern"])

for line in sys.stdin:
  article_id, content = line.split("\t", 1)
    words = re.findall(pattern, content)
    for word in words:
          print(word, 1, sep="\t")
```

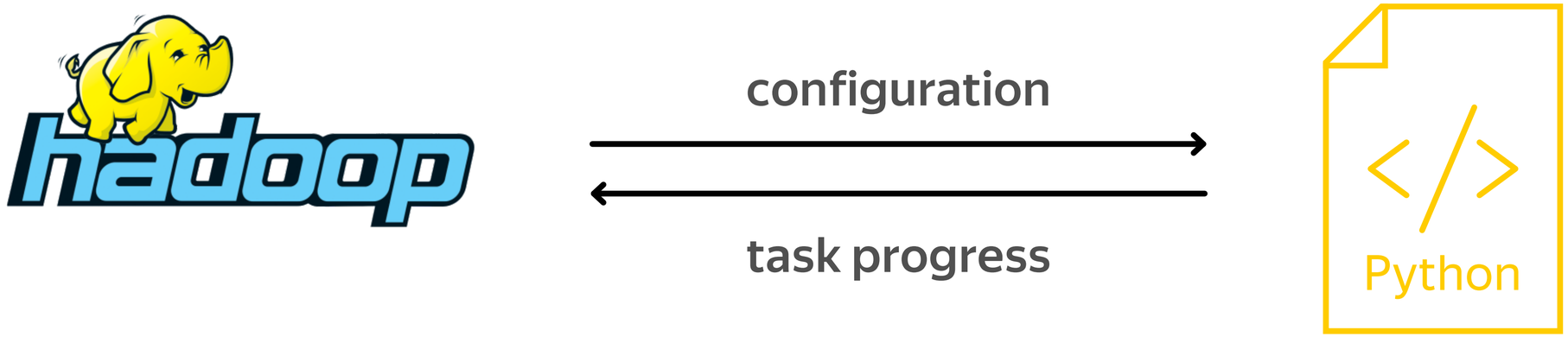
```
yarn jar $HADOOP_STREAMING_JAR -D word_pattern="\w+\d+" \
          -files mapper.py \
          -mapper 'python mapper.py' \
          -reducer 'python reducer.py' \
          -input /data/wiki/en_articles \
          -output word_count
```

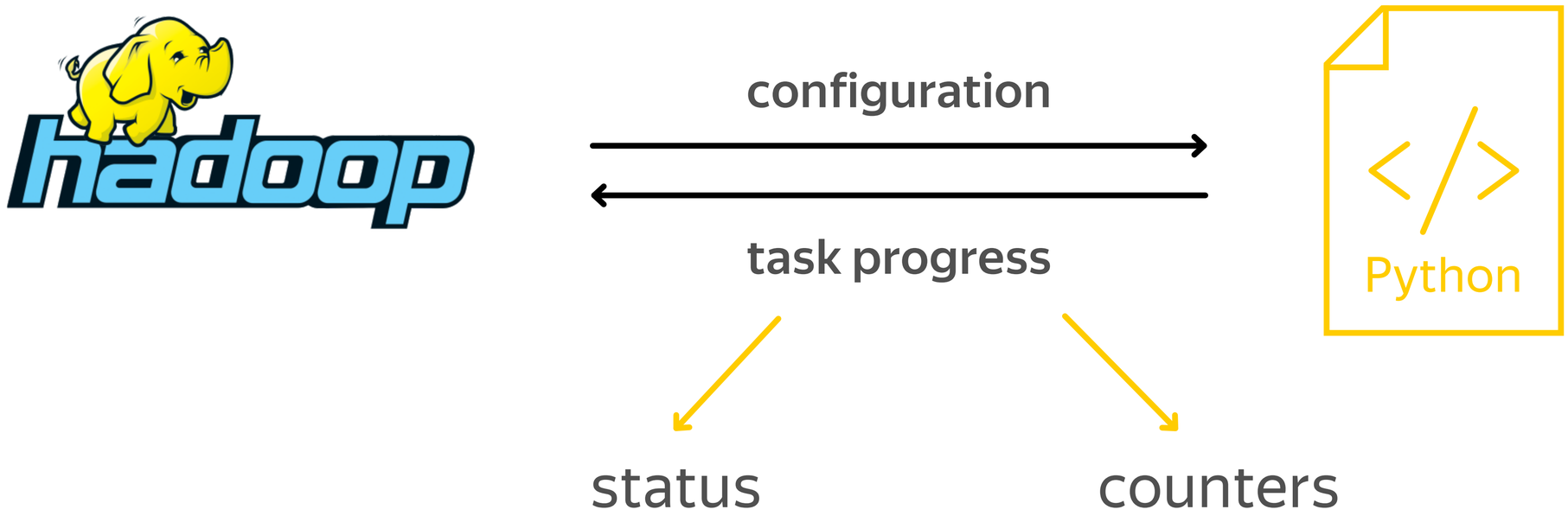
```
yarn jar $HADOOP_STREAMING_JAR -D word_pattern="\w+\d+" \
          -files mapper.py \
          -mapper 'python mapper.py' \
          -reducer 'python reducer.py' \
          -input /data/wiki/en_articles \
          -output word_count
```

```
$ hdfs dfs -text word_count/*
…
test2   4
times11 1
times48 3
tinctoria1      1
titan2
…
```


hadoop
configuration
Python


hadoop
configuration
Python


hadoop
configuration
task progress
Python


hadoop
configuration
task progress
Python
status
counters

Mapper (Python): reporter_mapper.py

```python
from __future__ import print_function
import re
import sys

for line in sys.stdin:
    article_id, content = line.split("\t", 1)
    words = re.findall("\w+", content)
    for index, word in enumerate(words):
        print(word, 1, sep="\t")
        print("reporter:status:processed {} words"
            .format(index + 1), file=sys.stderr)
```

Show 20 entries

Attempt	State	Status
attempt_1488734338480_1448_m_000000_0	SUCCEEDED	processed 6374 words
attempt_1488734338480_1448_m_000001_0	SUCCEEDED	processed 1778 words
Attempt	State	Status

Showing 1 to 2 of 2 entries

Mapper (Python): reporter_

```
from __future__ import print_functi
import re
import sys

for line in sys.stdin:
    article_id, content = line.split("\t", 1)
    words = re.findall("\w+", content)
    for index, word in enumerate(words):
            print(word, 1, sep="\t")
            print("reporter:status:processed {} words"
                .format(index + 1), file=sys.stderr)
```

Mapper (Python): reporter_mapper.py

```
from __future__ import print_function
import re
import sys

for line in sys.stdin:
    article_id, content = line.split("\t", 1)
    words = re.findall("\w+", content)
    for index, word in enumerate(words):
            print(word, 1, sep="\t")
            print("reporter:status:processed {} words"
                .format(index + 1), file=sys.stderr)
            print(" reporter:counter:Personal Counters,word found,1",
                file=sys.stderr)
                file=sys.stderr)
)
```

reporter:counter:<group>,<counter>,<amount>

	Name	Map	Reduce	Total
Map-Reduce Framework	Combine input records	0	0	0
	Combine output records	0	0	0
	CPU time spent (ms)	206620	63080	269700
	Failed Shuffles	0	0	0
	GC time elapsed (ms)	832	1533	2365
	Input split bytes	264	0	264
	Map input records	4100	0	4100
	Map output bytes	98534099	0	98534099
	Map output materialized bytes	9634808	0	9634808
	Map output records	12396473	0	12396473
	Merged Map outputs	0	20	20
	Physical memory (bytes) snapshot	3370938368	2752958464	6123896832
	Reduce input groups	0	306456	306456
	Reduce input records	0	12396473	12396473
	Reduce output records	0	306456	306456
	Reduce shuffle bytes	0	9634808	9634808
	Shuffled Maps	0	20	20
	Spilled Records	12396473	12396473	24792946
	Total committed heap usage (bytes)	3642228736	6243221504	9885450240
	Virtual memory (bytes) snapshot	7705399296	83205345280	90910744576
	Name	**Map**	**Reduce**	**Total**
Personal Counters	word found	12396473	0	12396473

```
    print(word, 1, sep="\t")
    print("reporter:status:processed {} words"
        .format(index + 1), file=sys.stderr)
    print("reporter:counter:Personal Counters,word found,1",
        file=sys.stderr)
        file=sys.stderr)
)
```

Summary

Summary

- You know how to:
- **provide environment variables** (global configuration) to your streaming scripts

Summary

- You know how to:
  - **provide environment variables** (global configuration) to your streaming scripts
  - **access job configuration** options (e.g. map input file)

Summary

- You know how to:
  - **provide environment variables** (global configuration) to your streaming scripts
  - **access job configuration** options (e.g. map input file)
  - **report progress** back to Hadoop MapReduce Framework

BigDATAteam