

Дата-инженер

Технология Map-Reduce

Николай Марков



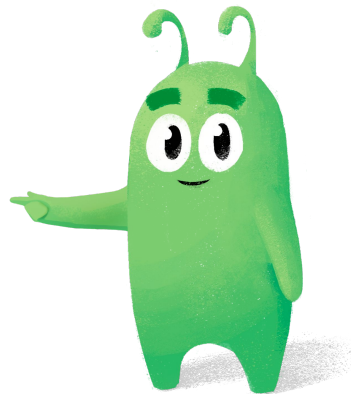
Что вы узнаете:

1 Как попробовать Hadoop и MapReduce на своей машине?

2 Что представляет собой парадигма вычислений MapReduce?

3 Как решать задачи разной сложности с использованием MapReduce?

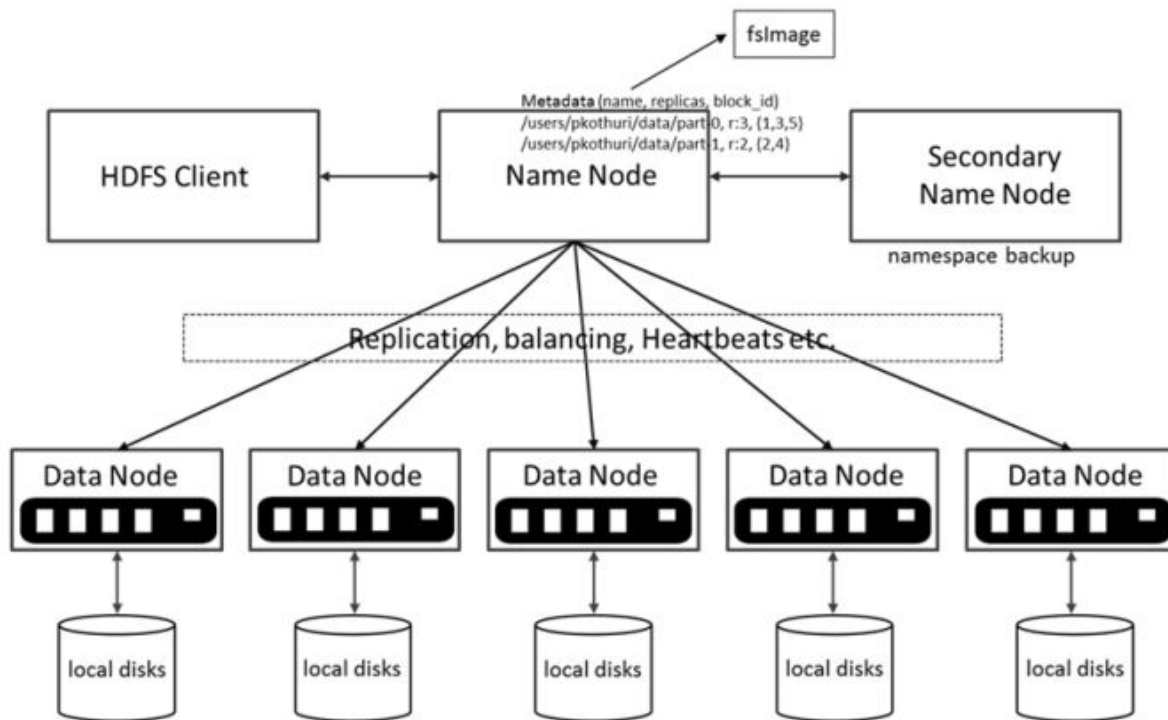
4 Зачем знать столь низкоуровневую парадигму в современном мире?



Поднимаем HDFS у себя



NameNode, SNameNode, DataNode



Основная концепция —
Data Locality

КупиСкачай слона

- В комплекте идут **HDFS**, фреймворк **MapReduce**, шедулер **YARN** и другие
- Можно [развернуть на одной машине](#)
- Поддерживается **Java** версий **8** и **11**
- Должен работать SSH на localhost без пароля

```
# добавить в etc/hadoop/hadoop-env.sh  
export JAVA_HOME=/usr/lib/jvm/default
```

У вас путь
может быть
другой



Переменные и конфиги

etc/hadoop/core-site.xml

```
<configuration>
  <property>
    <name>fs.defaultFS </name>
    <value>hdfs://localhost:9000 </value>
  </property>
</configuration>
```

etc/hadoop/hdfs-site.xml

```
<configuration>
  <property>
    <name>dfs.replication </name>
    <value>1</value>
  </property>
</configuration>
```

Переменные и конфиги

```
export JAVA_HOME=/usr/lib/jvm/default
export HADOOP_HOME=/opt/hadoop2
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export HADOOP_OPTS="$HADOOP_OPTS -Djava.library.path=$HADOOP_HOME/lib/native"
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
```

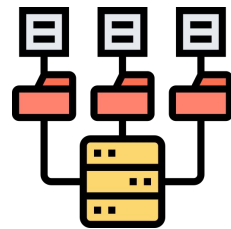
Запускаем HDFS

```
~$ bin/hdfs namenode -format
~$ sbin/start-dfs.sh
```

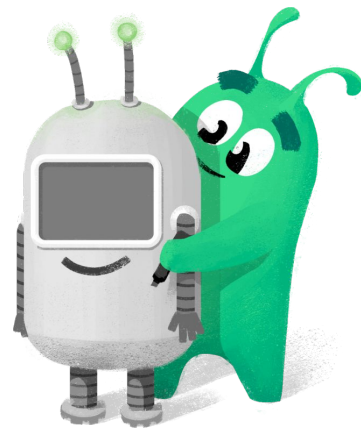
Работать с файлами в HDFS можно теми же командами, что в консоли, **добавляя префикс**

```
~$ bin/hdfs dfs -mkdir /user
~$ bin/hdfs dfs -mkdir /user/meow-nofer
~$ bin/hdfs dfs -mkdir /user/meow-nofer/input
~$ bin/hdfs dfs -put ./76-0.txt /user/meow-nofer/input
```

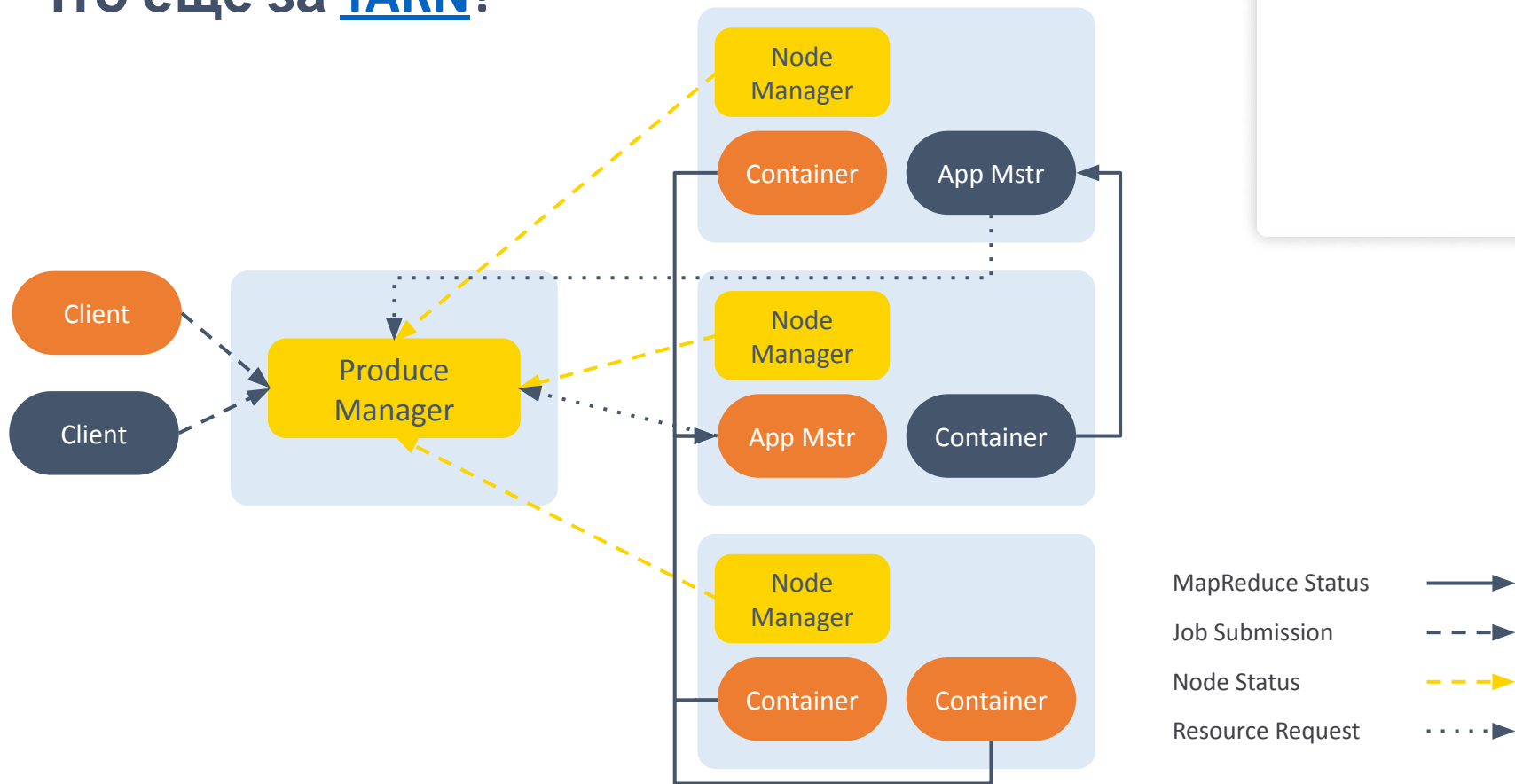
Проверить NameNode: <http://localhost:50070/>



Настраиваем YARN



Что еще за YARN?



Настраиваем YARN

etc/hadoop/mapred-site.xml

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
  <property>
    <name>mapreduce.application.classpath</name>
    <value>${HADOOP_MAPRED_HOME}/share/hadoop/mapreduce/*:${HADOOP_MAPRED_HOME}/share/hadoop/mapreduce/lib/*</value>
  </property>
</configuration>
```

etc/hadoop/yarn-site.xml

```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services </name>
    <value>mapreduce_shuffle </value>
  </property>
  <property>
    <name>yarn.nodemanager.env-whitelist </name>
    <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPATH_PREPEND_DISTCACHE,HADOOP_YARN_HOME,HADOOP_HOME,
    PATH,LANG,TZ,HADOOP_MAPRED_HOME </value>
  </property>
</configuration>
```

Запускаем YARN

```
~$ sbin/start-yarn.sh
```

Проверить YARN UI: <http://localhost:8088/>



MapReduce изнутри и снаружи



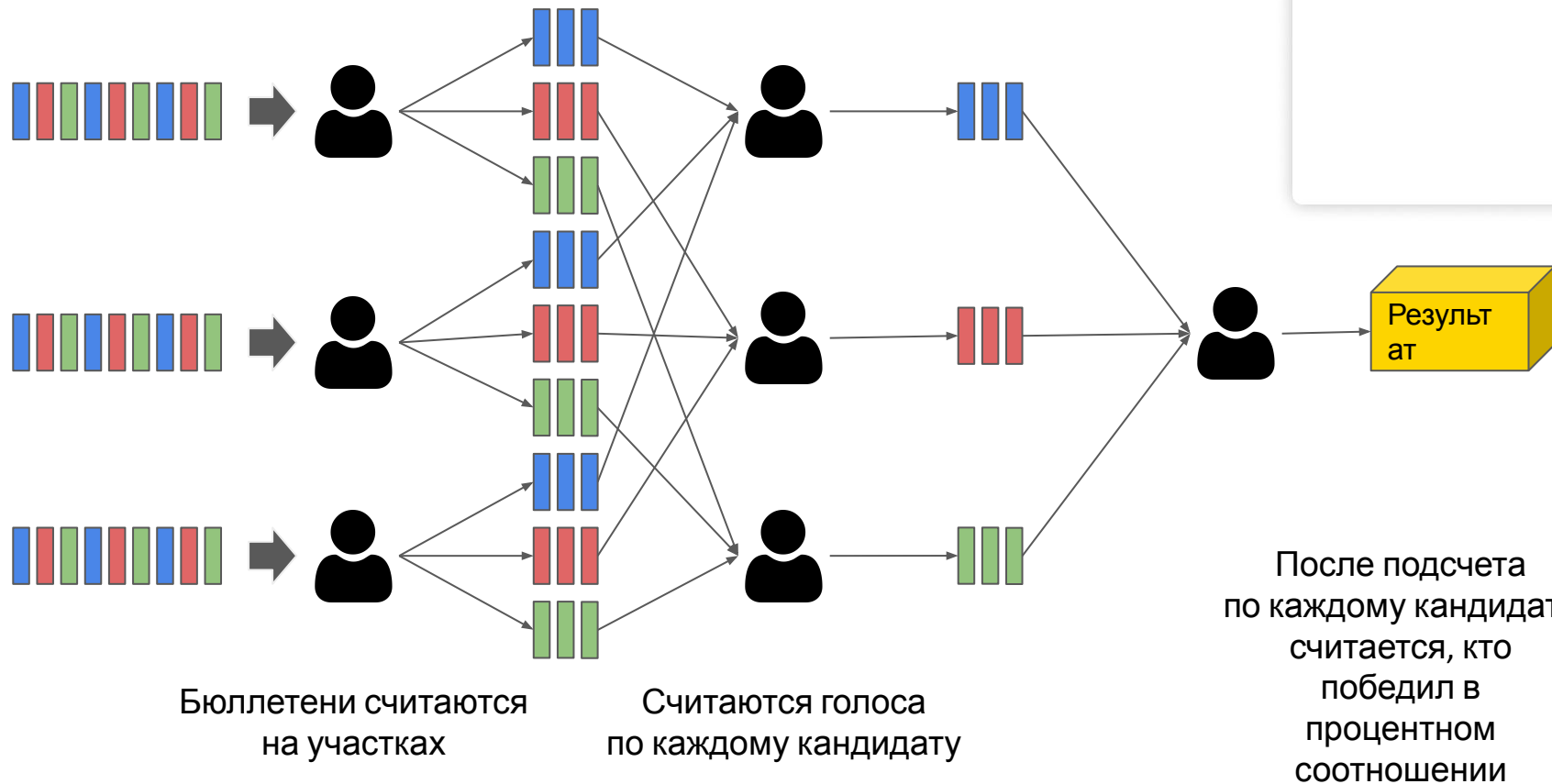
История и принципы

- До 2004 года было много проблем с хранением и обработкой данных на многих машинах
- Попытка распределенно посчитать и свести результат воедино, занимала несколько дней, а то и недель
- Каждое вычисление было уникальным, и для него требовалось каждый раз разрабатывать параллельный алгоритм



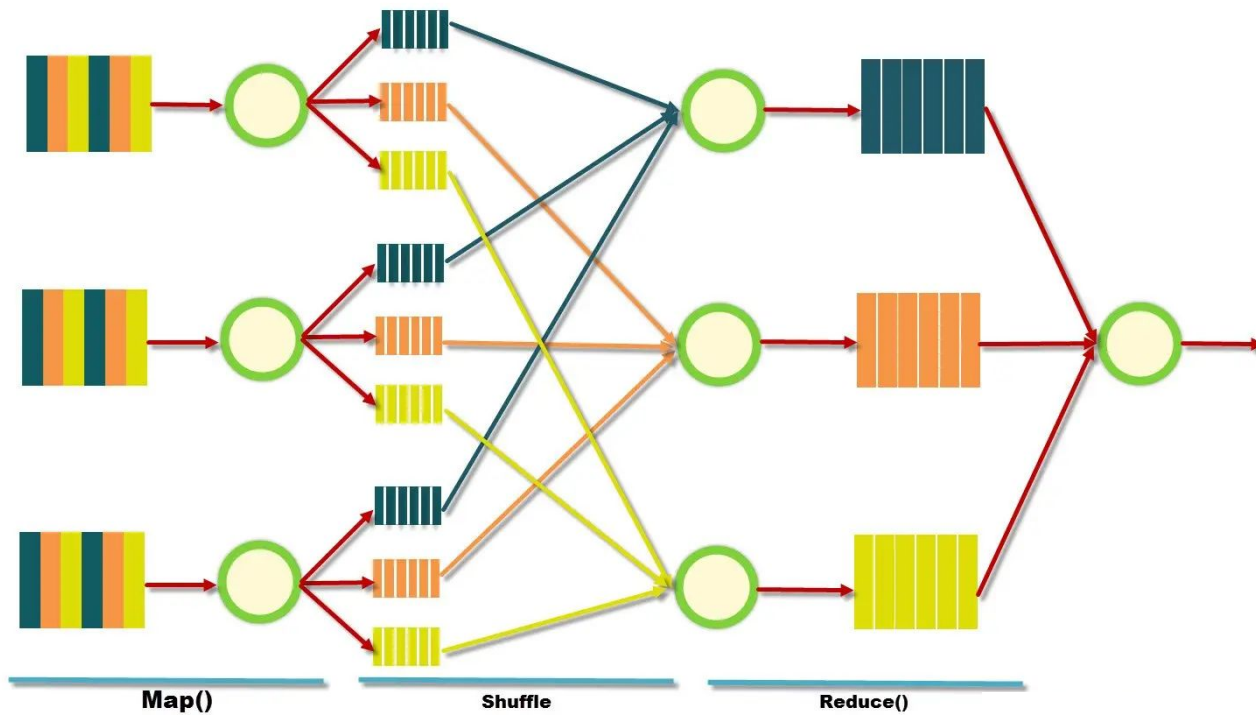
[Оригинальная статья](#)

История и принципы



Стадии в архитектуре

How MapReduce Works?



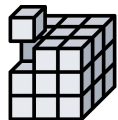
По шагам

Map



- Читаем данные с HDFS, преобразуя их в пары ключ-значение
- Одна и та же функция к каждой записи
- Рапортуем в центр о завершении

Partition



- Решаем, на какой редюсер отправить данные
- Все данные с одним ключом полетят на один редюсер
- Число партиций = число редюсеров

Shuffle



Перемещаем результаты с мапперов на соответствующие редюсеры

Sort



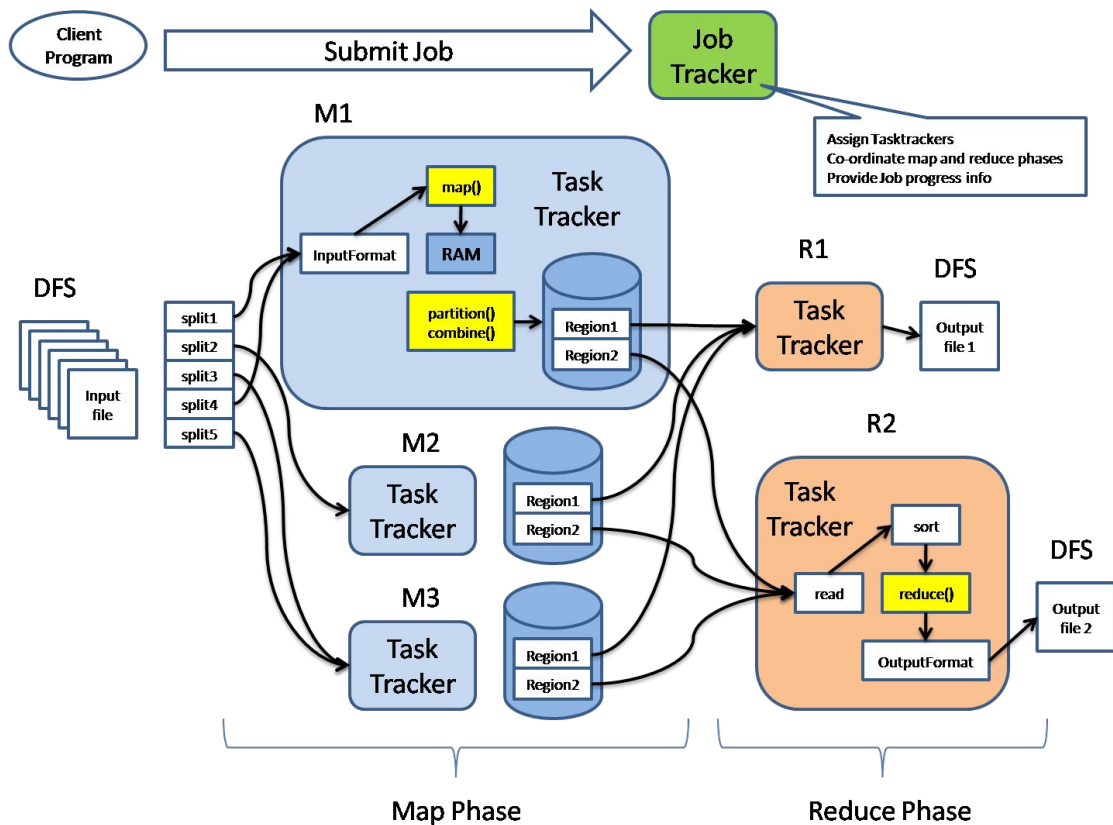
Линейно объединяем выхлопы Map в отсортированную последовательность

Reduce



Выполняем пользовательскую функцию Reduce на сортированных данных

Технически



Простые задачи



WordCount на псевдокоде

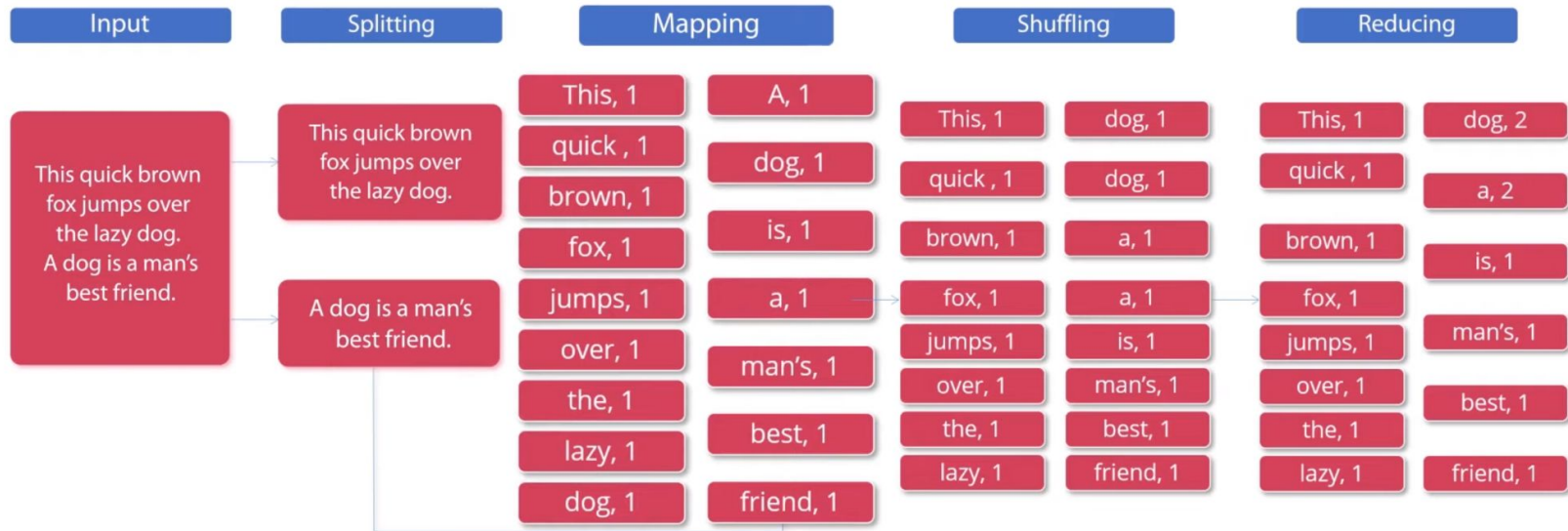
Сколько раз каждое из слов встречается в исходном тексте?

```
def map (doc) :  
  for word in doc :  
    yield word, 1
```

```
def reduce (word, values) :  
  yield word, sum (values)
```

WordCount в стадиях

MapReduce – Word Count



Отладка локально

Технология, которая нам поможет, называется [Hadoop Streaming](#)

Отлаживаем локально MapReduce на своей машине на кусочке данных

```
~$ cat input_file.txt | python3 mapper.py | sort -k1,1  
| python3 reducer.py > output_file.txt
```

```
~$ export HADOOP_STREAMING_JAR=$HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-2.7.3.jar
```



Запускаем все везде и сразу

На голом MapReduce

```
~$ hadoop jar $HADOOP_STREAMING_JAR \  
  -input /user/meow-nofer/input \  
  -output /user/meow-nofer/output \  
  -mapper ./01-wordcount/mapper.py \  
  -reducer ./01-wordcount/reducer.py
```

Через YARN

```
~$ yarn jar $HADOOP_STREAMING_JAR \  
  -input /user/meow-nofer/input \  
  -output /user/meow-nofer/output \  
  -mapper mapper2.py \  
  -reducer reducer2.py \  
  -file 01-wordcount/mapper2.py \  
  -file 01-wordcount/reducer2.py
```

Поиск отличников

Дан табель студентов с оценками:

Holmes	4	
Lopez	3	
Holmes	2	
Wilson	4	
Robinson		5
Weber	3	
Sanders	4	
Ball	2	
Butler	5	
Wilson	4	



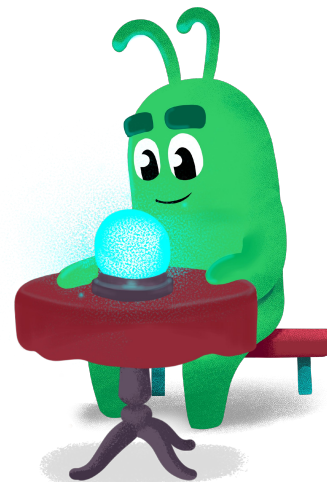
Отличником считается студент, у которого средний балл выше 4.5. Как найти всех отличников?

Синяя изолента

```
# mapper
map(x, score) →
    emit (x, score)
```

```
# reducer
reduce(x, scores) →
    avg = average(scores)
    if avg >= 4.5:
        emit(x, avg)
```

Секретные оптимизации



Combiner, Partitioner, Comparator, Compression

Combiner — промежуточный код, который запускается **между** маппером и редюсером. Может предагрегировать данные перед пересылкой их на редюсер

Partitioner — дополнительная логика, которая позволяет подтюнить партишенинг, например, в случае сложного составного ключа

Comparator — возможность указать более точно механизм сравнения ключей при шаффле и сортировке

Comparator

```
~$ yarn jar $HADOOP_STREAMING_JAR \  
  -D stream.num.map.output.key.fields=2 \  
  -D stream.num.reduce.output.key.fields=2 \  
  -D  
  mapreduce.job.output.key.comparator.class=org.apache.hadoop.mapreduce.lib.partition.KeyFieldBasedComparator \  
  -D mapreduce.partition.keycomparator.options="-k1,1n -k2,2" \  
  \
```

Partitioner

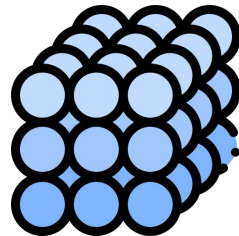
```
~$ yarn jar $HADOOP_STREAMING_JAR \  
  -D stream.num.map.output.key.fields=2 \  
  -D stream.num.reduce.output.key.fields=2 \  
  -D mapreduce.map.output.key.field.separator=.  
  -D  
  mapreduce.job.output.key.comparator.class=org.apache.hadoop.mapreduce.lib.par  
  tition.KeyFieldBasedComparator \  
  -D mapreduce.partition.keycomparator.options="-k1,1nr -k2,2" \  
  -D mapreduce.partition.keypartitioner.options=-k2.1,2.1 \  
  -partitioner org.apache.hadoop.mapred.lib.KeyFieldBasedPartitioner \  
  -
```

Количество мапперов и редьюсеров

- [Количество мапперов](#) по умолчанию зависит от разбиения входных данных, а разбиение зависит от размера блока (помните про 128Мб на HDFS?)
- Тем не менее, можно указать количество задач явно
- Если выставить количество редьюсеров в 0 — получится **map-only** задача, которая может быть нужна, например, для фильтрации

```
... -D mapred.map.tasks=5 -D mapred.reduce.tasks=2 ...
```

```
# также работает -numReduceTasks 1
```



Как работают Join'ы



Средняя оценка по предмету

- Есть табель и результат опроса студентов:
“Какой предмет самый любимый”
- Как посчитать для каждого предмета **среднюю оценку** среди любителей этого предмета?



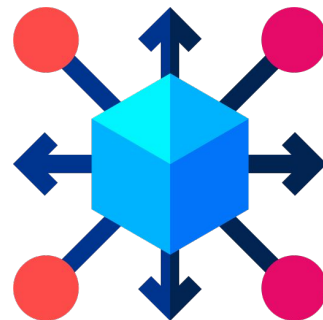
ReduceJoin

1. Делаем для каждого датасета составной ключ с типом значений
2. Считаем для каждого предмета среднюю оценку для каждого студента
3. Считаем среднюю оценку по всем студентам



MapJoin

1. Те же данные, что и в прошлой задаче
2. Как посчитать для каждого предмета среднюю оценку без усреднения по людям?
3. Подсказка — список любимых предметов помещается в память



Итоги. О чем поговорили:

- 1 Как развернуть у себя компоненты Hadoop
- 2 Как работает MapReduce и из каких стадий состоит
- 3 Что общего между MapReduce и синей изолентой
- 4 Секретные техники оптимизации вычислений
- 5 Зачем все это нужно знать DE в современном мире





**Спасибо
за внимание!**

