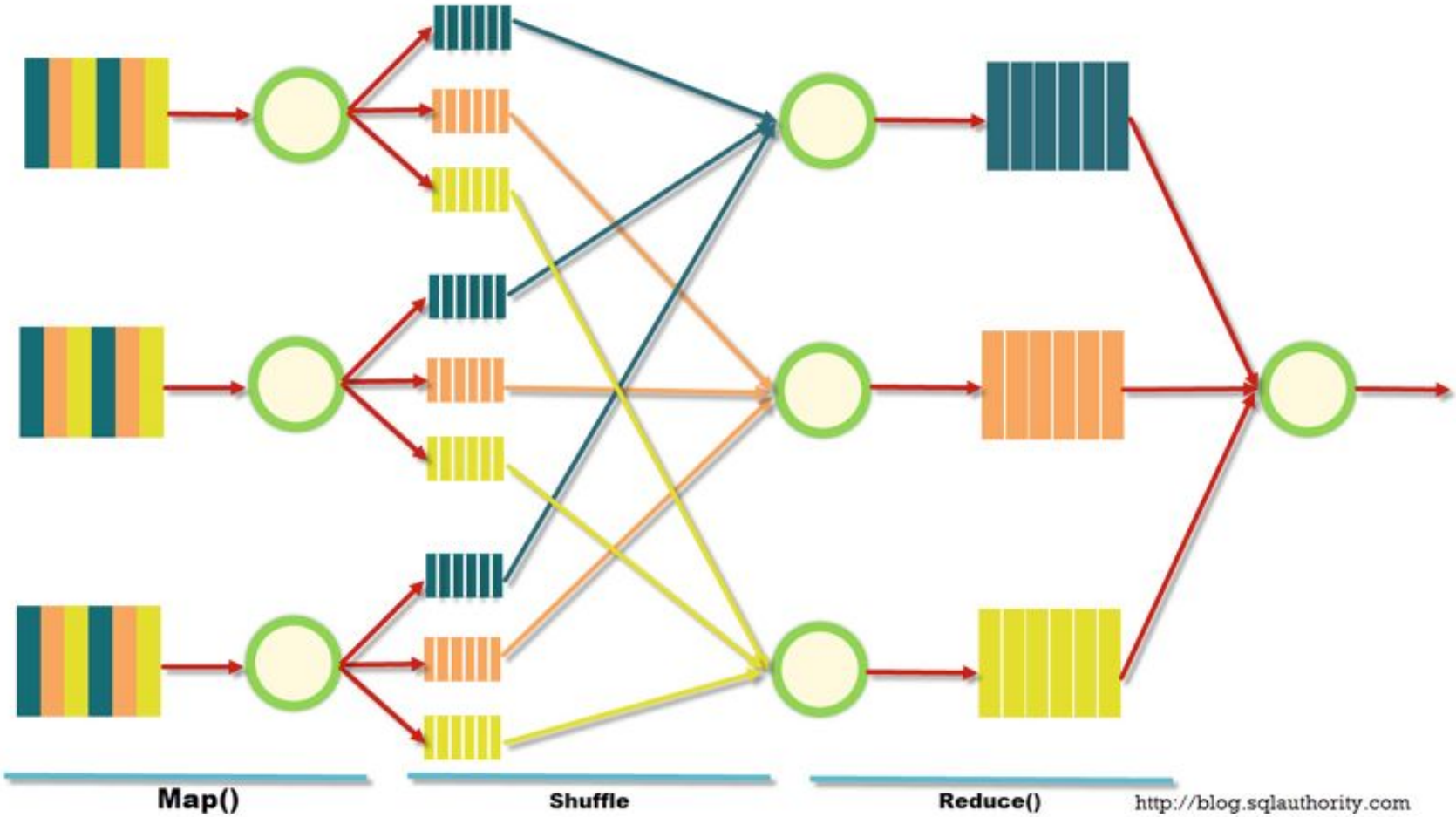
Приёмы и стратегии работы с MapReduce

Kirill Danilyuk, DS @ RnD Lab

How MapReduce Works?



Уровни абстракции MapReduce

Уровень 0: примитивы `map` и `reduce`

Уровень 1: концептуальная модель в рамках пейпы от Google 2004 года

Уровень 2: (сегодняшнее занятие) утилизация парадигмы MapReduce в виде кода, реализующего функции `map` и `reduce`

Уровень 3: реализация MapReduce в виде фреймворка, абстрагирующего инженерные аспекты работы

```
map(String key, String value):
  // key: document name
  // value: document contents
  for each word w in value:
    EmitIntermediate(w, "1");

reduce(String key, Iterator values):
  // key: a word
  // values: a list of counts
  int result = 0;
  for each v in values:
    result += ParseInt(v);
  Emit(AsString(result));
```

Map, `Mapper`, `map`

Map: шаг в терминах фреймворка

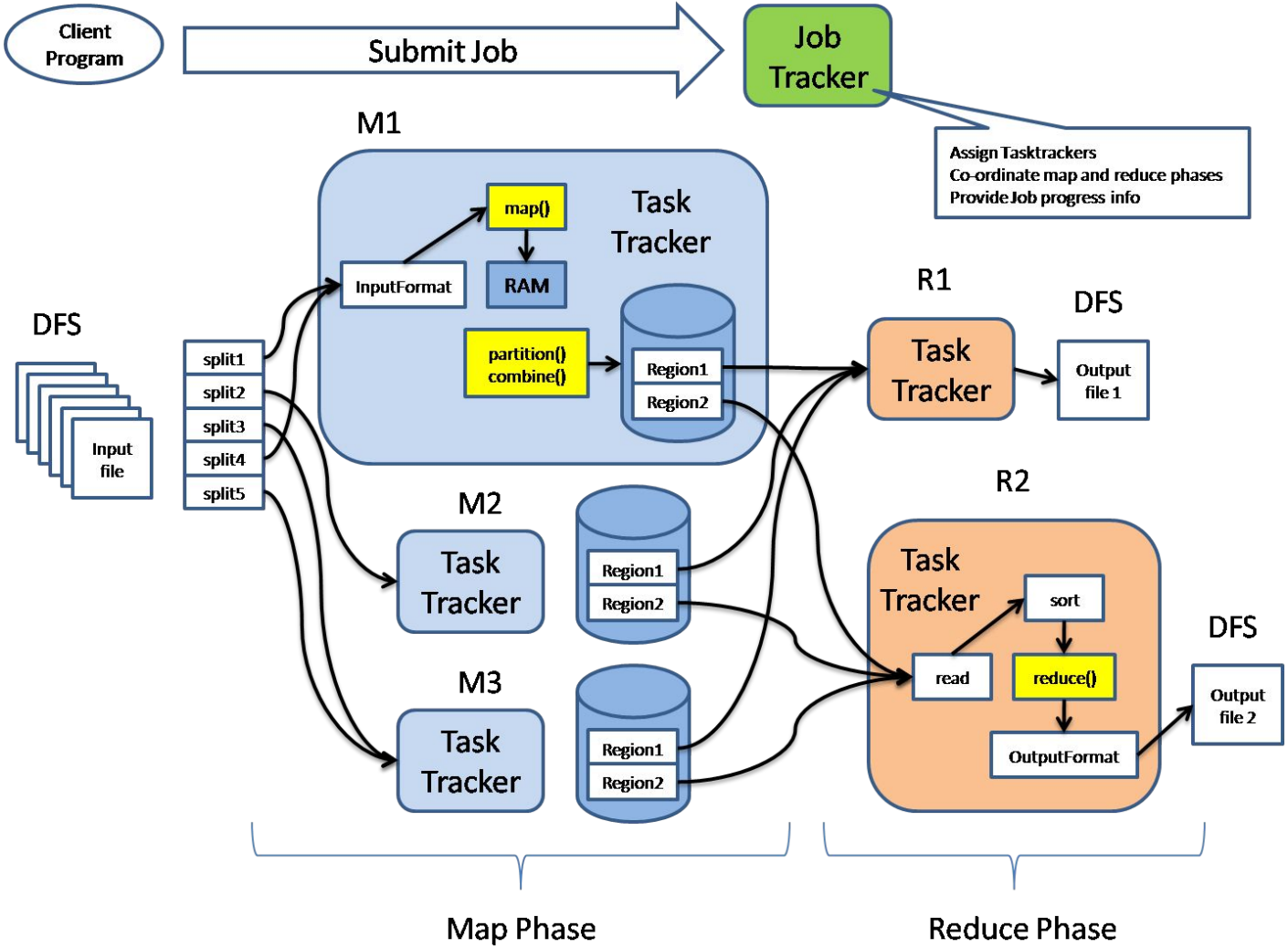
`Mapper`: интерфейс в терминах Java API. Отображает исходные key/value пары в набор intermediate key/value пар

`map`: процедура, которую необходимо реализовать в рамках интерфейса. Трансформирует одну пару key с value.

```
public class TokenCounterMapper
     extends Mapper<Object, Text, Text, IntWritable>{

   private final static IntWritable one = new IntWritable(1);
   private Text word = new Text();

   public void map(Object key, Text value, Context context)
       throws IOException, InterruptedException {
     StringTokenizer itr = new StringTokenizer(value.toString());
     while (itr.hasMoreTokens()) {
       word.set(itr.nextToken());
       context.write(word, one);
     }
   }
 }
```


Client Program
Submit Job
Job Tracker
Assign Tasktrackers
Co-ordinate map and reduce phases
Provide Job progress info
M1
map()
Task Tracker
InputFormat
RAM
partition()
combine()
Region1
Region2
DFS
Input file
split1
split2
split3
split4
split5
R1
DFS
Task Tracker
Output file 1
M2
Task Tracker
Region1
Region2
R2
Task Tracker
sort
read
reduce()
OutputFormat
DFS
Output file 2
M3
Task Tracker
Region1
Region2
Map Phase
Reduce Phase

0. Word Count (пример)

Работа на кластере

Работать будем на master-машине, зайдите на неё сейчас по ssh.

Скопируйте себе в домашнюю директорию master-машины данные:
`cp -R /tmp/mapreduce_seminar/data ~/.`

После занятия ответы будут доступны в директории
`/tmp/mapreduce_seminar/solutions` master-ноды.

Локальная отладка

```
$ cat data/doc.txt | python3 mapper.py | sort
-k1,1 | python3 reducer.py > data/result.txt
```

Вспомним Word Count

```python
# mapper.py

import sys

for line in sys.stdin:
    for token in line.strip().split():
        print(token + '\t1')
```

```python
# reducer.py

import sys

prev_key = None
sum = 0

for line in sys.stdin:
    key, value = line.split("\t")
    if key != prev_key and prev_key is not None:
        print(prev_key, sum)
        sum = 0
    sum += 1
    prev_key = key
if key != prev_key and prev_key is not None:
    print(prev_key, sum)
```

```python
import sys

# Accumulators
prev_key = None
values = []

# reduce() method
def do_reduce(key, values):
    yield key, sum(values)

# Reducer
for line in sys.stdin:
    key, value = line.split("\t")
    if key != prev_key and prev_key is not None:
        for k, v in do_reduce(prev_key, values):
            sys.stdout.write('{},{}\n'.format(k, str(v)))
        values = []
    values.append(int(value))
    prev_key = key

# Off by one correction
if key != prev_key and prev_key is not None:
    for k, v in do_reduce(prev_key, values):
        sys.stdout.write('{},{}\n'.format(k, str(v)))
```

Word Count (Reducer)

Оформление задачи

```
# mapper:
map(x) →
for token in x.split():
    emit(token, 1)
```

```
# reducer:
reduce(key, values) →
    emit(key, sum(values))
```

1. Поиск отличников

Поиск отличников

Дан файл вида студент<tab>оценка

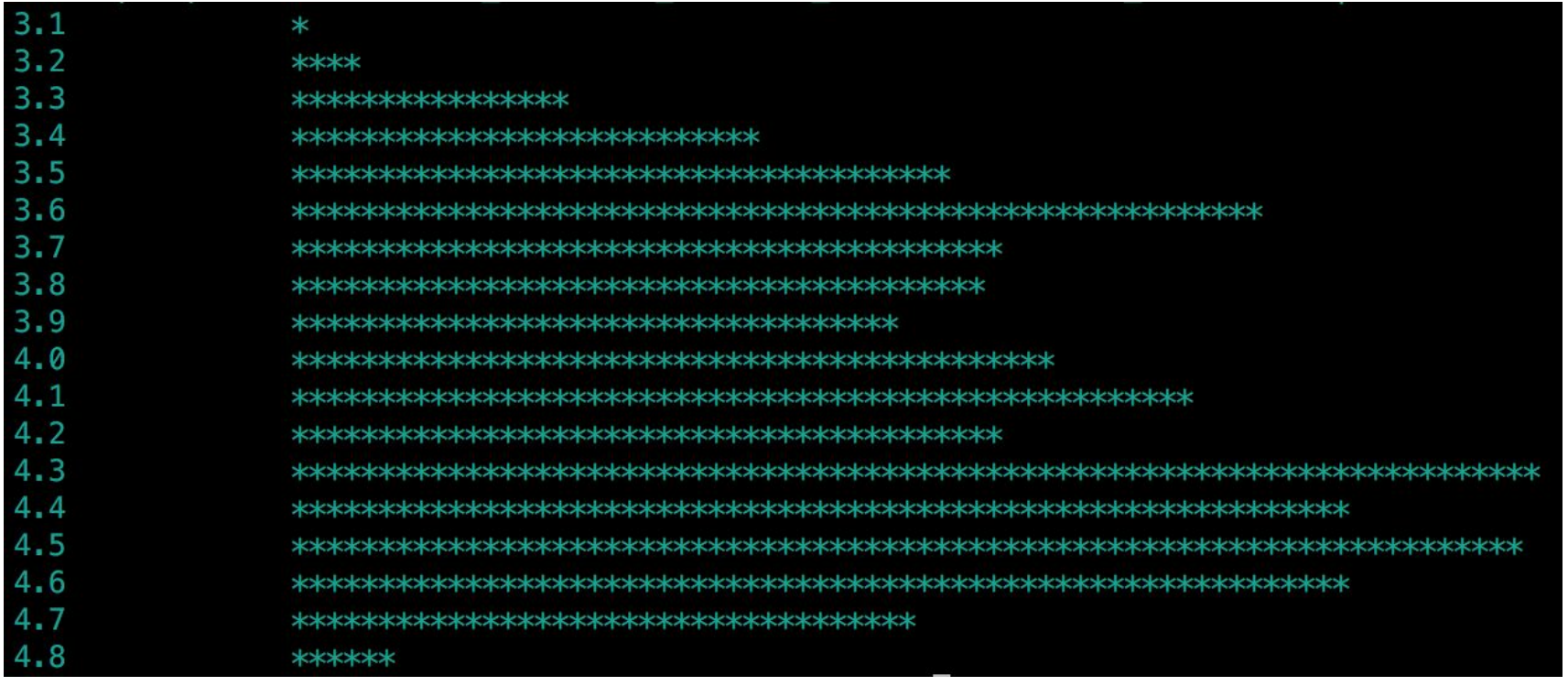
- Фейн 3
- Куликов 4
- Петровна 3
- Мордовина 4
- Коробейников 5
- Кремнев 5
- Мордовина 3

Студент считается отличником, если его средний балл >= **4.5**. Найти всех отличников в файле.

2. Гистограмма оценок

Гистограмма

Ожидаемый результат:



3. Map-Only Jobs

Map-Only Jobs

- Исходный файл - как в примере №1
- Оставить записи только по студентам с фамилией начинающейся на «П»

4. Reduce Joins

Средняя оценка vs любимый предмет

- **Файл 1** — как в первом задании
- **Файл 2** — <user>\t<любимый предмет>

Посчитать среднюю оценку среди любителей данного предмета

5. Map Joins

Map Join

Подсчет средней оценки по предмету без усреднения по пользователям

Спасибо!