

Практическое занятие №3

MapReduce – разбор ДЗ

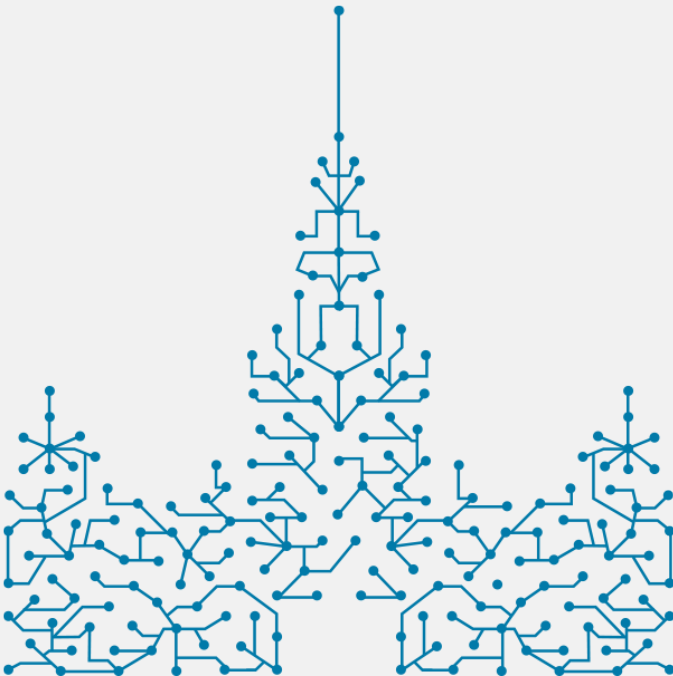
План занятия



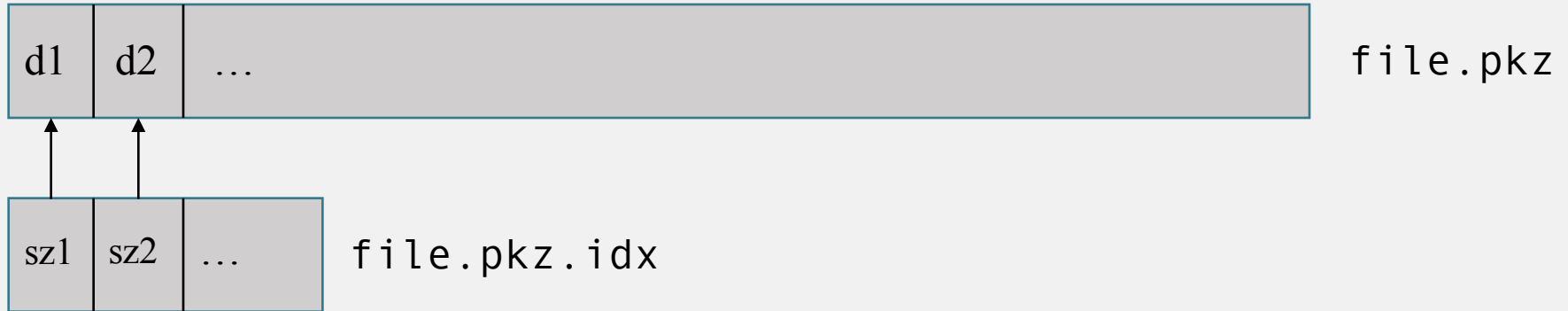
- Разбор ДЗ №1
- Еще раз про InputFormat-ы
- Решение chained задачи на map-reduce
- РК



Решение ДЗ#1



Custom Input Format



Размер `file.pkz.idx` \ll `file.pkz`

Реализовать `InputFormat` который:

1. Зачитывает `d1`, `d2`, ...
2. Позволяет дробить большие файлы на куски

Custom Input Format: этапы



1. `IndexedGzipInputFormat` **extends** `FileInputFormat<NullWritable, Text>`
2. Переопределяем `getSplits()`
3. Реализуем свой `RecordReader`:
 1. **class** `GzipBlockRecordReader` **extends** `RecordReader<NullWritable, Text>`
4. Наконец
`job.setInputFormatClass(IndexedGzipInputFormat.class);`
5. Map/Reduce ~ WordCount, быстрее с mapper-combiner

Вся разница лишь в поставке данных в `RecordReader`

RecordReader - зачитывание блоков #1



1. Считать что **bytespermap** большой и разбивает примерно на границе
2. `getSplits` просто примерно нарезает .pkz
3. `RecordReader` на `mapper`-е находит $\text{sum}(\text{len}) \geq X * \text{bytespermap}$
4. Позиционируемся на нужный `offset`

Т.е. действуем ровно как `TextInputFormat`

RecordReader - зачитывание блоков #2



1. .idx файл достаточно мал
2. А значит, его можно зачитать дважды:
 1. При разбиении (подогнать под **bytespermap**)
 2. В RecordReader-е «промотав» до нужного offset

RecordReader - зачитывание блоков #3



1. Обратить внимание на класс FileSplit
 1. Он уже сериализует путь, offset, len
 2. Вместе с этим можно передать кусок .idx
2. **class ChunkedPartsSplit extends FileSplit**
3. Заполнить ChunkedPartsSplit-ы в getSplits
4. Использовать в RecordReader-e

Существенная оптимизация

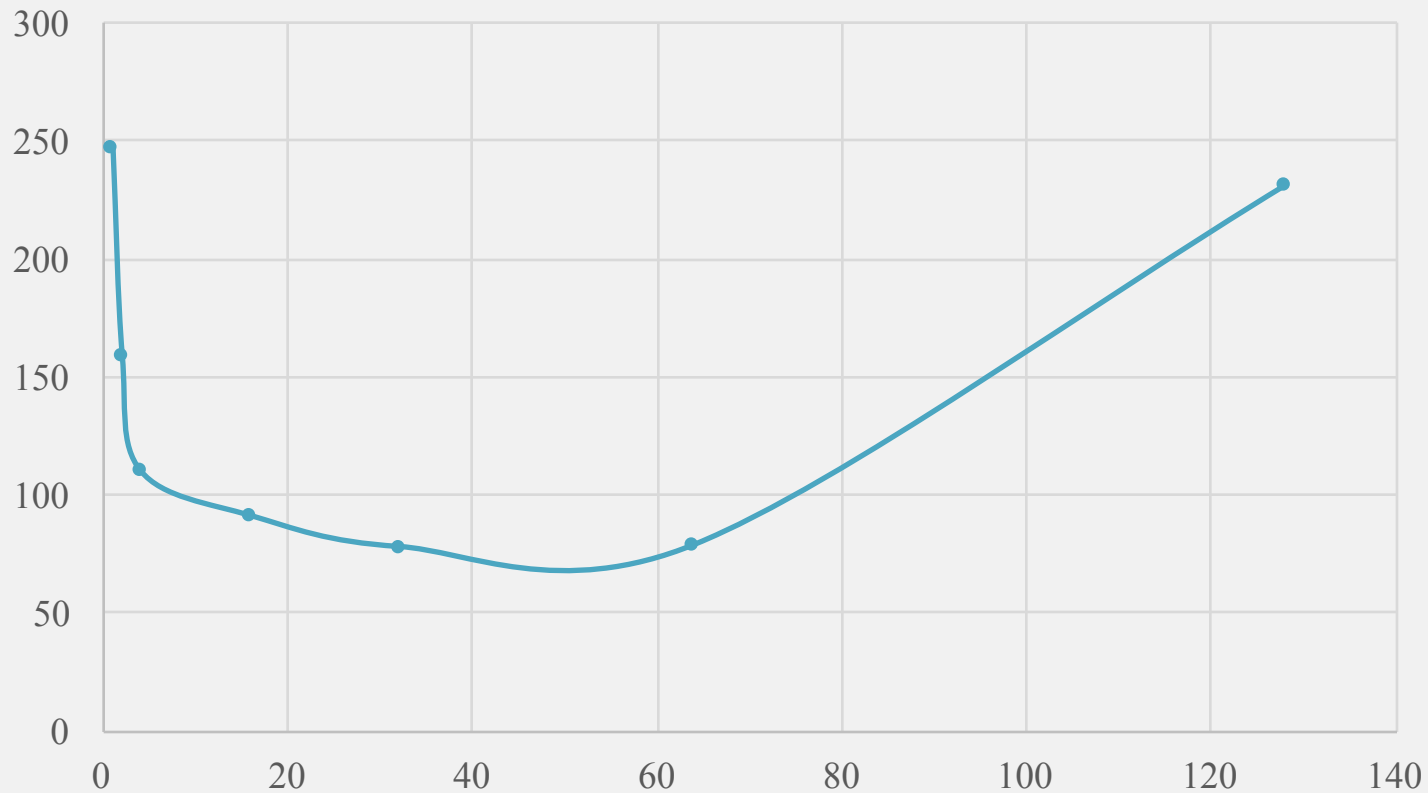


- Даже при in-mapper-combiner, у нас много док-ов содержащих одно слово:
 - Combiner сократит shuffle & reduce
 - Какая цифра для вашей реализации?

А что с bytespermap?

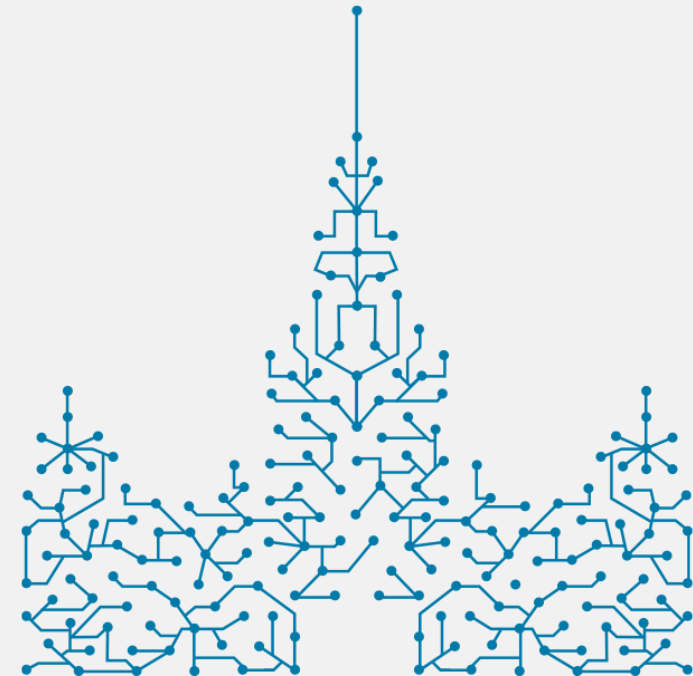


Зависимость времени работы от split-size(mb)





Еще раз про InputFormat-ы



Файлы для семинара



Семинар:

https://bitbucket.org/dkrot/hadoop_sem3

Нadoop – всегда про BigData



- Запуск JVM – 2 секунды => большой overhead
- Для выполнения task-а происходит много RPC и запусков JVM
- Поэтому работа таска должен быть:
 - Много меньше overhead-а
 - Но и не быть монолитом
- Отсюда и достаточно большой hdfs block-size
- И bytespermap в домашнем задании

Нadoop – в реальности много файлов



- Простой пример – частичные логи на физ. Машинах
- Надо статистику за 1 год
- Shuffle-стадия задействует $Nm * Nr$ копирований
- Чтобы избежать этого используется `CombineTextInputFormat`
- В streaming-задачах:
 - `org.apache.hadoop.mapred.lib.CombineTextInputFormat`

CombineTextInputFormat – попробуем



- /data/seminar3/texts/*.txt
 - 1000 файлов по 150К
- Запустим word_count без него и с ним

Streaming demo (python-streaming/):

```
$ hadoop jar /usr/lib/hadoop-mapreduce/hadoop-streaming.jar \  
-Dmapred.max.split.size=$((32*2**20)) \  
-inputformat org.apache.hadoop.mapred.lib.CombineTextInputFormat \  
-file mapper.py -file reducer.py \  
-input '/data/seminar3/texts/*.txt' -output out/wc_combined \  
-mapper mapper.py -reducer reducer.py -combiner reducer.py
```

Множество входов



- Задача – нужно отфильтровать по WhiteList
- Например, есть URL-ы и разрешенные URL-ы отдельно
- Решение:
 - White-list-ы поставить раньше с признаком
 - Далее на reduce должны идти обычные URL-ы

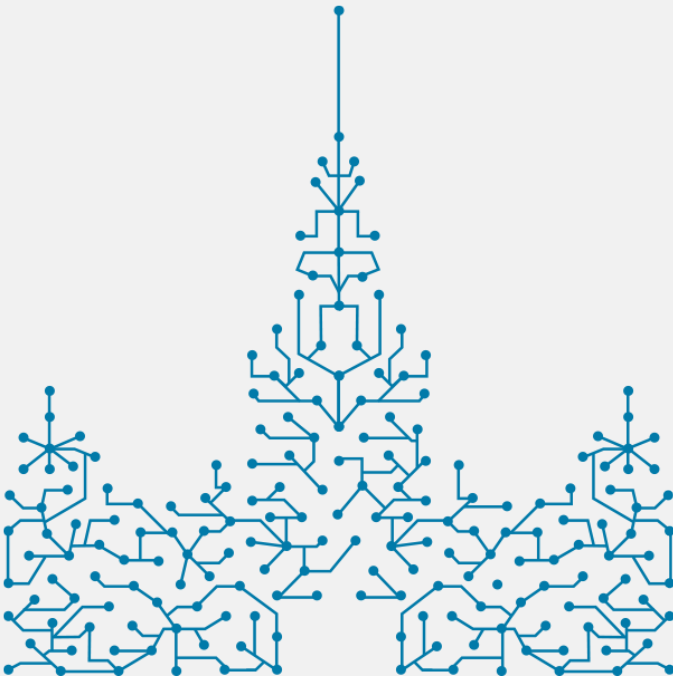
MultipleInputs



- Возможность в одной стадии map иметь несколько разнородных файлов
- Для каждого – свой mapper
- См. MultipleInputs



Chained-задачи



Подсчет относительных величин



- Классический пример – подсчет доли слов в wordcount
- 1 часть – wordcount
- 2 часть – по агрегации посчитать долю

Подсчет доли хостов в выдаче



- Задача #1:
 1. Map (urls) -> <hostname, 1>
 2. Multiple reducers: -> <hostname, cnt>
- Задача #2:
 1. Map <hostname, cnt> -> <hostname, cnt>
 1. А также частичную сумму на ЭТОМ mapper-e
 2. Single reducers: -> <hostname, %>
 1. Подумайте «как»

TODO: Write your code



1. `src/main/java/HostPercents.java`
2. Ваша задача – посчитать процент переходов на каждый из доменов в логе
 1. `/data/seminar3/access.log.small/`
 2. `/data/seminar3/access.log.big/`
3. Выбрать top:

```
$ hadoop fs -text \  
  out_dir/step2/part-r-0000 | \  
  sort -rgk 2,2
```

Пример вывода



mail.ru 59.9

hadoop.dkrot.pro 5.1

yandex.ru 5.0

...



Спасибо за внимание!
Ии...

Рубежный Контроль

