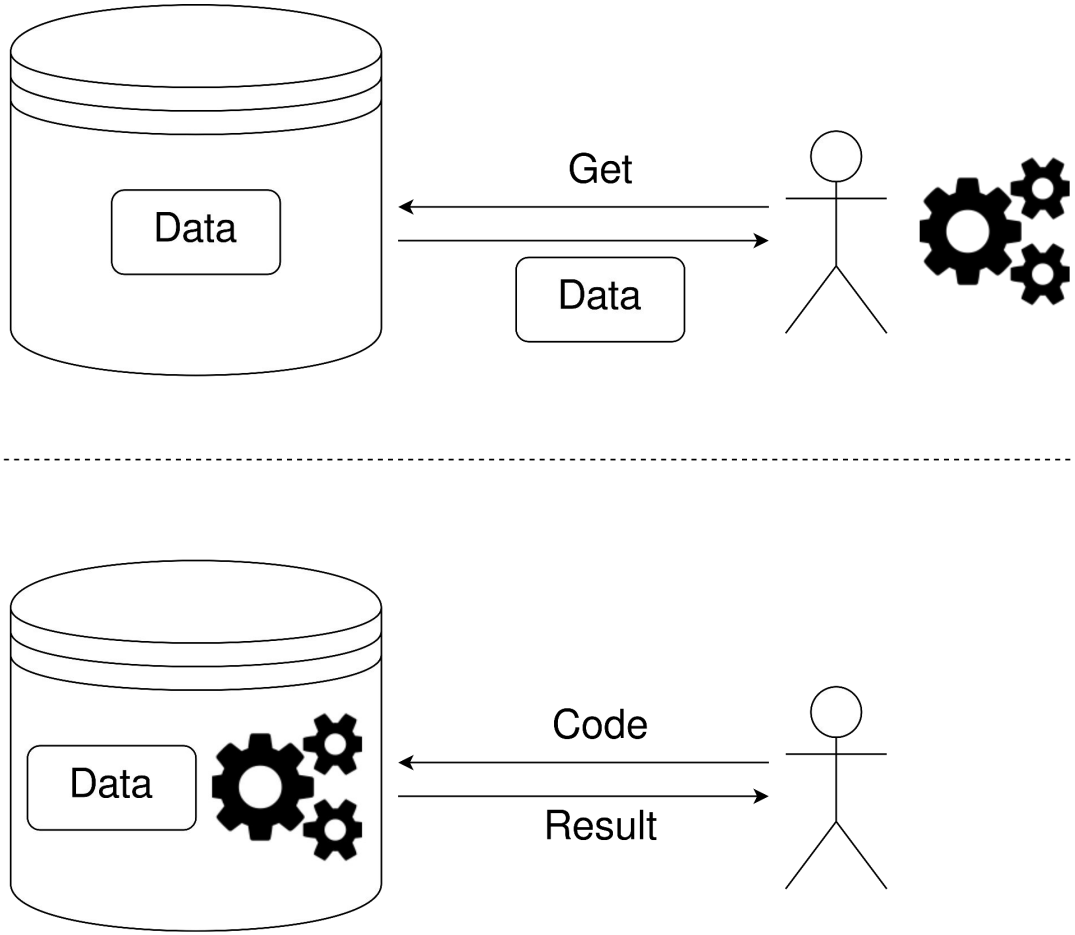
Hadoop MapReduce

Илья Кокорин

kokorin.ilya.1998@gmail.com

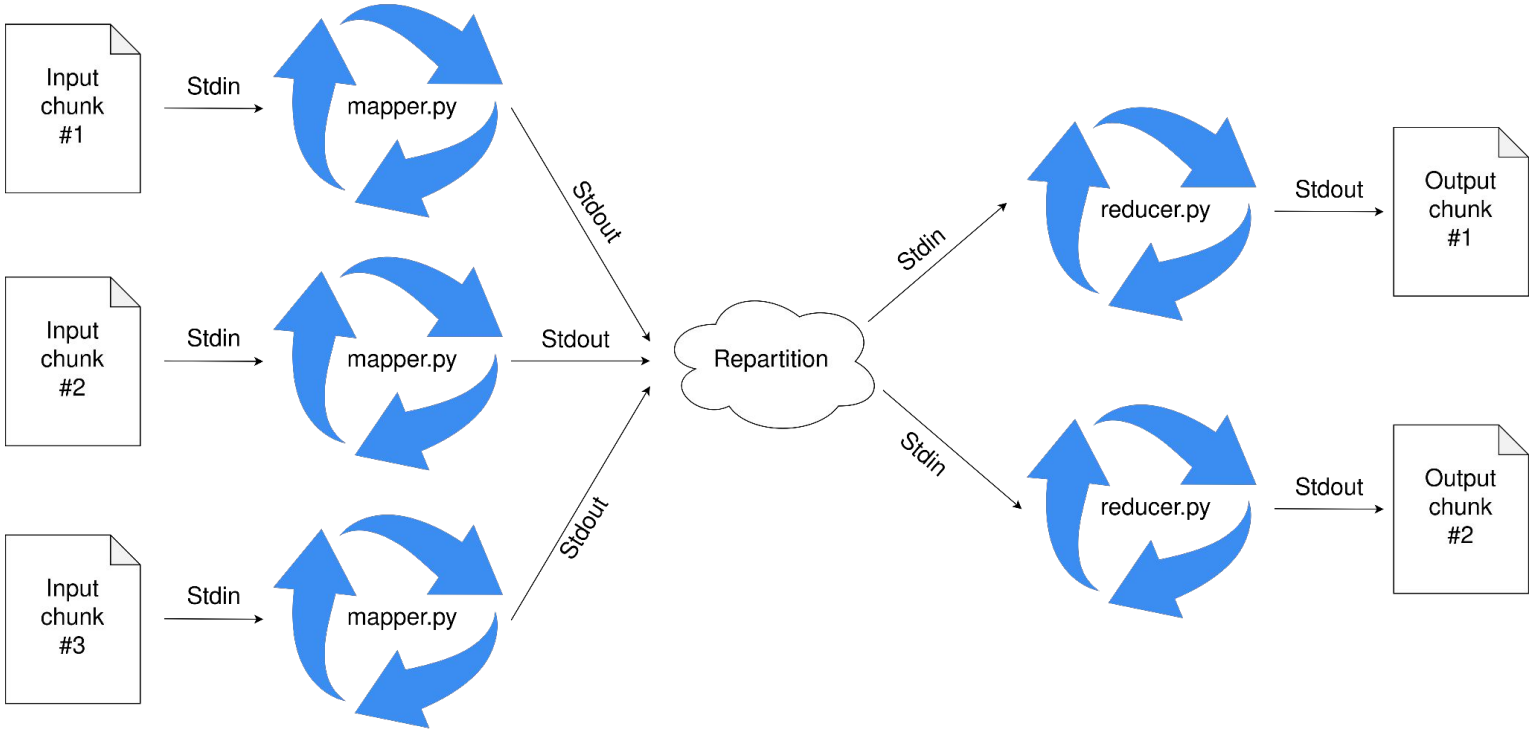
Data-to-Code vs Code-to-Data

- Привычный подход: скачать данные, обработать их
- Подход Hadoop: передать код обработки хранилищу данных
- Результат меньше данных
- Передача быстрее



Hadoop Streaming

- Пишем скрипт на любом ЯП
- Общение с MapReduce через stdin & stdout

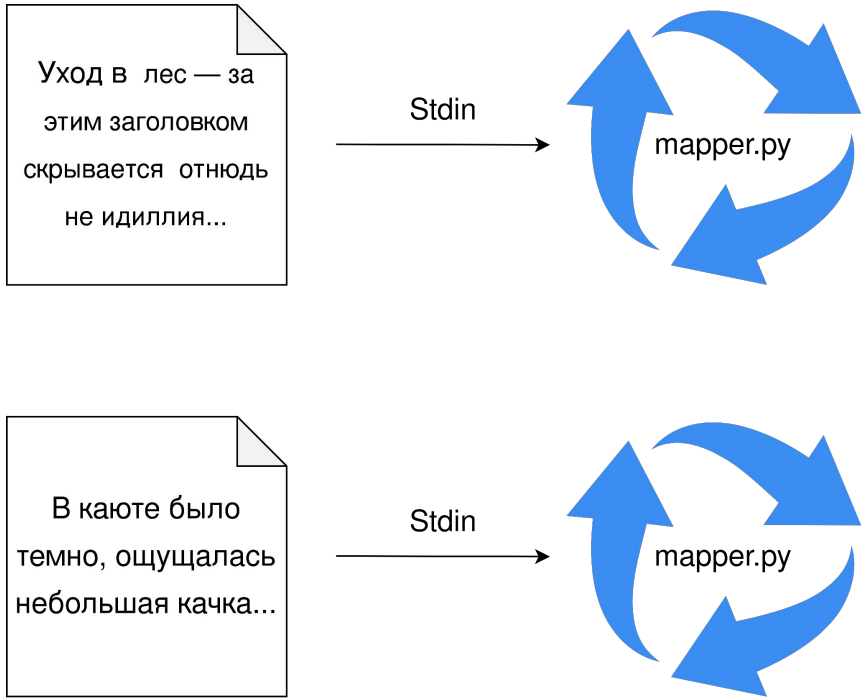


Hadoop Streaming: mapper

- Исходный документ поступает построчно
- `cat input/split-000.txt | python3 mapper.py`

```
#!/usr/bin/python3
import sys

for line in sys.stdin:
    for word in line.split():
        print(f'{word}\t1')
```

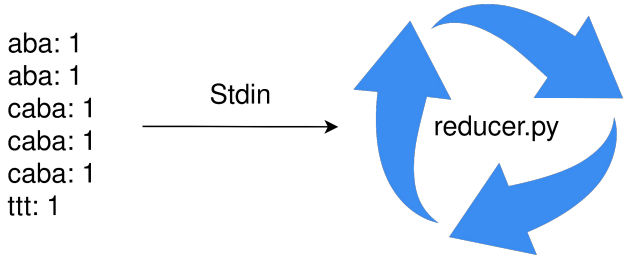


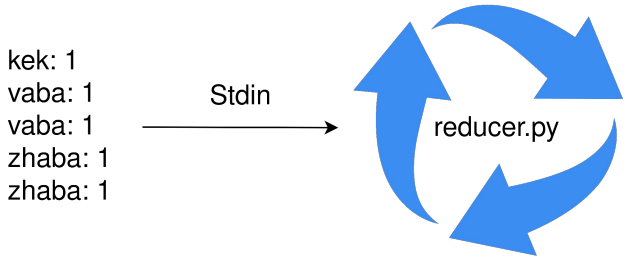
Hadoop Streaming: reducer

- Пары ключ-значение одной партиции поступают в stdin построчно в отсортированном порядке

```python
#!/usr/bin/python3
import sys

cur_word = None
cur_sum = 0
for line in sys.stdin:
    new_word, count = line.split('\t')
    if cur_word is not None and new_word != cur_word:
        print(f'{cur_word}\t{cur_sum}')
        cur_sum = 0
    cur_word = new_word
    cur_sum += int(count)
if cur_word is not None:
    print(f'{cur_word}\t{cur_sum}')
```

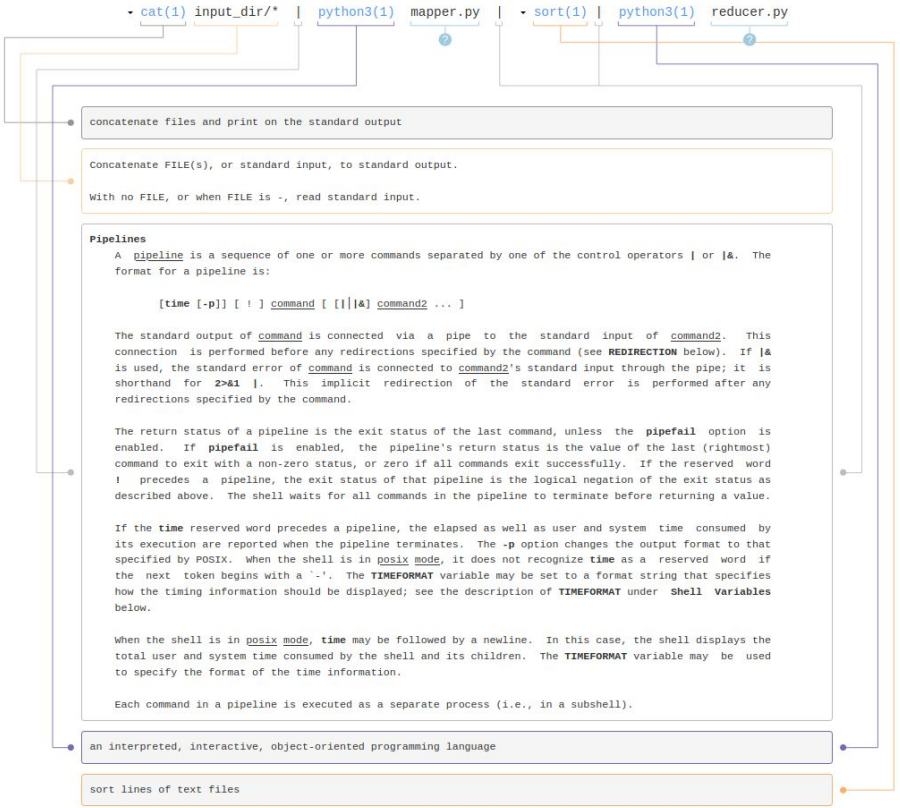




- `cat repart-data/* | sort | python3 reducer.py`

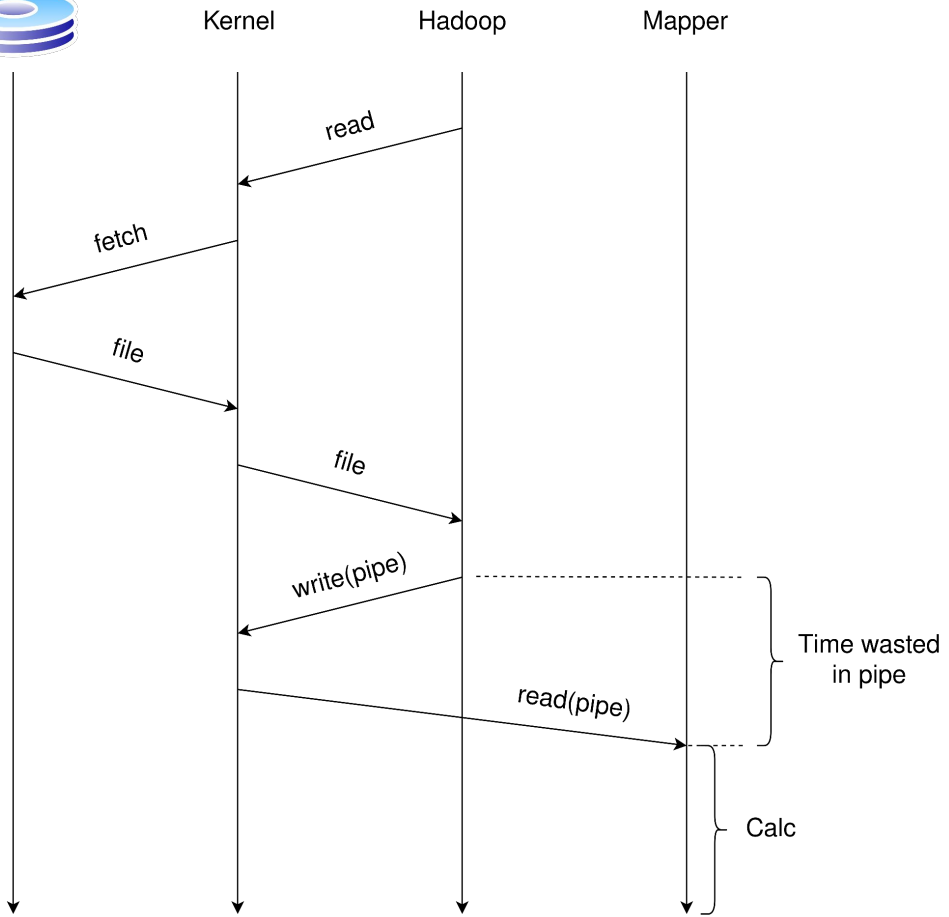
Hadoop Streaming: отладка

- `cat input/* | mapper.py | sort | reducer.py`



Hadoop Streaming: недостатки

- Данные должны лишний раз пройти через ядро ОС
- При передаче из hadoop-процесса в маппер-процесс
- Недостаток возможностей



-input directoryname or filename	Required	Input location for mapper
-output directoryname	Required	Output location for reducer
-mapper executable or JavaClassName	Required	Mapper executable
-reducer executable or JavaClassName	Required	Reducer executable
-file filename	Optional	Make the mapper, reducer, or combiner executable av
-inputformat JavaClassName	Optional	Class you supply should return key/value pairs of Text
-outputformat JavaClassName	Optional	Class you supply should take key/value pairs of Text c
-partitioner JavaClassName	Optional	Class that determines which reduce a key is sent to

Hadoop Java API

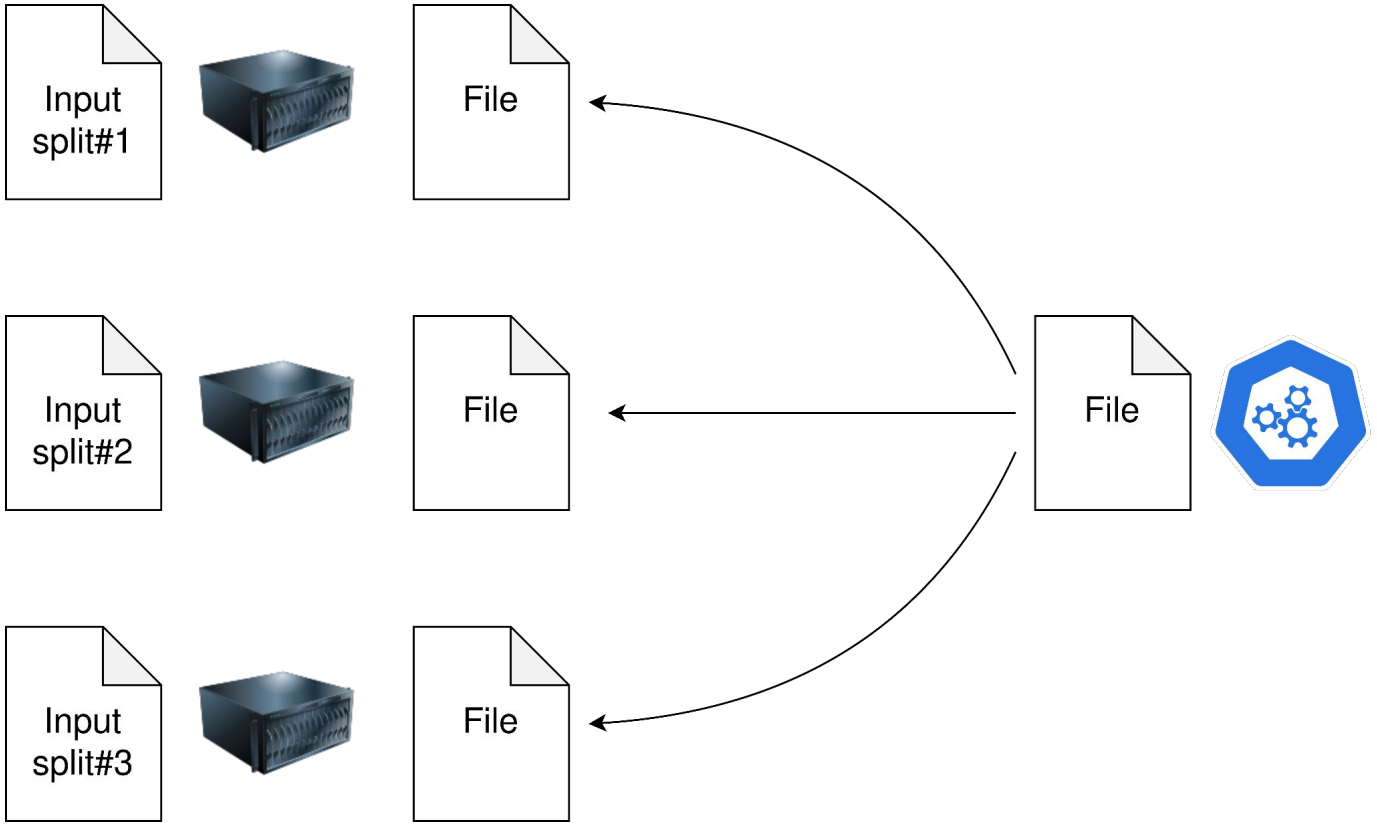
- `Mapper<IN_KEY, IN_VAL, OUT_KEY, OUT_VAL>`
  - `IN_KEY` — ключ входного файла (номер строки в файле)
  - `IN_VAL` — значения, читаемые из входного файла (строки файла)
  - `(OUT_KEY, OUT_VAL)` — тип выходных пар
-
```
void map(Long key, String value,
         Context ctx) {
   for (String word : value.split()) {
      ctx.write(word, 1);
   }
}
```

Hadoop Java API

- `Reducer<IN_KEY, IN_VAL, OUT_KEY, OUT_VAL>`
  - `(IN_KEY, IN_VAL)` — тип пар, получаемых из маппера
  - `(OUT_KEY, OUT_VAL)` — тип выходных пар
-
```
void reduce(String key, Iterable<Int> values,
            Context ctx) {
   int sum = 0;
   for (int x: values) {
      sum += x;
   }
   ctx.write(key, sum);
}
```

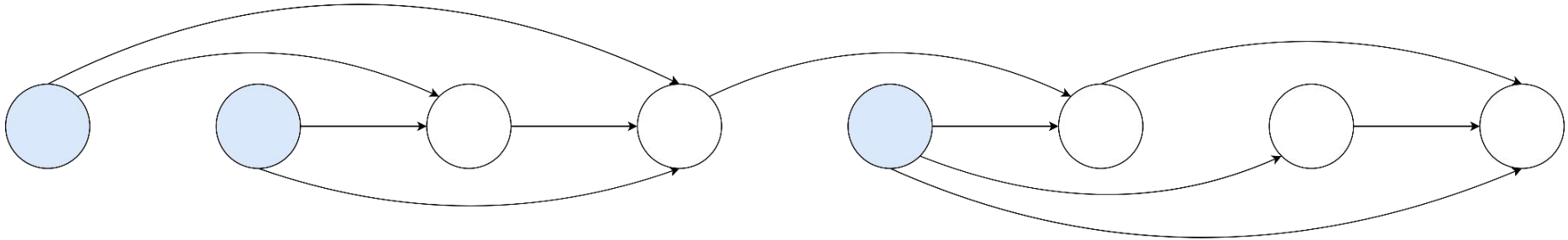
Hadoop: Distributed Cache

- Реплицируем небольшие файлики на все узлы кластера
- Map-side join
- Словари стоп-слов
- Веса моделей
- Центры кластеров
- …



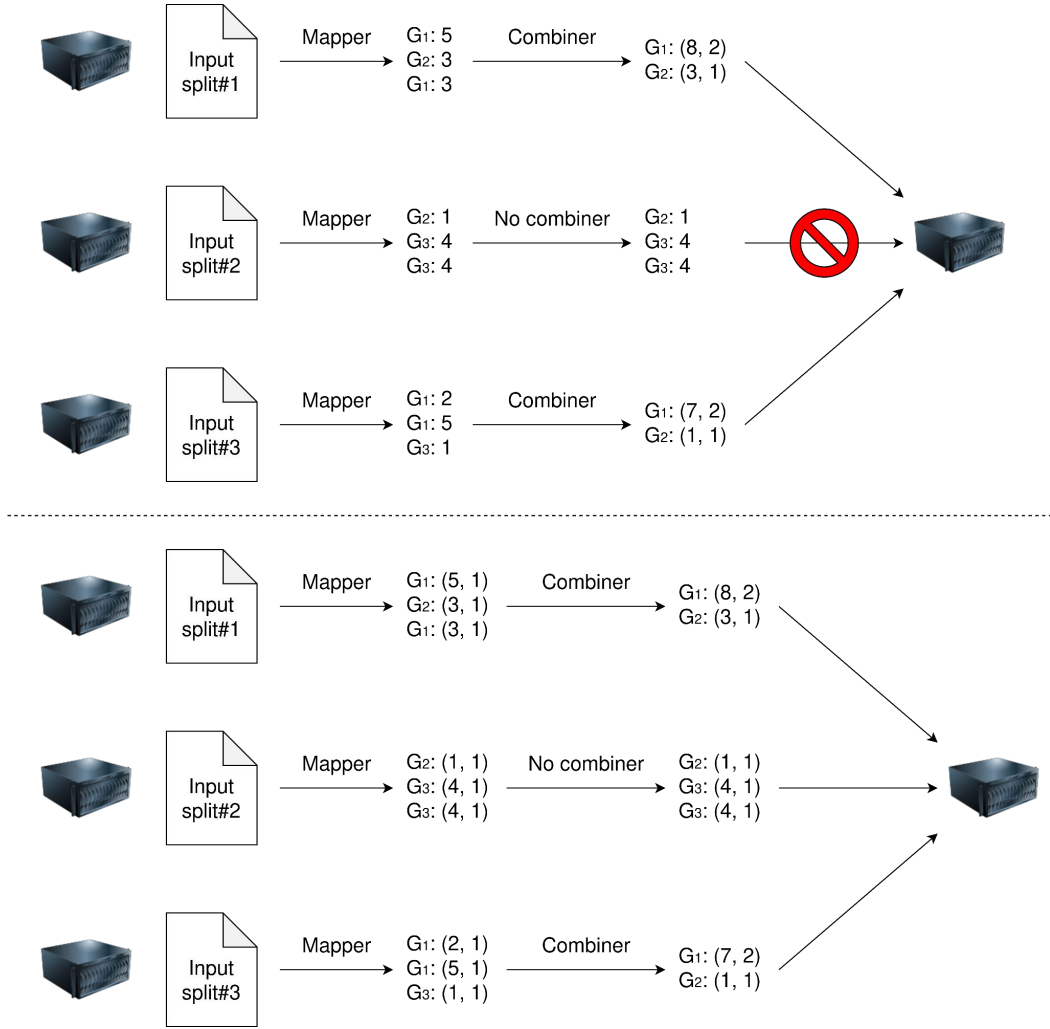
Hadoop: Job Chaining

- Есть поддержка исполнения графов задач
- `var a = new ControlledJob(..., List.of())`
- `var b = new ControlledJob(..., List.of(a))`
- `var c = new ControlledJob(..., List.of(a))`
- `var d = new ControlledJob(..., List.of(b, c))`
- `new JobControl(List.of(a, b, c, d)).run()`



Hadoop: Combiner

- Hadoop не гарантирует запуск
- На одного и того же редьюсера могут прийти данные как после combiner, так и непосредственно после маппера
- Выходные форматы маппера и combiner должны совпадать



Что почитать

- *Lam C.* Hadoop in action
- *White T.* Hadoop: The definitive guide
- *Miner D., Shook A.* MapReduce design patterns: building effective algorithms and analytics for Hadoop and other systems
- *Александр Петров.* Hadoop
- *Александр Петров.* Приемы и стратегии разработки MapReduce-приложений

Thanks for your attention