

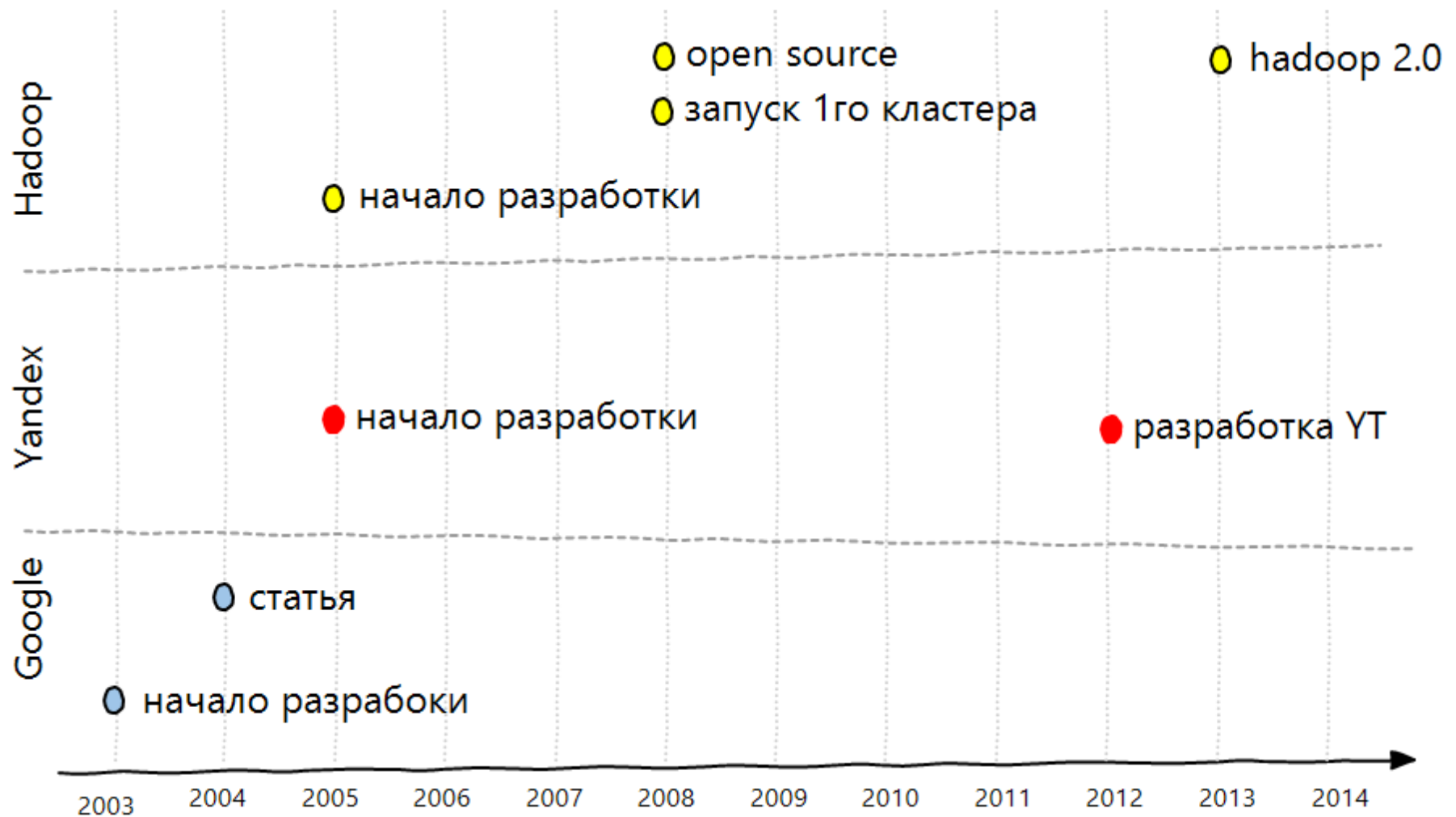
# mapReduce

Библиотека распределенной обработки данных

реализации:

- Google mapReduce
- Yandex mapReduce
- Apache Hadoop

# История



# Построение обратного индекса

Дано  
400 T

url	text
mult-online.org.ua	А Васька слушает да ест.
seowhite.org	А воз и ныне там.
koleravto.ru	А вы друзья как ни садитесь, все в музыканты не годитесь.
beautygirls.ucoz.net	А король-то — голый.

Надо:

word	url
a	i-sell.kz, alldentists.ru, abra-cadabra.ru, 23.ru, formtime.ru, megasoft.in.ua, tv-sign.ru, poiskradio.ru,.....
авось	radder.ru, myinternet.tabar.ru,
ад	bagaznik.shop.by, rozovijbegemotik.ru,
аз	daewoo-mag.ru, proletarsky.ru, ao-almatinskii-vagon.neobroker.ru,

# Подсчет слов

url	text
mult-online.org.ua	А Васька слушает да ест.
seowhite.org	А воз и ныне там.
koleravto.ru	А вы друзья как ни садитесь, все в музыканты не годитесь.
beautygirls.ucoz.net	А король-то — голый.
micsstart.com	А дарчик просто открывался.
lestnitsu.ru	и.

word	url
а	203
авось	2
ад	2
аз	3

# Выбор железа



# Выбор железа



# Выбор железа



- Дешево

# Выбор железа



- Дешево
- Линейный рост мощности



# Выбор железа



- Дешево
- Линейный рост мощности
- Вероятность отказа

# Требования

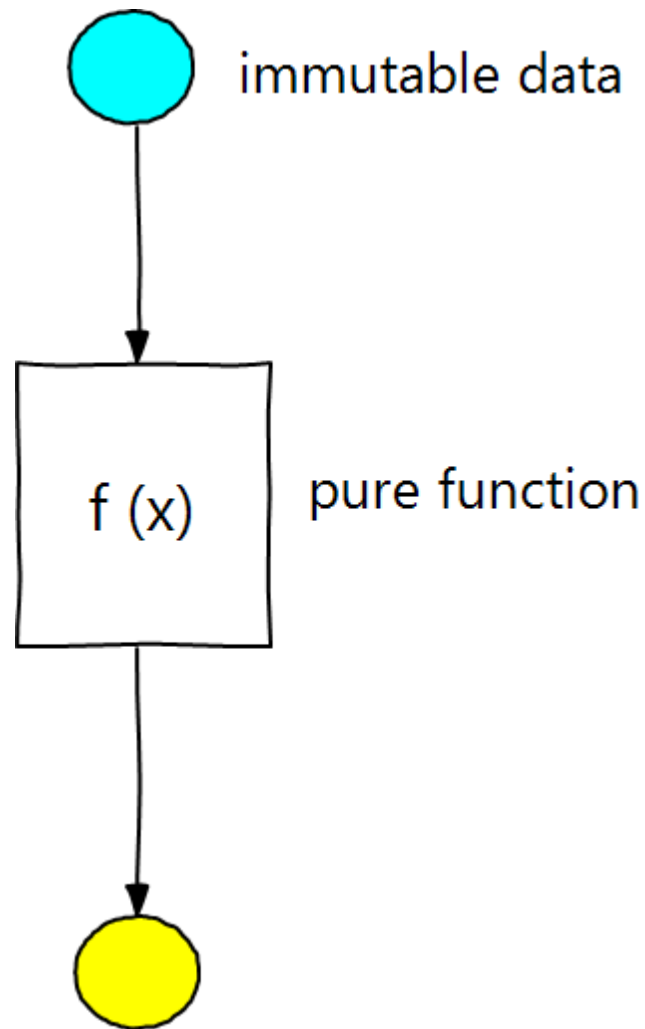
- Проста
- Распределенность
- Отказоустойчивость

# Функциональные парадигмы

- Неизменные данные
- Чистые функции

# Функциональные парадигмы

- Неизменные данные
- Чистые функции

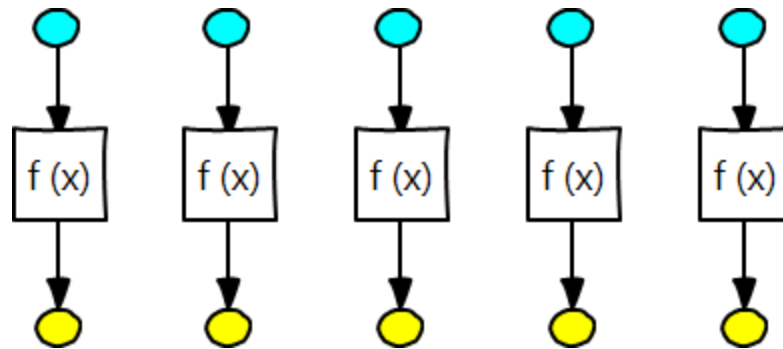


# Функции высокого порядка

# Функции высокого порядка.

## Map

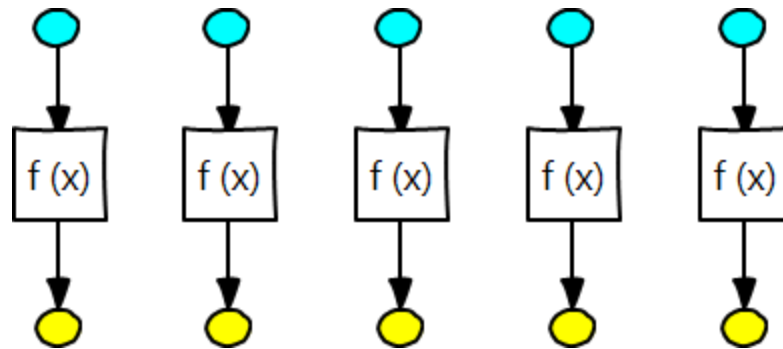
$\text{map } (f, [x]) \rightarrow [f(x)]$



# Функции высокого порядка.

## Map

$$\text{map } (f, [x]) \rightarrow [f(x)]$$



Пример:  $[x] = [1, 2, 3, 4, 5]$

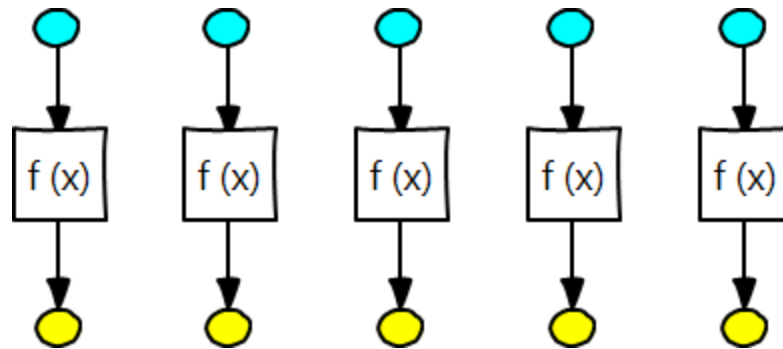
$$f(x) = x^2$$

$$\text{map } (f, [x]) \rightarrow [?]$$

# Функции высокого порядка.

## Map

$$\text{map } (f, [x]) \rightarrow [f(x)]$$



Пример:  $[x] = [1, 2, 3, 4, 5]$

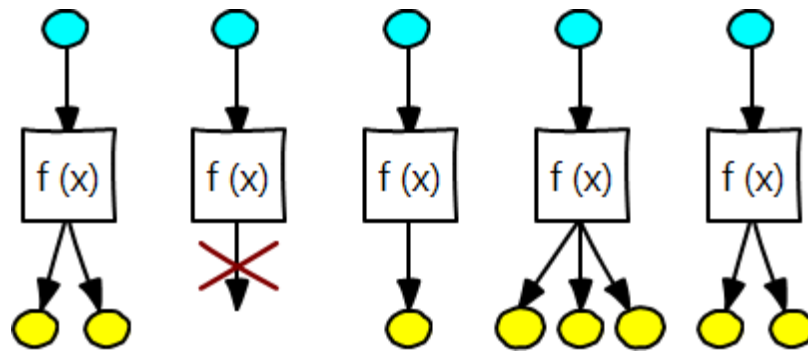
$$f(x) = x^2$$

$$\text{map } (f, [x]) \rightarrow [1, 4, 9, 16, 25]$$



# Функции высокого порядка

## Flat map



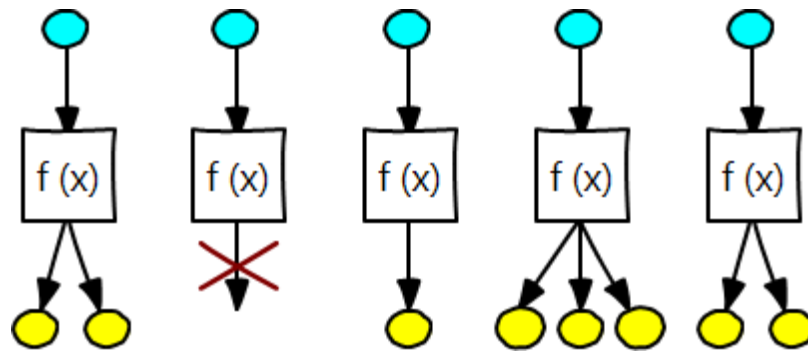
Пример:  $[x] = [1, 2, 3, 4, 5]$

$f(x) = x$  if  $x$  is simple

$\text{flat map}(f, [x]) \rightarrow ?$

# Функции высокого порядка

## Flat map



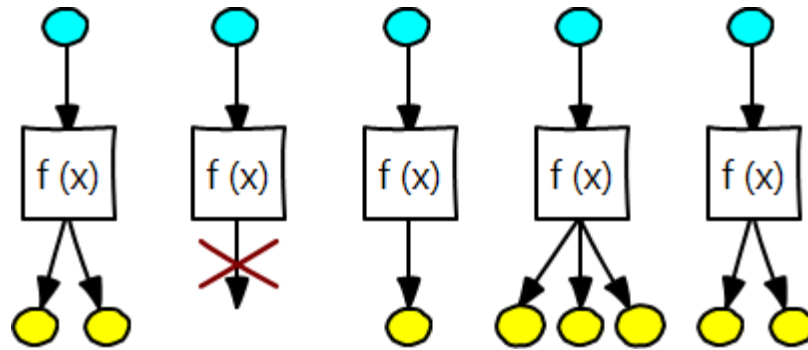
Пример:  $[x] = [1, 2, 3, 4, 5]$

$f(x) = x$  if  $x$  is simple

$\text{flat map } (f, [x]) \rightarrow [1, 2, 3, 5]$

# Функции высокого порядка

## Flat map

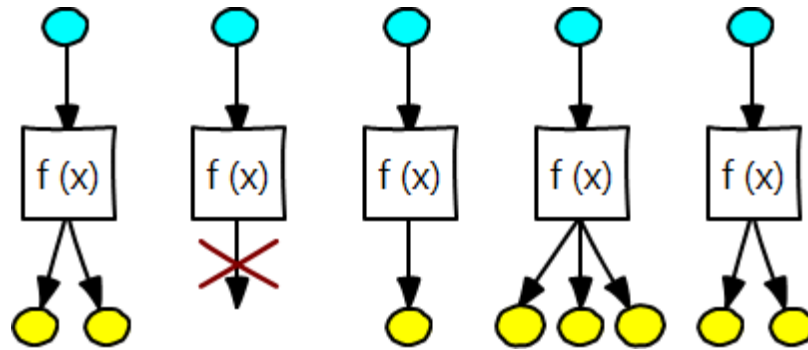


Пример:  $[x] = ['abc', 'def', 'gh']$

$f(x) = \text{split}$

$\text{map}(f, [x]) \rightarrow ?$

# Функции высокого порядка Flat map



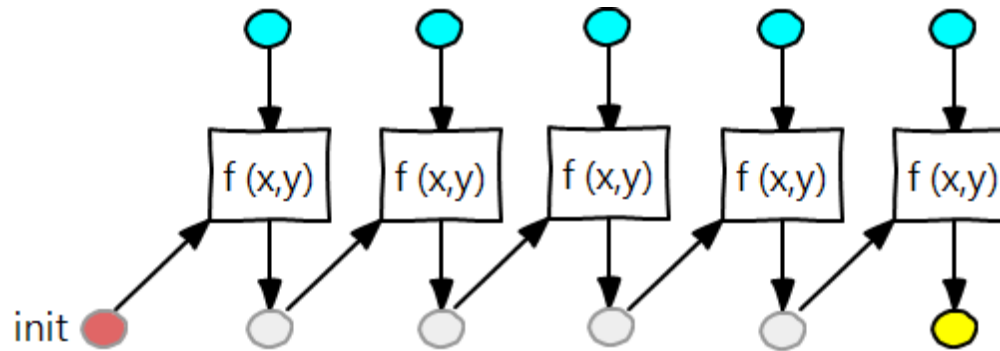
Пример:  $[x] = ['abc', 'def', 'gh']$

$f(x) = \text{split}$

$\text{map}(f, [x]) \rightarrow ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']$

# Функции высокого порядка Reduce

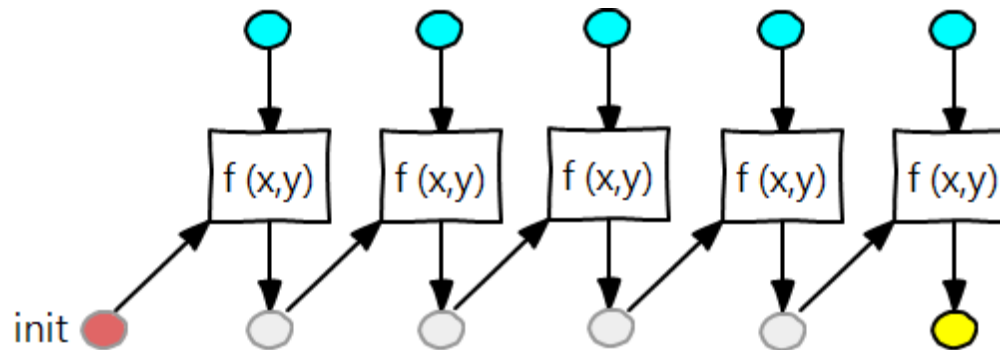
$\text{reduce}(f(x,y), [x], \text{init}) \rightarrow f(x_n, f(x_{n-1}, f(\dots, f(x_1, \text{init}))))$



# Функции высокого порядка

## Reduce

$\text{reduce}(f(x,y), [x], \text{init}) \rightarrow f(x_n, f(x_{n-1}, f(\dots, f(x_1, \text{init}))))$



Пример:  $[x] = [1,2,3,4,5];$        $\text{init} = 0$

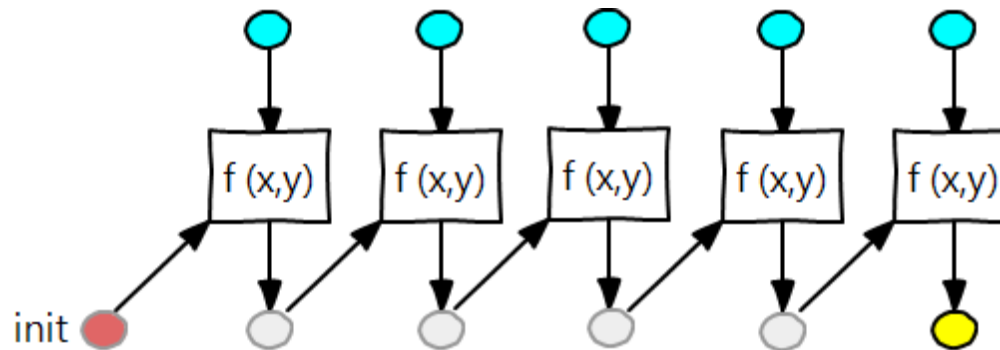
$f(x,y) = x+y$

$\text{reduce}(f, [x], \text{init}) \rightarrow ?$

# Функции высокого порядка

## Reduce

$\text{reduce}(f(x,y), [x], \text{init}) \rightarrow f(x_n, f(x_{n-1}, f(\dots, f(x_1, \text{init}))))$



Пример:  $[x] = [1,2,3,4,5];$        $\text{init} = 0$

$f(x,y) = x+y$

$\text{reduce}(f, [x], \text{init}) \rightarrow 15$

# Функции высокого порядка

## Group

$x = (k, v)$

$\text{group}([(k, v)]) \rightarrow [(k, [v])]$

Пример:

$[x] = [('a', 1), ('b', 2), ('a', 3), ('a', 4), ('b', 5)]$

$\text{group}([x]) \rightarrow ?$



# Функции высокого порядка

## Group

$x = (k, v)$

$\text{group}([(k, v)]) \rightarrow [(k, [v])]$

Пример:

$[x] = [('a', 1), ('b', 2), ('a', 3), ('a', 4), ('b', 5)]$

$\text{group}([x]) \rightarrow [('a', [1, 3, 4]), ('b', [2, 5])]$

# Функции высокого порядка

## Group

$x = (k, v)$

$\text{group}([(k, v)]) \rightarrow [(k, [v])]$

Как реализовать group?

# Функции высокого порядка

## Group

$x = (k, v)$

$\text{group}([(k, v)]) \rightarrow [(k, [v])]$

Как реализовать group?

Сортировка по ключу

# Функции высокого порядка

## Reduce By Key

`reduceByKey (f((k,v),y), [(k,v)], init)`

# Функции высокого порядка

## Reduce By Key

`reduceByKey (f((k,v),y), [(k,v)], init)`  
`group ([(k,v)])  $\rightarrow$  [(k,[v])]`

# Функции высокого порядка

## Reduce By Key

`reduceByKey (f((k,v),y), [(k,v)], init)`

`group ([(k,v)])  $\rightarrow$  [(k,[v])]`

`foreach (k,[v]) in [(k,[v])]:`

# Функции высокого порядка

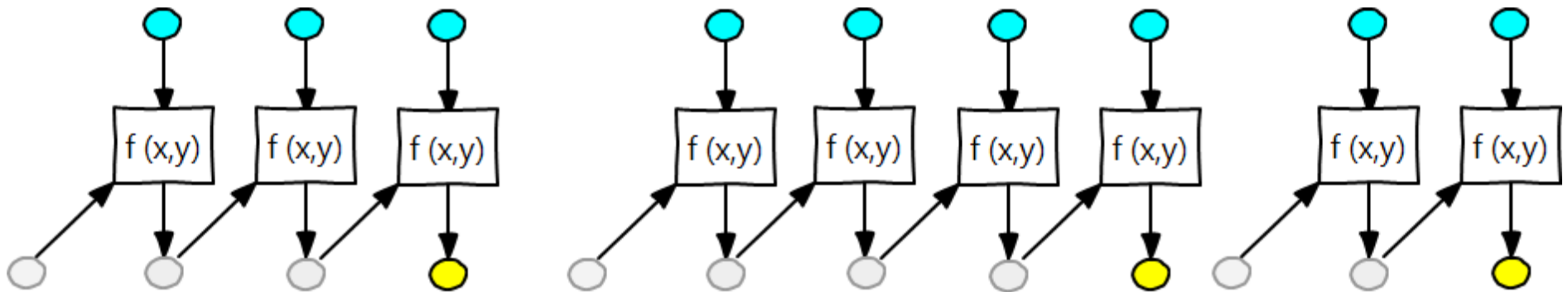
## Reduce By Key

`reduceByKey (f((k,v),y), [(k,v)], init)`

`group ([(k,v)])  $\rightarrow$  [(k,[v])]`

`foreach (k,[v]) in [(k,[v])]:`

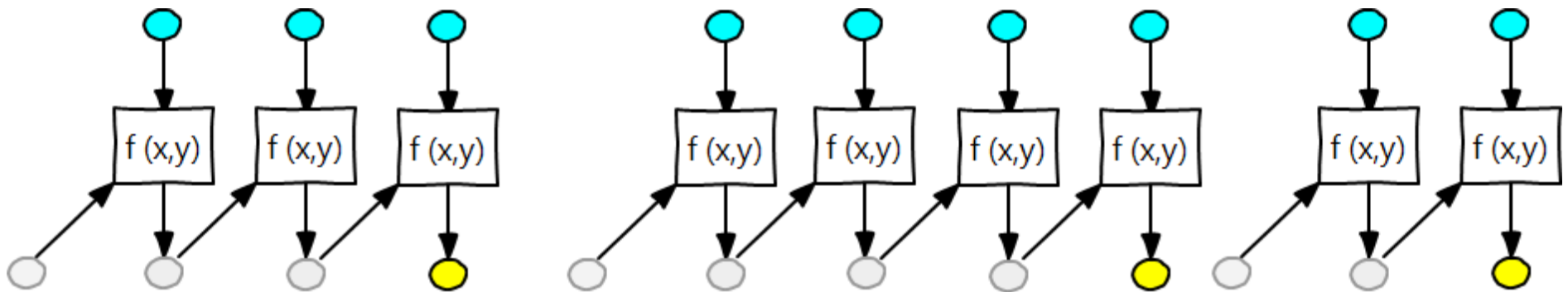
`reduce (f((k,v),y), k, [v], init)`



# Функции высокого порядка

## Reduce By Key

`reduceByKey (f((k,v),y), [(k,v)], init)`



Пример:  $[x] = [('a', 1), ('b', 2), ('a', 3), ('a', 4), ('b', 5)]$

$$f(x,y) = (x_1, x_2 + y_2)$$

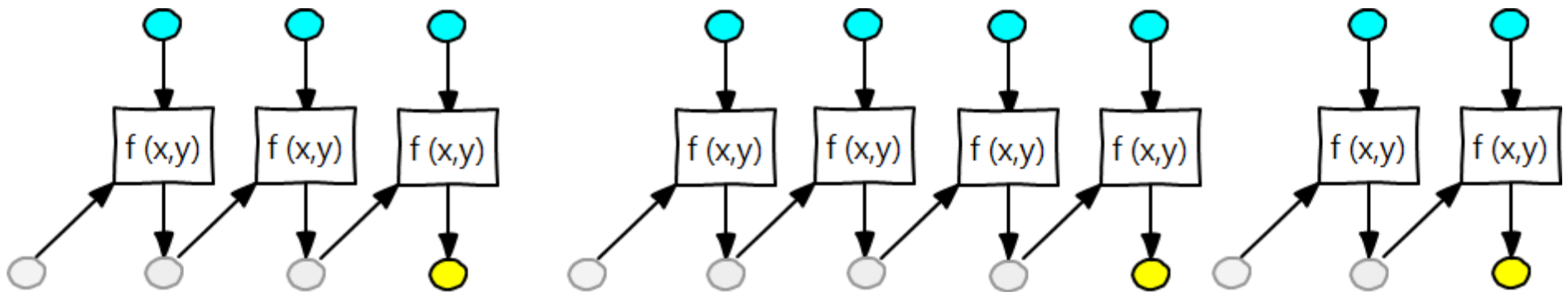
`ReduceByKey (f,[x],0)= ?`



# Функции высокого порядка

## Reduce By Key

`reduceByKey (f((k,v),y), [(k,v)], init)`



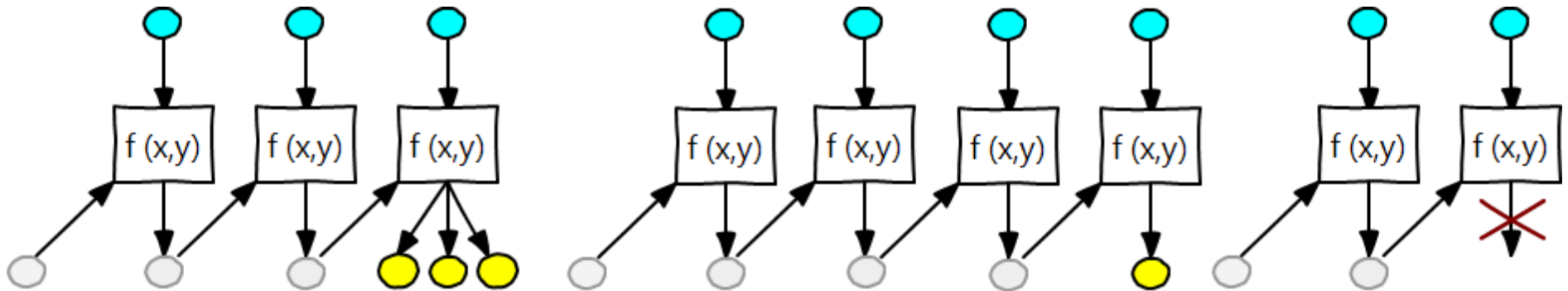
Пример:  $[x] = [(\text{'a'}, 1), (\text{'b'}, 2), (\text{'a'}, 3), (\text{'a'}, 4), (\text{'b'}, 5)]$

$$f(x,y) = (x_1, x_2 + y_2)$$

`ReduceByKey (f,[x],0)= [(\text{'a'},8), (\text{'b'},7)]`

# Функции высокого порядка

## Flat reduce By Key



# Логика mapReduce

- входные данные  $[(K1, V1)]$
- $\text{map}: (K1, V1) \rightarrow [(K2, V2)]$
- $\text{reduce}: (K2, [V2]) \rightarrow [(K3, V3)]$

# Решение задачи инвертированного индекса

Вход: (url, text)

Map: 

```
def map (url, text):  
    words = text.split (" ")  
    for w in words:  
        submit (w, url)
```

Промежуточные ключи: (word, url)

Reduce: 

```
def reduce (word, urlsIterator):  
    urls = join (" ", urlsIterator)  
    submit (w, urls)
```

Выходные ключи: (word, urls )

# Решение задачи подсчета слов

Вход: (url, text)

Map: 

```
def map (url, text):  
    words = text.split (" ")  
    for w in words:  
        submit (w, 1)
```

Промежуточные ключи: (word, count)

Reduce: 

```
def reduce (word, numsIterator):  
    total = sum (numsIterator)  
    submit (w, total)
```

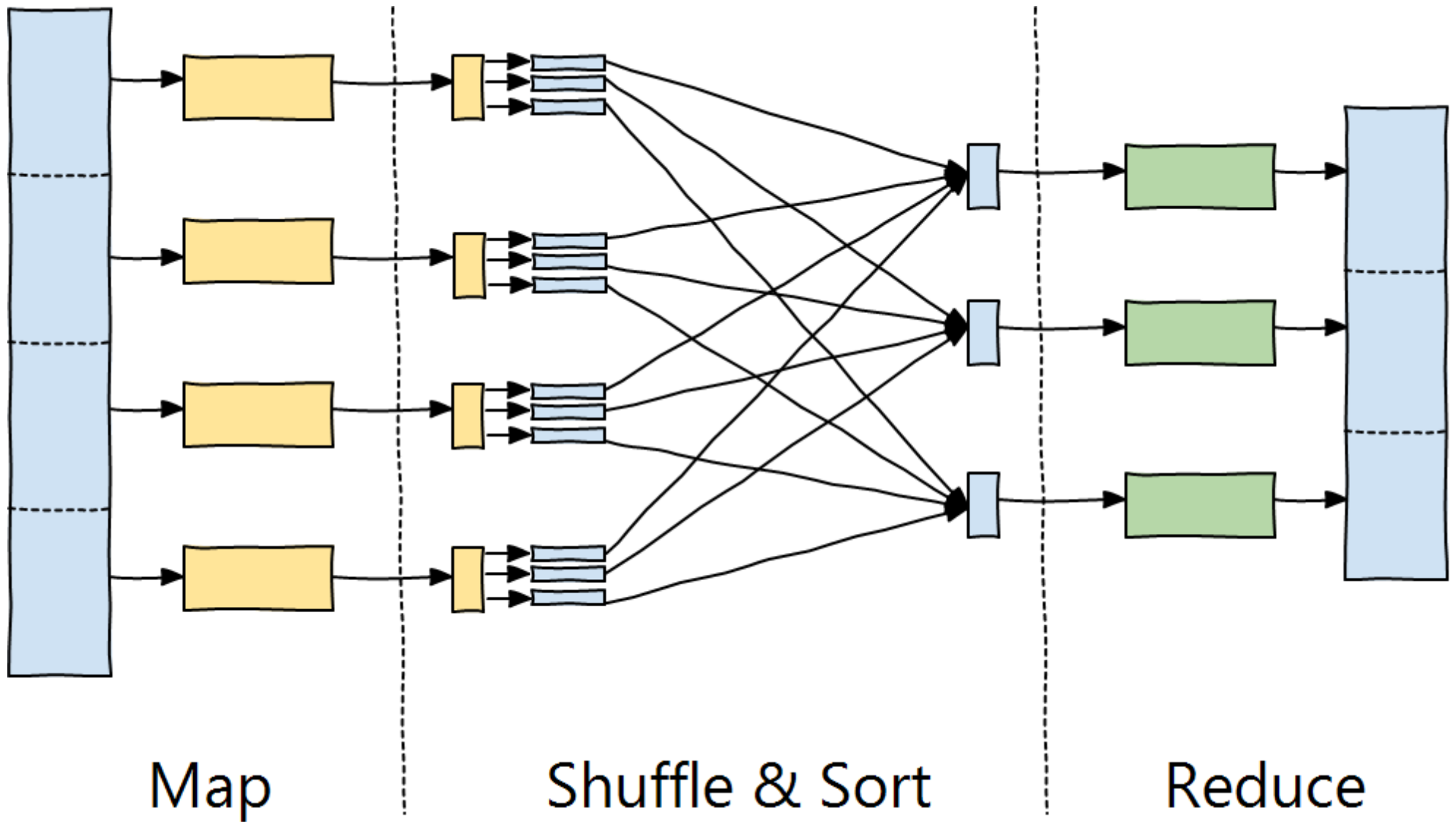
Выходные ключи: (word, count )

# Терминология mapReduce

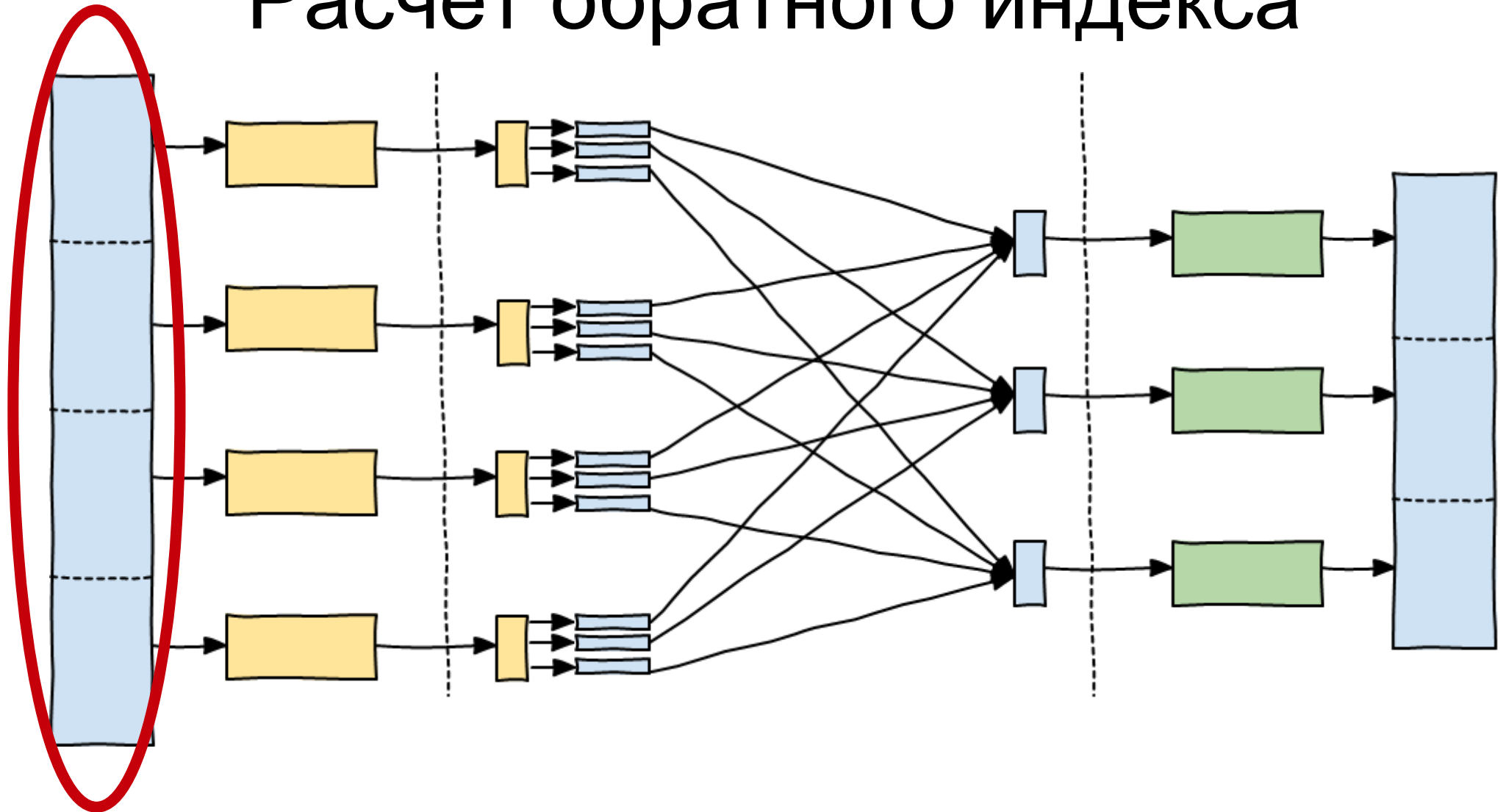
**Job** — выполняемое mapReduce задание

**Task** — отдельная map или reduce задача,  
над порцией данных

# Физическая схема mapReduce



# Расчет обратного индекса

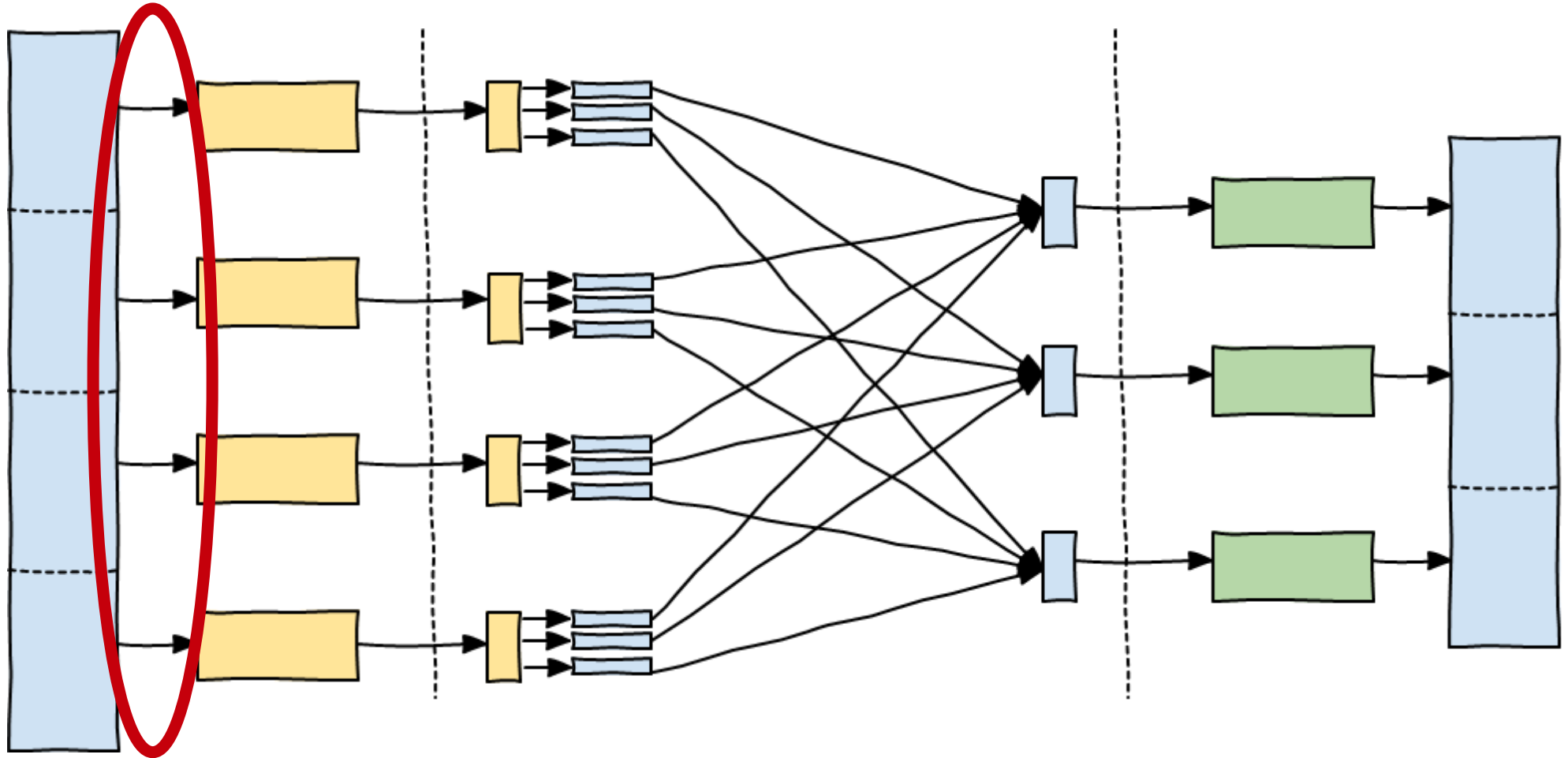


mult-online.org.ua  
seowhite.org  
koleravto.ru

А Васька слушает да ест.  
А воз и ныне там.  
А вы друзья как ни садитесь, ...

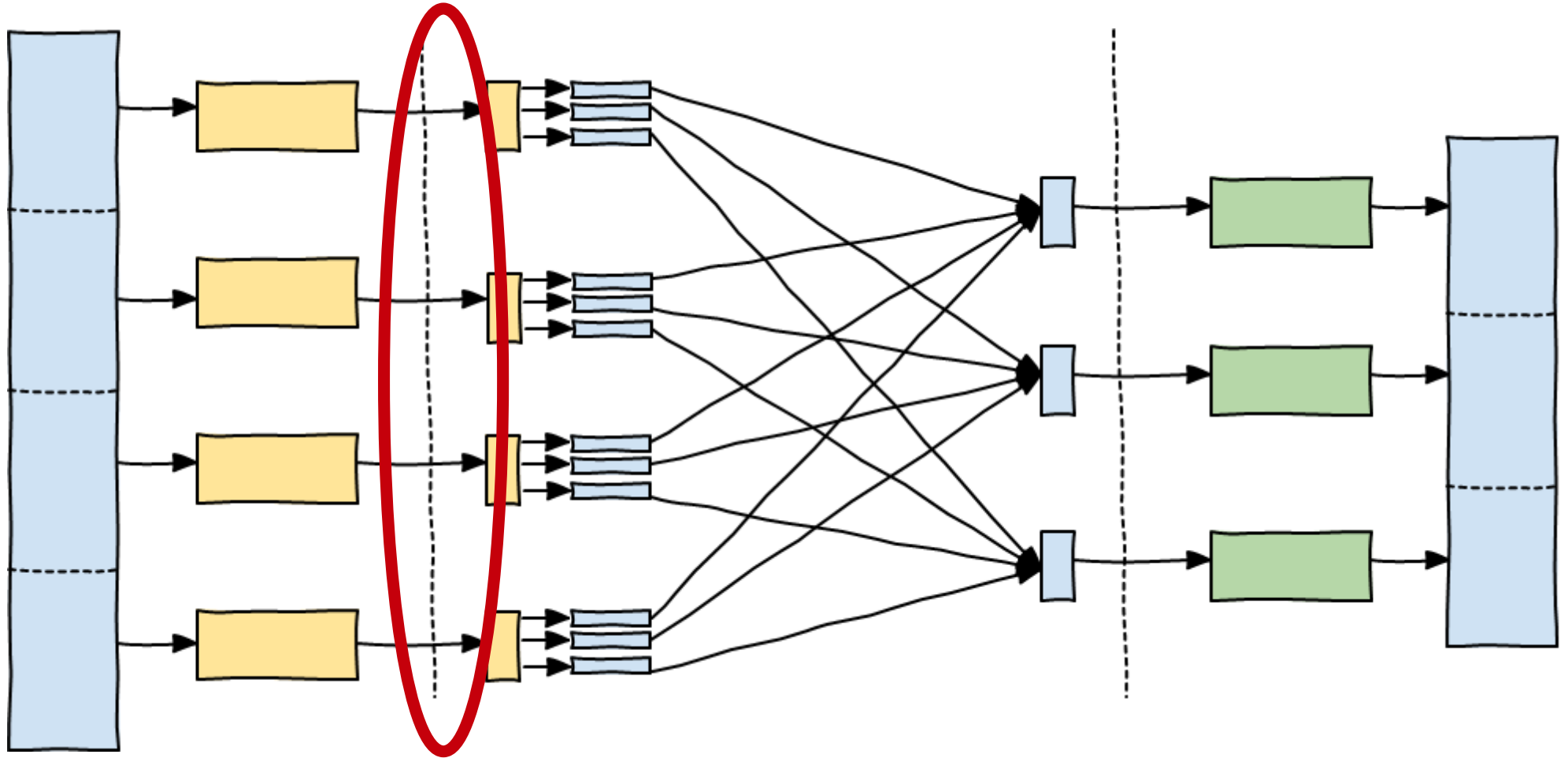


# Расчет обратного индекса



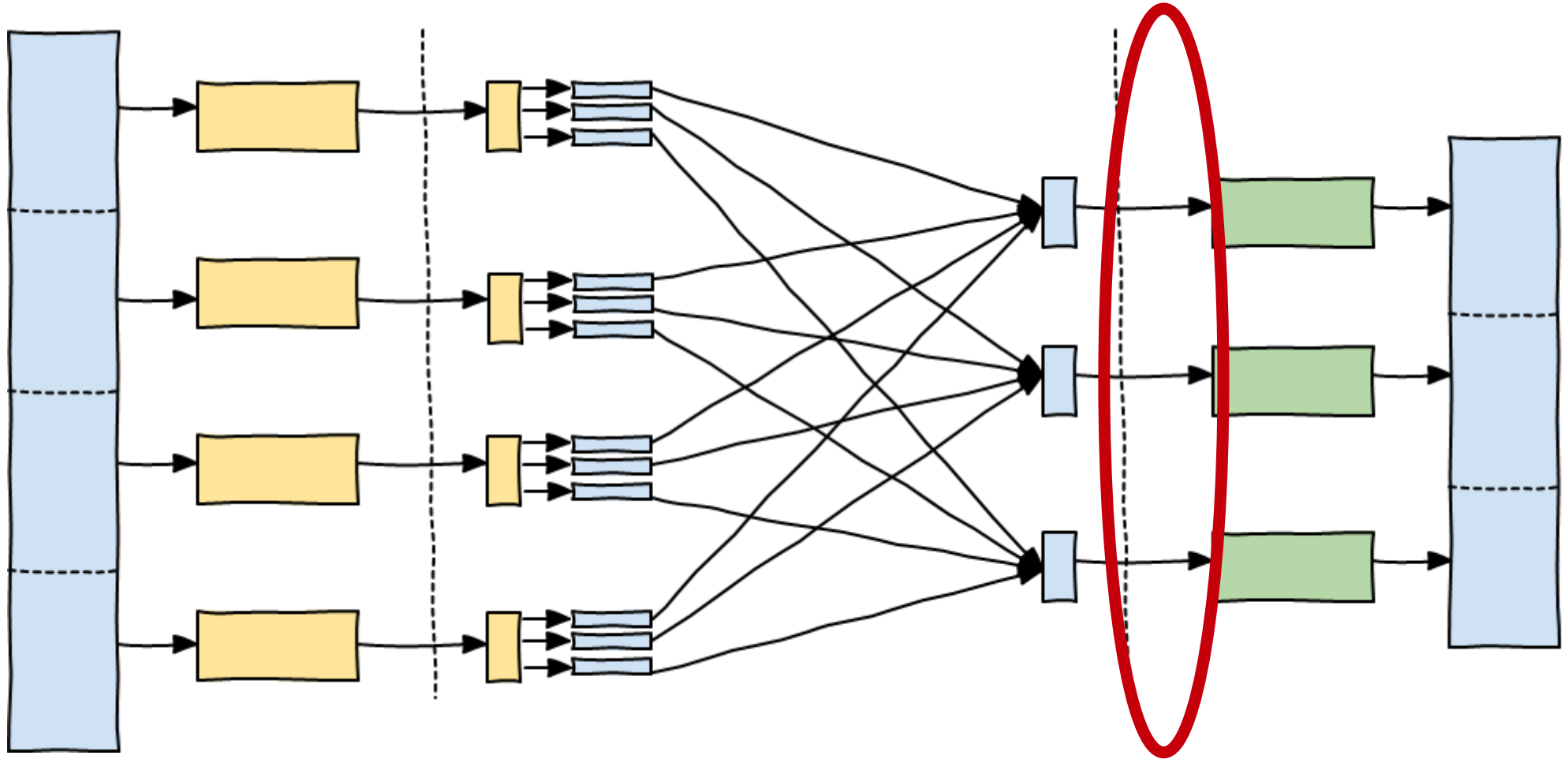
(0,	"mult-online.org.ua	А Васька слушает да ест")
(100,	"seowhite.org	А воз и ныне там.")
(213,	"koleravto.ru	А вы друзья как ни садитесь, ...")

# Расчет обратного индекса



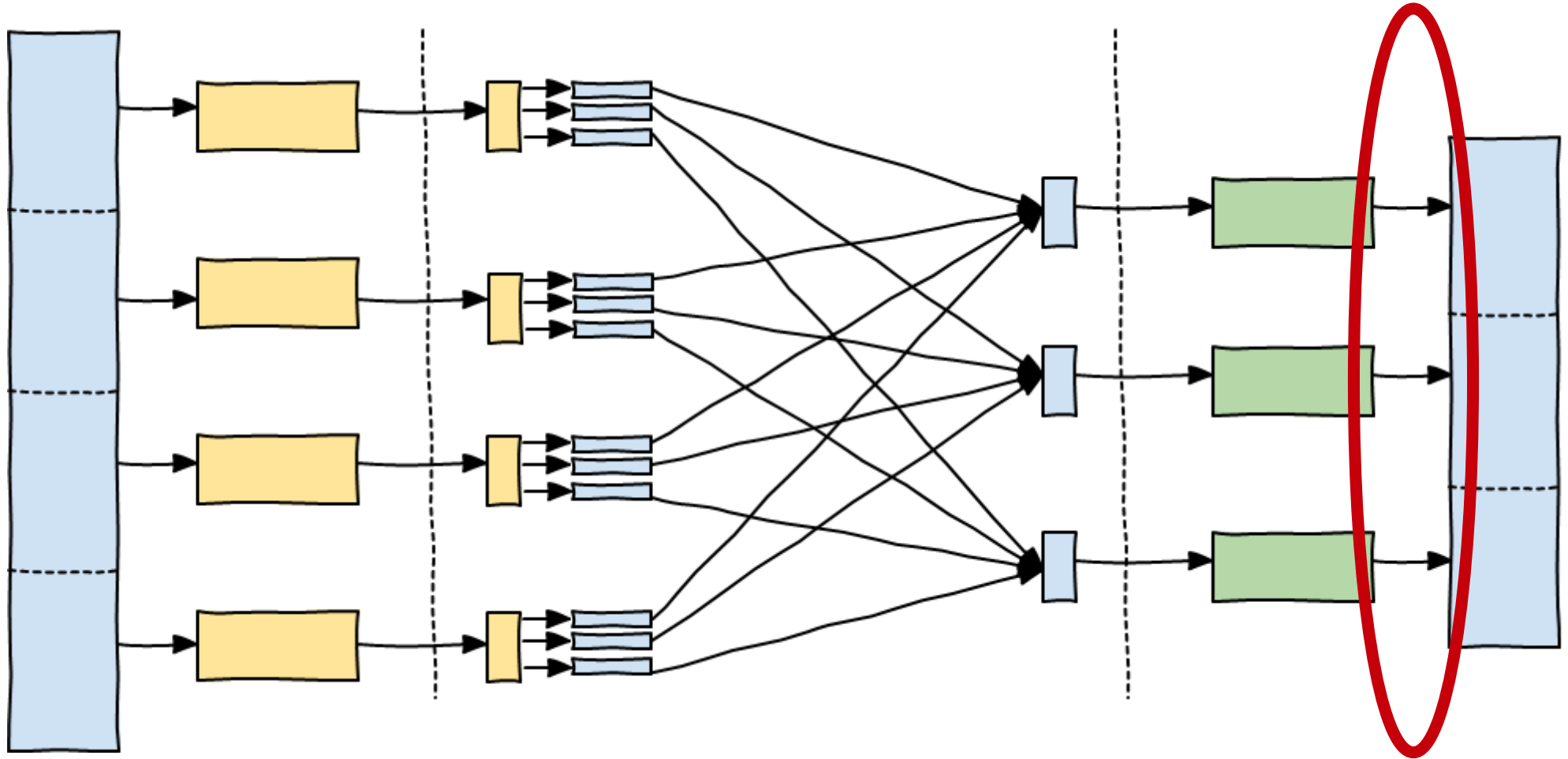
("а", "mult-online.org.ua")  
("васька", "mult-online.org.ua")  
("слушает", "mult-online.org.ua")

# Расчет обратного индекса



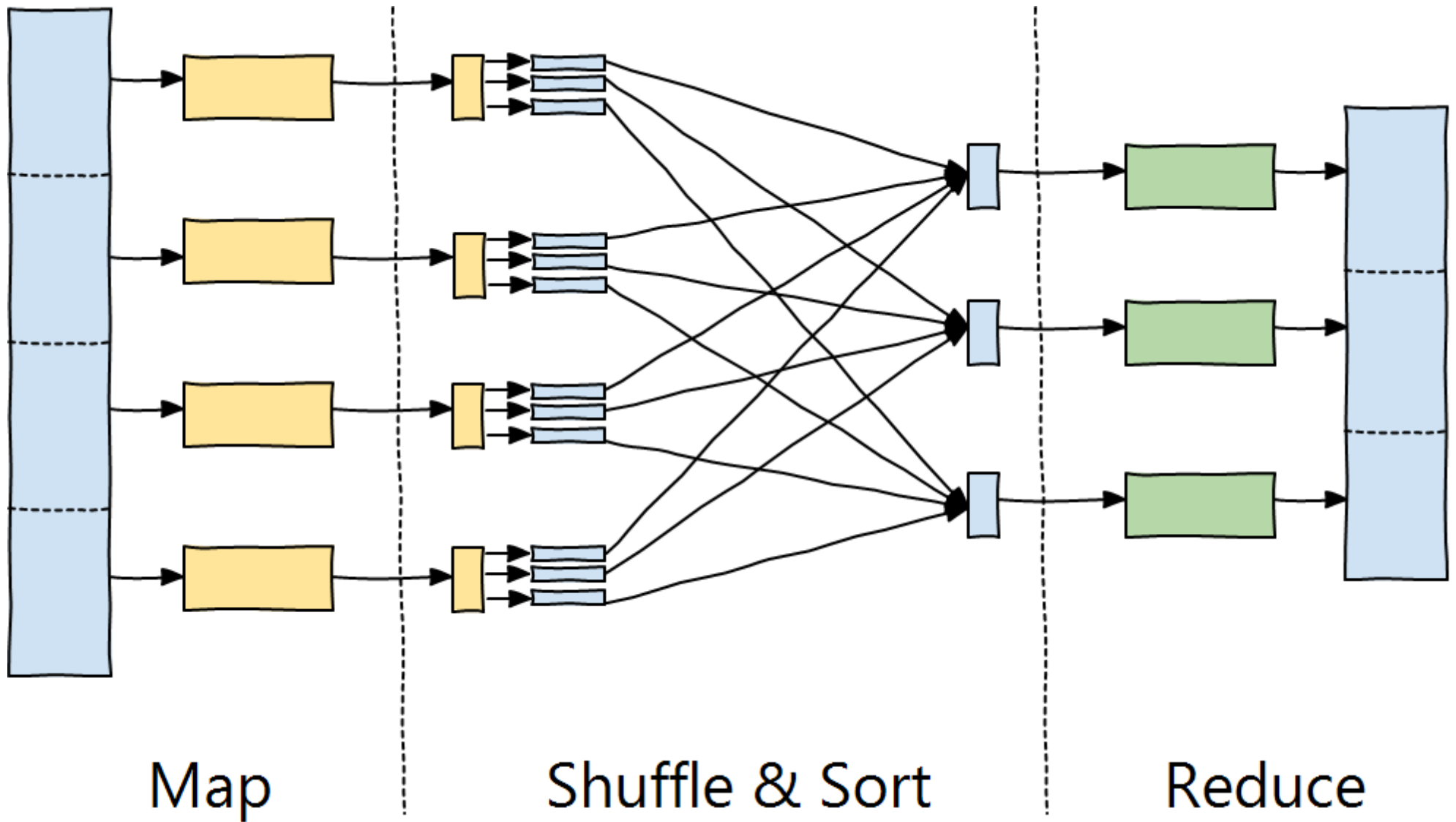
```
("a", ["mult-online.org.ua", "i-sell.kz", "alldentists.ru", ...])  
("васька", ["mult-online.org.ua"])  
("слушает", ["mult-online.org.ua", "game-deutsch.pusku.com"])
```

# Расчет обратного индекса

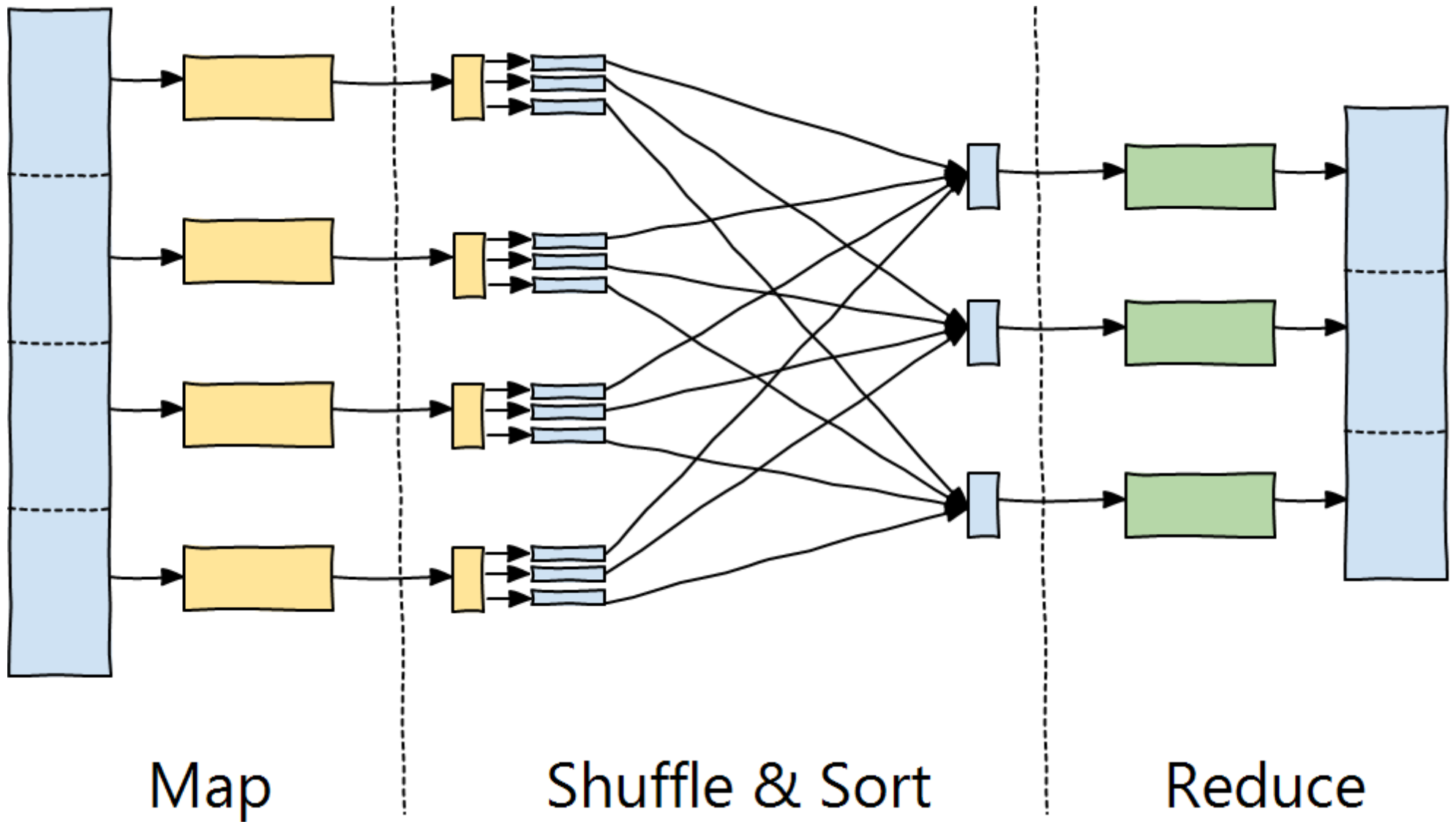


("a", "mult-online.org.ua, i-sell.kz, alldentists.ru, ...")  
("васька", "mult-online.org.ua")  
("слушает", "mult-online.org.ua", "game-deutsch.pusku.com")

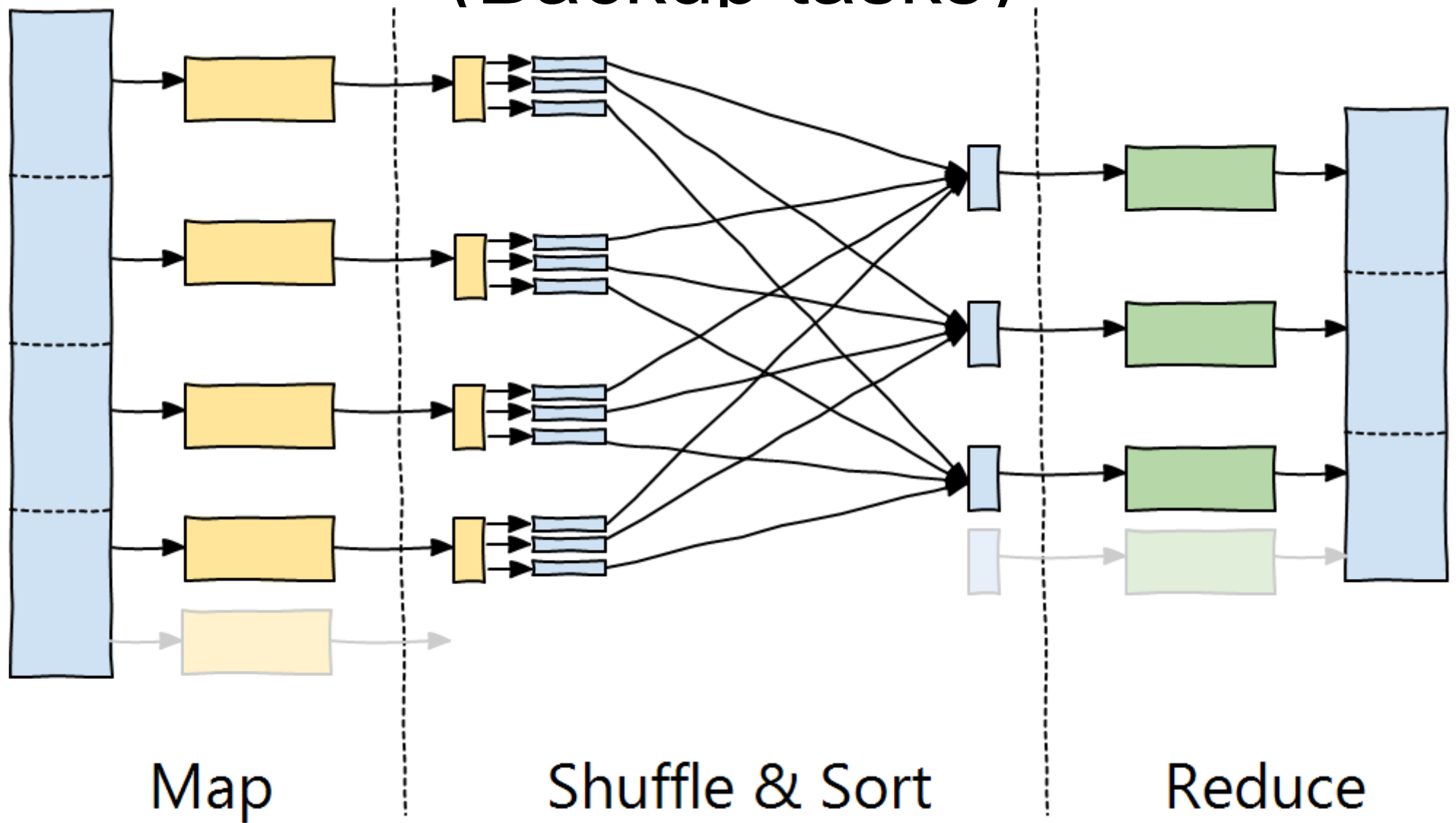
# Локальность данных



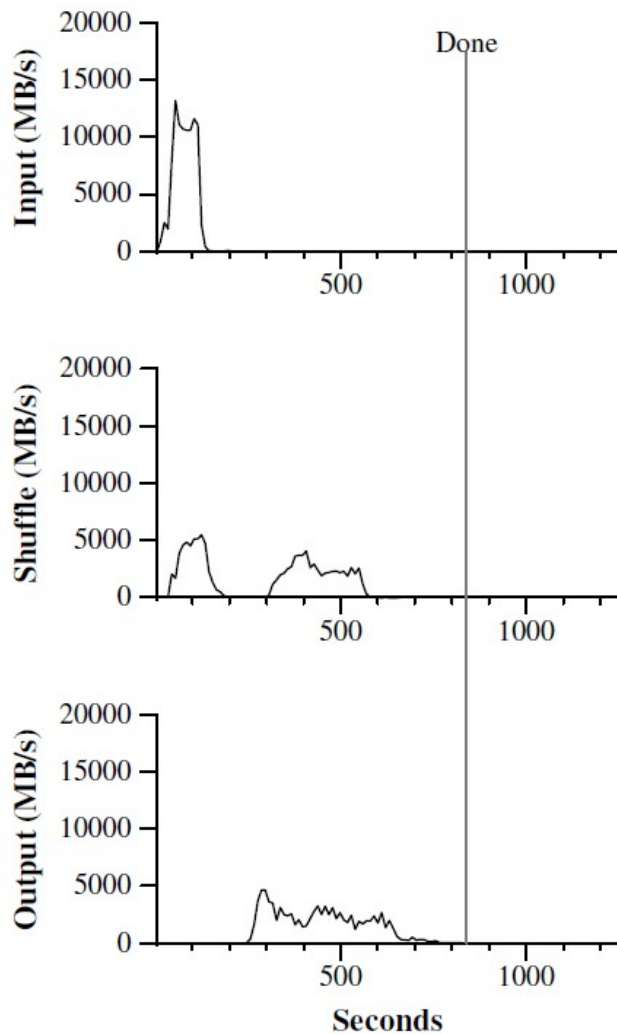
# Отказоустойчивость



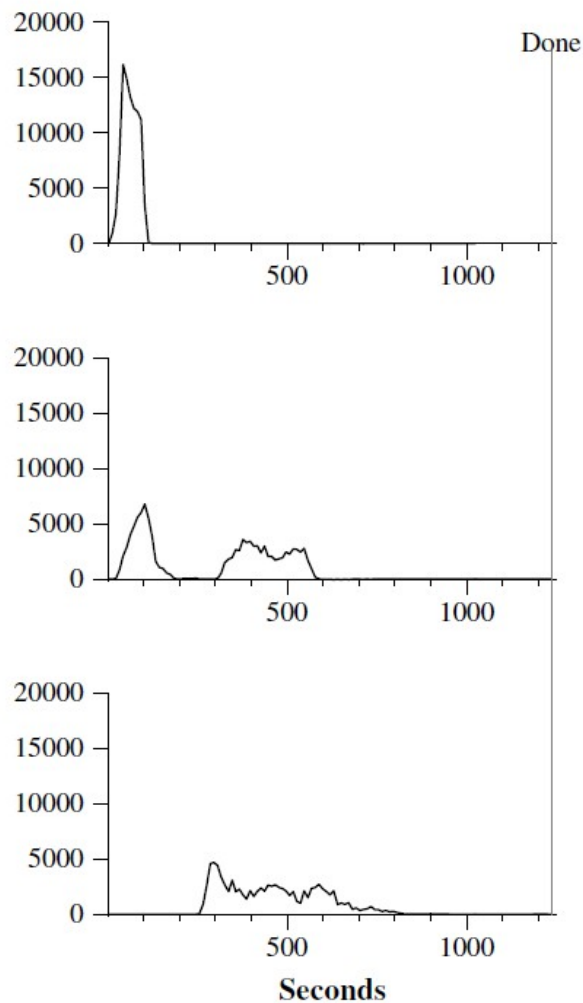
# Speculative Execution (Backup tasks)



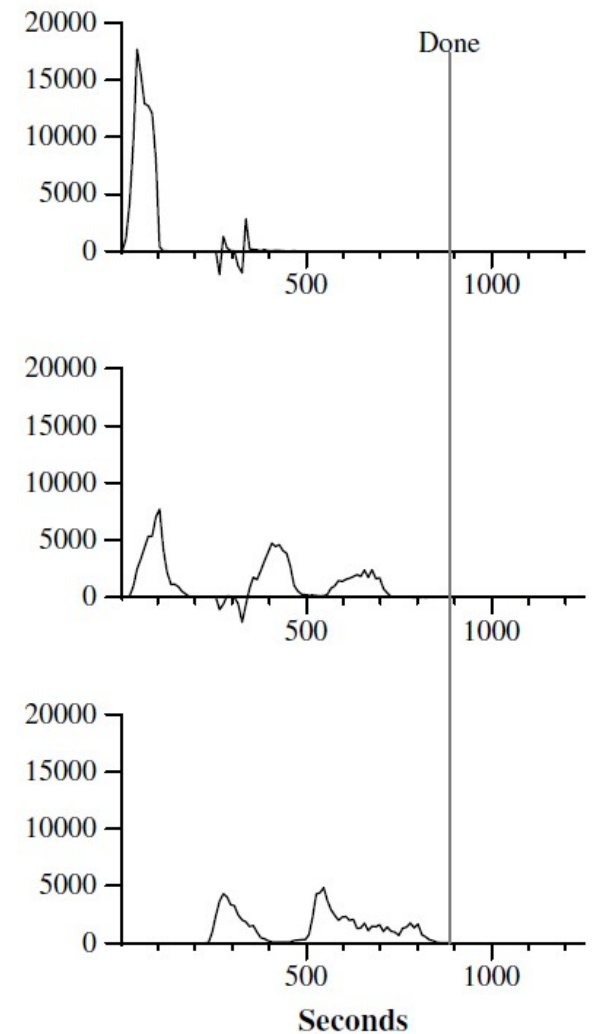
# Замер производительности



Обычный режим, 851 с



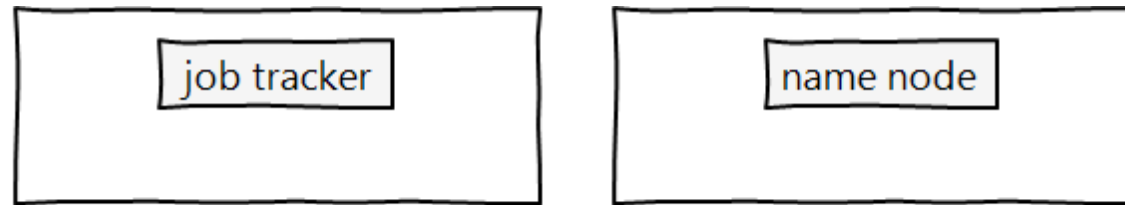
Без backup tasks, +44%



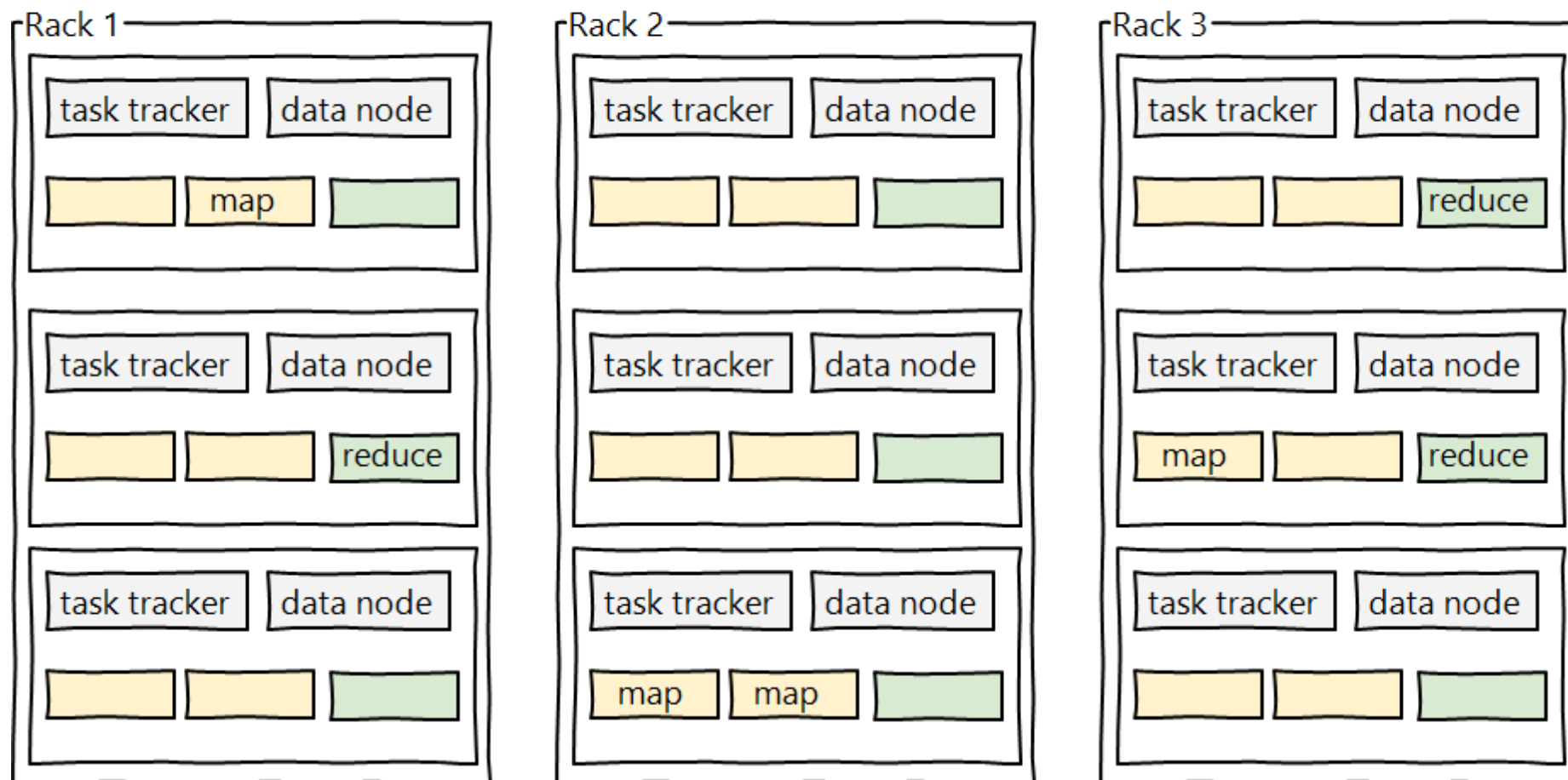
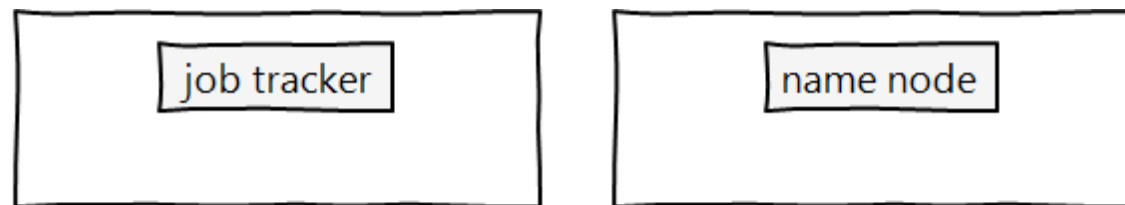
Падение 200 задач, +5%



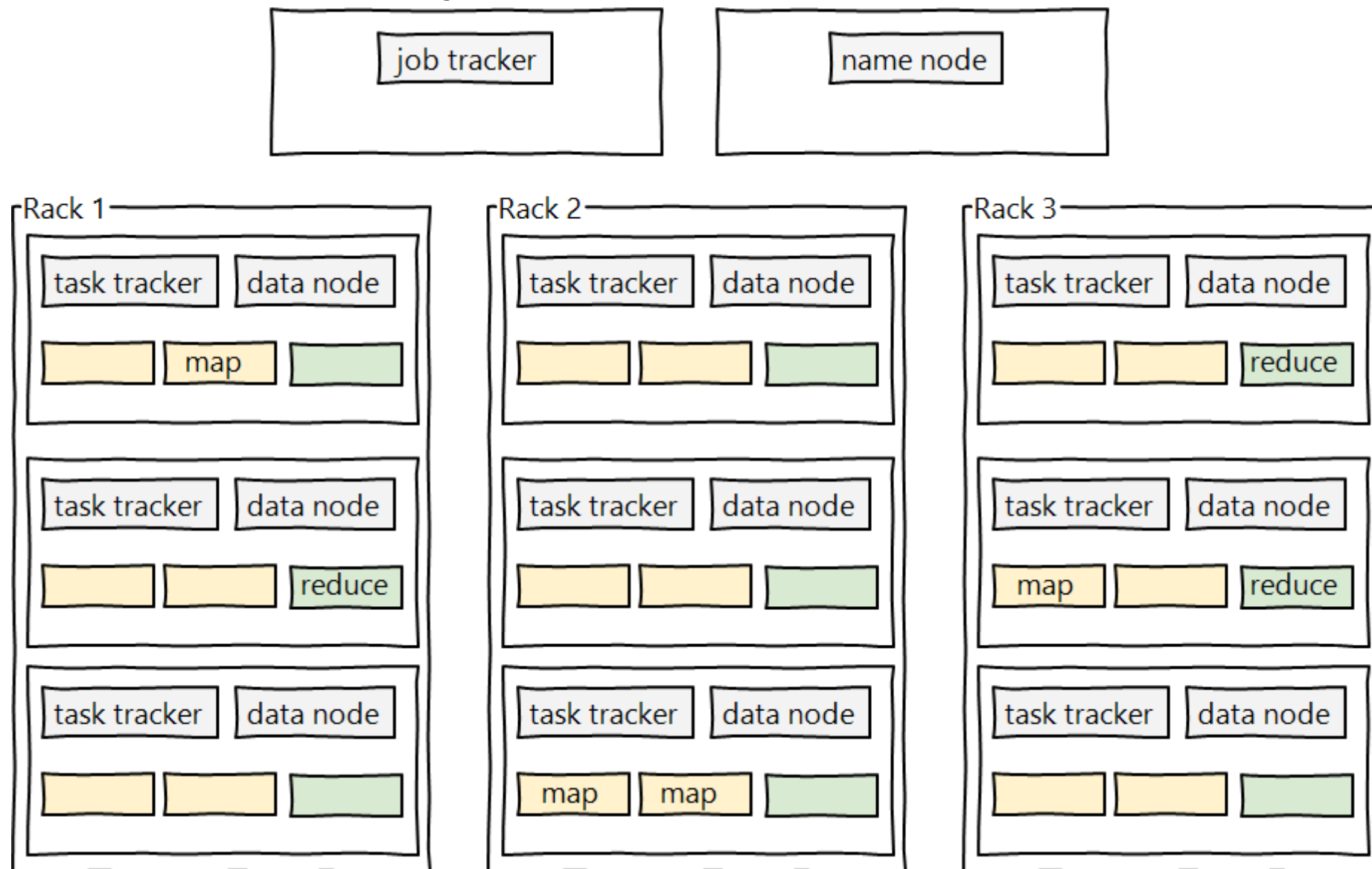
# Запуск на кластере



# Запуск на кластере

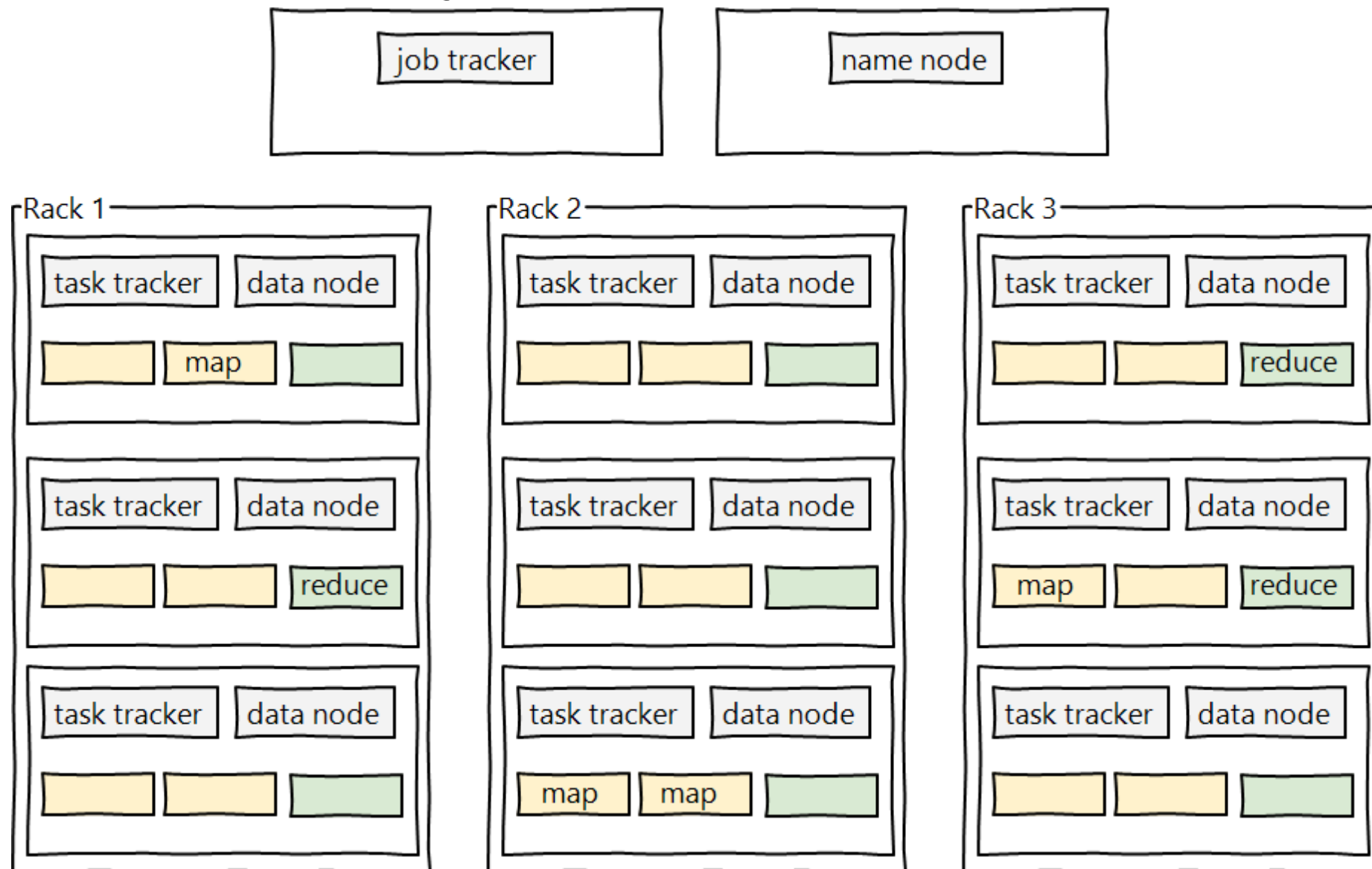


# Запуск на кластере



Что если задач больше чем слотов?

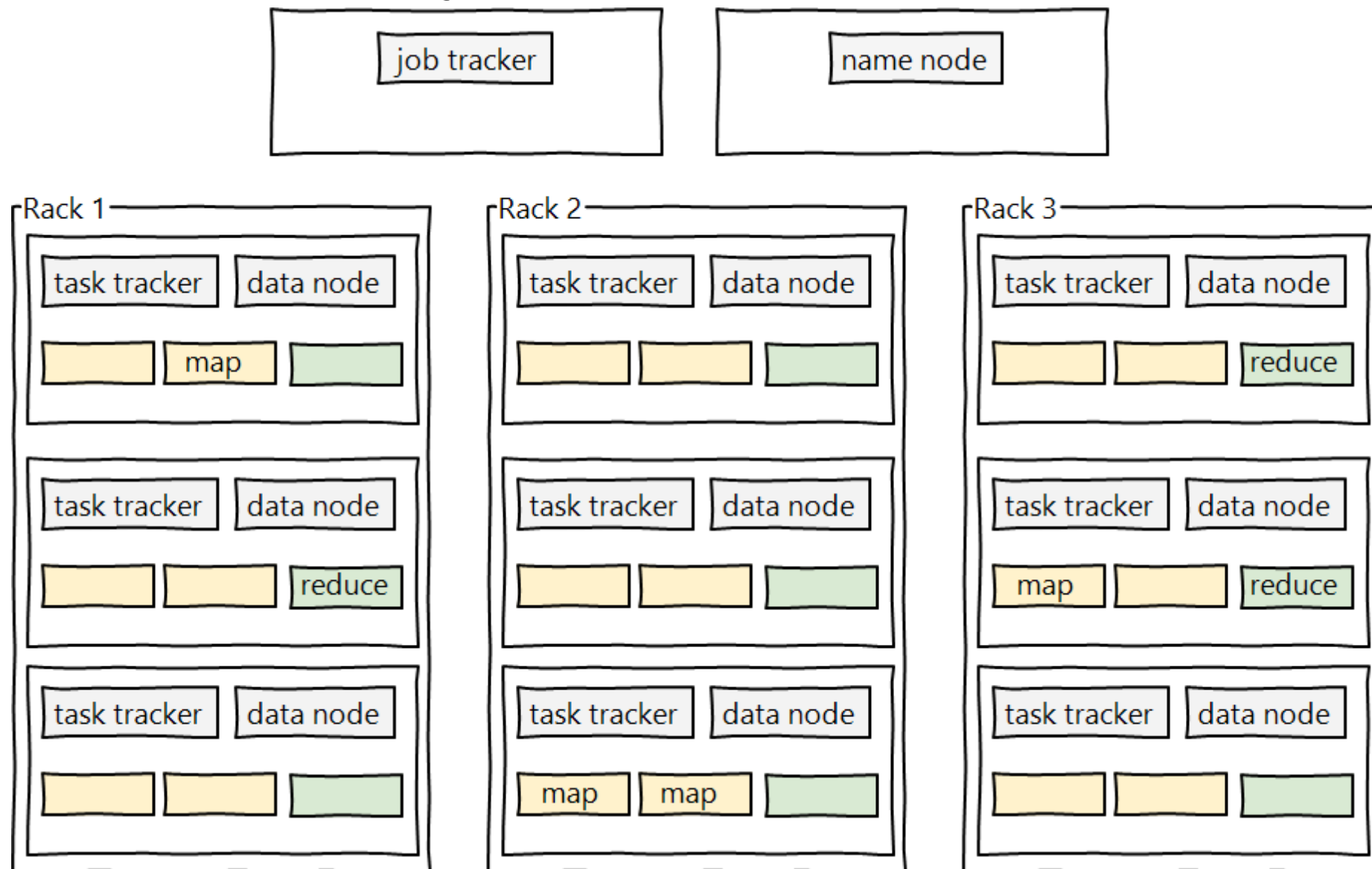
# Запуск на кластере



Что если задач больше чем слотов?

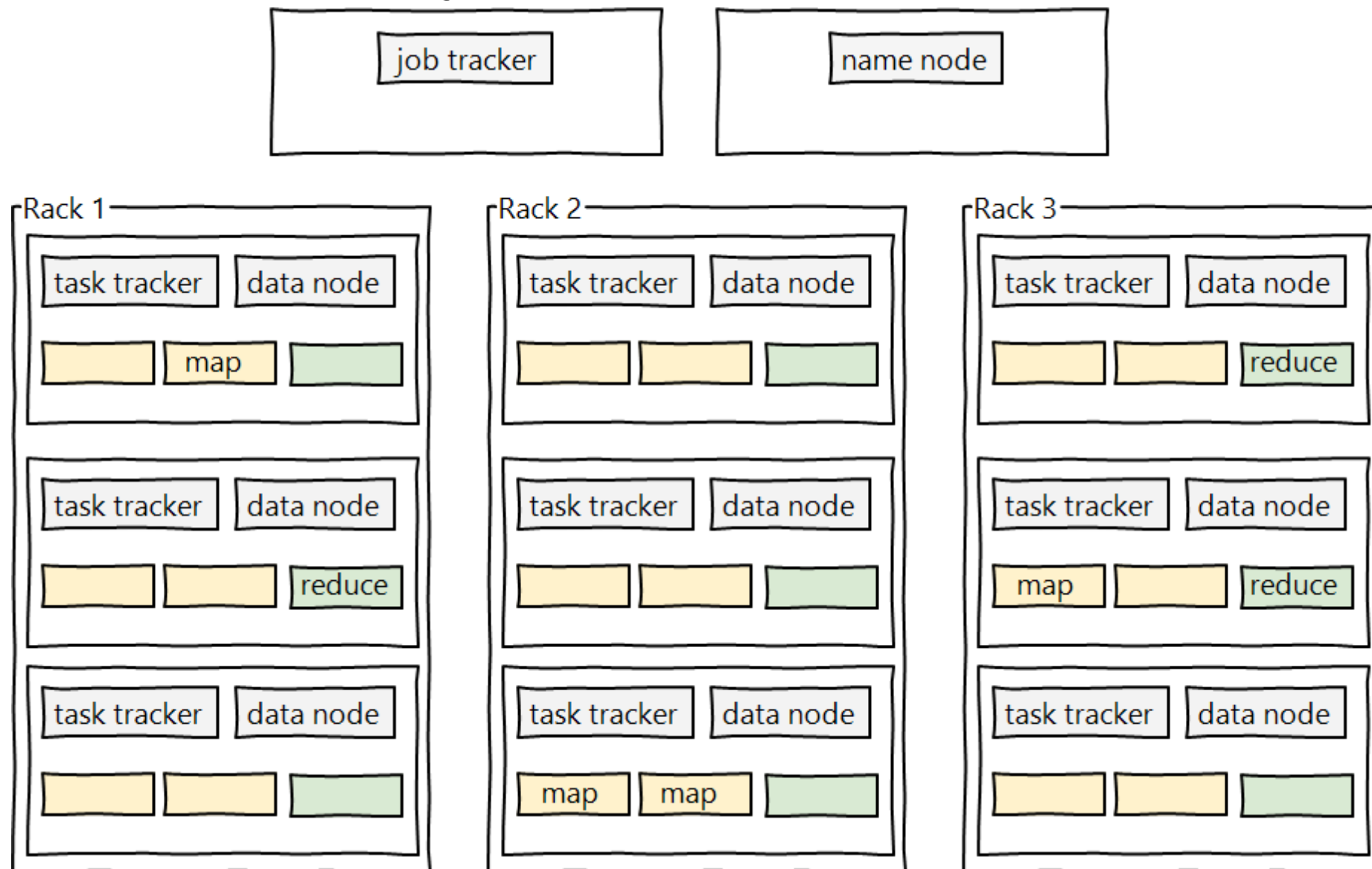
FIFO

# Запуск на кластере



Если запустить несколько job ?

# Запуск на кластере



Что если джобы не помещаются одновременно?

# Планирование заданий

- fifo scheduler
- fair scheduler

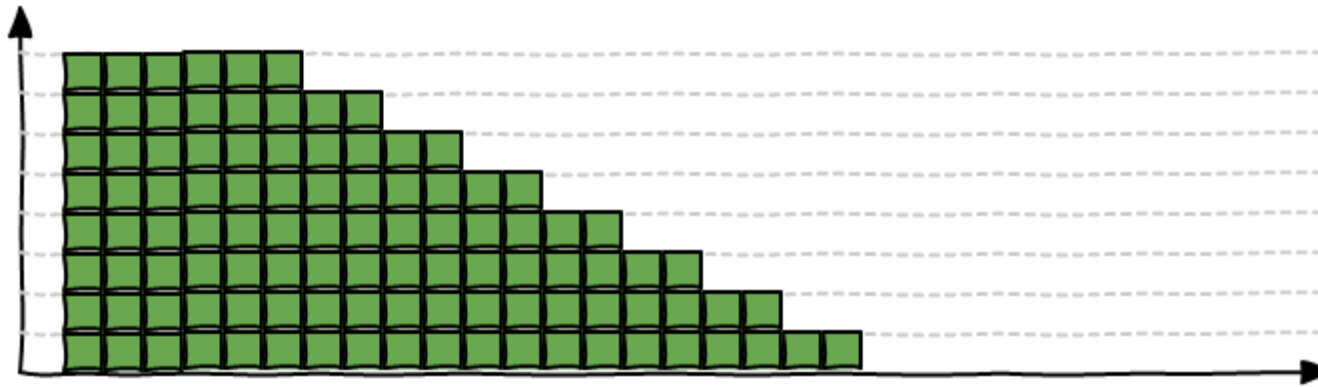
# Fifo scheduler

- Включен по умолчанию
- Все ресурсы достаются первому



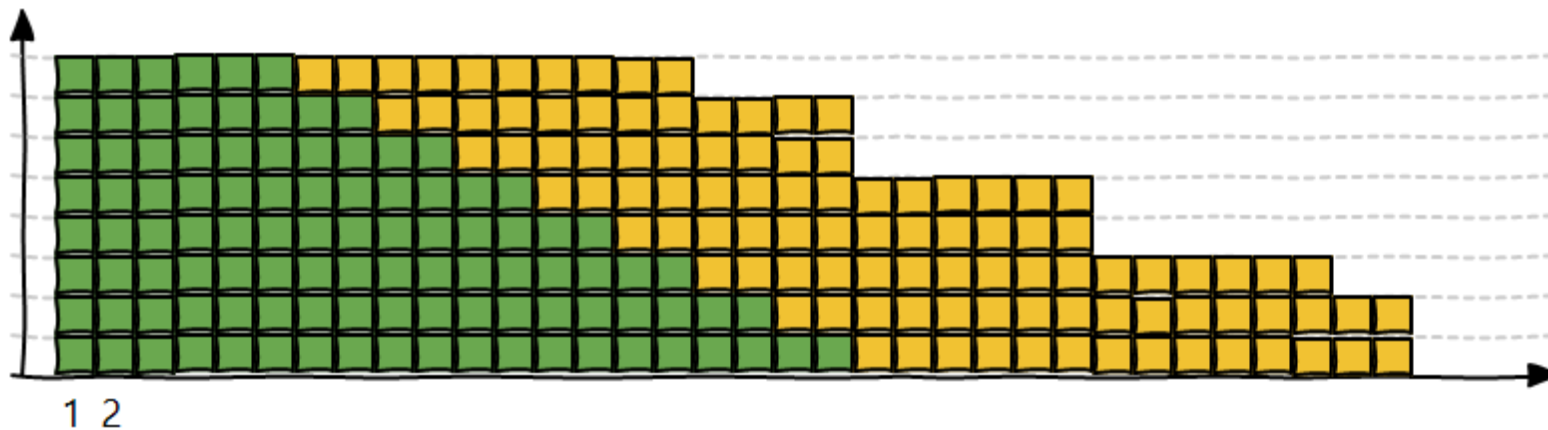
# Fifo scheduler

- Включен по умолчанию
- Все ресурсы достаются первому



# Fifo scheduler

- Включен по умолчанию
- Все ресурсы достаются первому



# Fair scheduler

- Основная идея: распределим ресурсы справедливо

# Fair scheduler

- Основная идея: распределим ресурсы справедливо
- Справедливо = поровну
- Но с учетом приоритетов!

# Fair scheduler

**Проблема:** Слотов у 1го job больше чем у 2го

# Fair scheduler

**Проблема:** Слотов у 1го job больше чем у 2го

**Решение:** после окончания задач 1го job передавать слоты 2му

# Fair scheduler

**Проблема:** 1й job занял кластер, второй не имеет ресурсов

# Fair scheduler

**Проблема:** 1й job занял кластер, второй не имеет ресурсов

**Решение:** preemption. Некоторое гарантированное количество задач 1го job убивается, передается 2му



# Fair scheduler

**Проблема:** у 1го job больше ресурсов, но задачи работают долго, нечего передавать

**Решение:** preemption после таймаута. Если сделать его сразу некоторое гарантированное количество задач 1го job убивается, передается 2му

# Fair scheduler

