Статистическая оптимизация MapReduce

Сергей Василинец
науч. рук. Д.В. Барашев

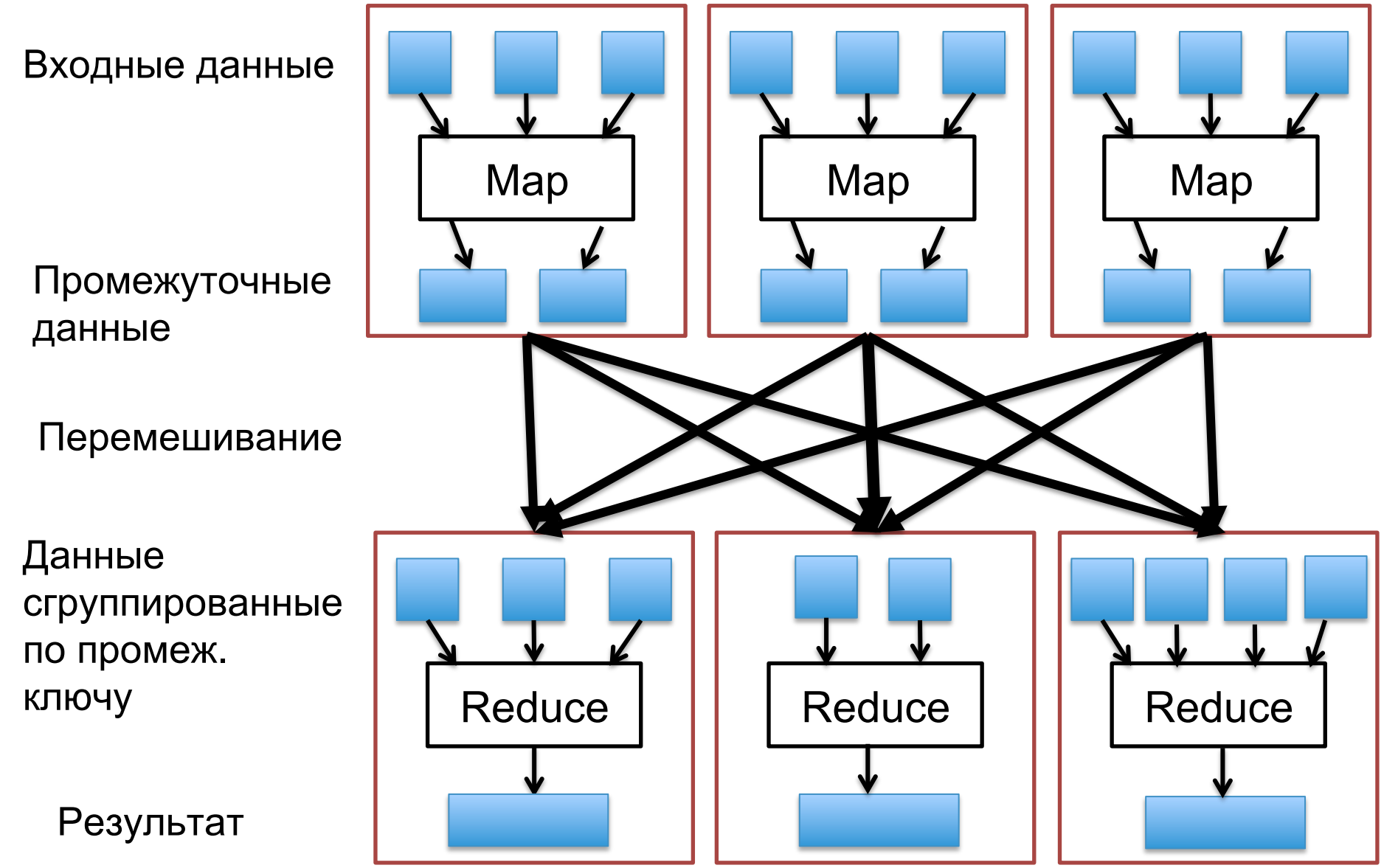
MapReduce

MapReduce **-** модель параллельной обработки большого количества данных на кластере.

Модель состоит из двух стадий:

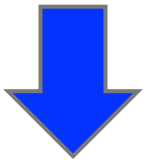
- Map
- Reduce

Модель MapReduce



Проблема

Функция распределения ключей по reducer: *key.hashCode() % n*



Одно или несколько reduce заданий выполняется значительно дольше остальных

Постановка задачи

- Реализовать сбор статистистических данных
- Реализовать Partitioner на основе собранной статистики
- Провести эксперименты

Алгоритмическая задача

Так разбить массив положительных чисел

$$a_1 \ldots a_p$$

на множества

$$M_1 \ldots M_k \text{ , что}$$

$$max\ S_i \ - min\ S_j \text{ минимальна,}$$

где $S_j = \Sigma a_i$, $a_i \in M_j$

Сбор статистики

- Сбор реализован на уровне Partitioner
- Реализована система, позволяющая собрать разные виды статистики

Эксперименты

Программы MapReduce:

- Word count
- First character
- Средняя длина сессии

Окружение экспериментов:
Кластер из 10 машин типа «small instances» на Amazon EC2.

Средняя длина сессии

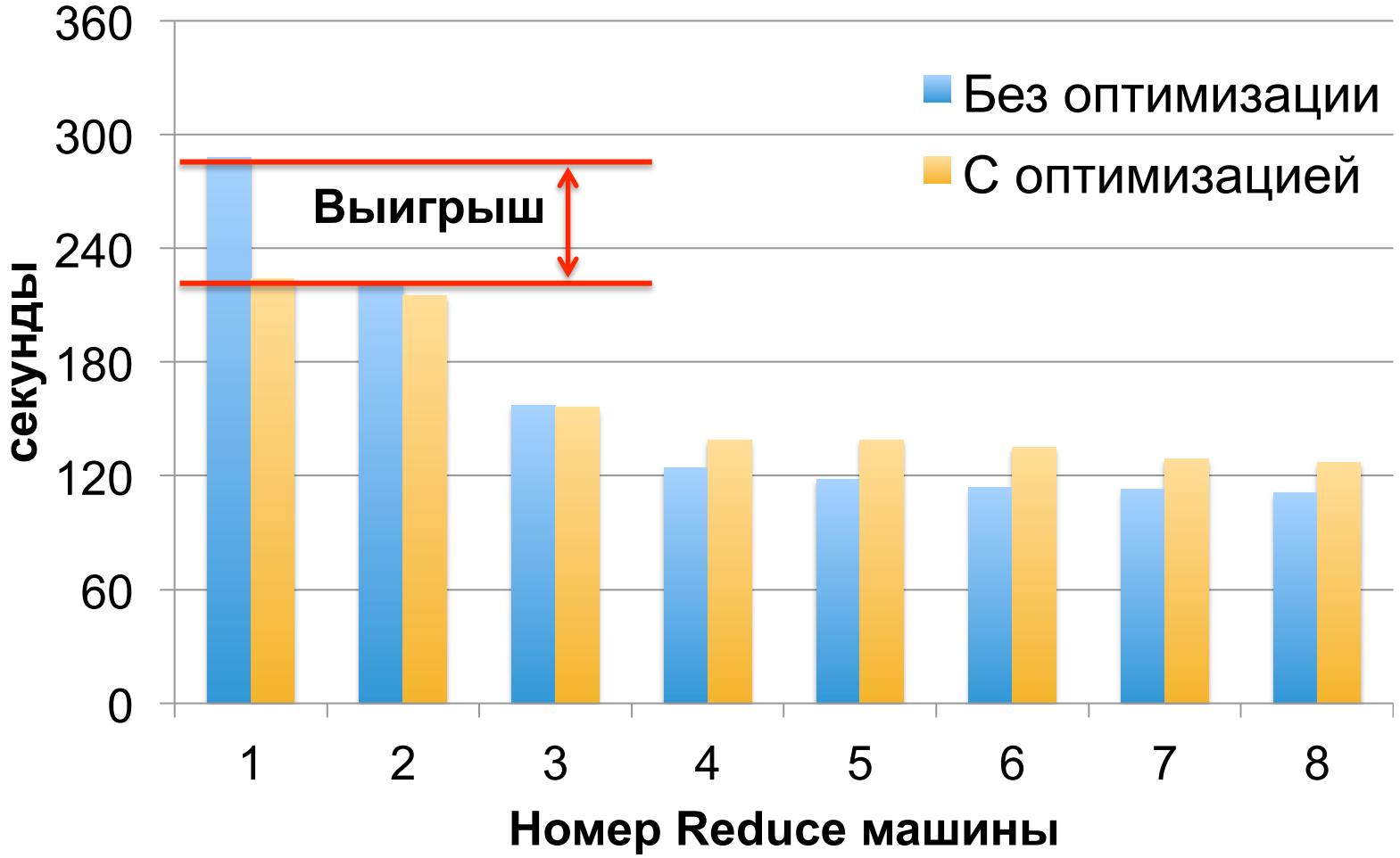
Цель:

Посчитать, сколько страниц за сессию в среднем просматривают люди из разных стран.

Интерес эксперимента:

Большое количество операций на стадии Reduce.

Время исполнения Reduce на 8 машинах



Результаты

- Реализован сбор статистики
- Реализован Partitioner на ее основе
- Проведены эксперименты

Заключение:

Оптимизация дает хорошие результаты при сложной стадии Reduce и промежуточных данных, распределенных неравномерно.