

Текстовая расшифровка видео:

MAPREDUCE ИЗНУТРИ И СНАРУЖИ

План:

- История и принципы;
- Пошаговый разбор MapReduce.

История и принципы

MapReduce как концепция была придумана в Google.

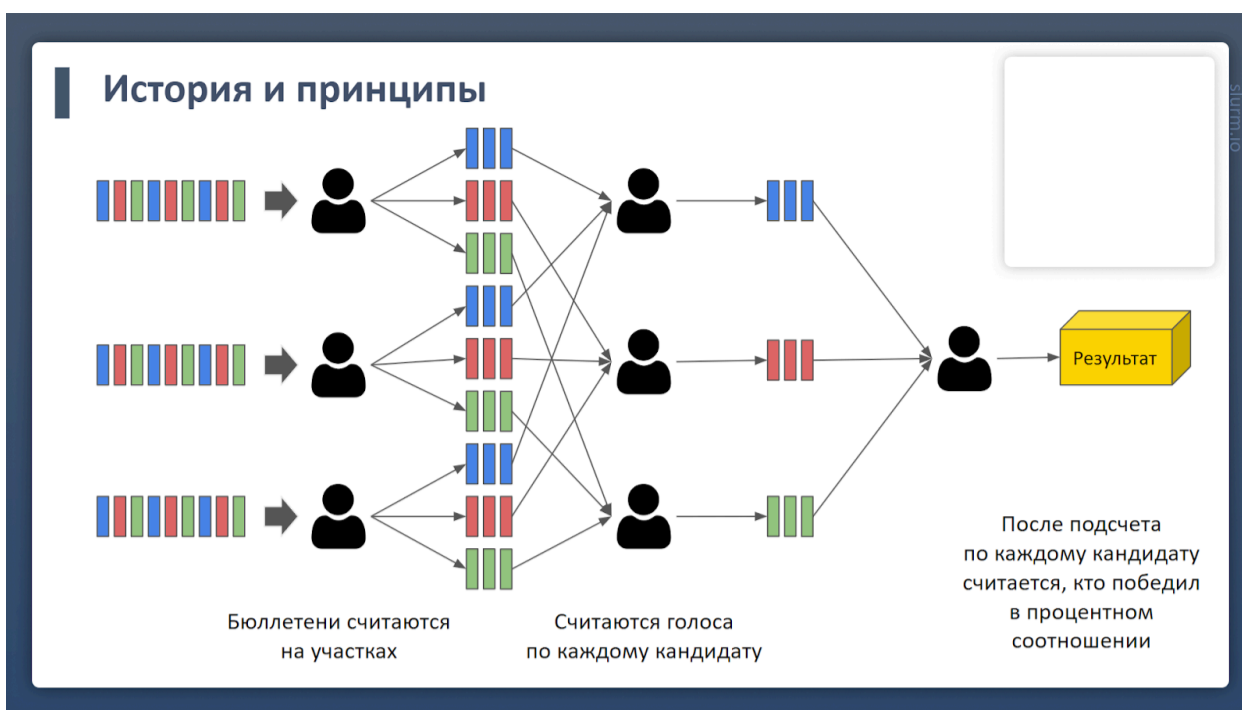
До 2004 года было много проблем с хранением и обработкой данных на многих машинах. Попытка распределенно посчитать и свести результат воедино занимала несколько дней, а то и недель. Каждое вычисление было уникальным, и для него требовалось каждый раз разрабатывать параллельный алгоритм.

С оригинальной статьей вы можете ознакомиться, перейдя по ссылке:

<https://static.googleusercontent.com/media/research.google.com/en/archive/mapreduce-osdi04.pdf>

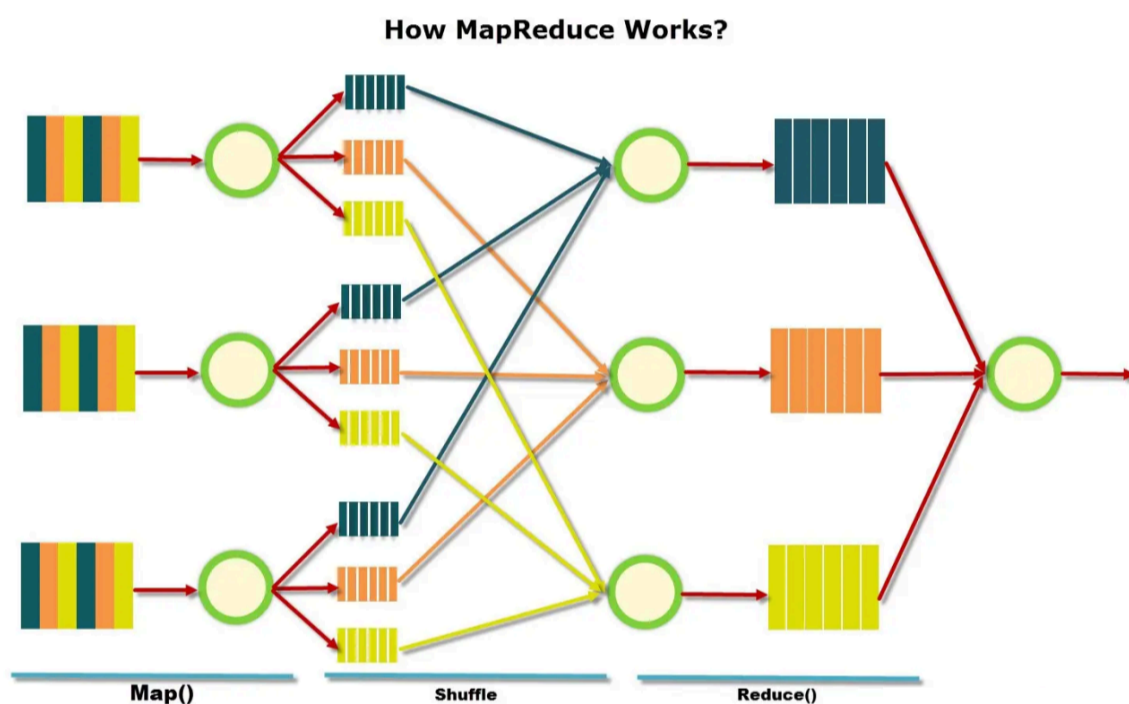
Приведем пример с аналогией выборов:





У нас есть много участков, на каждом из которых есть бюллетени и голосующие люди. По завершении голосования на каждом участке нужно посчитать количество голосов за кандидатов (процессинг, обработка). Все голоса сортируются по кандидатам. В глобальном масштабе суммируем общее число голосов за кандидатов. После подсчета по каждому кандидату считается, кто победил в процентном соотношении.

Если посмотреть на схему работы MapReduce, то можно заметить идентичность:



У нас есть данные, которые мы просчитываем и, группируя, пересылаем на другие машины. После этого по группам выводим конечный результат.

У MapReduce есть три основные стадии:

- **Map()** (в параллель процессим что-то на входе);
- **Shuffle** (стадия пересылки данных между машинами);
- **Reduce()** (делаем вывод на основе данных по ключам).

Пошаговый разбор MapReduce

Стадия «Map»:

- Читаем данные с HDFS, преобразуя их в пары «ключ-значение»;
- Применяем одну и ту же функцию к каждой записи;
- Рапортует в центр о завершении.

Стадия «Partition»:

- Решаем, на какой редюсер отправить данные;
- Все данные с одним ключом полетят на один редюсер;

- Число партиций = число редюсеров.

Стадия «Shuffle»:

- Перемещаем результаты с мапперов на соответствующие редюсеры.

Стадия «Sort»:

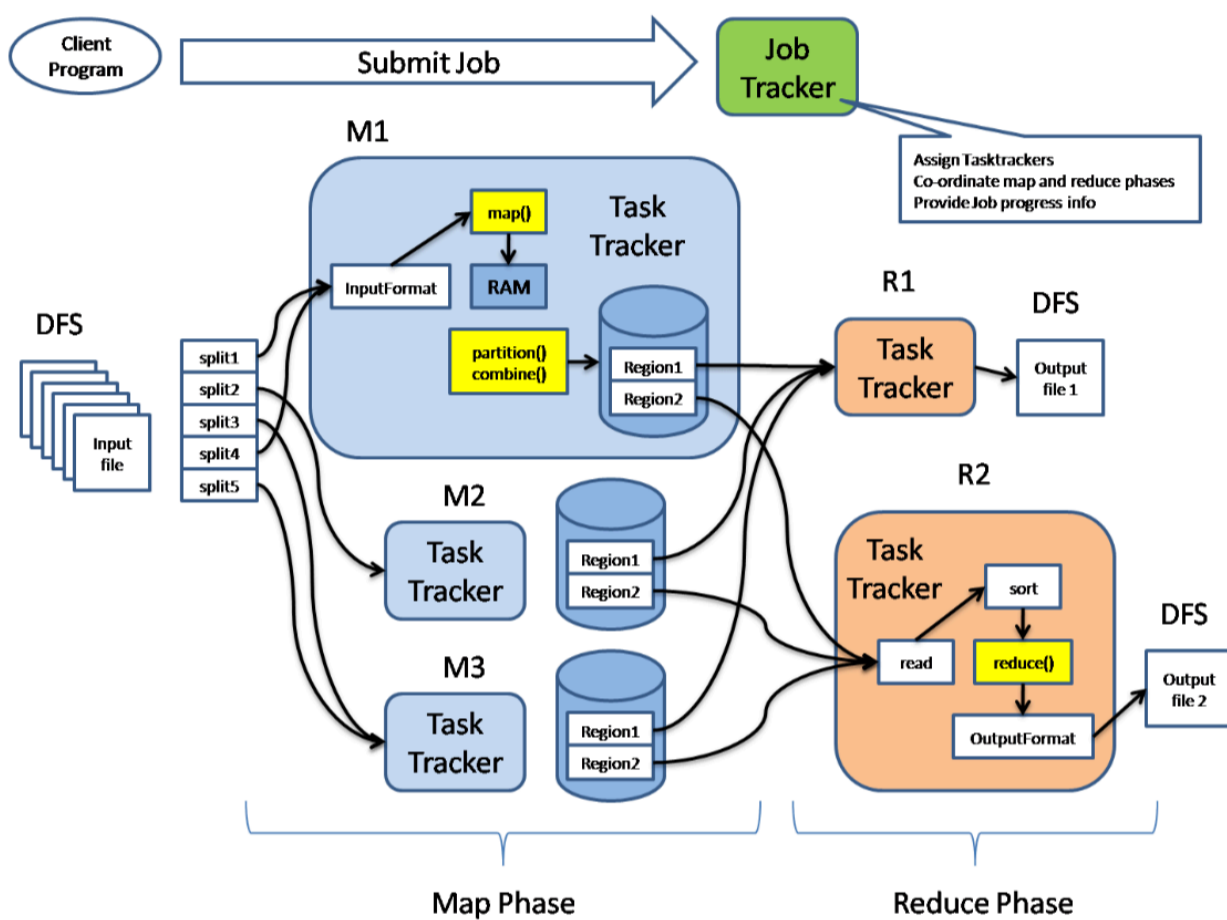
- Линейно объединяем выхлопы Map в отсортированную последовательность.

Стадия «Reduce»:

- Выполняем пользовательскую функцию Reduce на сортированных данных.

Далее, все, что остается сделать – применить конкретную операцию для набора значений.

Рассмотрим, как работает MapReduce с технической точки зрения:



На диаграмме можно увидеть наличие распределенной файловой системы. Видим файлы, которые бьются на кусочки (split'ы). Эти кусочки отправляются на мапперы, где все обрабатывается. В мапперах есть дополнительные промежуточные преобразования с partition и combine. За всем этим следят Task-трекеры. В итоге запускается фаза Reduce.

Чтобы сделать [YARN](#):

