

Введение в анализ данных: Поиск похожих объектов. MapReduce

Юля Киселёва
juliakiseleva@yandex-team.ru
Школа анализа данных



План на сегодня

- Поиск похожих объектов
 - Мотивация
 - Метрики расстояний
- MapReduce

Многомерные данные

Много реальных задач:

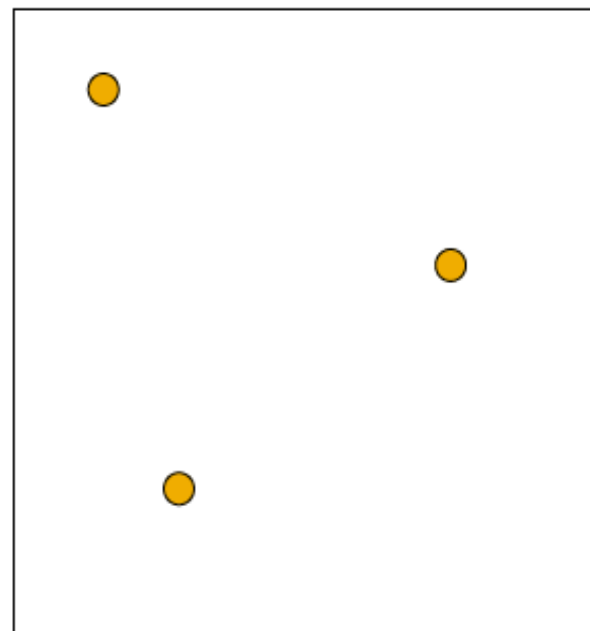
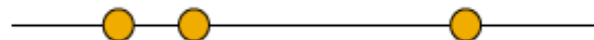
- Веб-поиск и анализа текста
 - Миллиард документов, миллион термов
- Системы рекомендаций для продуктов
 - Миллионы покупателей, миллионы продуктов
- Онлайн реклама, анализ поведения пользователя
 - Действия покупателей на сайте, поисковые запросы

Поиск похожих объектов

- Многие задачи, которые могут быть озвучены, как «найти похожие объекты»
- Примеры:
 - Веб-страницы с похожими словами (классификация, определение дубликатов)
 - Покупатели с «похожими интересами» (Netflix пользователи с похожими вкусами в кино)
 - Изображения с «похожими признаками»
 - Пользователи, которые посещают один и тот же веб-сайт

Многомерные данные

- Предположим, что есть набор данных с фиксированным числом N точек
- Если увеличить размерность пространства, в котором находятся данные точки, то средняя длина между точками в этом пространстве будет увеличиваться



Многомерные данные (2)

- Представление данных в векторном виде:
 - Все данные состоят из a, b, c, d
 - $ab = (1, 1, 0, 0)$
 - $bdc = (0, 1, 1, 1)$
- **Tf*Idf:**
 - $D = (d1, d2, \dots, di, \dots, dn)$
 - Tf (*term frequency*) = частота слова (терма) в данном документе
 - IDF (*inverse document frequency*) $IDF = \log \frac{|D|}{|(d_i \supset t_i)|}$
 - $di = (w1, w2, \dots, wn)$, где $wi = tf(t,d) * idf(t)$ для i -ого термина

Проблема разреженности данных

- Большинство покупателей не покупают большинство продуктов
- Большинство документов не содержат большинство слов
- **Простое решение:** Добавьте больше данных!
 - Больше покупателей, более длинная история покупок
 - Больше документов

Метрики расстояний

- Определим «ближайших соседей» как точки, которые имеют «маленькое расстояние» между друг другом
- Для каждого случая нужно определить, что означает «расстояние»
- Существует два основных класса расстояний:
 - Евклидово
 - Не-евклидово

Евклидово и Не-евклидово

- Евклидово пространство имеет определенную размерность
- *Евклидово расстояние* определяется на основе положения точек в пространстве
- *Не-евклидово расстояние* определяется на основе свойств точек, но не на основе их положения в пространстве

Аксиомы метрик расстояний

- d – является метрикой расстояния, если это функция, которая отображает пару точек в вещественное число и удовлетворяет следующим аксиомам:

1. Не отрицательность $d(p,q) \geq 0$
2. Тождественность $d(p,q) = 0$ iff $p = q$
3. Симметрия $d(p,q) = d(q,p)$
4. Неравенство треугольника $d(p,q) \leq d(p,r) + d(r,q)$

Некоторые евклидовы метрики расстояния

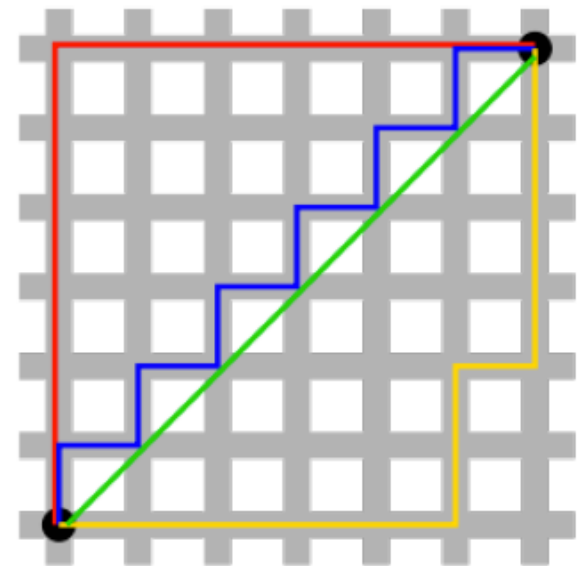
L_2 norm: $d(p, q)$

$$d(p, q) = d(q, p) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}.$$

L_1 norm: Сумма абсолютных разниц по каждому измерению

Манхеттановское расстояние (Manhattan distance) = расстояние, при котором вы можете путешествовать только по осям

$$d_1(p, q) = \|p - q\|_1 = \sum_{i=1}^n |p_i - q_i|,$$



Другие евклидовы расстояния

L_∞ norm: $d(x, y)$ = максимум разниц между x и y
по каждой оси

$$D_{\text{Chebyshev}}(p, q) := \max_i (|p_i - q_i|).$$

L_p norm:

$$d([x_1, x_2, \dots, x_n], [y_1, y_2, \dots, y_n]) = \left(\sum_{i=1}^n |x_i - y_i|^r \right)^{1/r}$$

Не-евклидовы метрики расстояний

1. Косинусное расстояние (Cosine Distance) = это угол между векторами
2. Edit distance = число вставок, удалений, которое нужно чтобы преобразовать одну строку в другую
3. Расстояние Хемминга (Hamming Distance) = число позиций, в которых соответствующие символы двух слов одинаковой длины различны

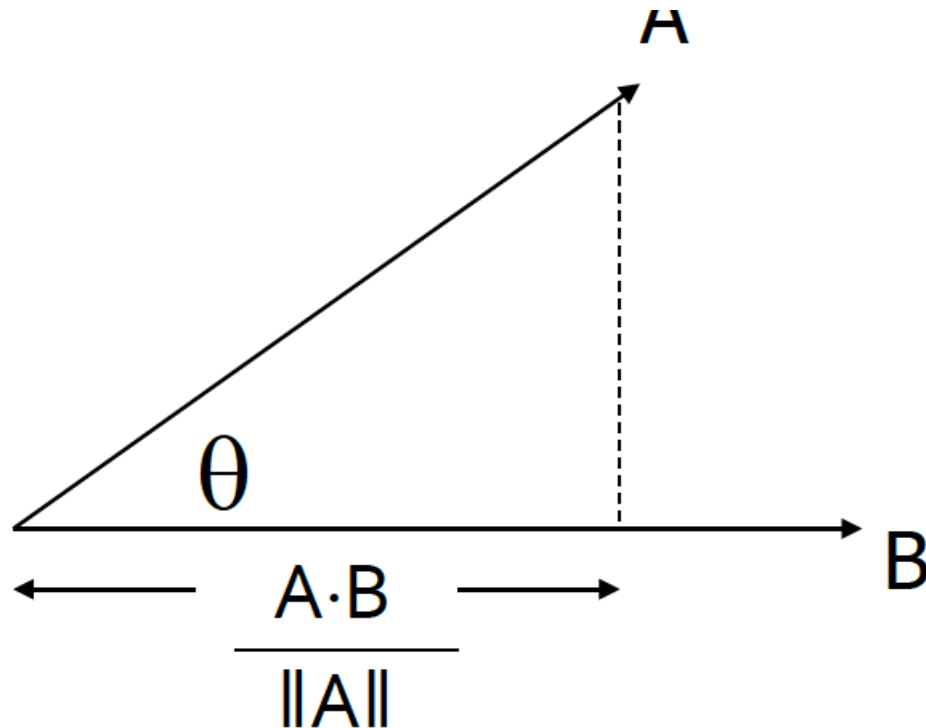
Косинусное расстояние

- Объект представлен в виде вектора
- Между двумя векторами образуется угол и значение косинуса этого угла – это нормализованное скалярное произведение:

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

- **Пример:** A = 00111; B = 10011
 - $A \cdot B = 2$; $\|A\| = \|B\| = \sqrt{3}$
 - $\cos(\theta) = 2/3$;

Косинусное расстояние: диаграмма



$$d(A, B) = \theta = \arccos\left(\frac{A \cdot B}{\|A\| \|B\|}\right)$$

Косинусное расстояние – это метрика

- $d(x, x) = 0$ потому что $\arccos(1)=0$
- $d(x,y) = d(y,x)$ по симметрии : очевидно, что угол между x и y равен углу между y и x
- $d(x,y) \geq 0$ потому что мы рассматриваем углы от 0 до 180 градусов
- **неравенство треугольника** (физический смысл)

Edit Distance (редакционное расстояние или дистанция редактирования)

- Edit distance между двумя строками – это число вставок и удалений символов, которые необходимо сделать, чтобы преобразовать одну строку в другую

Эквивалентно:

$$d(x,y) = |x| + |y| - 2|LCS(x,y)|$$

- **LCS** (longest common subsequence) = наибольшая общая подпоследовательность

Edit Distance: пример

- $x = abcde$; $y = bcduve$
- Преобразование x в y :
 - Удалить a
 - Вставить u и v после d
 - Edit distance = 3
- Или : $LCS(x,y) = bcde$

$$d(x,y) = |x| + |y| - 2|LCS(x,y)|$$
$$= 5 + 6 - 2 * 4 = 3$$

Edit Distance – это метрика

- $d(x, x) = 0$
- $d(x, y) = d(y, x)$: потому что вставка/удаление являются обратными по отношению друг к другу
- $d(x, y) \geq 0$: «отрицательное редактирование» не имеет смысла
- **неравенство треугольника**: преобразование x к z и затем к y – это один из путей преобразовать x к y

Расстояние Хемминга

- **Расстояние Хемминга (Hamming Distance)** – число позиций, в которых соответствующие символы двух слов одинаковой длины различны
- **Пример:** $x = 10101$; $y = 10011$
- **Ответ:**
 $d(x, y) = 2$

Jaccard Similarity

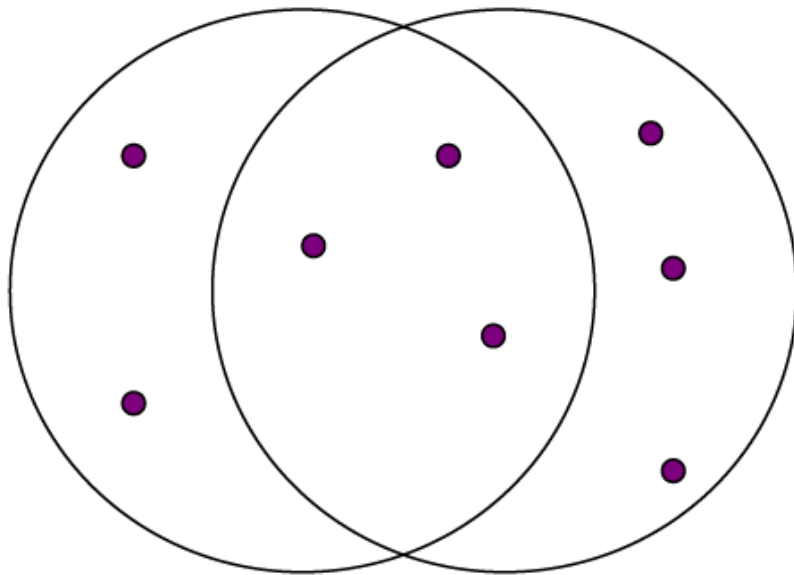
- **Jaccard Similarity** между двумя наборами – это размер их пересечения, разделенного на размер их объединения

- $Sim(C_1, C_2) = |C_1 \cap C_2| / |C_1 \cup C_2|$

- **Jaccard Distance** между двумя наборами – это 1 минус Jaccard Similarity

- $d(C_1, C_2) = 1 - |C_1 \cap C_2| / |C_1 \cup C_2|$

Пример: Jaccard Distance



Пересечение = 3

Объединение = 8

Jaccard Similarity = $3/8$

Jaccard Distance = $5/8$

План на сегодня

- Поиск похожих объектов
 - Мотивация
 - Метрики расстояний
- MapReduce

Мотивация

- 20+billon веб-страниц * 20 KB = 400+ TB
- 1 компьютер читает со скоростью 35 MB/sec с диска
~ 4 месяца, чтобы скачать веб
- **И еще нужно анализировать данные**

Пример: большие вычисления – большие машины

- **Traditional big-iron box (2003)**
 - 8 2GHz Xeons
 - 64GB RAM
 - 8TB disk
 - 758,000 USD
- **Prototypical Google rack (2003)**
 - 176 2GHz Xeons
 - 176GB RAM
 - ~7TB disk
 - 278,000 USD
- In Aug 2006 Google had ~450,000 machines

Масштабные вычисления

- Большие вычисления для АД на кластерах:
 - ПК соединены сетью
 - Процессинг большого объема данных на большом количестве машин
- Проблемы:
 - как организовать распределенные вычисления
 - Распределенные/параллельные вычисления – это сложно
 - Машины могут сломаться (из 1000 серверов в день ломается 1)
- MapReduce помогает бороться со всеми описанными проблемами

Идея и решение

– Идея:

- Перенести вычисления ближе данных ближе к данным
- Сохранить файлы несколько раз для надежности

– Что нужно:

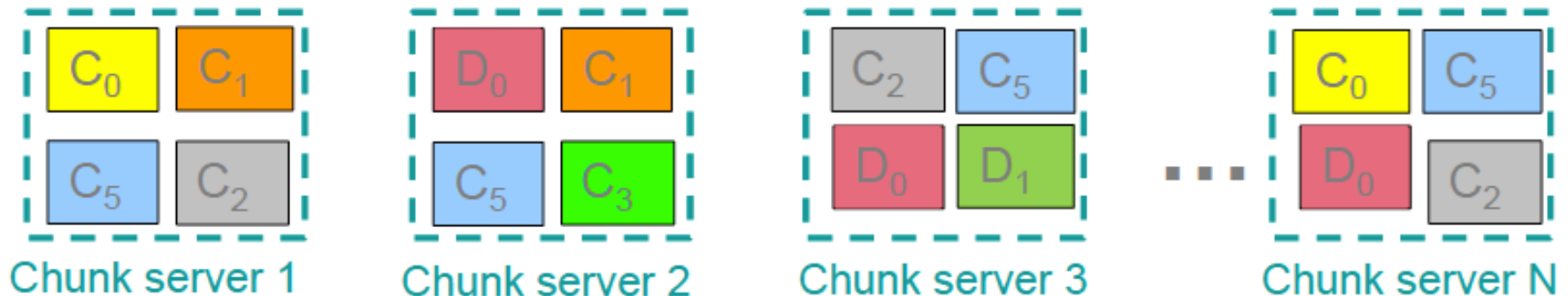
- Программная модель
 - Map-Reduce
- Инфраструктура – **Распределенная Файловая система**
 - Google: GFS
 - Hadoop: HDFS
 - Kosmix: KFS

– Типичный шаблон использования:

- Большие файлы (100s GB)
- Данные обновляются редко
- Чтение и добавление – это обычная операция

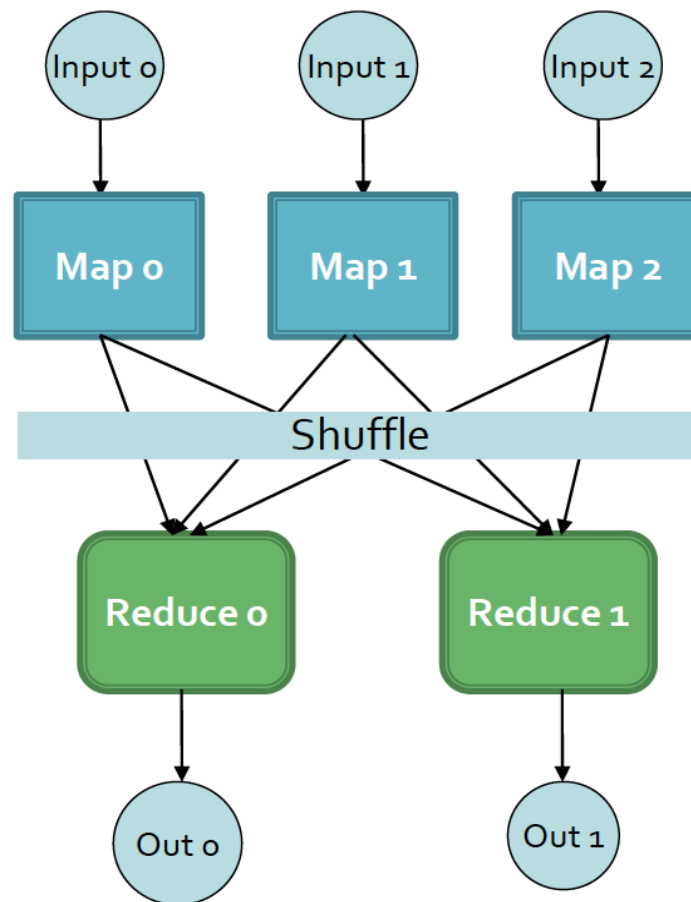
Распределенная файловая система

- **Chunk Servers:**
 - Файл разделен на «соседствующие» chunks
 - Типичный размер chunk – это 16-64MB
 - Каждый chunk реплицируется на разные машины (обычно 2 или 3 раза)
- **Master node:**
 - Хранение мета-данных

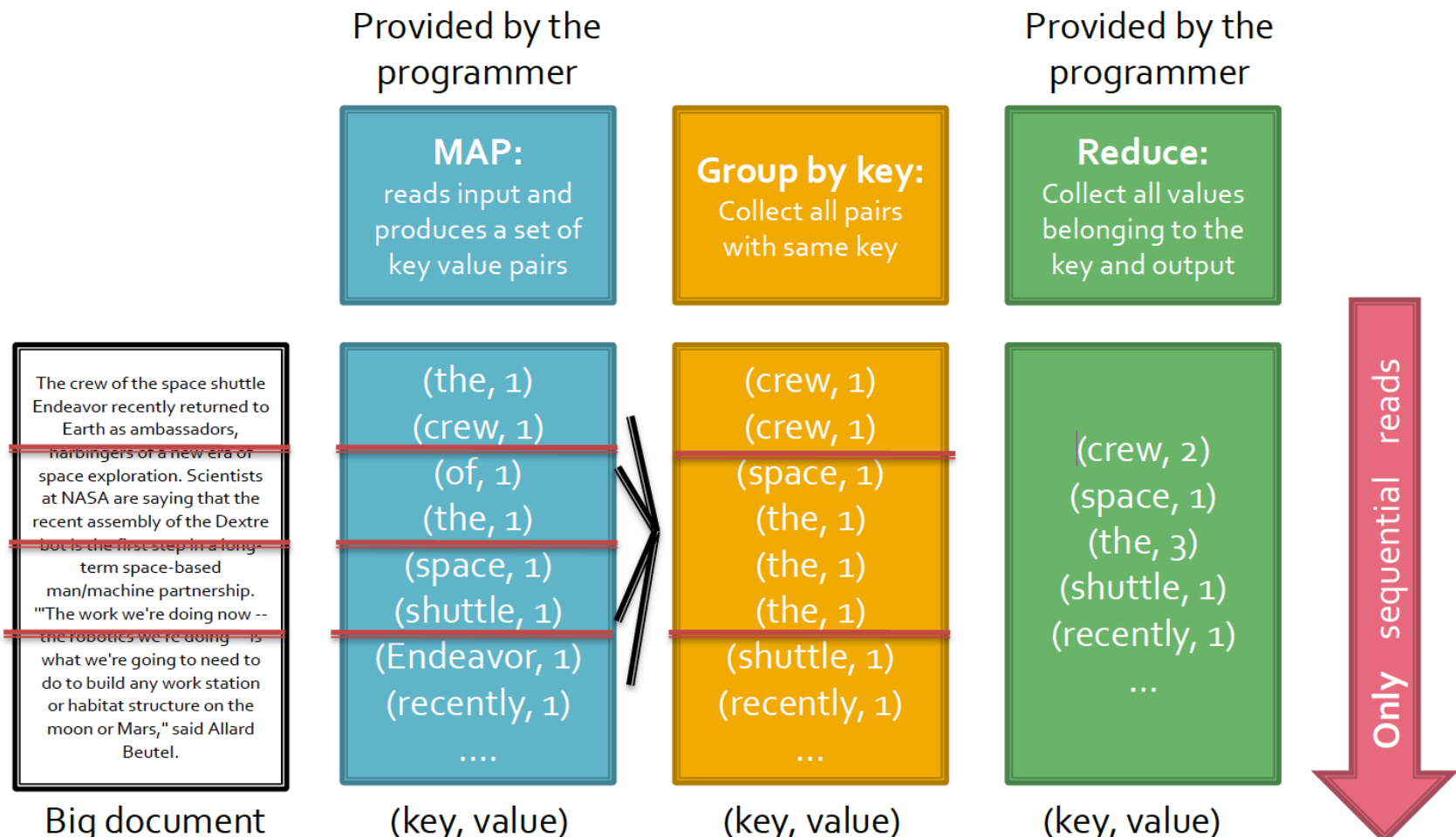


MapReduce: представление

- Читает данные
- **Функция – Map $\langle k, v \rangle$:**
 - Извлекает то, что Вам нужно
- Смешивает и сортирует
- **Функция – Reduce $\langle k, v \rangle \rightarrow \langle k, v' \rangle$:**
 - Агрегирует, суммирует, фильтрует и трансформирует
- Пишет результат



Пример: подсчет статистики по словам



Map Reduce: Enviroment

- MapReduce enviroment заботится о:
 - Partitioning входных данных
 - Расписании выполнения программ на наборе машин
 - Манипулирование «сломанными» машинами
 - Управление внутренней коммуникацией между машинами

Имплементация

- Google
 - Недоступна вне Google
- Hadoop
 - Открытая имплементация на Java
 - Используется HDFS for stable storage
 - Скачать: <http://lucene.apache.org/hadoop/>
- Aster Data
 - Cluster-optimized SQL Database которая также реализует MapReduce
- MongoDB

Резюме

- Познакомились с техникой Map Reduce
- Сейчас продолжим 😊
- Изучили или повторили метрики для нахождения похожих объектов
- Какие узнали метрики?