Формат данных ORC (Optimized Row Columnar)

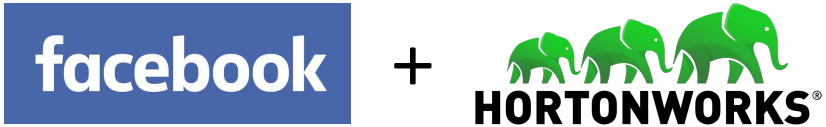
Драль Алексей, study@bigdatateam.org
CEO at BigData Team, https://bigdatateam.org
https://www.facebook.com/bigdatateam

Q&A

Как расшифровывается ORC?

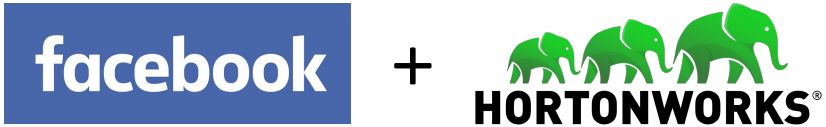
ORC и Parquet

- ORC = Optimized Row Columnar (File Format)



ORC и Parquet

- ORC = Optimized Row Columnar (File Format)



- Parquet - основан на статье Google Dremel
  (для вложенных структур данных, nested structures)

ORC

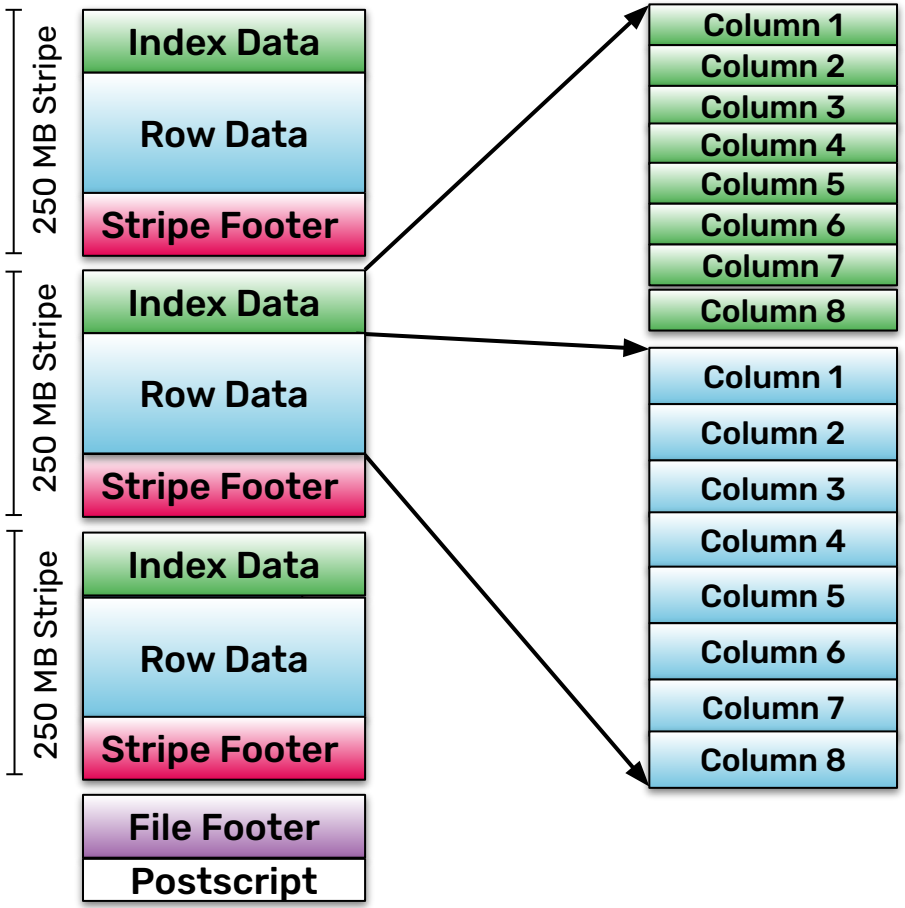
± 1. Быстрая загрузка данных в хранилище
✓ 2. Высокая скорость обработки запросов
✓ 3. Эффективное использование жесткого диска
✗ 4. Адаптивность к динамическому изменению паттернов аналитических запросов

↓

✓ ORC = Optimized Row Columnar (File Format)

BIGDATA TEAM

ORC



https://cwiki.apache.org/confluence/display/Hive/LanguageManual+ORC

BIGDATA TEAM

Статистики в ORC

```
message IntegerStatistics {
    optional sint64 minimum = 1;
    optional sint64 maximum = 2;
    optional sint64 sum = 3;
}
```

ORC v.1 спецификация: https://orc.apache.org/specification/ORCv1/

BIGDATA TEAM

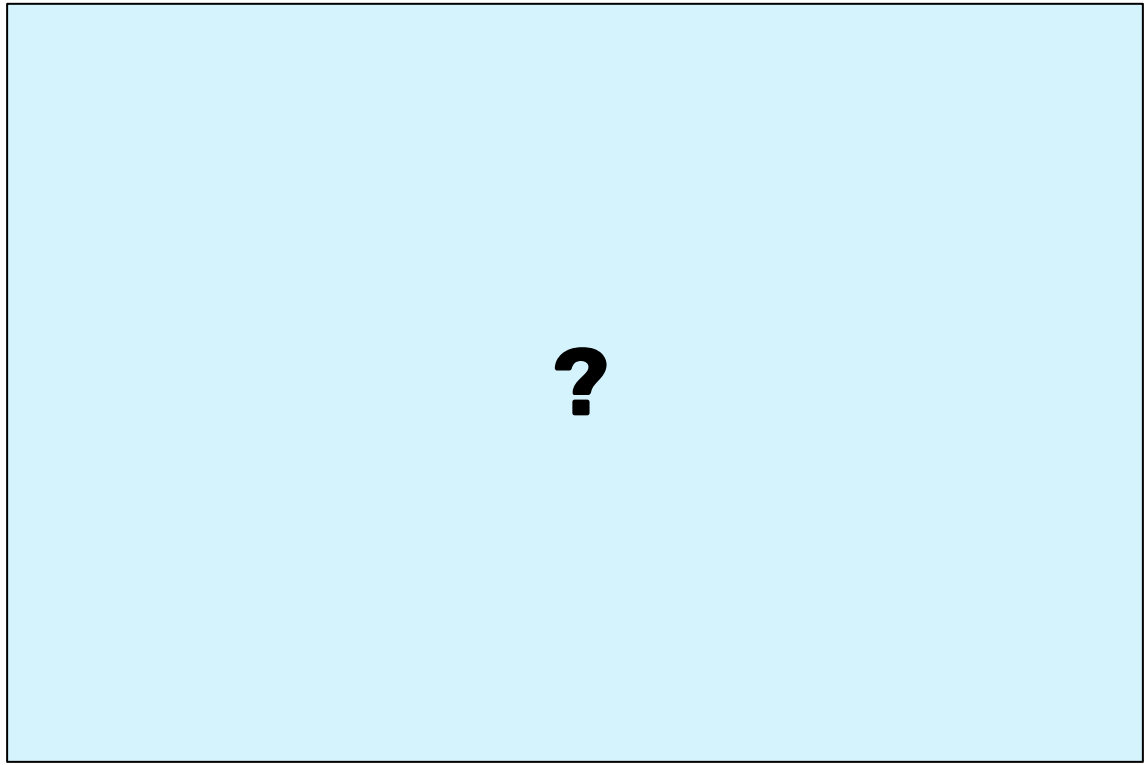
Статистики в ORC

```
message ColumnStatistics {
 optional uint64 numberOfValues = 1;
 optional IntegerStatistics intStatistics = 2;
 optional DoubleStatistics doubleStatistics = 3;
 ...
 optional bool hasNull = 10;
}
```

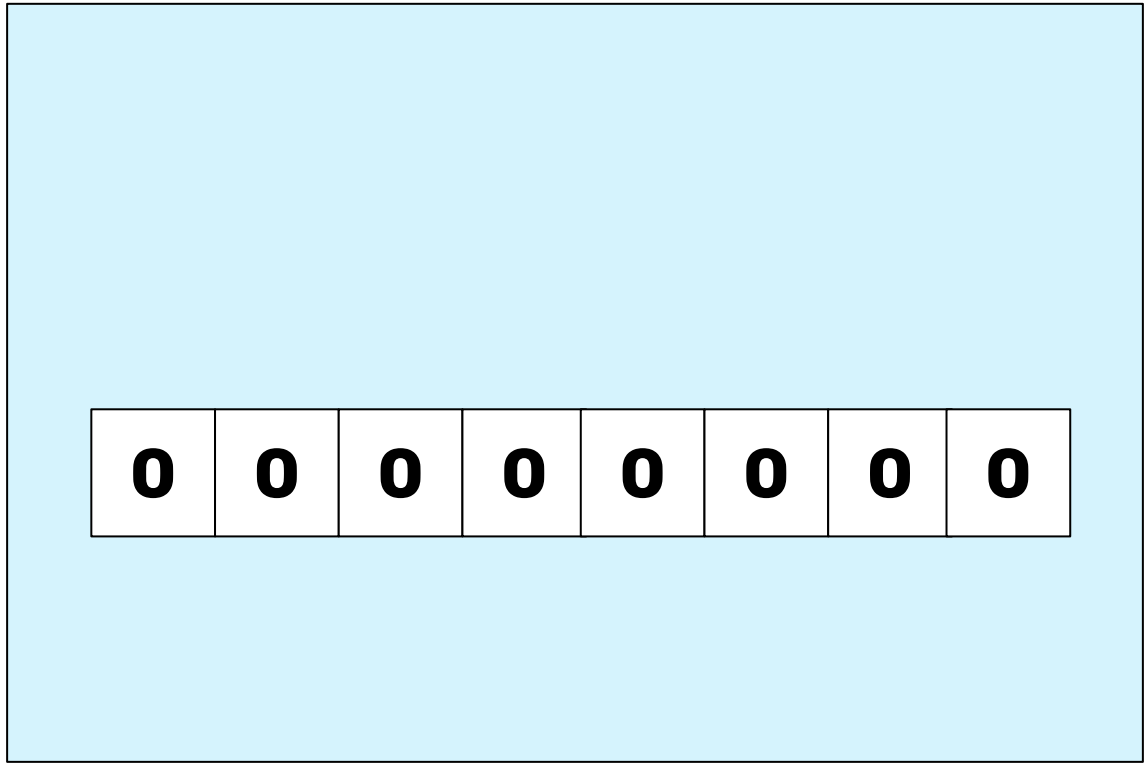
ORC v.1 спецификация: https://orc.apache.org/specification/ORCv1/

BIGDATA TEAM

Фильтр Блума (Bloom Filter)

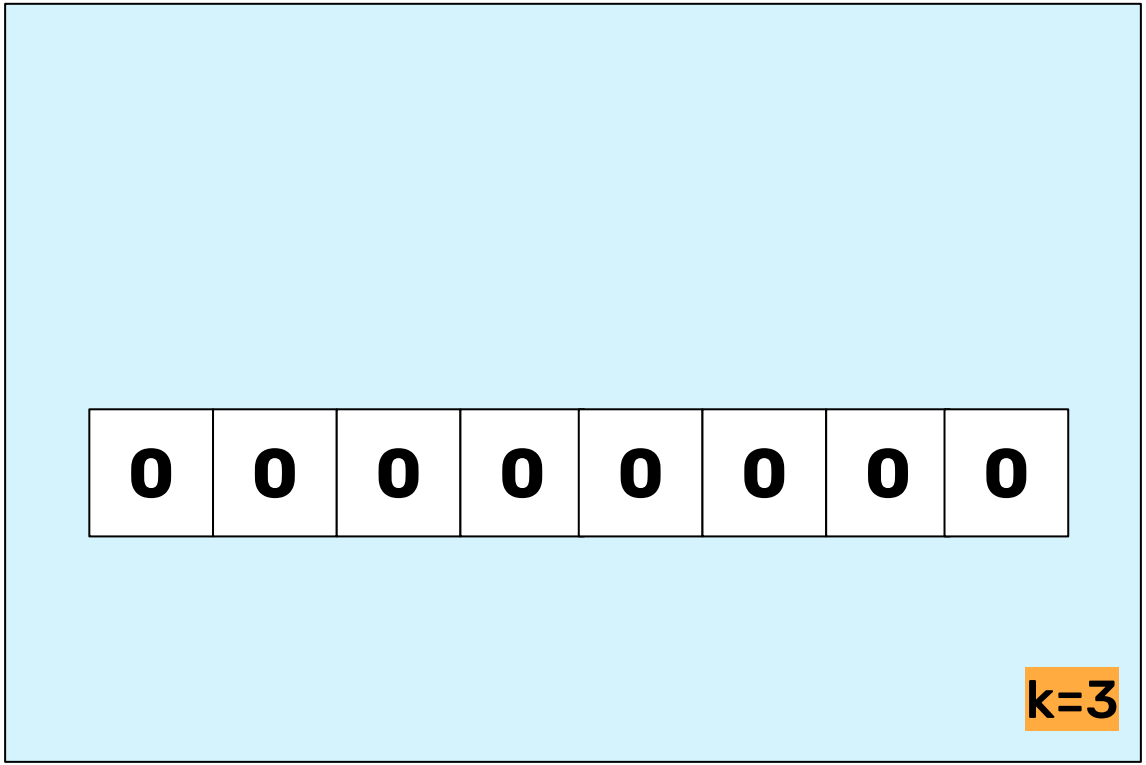


Фильтр Блума (Bloom Filter)



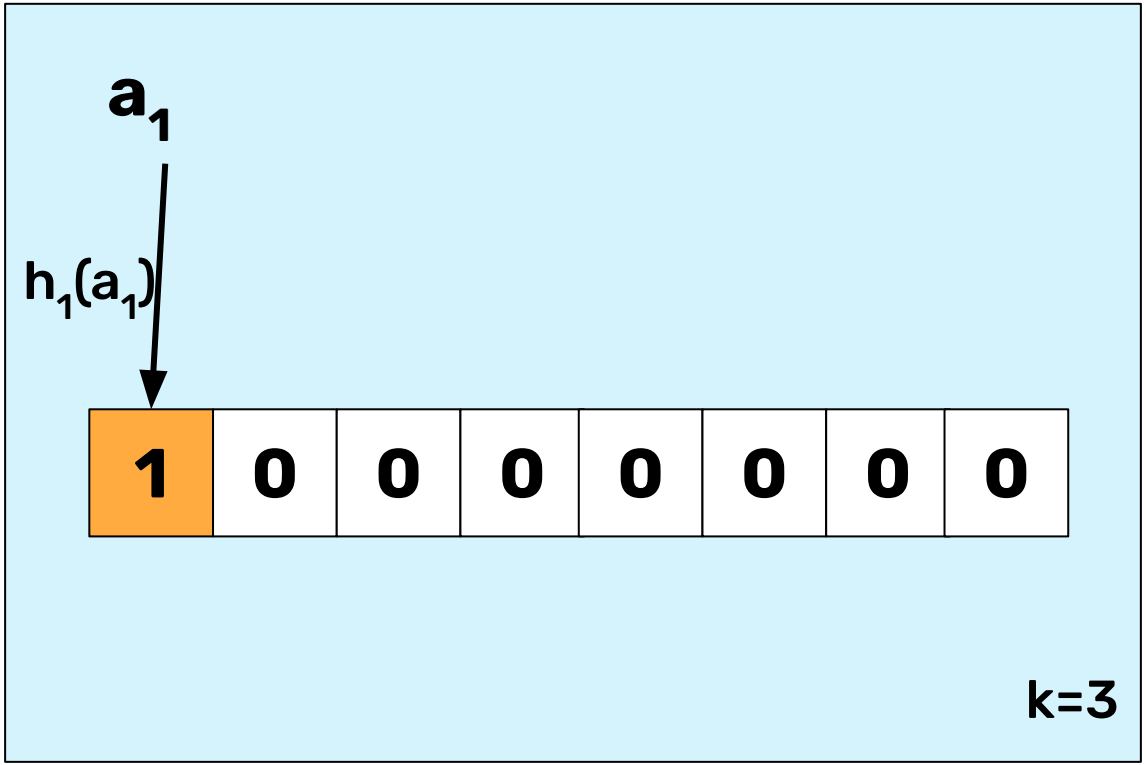
BIGDATA TEAM

Фильтр Блума (Bloom Filter)



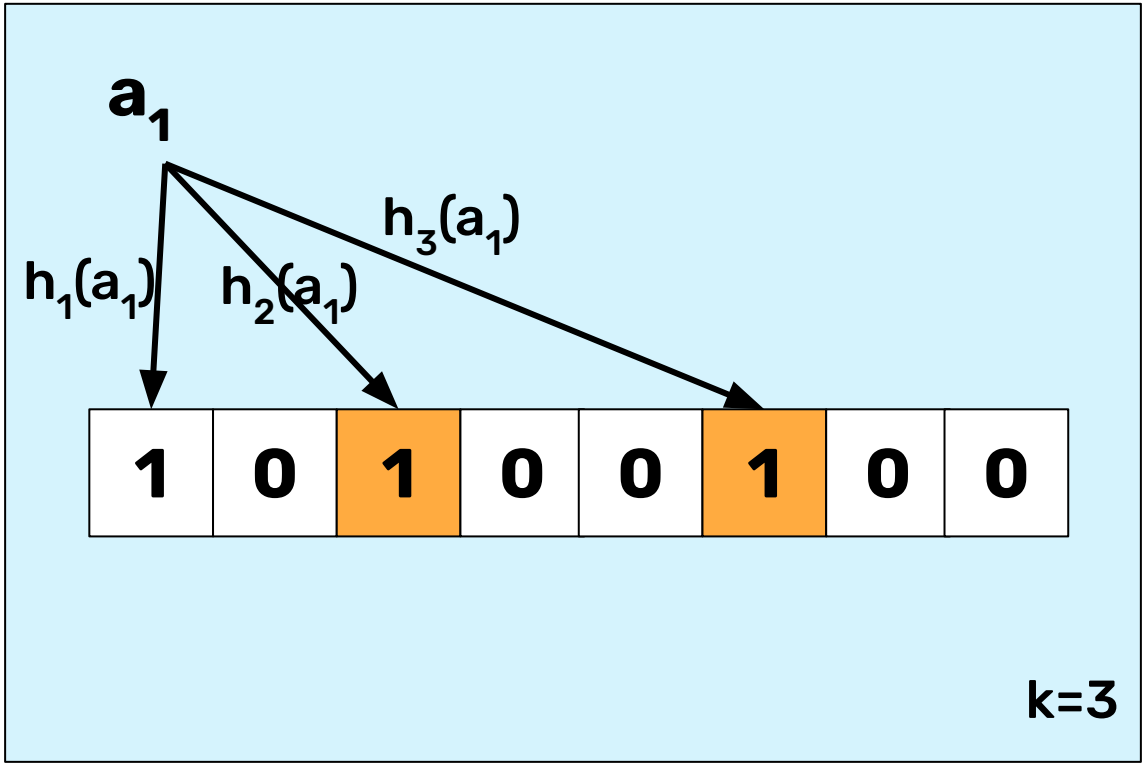
BIGDATA TEAM

Фильтр Блума (Bloom Filter)



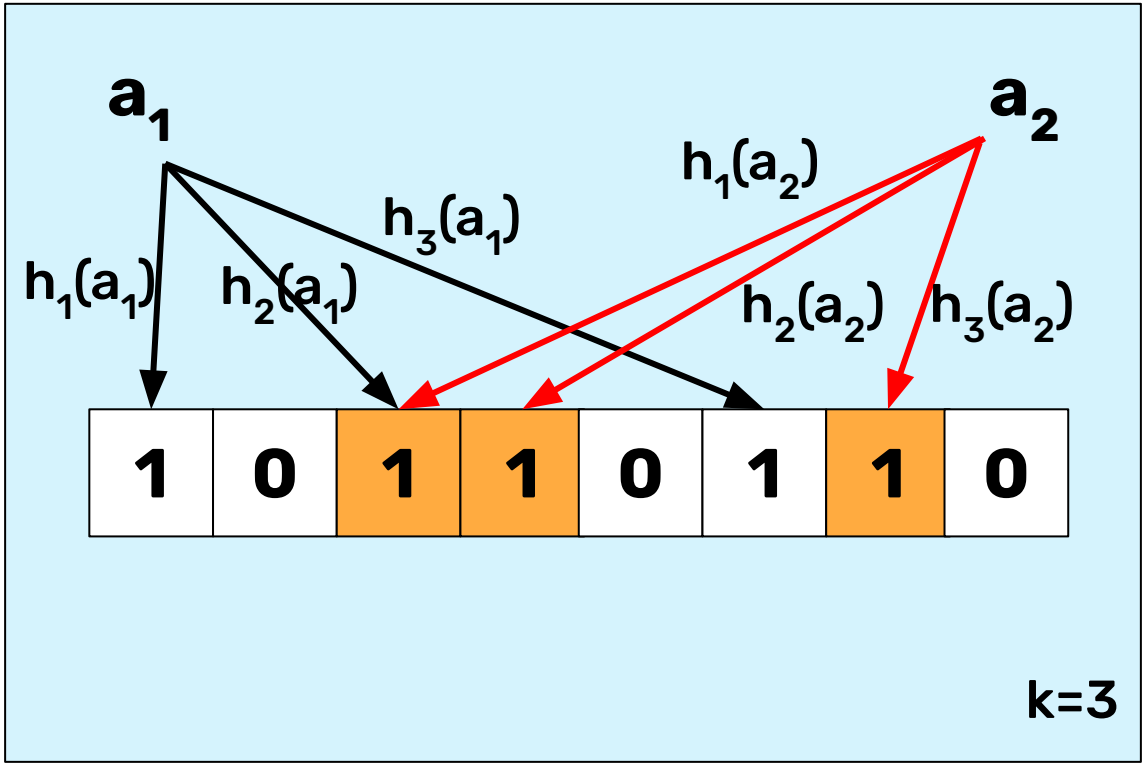
BIGDATA TEAM

Фильтр Блума (Bloom Filter)



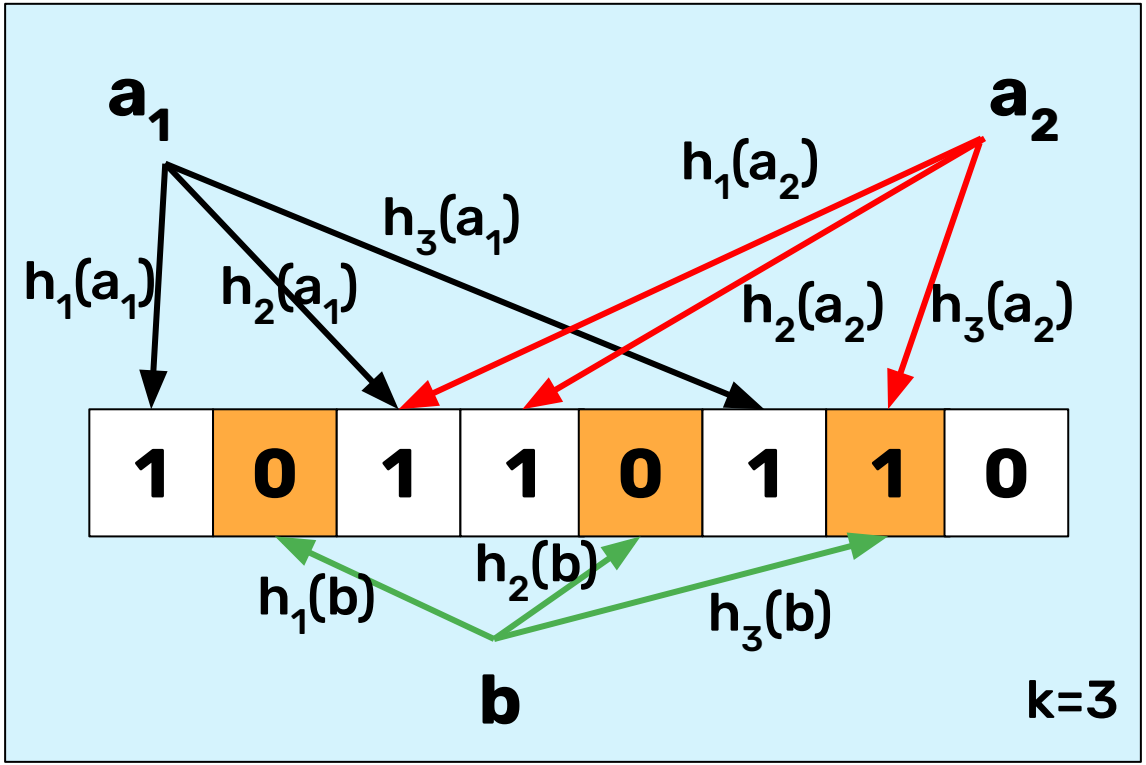
BIGDATA TEAM

Фильтр Блума (Bloom Filter)



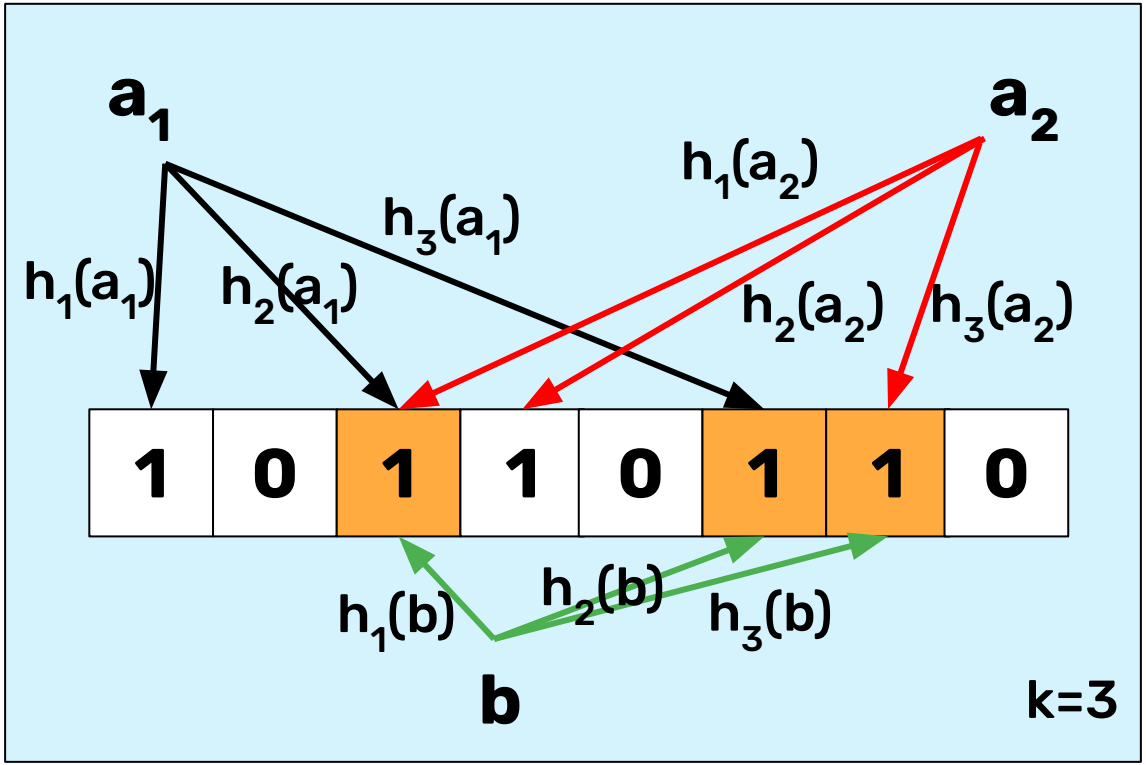
BIGDATA TEAM

Фильтр Блума (Bloom Filter)



BIGDATA TEAM

Фильтр Блума (Bloom Filter)



BIGDATA TEAM

Фильтр Блума (Bloom Filter)

Bloom Filter Calculator

Bloom filters are space-efficient probablistic data structures used to test whether an element is a member of a set.

They're surprisingly simple: take an array of **m** bits, and for up to **n** different elements, either test or set **k** bits using positions chosen using hash functions. If all bits are set, the element *probably* already exists, with a false positive rate of **p**; if any of the bits are not set, the element *certainly* does not exist.

Bloom filters find a wide range of uses, including tracking which articles you've read, speeding up Bitcoin clients, detecting malicious web sites, and improving the performance of caches.

This page will help you choose an optimal size for your filter, or explore how the different parameters interact.

n
Number of items in the filter (optionally with SI units: k, M, G, T, P, E, Z, Y)
4000

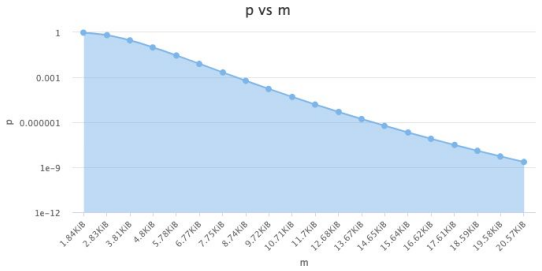
p
Probability of false positives, fraction between 0 and 1 or a number indicating 1-in-p
1.0E-7

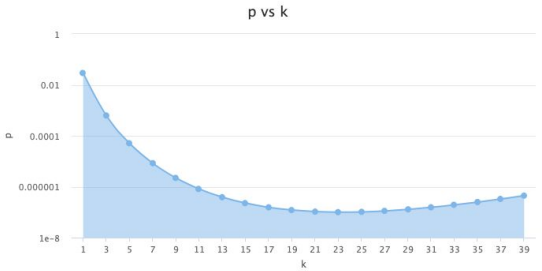
m
Number of bits in the filter (or a size with KB, KiB, MB, Mb, GiB, etc)

k
Number of hash functions
Submit

n = 4,000
p = 0.0000001 (1 in 9,994,297)
m = 134,191 (16.38KiB)
k = 23

```
n = ceil(m / (-k / log(1 - exp(log(p) / k))))
p = pow(1 - exp(-k / (m / n)), k)
m = ceil((n * log(p)) / log(1 / pow(2, log(2))));
k = round((m / n) * log(2));
```





https://hur.st/bloomfilter/

BIGDATA TEAM

NoSQL поверх больших данных

BIGDATA TEAM

Архитектура БД Cassandra

Cassandra - AP система в теореме CAP. На практике это означает:

- высокая доступность данных
- нет транзакций (не совсем)
- можно строить гео-кластера
- слабая согласованность (eventual)
- линейная масштабируемость
- высокая пропускная способность (особенно на запись)

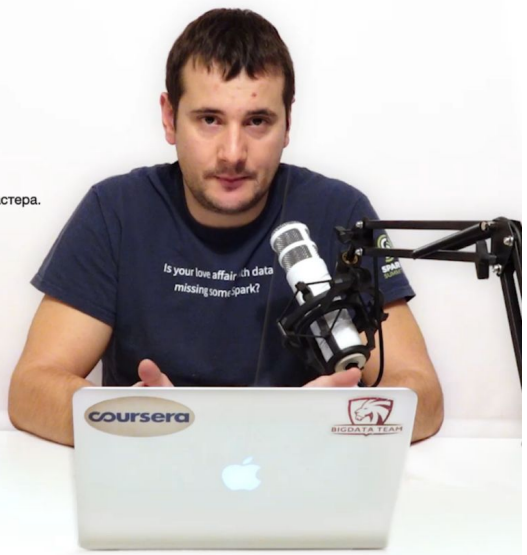
Cassandra имеет симметричную архитектуру. Каждый узел отвечает за хранение данных, обработку запросов и состояние кластера.

Расположение данных определяется значением хеш функции от Partition key.

Высокая доступность данных обеспечивается за счет репликации.

$$\begin{aligned} \text{token}(n_1) &= t_1 \\ \text{token}(n_2) &= t_2 \\ \text{token}(n_3) &= t_3 \\ &\dots \end{aligned} \qquad \begin{aligned} \text{range}(t_1, t_2] &\rightarrow \{n_2, n_3, n_4\} \\ \text{range}(t_2, t_3] &\rightarrow \{n_3, n_4, n_5\} \\ \text{range}(t_3, t_4] &\rightarrow \{n_4, n_5, n_6\} \\ &\dots \end{aligned}$$

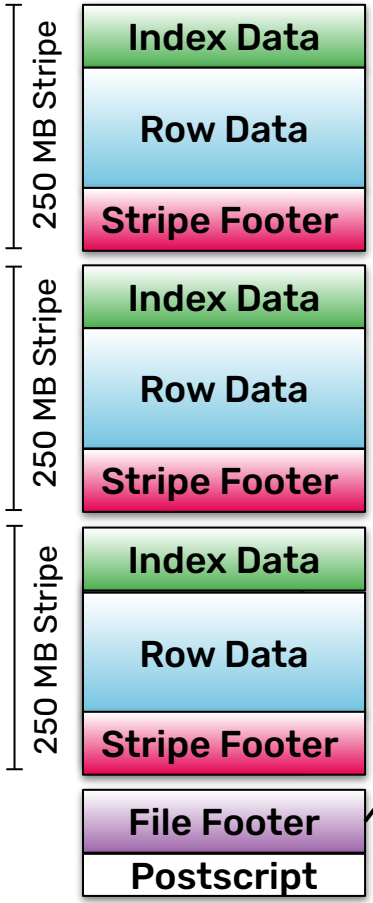
Андрей Титов
BigData Instructor



0:10 / 7:18 Скорость 1.0x

BIGDATA TEAM

ORC File Footer




```
enum CompressionKind {
    NONE = 0;
    ZLIB = 1;
    SNAPPY = 2;
    LZ0 = 3;
    LZ4 = 4;
    ZSTD = 5;
}
```

ORC v.1 спецификация: https://orc.apache.org/specification/ORCv1/

Параметры ORC

Ключ	Значение по умолчанию	Комментарии
orc.compress	ZLIB	верхнеуровневый кодек {NONE;ZLIB;SNAPPY}
orc.compress.size	262,144	число байт (чанк) для сжатия кодеком
orc.stripe.size	67,108,864	число байт в каждом stripe

Параметры ORC

Ключ	Значение по умолчанию	Комментарии
orc.compress	ZLIB	верхнеуровневый кодек {NONE;ZLIB;SNAPPY}
orc.compress.size	262,144	число байт (чанк) для сжатия кодеком
orc.stripe.size	67,108,864	число байт в каждом stripe

```
CREATE TABLE my_orc_table (
    ...
)
STORED AS orc
    TBLPROPERTIES ("orc.compress"="NONE")
...;
```