

Технологии и разработка СУБД

Введение в распределенные системы

Анастасия Лубенникова
Александр Алексеев

В этой лекции

- Движения NoSQL и NewSQL
- Распределенные транзакции
- Векторные часы
- CRDT
- Gossip
- И всякое такое
- Также см лекцию про репликацию и фейловер

Материала много



Движение NoSQL

Johan Oskarsson, then a developer at Last.fm, reintroduced the term NoSQL in early 2009 when he organized an event to discuss "open source distributed, non relational databases".

© <https://en.wikipedia.org/wiki/NoSQL>

В чем идея NoSQL баз данных

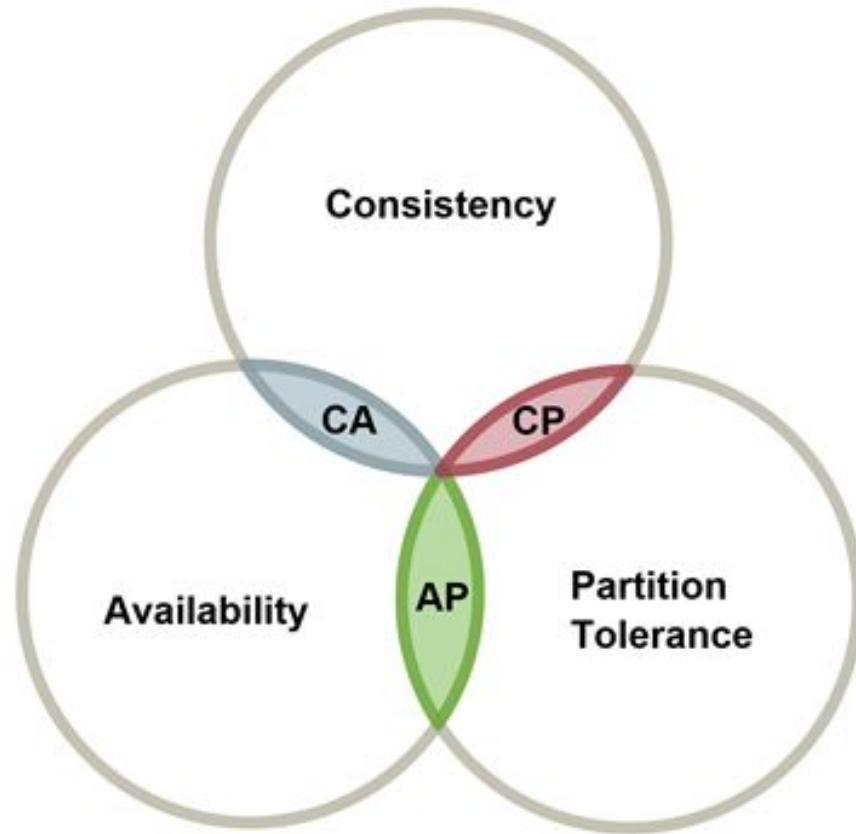
- Распределенные;
- Отказоустойчивые;
- Зачастую - жертвуя консистентностью или переопределяя ее значение;
- Упрощение схемы или полное ее отсутствие;

PACELC (a.k.a CAP-теорема)

To me, CAP should really be PACELC --- if there is a partition (P) how does the system tradeoff between availability and consistency (A and C); else (E) when the system is running as normal in the absence of partitions, how does the system tradeoff between latency (L) and consistency (C)?

<http://dbmsmusings.blogspot.ru/2010/04/problems-with-cap-and-yahoos-little.html>

CAP: неправильное объяснение



CRDT & Eventual Consistency

Conflict-free replicated data type (CRDT) is a data structure which can be replicated across multiple computers in a network, where the replicas can be updated independently and concurrently without coordination between the replicas, and where it is always mathematically possible to resolve inconsistencies which might result.

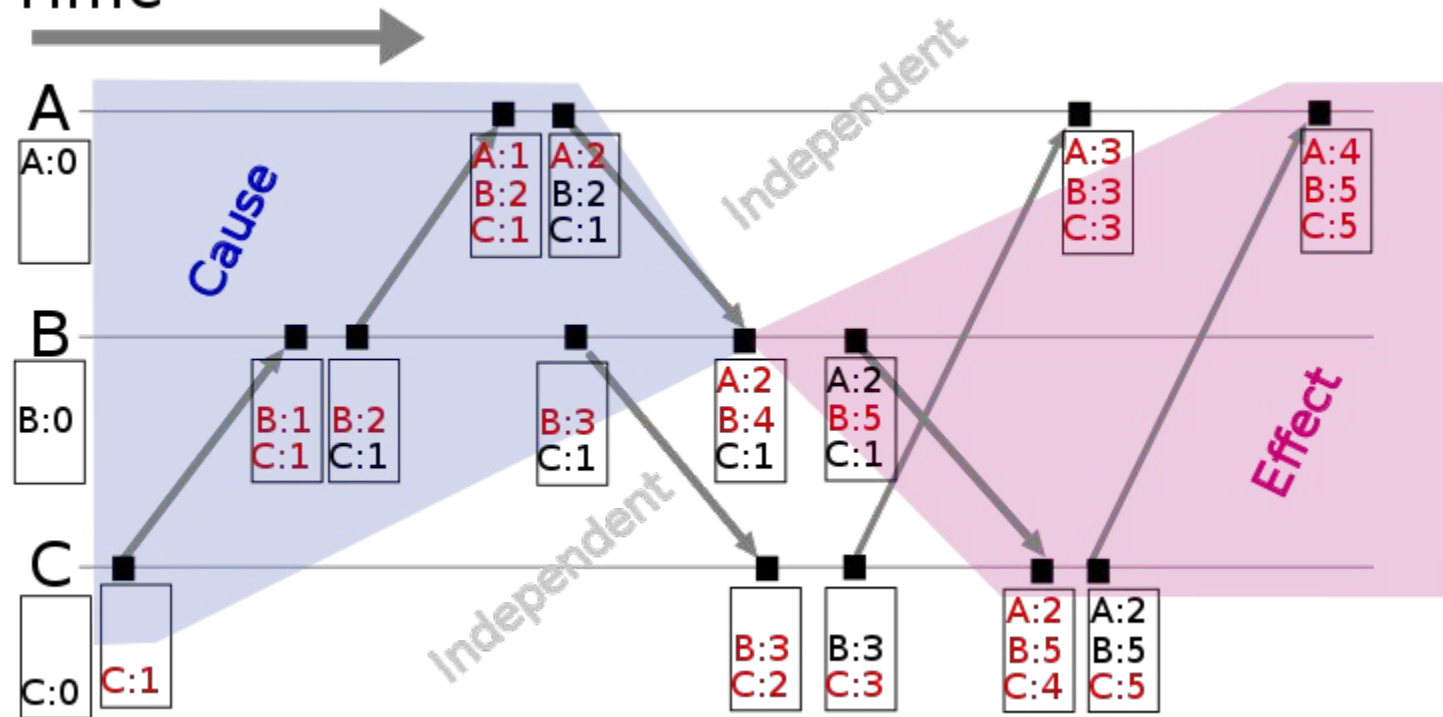
https://en.wikipedia.org/wiki/Conflict-free_replicated_data_type

Is it better to be alive and wrong or right and dead?

— Jay Kreps, A Few Notes on Kafka and Jepsen (2013)

Векторные часы

Time



Gossip Protocol



Jepsen



<https://aphyr.com/tags/Jepsen>

Пара слов о современном железе: RAM

- В один физический сервер влезает до 3 TB оперативы;
- AWS инстансы x1.32xlarge (128 vCPU, 1952 GB RAM, 2 x 1920 GB SSD) стоят 9603\$ в месяц [1];
- Также AWS анонсировал новые инстансы с 4-16 TB RAM [2][3].

[1]: <https://aws.amazon.com/ec2/pricing/on-demand/>

[2]: <https://aws.amazon.com/ec2/instance-types/x1e/>

[3]: https://www.theregister.co.uk/2017/05/16/aws_ram_cram/

Пара слов о современном железе: жесткие диски

- Сегодня можно купить 1 TB SSD за ~300\$ [1];
- В один физический сервер можно запихнуть до 900 TB данных;
- В следующем году - до 1.5 PB.

[1]: Samsung MZ-75E1T0BW, <https://market.yandex.ru/product/11929060>

Виды NoSQL баз данных

- Key-value (Memcached, Redis, Riak, ...)
- Документо-ориентированные (MongoDB, Couchbase, CouchDB, RethinkDB)
- Колоночные (ClickHouse)
- Графовые (Neo4j)
- Для полнотекстового поиска (ElasticSearch, Solr, Sphinx)
- Гибридные (Cassandra, Tarantool)
- Сюда же: месседж брокеры / очереди сообщений (RabbitMQ, Kafka)

Чуть подробнее: Memcached

- Key-Value
- Язык программирования: C
- Автор: Брэд Фитцпатрик
- Данные хранятся только в памяти
- Длина ключа до 250 байт, значения - до 1 Мб
- **Fun fact!** Ключ можно “расширить” с помощью хэш-функций, значения можно нарезать на части
- Данные вытесняются по алгоритму LRU
- Ввод-вывод осуществляется при помощи libevent
- Поддерживается TCP и UDP
- Есть текстовый и бинарный протокол

Чуть подробнее: Redis

- Key-Value
- Язык программирования: C
- Автор: Salvatore Sanfilippo (a.k.a. antirez)
- Данные хранятся в памяти + опционально снапшоты и WAL
- Есть репликация
- Есть поддержка массивов, множеств, словарей, bitmaps, можно указать TTL, также есть механизм publish / subscribe
- Сервер однопоточный
- Есть поддержка транзакций и пакетного выполнения команд
- См также Redis Cluster

Чуть подробнее: Riak

- Key-Value
- Язык программирования: Erlang (в основном)
- Основан на Dynamo-пейпере [1]
- Отсутствует единая точка отказа
- Все обмазано CRDT, векторными часами, gossip'ом, read-repair'ами и антиэнтропией
- Есть два протокола на выбор: REST API и Protobuf
- Есть несколько бэкендов: Bitcask, LevelDB, Memory
- Есть репликация между ДЦ а.к.а. XDC (была платной, теперь открыта)

[1]: <http://www.allthingsdistributed.com/files/amazon-dynamo-sosp2007.pdf>

Protobuf (1 / 2)

```
message Person {  
    required string user_name = 1;  
    optional int64 favorite_number = 2;  
    repeated string interests = 3;  
}
```

Protobuf (1 / 2)

```
message Person {
```

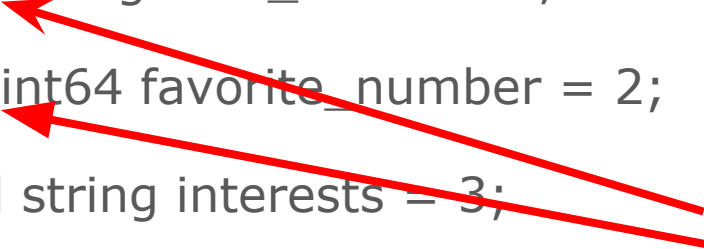
```
  required string user_name = 1;
```

```
  optional int64 favorite_number = 2;
```

```
  repeated string interests = 3;
```

```
}
```

Fun fact! В Protobuf 3 все поля всегда optional.

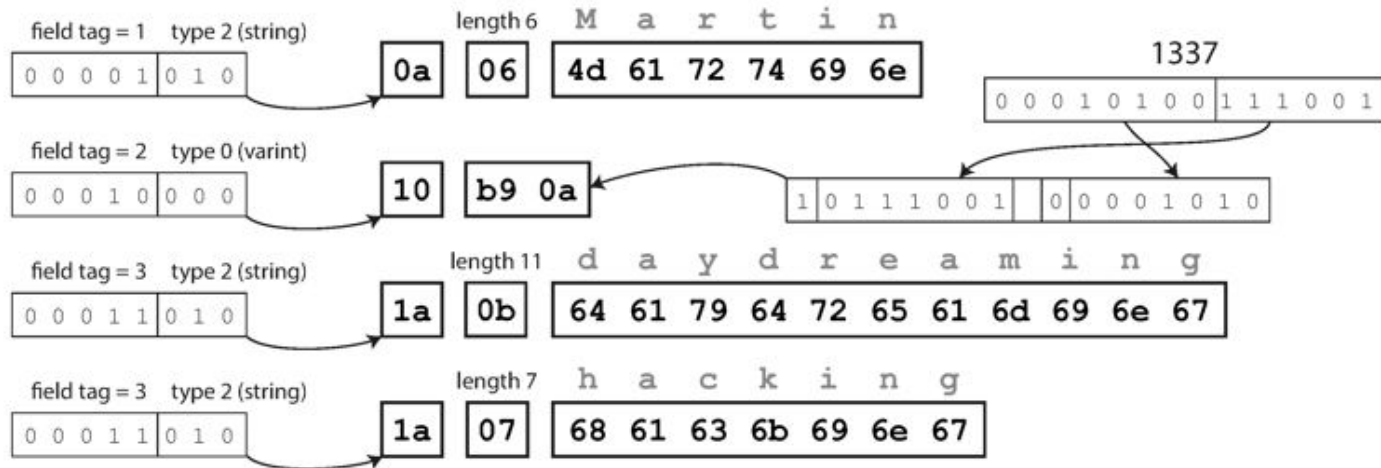


Protobuf (2 / 2)

Byte sequence (33 bytes):

0a	06	4d	61	72	74	69	6e	10	b9	0a	1a	0b	64	61	79	64	72	65	61
6d	69	6e	67	1a	07	68	61	63	6b	69	6e	67							

Breakdown:



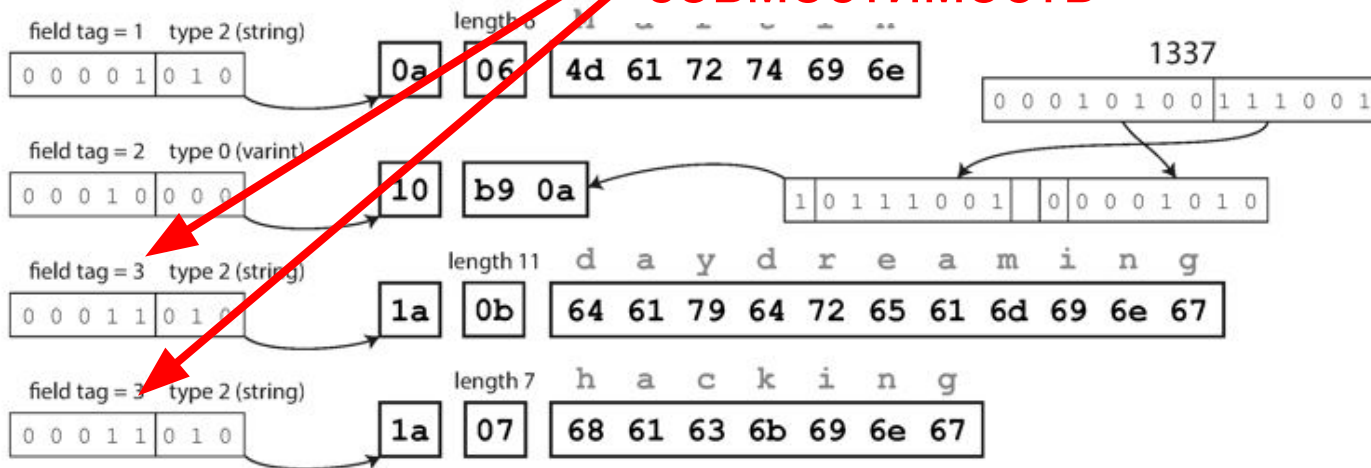
Protobuf (2 / 2)

Byte sequence (33 bytes):

0a	06	4d	61	72	74	69	6e	10
6d	69	6e	67	1a	07	68	61	63

Fun fact! Поле можно сделать repeated, не сломав обратную совместимость

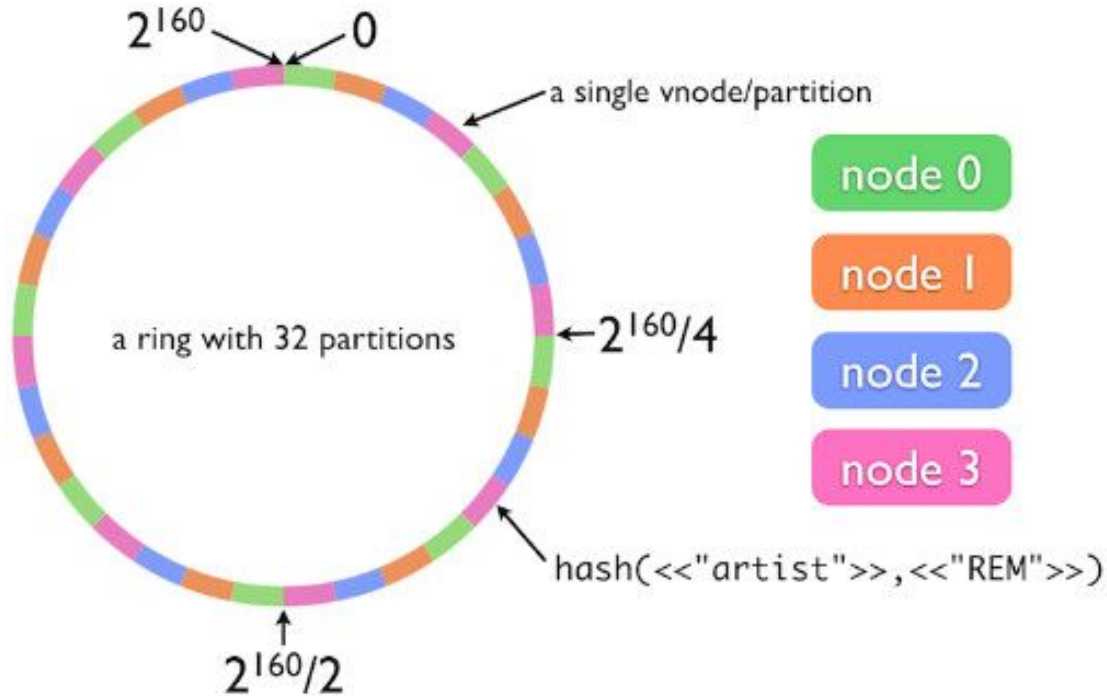
Breakdown:



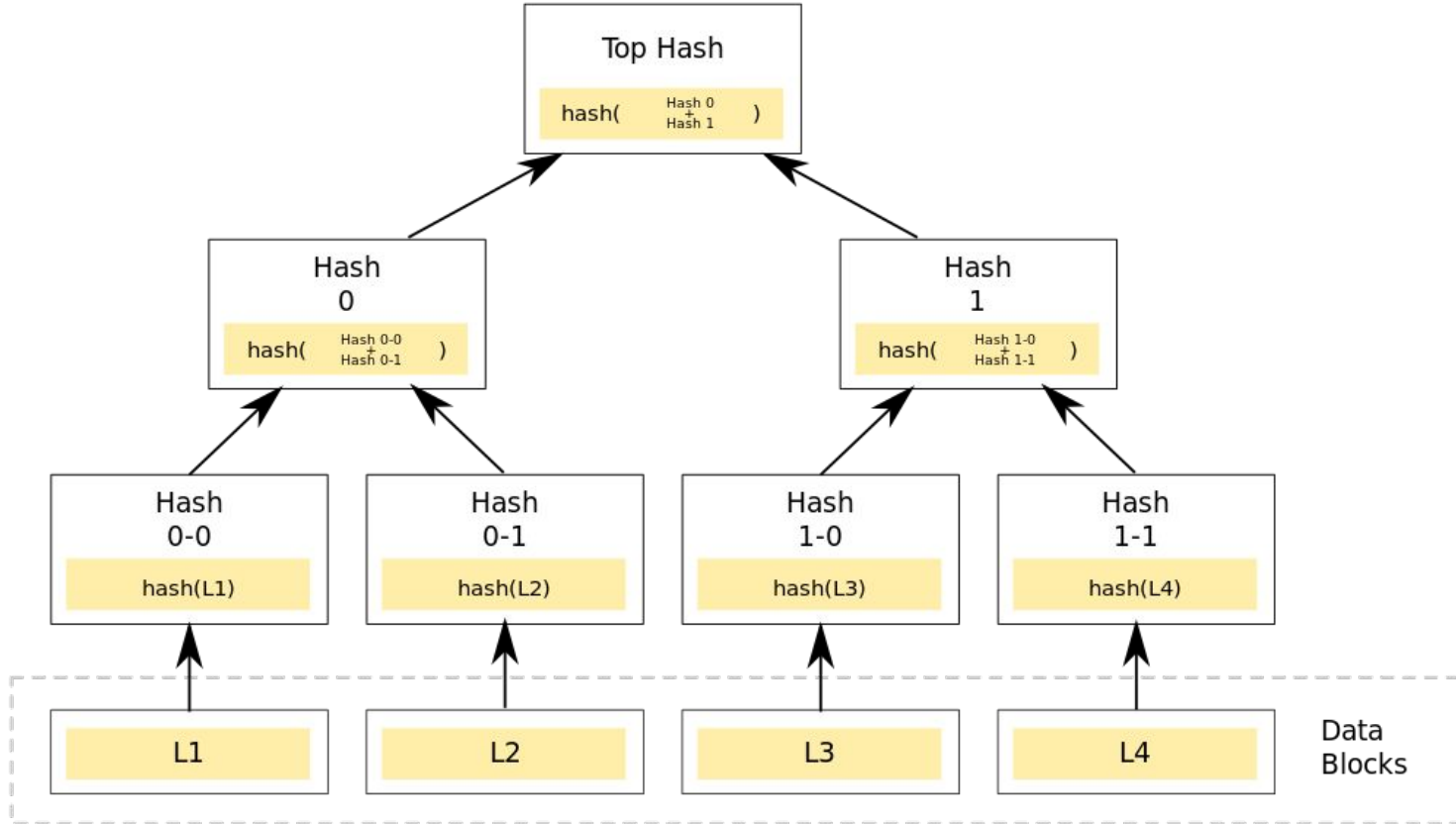
See Also

- Thrift
- Avro
- MessagePack
- Cap'n Proto
- ???

Riak: ring & vnodes



Riak: Merkle Tree (a.k.a. Hash Tree)



Чуть подробнее: MongoDB

- Документо-ориентированная СУБД
- Языки программирования: C++ и JavaScript
- Документы имеют JSON-подобную структуру
- Атомарность обеспечивается на уровне одного документа
- Buffer manager сделан на основе mmap
- Есть поддержка репликасетов с автофейловером
- Шардинг из коробки по диапазону, используется специальный shard key
- Есть сжатие, вторичные индексы и другие навороты
- В свое время прославилась склонностью терять данные и не проходить Jarsen (но вроде сейчас с этим стало лучше)

MongoDB: примеры запросов

```
> db.urls.insert({ code: 123, url: "https://google.com/" });
```

```
> db.urls.find();
```

```
> db.urls.ensureIndex({ code: 1 }, { unique: true });
```

```
> db.urls.getIndexes();
```

```
> db.urls.dropIndex({ code: 1 });
```

```
> db.urls.update({ code: 123 }, { url: "http://example.ru/" });
```

```
> db.urls.remove({ code: 123 });
```

Чуть подробнее: Couchbase

- Документо-ориентированная СУБД (на самом деле, ближе к Key-Value)
- Языки программирования: Erlang и C++
- Данные хранятся в vBuckets, по умолчанию их число равно 1024
- Каждый узел в кластере является мастером и репликой неких vBucket'ов
- Номер vBucket'а документа определяется по хэшу от ключа
- Горячие данные кэшируются в памяти по принципу LRU
- Есть поддержка Memcached-бакетов, хранящихся только в памяти
- Есть репликация между ДЦ
- Есть очень красивая веб-админка!
- См также N1QL, Couchbase Lite и другие навороты

Чуть подробнее: ClickHouse

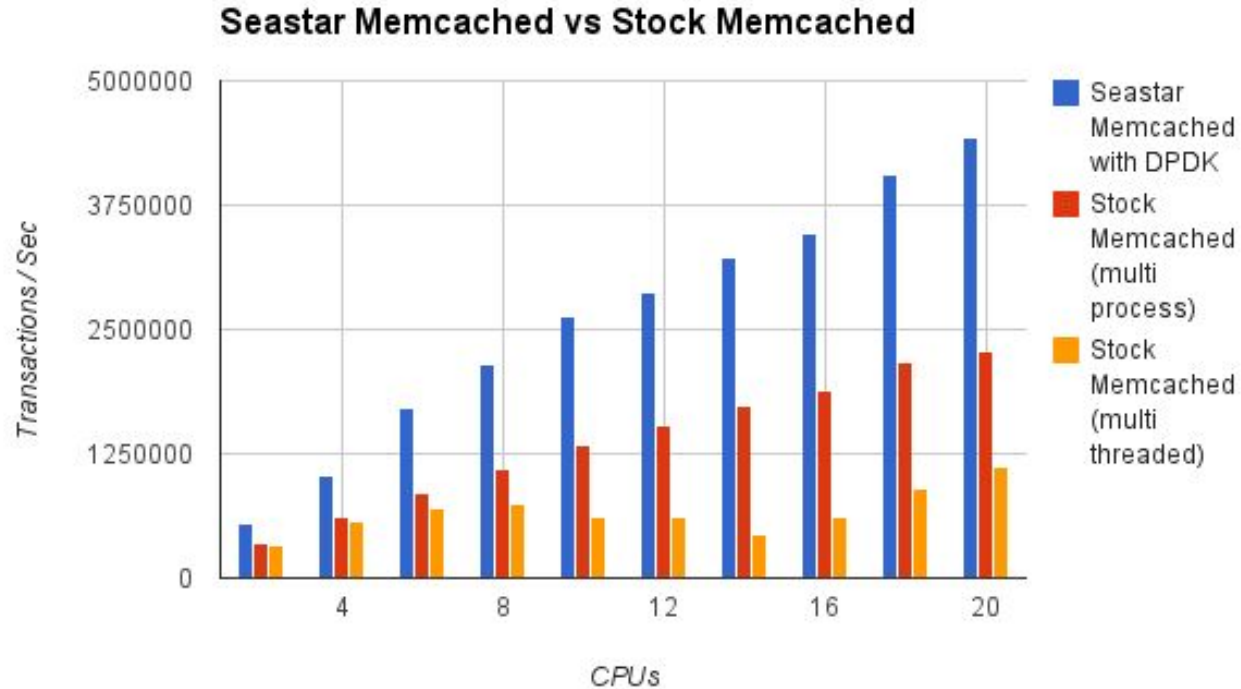
- Честная колоночная СУБД
- Язык программирования: C++
- Разрабатывается в Яндексе
- Поддерживает некий диалект SQL
- На ACID забили
- Есть сжатие колонок
- Есть распараллеливание и векторизация запросов
- Репликация данных зависит от ZooKeeper [1]

[1]: <https://github.com/yandex/ClickHouse/issues/479>

Чуть подробнее: Cassandra

- “Колоночно-ориентированная” СУБД
- Язык программирования: Java
- Создана в 2008-м году в Facebook
- Как и Riak, основана на Dynamo-пейпере
- Данные хранятся в виде троек (column_name, value, timestamp)
- Last Write Wins по умолчанию
- Поддерживает упрощенный SQL (Cassandra Query Language)
- Есть эффективные вторичные индексы, primary key разбивается на partition key (для шардинга) и clustering key (для сортировки внутри шарда).
- Данные хранятся в LSM-tree
- См также ScyllaDB, Seastar, Redis

Seastar (1 / 2)



Seastar (2 / 2)

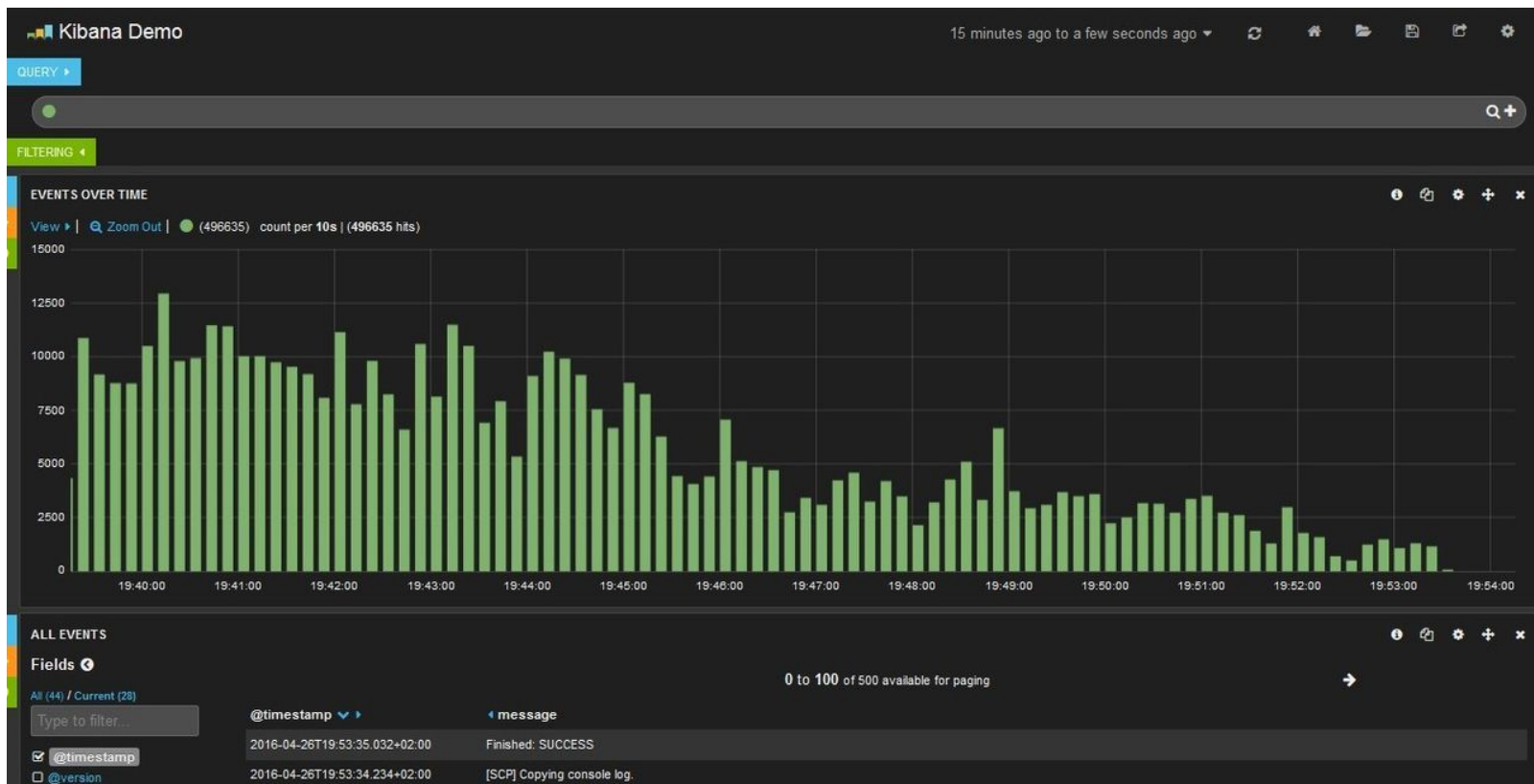
Пока что не занесли нормальной поддержки CMake :(

- <https://github.com/scylladb/seastar/issues/313> - ишью
- <https://github.com/scylladb/seastar/issues/264> - есть воркэраунд

Чуть подробнее: Elasticsearch

- Приложение для полнотекстового поиска
- Язык программирования: Java
- Основан на движке Apache Lucene
- Имеет довольно запутанный HTTP/JSON API
- Не очень хорошо переживает нетсплиты и падения узлов
- Часто используется в сочетании с Logstash и Kibana, так называемый ELK-стэк

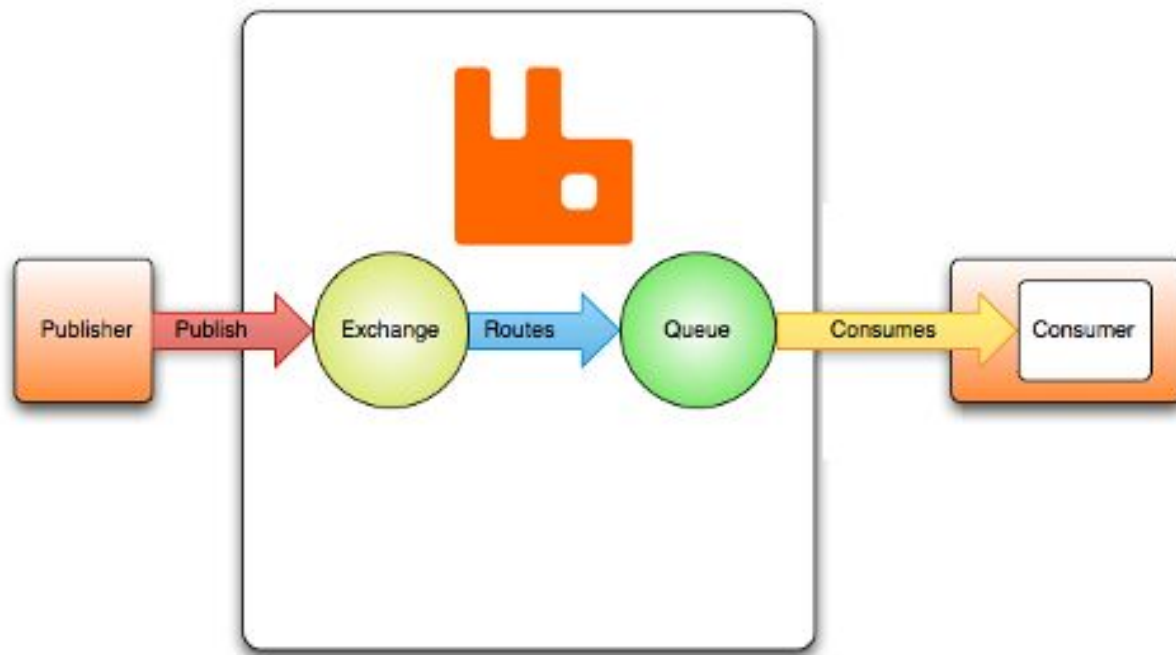
Kibana



Чуть подробнее: RabbitMQ

- Брокер сообщений
- Язык программирования: Erlang
- Реализует протокол AMQP
- Умеет хранить сообщения в памяти и персистентно (бэкенд - Mnesia)
- Как показывает опыт, в последнем случае не очень хорошо ведет себя при нетсплитах
- Области применения:
 - Раздача задач воркерам - например, рассылка SMS или Email, тяжелые вычисления и тд;
 - Подписка на события в системе - например, входящие сообщения для заданного пользователя;
 - Remote Procedure Call;

RabbitMQ: иллюстрация



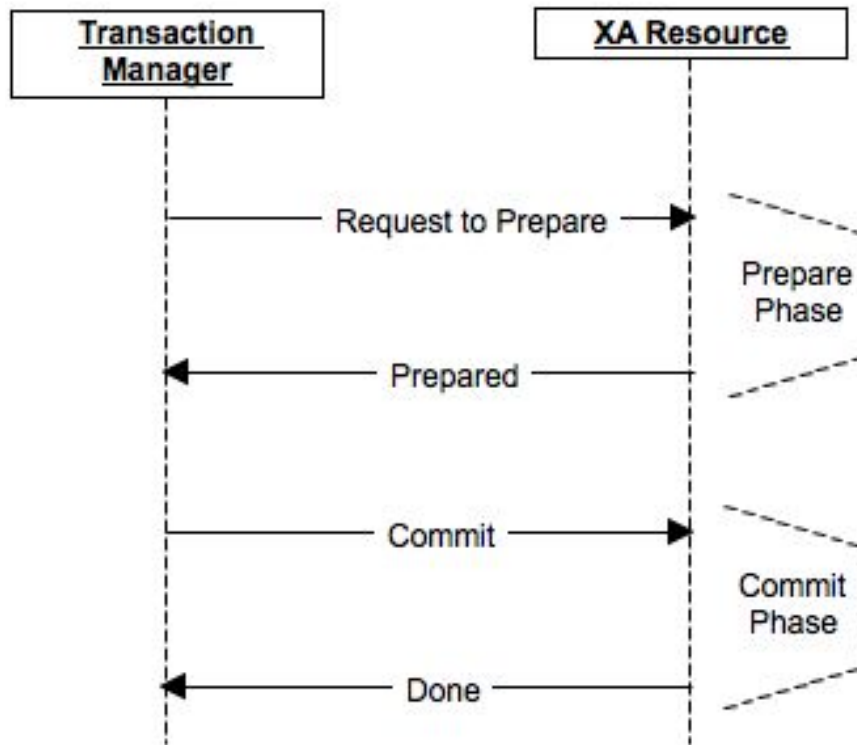
Транзакции в мире NoSQL

Можно сделать разными способами.

- Вести лог идемпотентных операций;
- Использовать Percolator-like подход [1];

[1]: <http://rystsov.info/2012/09/01/cas.html>

Чем плох 2PC?



NewSQL базы данных

Как традиционные РСУБД, только распределенные и шардированные.

Примеры NewSQL баз данных

- CockroachDB
- TiDB
- Amazon Aurora
- Google Spanner

Чуть подробнее: CockroachDB

- ACID с прозрачным фейловером, шардингом и распределенными транзакциями;
 - Анонсирован в 2014, написан на Go, разрабатывается экс-гугловцами;
 - Бесплатное и открытое ПО;
 - Совместимо с PostgreSQL на уровне протокола;
 - Проходит Jepsen [*];
 - Основывается на пейпере о Spanner [*];
-
- <https://www.cockroachlabs.com/blog/cockroachdb-beta-passes-jepsen-testing/>
 - <https://static.googleusercontent.com/media/research.google.com/en//archive/spanner-osdi2012.pdf>

Чуть подробнее: Amazon Aurora

- ACID с прозрачным фейловером и всяким таким;
- Анонсирована в 2014;
- Существует только в AWS;
- Совместима с MySQL и PostgreSQL [1] на уровне протокола;
- Есть пейпер [2];

[1]: с ноября 2016 <https://news.ycombinator.com/item?id=13072861>

[2]: <http://www.allthingsdistributed.com/files/p1041-verbitski.pdf>

Дополнительные материалы

- The Raft Consensus Algorithm

<https://raft.github.io/>

- Seastar

<http://www.seastar-project.org/>

- DPDK

<http://dpdk.org>

Рекомендуемые книги

- Seven Databases in Seven Weeks

<https://pragprog.com/book/rwdata/seven-databases-in-seven-weeks>

- Distributed Systems for Fun and Profit

<http://book.mixu.net/distsys/single-page.html>

- Designing Data-Intensive Applications

<http://dataintensive.net/>



Seven Databases in Seven Weeks

A Guide to Modern Databases
and the NoSQL Movement

Erie Redmond
and Jim R. Wilson

Series editor: Bruce A. Tate
Development editor: Jurgens J. Carter



O'REILLY

Designing Data-Intensive Applications

THE BIG IDEAS BEHIND RELIABLE, SCALABLE,
AND MAINTAINABLE SYSTEMS



Martin Kleppmann

Рекомендуемые блоги

- Martin Kleppman
<https://martin.kleppmann.com/>
- Adrian Colyer
<https://blog.acolyer.org/>
- Kyle Kingsbury
<https://aphyr.com/>
- Salvatore Sanfilippo
<http://antirez.com/>

Вопросы и ответы.

- a.lubennikova@postgrespro.ru
- a.alekseev@postgrespro.ru
- Telegram: <https://t.me/dbmsdev>