

CAP

Согласованность обновлений

- Конфликт “запись-запись”
 - Пессимистичный подход - транзакции
 - Медленная работа
 - Deadlock
 - Распределенные блокировки
 - Оптимистичный
 - Обновлять не всегда
 - Разрешение конфликтов
- Репликация повышает вероятность конфликтов “запись-запись”

Согласованность “чтение-запись”

- Когда возникает?
- В NoSQL нет транзакций - не совсем правда
- Есть атомарность операций с конкретным “агрегатом”
 - Как помогает денормализация?
- Окно несогласованности

Согласованность репликаций

- При чтении с разных реплик значение будет одним и тем же
- Как добиться?
 - Согласованность в рамках своей сессии.

Eventual Consistency

- Есть много серверов.
- В любой момент времени данные могут быть не согласованными, но, если нет новых обновлений, в конце концов все узлы получают одно и то же значение

Quorum

- Нужен для повышения вероятности правильного ответа
- Не обязательно “все или ничего”
 - Кворум записи ($W > N/2$)
 - Кворум чтения ($R + W > N$)

CAP

- Consistency
 - Записи атомарны
 - Все запросы после записи получают новое значение
- Availability
 - Должно возвращаться значение до тех пор, пока хотя бы один сервер работает
- Partition tolerance
 - Система продолжает функционировать, если распадается на части

Как выбирать?

1. Можно только 2 из 3
2. Выбираем, распределенная или нет
3. Выбираем, от чего отказываться (А или С)

Пример CAP

- CA - не выдерживают Partition Tolerance
- CP - должен ответить только если знает правильный ответ
- AP - должен всегда ответить, если спрашивают
 - Ответ может быть “не знаю”
 - Eventual Consistency

Примеры

- CA - PostgreSQL, Redis, Neo4J
- CP - MongoDB, HBase
- AP - CouchDB, ElasticSearch

Как выбирать БД?

- Смотреть на пары букв? Слишком просто!
- В некоторых БД можно выбирать интересующие буквы:
 - Cassandra - Replication Factor
 - MongoDB - количество узлов в Replica Set
- Настраивать часто можно еще и драйвер:
 - Cassandra
 - MongoDB