In - Memory базы данных (IMDB)

С развитием выч. техники (память стала больше и дешевле, начиная с 90-х)

Чтение / запись на диск - самая дорогая операция в БД

Произв-ть БД = фоновые процессы + основные процессы;

Удаление самое дорогое ⇒
⇒ давайте его не делать ⇒
⇒ давайте хранить всю БД в оперативной памяти;

Примеры: Redis, Tarantool; - постреляционные БД

ACID (транзакции)

данные в БД буферизуются на след. уровнях:

- м.б. приложения;
- СУБД;
- ОС;

→ С Durable и IMDB проблема, т.к. ОЗУ энергозависимая;

Методы повышения безопасности:

1) Snapshots; ⇒ часто: высокая нагрузка
⇒ редко: риск потерять данные;

2) Логи транзакций; (если БД маленькая, то логов м.б. больше, чем данных)

3) NVRAM и NVDIMM; ↳ а их тоже приходится дампить;
↳ батарейные модули, которые могут удерживать состояние ОЗУ при отключении питания;

Redis: key-value архитектура;

Одна большая хеш-таблица;
таблицы можно вкладывать, но больше подходит для плоских структур

Длинные ключи - плохо;

А зачем Redis, если я могу у себя в приложении всё хешировать?
- платформонезависимость;
- отв-ть не на нас

Надо придерживаться схемы при построении ключей: „object-type:id:field";

Redis:
- Умеет инкрементировать/декрементировать число в строковом представлении;
- Бинарно-безопасный формат хранения строк;

Master-slave репликация

Master: балансировка, хранение и изменение данных, распр-е API-запросов по Slave'ам;

Slave: не связаны, могут стать Master, хранение дубликатов (реплик), чтение данных;
(снимки изменений)

Всех можно разнести на разные сервера;

Master назначается вручную;

Redis Sentinel:
автоматический выбор Master;

Sentinel instance

Redis Cluster (для отказоустойчивости):

горизонтальное

Master'ов 3+ штук, все связаны между собой;

Master выбирается клиентом, и теперь клиент является балансировщиком (Redis Client);

В Postgres так тоже можно, но там придерживаются stand alone архитектуры;

Если гор. масштабировать Postgres, то получится Greenplum;

Рост науки — на стыке сфер

Ограниченная интенсивность — косяк приложения;

В OLTP строки таблиц считается целиком;

У Redis из коробки есть чат, организованный по принципу „Подписчик-издатель"

монитор „Читатели-писатели"