

Базы данных. Лекция 12.

@mikhirurg

May 2020

1 Теорема CAP

В распределённой информационной системе возможно обеспечить не более двух из перечисленных трёх свойств:

- Согласованность данных (Consistency)
- Доступность (Availability)
- Устойчивость к разделению (Partition Tolerance)

[Статья о теореме CAP](#)

Мы хотим реализовать систему, где люди могут записывать свои идеи, и в любой момент получить записанные им ранее факты.

Есть записная книжка, куда мы записываем факты.

Решения:

- **Решение 1.** Централизованная система
Один человек обрабатывает запросы
– **Проблема доступности при росте нагрузки (no Availability)**
- **Решение 2.** Система с независимыми узлами и балансировкой нагрузки
Мы берем на работу компаньона, даём ему книжку и распределяем запросы к нашей системе случайно.
Но если один человек сперва позвонит на линию и запишет данные у одного оператора, а потом попросит эти же данные у другого оператора, он их не получит. Система не консистентна.
– **Проблема согласованности (no Consistency)**

- **Решение 3.** Распределённая система с транзакционной репликацией данных
 Мы будем делиться полученными данными с нашим компаньоном, синхронизировать их. Только получив отклик от компаньона, мы можем гарантировать клиенту, что его данные записаны.
 Если компаньон ушел со своего рабочего места, система остановится на неопределённое время. Мы будем ждать синхронизации с другим узлом.
 – **Проблема доступности при недоступности хотя бы одного узла.**
- **Решение 4.1.** Распределённая система с гарантированной репликацией данных.
 Будем иногда проверять, находится ли наш компаньон на рабочем месте
 Но, если были проблемы со связью компаньон не ответит на нашу проверку и система остановится на неопределённое время.
 – **Невозможно реализовать, если потеряна связь с узлами (no Partition Tolerance). CA система.**
- **Решение 4.2.** Распределённая система с гарантированным откликом.
 Будем считать, что вероятность, того, что клиент запросит факт, который недавно записал очень мала. Тогда можно в пределах некоторого времени осуществлять синхронизацию.
 – **Целостность данных в конечном счёте (Eventually Consistency). AP система.**
- **Решение 4.3.** Распределённая система с гарантированной целостностью данных.
 Гарантируем целостность и блокируем обращение клиентов, если мы не можем достоверно подтвердить статус компаньона. Допускаем, что система может расщепляться, но если она станет таковой, перестает обрабатываться какой-то тип запросов от клиентов до разрешения конфликта.
 – **Невозможно обеспечить полную доступность. CP система.**

1.1 Решения CA, AP и CP.

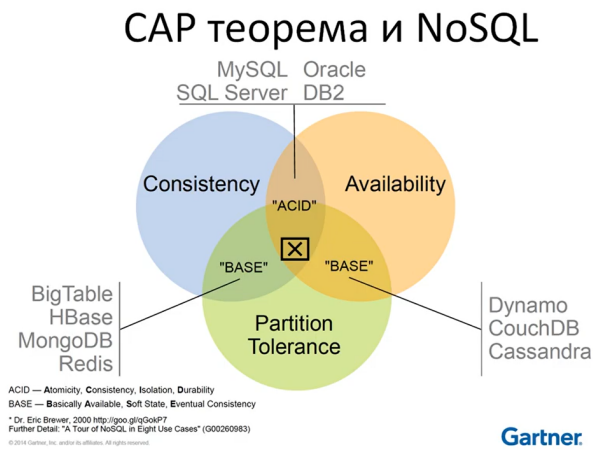


Рис. 1: Пример

1.1.1 Хранилище ключ-значение

- Использует принцип ассоциативного массива
- Базовые операции:
 - **INSERT** (ключ, значение)
 - **FIND** (ключ)
 - **REMOVE** (ключ)
- Как правило реализует класс AP.

1.1.2 Документоориентированные БД

- Хранение иерархических структур данных (документов)
- Использование стандартных нотаций для описания структур документов: XML, JSON.
- Документы идентифицируются ключами
- Есть CP и AP Реализации.

<pre>{ CustomerID: 17, OrderItems: [{ ProductID: 84, Quantity: 3 }, { ProductID: 19, Quantity: 8 }], OrderDate: 19.05.2020 }</pre>	<pre>{ CustomerID: 21, OrderItems: [{ ProductID: 11, Quantity: 43 }], OrderDate: 19.05.2020, PromoCode: 123456 }</pre>
--	--

Рис. 2: Пример

1.1.3 Колоночные БД

- Хранение всей таблицы, но с физическим расположением не записей построчно, а атрибутов по колонкам (может использоваться и с реляционной схемой).
- Хранение нескольких таблиц с дублированием ключевого атрибута (в пределе переходим к "ключ-значение").
- Как правило СА, но могут быть и CP.

RowId	EmpId	Lastname	Firstname	Salary
001	10	Smith	Joe	60000
002	12	Jones	Mary	80000
003	11	Johnson	Cathy	94000
004	22	Jones	Bob	55000

Рис. 3: Пример

1.1.4 Графовые БД

- Два типа хранимых объектов: узлы и дуги. У каждого могут быть свои атрибуты.
- Удобно для хранения соц. сетей, онтологий и баз знаний.

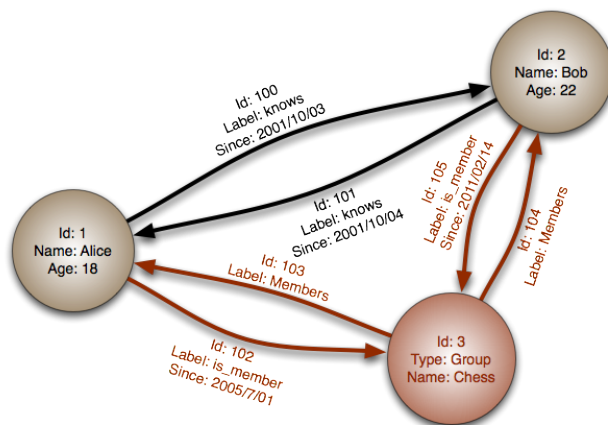


Рис. 4: Пример