

Поговорим о согласованности (это сейчас важно, так как мы выбираем базу данных).

Состояние согласованности данных - это когда во всех узлах системы (допустим если у вас несколько микросервисов и несколько баз данных) данные одинаковые (то есть, например, поле обновления баланса пользователя БалансА = 10 во всех узлах системы после проведения платежа).

Большинство современных систем строятся на основе множества компонентов — микросервисная архитектура и так далее. Таким образом, нам точно важно быть устойчивым к разделению (выполнять Р из теоремы CAP). И остаётся выбирать между согласованностью данных и доступностью системы. Это очень строгое ограничение теоремы CAP, и один из ее недостатков — она говорит нам строго выбирать между одним или другим. Также имеет место упрощение взгляда на современные системы.

Чтобы решить проблемы компромиссов и ограничений теоремы CAP, возникли два альтернативных подхода к обработке согласованности в распределенных системах - ACID и BASE. Эти подходы фокусируются на различных аспектах согласованности, помогая разработчикам выбрать наилучшую модель для конкретного случая использования.

- **ACID (Atomicity, Consistency, Isolation, Durability) - строгая согласованность**

ACID это акроним, представляющий собой набор свойств, которые обеспечивают надежную обработку транзакций в системах баз данных.

- "Атомарность" (Atomicity) гарантирует, что транзакция будет выполнена полностью или не будет выполнена вовсе. Это означает, что если в процессе транзакции происходит ошибка, все изменения, сделанные в рамках этой транзакции, откатываются, и система возвращается в состояние, в котором она была до начала транзакции.
- "Согласованность" (Consistency) обеспечивает, что каждая успешно завершенная транзакция приводит систему из одного согласованного состояния в другое. Система контролирует это, применяя серию правил или ограничений на допустимые операции.
- "Изолированность" (Isolation) обеспечивает, что каждая транзакция выполняется независимо от других. Это означает, что в процессе исполнения одной транзакции, результаты других транзакций, выполняющихся параллельно, не влияют на нее.
- "Надежность" (Durability) гарантирует, что после успешного выполнения транзакции все ее результаты остаются постоянно сохраненными в базе данных, даже в случае последующих сбоев или ошибок. Таким образом, результаты транзакции не могут быть потеряны после того, как она была зафиксирована.

ACID - это набор свойств, используемых традиционными базами данных, такими как реляционные базы данных, для обеспечения надежной согласованности. Он гарантирует, что транзакции базы данных обрабатываются надежно, сохраняя данные точными и непротиворечивыми. Приоритетом ACID является строгая согласованность, что означает, что данные должны быть обновлены и согласованы на всех узлах, прежде чем транзакция будет считаться успешной.

- ACID-совместимые инструменты
 - Реляционные системы управления базами данных (RDBMS), такие как PostgreSQL, MySQL, Oracle и SQL Server, обычно поддерживают ACID-транзакции.
 - Некоторые NoSQL базы данных, такие как Google Spanner и CockroachDB, также поддерживают ACID-транзакции.

Плюсы:

- Надёжность: ACID обеспечивает строгие гарантии в отношении согласованности и надёжности данных. Если транзакция успешно завершена, все изменения будут безопасно сохранены, и состояние данных будет согласованным.
- Простота разработки: Так как ACID транзакции обеспечивают строгую согласованность, разработчикам проще разрабатывать и реализовывать бизнес-логику, так как они могут полагаться на определенные гарантии.
- Отладка: В случае проблем с данными, проблему обычно проще найти и исправить, поскольку состояние системы более предсказуемо.

Минусы:

- Масштабирование: ACID транзакции могут создавать проблемы при масштабировании, так как они требуют строгой согласованности данных между всеми узлами в распределенной системе.
- Требования к инфраструктуре: ACID-системы обычно требуют более стабильной и надежной инфраструктуры, что может привести к увеличению затрат на обслуживание и поддержку.