Базы данных

Лекция 11

Мацнев Никита

Распределенная БД

Определение

- Распределенная база данных — это набор логически взаимосвязанных данных, которые физически распределены между несколькими узлами в некоторой компьютерной сети.

Распределенная БД

Мотивация

- Масштабируемость
- Высокая доступность и отказоустойчивость
- Производительность

Распределенная БД

Модели распределения данных

- Репликация
- Шардинг
- Replication + Sharding

Распределенная БД

РБД?

- Партиционирование — это вертикальное разделение таблицы по строкам внутри одной базы данных

Распределенная БД

Репликация

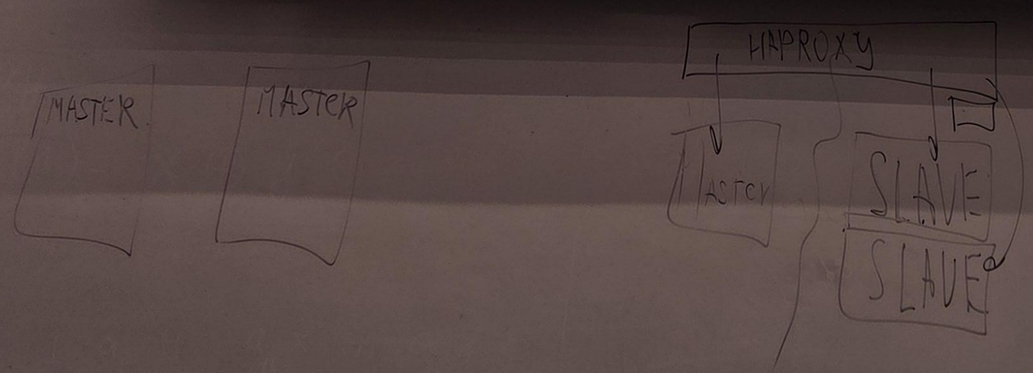
- Синхронная
- Асинхронная

Архитектуры реплицирования

- Master-Slave
- Multi-Master
- Peer-to-Peer

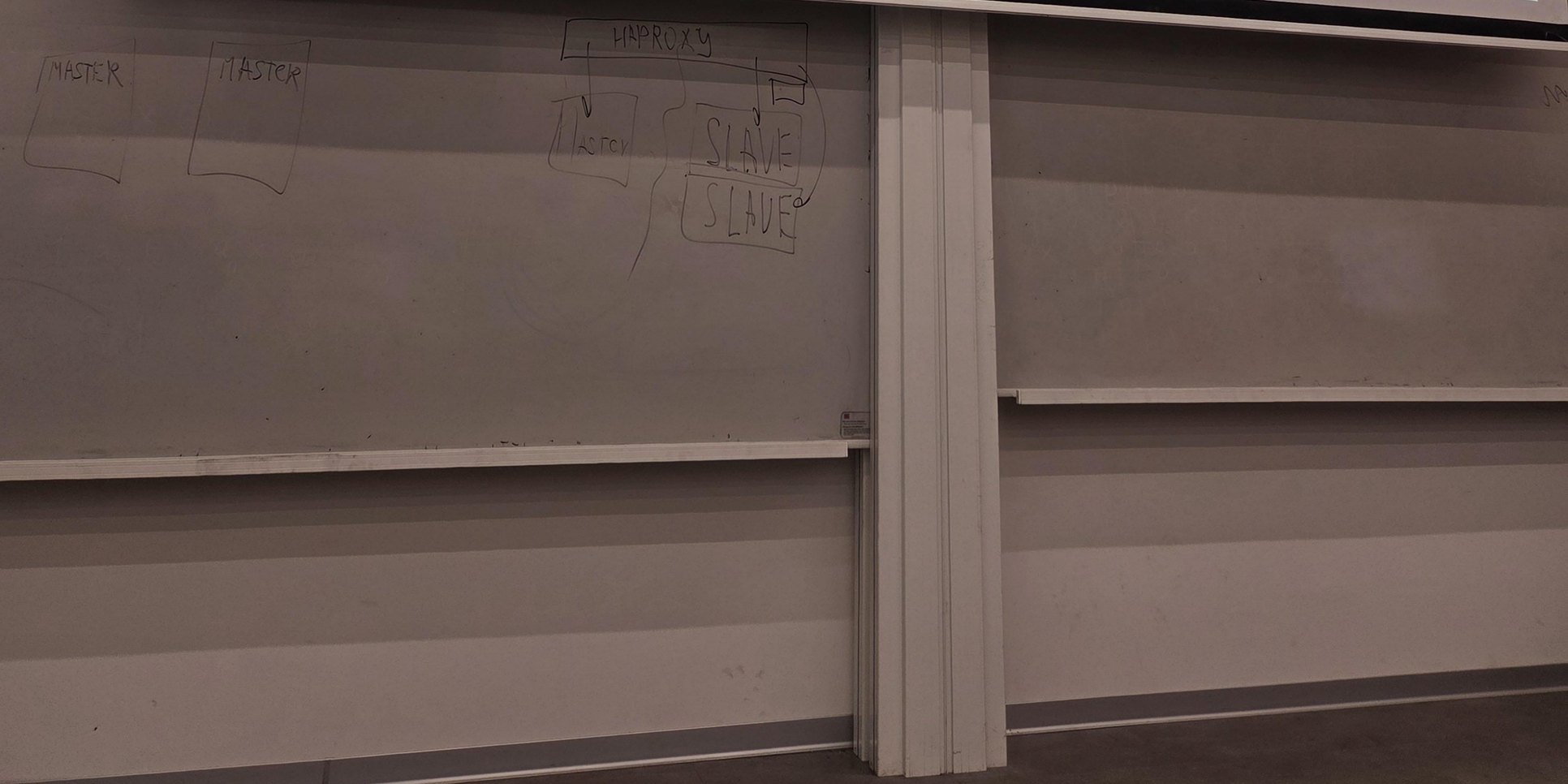
Стратегии шардирования

- Шардирование по диапазону
- Шардирование по хэшу
- Шардирование по справочнику
- Геошардинг



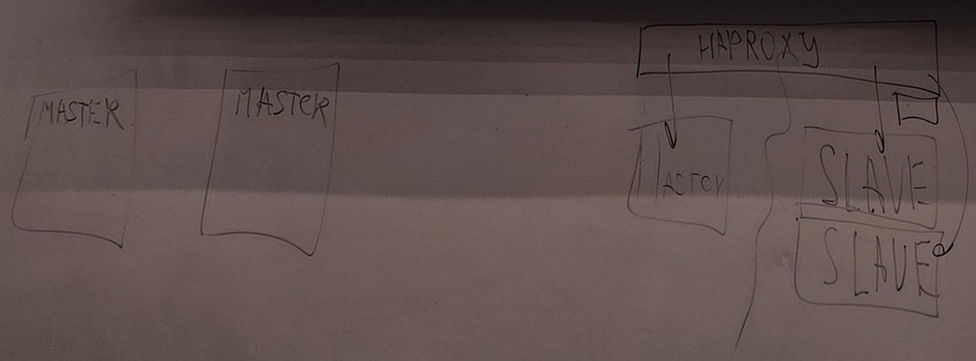
Проблемы шардирования

- Сложность запросов (JOINs)
- Транзакции (требует дорогих Two-Phase Commit)
- Решардинг
- Роутинг запросов (нужен координатор)
- Неподходящие ключи шардирования
- Кайфанешь от SELECT COUNT(*)



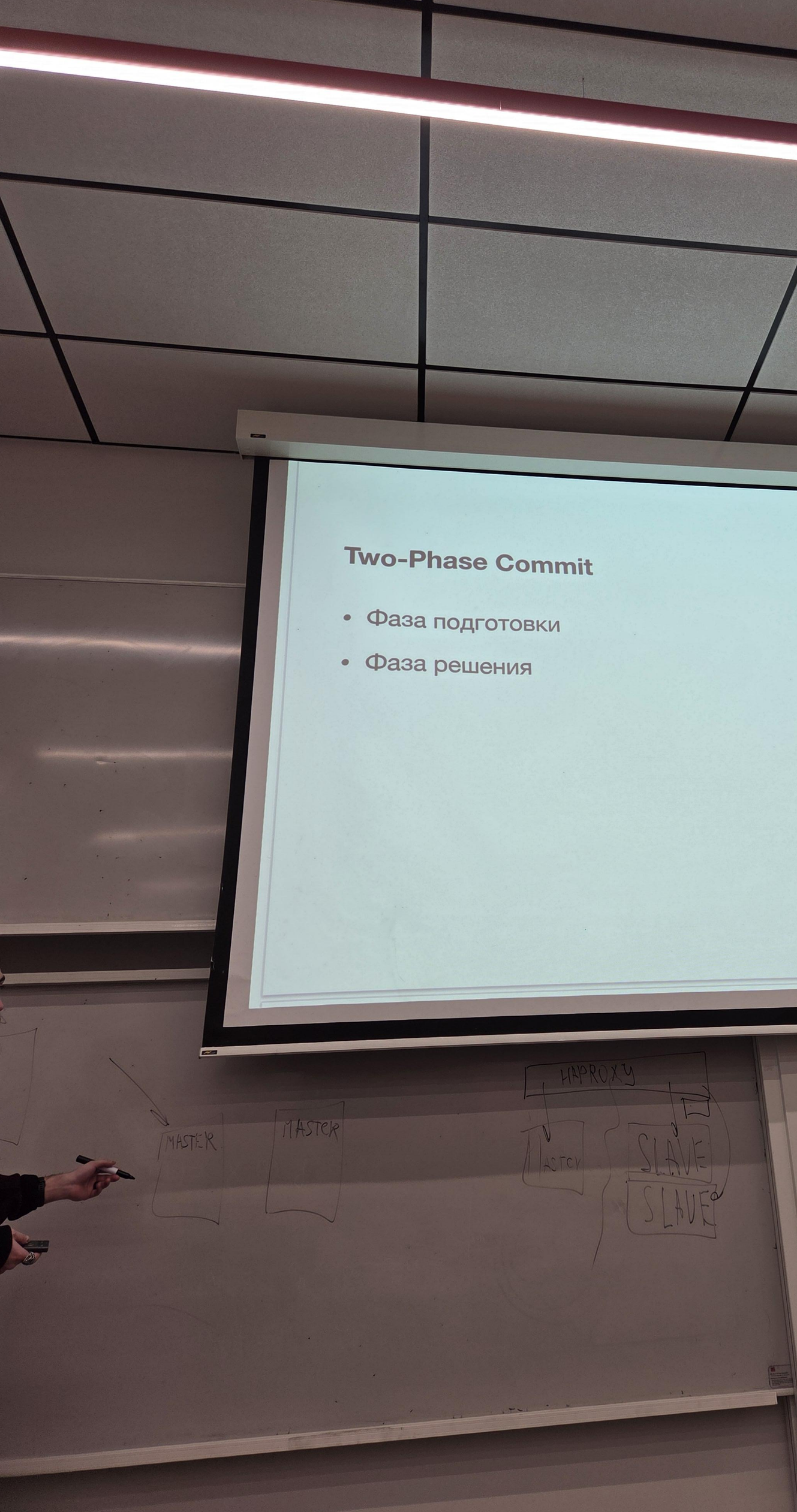
Типы распределенных СУБД

- Однородные (Homogeneous)
- Неоднородные (Heterogeneous)



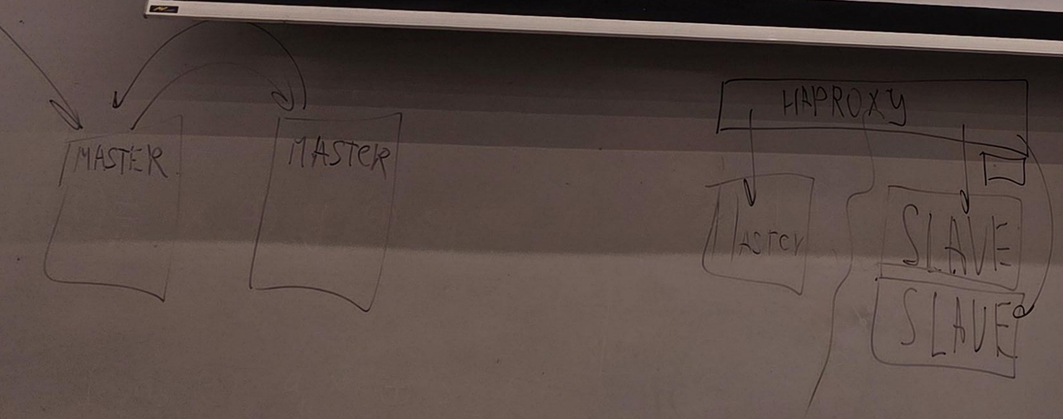
Two-Phase Commit

- Фаза подготовки
- Фаза решения



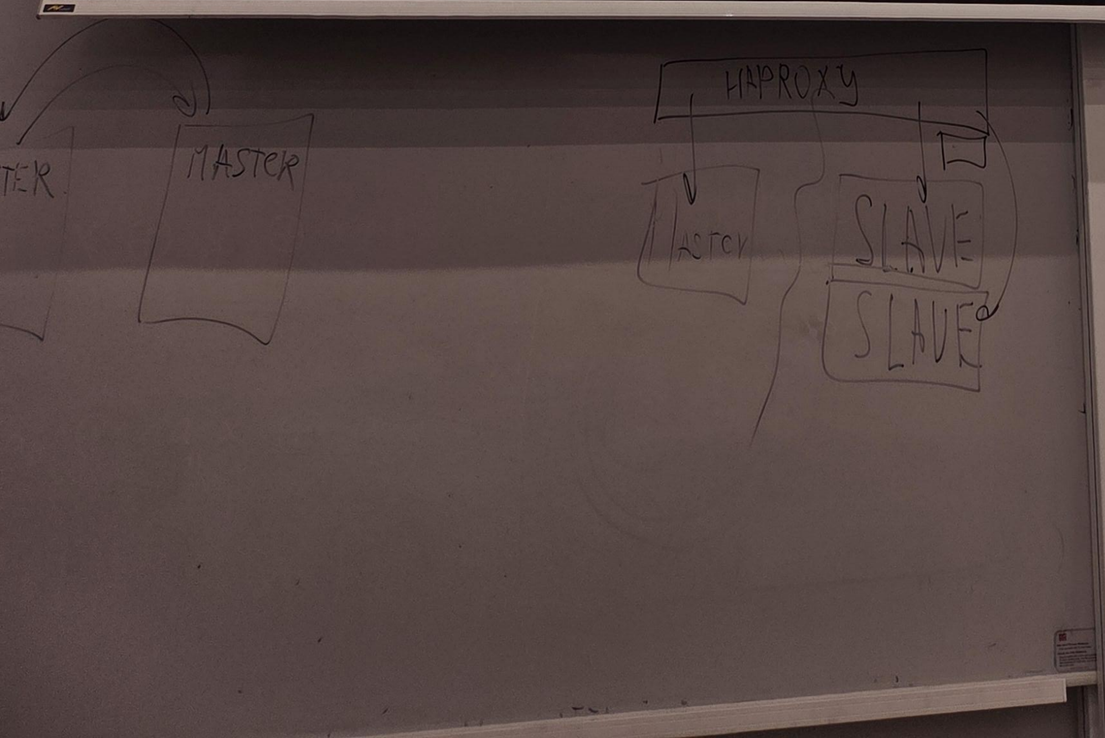
недостатки 2PC

- Протокол требует двух раундов сетевого взаимодействия
- Каждая транзакция ждет ответов от всех участников
- Координатор - единая точка отказа



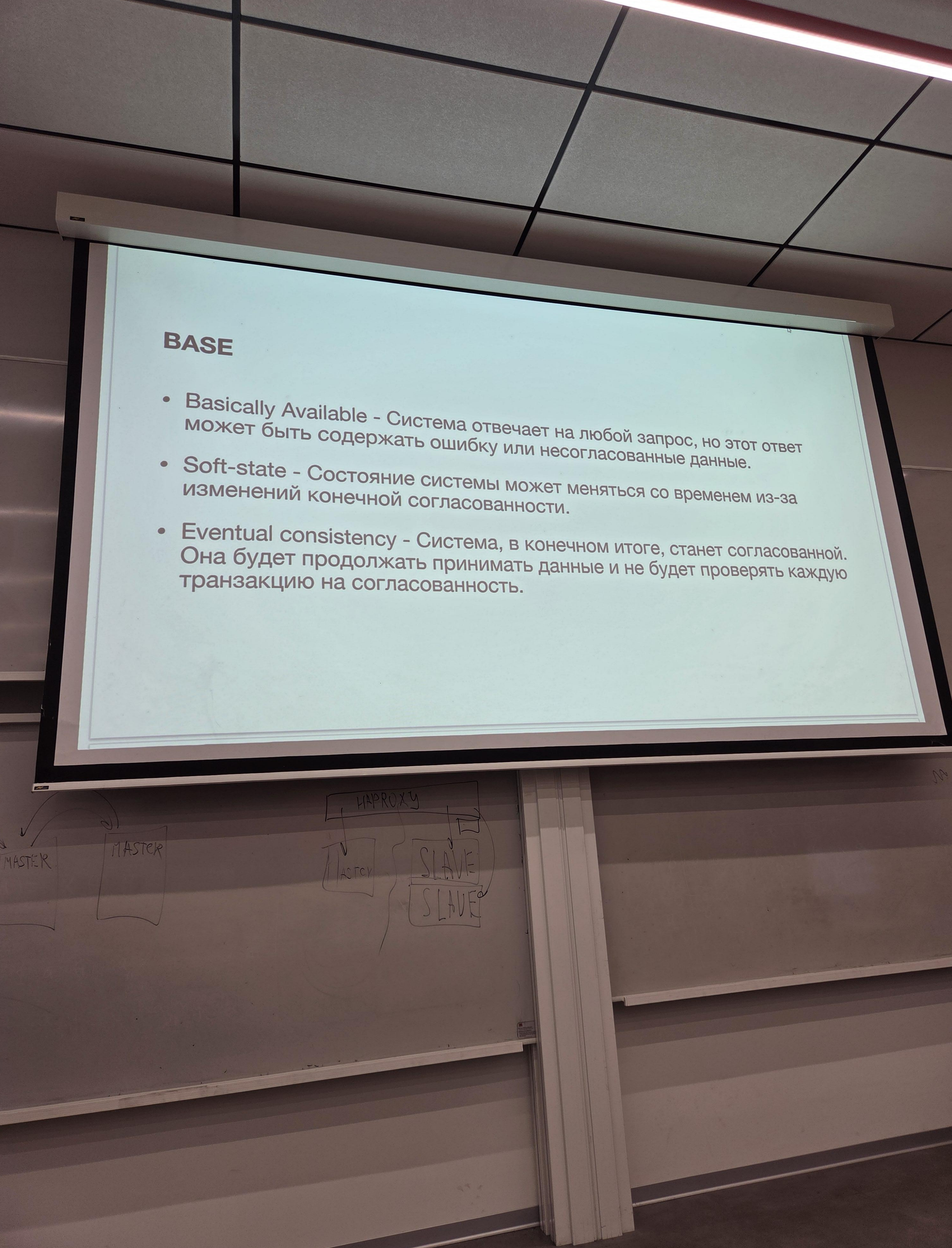
BASE

- BASE — это свойство, которая жертвует немедленной строгой согласованностью в обмен на высокую доступность и производительность.



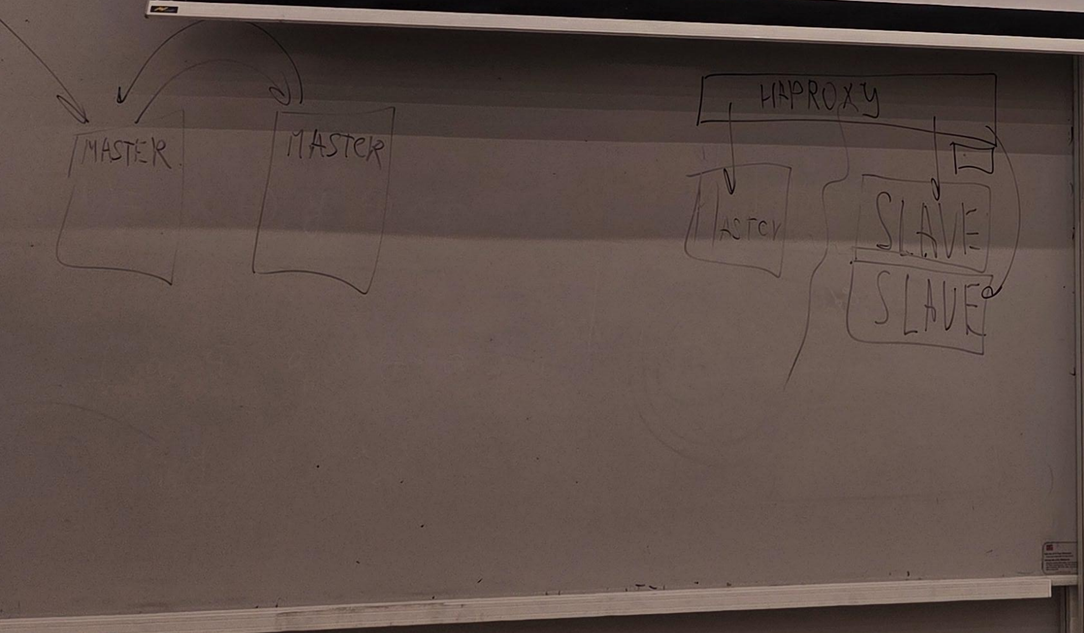
BASE

- Basically Available - Система отвечает на любой запрос, но этот ответ может быть содержать ошибку или несогласованные данные.
- Soft-state - Состояние системы может меняться со временем из-за изменений конечной согласованности.
- Eventual consistency - Система, в конечном итоге, станет согласованной. Она будет продолжать принимать данные и не будет проверять каждую транзакцию на согласованность.

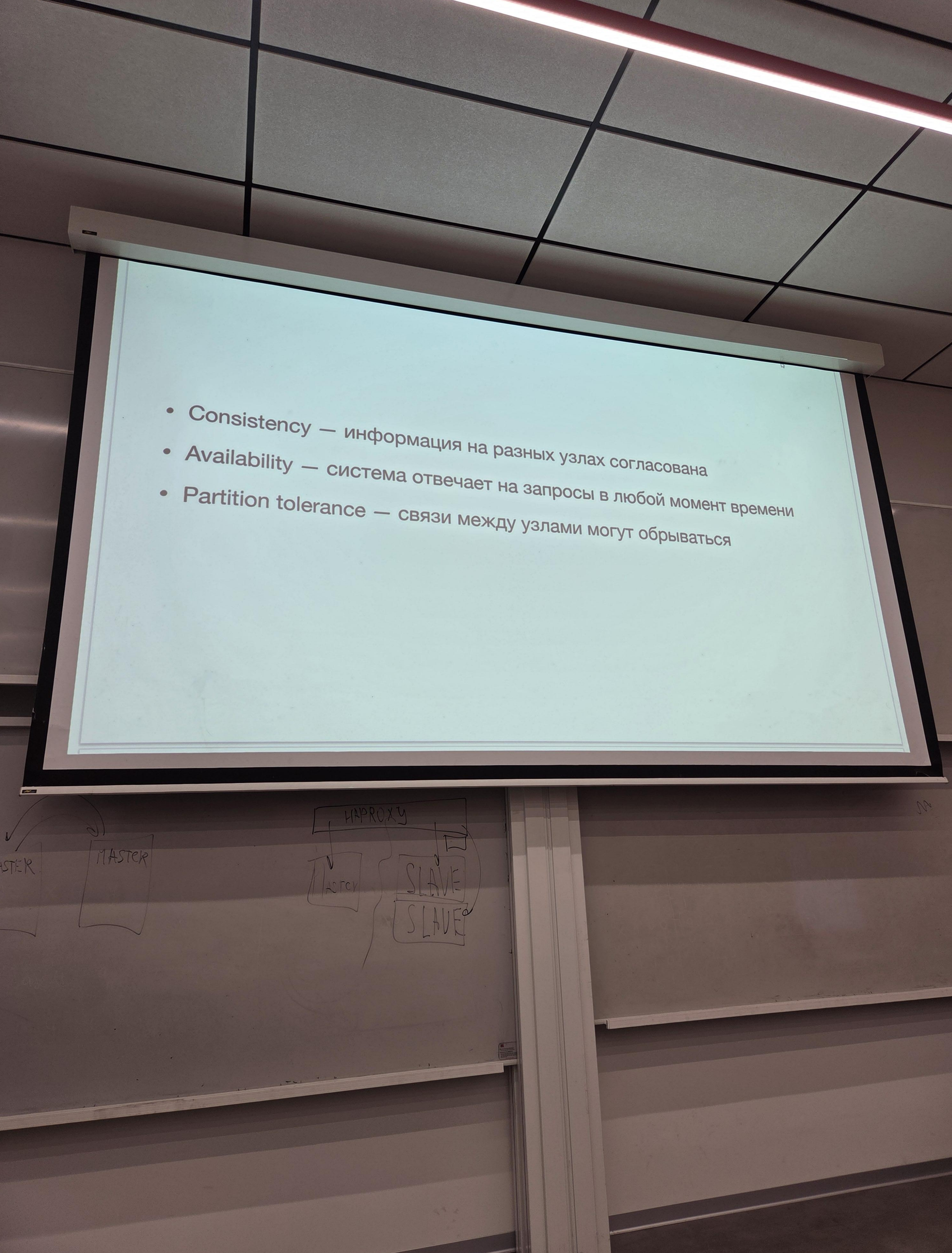


CAP теорема

- CAP теорема (теорема Брюера) — утверждает, что в любой распределенной системе, обеспечивающей хранение данных, невозможно одновременно гарантировать более двух из трех свойств.

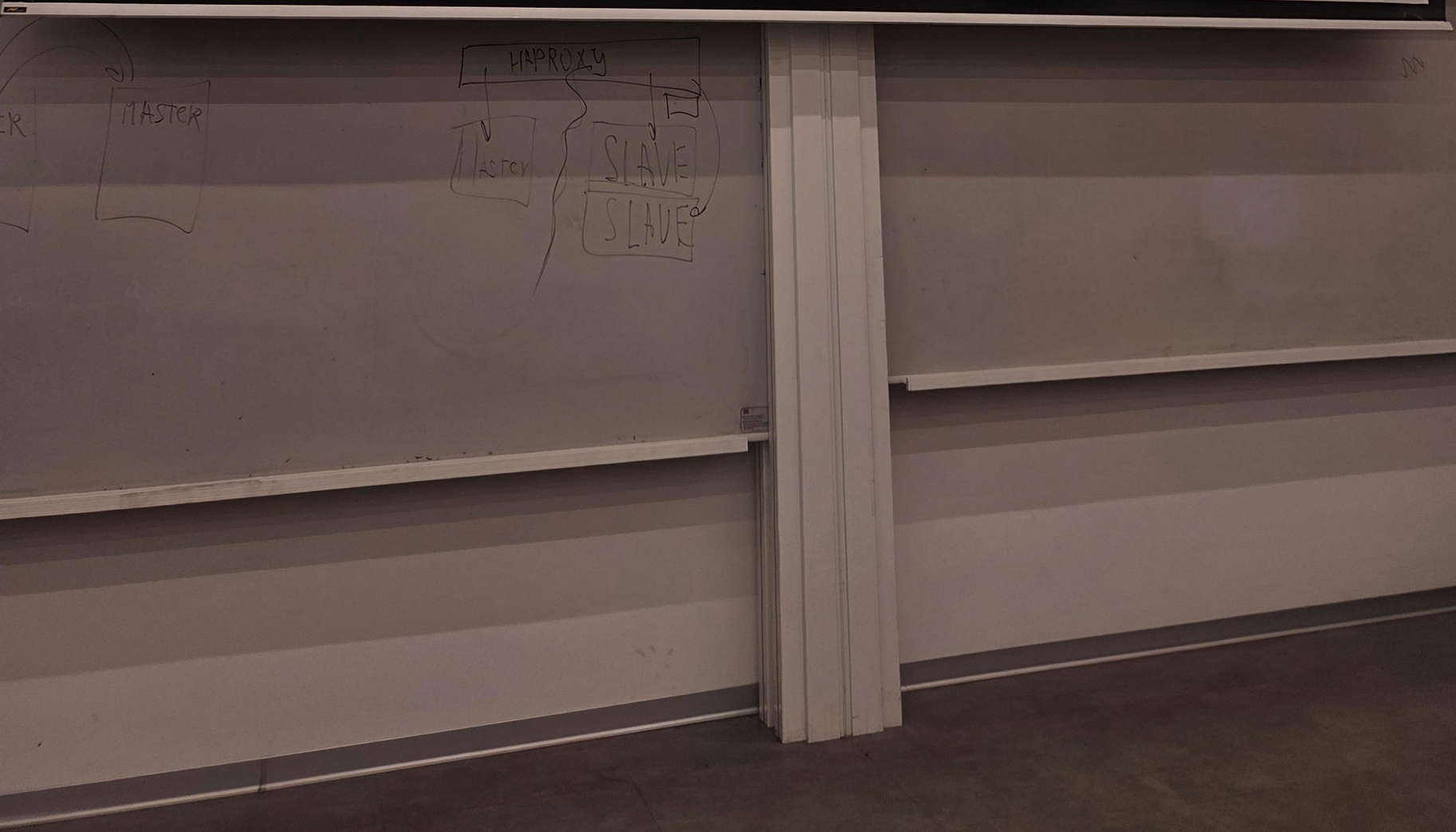


- Consistency — информация на разных узлах согласована
- Availability — система отвечает на запросы в любой момент времени
- Partition tolerance — связи между узлами могут обрываться



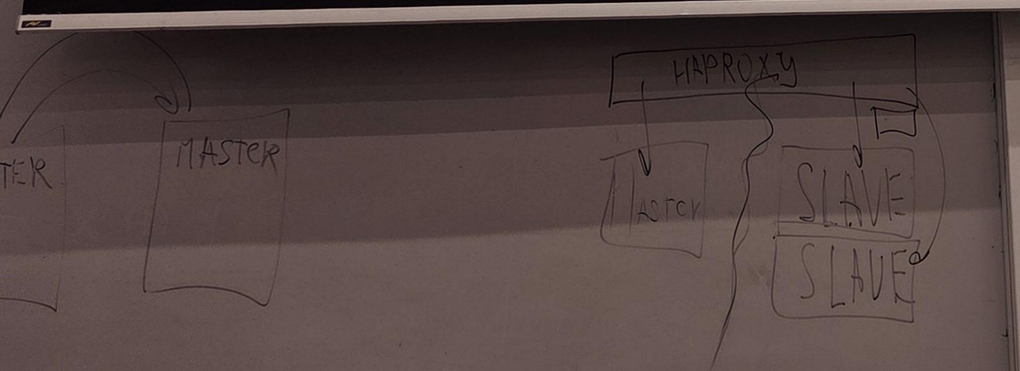
CP

- При разрыве связи между узлами система блокирует запись на часть узлов, чтобы гарантировать, что данные останутся согласованными.
- Примеры: MongoDB, Redis, ZooKeeper



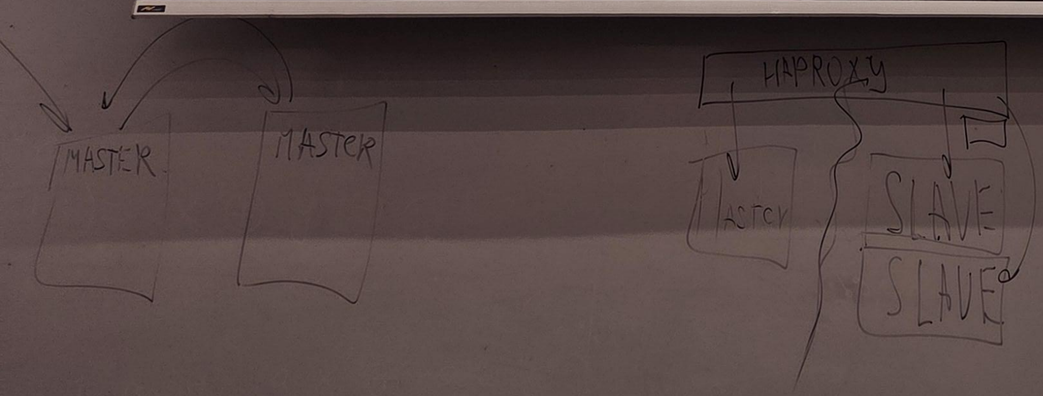
AP

- При разрыве связи все узлы остаются доступными для чтения и записи. Это может привести к тому, что разные узлы будут иметь разные версии данных
- Примеры: Cassandra, Scylla.



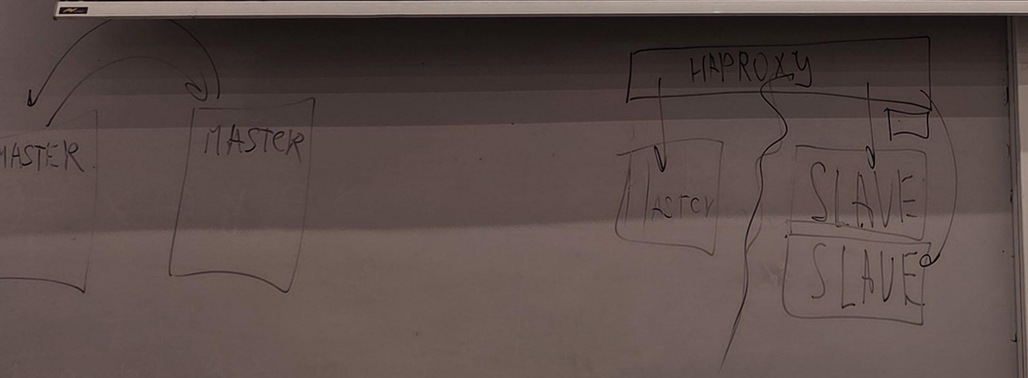
CA

- Такая система может существовать только при отсутствии сетевых сбоев.



Идея нового сервиса — «Позвони, напомню!»

- Система хранения фактов с разрешенными действиями:
  - Запрос на запись факта;
  - Запрос на изменение факта;
  - Запрос на чтение факта.



- Клиент: — Привет, не могли бы вы запомнить День Рождения моего соседа?
- Вы: — Конечно, какое число?
- Клиент: — Второго января.
- Вы: — *(Записываете дату в записную книжку)* Всё, записали. Звоните нам в любой момент, когда вам что-то понадобится.
- Клиент: — Кстати, мне тут пришла идея, что ему стоило бы подарить радиоуправляемый вертолёт, он обожает такие штуки!
- Вы: — Отличная идея для подарка, мы обязательно вам её напомним.
- Клиент: — Спасибо!
- Вы: — Всегда пожалуйста, с вашего счёта снято 3 рубля.

Прошло несколько месяцев...

- Клиент: — Привет, кажется я что-то забыл...
- Вы: — Добрый день, конечно, пару секунд... *(Ищем страничку клиента в записной книжке)*
- Вы: — Спасибо, что подождали. Через неделю день рождения вашего соседа, не забудьте его поздравить и подарить радиоуправляемую модель вертолёта.
- Клиент: — Точно, спасибо огромное! И подарок супер, как вы догадались?
- Вы: Всегда рады помочь, с вашего счёта снято 3 рубля.

рузки

Ура, дело пошло! Ваша идея настолько проста, насколько эффективна. Сервис выстреливает, вы получаете сотни заказов каждый день.
Всё бы хорошо, но появилась проблема: всё больше и больше клиентов ждёт своей очереди поговорить с вами. Некоторые не выдерживают ожидания и просто бросают трубку. Более того: когда вы заболели и не могли работать, пропал целый день бизнеса и вся выручка. И это ещё не говоря о неудовлетворённых клиентах, потерявших возможность получить информацию.
Решено — пора расширяться! Возьмём супругу в помощь.
Итак, план простой:

- Вручаем вам и жене по дополнительному телефону.
- Клиенты продолжают звонить по тому же номеру: нет необходимости запоминать несколько номеров.
- АТС перенаправляет звонки клиентов тому, кто в данный момент свободен.

Вы очень взволнованы этой идеей, ведь:

1. Можно обслуживать в два раза больше клиентов.
2. Если даже кто-то заболеет и не сможет работать, сервис не останавливается и продолжает функционировать.

Через два дня после внедрения новой системы вы получаете звонок от вашего постоянного клиента, Ивана Андреевича:

- Иван: — Добрый день.
- Вы: — Здравствуйте, «Позвони, напомню!», чем мы можем вам помочь?
- Иван: — Напомните, пожалуйста, я ничего там не забыл?
- Вы: — Секунду... *(смотрите в записную книжку, но на страничке Ивана Андреевича нет ничего того, о чём стоило бы напомнить)*
- Вы: — Нет, всё отлично, вы ни о чём не забыли, Иван Андреевич!
- Иван: — Замечательно, спасибо большое.

Через день Иван Андреевич снова звонит вам:

- Иван: — Вы меня сильно подвели, у вас ужасный сервис. У меня была запланирована командировка в Нью-Йорк по важным делам, а я пропустил самолёт. И главное, я просил вас напомнить, но нет, вы наврали. Я очень зол. *(гудки)*
- Вы: — Но как...

Часть 4
Решаем проблему согласованности

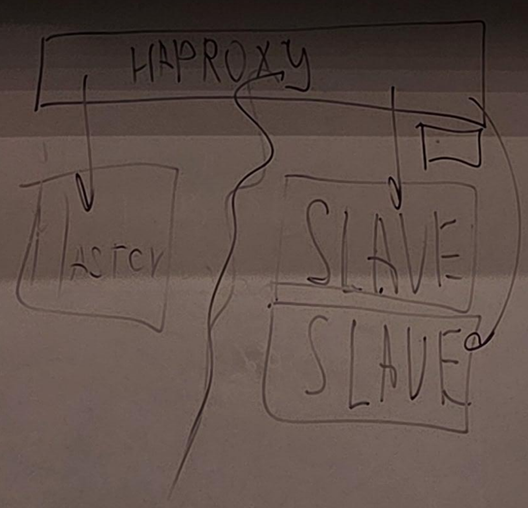
Часть 4: Решаем проблему согласованности

Ваши конкуренты могли бы спокойно игнорировать плохое обслуживание, но вы, конечно, заботитесь о своих клиентах и репутации. Пока ваша жена спала, вам не давала покоя проблема, вы всю ночь думали и пытались найти выход и ситуации. Бинго! Когда жена проснулась, вы в тот же момент ей говорите о новых планах:

- Когда кому-либо из вас звонит клиент и хочет что-нибудь запомнить, до того, как сказать «Спасибо, мы всё записали» вы звоните другому человеку и сообщаете о изменениях.
- Таким образом, у вас обоих записаны все последние обновления.
- Когда клиент звонит для того, что бы напомнить себе что-нибудь, вам не надо звонить супруге — у вас всегда при себе вся правильная и актуальная информация.

Вы замечаете единственную проблему — вы не можете работать параллельно. Каждый раз, когда вам звонит коллега для синхронизации, вы не можете отвечать на звонки клиентов — вы будете заняты. Но это не так уж и страшно, ведь большинство звонков поступает с просьбами что-то напомнить (поиск), а не что-то запомнить (обновление).
Главное — ответить клиенту правильно любой ценой.
«Отлично», говорит вам супруга, «Однако есть ещё одна брешь, о которой ты не подумал. Что, если в один прекрасный день кто-то не сможет выйти на работу? Тогда у нас не получится принять ни единой просьбы запомнить, ведь другой человек не сможет записать изменения в свою записную книжку. Это, братец, у нас получается проблема доступности, так как, к примеру, если запрос на обновление придёт ко мне, я не смогу завершить звонок клиента: даже если я записала изменения в свою записную книжку, у меня не выйдет записать их в твою. Таким образом, я не смогу попрощаться с клиентом!»

Вы замечаете единственную проблему — вы не можете работать параллельно. Каждый раз, когда вам звонит коллега для синхронизации, вы не можете отвечать на звонки клиентов — вы будете заняты. Но это не так уж и страшно, ведь большинство звонков поступает с просьбами что-то напомнить (поиск), а не что-то запомнить (обновление).
Главное — ответить клиенту правильно любой ценой.
«Отлично», говорит вам супруга, «Однако есть ещё одна брешь, о которой ты не подумал. Что, если в один прекрасный день кто-то не сможет выйти на работу? Тогда у нас не получится принять ни единой просьбы запомнить, ведь другой человек не сможет записать изменения в свою записную книжку. Это, братец, у нас получается проблема доступности, так как, к примеру, если запрос на обновление придёт ко мне, я не смогу завершить звонок клиента: даже если я записала изменения в свою записную книжку, у меня не выйдет записать их в твою. Таким образом, я не смогу попрощаться с клиентом!»



лему согласованности

Вы уже осознали, почему распределённые системы не такие простые, как вы думали. Сложно ли придумать решение, которое было бы одновременно «доступно и согласовано»? Кто-нибудь может и сдался, но не вы! Ваши конкуренты и не мечтали о том, что вы придумали. И вновь вы нетерпеливо будите свою жену...

«Смотри, это то, что нам нужно для доступности и согласованности.» План почти такой же, как в прошлый раз, но с важными изменениями:

- Когда кому-либо из вас звонит клиент и хочет что-нибудь запомнить, до того, как сказать «Спасибо, мы всё записали» вы проверяете, доступен ли коллега и, если доступен, звоните другому человеку и сообщаете о изменениях.
- Если коллега не доступен, то вы пишете письмо на электронную почту с информацией об новых обновлениях.
- Первым делом после отсутствия на работе вы или супруга проверяете почту и записываете изменения перед тем, как принимать звонки.

Гениально! Вы не можете найти ни одного недостатка в получившемся решении. Теперь «Позвони, напомню» одновременно доступный и согласованный сервис.

ему согласованности

Часть 6: Супруги иногда ссорятся

Кажется, всё отлично уже который день. Ваша система согласована. Всё работает, даже если кто-то из вас не может выйти на работу. Но что выйдет, если вы оба вышли на работу, но один из вас не может обновить информацию другого? Помните, как вы будили свою жену с очередным Гениальнейшим Бредом? Что, если ваша жена решится принимать звонки, но будет слишком обижена на вас и решит не разговаривать с вами весь день? Весь ваш бизнес опять превратится в тыкву! Ваша идея до сих пор хороша за её согласованность и доступность, но очень чувствительна к разделению коммуникаций! Вы конечно, можете не принимать ни одного звонка, пока вы в ссоре, но тогда ваша система будет недоступна всё это время...

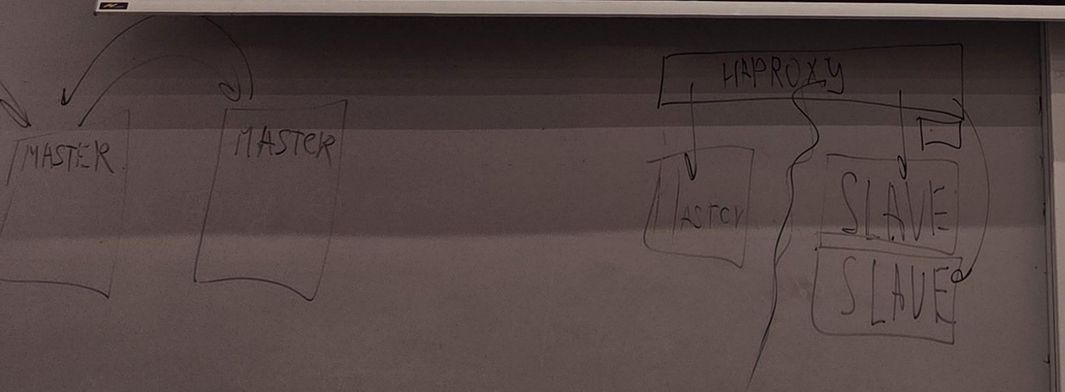
Часть 7: Выводы

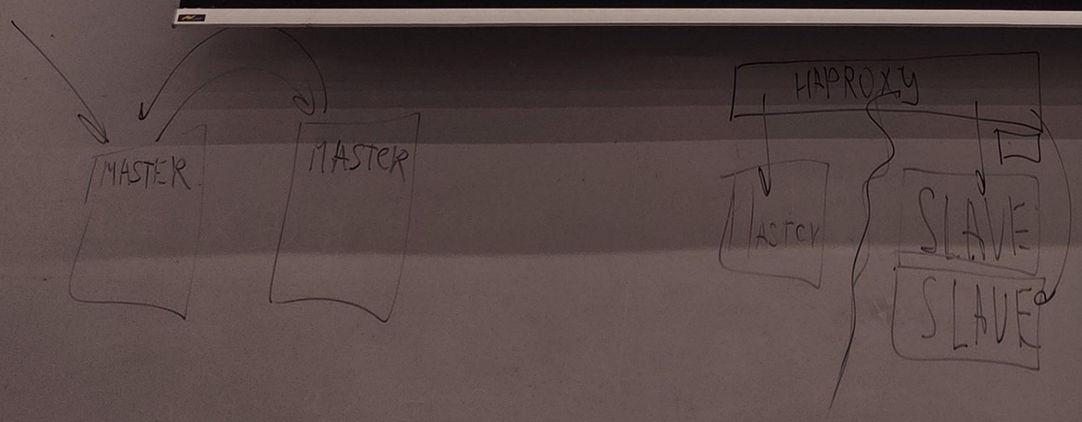
Итак, давайте теперь взглянем на CAP-теорему. Утверждается, что при разработке распределённой системы вы не можете достичь одновременно трёх свойств: доступность, согласованность и терпимость к разделению сети. Вы можете выбрать только два из:

- Согласованность (Consistency) — Ваши клиенты единожды обновив информацию всегда могут получить самые актуальные данные при их последующем запросе. И не важно, насколько быстро они вам перезвонят.
- Доступность (Availability) — «Позвони, напомню» всегда доступна для звонков в тот момент, когда хотя бы один из сотрудников вышел на работу.
- Терпимость к разделению сети. (Partition Tolerance) «Позвони, напомню» всегда работает корректно, даже если вы потеряли связь с вашей женой.

PACELC

- PACELC — расширяет CAP, описывая поведение системы не только во время сетевых сбоев, но и в нормальных условиях работы.



PACELC

- Если произошел P, то выбираем между A и C
- (E)lse выбираем между: L и C