



ОНЛАЙН-ОБРАЗОВАНИЕ

Онлайн-образование



Меня хорошо видно && слышно?

Ставьте  , если все хорошо
Напишите в чат, если есть проблемы

Проверить, идет ли запись!





Введение в распределенные системы



Тюменцев Евгений

Преподаватель



Тюменцев Евгений

- 9 лет руковожу компаний по разработке ПО
- в прошлом занимался разработкой многопоточных кросс-платформенных приложений на C++, серверных приложений на C#
- 20 лет преподаю ООП, паттерны, C++, C#, Kotlin, многопоточные и распределенные приложения
- руководитель курса Архитектура и шаблоны проектирования

Правила вебинара



Активно участвуем



Задаем вопрос в чат или голосом



Off-topic обсуждаем в Slack #канал группы или #general



Вопросы вижу в чате, могу ответить не сразу

Маршрут вебинара

Распределенные системы



CAP и PACELC теоремы



ACID. BASE

Цели вебинара | После занятия вы

1

будете знать фундаментальные ограничения распределенных систем

2

иметь представление как эти ограничения влияют на выбор тех или иных проектных решений

Смысл | Зачем вам это уметь

1

Выбирать наиболее подходящую архитектуру приложений

2

Уметь управлять ожиданиями пользователей

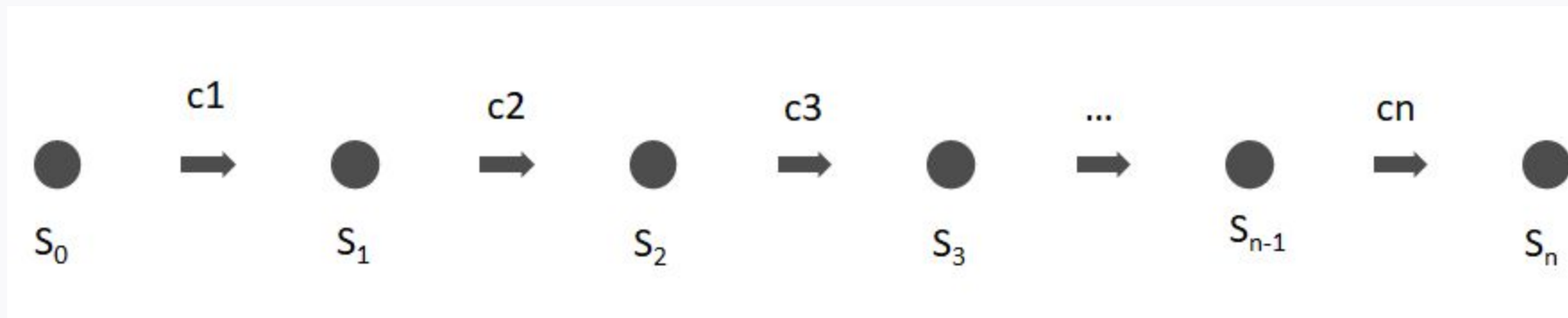
The image features a high-angle, blue-tinted aerial photograph of a dense urban skyline, likely New York City, with numerous skyscrapers. A semi-transparent blue band with a white geometric network pattern of dots and lines runs horizontally across the center of the image. The word "Вычислимость" is written in white, bold, sans-serif font within this band.

Вычислимость

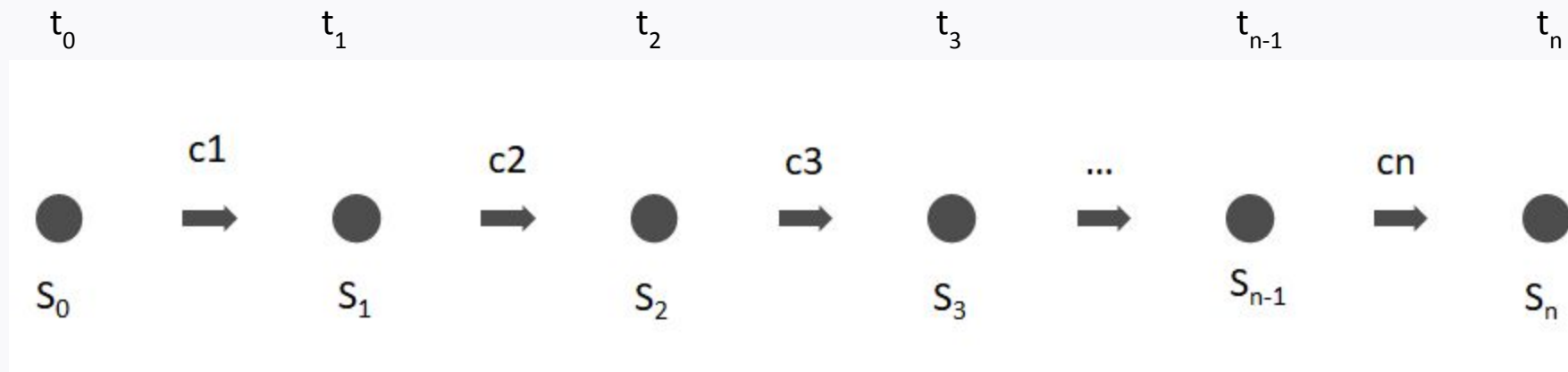
Вопрос

Почему возникает потребность в распределенной системе?

Синхронизация



Ограничение: Глобальное время



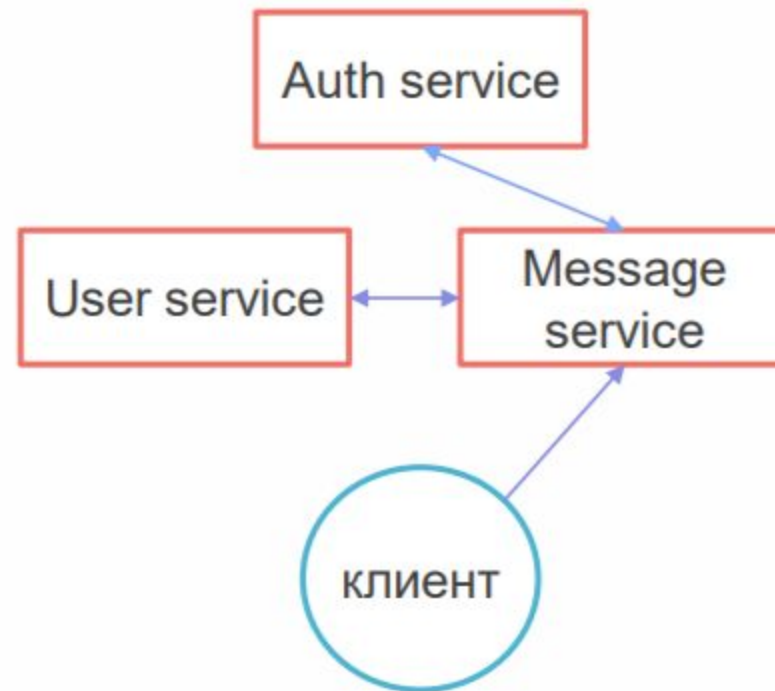
Особенность: Ограниченный недетерминизм



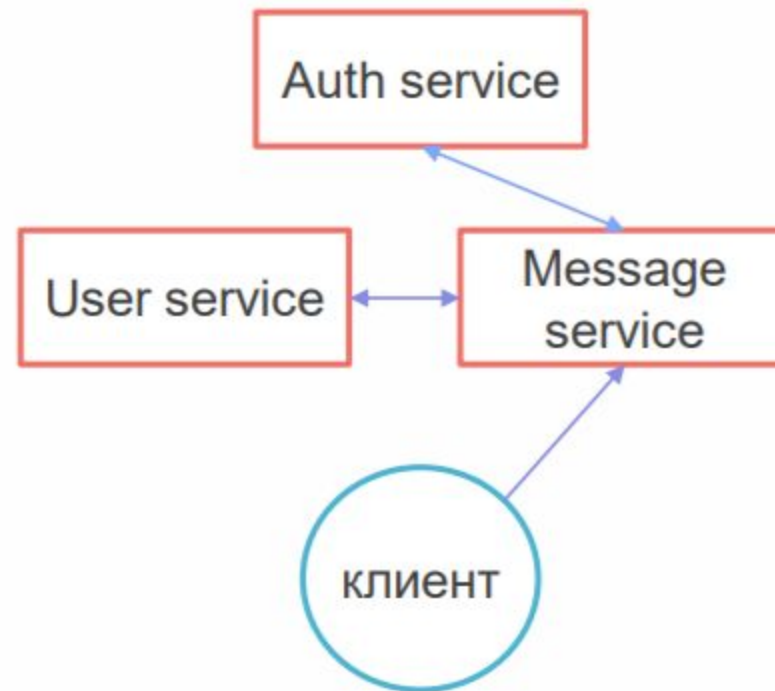


Виды распределенных систем

Stateless-системы

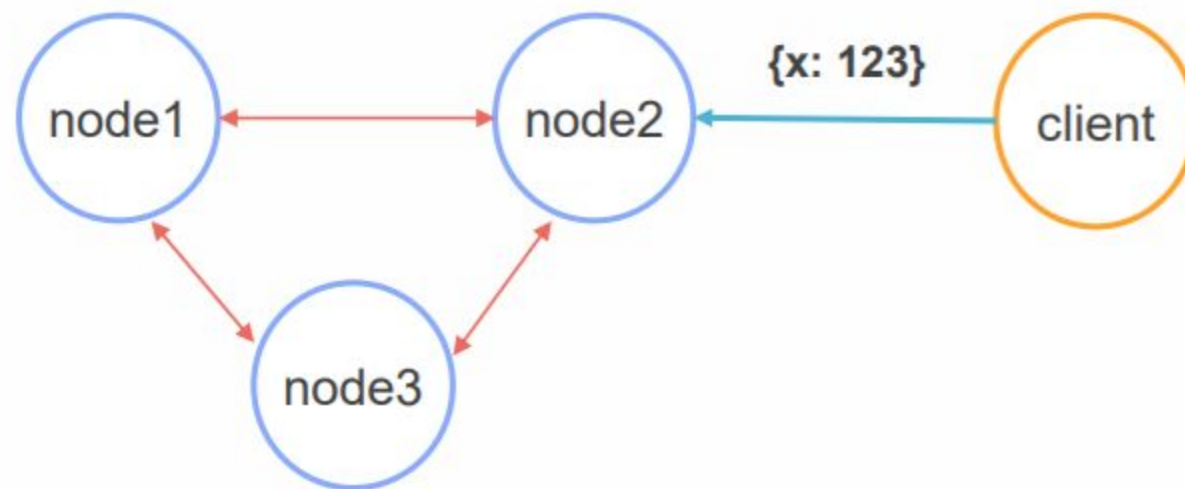


Stateless-системы

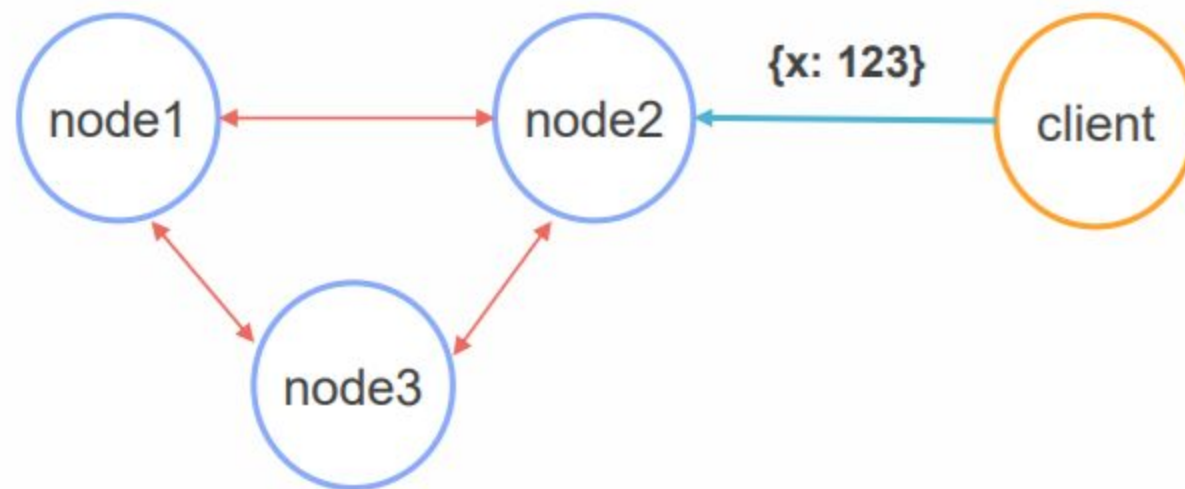


Web-приложение

Statefull-системы



Statefull-системы



СУБД



САР и RASCLS-теоремы

Свойства распределенной системы

- **C** = Consistency

Каждое чтение дает самую последнюю запись

Свойства распределенной системы

- C = Consistency

Каждое чтение дает самую последнюю запись

- A = Availability

Каждый узел (не упавший) всегда успешно выполняет запросы на чтение и запись

Свойства распределенной системы

- C = Consistency

Каждое чтение дает самую последнюю запись

- A = Availability

Каждый узел (не упавший) всегда успешно выполняет запросы на чтение и запись

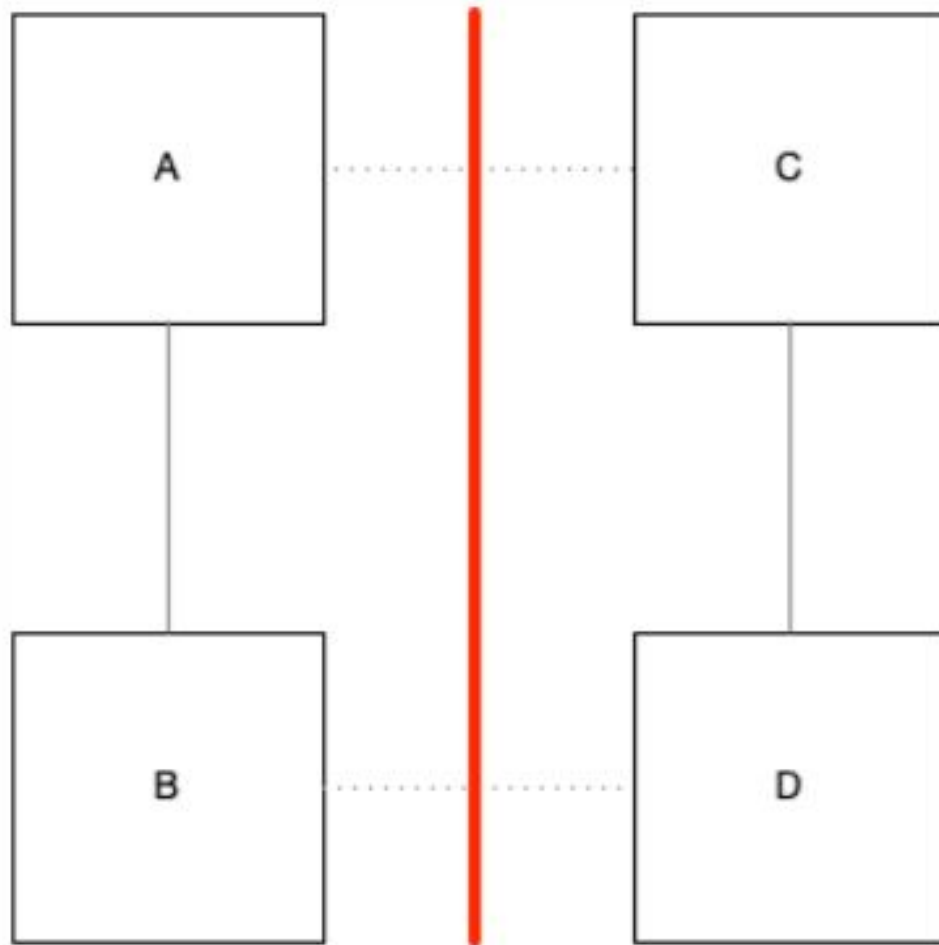
- P = Partition tolerance

Даже, если между узлами нет связи, то они продолжают работать независимо друг от друга

САР-теорема

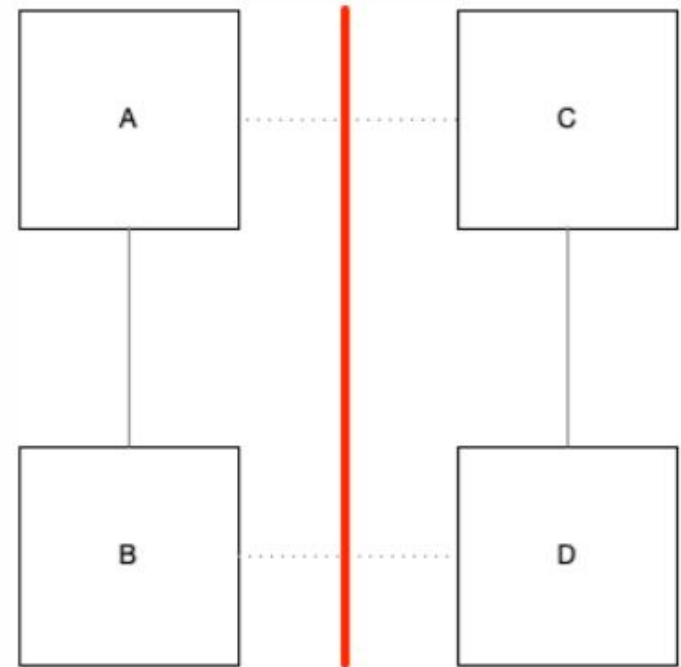
Из трех свойств одновременно можно удовлетворять только двум.

Что делать?



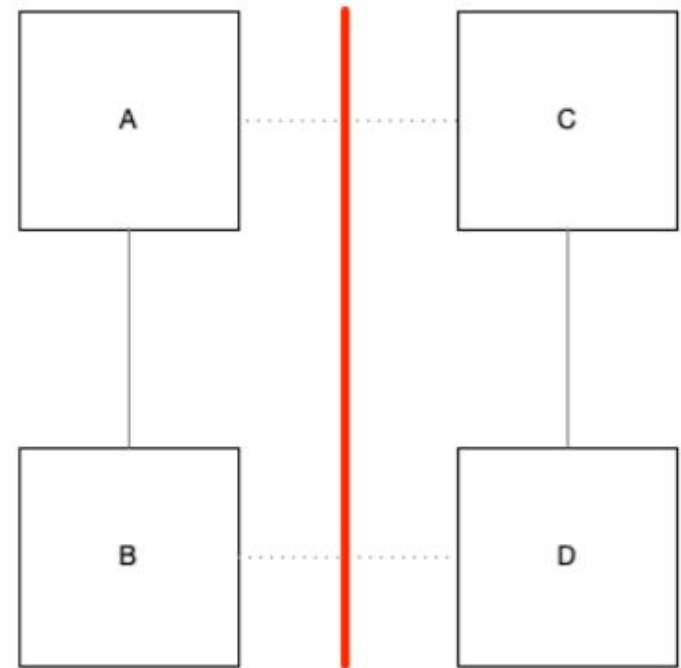
Варианты

- АС - не принимаем запросы



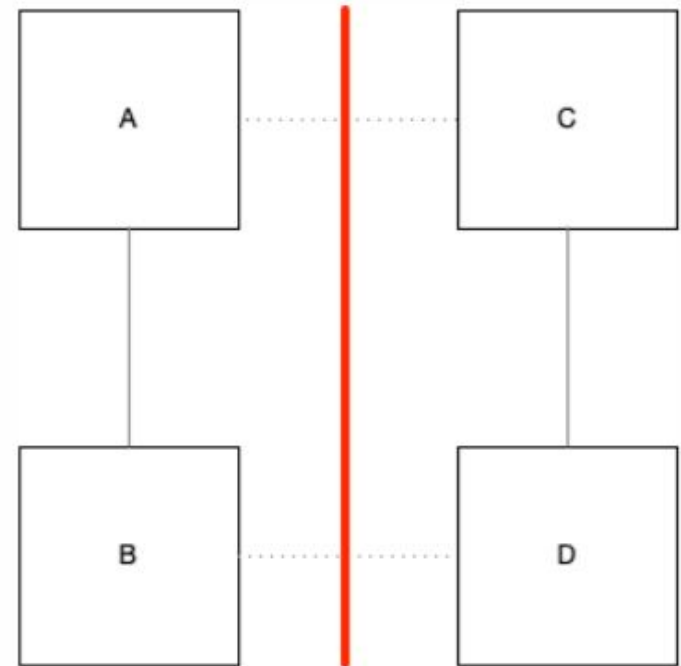
Варианты

- ~~АС - не принимаем запросы~~



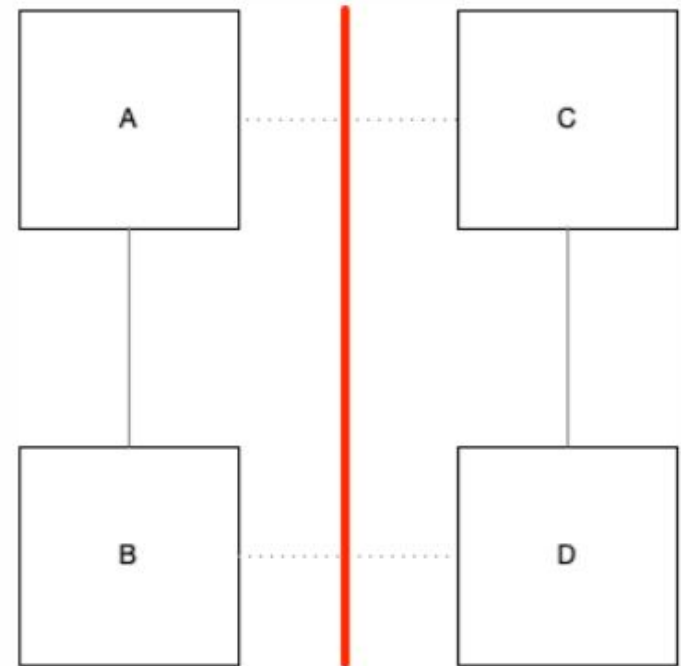
Варианты

- ~~АС - не принимаем запросы~~
- CP - Разрешаем чтение, запрещаем запись



Варианты

- ~~АС - не принимаем запросы~~
- CP - Разрешаем чтение, запрещаем запись
- AP - Разрешаем и чтение, и запись



Проблемы

- далекие от реального мира определения

Проблемы

- далекие от реального мира определения
- выбор только между АР и СР

Проблемы

- далекие от реального мира определения
- выбор только между АР и СР
- множество систем просто Р

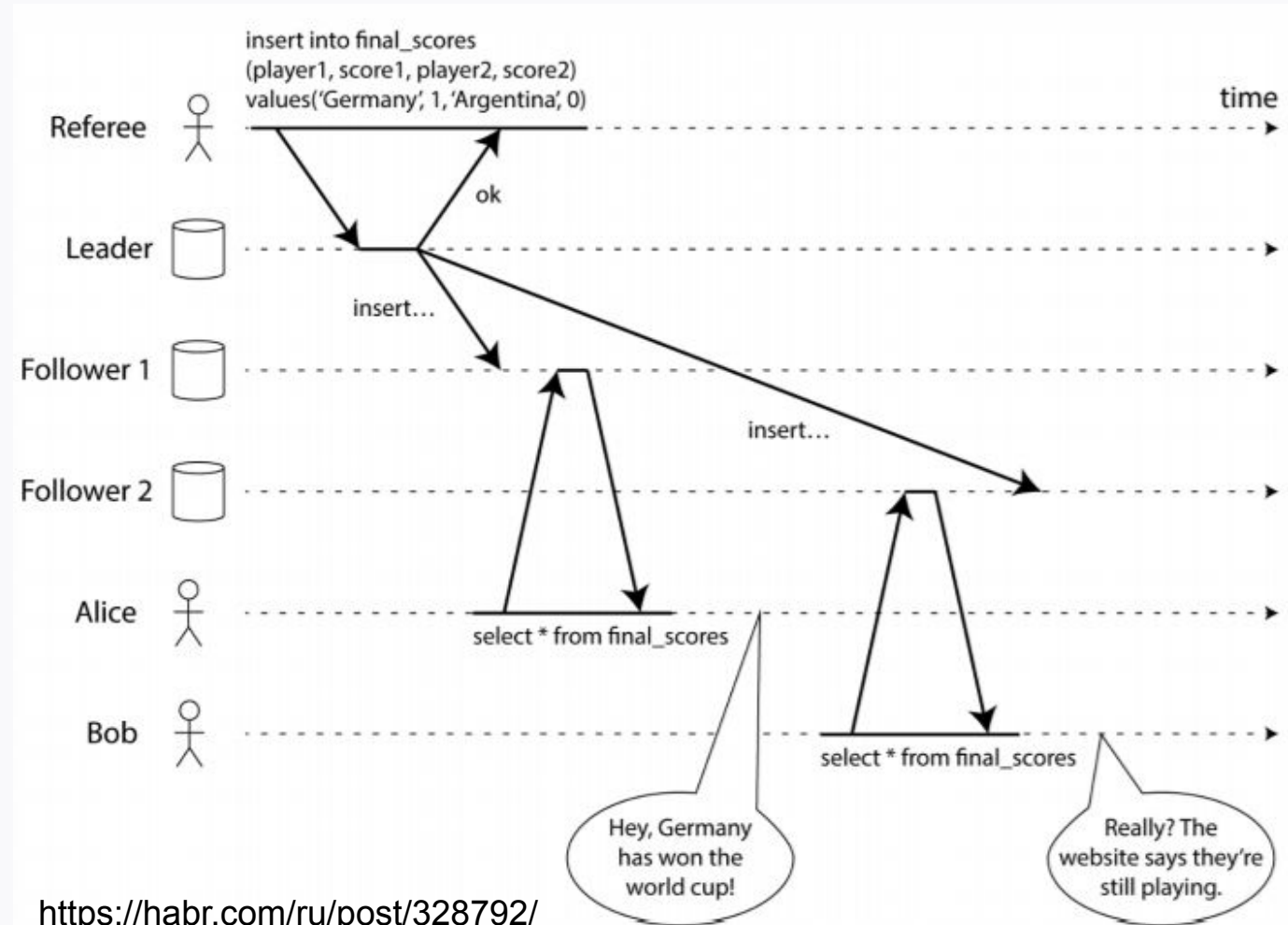
Проблемы

- далекие от реального мира определения
- выбор только между AP и CP
- множество систем просто P
- **чистые AP и CP системы могут быть не тем, чем ожидаешь**

Далекие от реального мира определения

consistency = линейризуемость

- тяжело достичь
- а надо?



Далекие от реального мира определения

availability

- как быть с частичной доступностью

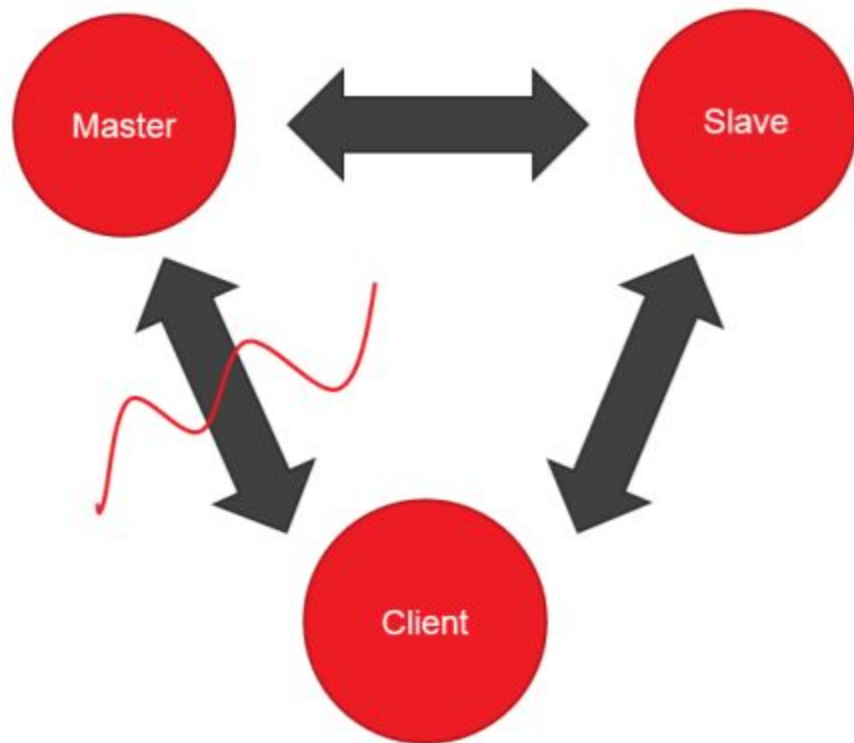
Далекие от реального мира определения

availability

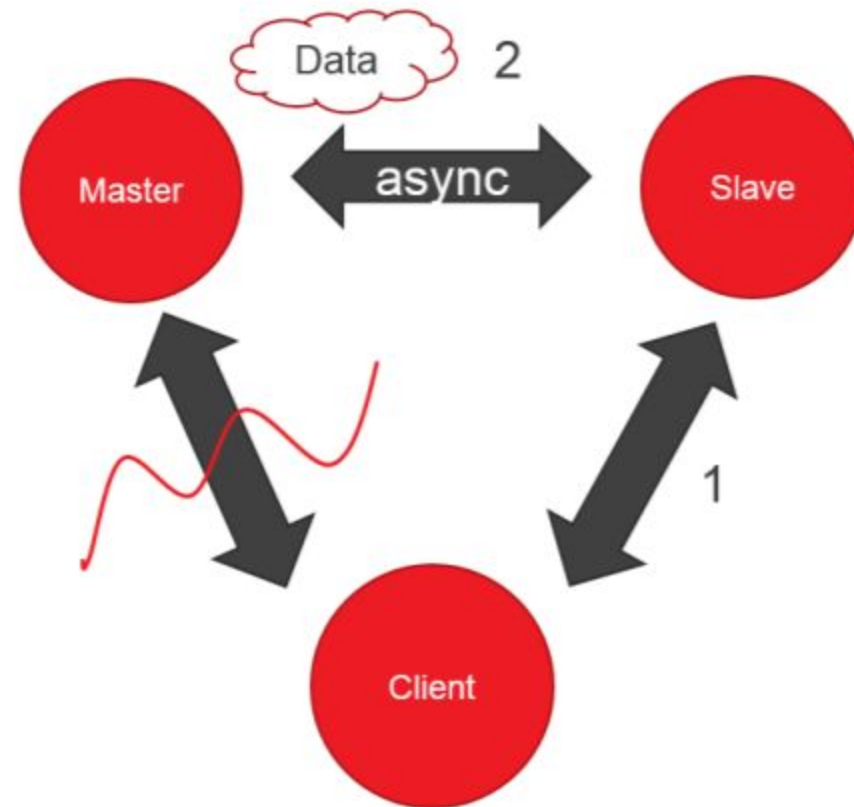
- как быть с частичной доступностью
- **хотелось бы ответа за разумное время**

Многие системы просто Р

Not CAP-available



Not CAP-consistent



Вопрос

Как классифицируется Google Docs согласно CAP-теореме?

Вопрос

Как классифицируется Facebook согласно CAP-теореме?

Вопрос

Как классифицируется CDN согласно CAP-теореме?

Вопрос

Приведите пример просто Р системы?

Вопрос

Приведите пример просто Р системы?

<https://martin.kleppmann.com/2015/05/11/please-stop-calling-databases-cp-or-ap.html>

<https://habr.com/ru/post/322276/>

<https://aphyr.com/tags/jepsen>

Вопрос

А как обычный пользователь интуитивно воспринимает любую систему?

Что делать?

- CAP как идея о том, что любая система имеет фундаментальные ограничения
- нужны уточняющие варианты (PACELC-теорема, принципы ACID/BASE)
- предметная область может подсказать идеи

Пример. Сайт бесплатных объявлений

Пользователи жалуются на то, что размещают объявление, потом идут его посмотреть, а оно то появляется, то исчезает в выдаче.

Анализ: БД Master + 4 Slave, время репликации 40 секунд, этого достаточно, чтобы пользователь успел открыть страницу с выдачей

Что делать?

Пример. Сайт бесплатных объявлений

Время модерации объявления 24 часа, из-за этого приходится публиковать объявления до его модерации. Это создает риски получить большие штрафы от надзорных органов за размещение контента, который нельзя было размещать, а сейчас еще и риски блокировки ресурса.

Анализ: Модераторы не успевают проверять, так как у них есть еще другие более приоритетные операции, в некоторых случаях не берут объявление на модерацию, так как боятся брать ответственность, система оплаты сдельная, поэтому выгодно брать простые случаи и набирать объем.

Что делать?

РАСЕЛС-теорема

IF P \rightarrow (C xor A), ELSE (C xor L).

Latency — это время, за которое клиент получит ответ и которое регулируется каким-либо уровнем consistency. Latency (задержка), в некотором смысле представляет собой степень доступности.

<https://habr.com/ru/post/328792/>

The background of the entire image is an aerial photograph of a dense city skyline, likely New York City, with numerous skyscrapers. A semi-transparent blue overlay covers the entire image. In the center, there is a network of white lines connecting various points, creating a geometric pattern that resembles a molecular structure or a data network. The text "BASE и ACID" is centered within this network.

BASE и ACID

ACID

- A = Atomicity
Все или ничего

ACID

- A = Atomicity
Все или ничего
- C = Consistency

Выполнение инварианта. Поддержание согласованности - задача приложения.

Рекомендация



ACID

- A = Atomicity
Все или ничего
- C = Consistency

Выполнение инварианта. Поддержание согласованности - задача приложения.

- I = Isolation
Параллельные транзакции не должны оказывать влияние друг на друга

ACID

- A = Atomicity
Все или ничего
- C = Consistency

Выполнение инварианта. Поддержание согласованности - задача приложения.

- I = Isolation
Параллельные транзакции не должны оказывать влияние друг на друга
- D = Durability

Данные подтвержденных транзакций сохраняются после восстановления системы после сбоев

BASE

- Basic Availability. Система отвечает на любой запрос, но этот ответ может быть содержать ошибку или несогласованные данные.

BASE

- Basic Availability. 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.**

BASE

- Basic Availability. 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 (**конечная согласованность**). Система, в конечном итоге, станет согласованной. Она будет продолжать принимать данные и не будет проверять каждую транзакцию на согласованность.

Пример. Сайт бесплатных объявлений

Пользователи жалуются, что много сбоев.

Анализ: у нас 7 студентов пилят проект

Что делать?

Пример. Сайт бесплатных объявлений

Архитектура - акторы



Карл Хьюит

1973 г. A Universal Modular ACTOR
Formalism for Artificial Intelligence

Актор – вычислительная сущность

- Отправить конечное число сообщений другим акторам
- Создать конечное число акторов
- Выбрать поведение для приема следующего сообщения

Пример. Сайт бесплатных объявлений

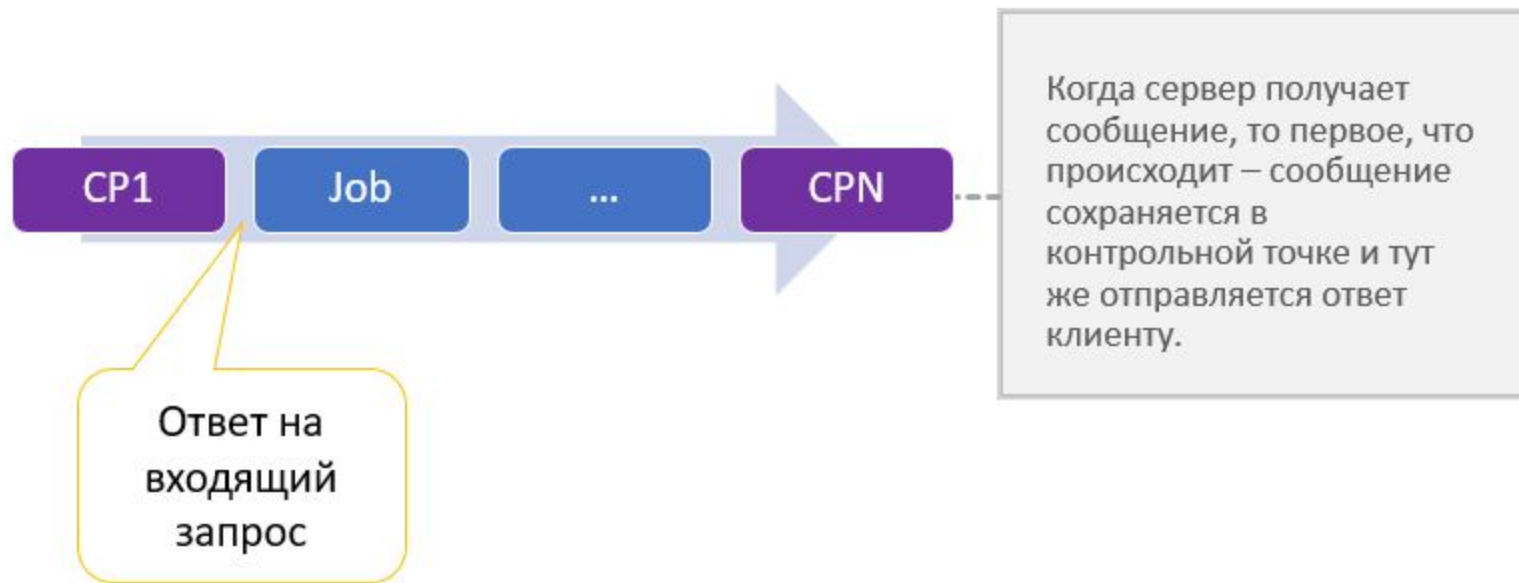
Глобальное время VS акторы



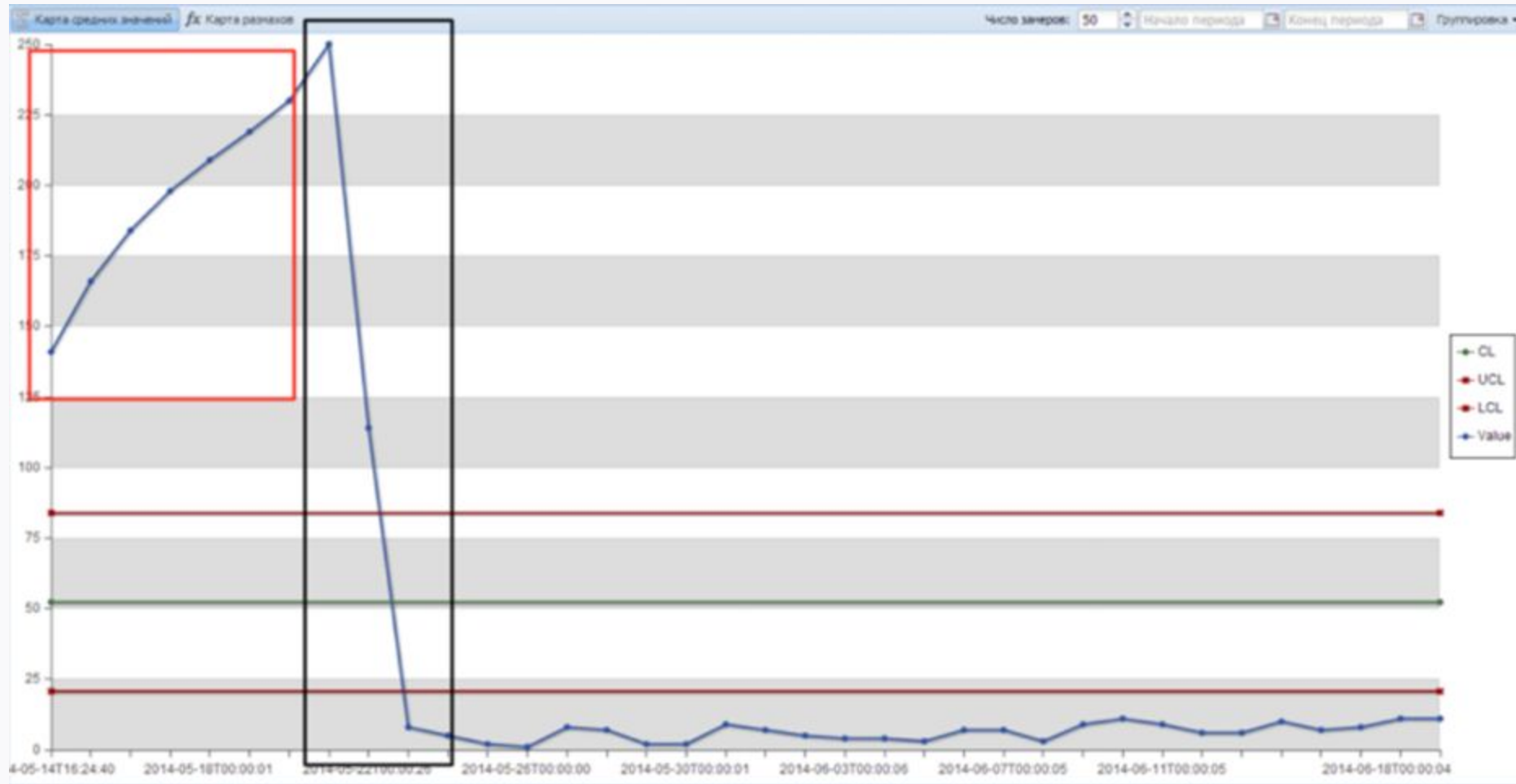
Пример. Сайт бесплатных объявлений

Контрольные точки

Если в момент обработки произойдет ошибка, то пользователь этого никогда не заметит, а разработчики всегда могут повторить данную операцию без участия пользователя, когда исправят ошибку.



Пример. Сайт бесплатных объявлений



Пример. Сайт бесплатных объявлений

Пользователи жалуются, что не проходит оплата услуг

Анализ: есть сбои, а платежная система сообщает только один раз о факте оплаты

Что делать?

Сайт периодически падает из-за стороннего сервиса погоды

Анализ: то бухгалтерия забыла оплатить, то API поменялся

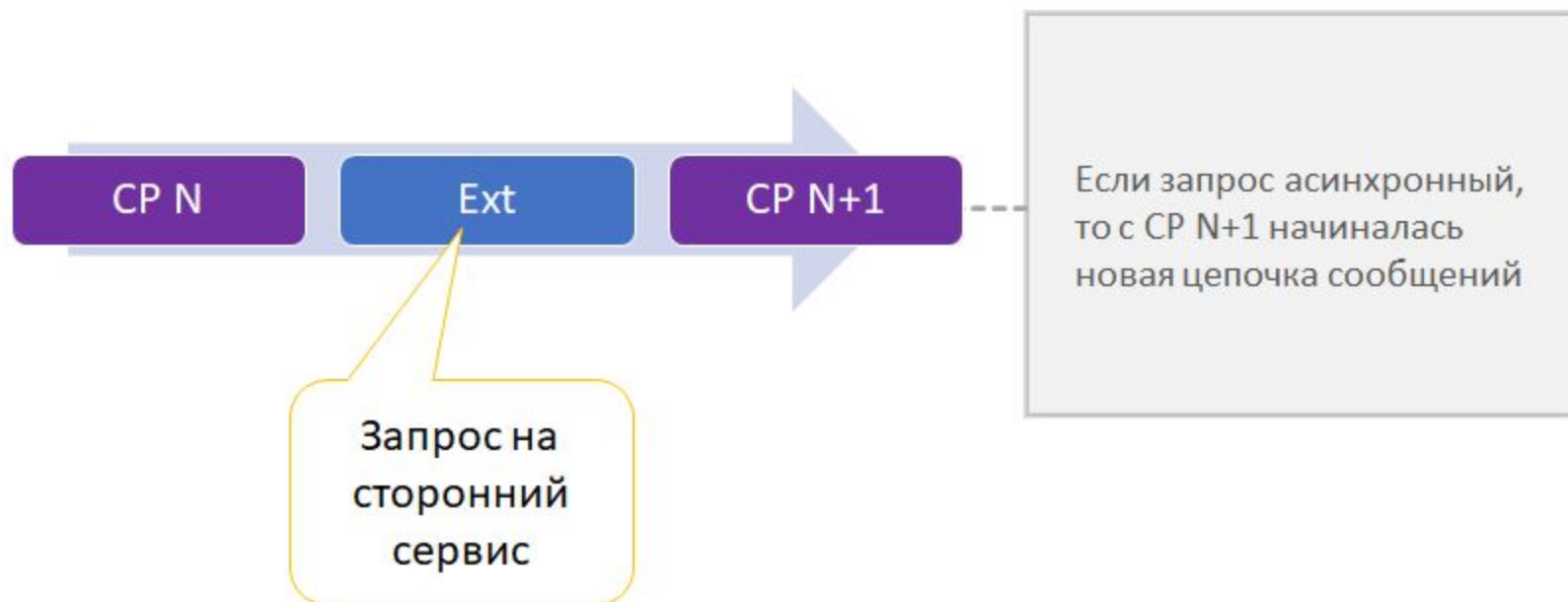
Что делать?

Пример. Сайт бесплатных объявлений

Если результат сервиса потерять нельзя

Перед вызовом сервиса и после ставим контрольную точку.

Перед – чтобы не бомбардировать сторонний сервис, если у нас пойдет что-нибудь не так, а после, чтобы не потерять ответ сервиса, если у нас, что-нибудь сломается после.



Практика. Mentimeter.com

Нужно создать инструмент для интерактивных презентаций.

В начале занятия слушатели подключаются к сайту, на котором идет демонстрация. На телефоне можно будет видеть слайды, которые переключает преподаватель. К некоторым слайдам преподаватель приготовил разные виды вопросов. Слушатели с телефона могут отвечать на эти вопросы. Часть ответов на вопросы тут же отображаются на экране, а часть используются для контроля знаний.


Что делать?



Отметьте 3 пункта, которые вам
запомнились с вебинара



Что вы будете применять в работе
из сегодняшнего вебинара?



Спасибо за внимание!
Приходите на следующие вебинары

Тюменцев Евгений

фото