OTUS

Онлайн образование

otus.ru

REC **Проверить, идет ли запись**

Меня хорошо видно && слышно?

Тема вебинара

Очереди и отложенное выполнение

Преподаватель

Олег Мифле

Backend Developer VK, ВКонтакте

Об опыте:
Более 15 лет в IT, более 10 лет в разработке. Основной стек - php, TS, Golang, PostgreSQL, MySql, redis.

Докладчик на разных конференциях, в т.ч. крупнейших (PhpRussia, HighLoad++), пишу в блог (https://t.me/ask_for_oleg), менторю

Телефон / эл. почта / соц. сети:
https://t.me/mifleo
mifleov@gmail.com

Правила вебинара

Активно
участвуем

Off-topic обсуждаем
в TG #канал группы

Задаем вопрос
в чат или голосом

Вопросы вижу в чате,
могу ответить не сразу

Условные обозначения

Индивидуально

Время,
необходимое на
активность

Пишем в чат

Говорим голосом

Документ

Ответьте себе или
задайте вопрос

Маршрут вебинара

- Знакомство
- Единицы измерения нагрузки, их плюсы и минусы;
- Средства мониторинга, логирование.
- Виды масштабирования
- Рефлексия

Цели вебинара

После занятия вы сможете

1. Проанализировать в каких единицах можно измерять нагрузку
2. Рассмотреть преимущества и недостатки различных подходов к масштабированию
3. Рассмотреть проблемы высоконагруженных проектов

В чем измерить нагрузку?

Вариант №1

Количество запросов в единицу времени:

- rps
- rpm

Какие здесь могут быть проблемы?

Вариант №1

Количество запросов в единицу времени:

- rps
- rpm

Какие здесь могут быть проблемы?
rps бывают разные (выдача прогноза погоды из кэша vs сложная аналитика)

Вариант №2

Количество данных в единицу времени:

- packets per second
- megabytes per second

Какие здесь могут быть проблемы?

Вариант №2

Количество данных в единицу времени:

- packets per second
- megabytes per second

Какие здесь могут быть проблемы?
Видео сервис выдает одинаковые трансляции (один и тот же видео поток) vs Zoom для видеоконференций (стохастические наборы пакетов разные по набору и содержанию, кэш не подойдет)

Вариант №3

Количество одновременных соединений (кол-во пользователей):

- Simultaneous connections
- Concurrency

Какие здесь могут быть проблемы?

Вариант №3

Количество одновременных соединений (кол-во пользователей):

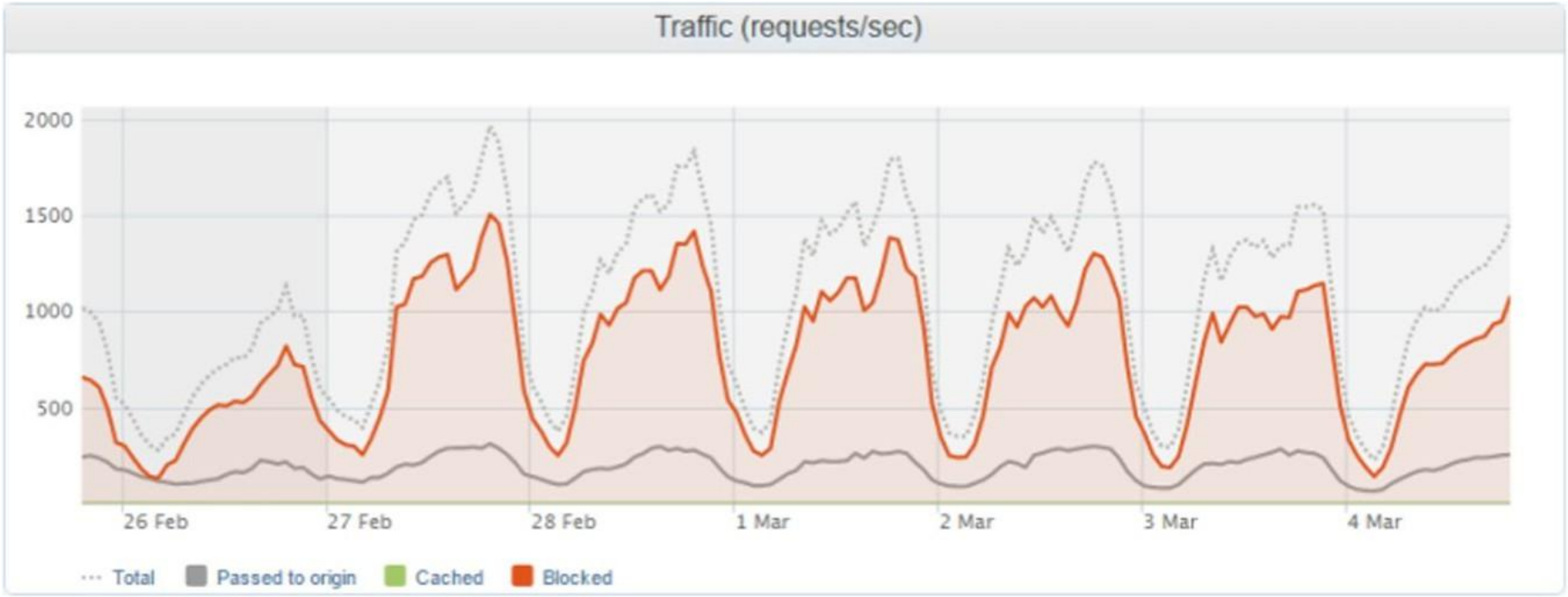
- Simultaneous connections
- Concurrency

Какие здесь могут быть проблемы?
В Zoom в определенный момент приходят новые популярные ведущие (наплыв пользователей)

Выводы

- Чаще всего используется rps
- Немного в прошлом - хороший для измерения простых веб-приложений, в которых есть синхронный запрос-ответ

График нагрузки



Latency

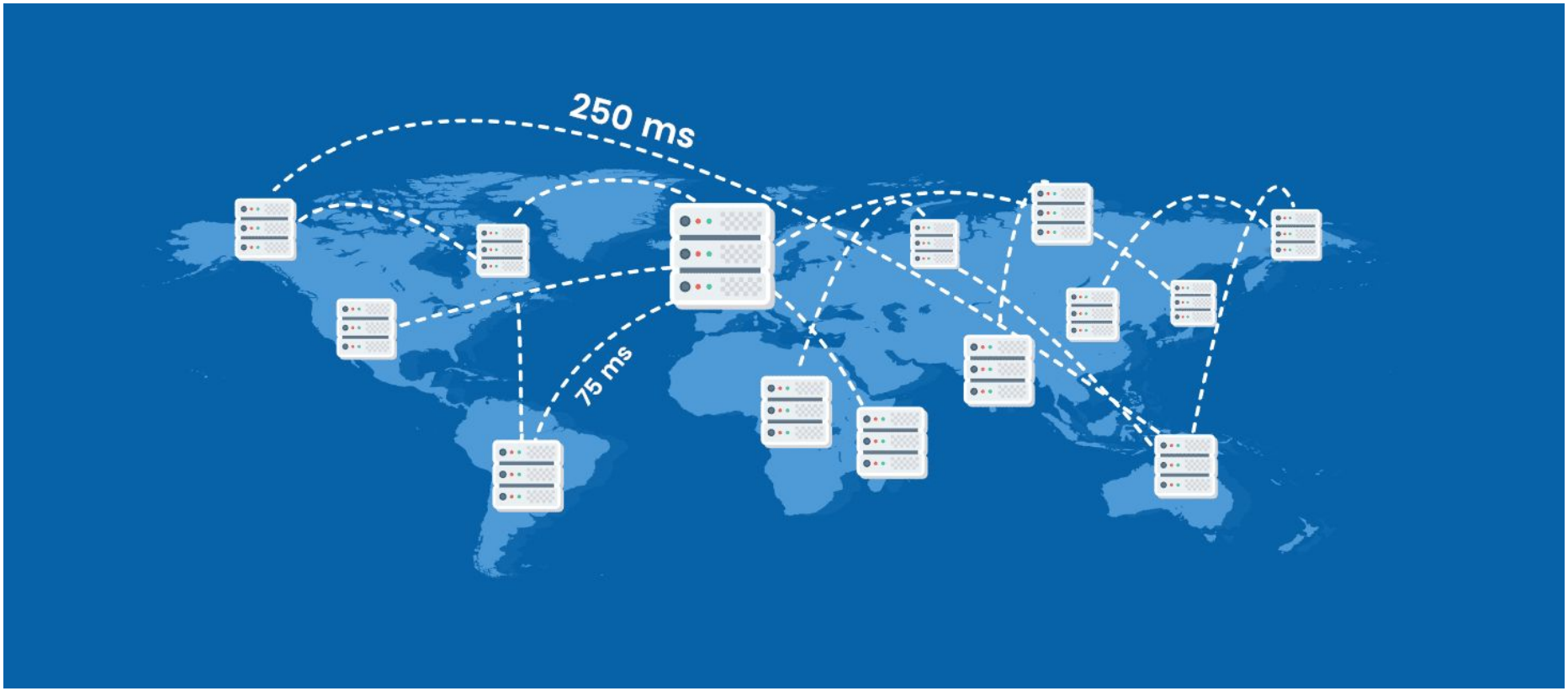
Время, которое проходит от запроса, до получения ответа. Чем оно **ниже**, тем **лучше**.

Что влияет?

Latency

Что влияет?

- География (распространение)



Latency

Что влияет?

- **География** (распространение)
- **Перегрузка сети**: Увеличение трафика в сети приводит к тому, что пакеты отправляются к месту назначения по более длинным маршрутам.
- **Эффективность протокола**: Дополнительные шаги рукопожатия создают задержку.
- **Сетевая инфраструктура**: Когда пакеты задерживаются или теряются, устройства повторно передают их. Это увеличивает задержку.

Throughput (пропускная способность)

- Средний объем данных, который фактически может пройти через сеть (или обработаться системой) в течение определенного времени.
- Он показывает количество пакетов данных, которые успешно прибывают в места назначения, и потерю пакетов данных.

Что влияет?

Throughput

Что влияет?

- Вычислительные мощности
- Потеря пакетов
- Топология сети и вычислительного кластера

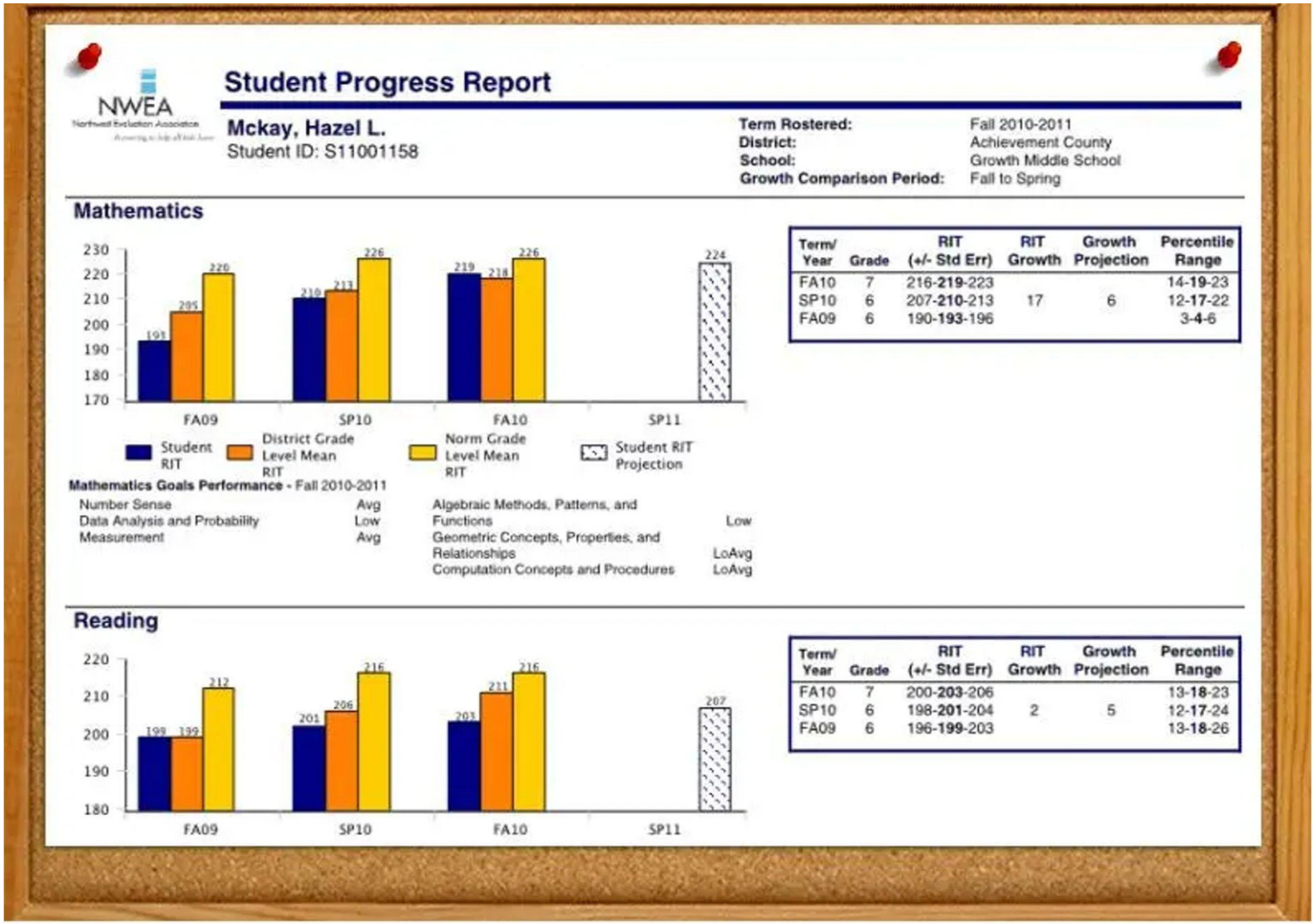
Latency vs throughput

- В реальных проектах приходится выбирать между этими параметрами
- В highload-системах, как правило, предпочтение отдают throughput
- Причины выбора - разные архитектуры.

А что, собственно, измерять?

- Правильно измерять перцентиль.
- Как правило 95 или 99. (а почему?)
- **Перцентиль (процент)** — значение антропометрического признака для сотой доли совокупности измеренных людей.
- Если кривую распределения всей совокупности измеренных людей разделить на 100 равных частей, то получим 99 площадей, в каждой из которых будет свое значение признака и частота ее встречаемости.

Перцентиль



NWEA
Northwest Evaluation Association

Student Progress Report

Mckay, Hazel L.
Student ID: S11001158

Term Rostered: Fall 2010-2011
District: Achievement County
School: Growth Middle School
Growth Comparison Period: Fall to Spring

Mathematics

Term/ Year	Grade	RIT (+/- Std Err)	RIT Growth	Growth Projection	Percentile Range
FA10	7	216-**219**-223			14-**19**-23
SP10	6	207-**210**-213	17	6	12-**17**-22
FA09	6	190-**193**-196			3-**4**-6

Mathematics Goals Performance - Fall 2010-2011

Number Sense	Avg	Algebraic Methods, Patterns, and Functions	Low
Data Analysis and Probability	Low	Geometric Concepts, Properties, and Relationships	LoAvg
Measurement	Avg	Computation Concepts and Procedures	LoAvg

Reading

Term/ Year	Grade	RIT (+/- Std Err)	RIT Growth	Growth Projection	Percentile Range
FA10	7	200-**203**-206			13-**18**-23
SP10	6	198-**201**-204	2	5	12-**17**-24
FA09	6	196-**199**-203			13-**18**-26

https://youtu.be/3ORW40CshBM?t=195

Медиана

Медианой ряда чисел (или медианой числового ряда) называется число, стоящее посередине упорядоченного по возрастанию ряда чисел — в случае, если количество чисел нечётное. Если же количество чисел в ряду чётно, то медианой ряда является полусумма двух стоящих посередине чисел упорядоченного по возрастанию ряда.

Исходим из предположения, что распределение данных внутри медианного интервала равномерное (т.е. 30% ширины интервала – это 30%, и т.д.)

$$Me = x_{Me} + i_{Me}\frac{\frac{\sum f}{2} - S_{Me-1}}{f_{Me}}$$

https://statanaliz.info/statistica/opisanie-dannyx/mediana-v-statistike/

DEMO

https://github.com/rakyll/hey
http://www.itsecgames.com/download.htm

rakyll/hey: HTTP load generator, ApacheBench (ab) replacement

`hey` is a tiny program that sends some load to a web application.

DEMO

https://github.com/rakyll/hey
http://www.itsecgames.com/download.htm

bWAPP, a buggy web application!

bWAPP, or a *buggy web application*, is a free and open source deliberately insecure web application.
It helps security enthusiasts, developers and students to discover and to prevent web vulnerabilities.
bWAPP prepares one to conduct successful penetration testing and ethical hacking projects.

What makes bWAPP so unique? Well, it has over 100 web vulnerabilities!
It covers all major known web bugs, including all risks from the OWASP Top 10 project.

DEMO

https://github.com/rakyll/hey
http://www.itsecgames.com/download.htm

```
wget https://hey-release.s3.us-east-2.amazonaws.com/hey_linux_amd64
-O ~/.local/bin/hey chmod +x !$

docker run --rm -p 8080:80 raesene/bwapp
curl -sI http://localhost/sm_dos_1.php
hey http://bwapp/sm_dos_1.php

https://reqres.in/api/users
```

DEMO

https://github.com/rakyll/hey
http://www.itsecgames.com/download.htm

```
hey http://localhost/sm_dos_1.php
```

```
Response time histogram:
  0.001 [1]     |■
  0.003 [38]    |■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
  0.006 [24]    |■■■■■■■■■■■■■■■■■■■■■■■■
  0.009 [39]    |■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
  0.011 [30]    |■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
  0.014 [36]    |■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
  0.017 [21]    |■■■■■■■■■■■■■■■■■■■■■
  0.019 [5]     |■■■■■
  0.022 [1]     |■
  0.024 [2]     |■■
  0.027 [3]     |■■■

Latency distribution:
  10% in 0.0019 secs
  25% in 0.0043 secs
  50% in 0.0086 secs
  75% in 0.0132 secs
  90% in 0.0156 secs
  95% in 0.0174 secs
  99% in 0.0247 secs
```

Инструменты

ab - Apache HTTP server benchmarking tool

ab is a tool for benchmarking your Apache Hypertext Transfer Protocol (HTTP) server. It is designed to give you an impression of how your current Apache installation performs. This especially shows you how many requests per second your Apache installation is capable of serving.

Инструменты

ab - Apache HTTP server benchmarking tool

- 100 000+ RPS Use phantom load engine written in pure C++ to generate big amount of load from one machine.
- Interactive reports See how your system behaves under load while running the test. Save report when the test is over and email it to your colleagues.
- Monitoring plugin Collect all your system and business metrics by configuring monitoring plugin. The resources of the host you shooting at are monitored by default.
- Multiple load engines Use JMeter to test complex scenarios or BFG (experimental) for exotic protocols. Implement your own module for your favorite tool and use Tank's features like OnlineReport with it.
- Autostops Save your time by stopping your tests automatically using customizable criteria.
- Simple ammo format

Инструменты

wrk2

- giltene/wrk2: A constant throughput, correct latency recording variant of wrk
- a HTTP benchmarking tool based mostly on wrk
- wrk2 is wrk modifed to produce a constant throughput load, and accurate latency details to the high 9s (i.e. can produce accurate 99.9999%'ile when run long enough). In addition to wrk's arguments, wrk2 takes a throughput argument (in total requests per second) via either the --rate or -R parameters (default is 1000)

User-centric подход

Альтернативный взгляд - user-centric

Core Web Vitals: Web Vitals

- Core Web Vitals are the subset of Web Vitals that apply to all web pages, should be measured by all site owners, and will be surfaced across all Google tools.
- Each of the Core Web Vitals represents a distinct facet of the user experience, is measurable in the field, and reflects the real-world experience of a critical user-centric outcome. The metrics that make up Core Web Vitals will evolve over time.
- The current set for 2020 focuses on three aspects of the user experience—loading, interactivity, and visual stability—and includes the following metrics (and their respective thresholds):

https://web.dev/user-centric-performance-metrics/#how-metrics-are-measured

https://web.dev/vitals/#evolving-web-vitals

Инструменты

Core Web Vitals: Web Vitals

- **Largest Contentful Paint (LCP):** measures loading performance. To provide a good user experience, LCP should occur within 2.5 seconds of when the page first starts loading.
- **First Input Delay (FID)**: measures interactivity. To provide a good user experience, pages should have a FID of 100 milliseconds or less.
- **Cumulative Layout Shift (CLS)**: measures visual stability. To provide a good user experience, pages should maintain a CLS of 0.1. or less.

Инструменты

jamstack https://jamstack.org/

Jamstack is an architectural approach that decouples the web experience layer from data and business logic, improving flexibility, scalability, performance, and maintainability.

Jamstack removes the need for business logic to dictate the web experience. It enables a composable architecture for the web where custom logic and 3rd party services are consumed through APIs.

1. https://vitepress.vuejs.org/
2. https://pages.cloudflare.com/
3. https://workers.cloudflare.com/

Инструменты

lighthouse
https://chrome.google.com/webstore/detail/lighthouse/blipmdconlkpinefehnmjammfjpmpbjk?hl=en

Lighthouse is an open-source, automated tool for improving the performance, quality, and correctness of your web apps.

Инструменты

overload (Yandex Load Testing):
https://overload.yandex.net/login/?next=/

yandextank

Инструменты

webpagetest
https://www.webpagetest.org/

Latency vs Throughput

Напишите, в каком случае, что использовать?

1. IoT
2. BigData
3. Mobile
4. Web
5. Gaming
6. Analytics

Latency vs Throughput

Напишите, в каком случае, что использовать?

1. IoT
2. BigData
3. Mobile
4. Web
5. Gaming
6. Analytics

Latency	Throughput
IoT	BigData
Mobile	Web
Gaming	Analytics

Виды масштабирования

Виды масштабирования

- **Вертикальное -** увеличение мощности сервера.
- **Горизонтальное -** использование бОльшего количества серверов.

Вертикальное масштабирование

- **Вертикальное масштабирование** — увеличение производительности каждого компонента системы с целью повышения общей производительности.
- Масштабируемость в этом контексте означает возможность заменять в существующей вычислительной системе компоненты более мощными и быстрыми по мере роста требований и развития технологий.
- Это самый простой способ масштабирования, так как не требует никаких изменений в прикладных программах, работающих на таких системах.

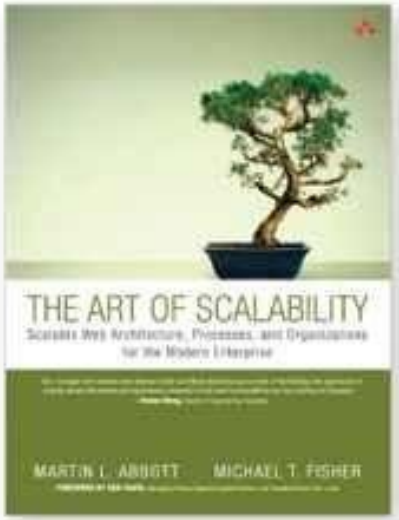
Горизонтальное масштабирование

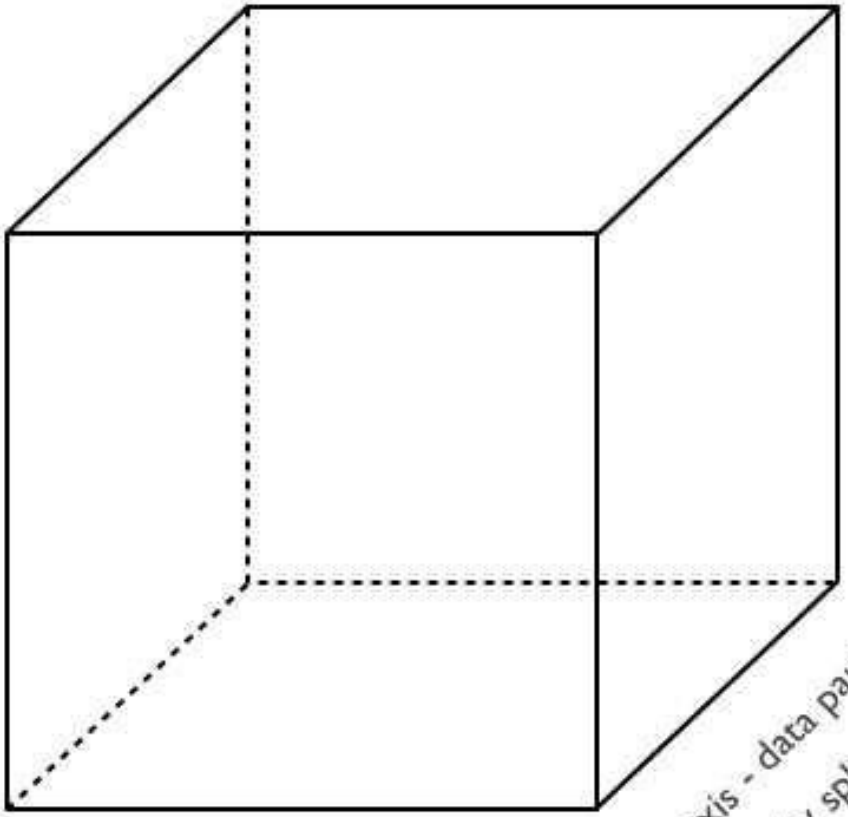
- **Горизонтальное масштабирование (X-axis scaling)** позволяет запускать только множество экземпляров всего приложения, а не той ее части которая требуют большего ресурса, за балансировщиком нагрузки.
- Если есть N копий приложения, то каждая копия обрабатывает 1 / N нагрузки. Это простой и часто используемый подход масштабирования приложение.
- Одним из недостатков такого подхода заключается в том, что, поскольку каждая копия потенциально получает доступ ко всем данным, чтобы быть эффективным, кэши требуется больше памяти.
- Еще одна проблема этого подхода заключается в том, что он не учитывает последующее развития и увеличения сложности приложений.

The Scale Cube

The book, The Art of Scalability, describes a really useful, three dimension scalability model: the scale cube.

3 dimensions to scaling





Y axis - functional decomposition

Scale by splitting different things

Z axis - data partitioning

Scale by splitting similar things

X axis - horizontal duplication

Scale by cloning

https://microservices.io/articles/scalecube.html

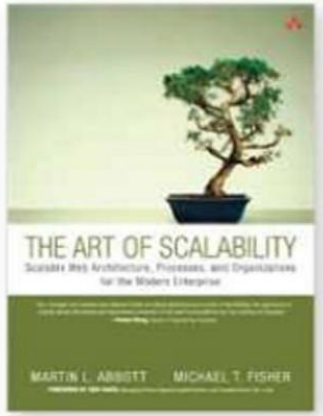
Параметрическое масштабирование

- **Параметрическое масштабирование (Z-axis scaling)** — каждый экземпляр выполняет идентичную копию кода. В этом отношении он похож на (X-axis scaling). Большая разница в том, что каждый сервер отвечает только за своё подмножество данных.
- Некоторый компонент системы отвечает за маршрутизацию каждого запроса на соответствующий экземпляр. Один из критериев маршрутизации является как правило атрибут запроса, другим распространенным критерием маршрутизации критерием является тип клиента. (Z-axis scaling) имеет ряд преимуществ и обычно используются для масштабирования баз данных.
- Каждый экземпляр имеет дело только со своим подмножеством данных. Это улучшает использование кэш-памяти и уменьшает использование трафика ввода/вывода, а также улучшает масштабируемость транзакций, так как запросы, как правило, распределяются между несколькими экземплярами.
- Кроме того, (Z-axis scaling) улучшает изоляцию сбоев, поскольку отказ одного экземпляра оставляет часть данных доступной.

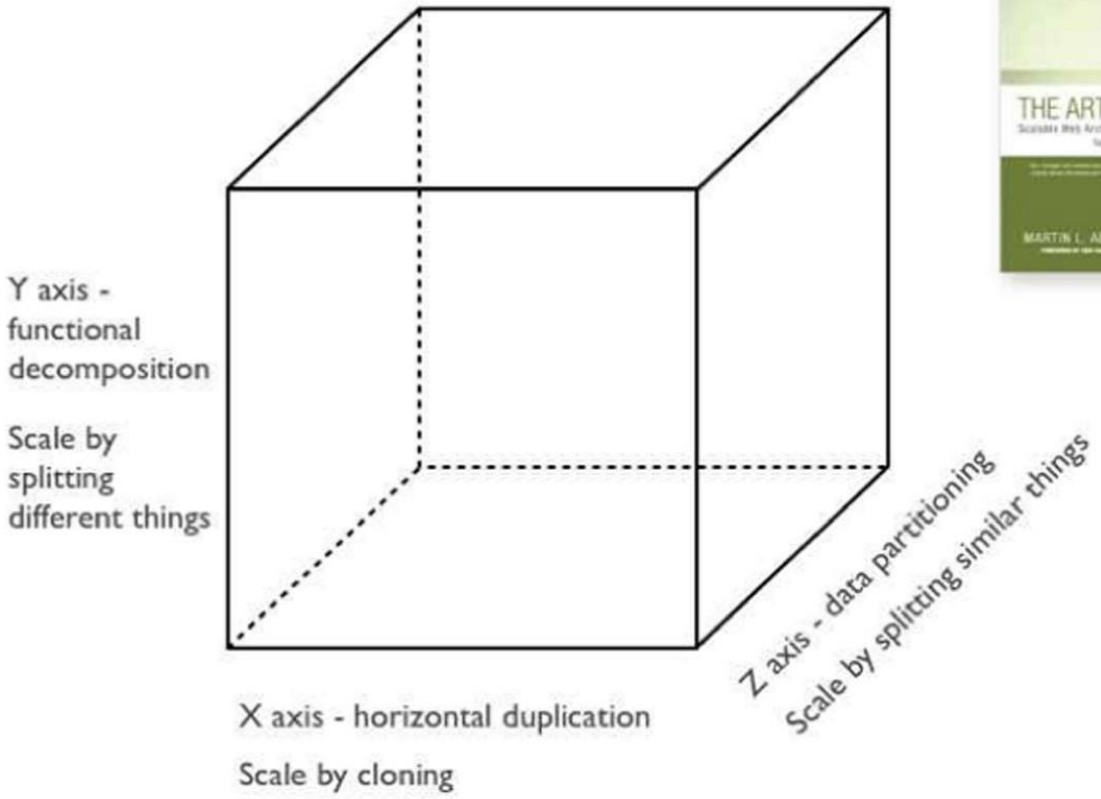
Функциональное масштабирование

- **Функциональное масштабирование (Y-axis scaling)** в отличии от (X-axis and Zaxis), которые состоят из нескольких, работающих идентичных копий приложения, последний тип масштабирования разбивает приложение на несколько разных услуг.
- Каждая служба несет ответственность за одну или несколько функций, взаимодействуя друг с другом.

3 dimensions to scaling



Microservices



Sharding

Mirroring

[Microservices. Как правильно делать и когда применять?]

Sharding

- **Sharding («шардинг», «разбиение»)** — расположение однотипных, но разных данных на разных узлах.
- Те, кто работал с NoSQL-базами, знают, что это такое.
- У вас есть ключ шардирования, по которому вы определяете, что у вас, например, данные на А и Б хранятся на одном узле, на В и Г — на другом узле и т. д.
- Таким образом, используя интеллектуальный выравниватель нагрузки, вы можете ее распределять по вашей системе и добиться более высокой производительности.

Mirroring

- **Mirroring («зеркалирование»)** — горизонтальное дублирование или клонирование всех данных, когда вы ставите рядом совершенно одинаковые хосты.
- Таким образом вы полностью копируете данные.
- Нужно это, прежде всего, чтобы система отвечала на запросы с каким-либо ожидаемым временем отклика.

MSA

- **Microservices (микросервисы)** — вы разбиваете функциональность по бизнес-задачам.
- Каждый сервис будет выполнять определенные задачи.
- Это и есть микросервисный подход, который мы здесь разберем.

3D Масштабирование микросервисов

Optimizing web servers for high throughput and low latency

// By Alexey Ivanov • Sep 06, 2017

> All optimizations that were described in this post are local to a single web server box. Some of them improve scalability and performance. Others are relevant if you want to serve requests with minimal latency or deliver bytes faster to the client. But in our experience a huge chunk of user-visible performance comes from a more high-level optimizations that affect behavior of the Dropbox Edge Network as a whole, like ingress/egress traffic engineering and smarter Internal Load Balancing. These problems are on the edge (pun intended) of knowledge, and the industry has only just started approaching them.

https://dropbox.tech/infrastructure/optimizing-web-servers-for-high-throughput-and-low-latency

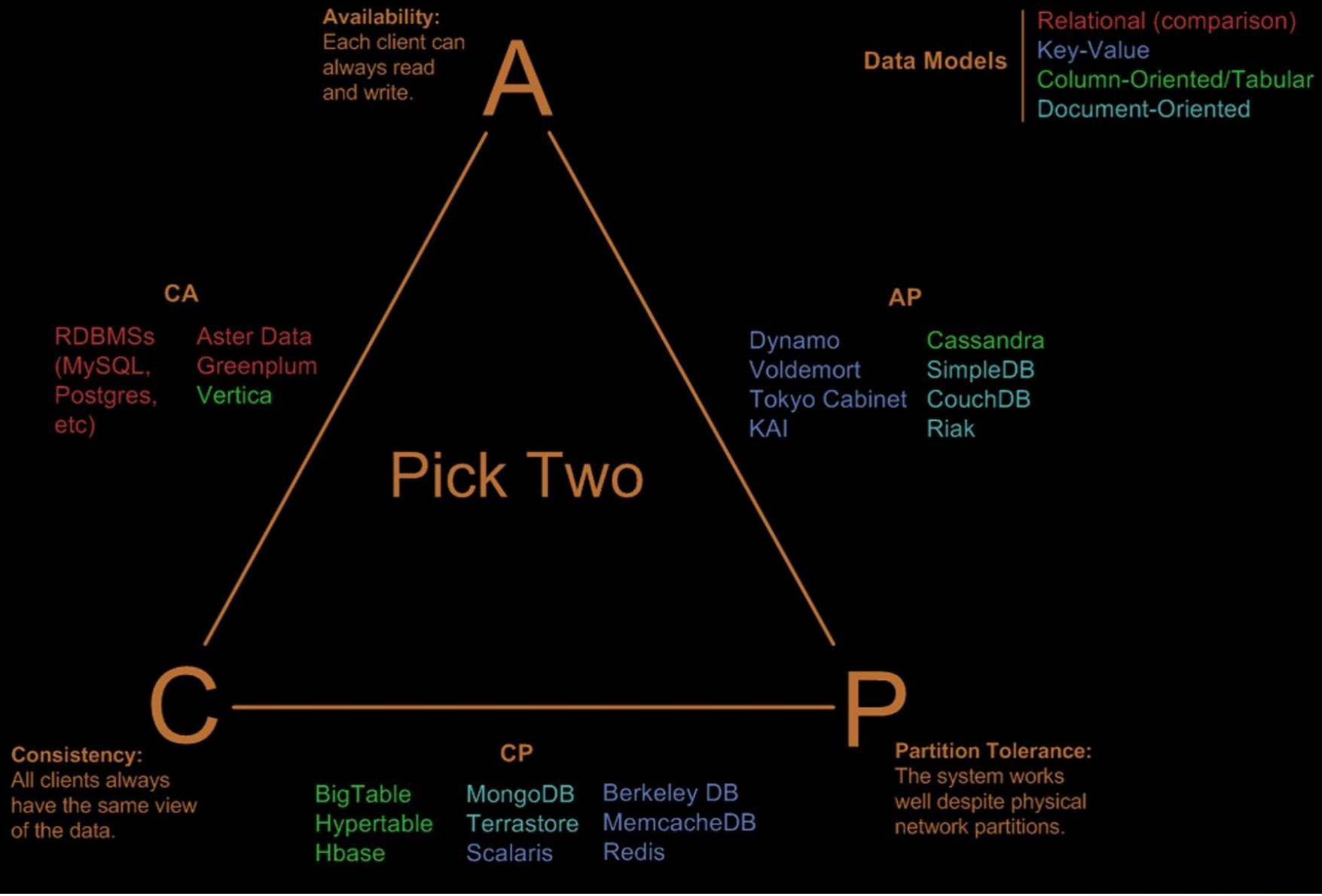
CAP

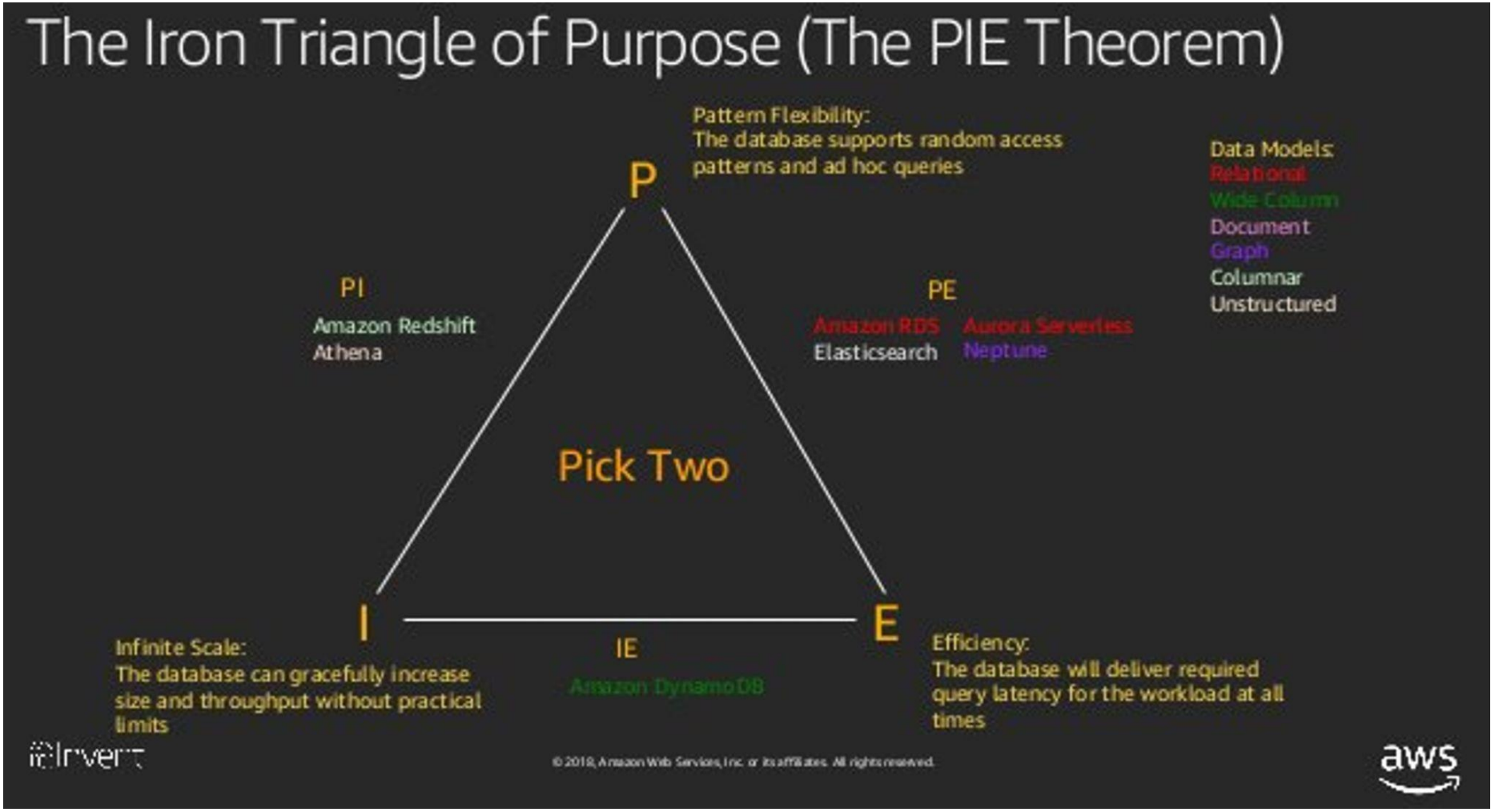
CAP/PIE theorems, BASE-approach

CAP — невозможно создать систему, которая одновременно будет корректно отвечать на каждый принятый запрос, устойчива к разделению на сегменты и будет обеспечивать при этом линеаризуемость операций. В то же время в жизни при обсуждении гарантий консистентности и доступности мы обычно говорим про положение на спектре и SLA.

CAP Theorem - two decades and few clouds later

Visual Guide to NoSQL Systems





https://www.alexdebrie.com/posts/choosing-a-database-with-pie/

CAP Theorem - two decades and few clouds later

Spanner, TrueTime & The CAP Theorem

Spanner, TrueTime & The CAP Theorem - Google Research https://research.google.com › pubs › archive

Spanner, TrueTime & The CAP Theorem

“A Critique of the CAP Theorem”

Availability: The system is available 100% of the time, in the scope of network partitions. An important detail of this definition is that there is no limit to latency. “A Critique of the CAP Theorem”
pdf, September 18, 2015

https://arxiv.org/pdf/1509.05393.pdf

CAP & Spanner & MythBusters

Про Google Spanner говорят, что он опровергает CAP, беря от жизни всё. Однако в реальности у него не полная доступность всегда, а «всего лишь» 99.999%. Технически это CP-система, но на самом деле CA.
Kafka позволяет нам определять гарантии консистентности тонкой настройкой клиентов, а Cassandra и того пуще даёт возможность выбрать независимые гарантии для каждой операции.
К какому классу их отнести? Выбирая систему хранения данных для нашего проекта, мы должны принимать всё это во внимание.
SLA is the new CAP!

Spanner, TrueTime & The CAP Theorem
https://storage.googleapis.com/pub-tools-public-publication-data/pdf/45855.pdf

BASE

S3 поддерживает strong consistency. WAT?
AP vs AC?

https://aws.amazon.com/ru/blogs/aws/amazon-s3-update-strong-read-after-write-consistency/

Интересные ссылки

- https://www.comp.nus.edu.sg/~gilbert/pubs/BrewersConjecture-SigAct.pdf
- https://sites.cs.ucsb.edu/~rich/class/cs293b-cloud/papers/brewer-cap.pdf
- https://arxiv.org/pdf/1509.05393.pdf
- https://storage.googleapis.com/pub-tools-public-publication-data/pdf/45855.pdf

Вопросы?

Ставим “+”,
если вопросы есть

Ставим “–”,
если вопросов нет

Рефлексия

Рефлексия

С какими впечатлениями уходите с вебинара?

Как будете применять на практике
то, что узнали на вебинаре?

О чем мы говорили сегодня?

Подведем итоги

Следующий вебинар

Нагрузочное тестирование

- Ссылка на вебинар будет в ЛК за 15 минут
- Материалы к занятию в ЛК — можно изучать
- Обязательный материал обозначен красной лентой

Заполните, пожалуйста, опрос о занятии https://otus.ru/polls/86414/

Спасибо за внимание!

Приходите на следующие вебинары

Олег Мифле
Backend Developer VK, ВКонтакте

Телефон / эл. почта / соц. сети:
https://t.me/mifleo
mifleov@gmail.com