

Как выжить под нагрузкой: отказоустойчивый сервер, умный клиент

Игорь Луканин

Surviving overload: fault-tolerant servers, smart clients

Igor Lukanin

Intro

Surviving overload

Fault-tolerant servers

Smart clients

Takeaways

Kontur

The largest .NET product company in Russia:

- 40+ products for 2 000 000 clients
- 1000+ engineers in 70 teams

Kontur

The largest .NET product company in Russia:

- 40+ products for 2 000 000 clients
- 1000+ engineers in 70 teams
- 5000+ microservice replicas under load
- 800+ analysed post-mortems over last two years

yt.skbkontur.ru			+			
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Tree view	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	★	fuckups-877	Часть эвентов по продажам в определенные даты не долетела до сервиса расчета остатков			Nov 19
<input type="checkbox"/>	★	fuckups-885	Пустые письма от аутха			Nov 19
<input type="checkbox"/>	★	fuckups-871	При рестарте демона BIRD на релее часть метрик была задержана. Пара точек на некоторых графиках была потеряна			Nov 19
<input type="checkbox"/>	★	fuckups-879	Не работала регистрация в Отчёт.ру			Nov 19
<input type="checkbox"/>	★	fuckups-854	Ошибка в получении электронных сертификатов от КаБУЦ			Nov 16
<input type="checkbox"/>	★	fuckups-880	Кратковременная потеря связности с цодом BST	<input type="checkbox"/>		Nov 14
<input type="checkbox"/>	★	fuckups-881	таймауты при работе с pfrservice			Nov 14
<input type="checkbox"/>	★	fuckups-873	Ошибка при конвертации базы Томлайн с релиза 148			Nov 12
≡ <input type="checkbox"/>	★	fuckups-876	Пропускали часть данных при миграции из Evrika в ElbaStaff			Nov 12
<input type="checkbox"/>	★	fuckups-875	Не работал сайт Контур.1С			Nov 12
<input type="checkbox"/>	★	fuckups-874	Магнит. Встала обработка документов			Nov 12
<input type="checkbox"/>	★	fuckups-872	Скриптовая ошибка при открытии редактора PCB "Context not found" (не мешает работе страницы, просто пишется в	<input type="checkbox"/>		Nov 12
<input type="checkbox"/>	★	fuckups-861	В IE не открывалась карточка клиента в Биллинге			Nov 09
<input type="checkbox"/>	★	fuckups-870	Недоступность гипервизора hv4-15			Nov 09
<input type="checkbox"/>	★	fuckups-869	После восстановления виртуальной машины ft.zebra.exporter удачно запустился с испорченным состоянием	<input type="checkbox"/>		Nov 09
<input type="checkbox"/>	★	fuckups-868	Недоступность поиска в сервисе реквизитов (тестовая)			Nov 09
<input type="checkbox"/>	★	fuckups-865	Недоступность sql-сервера для коннекторов Ритейла и Диодока			Nov 07
<input type="checkbox"/>	★	fuckups-859	В Диодоке не работали списки документов	<input type="checkbox"/>		Nov 06
<input type="checkbox"/>	★	fuckups-867	Временная недоступность сервисов Контур.Алкодекларация и Контур.Алкосверка из за отказа нод кассандры.			Nov 06
<input type="checkbox"/>	★	fuckups-677	500 ошибка при ответе на требование о предоставлении пояснений по НДС у ИП	<input type="checkbox"/>		Nov 06
<input type="checkbox"/>	★	fuckups-863	Проблемы с шифрованием ГОСТ в КОПФ			Nov 06

yt.skbkontur.ru

Showing issue #53 of 869 found

Created by [Жиров Евгений Вячеславович](#) 13 Nov 2018 14:55 Updated by [Кармазин Дмитрий Иванович](#) 14 Nov 2018 visible to: [All Users](#)

★ fuckups-880 Кратковременная потеря связности с цодом BST

Summary

Датацентр BST отрезало от остальных датацентров примерно на 15 секунд. TCP соединения между датацентрами устанавливались с перебоями в течении 90 секунд.

@dc-нос: BST отрезало только от SD2, из других цодов связь была.

Мастер Кансо находился в BST и потерял соединение с тенью в другом датацентре, в результате чего он в течении 120 секунд находился в режиме readonly, пока не смог инициализировать новую тень. Мастер Зебры не смог сделать запись в оплог в Кансо и потерял свое состояние. Он успешно переподнял его через несколько минут после восстановления мастера Кансо.

Ущерб

~20k клиентских Ошибок в Кансо
574 клиентских Ошибок в Зебре

Также пострадала телеметрия: не доходили данные на графики и в систему алертинга.

Графики

[Клиентски ошибки в Кансо](#)
[Клиентски ошибки в Зебре](#)

Как заметили

Project	fuckups
Fuckup State	Investigation
Teams	Infrastructure
Services	Many services
Damage type	Недоступность сервиса
Trigger	Падение сети
Noticed by	Алерты
Author	Жиров Евгений Вячеславович
QTeam assignee	No qteam assignee
Problem appeared MSK	13/11/18 12:20
First noticed MSK	13/11/18 12:21
Investigation started MSK	13/11/18 12:21
Problem resolved MSK	13/11/18 12:35
Obsolete_Reason	Changes
Obsolete_Type	Bug
Obsolete_Subsystems	No Subsystem

Created by Ионов Денис Андреевич 30 Nov 2015 17:06 Updated by Aleksandr Kazakov 11 Dec 2015 14:30 visible to: All Users

fuckups-60 Отказ в обслуживании сервиса реквизитов

Симптомы

Перестали отвечать за разумное время реплики сервиса реквизитов, следом отвалился аутх

Время начала: В 14:57 по Екб стрельнуло на тестовой, следом в 15:58 по продакшену

Время конца: на тестовой в 15:03, на боевой 16:30 Екб

Проблема снова случилась, на этот раз на тестовой площадке 1го декабря. Был запущен индексадор бананы, который читал данные из сервиса реквизитов как раз с не оптимальным фильтром. Проблема продолжалась 20 минут. Причины и меры те же самые.

Причины

В поисковые индексы, расположенные на сервисе реквизитов начали прилетать запросы, приводившие к полному рескану поискового индекса (запросы вида поле~*значение*). Сначала по тестовой прилетели запросы, потом по боевой.

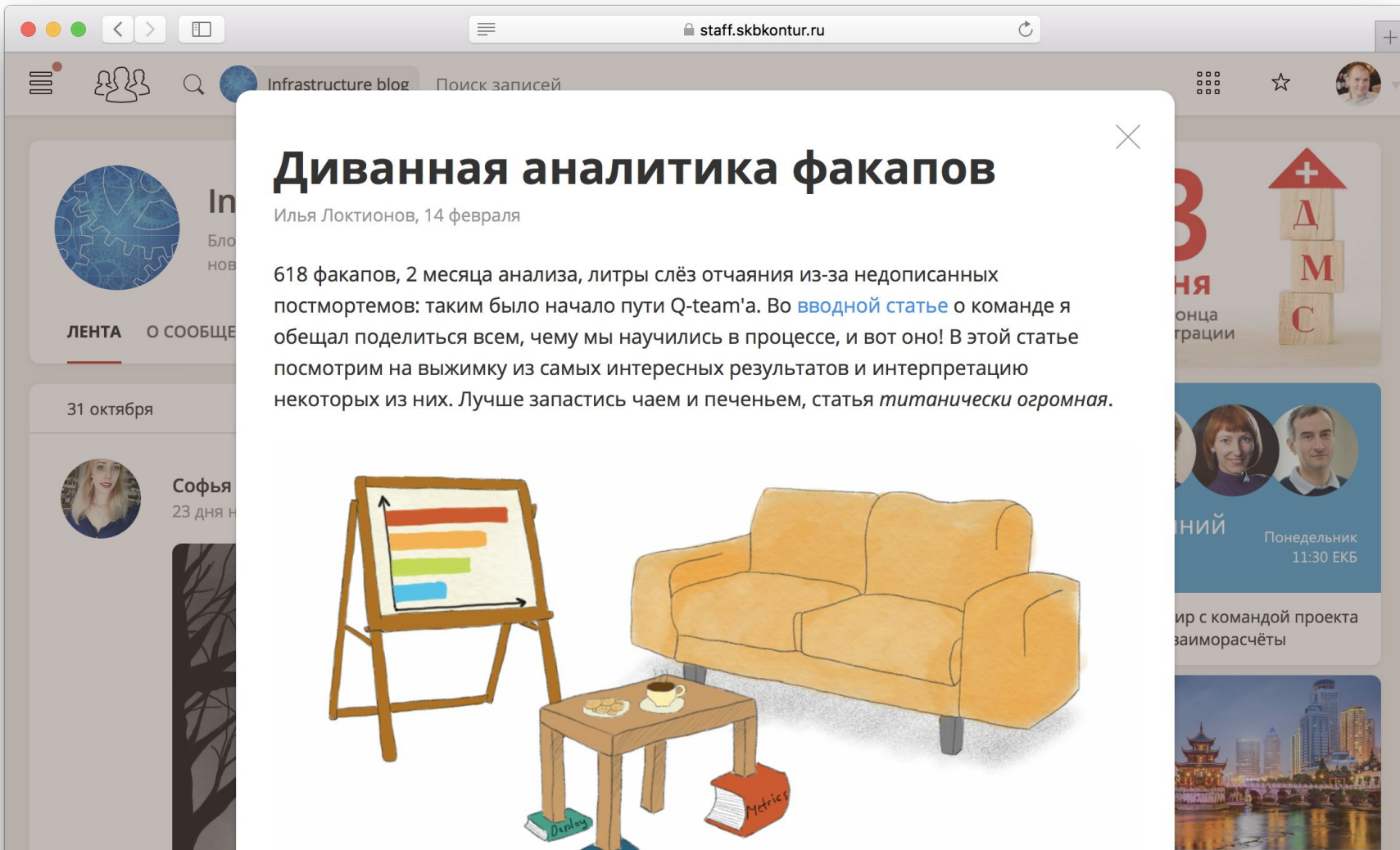
Как заметили

На тестовой о подозрительной активности через 4 минуты отписались парни из СПС. По логам мы не смогли оперативно разобраться, что произошло. При возникновении активности на боевой сразу же получил уведомление от мойры на телефон после первой же пятисотки и начал разбираться (каждая пятисотка это ЧП, по которому начинается разбирательство)

Что предприняли

Посмотрели в мониторинге, что все реплики постепенно начинают приближаться к 100% CPU. По логам начали размащивать, кому мы отвечали статус кодом 503 и какой запрос при этом приходил. Примерно через 5 минут

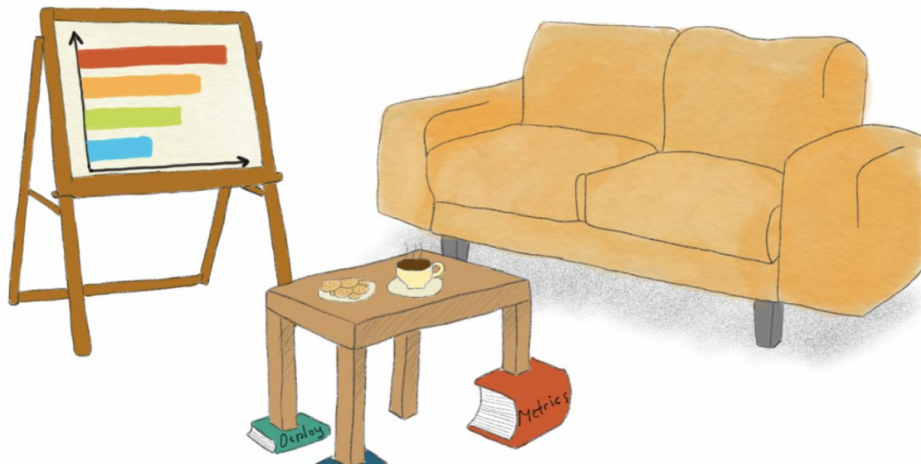
Project	fuckups
Fuckup State	Investigation
Teams	Not specified
Services	No services
Damage type	No damage type
Trigger	No trigger
Noticed by	Заметили случайно
Author	No author
QTeam assignee	No qteam assignee
Problem appeared MSK	dd/MM/YY HH:mm
First noticed MSK	dd/MM/YY HH:mm
Investigation started MSK	dd/MM/YY HH:mm
Problem resolved MSK	dd/MM/YY HH:mm
Obsolete_Reason	Changes
Obsolete_Type	Bug
Obsolete_Subsystems	No Subsystem



Диванная аналитика факапов

Илья Локтионов, 14 февраля

618 факапов, 2 месяца анализа, литры слёз отчаяния из-за недописанных постмортемов: таким было начало пути Q-team'a. Во [вводной статье](#) о команде я обещал поделиться всем, чему мы научились в процессе, и вот оно! В этой статье посмотрим на выжимку из самых интересных результатов и интерпретацию некоторых из них. Лучше запастись чаем и печеньем, статья *титанически огромная*.



Найдено п

ВСЕ ЗАПИСИ РА



Мы пода
неравноду
директор
Добрые де



Докладь
а алерты в
сервис-че
Летний Ко



Новости
и сервис
Мойдоды
Группа под



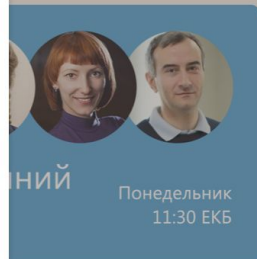
Дайджес
времени.
компиляц
Infrastructu



Тизеры д

А Мойдодыр здесь причем?

Мойдодыр — это сервис-чеклист для гигиены твоего приложения. Он подскажет как сделать решение надежнее: например, даст советы по сбору метрик и нагрузочному тестированию. А еще Мойдодыр сравнит чистоплотность твоего сервиса с другими.



ир с командой проекта
заиморасчёты



Intro

Surviving overload

Fault-tolerant servers

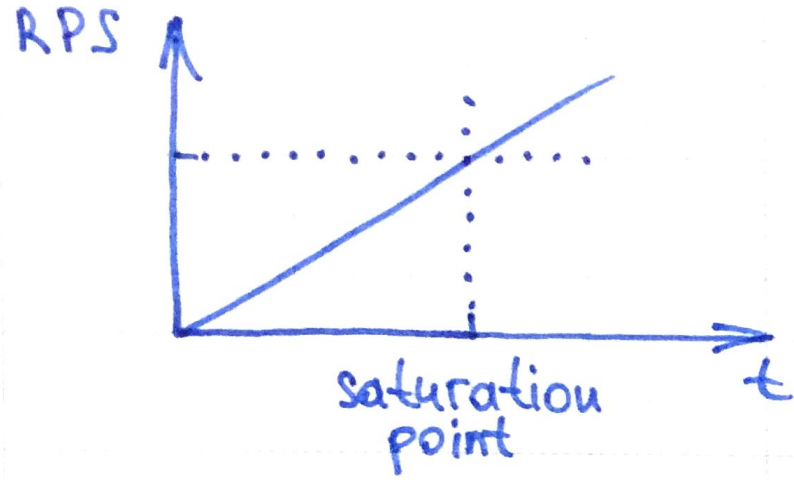
Smart clients

Takeaways

Server

An HTTP backend, which

- serves client requests
- responds with HTTP 200 status codes
- consumes resources: CPU, memory, I/O, etc.





Overload effects

Clients would experience:

- increased response latency
- HTTP 500 error codes
- timed-out connections
- rejected connections
- application crash

Resource utilization

High is good: there's a bottleneck

Low is bad: review the code and environment

Resource underutilization

Check:

- resource contention — e.g., lock contention

Resource underutilization

Check:

- resource contention — e.g., lock contention
- shared objects — e.g., thread and connection pools

Resource underutilization

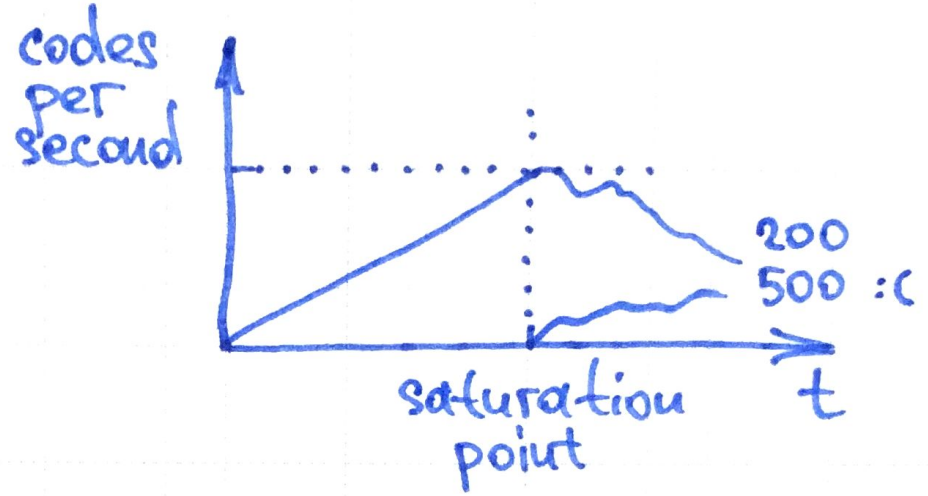
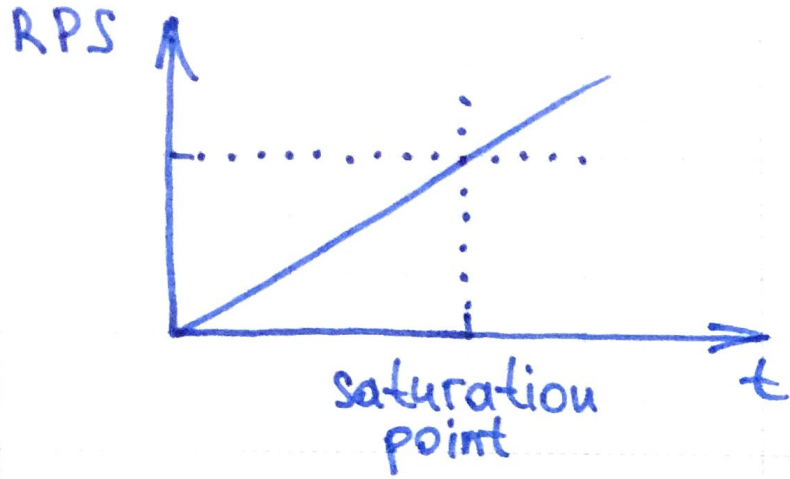
Check:

- resource contention — e.g., lock contention
- shared objects — e.g., thread and connection pools
- synchronous operations — e.g., network and disk I/O

Resource underutilization

Check:

- resource contention — e.g., lock contention
- shared objects — e.g., thread and connection pools
- synchronous operations — e.g., network and disk I/O
- runtime configuration — e.g., GC settings



Resource overutilization

Await for high CPU consumption:

- most tasks are CPU-bound*
- other resources can be over-provisioned*
- with GC, memory pressure → CPU consumption

* It depends.

Resource limit

CPU consumption is better than RPS.

Beware of:

- heterogeneous requests

Resource limit

CPU consumption is better than RPS.

Beware of:

- heterogeneous requests
- heterogeneous replicas

Resource limit

CPU consumption is better than RPS.

Beware of:

- heterogeneous requests
- heterogeneous replicas
- heterogeneous environments — e.g., staging

Resource limit

CPU consumption is better than RPS.

Beware of:

- heterogeneous requests
- heterogeneous replicas
- heterogeneous environments — e.g., staging
- co-hosted applications — e.g., other processes

Resource limit

CPU consumption is better than RPS.

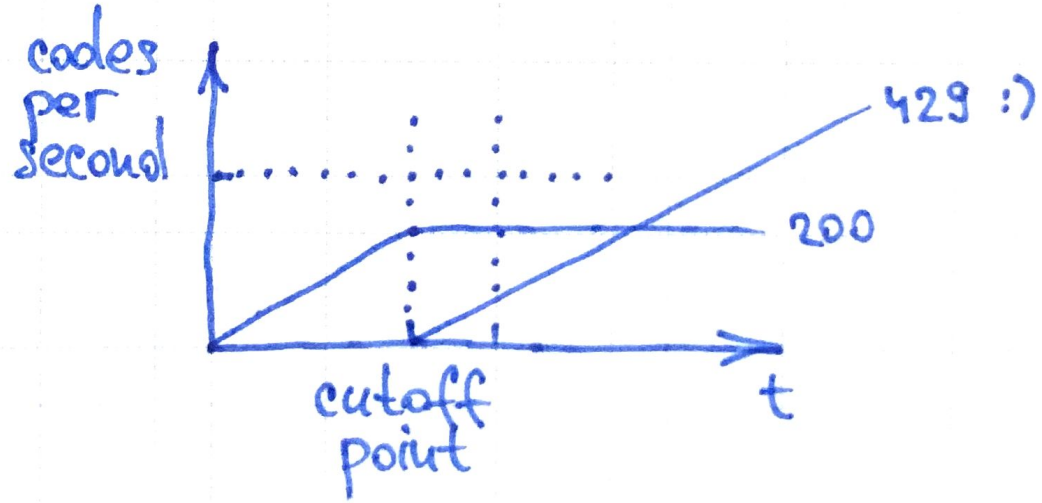
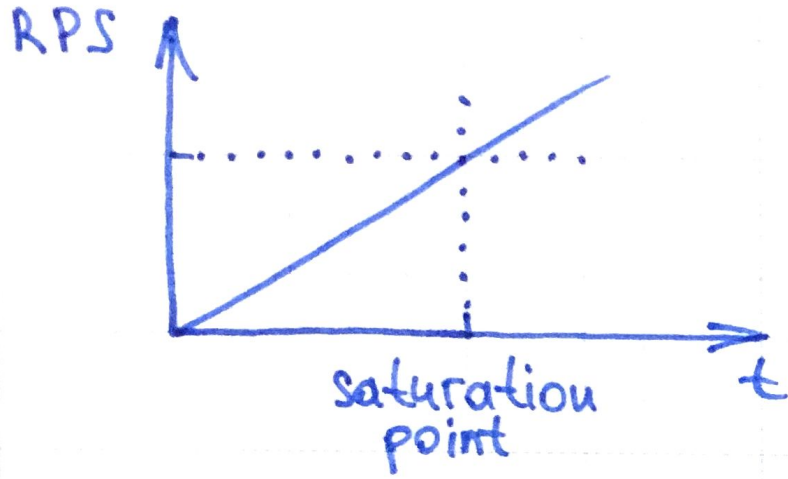
Beware of:

- heterogeneous requests
- heterogeneous replicas
- heterogeneous environments — e.g., staging
- co-hosted applications — e.g., other processes
- unusual operation modes — e.g., startup, GC cleanup

Fault-tolerant server

An HTTP backend, which

- measures resource utilization
- serves degraded responses, if applicable
- reliably responds with HTTP 429 error codes



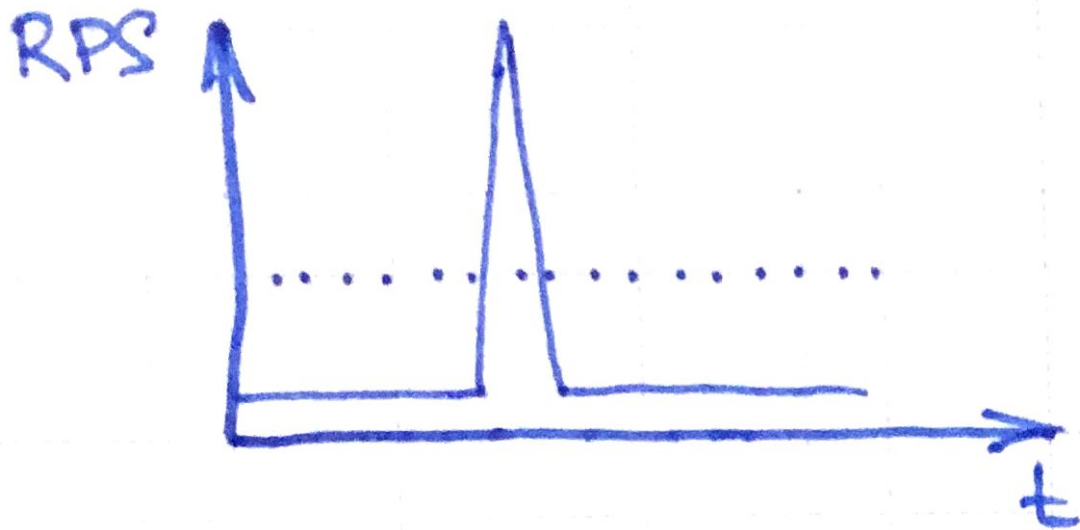
Intro

Surviving overload

Fault-tolerant servers

Smart clients

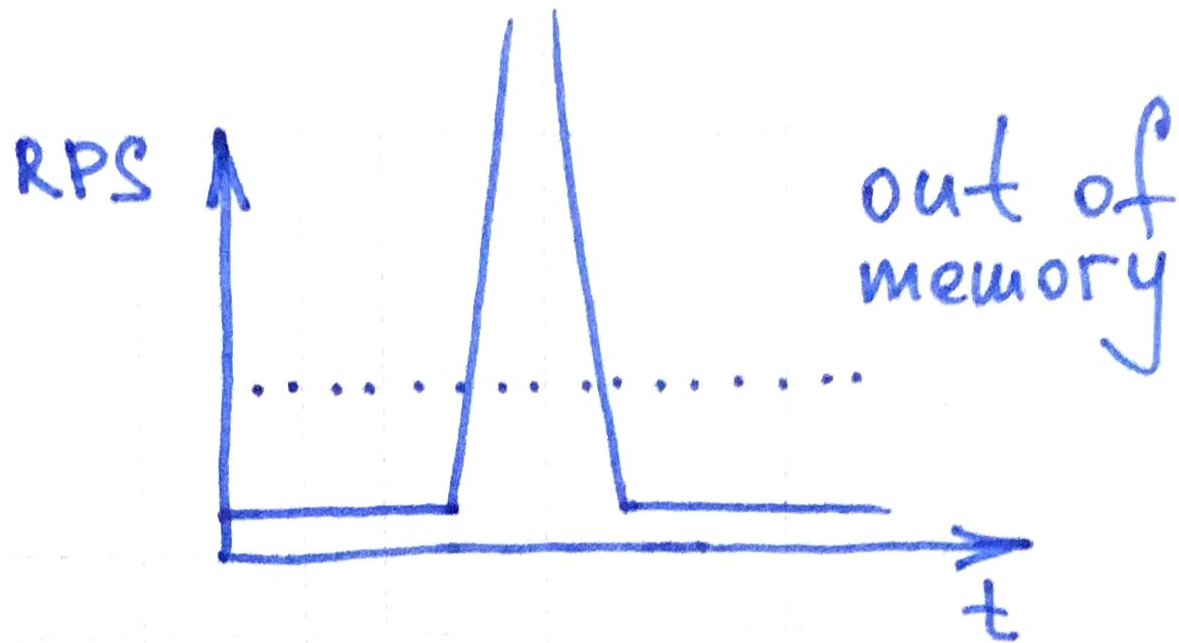
Takeaways



Smoothing the load

Beware of uneven load.

Use a task queue.



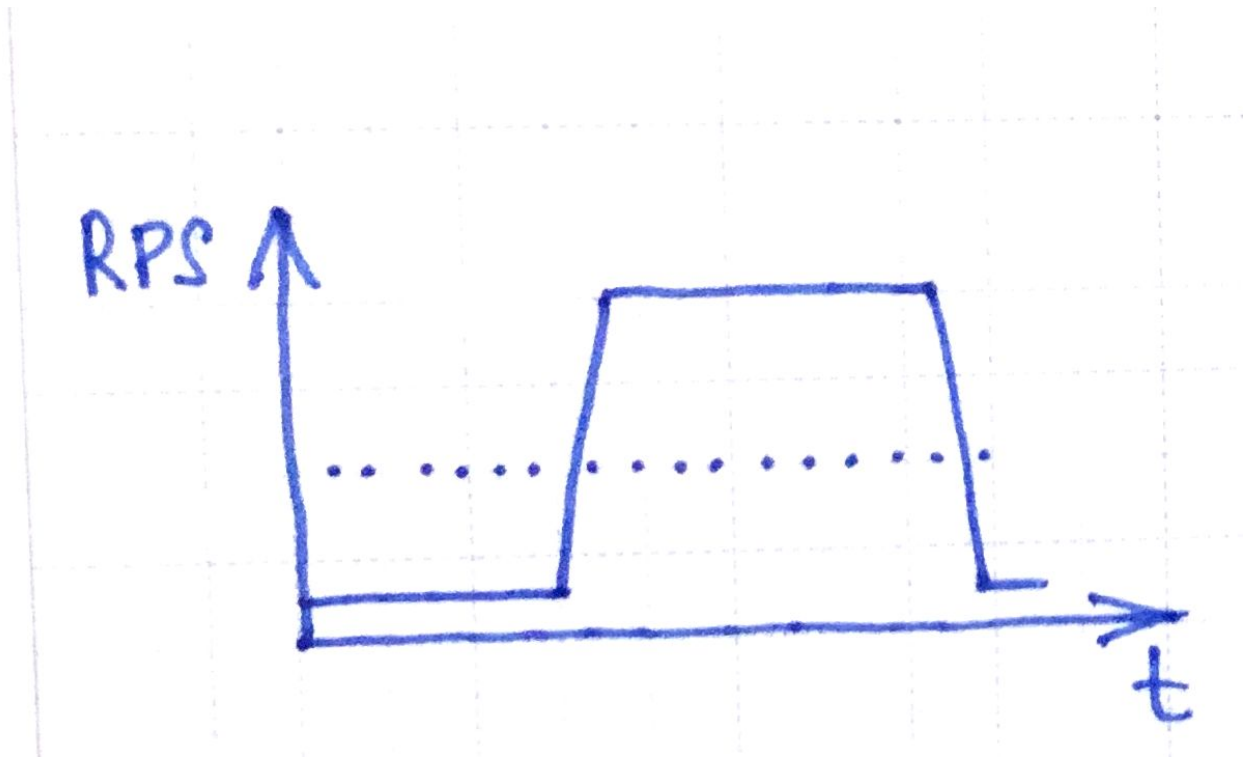
Smoothing the load

Beware of uneven load.

Use a task queue, which is

- limited in capacity

Reject the remaining requests.



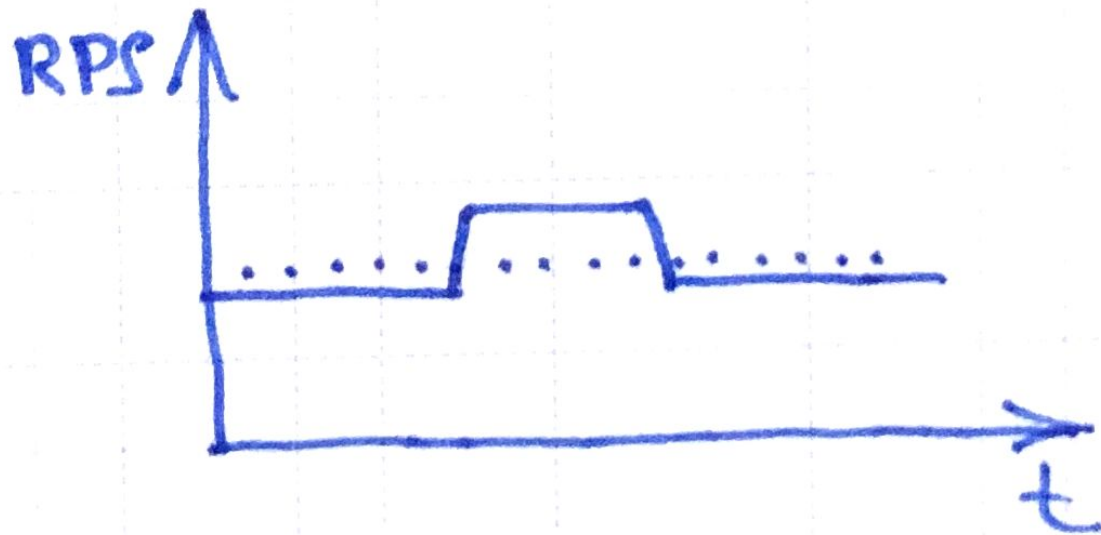
Smoothing the load

Beware of uneven load.

Use a task queue, which is

- limited in capacity
- limited in task TTL

Reject the remaining requests.



Smoothing the load

Beware of uneven load.

Use a task queue, which is

- limited in capacity
- limited in task TTL
- LIFO rather than FIFO

Reject the remaining requests.

Request criticality

Beware of different business scenarios.

Use request priorities:

- set by the impact on users
- use 2–4 levels — e.g., low, normal, critical
- apply quotas per level — e.g., drop low-level tasks

контур.фокус

джуг ру груп

RUS



Новости
наблюдения



Списки



Заметки



Проверка
паспортов



Недвижимость



Поиск новых
клиентов

Уплаченные налоги и сборы, сумма доходов и расходов по данным ФНС появились в Фокусе

Сообщения о намерении подать заявление о банкротстве, бухотчетность 2017, вакансии и другие новости сервиса

Информация о среднесписочной численности и специальных налоговых режимах появилась в Фокусе

Финотчётность за 2017 год по более чем 105 тысячам компаний уже в сервисе

Найдено 1 за 0.06 сек. — [показать аналитику](#)

1 **ООО "ДЖУГ РУ ГРУП"**

Деятельность по организации конференций и выставок
199004, г Санкт-Петербург, линия 9-я В.О, 34А, оф 500
ИНН: 7801341446 ОГРН: 1177847388465 — 01.12.2017

Россия 1

- Беларусь
- Казахстан

Все регионы

- Москва
- Санкт-Петербург
- Свердловская обл
- Пермский край
- Новосибирская обл
- ...

Все отрасли

- Вся торговля
- Строительство

focus.kontur.ru

kontur.фокус

lukanin@kontur.ru | Выйти

Введите ИНН, ОГРН, название, адрес или ФИО

ООО "ДЖУГ РУ ГРУП"

+ Списки и наблюдение Заметки Выгрузка

Сводка Связи

Общество с ограниченной ответственностью "Джуг Ру Груп"

Действующее предприятие

Микропредприятие

Сотрудники: 1 человек

ИНН 7801341446
КПП 780101001
ОГРН 1177847388465

Дата образования: 1 декабря 2017

199004
г Санкт-Петербург
[линия 9-я В.О, 34А](#) 133 [Осмотреть](#)

Автоматическая проверка

[Экспресс-отчёт](#) (Eng)

Налоги и сборы в 2017

Применяет УСН

Записи в ЕГРЮЛ

- 04.12.2017 Внесение сведений о регистрации в ПФР
- 04.12.2017 Внесение сведений о регистрации в ФСС
- 01.12.2017 Внесение сведений об учете в налоговом

Load quotas

Beware of heterogeneous clients.

Use per-client quotas:

- identify clients by revocable API keys
- measure per-client load
- apply quotas per client — e.g., drop non-SLA tasks

Fault-tolerant server

Ready for:

- uneven load
- business scenarios
- heterogeneous clients

May still fail to survive overload.

Fault-tolerant cluster

Cluster of server replicas:

- provides more capacity than a single replica
- provides backup if a replica fails — for any reason

Use 3+ replicas in a cluster. If one replica fails, there's still more than 50 % of capacity

Intro

Surviving overload

Fault-tolerant servers

Smart clients

Takeaways

Load balancing

Distribute requests between replicas.

Use round-robin.

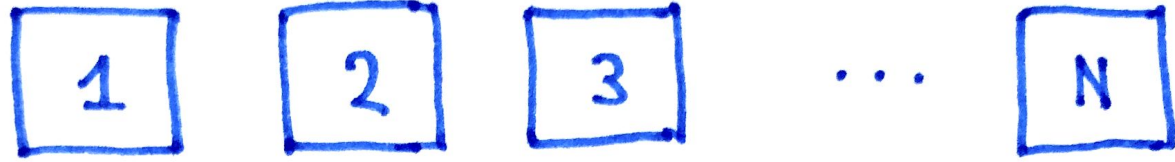
1

2

3

...

N



Load balancing

Distribute requests between replicas.

Use round-robin:

- simple — i.e., stateless

Distributes load unevenly.

1

2

3

...

N

active
requests

6

3

7

4

1

2

3

...

N

active requests

6

3

7

4



Load balancing

Distribute requests between replicas.

Use round-robin:

- simple — i.e., stateless
- least-loaded — i.e., tracking active requests

Client has limited view of replica states.

Prone to traffic sinkholing.

1

2

3

...

N

used
capacity

50%

40%

10%

60%

1

2

3

...

N

used
capacity

50%

40%

10%

60%



Load balancing

Distribute requests between replicas.

Use round-robin:

- simple — i.e., stateless
- least-loaded — i.e., tracking active requests
- weighted — i.e., tracking real-time replica capacity

Request Submission

Choose a strategy for each request. Send to:

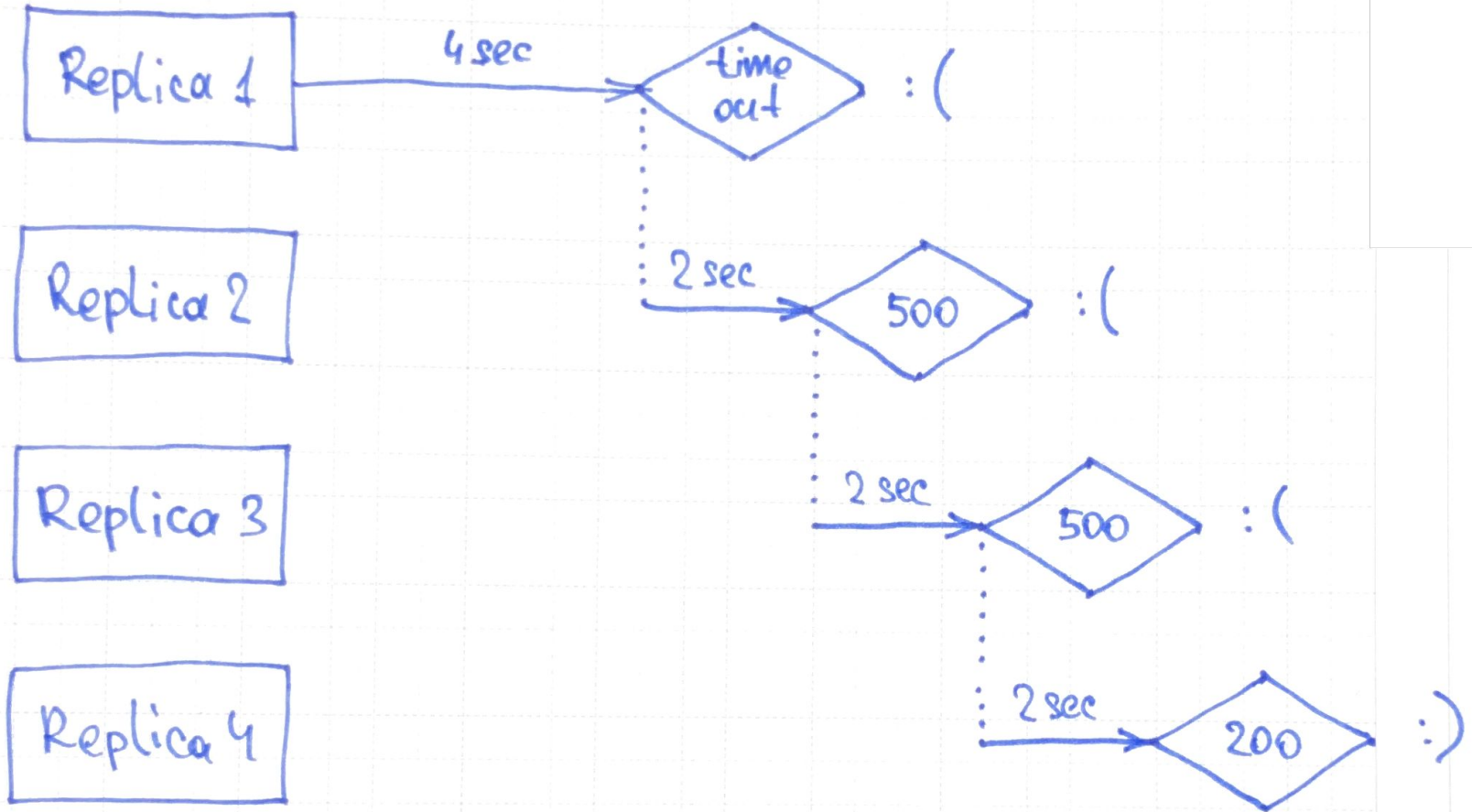
- a single replica



Request Submission

Choose a strategy for each request. Send to:

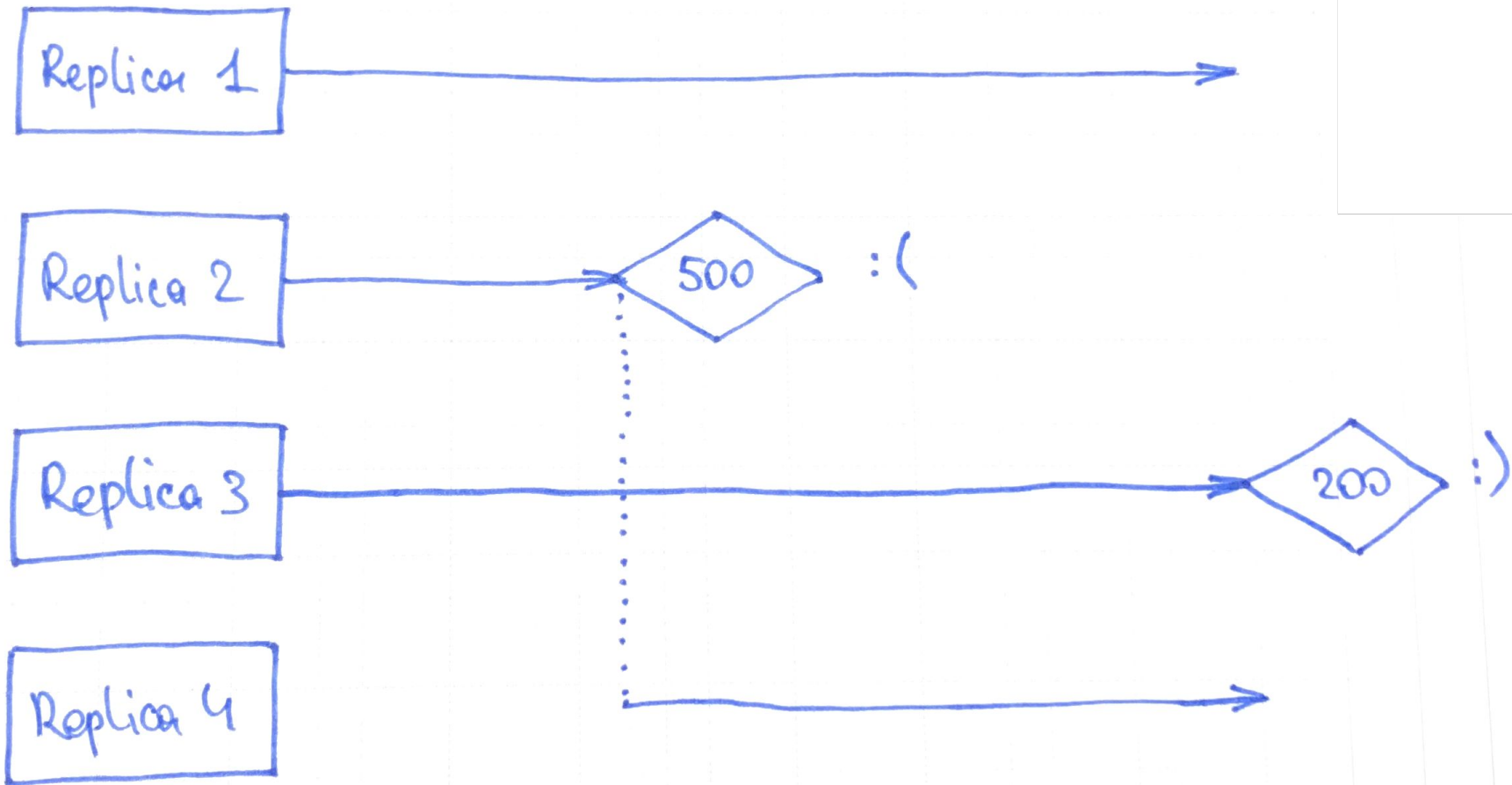
- a single replica
- multiple replicas, sequentially



Request Submission

Choose a strategy for each request. Send to:

- a single replica
- multiple replicas, sequentially
- multiple replicas, concurrently



Request Submission

Choose a strategy for each request. Send to:

- a single replica
- multiple replicas, sequentially
- multiple replicas, concurrently
- multiple replicas, concurrently and adaptively

Request Retrial

Choose a strategy for each failed request:

- set the limit — e.g, 1-3 attempts
- use linear or exponential backoff
- use jitter

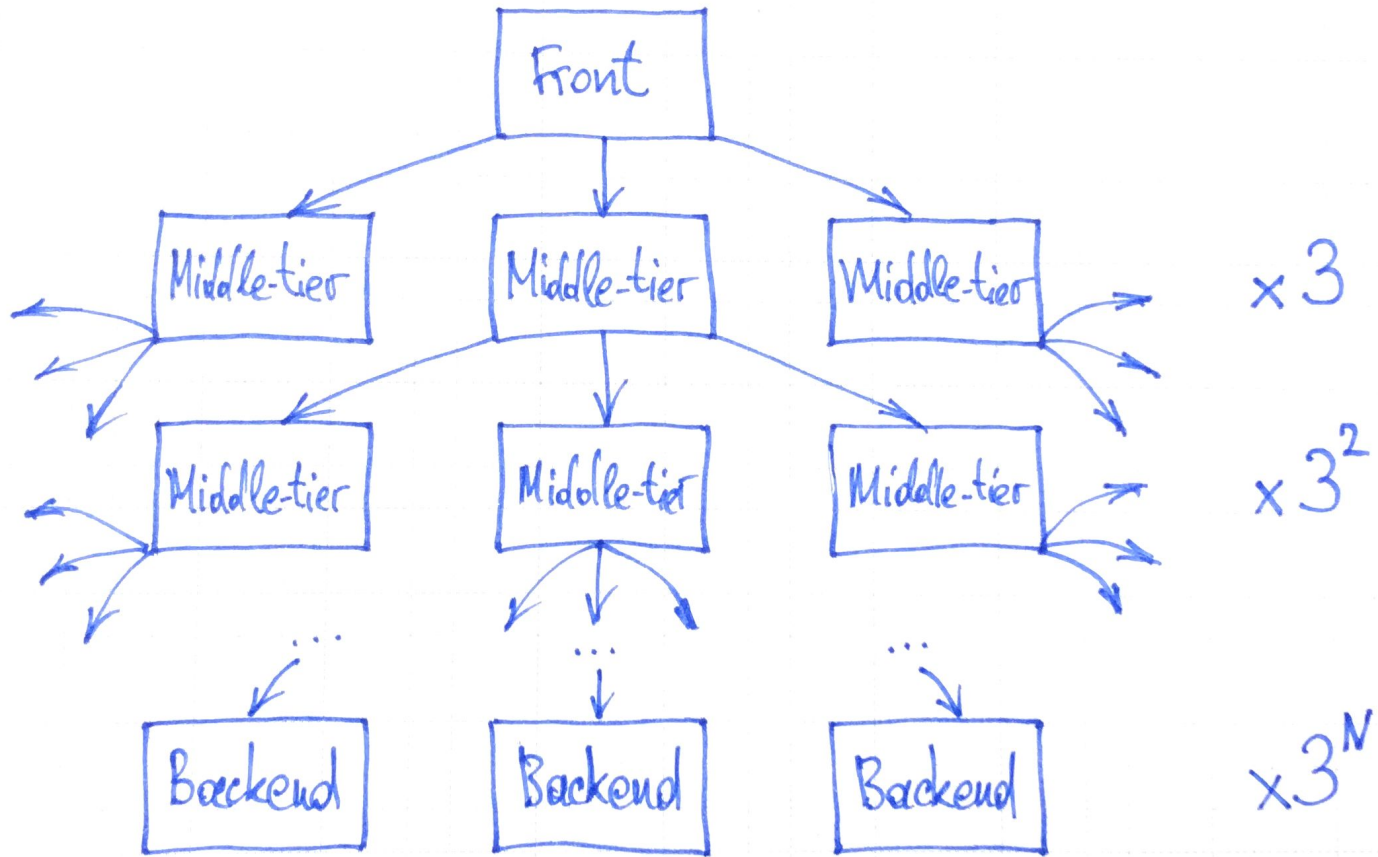
Overloaded cluster

Request strategies and retries:

- **multiply load $\times 3^*$** on an overloaded cluster
- **multiply load $\times 3^N$** on an overloaded N-tier system,
and lead to cascading failure

(10–20 % of overloaded replicas are okay.)

* Depends on concurrency level and retry limit.



Load multiplication factor

Don't retry requests automatically:

- on all replicas — e.g., use replica subsetting
- when server asked to — e.g., via an HTTP header
- when non-critical — e.g., a user can retry manually

Load multiplication factor

Propagate metadata through system tiers:

- request priorities
- request timeouts

Adaptive replica subsetting

Limit replicas available for retrieval.

Track the factor $K = \text{sent requests} / \text{used replicas}$.

Adaptive replica subsetting

Limit replicas available for retrieval.

Track the factor $K = \text{sent requests} / \text{used replicas}$:

- if $K \approx 1$, everything is okay
- if $K > 1$, some replicas are overloaded
- if $K \geq K_{\text{critical}}$, a client doesn't retry requests

Compare $3^5 = 243$ vs. $1.3^5 = 3.7^*$

* Threshold of 1.3 allows for 8 healthy and 2 overloaded replicas in a cluster.

Adaptive request throttling

Limit the probability of request submission.

Track the factor $K = \text{requests} / \text{accepts}$.

Adaptive request throttling

Limit the probability of request submission.

Track the factor $K = \text{requests} / \text{accepts}$.

Calculate request rejection probability:

$$P_{\text{reject}} = \max \left(0, \frac{\text{requests} - K \cdot \text{accepts}}{\text{requests} + 1} \right)$$

- if $K \approx 1$, everything is okay
- if $K > 1$, some replicas are overloaded
- if $K \geq K_{\text{critical}}$, a client drops requests with P_{reject}

Intro

Surviving overload

Fault-tolerant servers

Smart clients

Takeaways

Test your system

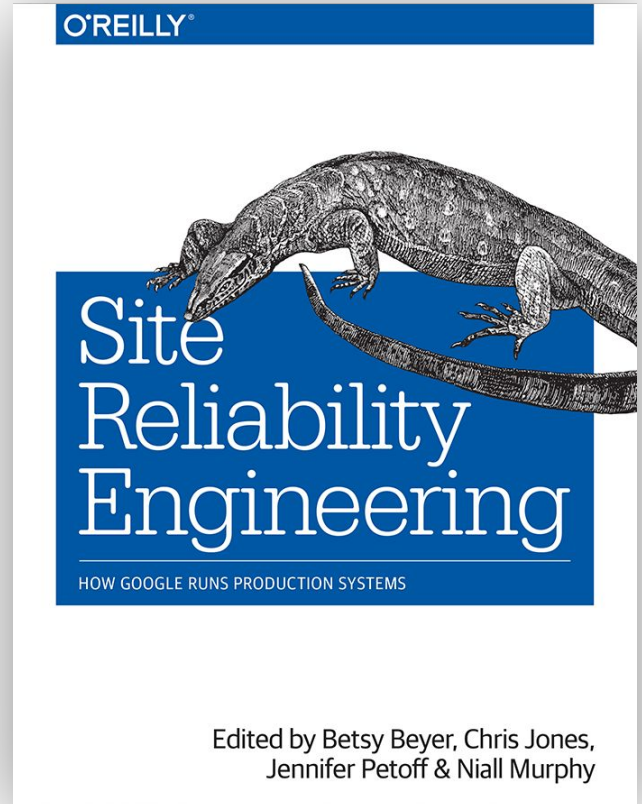
- Test a single server replica under load
- Test the cluster and its client
- Test unstable cluster configurations
- Test the system as a whole

Then make it survive the overload.

Read the SRE book

Chapters 19–22
on load balancing
and handling overload


landing.google.com/sre/books



Edited by Betsy Beyer, Chris Jones,
Jennifer Petoff & Niall Murphy

github.com

Search or jump to... Pull requests Issues Marketplace Explore



Vostok

http://vostok.tools | opensource@skbkontur.ru


Repositories 64 | People 30 | Teams 1 | Settings

Find a repository... Type: All Language: All Customize pinned repositories [New](#)

sys.metrics.windows

Classes for collecting system metrics on Windows.

C# ★ 2 MIT Updated 23 minutes ago



sys.metrics.perfcounters

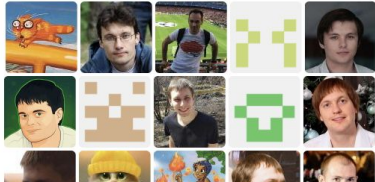
The Vostok.Sys.Metrics.PerfCounters library provides interface for efficient consuming Windows Performance Counters data. It uses Performance Data Helpers (PDH) API under the hood.

C# MIT Updated 5 hours ago

Top languages

- C#
- JavaScript
- Java
- CSS
- Dockerfile

People 30 >



Have a look at Vostok

Vostok is a toolbox for production-ready and fault-tolerant .NET microservices.

Have a look at:

- sys.metrics
- clusterclient
- throttling

Have a look at Vostok

["Collecting telemetry from .NET microservices"](#) —

Alexey Kirpichnikov @ CodeFest 2017

["Microservice interaction with HTTP/2"](#) —

Evgeny Zhironov @ DotNext 2018

FMS Elasticsearch HeapUsedPercent

[Edit](#) [Duplicate](#)

Target aliasByNode(FMS.elasticsearch.vm-fms-*.jvm.mem.heap_used_percent, 2, 3)

Value Warning: 90. Error: 95. Set NODATA if has no value for 600 seconds

Schedule Everyday 00:00–23:59

Tags FMS DevOps elasticsearch

Current state

Events history

Name	Last event	Value	Delete NODATA metrics
● vm-fms-el01.fms-es	October 5, 07:51:35	55	Maintenance Delete
● vm-fms-el02.fms-es	April 6, 15:31:44	60	Maintenance Delete
● vm-fms-el03.fms-es	October 5, 07:51:35	51	Maintenance Delete

Have a look at Moira

**Moira is a real time alerting system
based on Graphite data.**

Collect metrics and respond to alerts.

["What we learned making Moira"](#) —

Alexey Kirpichnikov @ HighLoad++ 2018

Thank you!

- tech.kontur.ru — more open source
- t.me/KonturTech — news and events
- t.me/igorlukanin — ping me!