

Отказоустойчивость и балансировка трафика для MySQL-бэкенда

Анастасия Распопина,
Владимир Федорков
Big Data Days Moscow,
09.10.2019



О докладчиках

- **Владимир Федорков**
- 19+ лет опыта с MySQL.
- Вытягиваю проекты из сложных жизненных ситуаций
 - в высоконагруженных проектах простых не бывает
- Специализация - MySQL
 - Также – ProxySQL и полнотекстовый поиск
- **Анастасия Распопина**
- Коммуникации в СУБД-индустрии.
- Придумываю интересные доклады (DevRel)
 - ориентирую экспертов в их морях смыслов
- Специализация – мероприятия
 - В РФ и за рубежом

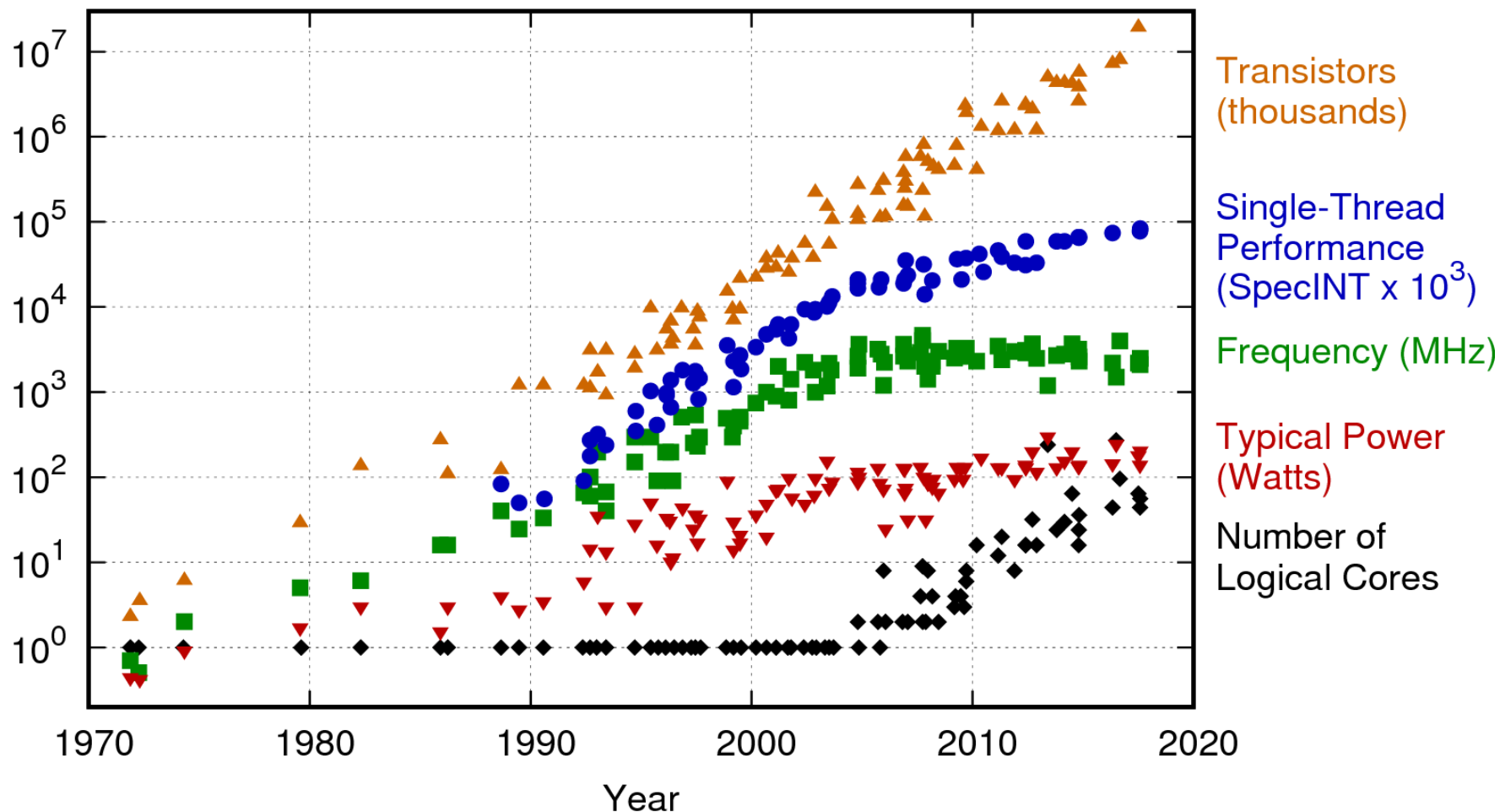


Информационная эпоха, 2019

1. Microsoft 904,860
2. Apple Inc. 895,670
3. Amazon.com 874,710
4. Alphabet Inc. 818,160
5. Berkshire Hathaway 493,750
6. Facebook 475,730
7. Alibaba Group 472,940
8. Tencent 440,980
9. Johnson & Johnson 372,230
10. ExxonMobil 342,170
- ... Ford motors 38,221

Изменились скорости

42 Years of Microprocessor Trend Data



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

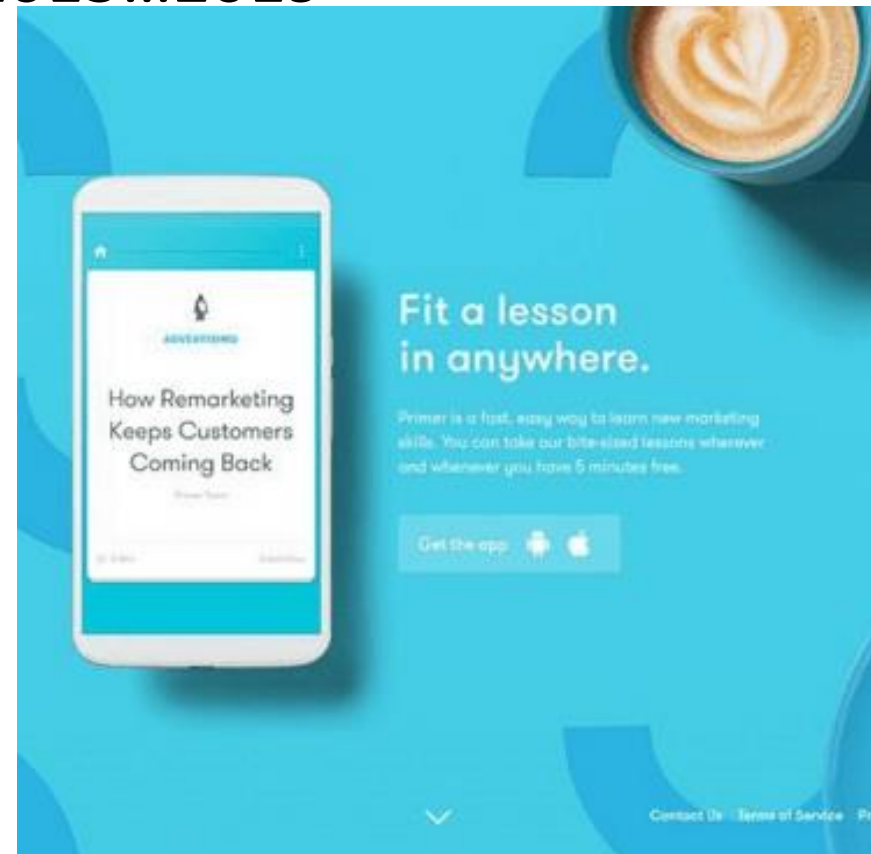


Поменялся внешний вид сайтов

1998...2000



2015...2019



Интернет-магазин 2001-го года

- Физический магазин в центре Москвы.
- На сайте:
 - каталог
 - страница товара
 - корзина
 - страница пользователя
- Заказ оформляется после созвона с клиентом.
- Один сервер, потому что если вдруг умрёт – поднимем.

Интернет-магазин 2019

- Бесперебойная работа сайта 24/7.
- Основная точка продаж – Сеть.
- Заказ оформляется и оплачивается прямо в Сети.
- На сайте:
 - сервисы прямого поиска (три штуки);
 - интеграция с 1С (бухгалтерия, склады);
 - сервисы продвижения товаров (платные объявления и «с этим товаром также покупают»);
 - сервис рейтинга.
- Никто даже не считает, сколько там страничек (тысячи!).
- И это только софт!

Ещё кое-что:

- DDoS и сетевая отказоустойчивость.
- Многоуровневое кэширование:
 - CDN-сети, внутренние кэши данных, кэширование запросов.
- Многократное дублирование:
 - Много серверов приложений (stateless)
 - Много серверов уровня хранения данных (stateful)
- И это средний уровень
 - только один датацентр

#Тыжпрограммист! Почини тостер!

- Искусственный интеллект
 - Распознавание образов, навигационные алгоритмы, статистические системы
- Поисковые системы
- Финансовые системы и HFT
- Биоинформатика
- Просчет и оптимизация процессов

Уровень задач изменился

- Один сервер не может и не должен держать всю нагрузку.
 - Даже со 128-ю ядрами и 512-ю гигами памяти
- Много серверов – много отказов
 - Отказаться может всё – и железо, и софт.
 - Единая точка отказа может стать и станет проблемой.
- “Capacity” более важна, чем производительность.
 - Достаточно ли у нас запаса мощности чтобы обслужить пользователей?
 - Сколько пользователей мы можем обслужить с текущим железом?
 - Сколько пользователей выдержит наша архитектура?
- “Capacity” – это сколько и каких серверов нам нужно, чтобы обслуживать клиентов с нужным временем отклика.
- Умения настраивать один сервер недостаточно!
 - Про настройку отдельных серверов есть слайды на <http://astellar.com/>

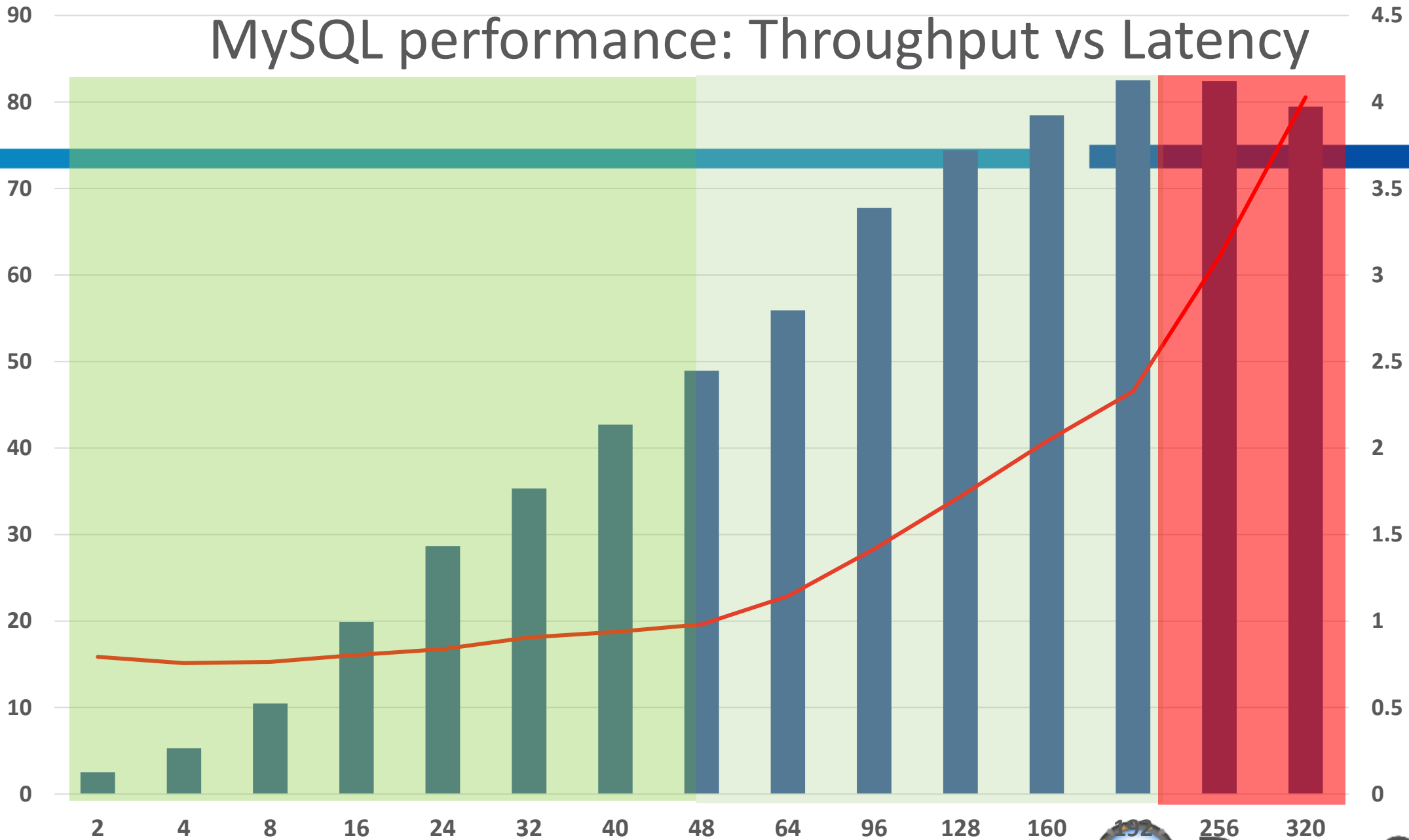
Факторы нагрузки в MySQL

- Один поток на соединение
- Каждый запросы выполняется в один поток
- Каждый запрос – это комбинация нагрузки на CPU и IO.
- IO может быть очень медленным
 - Поэтому MySQL кеширует данные в память
- В целом запас мощности определяется комбинацией времени отклика и пропускной способности.

MySQL performance: Throughput vs Latency

Thousand QPS (kQPS)

Latency (milliseconds)



Active clients (connected processes)



ProxySQL

Все отлично работало, НО!!!

- В пятницу вечером выкатили новый build!
- Наконец-то сделали автоматические миграции!!!
 - «Я поле просто добавил!»
- «Просто у нас слишком много пользователей!!!»

Разработка vs. эксплуатация

- Две разные Вселенные
- Решают разные задачи
 - Поэтому друг друга понимают плохо...
- Иногда кажется, что в реальном мире не пересекаются

Почему так происходит?



У разработки – минимум
несколько часов

VS.



У DBA – максимум несколько минут

Переходный процесс

- Называется тестирование
- Работает не всегда
 - Даже если сделан грамотно
- Учесть все можно, но очень дорого
 - И долго
 - И все равно не гарантирует 100% результата
- **И всё равно падает!**

MySQL все равно падает!

- Почти всегда – самое узкое место.
 - Потому что stateful.
 - Поэтому сложно масштабируется.
 - Нужна поддержка приложения.
 - Многогранная система в целом
 - MVCC, ACID, и много других страшных и непонятных слов.

В чём сложности-то?

- Писать можно не везде.
 - Galera и Group replication тоже не панацея.
- Можно прочитать старые данные.
 - А можно и прочитать, но надо смотреть в GTID.
- Запросы тормозят.
 - Скорость не всегда консистентна.
- В результате тормоза, падения, звонки!

The DevOps Borat

«In startup we are practice Outage Driven Infrastructure»



Как сделать, чтобы не падало?

- Вдумчиво разрабатывать.
- Серьезно тестировать.
- Иметь большой запас по мощностям.

Как сделать, чтобы не падало?

- ~~Вдумчиво разрабатывать~~ - бестолково.
- Серьезно тестировать.
- Иметь большой запас по мощностям.

Как сделать, чтобы не падало?

- ~~Вдумчиво разрабатывать~~ - бестолково.
- ~~Серьезно тестировать~~ - долго.
- Иметь большой запас по мощностям.

Как сделать, чтобы не падало?

- ~~Вдумчиво разрабатывать - бестолково.~~
- ~~Серьезно тестировать - долго.~~
- ~~Иметь большой запас по мощностям - очень дорого!~~

Как сделать, чтобы не падало?

- ~~Вдумчиво разрабатывать - бестолково.~~
- ~~Серьезно тестировать - долго.~~
- ~~Иметь большой запас по мощностям - очень дорого.~~
- Уметь менять всё, что нужно, на лету!
 - Без изменения кода

Балансировщики важны для инфраструктуры

- Для HTTP-трафика их много
 - Потому что это уровень TCP/UDP
- Трафик для базы данных проксировать сложнее.
- У пользователей MySQL всё равно есть выбор: HA Proxy, MaxScale, MySQL Router, Vitesse и т.д.
- Ну и, конечно, ProxySQL.

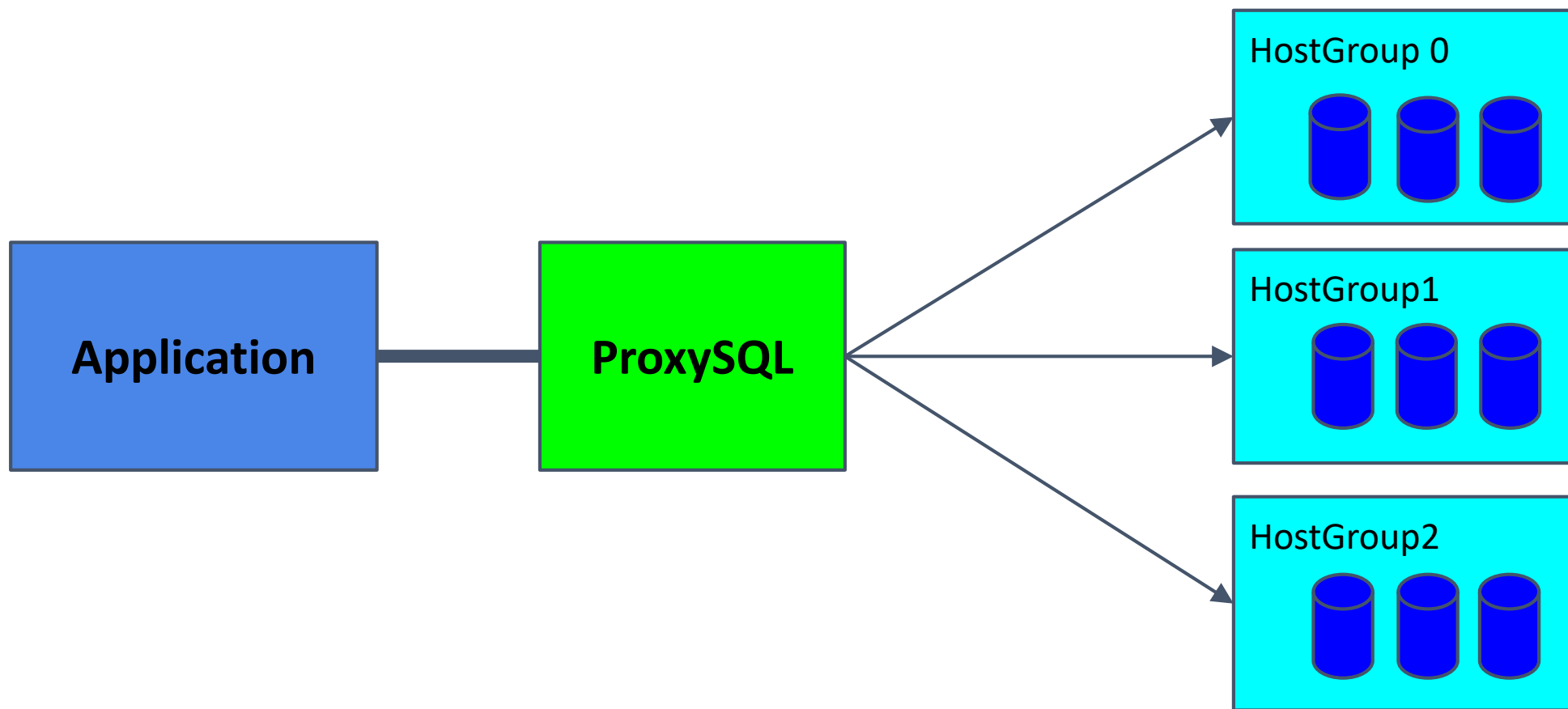


ProxySQL: open source, лицензия - GPL

- ProxySQL «понимает» язык SQL
 - В отличие от прокси слоя 4 ISO/OSI, работающих на уровне транспорта
- ProxySQL «знает» всё о запросе, его обработке, состоянии соединения, авторизации и результатах.
- ProxySQL использует внутренний пул соединений с мультиплексированием для повторного использования существующих соединений.
- ProxySQL может маршрутизировать запросы на основе различных фильтров:
 - By user, by database (schema name), by query itself



Базовый вариант конфигурации



Что может ProxySQL?

- Переписывает запросы «на лету»
- Поддерживает пул соединений и мультиплексирование
- Маршрутизирует сложные запросы
- Разделяет чтение и запись
- Балансирует нагрузку между нодами кластера
- Умеет работать в облаке
- Предоставляет статистику в реальном времени
- Обновляется без downtime
- **Большая часть работает из коробки вообще сразу!**

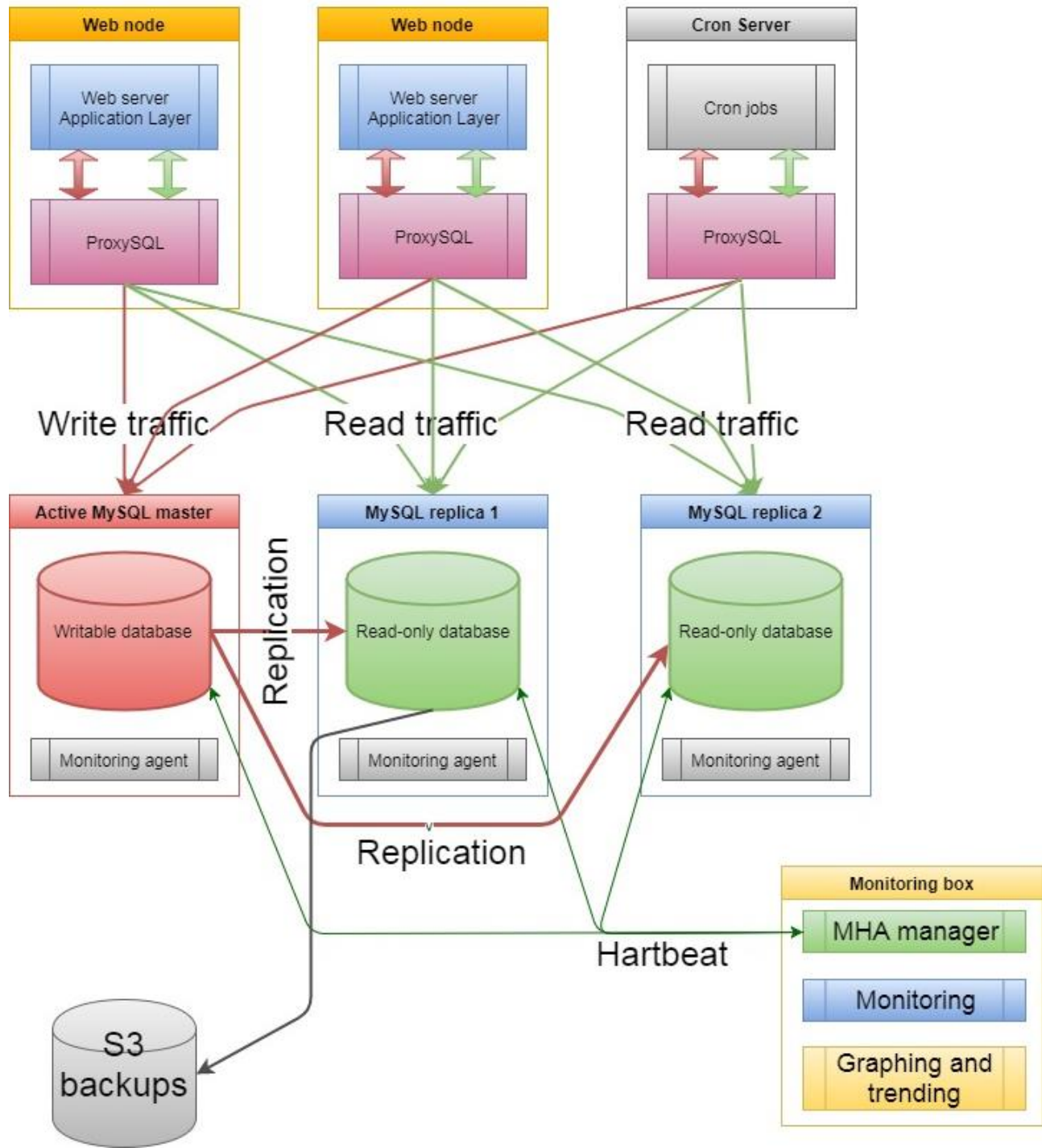
Что-то ещё?

- Кэширование запросов
- Обеспечение отказоустойчивости незаметно для пользователя
- Файервол
- Запросы:
 - регулирование (throttling)
 - зеркалирование
- Изменение конфигурации без задержек
- Поддержка конфигурирования на уровне всего кластера
- Поддержка Galera/PXC и Group Replication

Функция Read/Write split

- Пишем на ноды, которые для этого предназначены.
- Читаем со всех нод
 - Нодам можно присваивать весовые коэффициенты
 - Конфиг можно менять на лету
- Вводим понятие `hostgroups`.
 - `HGxx0`: Write masters
 - `HGxx1`: Read instances
- ProxySQL знает, куда можно писать через флаг `read_only`.
 - Нужно быть аккуратным.

Красивая диаграмма



О ТОМ, КАК ВСЕ МОГЛО БЫ БЫТЬ

Зачем такие сложности?

- HG0: мастера для записи (RW)
- HG1: ноды для основных боевых запросов (RO)
- HG2: reporting реплики

Что в результате?

- Отказы серверов БД перестают значительно влиять на пользователей.
- Потеря данных сводится к минимуму.
- Нагрузка распределяется равномерно между нодами в кластере.
- «Плохие» запросы отлавливаются «на лету».
 - И переписываются там же!
- Если нельзя переписать, то можно отправить на отдельную ноду.
 - Даже внутри транзакции!
- А где здесь спрятаны грабли, спросите после доклада, потому что там глубины и зыбучие пески, а время кончилось :)

Где почитать?

- <https://proxysql.com/blog/>
- На прошлой неделе вышла ProxySQL версии 2.0.7
- <https://twitter.com/proxysql>
- Спросить у нас!
- Будет вебинар
 - запрос можно отправить на info@proxysql.com

Вопросы!

Спасибо!