

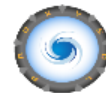
# Как справляться с очень высокими нагрузками и упасть не сразу

---

Федорков. ProxySQL.

Конфиденциально. Местами нецензурно.

Москва 2019



**ProxySQL**

# Что такой дерзкий?

- Не высыпаюсь сильно
- ProxySQL-то помогает
- но не все про него знают
  - Хотя он бесплатный
  - Совсем бесплатный
  - И исходники открыты



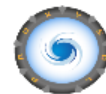
# Кого знаешь?

---

- Нахожу и убираю узкие места в разных проектах
  - Иногда ночью
    - Часто на выходных
- Работаю в ProxySQL
  - До этого в Sphinx и в Percona
- Наболело, поэтому буду эмоционален

# Матом не ругаемся

---



# Но детей лучше убрать

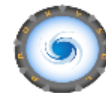
---



# Все отлично работало НО!!!

---

- В пятницу вечером выкатили новый билд!
- 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 реальном мире не пересекаются

# Работал и там и там, всех ненавижу одинаково

---





# Почему так происходит?



Минимум несколько часов

VS



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 все равно падает

# MySQL все равно падает

---

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

# В чем сложности-то?

---

- Писать можно не везде
  - Galera и Group replication тоже не панацея
- Можно прочитать старые данные
- Запросы тормозят
  - Скорость не всегда консистентна
- В результате тормоза, падения, звонки!

# Как сделать чтоб не падало?

---

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

# Как сделать чтоб не падало?

---

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

# Как сделать чтоб не падало?

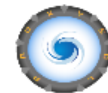
---

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

# Как сделать чтоб не падало?

---

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





# Как сделать чтоб не падало?

---

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

# Плохой запрос

---

- Переписать на лету
- Закешировать
- Отправить на реплику

# Как так на реплику-то?!

---

- Реплики не синхронны
  - Можно что б синхронны, но дороже
- Но есть GTID
  - Мы ТОЧНО знаем с какой реплики можно читать

# Как «переписать»?

---

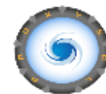
- Посмотреть уникальный digest

Из

```
SELECT * FROM ... ORDER BY RAND()
```

Сделать

```
SELECT * FROM ...  
WHERE a < RAND() ORDER BY a DESC  
LIMIT 10;
```



# Почему так можно?

---

- ProxySQL понимает MySQL протокол
- Понимает откуда, что и куда летит
- Полностью скрывает MySQL от приложения
  - Приложение работает только с ProxySQL
  - В том числе обрабатывая отказы

# Экономия на железе

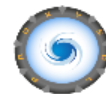
---

- Можем читать с реплик
- Можем отправлять запросы на нужную группу серверов
  - Или просто кешировать
- Можем вообще затротлить насмерть

# Можем закешировать

---

- Тупо на определенное время
  - Секунда, минута, час и т.д.
  - Запись не инвалидирует кеш!
- Когда положение безвыходное
  - Или для некритичных данных
- Временное решение



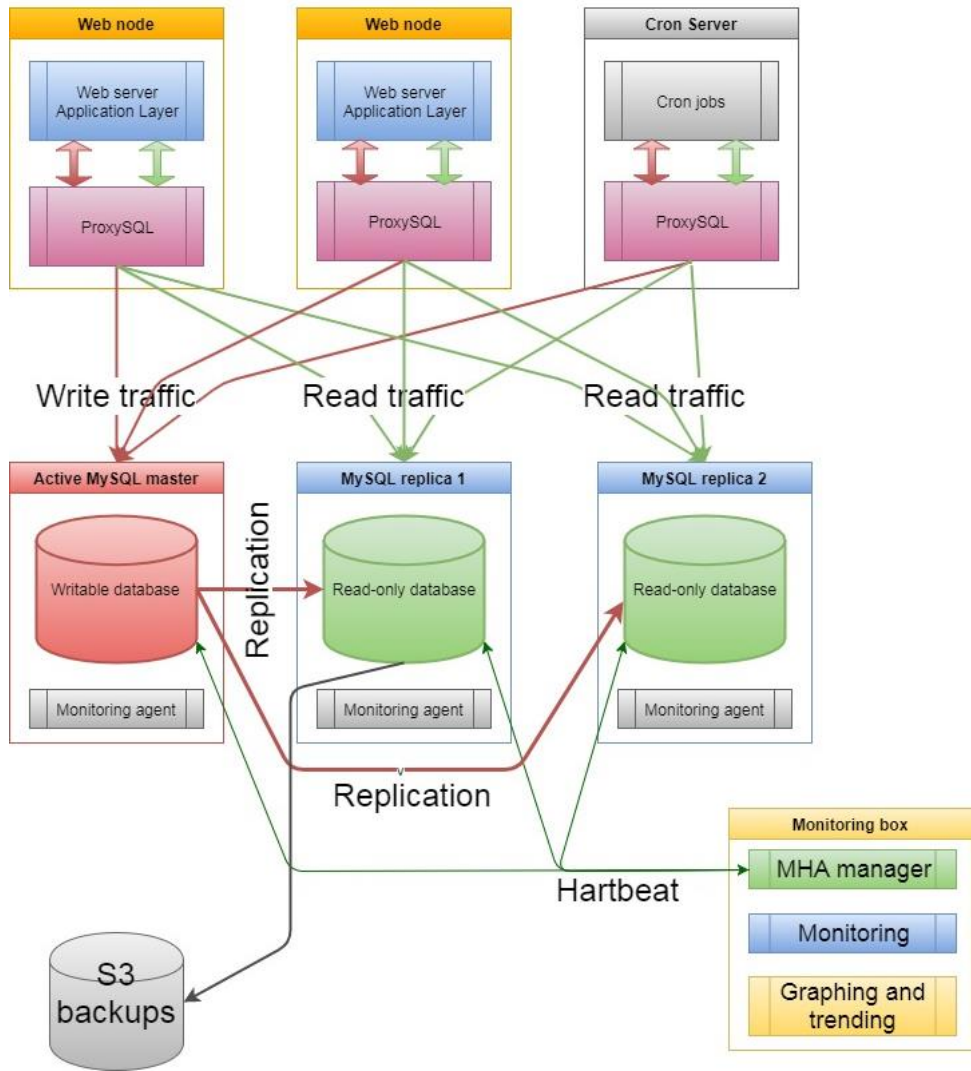
# Если MySQL упал

---

- Есть мониторинг и поддержка failover
  - Проверка отклика сервера
  - Проверка read\_only
  - Если сервер упал во время работы запроса, он будет перевыполнен
- Хорошо работает с МНА / Orchestrator
- В облаке тоже работает

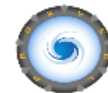
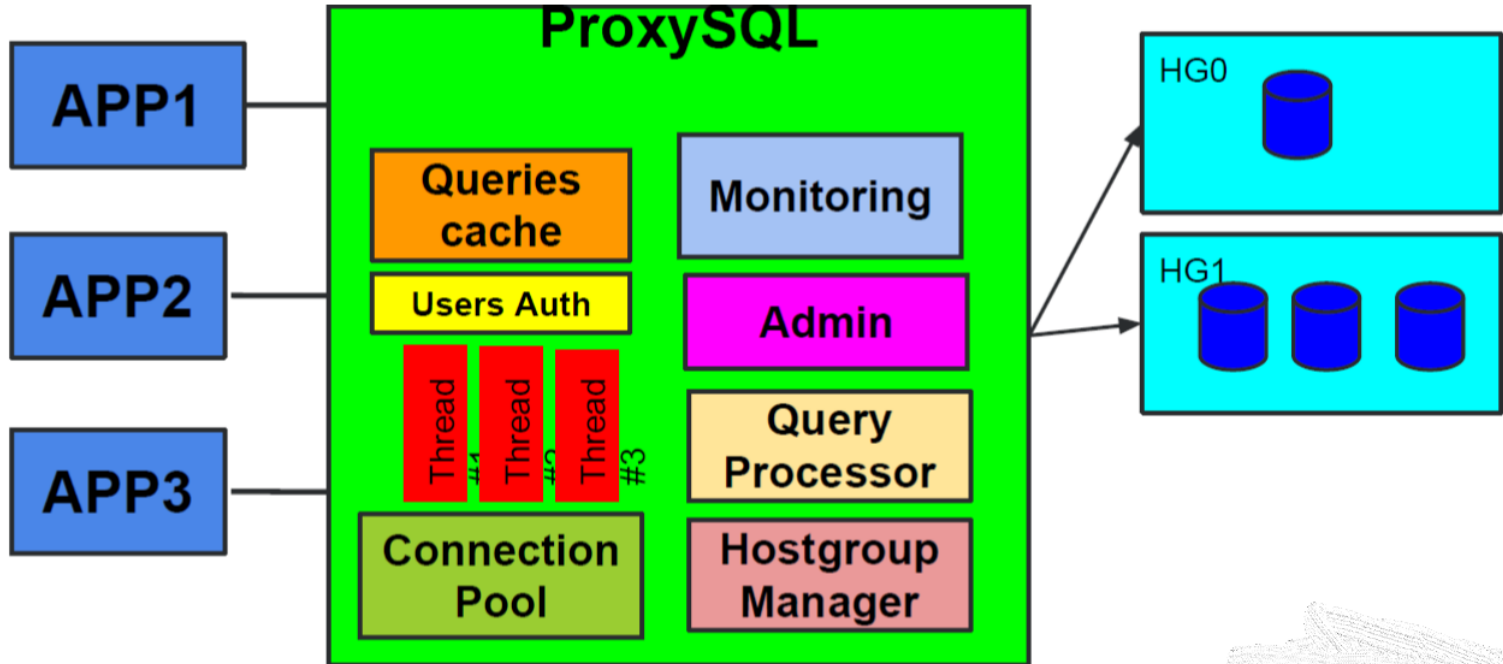


# Красивая диаграмма



О том как все могло бы быть

# ProxySQL изнутри



# Что мы теперь знаем?

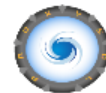
---

- Изменение запросов больше не требует срочного изменения кода
- Нагрузку на MySQL можно гибко распределять
- Если что-то упало - даунтайм можно сократить до секунд
- Какая-то часть ситуаций больше не будет мешать спать

# Где еще?

---

- Задать вопрос мне прямо сейчас
- Написать мне в вконтактИК ПОТОМ
  - [vk.com/vftimes](https://vk.com/vftimes)
- Рассказать про Ваш опыт с ProxySQL
  - на [vlad@proxysql.com](mailto:vlad@proxysql.com)
- Подписаться
  - [Facebook/proxysql](https://facebook.com/proxysql)
  - [twitter.com/proxysql](https://twitter.com/proxysql)



# ВОПРОСЫ!

# СПАСИБО!