

# Как Badoo перешли на PHP7 и сэкономили \$1M

Юрий Насретдинов, старший PHP-разработчик



# О компании

- 305 млн пользователей
- 50 языков интерфейса
- 1,5 Пб фотографий
- 4 ДЦ
- PHP7, MySQL, Memcache, Tarantool



# Схема доклада

- Архитектура
- HNVN
- Ядро RNP7 и расширения
- Тестирование
- Утилиты, код приложения
- «Запуск в бой» и результаты

4th Edition



# АРХИТЕКТУРА

**КАКАЯ ЕЩЕ АРХИТЕКТУРА?**

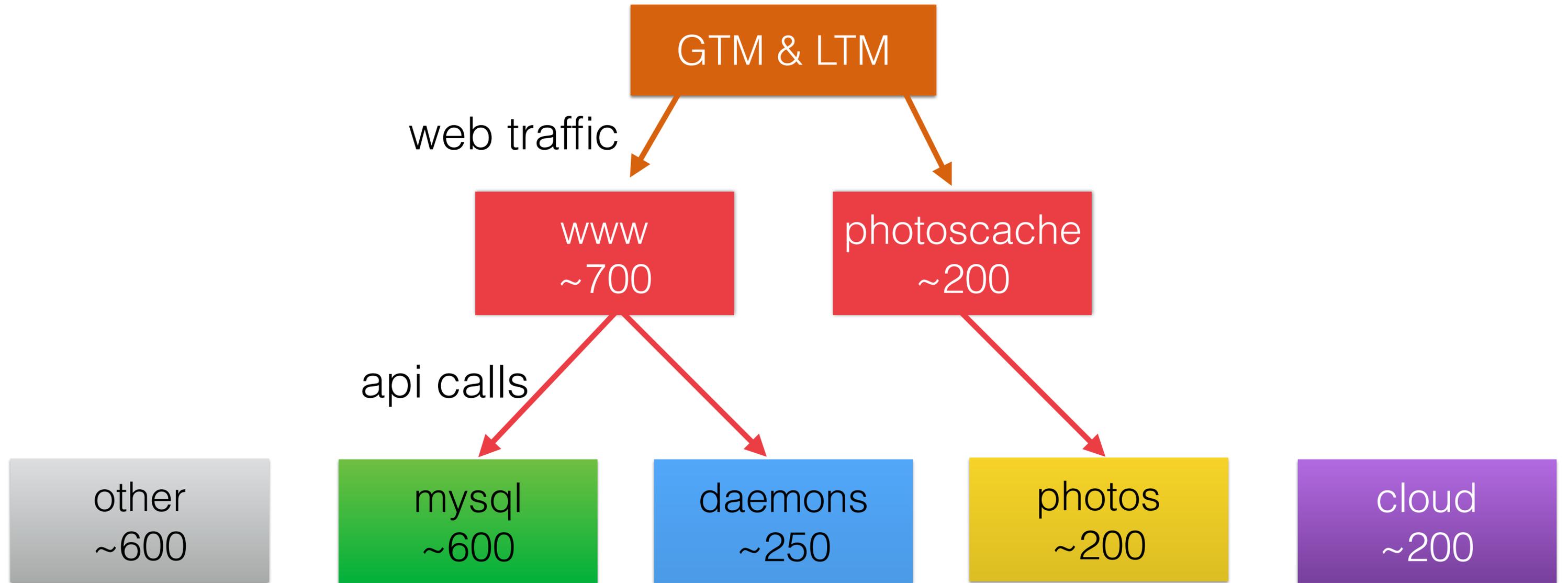
*все о трех буквах на которые вас послал  
проджект менеджер*

O'REILLY®

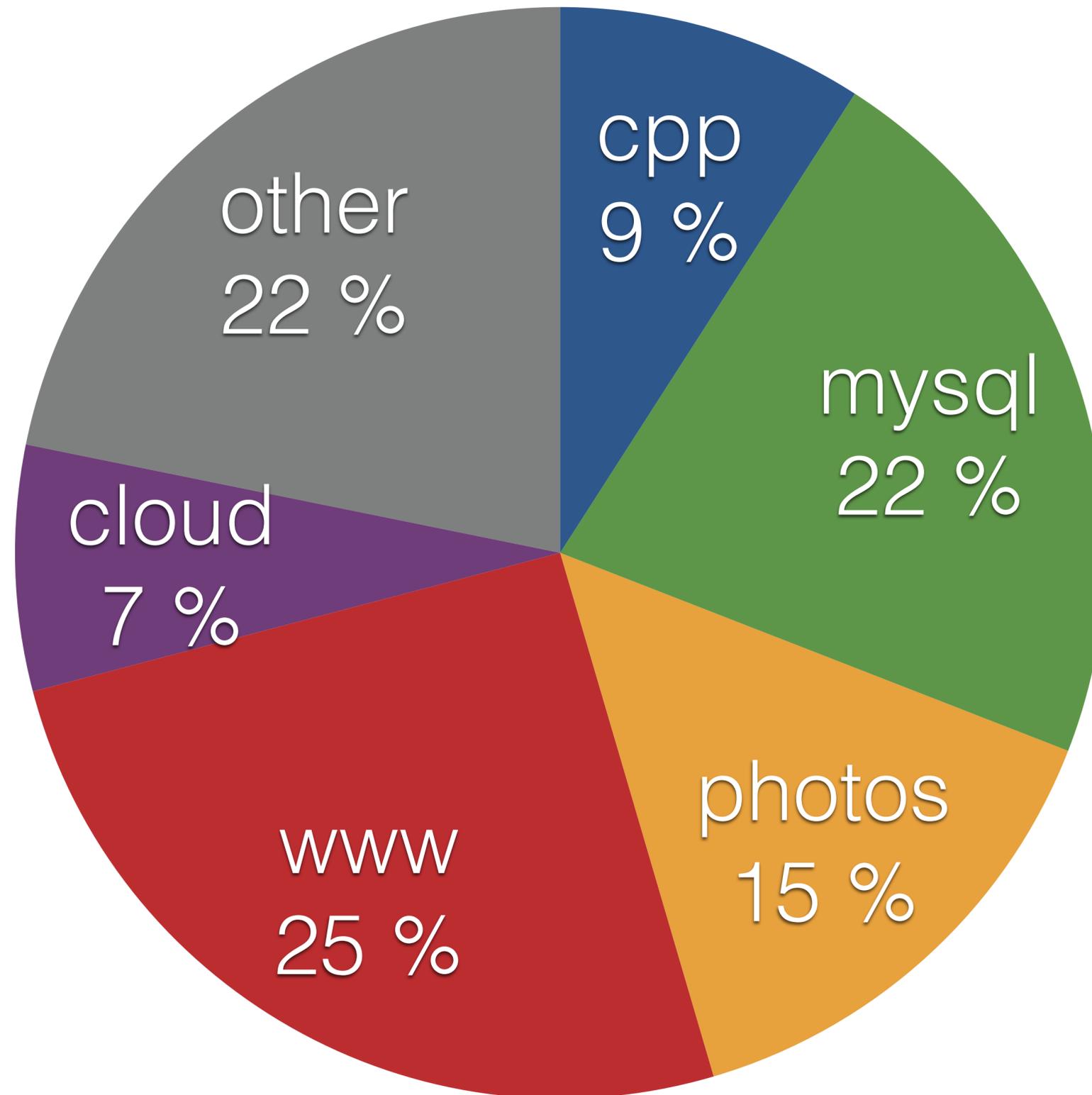
Василий Пупкин



# Архитектура



# Архитектура





HHVM

# HHVM

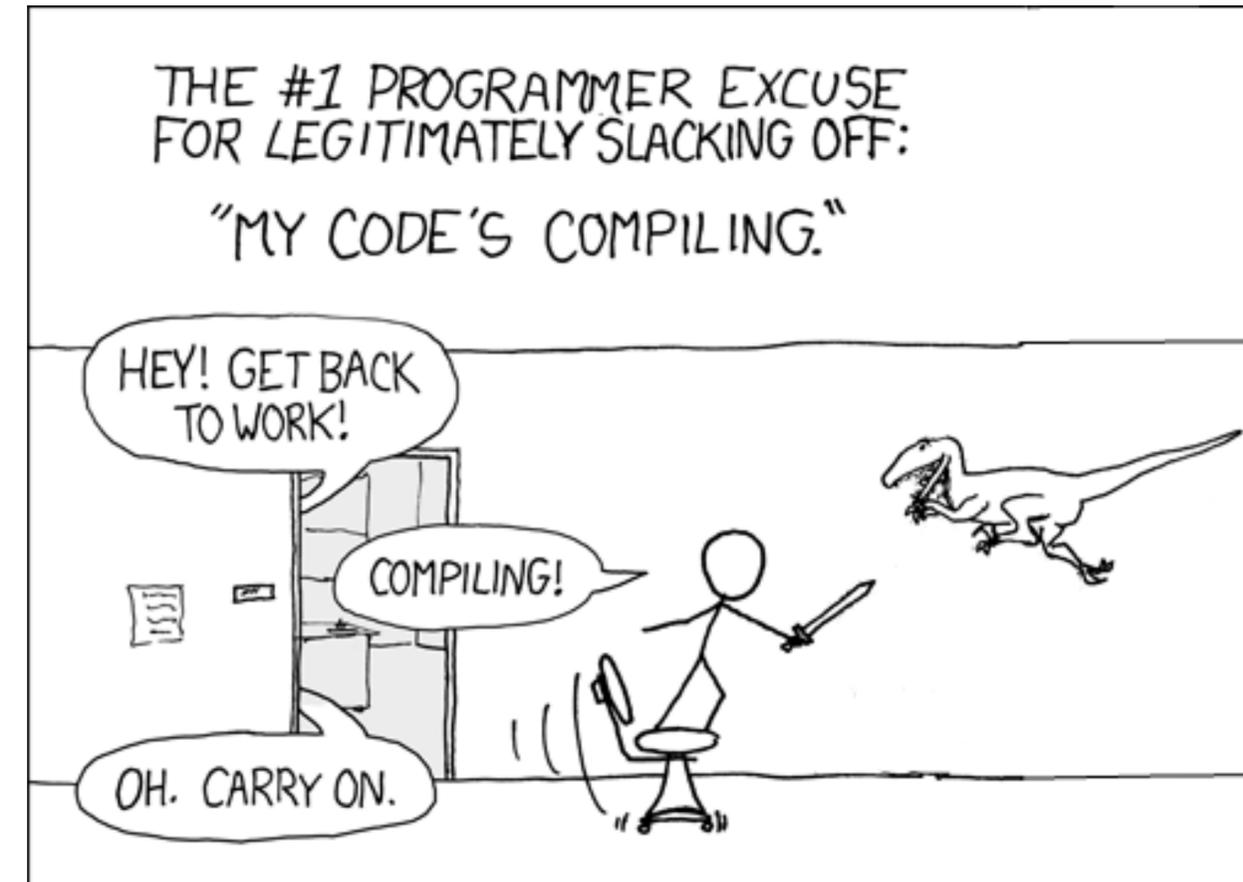
- Разработка Facebook
- Другая реализация интерпретатора PHP
- JIT-компиляция!
- Язык Hack (PHP с типами, в PHP7 что-то похожее)

# HHVM: плюсы

- Быстрый (4x RPS по сравнению с PHP 5.5)
- Почти полная совместимость с PHP 5.5
- Возможность деплоя одного .hhbc-файла
- Асинхронные mysql и memcache клиенты

# HHVM: МИНУСЫ

- Долго стартует в CLI-режиме
- Совсем другой внутренний API
- Нет поддержки runkit
- C++: долгая (пере)компиляция
- Нужно прогревать JIT



*php7*

# PHP7

- Новая версия PHP
- Радикальные переработки ядра
- Ускорение в среднем в 2 раза
- Переработка парсера (AST-деревья)
- Аннотации типов



# PHP7: ядро

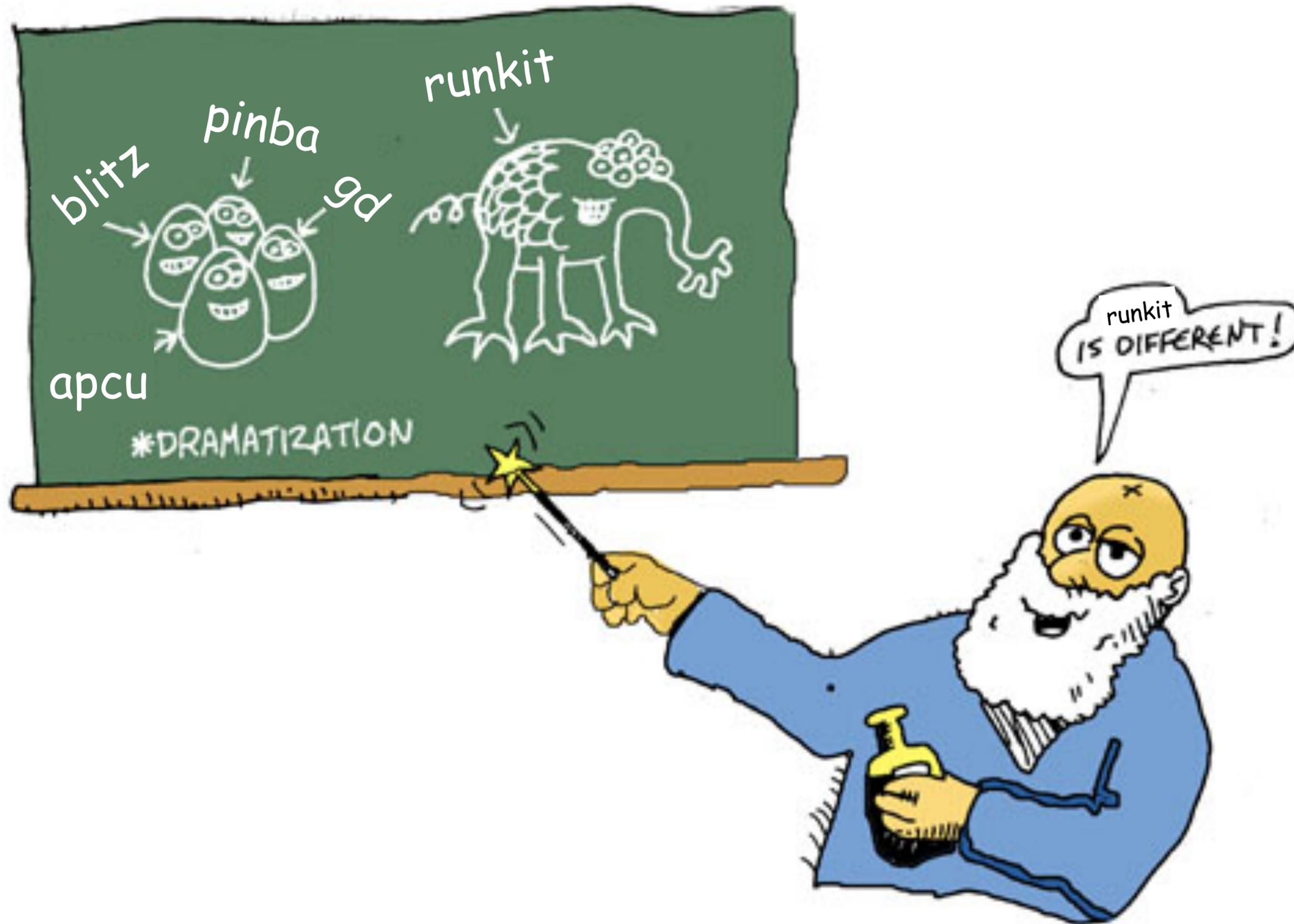
- `zval*` -> `zval`: на одно разыменование меньше
- `char*` -> `zend_string`: всегда известна длина строк
- Оптимизация `zval`
- Оптимизация парсинга параметров
- Изменение API работы с массивами (настоящие массивы, `zend_string` для ключей)
- <https://www.youtube.com/watch?v=LCIGhrci5V4>



# RNR7: расширения

- Использовали около 40 расширений
- Выкинули 11 из них
- Портировали свои: blitz, арси, pinba, несколько внутренних
- Для runkit создали альтернативу

# PHP7: расширения



# RНР7: проблемы с фронт

- Проверили все свои экстеншены
- Долго дебажили
- Правда, долго. Несколько недель
- Написали Стогову. Проблема оказалась в орсасхе
- Выкатились с версией RНР 7.0.4 и всё заработало



# Soft Mocs



# Тестирование

- 60 000 юнит-тестов
- Проходят за 1 минуту
- Активно использовался `runkit`
- Также есть функциональные, интеграционные, `calabash`, `selenium`-тесты



# Тестирование: PHP7 vs. runkit

- runkit не работает с PHP7
- говорят, корз починили 2 недели назад
- 2,5 месяца назад не было ничего
- Пришлось написать Soft Mocks

# Soft Mocks

- Решение на чистом PHP. Работает даже на HHVM\*
- На лету переписывает исходный текст, позволяя вставлять моки в любом месте (вызовы функций и методов, обращения к константам)
- Использует PHP-Parser
- <https://habrahabr.ru/company/badoo/blog/279617/>



\*мы не даем гарантий, что на HHVM всё работает идеально

# Soft Mocks

- 1,5 недели разработки
- 3 недели переписывания тестов
- нет проблем с производительностью, как у runkit
- тесты больше не дают segfault'ов\*, проходят стабильнее и быстрее
- всё работает под PHP5.5 и PHP7



\*кроме бесконечной рекурсии

# Переписывание кода

- mysql -> mysqli
- migration guide
- php7mar (<https://github.com/Alexia/php7mar>)
- func\_get\_args(), foreach by ref
- у нас были классы string, error, null
- конструктор теперь всегда \_\_construct()





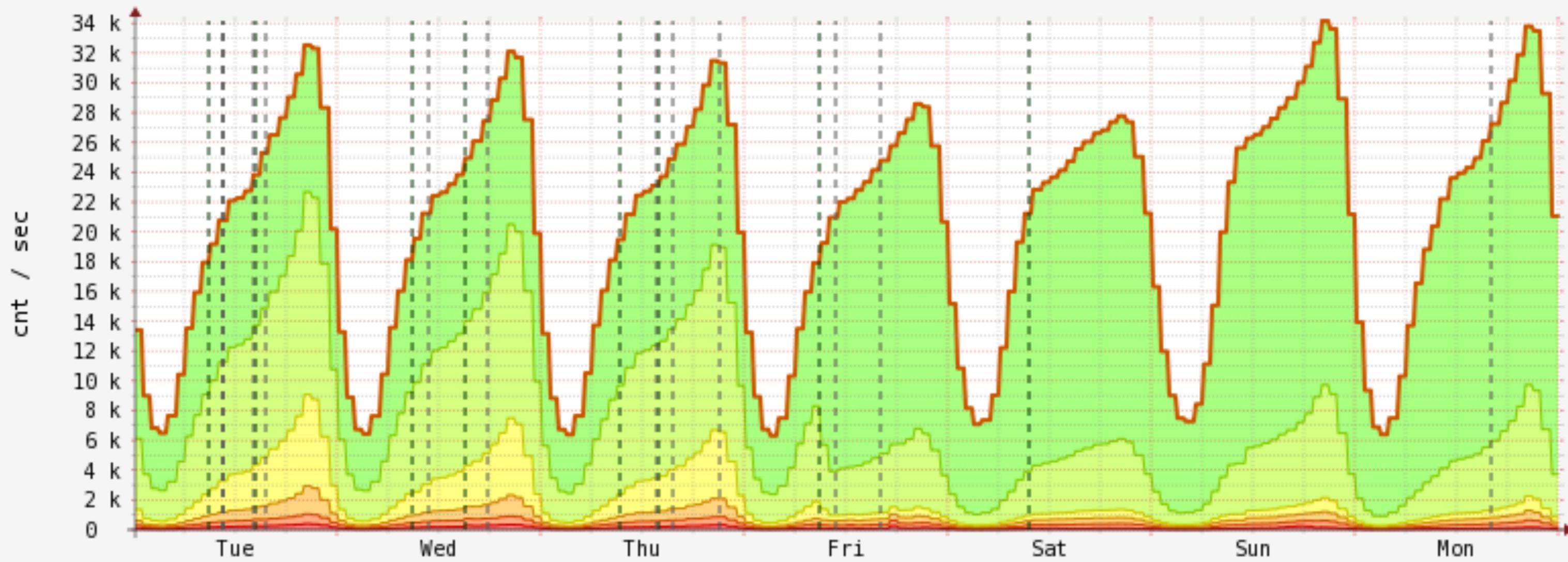
# Деплой

- selenium-тесты на стейджинге
- запуск на паре машин в каждом кластере
- запуск в «облаке» (cli)
- очень долгий дебаг для веб-кластера (fpm)
- апгрейд веб-кластера



# Pinba requests: BMA: Total requests [Prague] [NOT FLUSHED]

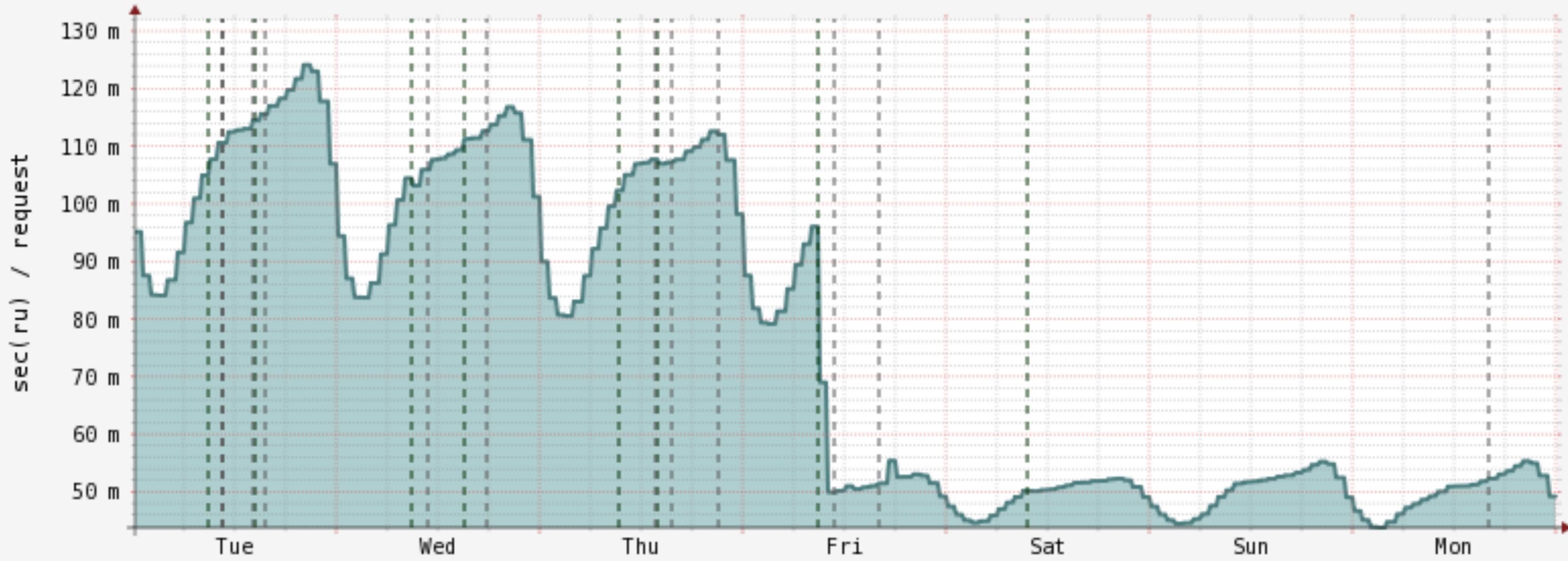
RRDTOOL / TOBI OETIKER



	last	first	avg	max
■ 1.0 - inf sec	= 0.32%	0.75%	0.57%	1.21%
■ 0.5 - 1.0 sec	= 0.87%	1.43%	1.32%	2.02%
■ 0.3 - 0.5 sec	= 1.19%	2.34%	2.09%	5.83%
■ 0.2 - 0.3 sec	= 1.47%	5.68%	5.25%	18.84%
■ 0.1 - 0.2 sec	= 11.05%	35.41%	25.32%	41.88%
■ 0.0 - 0.1 sec	= 85.11%	54.39%	65.44%	86.55%
■ All request	= 13544.3	13411.3	20376.0	34163.5



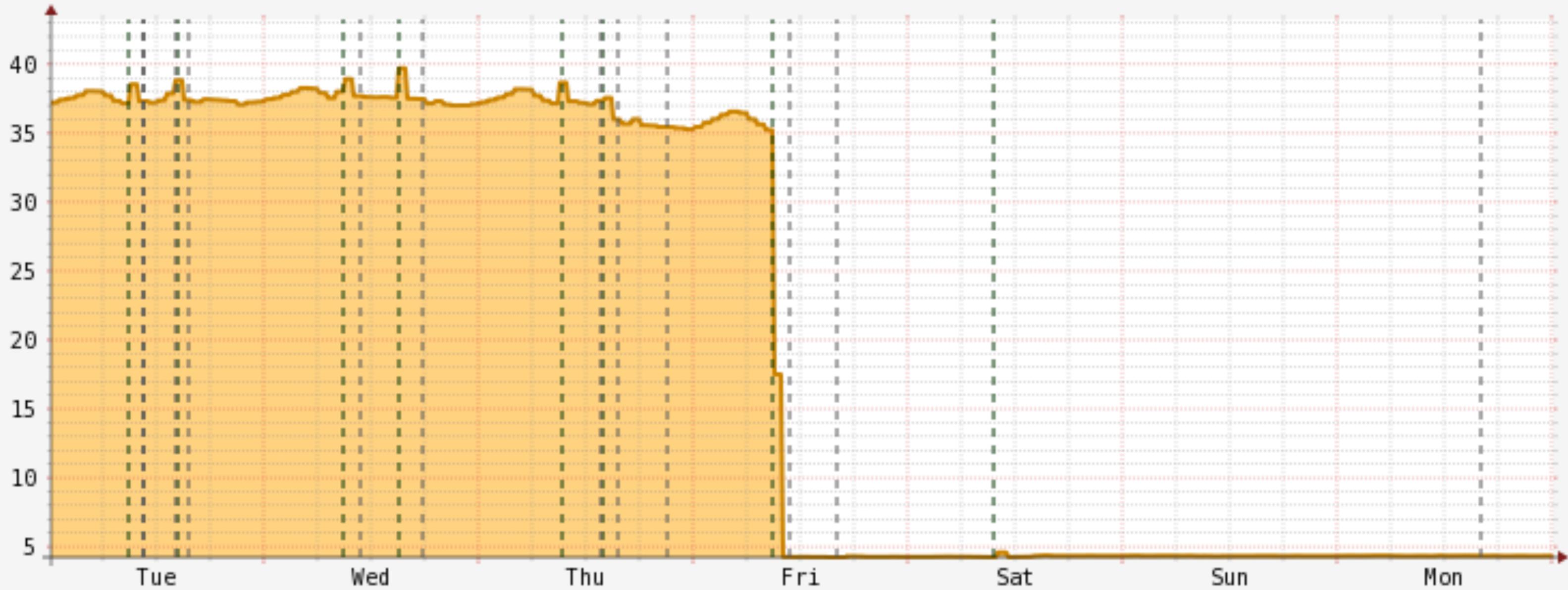
### Pinba requests: BMA: Average RUsage [Prague] [NOT FLUSHED]



	last	first	avg	max
ru_time	0.047	0.095	0.075	0.124



# Pinba requests: BMA: Average script memory [Prague] [NOT FLUSHED]



	last	first	avg	max
memory	4.35	37.22	20.22	39.71



# CPU Load WWBMA (mlan) [NOT FLUSHED]

RRDTOOL / TOBI OETIKER



		last	first	avg	max
■	amean	= 14.32	27.35	27.46	69.49
■	amean trend	= 14.32	27.35	27.46	69.49
■	median	= 14.34	27.40	27.45	69.45
■	median trend	= 14.34	27.40	27.45	69.45
■	percentile75	= 14.57	27.72	27.89	70.07
■	percentile95	= 14.93	28.43	28.59	71.47



# ЭКОНОМИЯ

- 700 серверов в FPM-кластере
- теперь нужно в 2 раза меньше
- каждый сервер ~\$4K
- $350 * 4K = \$1,4M$







**АНТОН**

17 марта 2016 в 18:55



0



Мне одному кажется, что вы сэкономили не 300 серверов, а 600?

Вы увеличили производительность вдвое, как если бы вы без оптимизации добавили столько же серверов, сколько у вас было.

То есть текущая производительность после оптимизации примерно равна двойной производительности до, то есть 600+600 серверов.

В итоге экономия в \$2 млн на железе и \$200к на хостинге.

# Вопросы

Юрий Насретдинов

[y.nasretdinov@corp.badoo.com](mailto:y.nasretdinov@corp.badoo.com)

