

# ОБЗОР РИТ 2014

Максим Мельниклв

# СОЧИ 2014

- ▶ сочи2014 с точки зрения it: с нуля, непонятно на чём, высокая нагрузка, несдвигаемые сроки, высокий уровень угроз ИБ
- ▶ если техническую платформу выбирают менеджеры, то они выбирают bitrix, беда-беда (и не думают о технологиях вообще)
- ▶ технологический тендер сочи2014 выиграл Microsoft с Azure, вот так вот
- ▶ сочи2014 194тыс посетителей - пик, 55 млн уникальных посетителей (Google Analytics)
- ▶ предыдущие цифры без учёта Китая, где с осени 2012 google analytics не работает
- ▶ сочи2014: 100% uptime, 4 мегабайта на главную и около 4 секунд на страницу

## email-рассылки от Badoo

- ▶ продуктовых готовых рецептов про email рассылки в общем-то нет, каждый проект уникален
- ▶ email-рассылка: нужно понимать нагрузку, какие ip-адреса и домены используются, не забудьте про: DKIM, SPF, PTR, DMARC!
- ▶ email-рассылка: самим, если большая техническая команда, полный контроль, и тд; готовое - если интеграция с CRM и/или малый бизнес
- ▶ грубая оценка, 5 миллионов в день с одного сервера; имейте второй сервер если у вас доход от потери писем больше \$2000
- ▶ хороший сервис рассылок обязан: доставлять за минуты, не должно быть проблем с доставкой, работает: bounce, FBL, ...
- ▶ статистика обязательна(sic!): процент открытий, кликов; размер очереди отправки; скорость доставки; число ошибок отправки
- ▶ микрофреймворк для писем - the must: подстановка параметров авторизации, проверка получателя — позволит просто расширять письма
- ▶ все письма в одном конфиге! ссылка на описание; приоритет письма; тип уведомлений: контакт "автора" письма

## заметки практекующих e-mail маркетологов, стратоплан

- ▶ как умным людям работать с другими умными людьми?! это между прочим сложно ...
- ▶ почему email-маркетинг?! мы не умеем пользоваться другими каналами; email обычно есть, it-шники читают письма каждый день, 1:1
- ▶ habrahabr даёт не плохой поток подписчиков-it-шников, если писать хорошо!
- ▶ после набора абоненской базы нужно сначала настроить доверительные отношение, иначе bullshit detector, сведёт эффект на ноль!
- ▶ психологи говорят: о хорошем человек говорит 5-7 знакомым, а о плохом 10-12
- ▶ email-маркетинг, конкретная полезная метрика - Open Rate писем; процент отписок; количество повторных продаж
- ▶ .oO андрогогика — теория обучения взрослых людей

## pulsedb

- ▶ база данных для хранения временных рядов
- ▶ проблема: хранение и обработка большого количества временных статистик, +графики
- ▶ pulsedb vs rrdtool, graphite, opentsdb, influxdb, ...
- ▶ админы любят то, что на C; программисты наоборот, любят то, что не на C
- ▶ о rrdtool: медленный (fork на замер), нельзя склеить метрики, примитивный
- ▶ о graphite: сложно склеивать метрики на лету, графики строит сам
- ▶ о opentsdb: hadoop, рисует графики, склеивает ряды, нет realtime-а, огромный overhead на данные
- ▶ о influxdb: go, развитие решение, SQL-like, агрегация рядов, умеет события
- ▶ pulsedb - библиотека/демон, замер: UTC+имя+теги, компактное неточное хранение, websocket-подписка
- ▶ 14bit на замер хватит всем! (с) Макс Лапшин
- ▶ pulsedb: HTTP Upgrade + текстовый протокол, собственный формат хранения, последняя минута в памяти
- ▶ pulsedb: 2 байта на замер, 14 бит на значение, 2 бита на порядок: 25 GB в

## schema-less в PostgreSQL9.4, Бартунов

- ▶ нашумевший доклад вживую
- ▶ сервисная архитектура изменила подход к СУБД, ACID -> BASE, масштабирование и производительность
- ▶ hstore в postgresql с 2003-го года, Key-Value, бинарное хранилище
- ▶ json в postgresql - только строка as is, это медленно и не эффективно
- ▶ а вот jsonb - это быстро и эффективно, и в самом ядре; а hstore и json - legacy
- ▶ text vs json vs jsonb: хранения совпадает, вставка незначительно медленнее, а выборка намного быстрее
- ▶ jsonb в postgresql в перфоманс тестах впечатляет! в зависимости от case-а частенькое эффективнее mongodb
- ▶ "хочется купить что-нибудь красненькое women oriented query: create index ... using vodka
- ▶ граница между rational и nosql решениями стирается
- ▶ postgresql9.4: jsonb, dynamic background workers, in-memory columnar storage, logical replication

## распределённые fs

- ▶ распределённые отказоустойчивые fs: glusterfs, ceph, parallels cloud storage
- ▶ wtf?!: объём, скорость, доступность, стоимость
- ▶ glusterfs: posix-compatible, no metadata server, heavy portable, nfs/cifs/native
- ▶ ceph — стабильность под большим вопросом, неочевидный процесс восстановления
- ▶ pcs (parallels cloud storage) в первую очередь ориентирован на хранение образа VM

## анатомия web-сервиса

- ▶ backend: принять соединение, аутентификация, авторизация, сессия...
- ▶ ..., распарсить url, rate limiting, бизнес-логика, кэширование, формирование ответа, шаблона, всё ...
- ▶ неблокирующий ввод/вывод: select => ready to read/write, do read/write until EAGAIN
- ▶ многозадачность (многопользовательность): процессы, нити(threads) на уровне ОС, event-loop, кооперативная многозадачность, реактор
- ▶ gevent - с точки зрения кода выглядит хорошо, и работает неплохо - переключение происходит именно тогда, когда надо
- ▶ ещё одна работа web-сервиса: общение с "db": request/response vs pipeling; N: одно vs pool, etc ...
- ▶ взаимодействие в web-сервисе: очереди, деление на компоненты, обращение к другим сервисам
- ▶ pubsub vs pubconsume, redis, pgq, rabbitmq, apache kafka и другие buzzwords на тему очередей...
- ▶ JavaScript - однопоточный. явная кооперативная многозадачность. ajax.



## Андрей Аксёнов: цена абстракции

- ▶ личный странный опыт - жжот, всё остальное деньги-на-ветер
- ▶ `vector<int>` vs `vector<int>+reverse` vs `int`

12

`+int*res+EOF` === 0 vs +20% vs +40%

- ▶ на самом деле `vector` разумеется тормозит, а `reverse(+=1000)` очень быстр, попали в кэш?!...
- ▶ ай, доклады Аксёнова на бесполезно конспектировать, это надо видеть
- ▶ CRC32 - очень старый, медленный, и вообще не хеш-функция, рулят: FNV, Jenkins, MurMur, etc... теоретически, но не практически!

## newsqI от echo

- ▶ comeback классических подходов
- ▶ sharding in app - дополнительная логика, больше серверов для поддержки, сложность поддержки
- ▶ архитектурное изменение: база данных уходит в облако, сервера становятся менее мощными и их количество возрастает
- ▶ Р в CAP вообще-то не дискретно, границу каждый себе может провести сам; NoSQL базы обычно акцентируются на А
- ▶ NewSQL - DBMS с масштабируемостью и гибкостью, которая поддерживает SQL и/или поддерживает ACID транзакции
- ▶ shared nothing - каждая нода независима, бесконечное масштабирование, но любой запрос во все ноды будет очень медленным
- ▶ In-memory storage: быстрый доступ, высокая пропускная способность, никакого буфер-management-a, нет проблема с блокировками и тд
- ▶ newsqI категории: новые решения: VoltDb, Clusterix, NuoDB; новый storage engine: TokuDB, scaleDB; прозрачная кластеризация: ...
- ▶ NewSQL это уже установившийся тренд: но поскольку решения сильно

# Как устроен NoSQL, Андрей Аксёнов

- ▶ строки == документы == объекты; колонки == поля == значения == ..., CSV, SQL, XML, JSON, WTF... а суть одна - документы и их части
- ▶ структуры данных: как хранятся данные (документы)? какой внутри формат строки? как хранятся индексы?
- ▶ NoSQL "revolution было: plain text, B-tree. Добавились: LSM, Bitcask, column-based. И алгоритмы компрессии: LZO, LZ4, snappy
- ▶ индексы: было B-tree, добавились: LSM+Bloom (псевдоиндекс по PK), Column Based... NB: вторичный - всегда btree(sic!)
- ▶ про мишуру, NoSQL "революция отсутствие схем, сплошной json, денормализация, шардинг, плюс мишура: rest/json синтаксис запросов
- ▶ Не забудь почитать в wikipedia про: B-Tree, LSM, Bloom
- ▶ а так же о Bitcask, column based, pfor

# Tarantool

- ▶ Tarantool: redis + node.js для lua, все данные в одной процессе, кооперативная многозадачность, вторичные и multipart ключи
- ▶ tarantool 1.6 - база данных встроенная в интерпритатор lua - как drop-in replacement
- ▶ так или иначе, если убрать в сторону весь хайп вокруг json, всё более популярным становится msgpack
- ▶ Tarantool + LuaJit = хранимые процедуры на языке общего назначения со скоростью C, интеграция с C => большое ускорение
- ▶ tarantool 1.6 - мастер-мастер асинхронная репликация(но синхронизация конфликтов на стороне app) - проще делать failover

## биллинг в Badoo

- ▶ badoo - нужно понимать платежи со всего мира, монетизация - микроплатежи, импульсивные платежи, из-за объёма важен каждый %
- ▶ в Badoo - отдел "биллинга 15 человек, каждый может сравнить со своей компанией, я со своей сравнил - очень похоже :) ...
- ▶ биллинг в badoo - отдельный сервис с API - HTTP/JSON для внешних и внутренних клиентов, интеграция с агрегаторами...
- ▶ любой большой проект интегрируется с кучей партнёров - the must, ибо риски, ибо комиссия, разная реализация, ...
- ▶ совет для маленьких которые стремятся в большие: в каждой стране самый популярный метод оплаты - свой!
- ▶ MCC (Merchant Category Code) оказывается очень важная штука для повышения процента успешных платежей
- ▶ в США после включения 3d secure платежи упали в 0.
- ▶ от отключения 3d secure в РФ число успешных платежей выросло с 70 до 85%
- ▶ работа с локальным эквайером: +20% сразу во многих странах
- ▶ у Badoo платежей с Android-ов больше чем с iOS. Google позволяет не

# SOA от Wargaming

- ▶ мы стали это делать ещё до этого как подход сдох и снова стал быть модным :) ...

## vkontakte в Open Source

- ▶ KPHP; набор NoSQL движков: pmemcached, texts, lists, hints и т.д; набор утилит db-proxy, mc-proxy, antispam, cache
- ▶ KPHP - очень урезанный php, в 4 раза HHVM, в 25 раз PHP, +статический анализ кода; но -oop и отсутствие части стандартный lib
- ▶ NoSQL от VK - снимки и бинлоги, memcache протокол, rpc протокол, масштабируемость, скорость; -rdbms, только Linux
- ▶ pmemcached: KV + высокая скорость работы + набор ключей по префиксов + метафайлы
- ▶ lists - движок для списков, в качестве значения int и 256 символов текста
- ▶ Search - некоторая альтернатива sphinx: произвольные теги для выборки, произвольные поля для сортировки. Search-X и Search-Y - моды
- ▶ Storage - хранение медиа-контента, быстрее чем в отдельных файлах, работа через MC и RPC протоколы, администрирование+миграция
- ▶ hints - быстрый поиск по префиксам слов, поиск по объектам, сортировка объектов по рейтингу
- ▶ Queue - позволяет организовать realtime общения между браузером и

# СПАСИБО ЗА ВНИМАНИЕ. ВОПРОСЫ

Максим Мельников

[mailto:m\\_melnikau@wargaming.net](mailto:m_melnikau@wargaming.net)

<https://plus.google.com/+MaksimMelnikau>

[https://twitter.com/max\\_posedon](https://twitter.com/max_posedon)

<http://wargaming.com>



**WARGAMING.NET**  
LET'S BATTLE