

Дослідження та аналіз характеристик сервісів соціальної мережі «В контакте»

Юшкевич О.Е.

ІК-51м

- ▶ «ВКонтакте» — социальная сеть, принадлежащая Mail.Ru Group. Располагается по адресу vk.com.
- ▶ По данным SimilarWeb, «ВКонтакте» является первым по популярности сайтом в России и на Украине, 4-м — в мире.
- ▶ По данным Alexa Internet, третий по популярности сайт в России и на Украине, второй — в Белорусии, 21-й — в мире.
- ▶ Запущенный 10 октября 2006 года, ресурс изначально позиционировал себя в качестве социальной сети студентов и выпускников российских вузов, позднее стал называть себя «современным, быстрым и эстетичным способом общения в сети».
- ▶ В январе 2014 года ежедневная аудитория «ВКонтакте» составляла около 60 миллионов человек, а в январе 2015 года — 70 миллионов человек в день. По данным на 15 июля 2015, среднесуточная аудитория составляет 64 525 950 посетителей.

Статистика ВКонтакте

- ▶ 99,5 миллионов учетных записей.
- ▶ 40 миллионов активных пользователей во всем мире
- ▶ 11 миллиардов запросов в день.
- ▶ 200 миллионов личных сообщений в день.
- ▶ Видеопоток достигает 160Гбит/с.
- ▶ Более 10 тысяч серверов, из которых только 32 — фронтенды на nginx
(количество серверов с Apache неизвестно).
- ▶ 30-40 разработчиков, 2 дизайнера, 5 системных администраторов, многолюдей в датацентрах.
- ▶ Каждый день выходит из строя около 10 жестких дисков.

Платформа

- ▶ [Debian Linux](#) - основная операционная система
- ▶ [nginx](#) - балансировка нагрузки
- ▶ [PHP](#) + [XCache](#)
- ▶ [Apache](#) + [mod_php](#)
- ▶ [memcached](#)
- ▶ [MySQL](#)
- ▶ Собственная СУБД на [C](#)
- ▶ [node.js](#) - прослойка для реализации XMPP, живет за [HAProxy](#)
- ▶ Изображения отдаются просто с файловой системы [xf](#)
- ▶ [ffmpeg](#) - конвертирование видео

Общие принципы

Сервера многофункциональны и используются одновременно в нескольких ролях:

- ▶ Перебрасывание полуавтоматическое
- ▶ Требуется перезапускать даемон'ы
- ▶ Генерация страниц с новостями (микроблоги) происходит очень похожим образом с [Facebook](#), основное отличие - использование собственной СУБД вместо MySQL
- ▶ При балансировке нагрузки используются:
- ▶ Взвешенный round robin внутри системы
- ▶ Разные сервера для разных типов запросов
- ▶ Балансировка на уровне DNS на 32 IP-адреса

Общие принципы

Большая часть внутреннего софта написано самостоятельно, в том числе:

- ▶ Собственная СУБД (см. ниже)
- ▶ Мониторинг с уведомлением по СМС
- ▶ Автоматическая система тестирования кода
- ▶ Анализаторы статистики и логов

Мощные сервера:

- ▶ 8-ядерные процессоры Intel (по два на сервер, видимо)
- ▶ 64Гб оперативной памяти
- ▶ 8 жестких дисков (соответственно скорее всего корпуса 2-3U)
- ▶ RAID не используется
- ▶ Не брендированные

Общие принципы

Вычислительные мощности серверов используются менее, чем на 20%

Сейчас проект расположен в 4 датацентрах в Санкт-Петербурге и Москве, причем:

- ▶ Вся основная база данных располагается в одном датацентре в Санкт-Петербурге
- ▶ В Московских датацентрах только аудио и видео
- ▶ В планах сделать репликацию базы данных в другой датацентр в ленинградской области
- ▶ CDN на данный момент не используется, но в планах есть
- ▶ Резервное копирование данных происходит ежедневно и инкрементально

Волшебная база данных на С

Этому продукту, пожалуй, уделялось максимум внимания аудитории, но при этом почти никаких подробностей о том, что он собственно говоря собой представляет, так и не было обнародовано. Известно, что:

Разработчики:

- ▶ Андрей Лопатин
- ▶ Николай Дуров
- ▶ Арсений Смирнов
- ▶ Алексей Левин

Используется в огромном количестве сервисов:

- ▶ Личные сообщения
- ▶ Сообщения на стенах
- ▶ Статусы
- ▶ Поиск, Приватность, Списки друзей

Волшебная база данных на C

- ▶ Нереляционная модель данных
- ▶ Большинство операций осуществляется в оперативной памяти
- ▶ Интерфейс доступа представляет собой расширенный протокол [memcached](#), специальным образом составленные ключи возвращают результаты сложных запросов (чаще всего специфичных для конкретного сервиса)
- ▶ Кластеризация осуществляется легко
- ▶ Есть репликация

Аудио и видео

- ▶ Эти подпроекты являются побочными для социальной сети, на них особо не фокусируются. В основном это связано с тем, что они редко коррелируют с основной целью использования социальной сети - *общением*, а также создают большое количество проблем: видео трафик - основная статья расходов проекта, плюс всем известные проблемы с нелегальным контентом и претензиями правообладателей.
- ▶ Медиа-файлы банятся по хэшу при удалении по просьбе правообладателей, но это неэффективно и планируется усовершенствовать этот механизм.
- ▶ 1000-1500 серверов используется для перекодирования видео, на них же оно и хранится.

XMPP

- ▶ XMPP (Extensible Messaging and Presence Protocol – расширяемый протокол обмена сообщениями и информацией о присутствии, ранее известный как Jabber^[1] (джа́ббер – «болтовня», «трёп») – открытый, основанный на XML, свободный для использования протокол для мгновенного обмена сообщениями и информацией о присутствии (см. список контактов) в режиме, близком к режиму реального времени.
- ▶ Изначально спроектированный легко расширяемым, протокол, помимо передачи текстовых сообщений, поддерживает передачу голоса, видео и файлов по сети.
- ▶ В отличие от коммерческих систем мгновенного обмена сообщениями, таких как AIM, ICQ, WLM и Yahoo, XMPP является децентрализованной, расширяемой и открытой системой. Любой желающий может открыть свой сервер мгновенного обмена сообщениями, зарегистрировать на нём пользователей и взаимодействовать с другими серверами XMPP. На основе протокола XMPP уже открыто множество частных и корпоративных серверов XMPP. Есть достаточно крупные проекты, такие как Google Talk, WhatsApp, Одноклассники.ru, LiveJournal, Juick и др.

XMPP

- ▶ Как известно, некоторое время назад появилась возможность общаться на Вконтакте через протокол Jabber (он же XMPP). Протокол совершенно открытый и существует масса opensource реализаций.
- ▶ По ряду причин, среди которых проблемы с интеграцией с остальными сервисами, было решено за месяц создать собственный сервер, представляющий собой прослойку между внутренними сервисами Вконтакте и реализацией XMPP протокола. Основные особенности этого сервиса:
- ▶ Реализован на [node.js](https://nodejs.org/) (выбор обусловлен тем, что JavaScript знают практически все разработчики проекта, а также хороший набор инструментов для реализации задачи)
- ▶ Работа с большими контакт-листами - у многих пользователей количество друзей на Вконтакте измеряется сотнями и тысячами
- ▶ Высокая активность смены статусов - люди появляются и исчезают из онлайн чаще, чем в других аналогичных ситуациях

XMPP

- ▶ Высокая активность смены статусов - люди появляются и исчезают из онлайн чаще, чем в других аналогичных ситуациях
- ▶ Аватарки передаются в base64
- ▶ Тесная интеграция с внутренней системой обмена личными сообщениями Вконтакте
- ▶ 60-80 тысяч человек онлайн, в пике - 150 тысяч
- ▶ [HAProxy](#) обрабатывает входящие соединения и используется для балансировки нагрузки и развертывания новых версий
- ▶ Данные хранятся в [MySQL](#) (думали о MongoDB, но передумали)
- ▶ Сервис работает на 5 серверах разной конфигурации, на каждом из них работает код на [node.js](#) (по 4 процесса на сервер), а на трех самых мощных - еще и [MySQL](#)
- ▶ Группы друзей в XMPP не связаны с группами друзей на сайте - сделано по просьбе пользователей, которые не хотели чтобы их друзья из-за плеча видели в какой группе они находятся

Интеграция со внешними ресурсами

Во Вконтакте считают данное направление очень перспективным и осуществляют массу связанной с этим работы. Основные предпринятые шаги:

- ▶ Максимальная кроссбраузерность для виджетов на основе библиотек easyXDM и fastXDM
- ▶ Кнопка "поделиться с друзьями", поддерживающая openGraph теги и автоматически подбирающая подходящую иллюстрацию (путем сравнение содержимых тега <title> и атрибутов alt у изображений, чуть ли не побуквенно)
- ▶ Возможность загрузки видео через сторонние видео-хостинги (YouTube, RuTube, Vimeo, и.т.д.), открыты к интеграции с другими

Интересные факты

Во Вконтакте считают данное направление очень перспективным и осуществляют массу связанной с этим работы. Основные предпринятые шаги:

- ▶ Ядро операционной системы модифицировано (на предмет работы с памятью), есть своя пакетная база для [Debian](#)
- ▶ Фотографии загружаются на два жестких диска одного сервера одновременно, после чего создается резервная копия на другом сервере
- ▶ Есть много доработок над [memcached](#), в.т.ч. для более стабильного и длительного размещения объектов в памяти; есть даже persistent версия
- ▶ Фотографии не удаляются для минимизации фрагментации

memcached

memcached – программное обеспечение, реализующее сервис кэширования данных в оперативной памяти на основе хэш-таблицы.

С помощью клиентской библиотеки (для C/C++, Ruby, Perl, PHP, Python, Java, .Net и др.) позволяет кэшировать данные в оперативной памяти множества доступных серверов.

Распределение реализуется путём сегментирования данных по значению хэша ключа по аналогии с сокетом хэш-таблицы.

Клиентская библиотека, используя ключ данных, вычисляет хэш и использует его для выбора соответствующего сервера.

Ситуация сбоя сервера трактуется как промах кэша, что позволяет повышать отказоустойчивость комплекса за счет наращивания количества memcached серверов и возможности производить их горячую замену.

memcached

- ▶ В API memcached есть только базовые функции: выбор сервера, установка и разрыв соединения, добавление, удаление, обновление и получение объекта, а также Compare-and-swap.
- ▶ Для каждого объекта устанавливается время жизни, от 1 секунды до бесконечности. При исчерпании памяти более старые объекты автоматически удаляются.
- ▶ Для PHP также есть уже готовые библиотеки PECL для работы с memcached, которые дают дополнительную функциональность.
- ▶ По умолчанию memcached использует порт 11211.

memcached

- ▶ Запрос к базе данных (без использования memcached)

```
function get_foo(int userid) {  
    result = db_select("SELECT * FROM users WHERE userid = ?", userid);  
    return result;  
}
```

- ▶ После введения использования memcached:

```
function get_foo(int userid) {  
    /* вначале проверить кэш */  
    data = memcached_fetch("userrow:" + userid);  
    if (!data) {  
        /* не найдено: запросить БД */  
        data = db_select("SELECT * FROM users WHERE userid = ?", userid);  
        /* сохранить в кэше для будущих запросов */  
        memcached_add("userrow:" + userid, data);  
    }  
    return data;  
}
```

nginx

- ▶ **nginx** (англ. engine x) (произносится как э́нжин-э́кс^[5] или э́нжин-й́кс^[6]) – веб-сервер и почтовый прокси-сервер, работающий на Unix-подобных операционных системах (тестировалась сборка и работа на FreeBSD, OpenBSD, Linux, Solaris, Mac OS X, AIX и HP-UX). Начиная с версии 0.7.52 появилась экспериментальная^[4] бинарная сборка под Microsoft Windows.
- ▶ Игорь Сысоев начал разработку в 2002-м году.

HTTP-сервер

- ▶ обслуживание статических запросов, индексных файлов, автоматическое создание списка файлов,
- ▶ акселерированное проксирование без кэширования, простое распределение нагрузки и отказоустойчивость
- ▶ поддержка кеширования при акселерированном проксировании и FastCGI
- ▶ акселерированная поддержка FastCGI и memcached серверов, простое распределение нагрузки и отказоустойчивость
- ▶ модульность, фильтры, в том числе сжатие (gzip), byte-ranges (докачка), chunked ответы, HTTP-аутентификация, SSI-фильтр
- ▶ несколько подзапросов на одной странице, обрабатываемые в SSI-фильтре через прокси или FastCGI, выполняются параллельно
- ▶ поддержка SSL
- ▶ поддержка PSGI, WSGI
- ▶ экспериментальная поддержка встроенного Perl

nginx

- ▶ Прокси-сервер (от англ. *proxy* — «представитель, уполномоченный») — сервер (комплекс программ) в компьютерных сетях, позволяющий клиентам выполнять косвенные запросы к другим сетевым службам. Сначала клиент подключается к прокси-серверу и запрашивает какой-либо ресурс (например, e-mail), расположенный на другом сервере. Затем прокси-сервер либо подключается к указанному серверу и получает ресурс у него, либо возвращает ресурс из собственного кэша (в случаях, если прокси имеет свой кэш). В некоторых случаях запрос клиента или ответ сервера может быть изменён прокси-сервером в определённых целях. Прокси-сервер позволяет защищать компьютер клиента от некоторых сетевых атак и помогает сохранять анонимность клиента.

HAProxy

- ▶ серверное программное обеспечение для обеспечения высокой доступности и балансировки нагрузки для TCP и HTTP-приложений, посредством распределения входящих запросов на несколько обслуживающих серверов.^[1] Программа написана на C.
- ▶ HAProxy используется в ряде высоконагруженных веб-сайтов, включая Twitter^[3], Instagram^[3], Github^[4], Stack Overflow,^[5] Reddit, Tumblr и OpsWorks product из Amazon Web Services^[6], W3C (W3C Validator)^[3], а также является составной частью облачной платформы Red Hat OpenShift^[7] и балансировщиком по умолчанию в облачной платформе OpenStack.
- ▶ HAProxy является программой с открытым исходным кодом и распространяется в соответствии с GNU General Public License (GNU GPL v2).

HAProxy

Возможности:

- ▶ Периодическая проверка доступности обслуживающих (back-end) серверов, на которые перенаправляются запросы пользователей;
- ▶ Несколько алгоритмов определения доступности сервера: tcp-check, http-check, mysql-check;
- ▶ Балансировка HTTP / HTTPS / TCP-запросов между «живыми» серверами;
- ▶ Возможность закрепления определенных клиентов за конкретными обслуживающими серверам (stick-tables);
- ▶ Поддержка: [IPv6](#) и UNIX sockets, HTTP/1.1 сжатие (deflate, gzip, libslz), [SSL](#)-шифрование, полная поддержка [постоянного HTTP-соединения](#);
- ▶ Поддержка переменных, цитирования блоков и [Lua](#)-скриптов в конфигурации сервера;
- ▶ [Веб-интерфейс](#) с актуальным состоянием и статистикой работы программы;