

# Доставка данных в реальном времени

Александр Бешкенадзе

# Структура

Для чего нам это нужно?

Прощаемся с Comet: анализ популярных решений, критика, поиски альтернативы.

Встречаем и знакомимся с WebSocket и Push: анализ поддержки браузерами, библиотеки, решение проблем.

Работа с Node.js и Redis: запуск и настройка, использование pub/sub, репликация.

Создаем готовое приложение.

# Для чего нам это нужно?

Вещание в реальном времени (текстовые трансляции)

Онлайн - Игры

Синхронизация действий пользователей

Оповещение пользователя о событии

# Прощаемся с Comet: анализ популярных решений, критика, поиски альтернативы.

APE (<http://www.ape-project.org>,  
<http://habr.ru/blogs/webdev/60803/>)

DKLab Realplexor ([http://dklab.ru/lib/dklab\\_realplexor/](http://dklab.ru/lib/dklab_realplexor/), [http:// habrahabr.ru/blogs/hi/79189](http://habrahabr.ru/blogs/hi/79189))

Протокол Bayeux  
(<http://svn.cometd.com/trunk/bayeux/bayeux.html>) .

Протокол BOSH  
(<http://xmpp.org/extensions/xep-0124.html>)

Lightstreamer ( [www.lightstreamer.com](http://www.lightstreamer.com))

# Прощаемся с Comet: анализ популярных решений, критика, поиски альтернативы.

Long polling — сервер не отвечает сразу на XHR-запрос, а по событию.

Forever frame — создает iframe, в который сервер постоянно дописывает события по мере их поступлений.

Script tags — динамически создаваемые JS-блоки, что позволяет кросс-доменные коммуникации.

ActiveXObject("htmlfile") — метод доступный в IE и использующий ActiveX

JSONRequest object ( [www.json.org/JSONRequest.html](http://www.json.org/JSONRequest.html) ) — реализует двухстороннее соединение, посредством двух одновременных запросов (передача/прием).

Прощаемся с Comet: анализ  
популярных решений, критика,  
поиски альтернативы.

Возможные задержки между  
событием и уведомлением.

Входящий трафик на сервер

Comet — один большой хак

# Прощаемся с Comet: анализ популярных решений, критика, поиски альтернативы

Web Socket как альтернатива Comet  
(Full-Duplex)

EventSource как альтернатива Comet (Push)

# Встречаем и знакомимся с WebSocket и Push: анализ поддержки браузерами, библиотеки, решение проблем

WebSocket — это стандарт HTML5 (<http://dev.w3.org/html5/websockets/>)

Server-Sent Events или EventSource — так же стандарт HTML5 (<http://dev.w3.org/html5/eventsource/>)



# Встречаем и знакомимся с WebSocket и Push: анализ поддержки браузерами, библиотеки, решение проблем

WebSocket

Неограниченное количество подключений на странице

Full-Duplex

Простой API

# Встречаем и знакомимся с WebSocket и Push: анализ поддержки браузерами, библиотеки, решение проблем

## Поддержка WebSocket

Google Chrome (начиная с версии 4.0.249.0)

Apple Safari (начиная с версии 5.0.7533.16)

Mobile Safari (iOS 4.0 beta 2, позднее убрана)

Mozilla Firefox (начиная с версии 3.7)

Opera (до конца года)

Internet Explorer (начиная с версии 9)

# Встречаем и знакомимся с WebSocket и Push: анализ поддержки браузерами, библиотеки, решение проблем

Поддержка EventSource

Google Chrome (начиная с версии 6)

Apple Safari (начиная с версии 5)

Opera (начиная с версии 9, через тег `<event-source/>`)

# Встречаем и знакомимся с WebSocket и Push: анализ поддержки браузерами, библиотеки, решение проблем

Проблема : Как запустить, там где нет поддержки WebSocket

Использовать библиотеку Socket.IO (<http://socket.io/>)

# Работа с Node.js и Redis: запуск и настройка, использование pub/sub, репликация

Node.js — событийно-ориентированный фреймворк на движке V8 JavaScript.

Redis — высокопроизводительное нереляционное распределённое хранилище данных.

# Работа с Node.js и Redis: запуск и настройка, использование pub/sub, репликация

## Установка

Node.js — `./configure && make && make install`

Redis — `make`

# Работа с Node.js и Redis: запуск и настройка, использование pub/sub, репликация

Node.js — запускается без файла настроек.

Redis — использует настройки по умолчанию или из файла (менее 30 строк).

# Работа с Node.js и Redis: запуск и настройка, использование pub/sub, репликация

Redis Publish/Subscribe

SUBSCRIBE канал1, канал2 - подписка

UNSUBSCRIBE канал1, канал2 (пусто - от всех) - отписка

PSUBSCRIBE шаблон\*, шаблон\* - подписка

PUNSUBSCRIBE шаблон\*, шаблон\* (пусто - от всех) - отписка

PUBLISH канал1 сообщение



# Работа с Node.js и Redis: запуск и настройка, использование pub/sub, репликация

## Репликация

Несколько slave-серверов, цепочки slave серверов

Master не блокируется

redis.conf: slaveof адрес порт

# Создаем готовое приложение

Чат (ClientSide) - 4 кб.

Чат (ServerSide) - 49 кб.

Размер приложение 53 кб.

# Создаем готовое приложение

```
$( "send" ).addEventListener( "click", function () {
    if (conn) conn.send($( "messge" ).value);
    return false;
}, true);
var conn;
window.onload = function () {
    if (window[ "WebSocket" ]) conn = new WebSocket( "ws://localhost:8000/
test" );
    else return;
    conn.onmessage = function (evt) {
        var msg = document.createElement( "p" );
        msg.innerHTML = JSON.parse(evt.data).message;
        $( "log" ).appendChild(msg);
    };
}

function $(id) {
    return document.getElementById(id);
}
```

# Создаем готовое приложение

Чат (ServerSide) - 4 кб.

Библиотека WebSocket (<http://github.com/miksago/node-websocket-server>) - 16 кб.

Библиотека Redis Node Client (<http://github.com/fictorial/redis-node-client>) - 33 кб.

# Создаем готовое приложение

```
var sys = require("sys"),
    ws = require('./lib/ws'),
    client = require("./lib/redis-client").createClient('6380'),
    client2 = require("./lib/redis-client").createClient('6380');
var server = ws.createServer(), conn = false;
server.addListener("connection", function (conn) {
  say(conn, " connected");
  conn.addListener("close", function () {
    say(conn, " disconnected");
  });
  conn.addListener("message", function (message) {
    say(conn, message);
  });
});
server.listen(8000, "localhost");
client.subscribeTo("chat", function (channel, message, subscriptionPattern) {
  if (server) server.broadcast(message);
});
function say(conn, message) {
  client2.publish("chat", JSON.stringify({
    message: "<" + conn._id + "> " + message
  }));
}
```

# Вопросы?

email/gtalk: [beskhenadze@gmail.com](mailto:beskhenadze@gmail.com)