

Node.js高并发聊天 服务实战

王文明

Github: [wangwenming](#)

2015-03-28

大纲

- ❖ 背景
- ❖ 技术选型&极简架构
- ❖ 性能优化工具

背景

产品:

- <http://www.haosou.com/s?q=exo&src=node>
- <http://t.haosou.com/chat/tantanmobile?topic=exo>
- <http://xiaohua.hao.360.cn/daojie> 点进去再滚到底部
- http://tatan.360.news.ifeng.com/news?nid=54d82f5ca15903ec558b4568#_zbs_360_tatan
- 世界杯期间 cntv 直播页面

数据:

- 同时在线约 100万
- 每秒接入连接数 5000
- 最大的一个Topic有10万人 (m)
- 每秒广播消息 20条 (n)
- 每秒广播消息数 200万条 ($m * n$)

技术选型

长连接

❖ Q1: 长连接、轮询?

- 举个例子: 微博PC版, 有2个:
 1. 30S一次的轮询(push_count)
 2. im的WebSocket

❖ Q2: 长连接方案

- Flash (双向通信)
- WebSocket (双向通信) [rfc6455](#)(2011.12), 手机上混合开发无法跨域
[faye-websocket@0.7.2](#) ([websocket-driver@0.5.3](#))
[6455在手机上的支持](#) Android>=4.4 Mobile Safari>=6.0
- Long Pull
 - xhr streaming (responseText 非一次返回, onreadystatechange, xhr.readyState=3) Chrome Firefox
 - Forever Frame(iframe / HTTP 1.1 Chunked Encoding) IE, 但是会一直Loading
 - HTMLFile (ActiveXObject对象)
 - JSONP, Old IE下要跨域
- 兼容性一栏
- “Even Faster Web Sites: Performance Best Practices for Web Developers”

技术选型

长连接框架

❖ <https://github.com/sockjs/sockjs-node>
Star: 900
@0.3.15

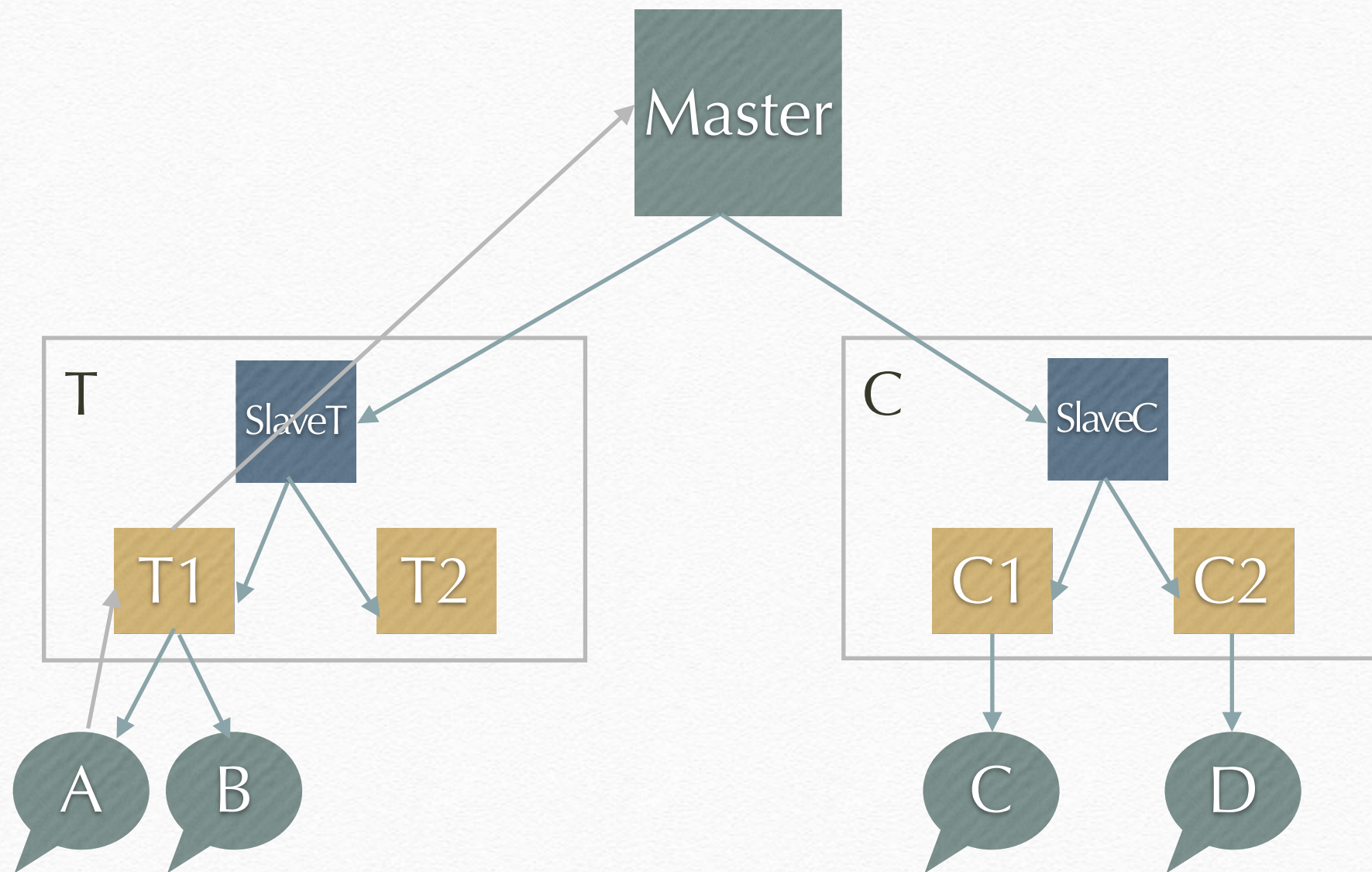
❖ [https://github.com/Automattic/socket.io/](https://github.com/Automattic/socket.io)
Star: 16, 134
@1.3.2

1. #438 2011-08-01提出，作者rauch在2014-08-19关闭了这个issue，也没有解决，有301个讨论
2. socket.io 的封装功能更多，包括Redis，消息的封装等
on('任意') emit('任意消息')，SockJS有且只有1个 on('message')
3. IP地址获取： x-forwarded-for sockjs/lib/transport.js:159

```
$ mocha test/echo-server.js
```


服务架构

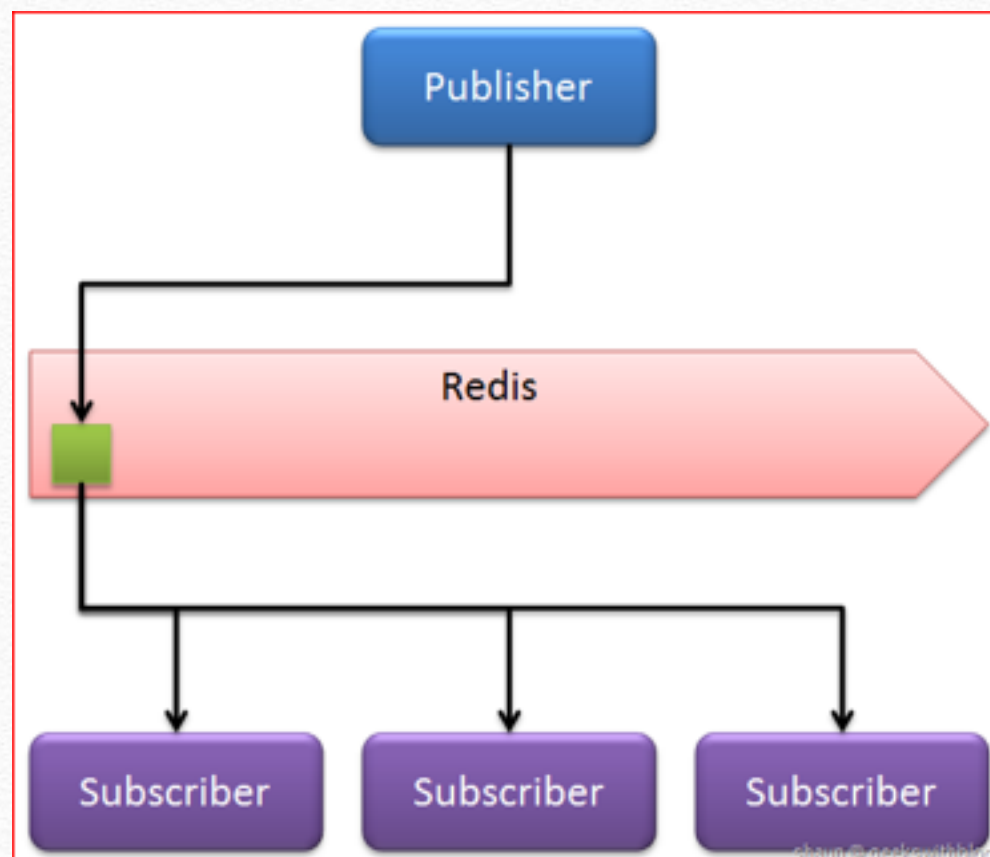
尽量简单的架构



服务架构

再简单一点

http://en.wikipedia.org/wiki/Publish_subscribe_pattern



RedisManager.js

hmdel
ping
retry

数据结构

Hash - hmget hdel

Set - sadd srem

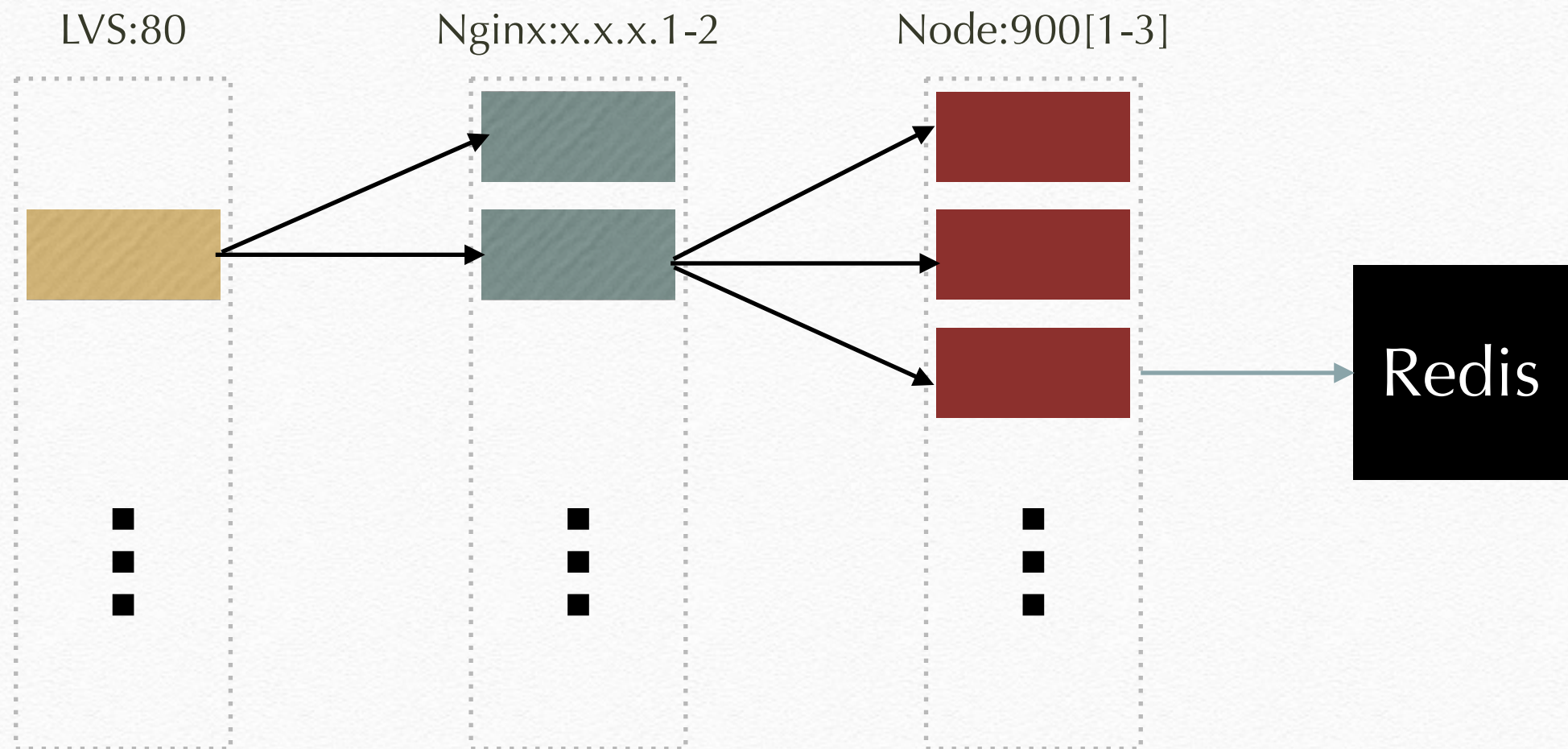
publish psubscribe

使用O(N)复杂度的操作

大数据操作使用cursor

服务架构

部署



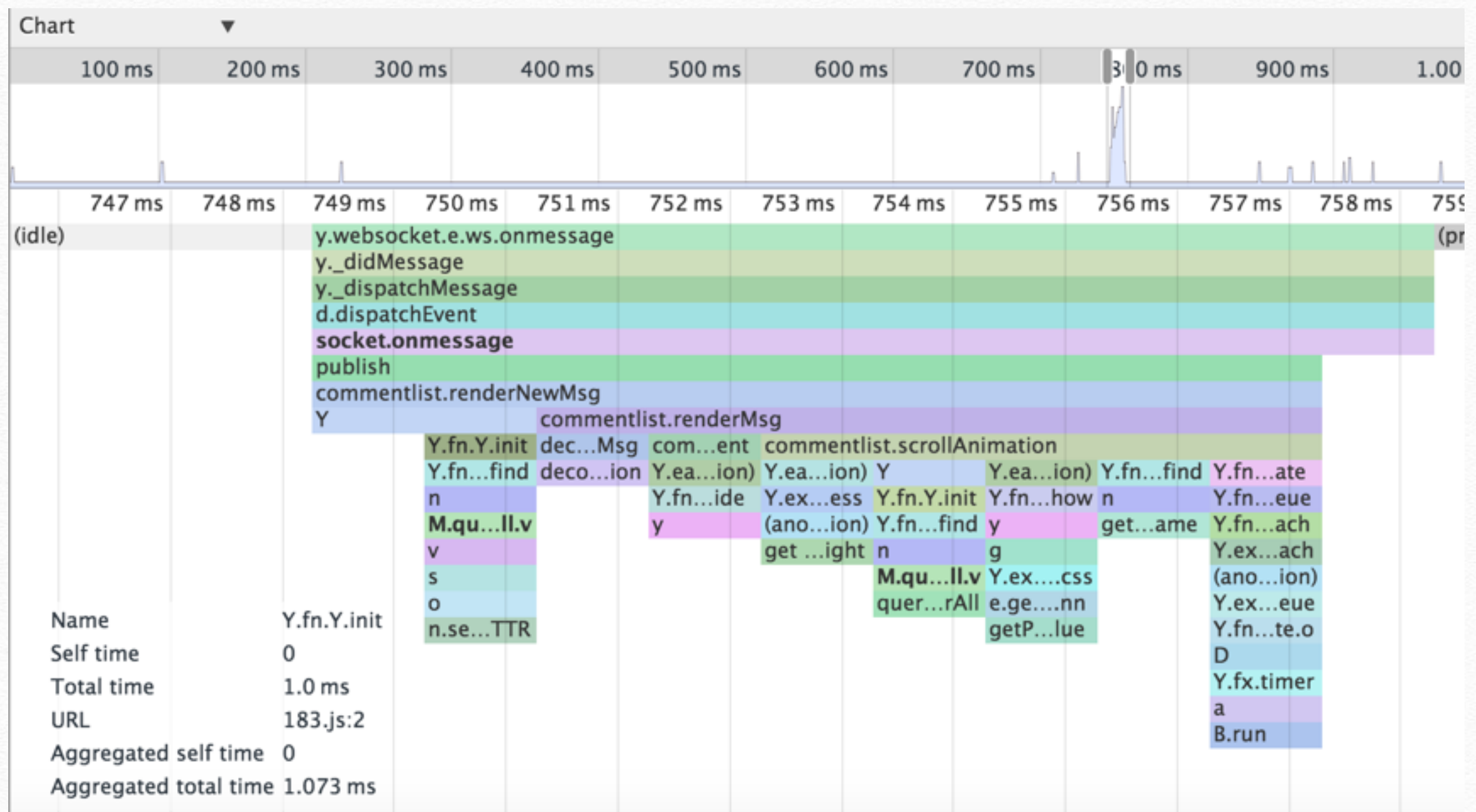

```
$ mocha test/chat-server.js
```

<http://localhost:8080/tantan/chat-client.html?ui>

```
> socket.post(new Date());  
> setInterval(function() {socket.post(new  
Date());}, 2000);
```


性能优化

Chrome DevTools



性能优化

v8

```
$ d8 --prof --log-timer-events script.js  
# 生成 v8.log  
$ mac-tick-processor v8.log  
$ plot-timer-events v8.log  
# 需要 Linux with gnuplot 4.6 来绘图
```

```
$ npm install -g tick  
$ node-tick-processor v8.log
```

http://v8.googlecode.com/svn/branches/bleeding_edge/tools/profviz/profviz.html

性能优化

v8-profiler

```
$ npm install v8-profiler
```

```
$ npm install -g node-inspector
```

```
$ node --prof --prof_lazy --log index.js 9011 /profile
```

```
# 会立即生成一个没有真实数据的 v8.log
```

```
$ node-debug --web-port 8888 index.js 9011 /profile
```

语句上的优化:

<http://s3.mrale.ph/nodecamp.eu/#12>

性能优化

压缩数据包

- ❖ JSON是比较冗余的，可从文本上压缩

```
$ mocha test/unserialize.js
```

```
匿名压缩比: 30.8%, 253B ==> 78B
```

```
登录压缩比: 40.3%, 258B ==> 104B
```

- ❖ GZIP压缩

Gzip有收益的建议最小值是150-1000字节，150字节左右压缩了反而会变大

- ❖ 如果不考虑旧浏览器，可以直接使用ArrayBuffer

- ❖ Redis操作: $O(m,n) \Rightarrow O(n)$

- ❖ strace(DTrace) ->

```
$ sudo dtruss -cnp 29663
```

strace发现80%操作在 epoll_wait (只有1个线程)