NCHAN 高效实现 WEB 实时通信

薄荷 Vincent

2017.3.7

关于我

```
name = 'Vincent'
real_name = '谢文威'

company = '薄荷科技 - 健康领域领先的移动互联网公司'
title = '联合创始人, CTO, Rubyist'

blog = 'https://xiewenwei.github.io'
github = 'https://github.com/xiewenwei'
ruby_china = 'https://ruby-china.org/vincent'

while true
  write_programs
  read_books
  watch_movies
  enjoy
end
```

提纲

- 即时通信 IM 需求
- 各种方案对比
- Nginx Nchan 介绍
- Nginx Nchan 演示
- 实践中的问题
- 总结 & QA

即时通信 IM 需求

- 薄荷的顾问和用户通过 IM 交流完成服务
- IM Client 端支持 iOS，Android 和 Web
- 消息类型丰富，包括文本，图片，商品，视频和红包
- 服务端存储所有消息，以便分析，监控和审计

可选择的方案

Web 实时通信方案很多，大体分成三类

- 基于 Http 连接
- 第三方 Socket 服务
- Websocket

方案1：HTTP 轮询

- 定期请求服务器刷新消息，伪实时
- 优点：简单
- 缺点：效率太低，体验不好

方案2：第三方IM服务

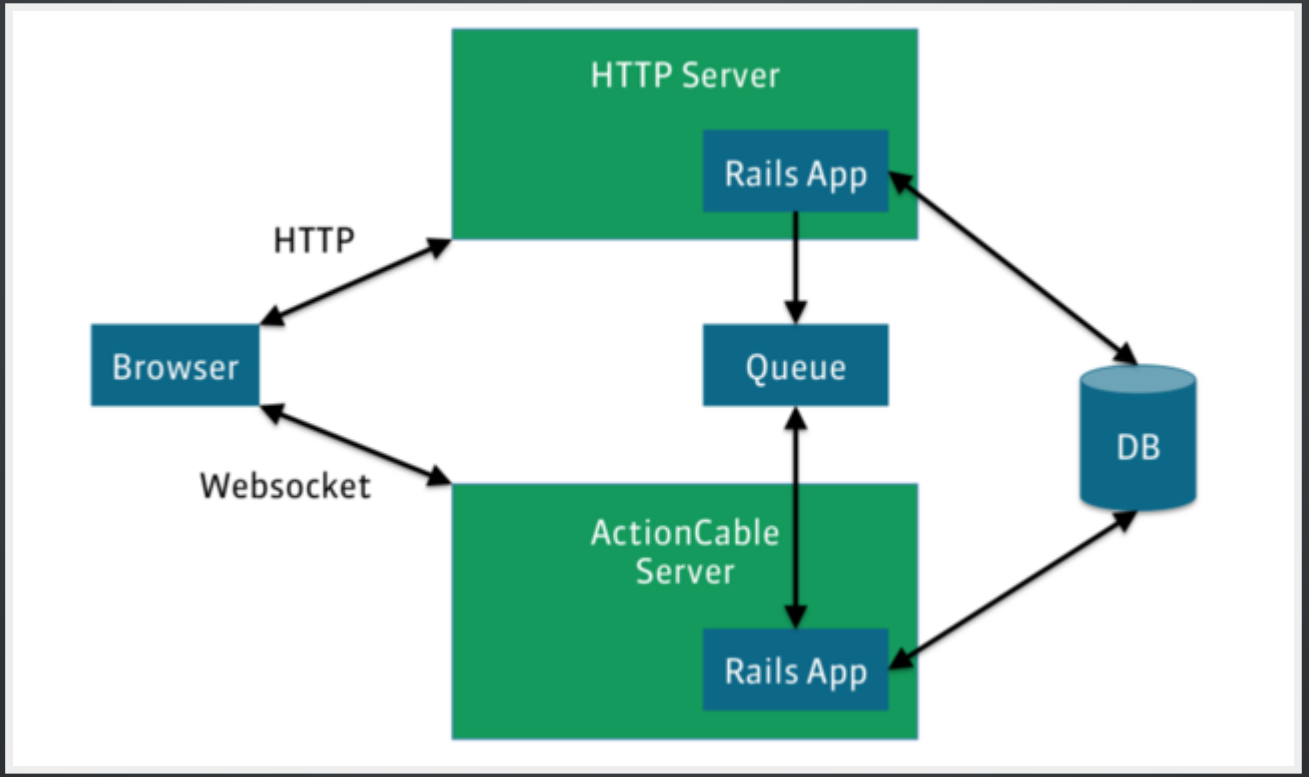
- 使用第三方服务，如环信、融云等
- 优点：简单，有很多现成功能
- 缺点：集成复杂，不够灵活，收费

方案3： WEBSOCKET - ACTIONCABLE

- Rails 5 引入的重量级特性，Rails 推荐的实时通信方案
- 优点：与 Rails 良好集成
- 缺点：复杂，成熟度低，并发能力差，性能低

ACTIONCABLE 结构

ActionCable 自建 websocket server，并发和性能受限于 ruby



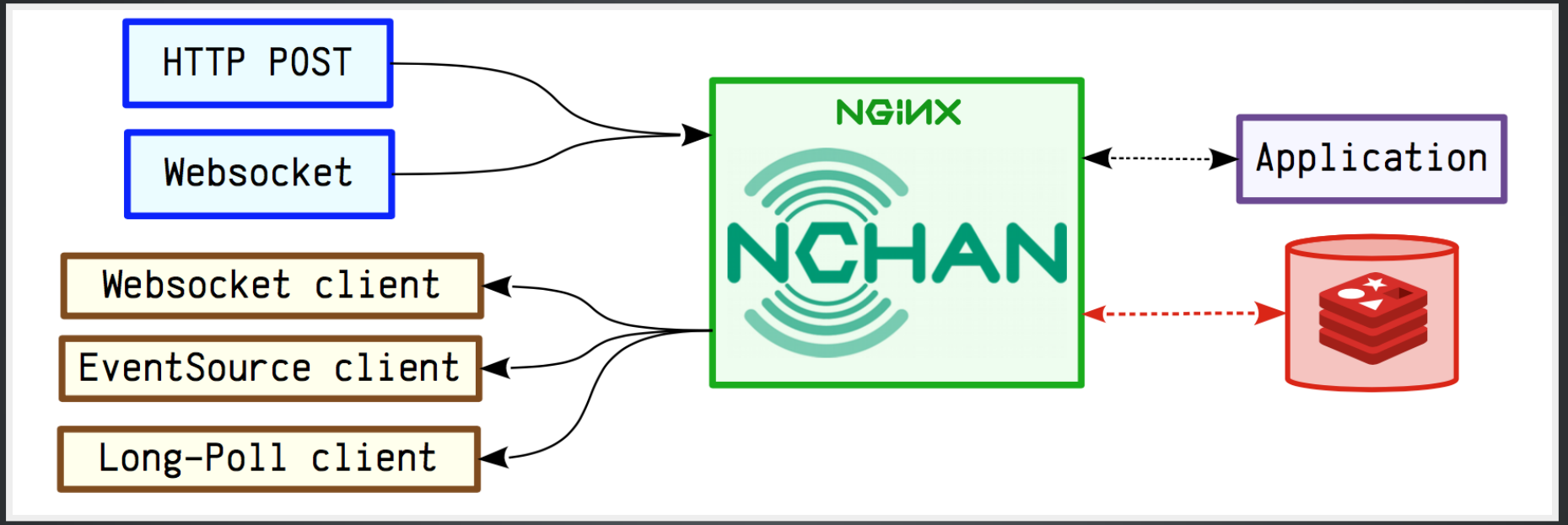
方案4： WEBSOCKET - NGINX NCHAN

- 基于 Nginx 的 订阅/发布 pub/sub 服务器
- 优点：简单，并发能力强，性能好，跨语言
- 缺点：成熟度低

NGINX NCHAN 简介

- 一个消息订阅/发布 pub/sub 服务器
- 主要用于 Web 实时应用，开源免费
- 基于 Nginx，是 Nginx 一个模块
- 灵活，高性能，高并发，可伸缩，易于使用

NCHAN 的结构示意图



nchan 作为中间层，避免了 application 和 client 直接建立 websocket 连接

application 通过 http pub 消息给 nchan，再由 nchan 传递 client 端

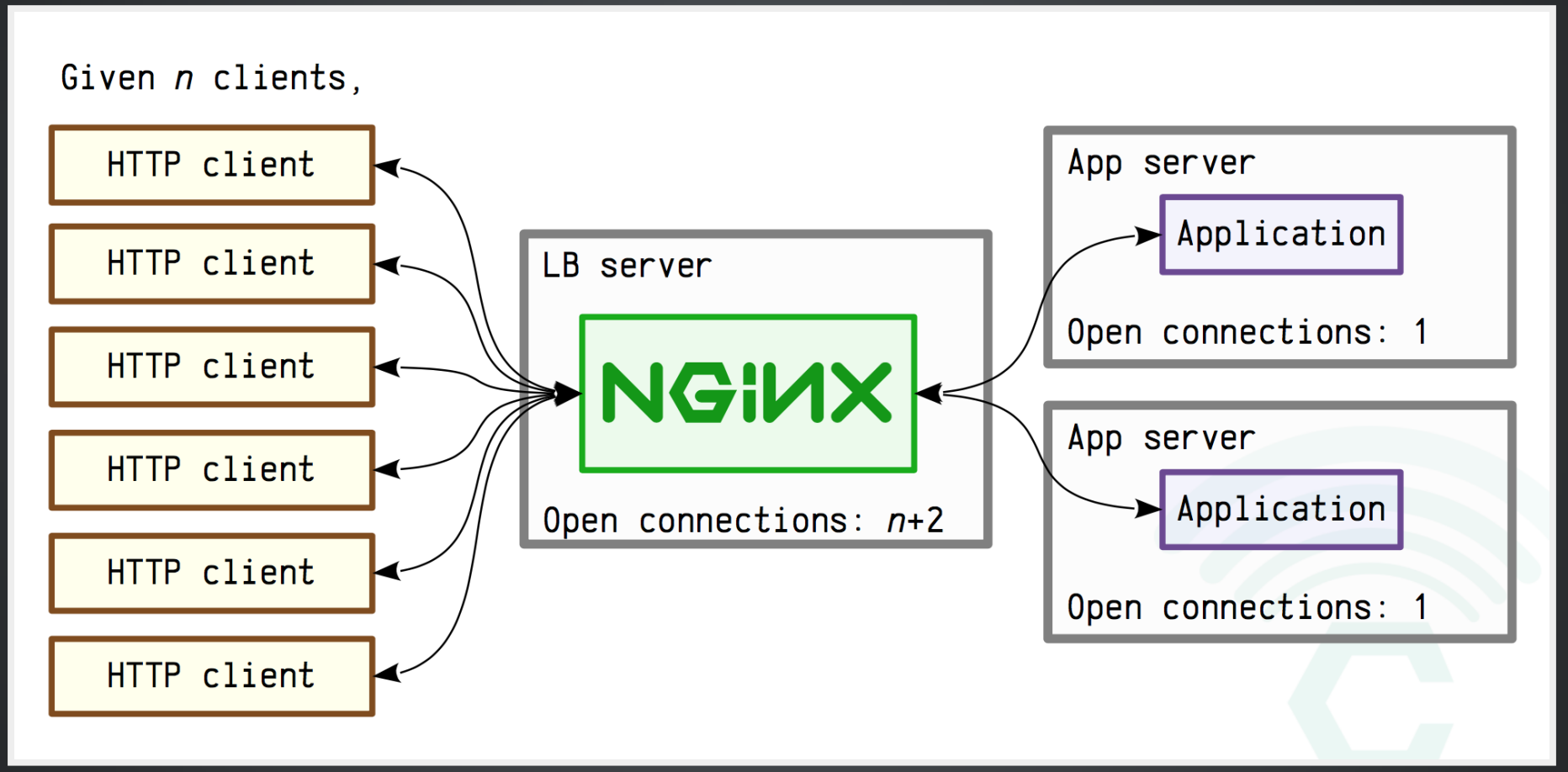
其它 WEBSOCKET 方案

- ActionCable (Ruby on Rails)
- socket.io (Node.js)
- Lightstreamer (Java)
- Faye (Node.js and Ruby)
- ...

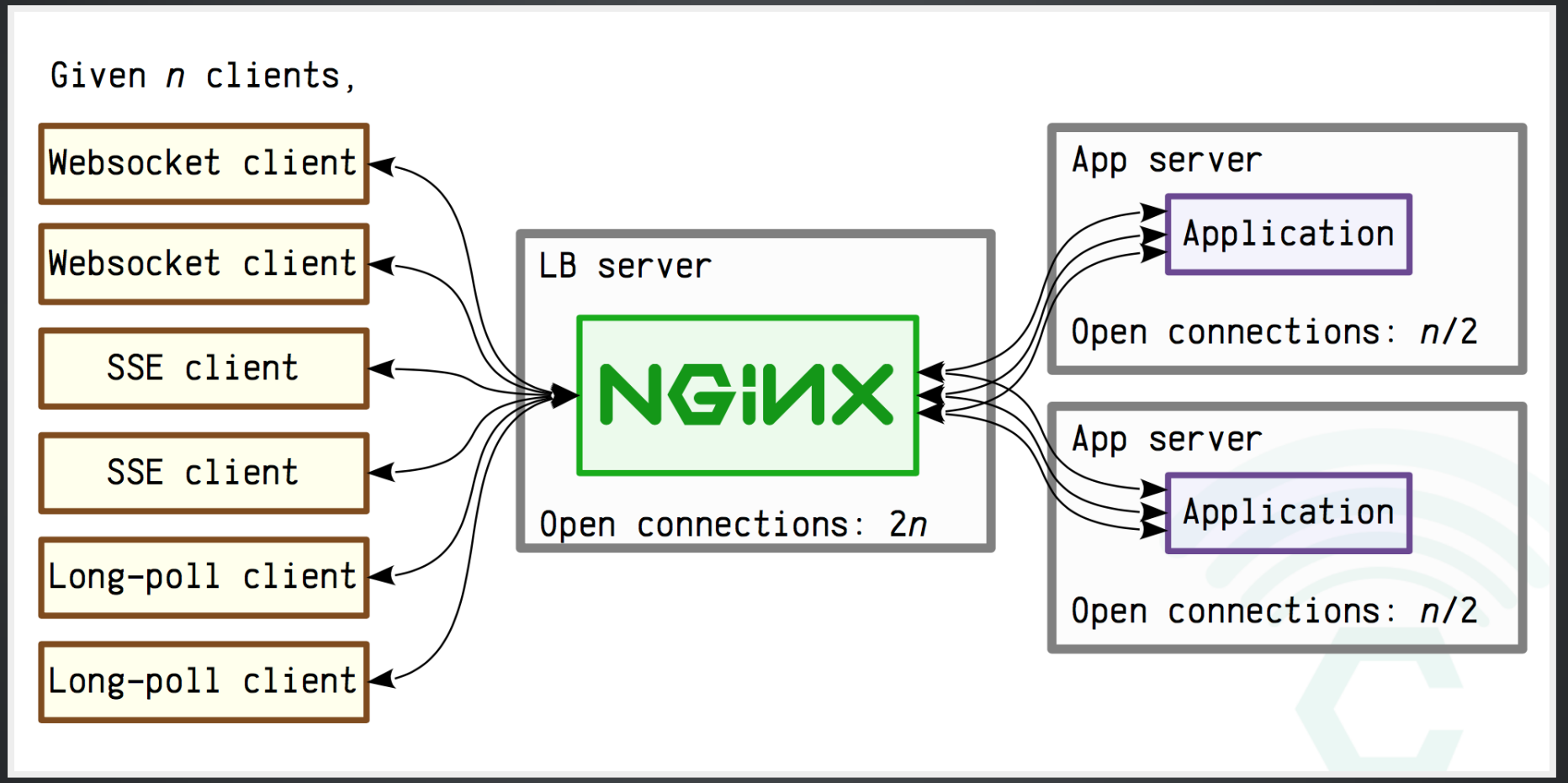
NCHAN 的特别之处

- 可使用 Restful API，完全利用现有技术架构
- 支持各种 hook，配置灵活
- 同时支持 Websocket, EventSource 和 Long Poll
- 高并发，高性能，易于伸缩

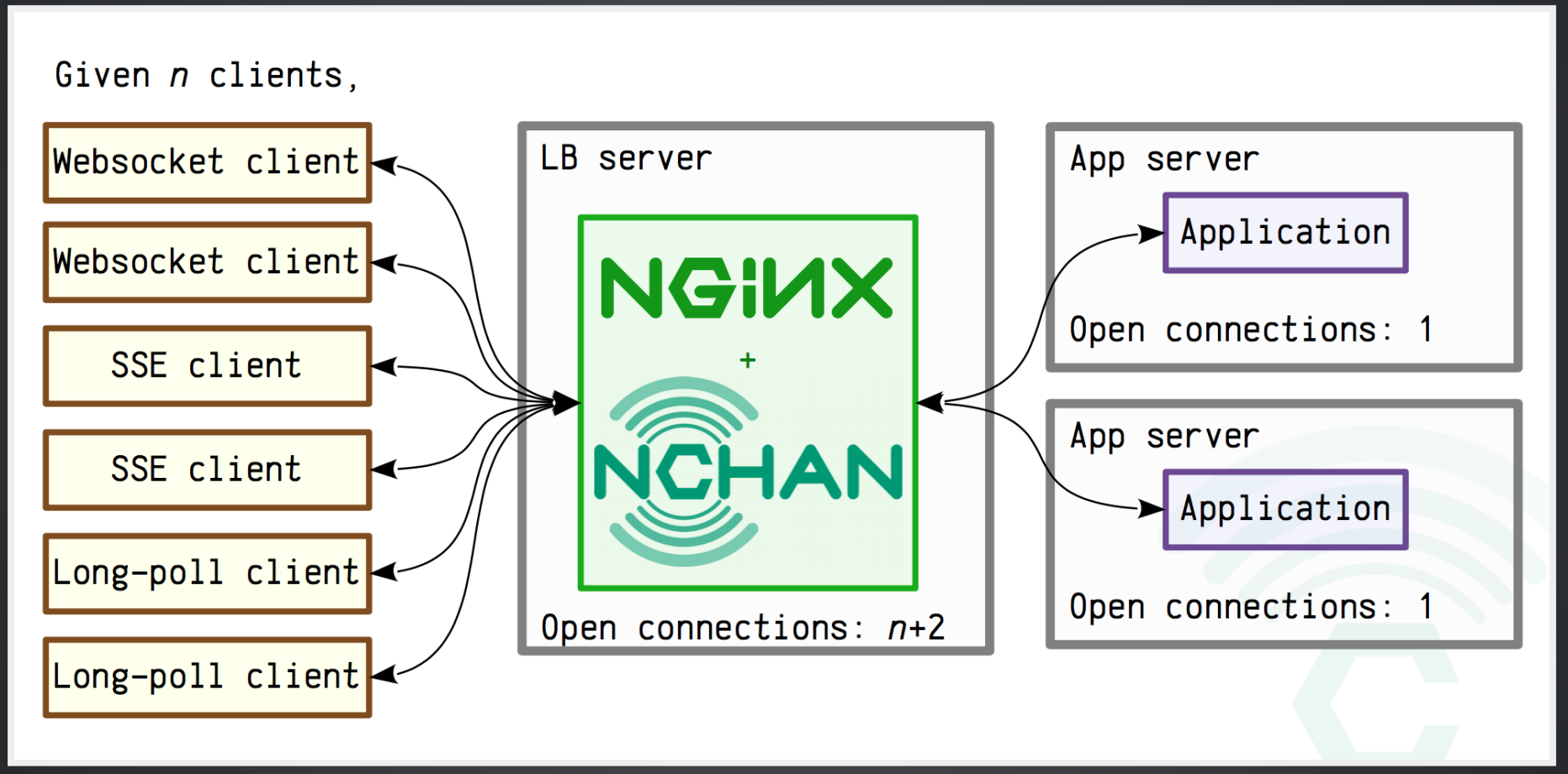
最常使用的 LOAD BALANCING HTTP



常见的 LOAD BALANCING WEBSOCKETS

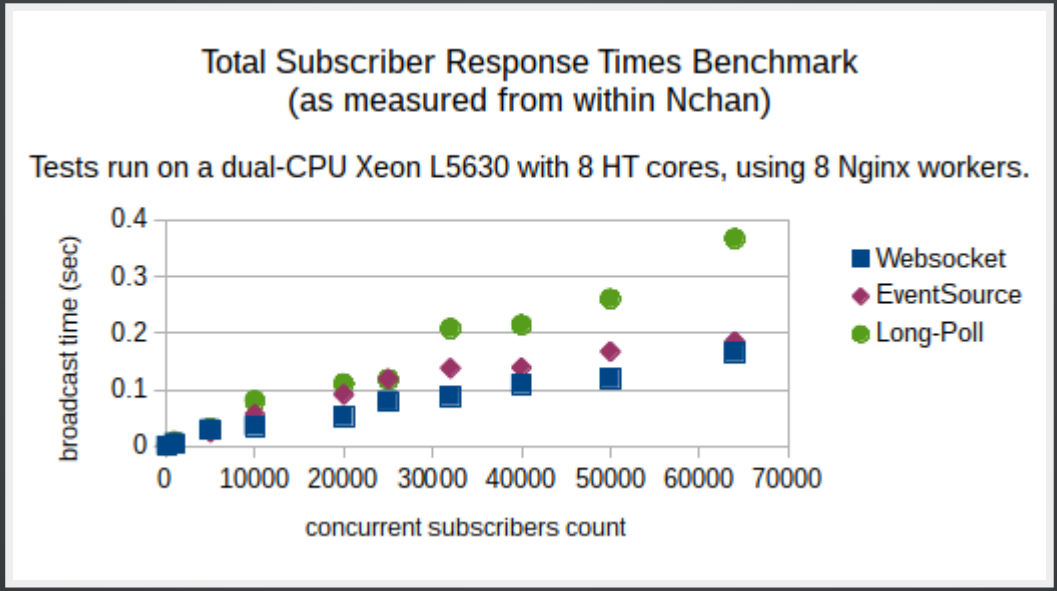


巧妙的 NCHAN LOAD BALANCING



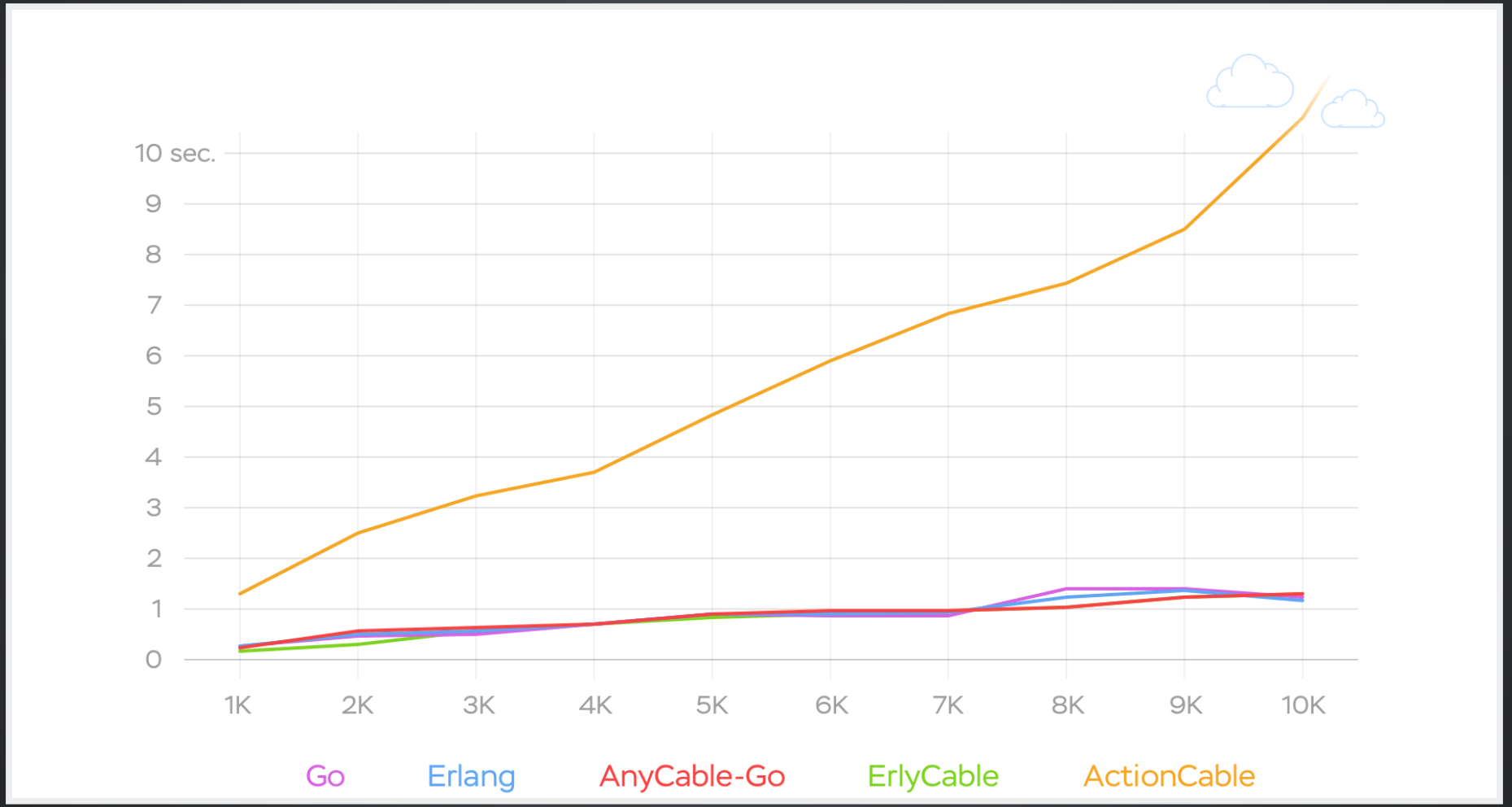
NCHAN 的性能 BENCHMARK

Nchan 的性能非常突出



其它方案的 BENCHMARK

ActionCable 的数据有点惨不忍睹



NCHAN 简单演示

```
#very basic nchan config
worker_processes 5;
http {
  server {
    listen        80;

    location ~ /pub$ {
      nchan_publisher;
      nchan_channel_id test;
    }

    location ~ /sub$ {
      nchan_subscriber;
      nchan_channel_id test;
    }
  }
}
```

```
curl -X POST http://localhost/pub -d hi

  queued messages: 1
  last requested: 0 sec. ago
  active subscribers: 1
  last message id: 1461622867:0
```

```
var ws = new WebSocket("ws://127.0.0.1/sub");
ws.onmessage = function(e) {
  console.log(e.data);
};


hi
```

实践中的问题

- 识别用户身份
- 订阅安全性
- 跟踪在线用户
- 处理消息发送
- 处理群组消息

识别用户身份

- 为用户分配唯一固定的 Channel ID
- 使用频道 ID 建立 Websocket

```
location ~ /channels/(\w+)/sub$ {
  nchan_subscriber;
  nchan_channel_id "$1";
}
```

订阅安全性

- 频道 ID 使用比较长的随机字符串，如 uuid
- 给订阅加上 hook，通过额外的 token 认证

```
location ~ /channels/(\w+)/sub$ {
  nchan_authorize_request /ws_authorize;
  #...
}
location = /ws_authorize {
  proxy_pass http://hook_demo/websockets/auth;
  proxy_set_header X-Channel-Id $nchan_channel_id;
  #...
}
```

跟踪在线用户

- 给订阅加上 hook，把连接的用户 ID 加入 Redis Set
- 通过 Redis Set 判断用户是否在线，遍历在线用户
- 给退订加上 hook，把用户 ID 从 Redis Set 移除

处理消息发送

- 使用 Http 短连接请求
- 再建一条用于发送的 Websocket

处理群组消息

- 多个用户订阅同一个群 Channel ID
- 遍历群组用户逐个发送消息

其它问题

- 消息缓存
- Redis 存储
- 水平扩展
- 高可用
- ...

总结

- Nchan 巧妙利用 Ngnix 的能力，利用现有技术架构，简单高效解决 Web 实时通信问题
- Nchan 是一种轻量级、高性能、高并发和跨语言的解决方案，值得研究使用，深入挖掘

THANK YOU & QA

薄荷诚聘 Ruby 工程师，欢迎自荐或推荐，非常感谢！

这里是招聘贴

相关资料

- Nginx Nchan 主站
- Rails ActionCable 优劣分析
- Real-time Web Applications with Ruby on Rails
- Websocket Shootout: Clojure, C++, Elixir, Go, NodeJS, and Ruby