

用Golang

# 写一个操作系统

复睿：李四保



# 自我介绍



- 开发过银行设备
- 同花顺移动证券
- 前后参与创立过多家公司
- 用Golang写了一个云操作系统



# 自我介绍



- 多个行业 电力、石化、银行、证券、广电、电商
- 多种职业 研发、产品、运营、公司运营
- 喜欢各种宗教
- 喜欢在虚无中创造一个世界





- 牛B的作者
- 曾经的痛点
- 上帝般的手法



# 结缘Golang：上帝的手法



- 斯宾诺莎
- 盖娅
- 文明
- 互联网



# 结缘Golang: 上帝的手法



- 大道至简
- 无用的都会被舍弃
- 互动和映照
- 迭代中前行



# 小试牛刀: **monitor**



## 简介

监控在线系统所有请求的性能，并通过网页进行提供查询功能

## 功能

模拟ssh登录各server, 采集所有请求的数据

根据服务器、日期、请求，显示所选请求的性能走势图，包括平均和最大响应时间。

选定时期，显示这一天所有请求的请求数目，平均响应时间中，最大响应时间。

可以根据请求数、平均响应时间，最大响应时间对数据时行排序。

## 代码

1个文件服务器、5个webapi、6个goroutine

部署简单，可以同时 windows 和 linux 上运行

去掉美观的行，只有200来行



# 来份硬菜：用Golang写一个操作系统



- 程序员 VS 码农
- Golang VS C++
- 和朋友的一场对话

说一万句，不如做一件事：用Golang写一个操作系统





# 操作系统历史



- 操作系统  
相对完整的控制系统资源，有应用体系  
Unix,CP/M,dos、windows,
- 生态操作系统  
在原有操作系统内核基础上，构建新的应用体系及应用生态  
ios、android
- 云操作系统  
chrome os和阿里云os。在上一层操作系统基础之上和自家的云系统进行整合
- 去中心化的云操作系统 Leither  
架构基本和**chrome os**类似，只是将后端的云系统去中心化，使之脱离巨头的控制



# 国内互联网现状



- 互联网正在整合各个行业，对各个传统行业进行重构
- 互联网的格局正在固化  
行业巨头垄断互联网资源，全面压制创新空间
- 人工智能和智能硬件的快速发展，引起了人们对未来的担忧
- 过度行政监管无处不在，市场机制无法顺畅运行



# 互联网的核心：数据



- 数据是互联网的核心，是用户的主权
- 数据日益集中化，规则垄断化，使互联网失去了多样性
- 《失控》作者：未来会有天量的数据只存在用户本地
- 西瓜大丸子汤：大数据处在奴隶社会

无底限的运用规则日益伤害用户的权益，普通用户失去了自主性  
案例：马云，小米，Facebook



# 高压差之下的机会



- 广大网民的各种**需求**被抑制，包括一些强烈需求
- 上千万的互联从业人员的**创造性**被抑制
- 中小企业和个人对**自身数据安全**的担忧日益加强





Leithar为此而生!

---

# Leither是什么?



- 它是一个小程序

前后引用300多M开源代码，多重压缩之后，  
目前可执行程序大约只有2到3M

- 它可以运行在各种常见操作系统下

windows    linux    android

- 它可以跑在用户的路由器、PC、甚至手机上



# Leither是什么?



- 它是一个操作系统

Leither OS

- 它有自己的资源管理体系

动态管理众多在线网民的网络资源、存储资源、运算资源

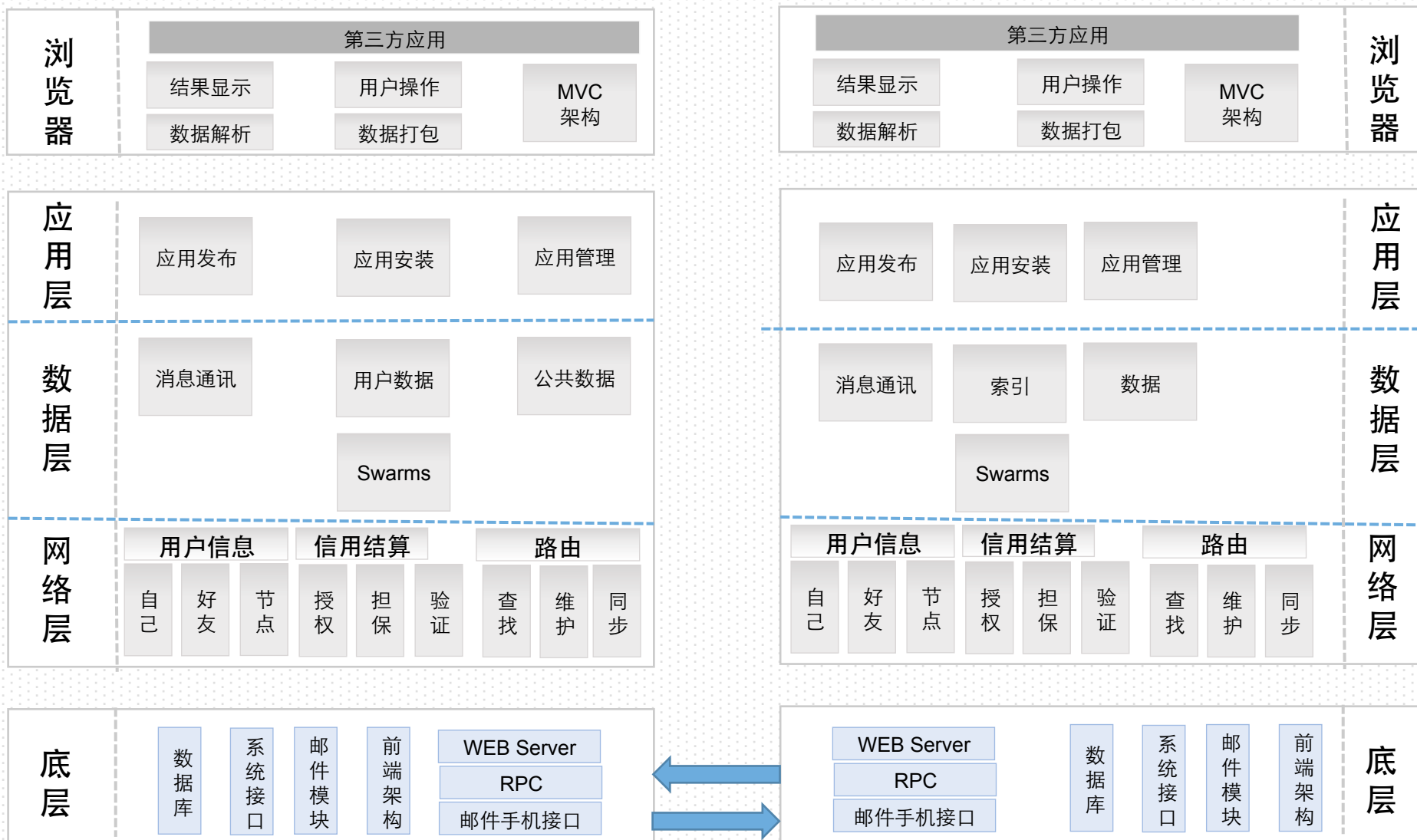
- 它有自己的应用体系

现有互联网的大部分服务形态，都可以在Leither上快速构造出来。  
可以构造视频网站、微博、微信、IM。。。。。

所有这些应用，用户体验基本不变的情况下  
**不需要中心服务器存在**



# Leither的技术架构





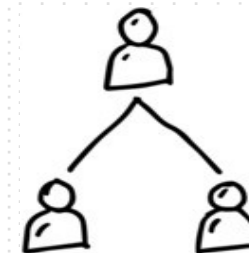
# Leither是什么？



- ✓ 它是一个生态
- ✓ 它有自己的生态成员  
应用开发、内容维护、服务提供、系统支撑、传递者、消费者
- ✓ 它有自己的金融体系  
通过诺币，将不同生态成员整合在一起
- ✓ 它是自调节的



# Leither的生态结构





1诺币=1M流量信用

1毫=1K流量信用

1K=1G流量信用





# 效率：降低基础设施成本

阿里云2万台服务器

新开的北京中心1万台

<http://server.zol.com.cn/452/4522831.html>

09年，Google9万台服务器（另说06年45万台）

构建同行规模（网络，存储）大约只需要数千万人民币

类比：

一个是现有的中国电力系统（电厂+电网）

一个是成本低于电费的个人太阳能加自动组网技术





# 效率：降低应用开发维护成本

- 可以支持各种开发语言  
php golang java c# js python html5  
目前优先支持html5架构
- 没有前端和后端之分  
只需要对编程要求最低的js程序员再加美工，便可以完成所有开发工作
- 自动支持大数据

相比于传统的互联网应用，用户越多，系统的支撑能力越强。同时不需要考虑应用规模变大之后引发的开发团队失控问题。

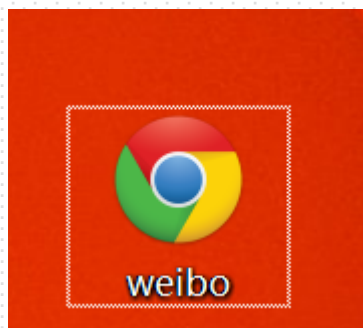
# Leither-应用演示:微博



保存一个小文件到桌面，通过浏览器打开。

用户体验和传统的网页浏览没太大区别：

微博，有新浪微博的大部分功能。



# 应用的内部代码展示



api

开发方式

应用发布





# 目前阶段：

- 小范围内测已过，架构的各体系基本完备，可行性已经验证
- 自建节点，自建应用，万人规模左右的内测  
现有的微博应用，继续强化





# 选Golang的理由



- ✓ 其他语言不适合
- ✓ 丰富的开源资料
- ✓ 跨平台
- ✓ 编译速度不到3秒
- ✓ 模块化很好，耦合度很低
- ✓ 大量使用google的技术
- ✓ 我喜欢Golang  
价值观，哲学思想都比较符合





# 其它语言不合适

- **C或C++**  
开发效率低下，最少十人以上，并且是痛苦的经历
- **Java**  
代码臃肿，部署麻烦，不便分发扩散
- **其它**  
性能不足，底层接口不足，无法进行系统级的开发
- **Golang**  
性能足够，可以操作底层，生成程序足够小，易于分发



# 丰富的开源资料



拉下来的github.com目录有300M

- Hprose
- Leveldb
- Ledis
- Beego



# 跨平台



按设计要求系统的运行场景有：传统服务器、PC、手机或路由器。  
能满足的语言只有：

- C/C++
- Java
- Golang



# 迭代和重构



前后发布了数十个版本，上百次重构，用golang毫无压力。主要得力于以下特征：

- 极简的语法
- 快速的编译  
以前的同花顺客户端编译要近20分钟，现在小于3秒
- **Package**和易用的接口  
耦合度低  
模块化好





- **初觉的很好**  
错误和异常是不同的

- **后用起来很烦**  
大量的重复代码

- **最后觉得很合理**

写代码的时间很短，测试和查问题的时间很多，相当多的时间是因为某个错误忽略或处理的缺失。多加几行代码占位，视觉提醒和后期查问题都很方便





# Golang的总结

- **大道至简**

goroutine, 接口, 数据结构, Web

- **无用的都会被舍弃**

语法, 书写格式, ++ --的一些特性

- **互动和映照**

控制机器不再重要, 弱化了指针, 弱化了性能极致追求。  
强化了多核, 强化了网络

- **迭代中前行**

Golang很早就形成了闭环; Golang的项目也很适合重构和迭带



硬件时代最佳的系统语言是C/C++;云时代最佳的系统级语言是Golang

# 结束语



希望能和大家成为朋友

知乎ID: 李四保;

微博ID: 李四保

希望大家加入到这个项目中





THANK YOU