利用开源社区
打造微服务生态体系

作者：敖小剑

前言

今天讲什么?

在微服务方兴未艾如火如荼之际，
如何打造合适自己公司实际情况的微服务框架?
如何实现以微服务微核心的完整的开发生态体系?
开源社区为此提供了哪些支持?
面对眼花缭乱的众多技术，该如何抉择?

今天的内容：看看微服务的世界

基础设施

微服务框架

核心技术

- 内容：只罗列，不展开
- 范围：遗漏是必然的
- 建议：一家之言，仅参考

璀璨群星

SmartStack
Dropwizard
Zuul
etcd
Spark Framework
Spring MVC
Jersey
Netflix OSS
zookeeper
Spring Cloud
Apache
Doozer
Consul
Prometheus
gRPC
XNIO
Qconf
HTTP/2
Diamond
Archaius
Eureka
Governator
Motan
HTrace
Kafka
HA proxy
ELK
Zipkin
Spotify
Serf
Mantl
Ribbon
CAT
Grafana
F5
Thrift
Spring Boot
SkyDNS
Disconf
Karyon
Spring HATEOAS
Vamp
Pinpoint
Spring Cloud Netflix
RabbitMQ
Netty
Prana
Vintage
Hessian
NSQ
okhttp
Mina
ActiveMQ
NSQ
Dubbo
LVS

第一部分

核心技术

微服务中必然涉及到的部分核心技术，必不可少

每个技术都有很多种实现，各具特色

在目前的开源界，选择众多，可谓繁星璀璨

核心技术

进程间通讯

服务注册与发现

负载均衡

熔断

故障转移
Failover

失败重试

快速失败
Failfast

微服务中的进程间通讯

对比

在这点上，微服务和以OSGi、jigsaw为代表的Java模块化方案形成鲜明对比

本质差异

在微服务架构中，为了彻底隔绝不同服务，采用了最坚决的方案，强制要求服务之间：

通过 **远程访问** 方式进行通讯

OSGi

Jigsaw

微服务中通讯的多样性

两个维度

交互对象的数量

一对一，一对多

应答返回的方式

同步，异步

两个维度组合的可能性

	一对一	一对多
同步	Request/Response	
异步	Notification	Publish/subscribe
	Request/Async response	Publish/Async responses

再结合三种类型，现在大家对“多样性”有体会有吧？……

网络类库的选择

如日中天 Netty

由JBOSS提供

最新版本4.1,2016年发布

英雄迟暮 Mina

来自Apache

几乎不再有大版本更新

鲜为人知的 XNIO

来自Redhat

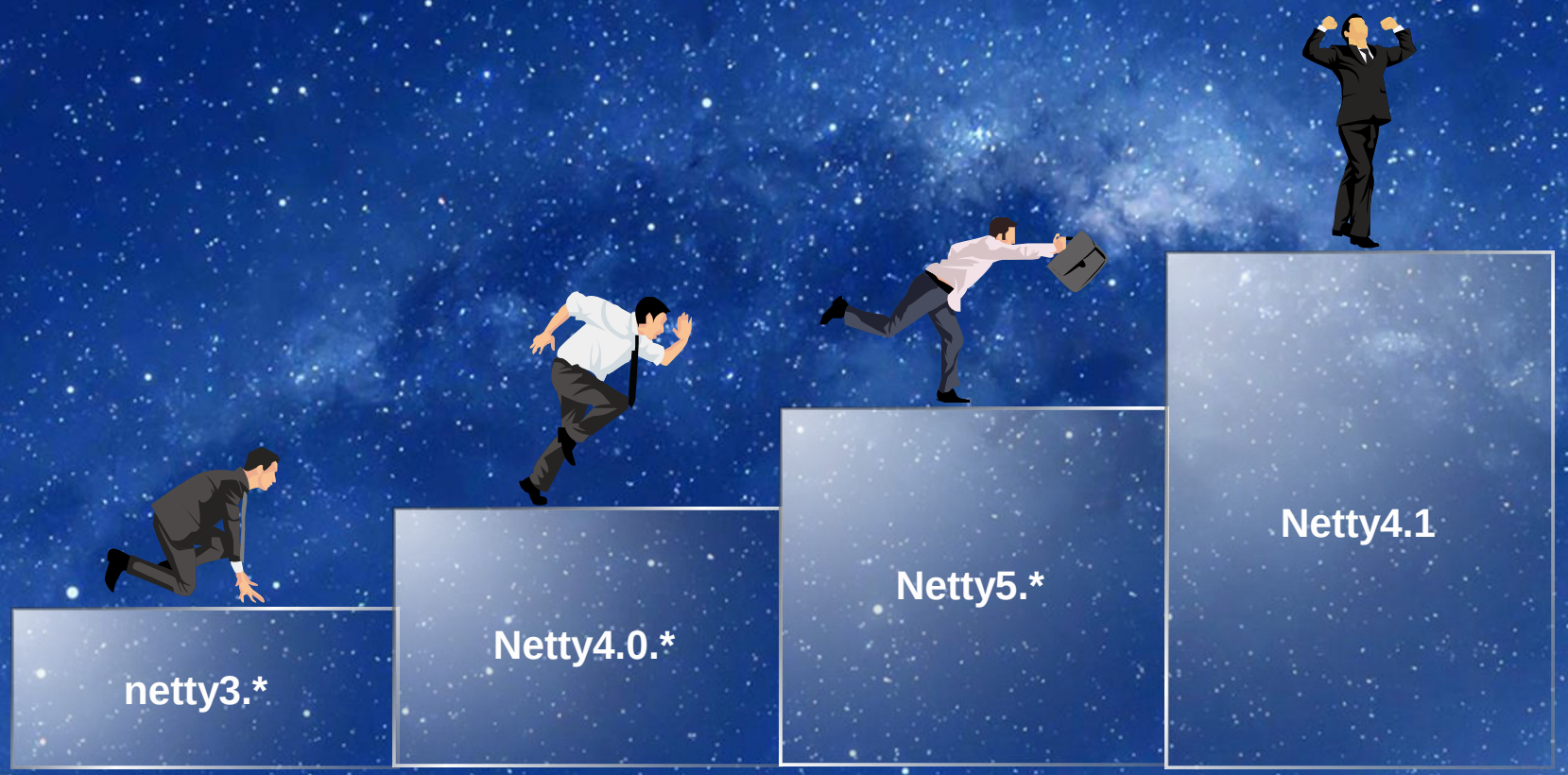
主要服务于自家的undertow

支持Android Okhttp

新兴势力，发展迅猛

支持Android，支持HTTP2

Netty版本的选择



老版本

推荐升级

新项目不要选

主流版本

追求稳定的选择

已废弃版本

切记不要选!

已上贼船的赶紧下来

最新版本

支持HTTP/2

有需要再上

Rest 框架

因为个人原因偏好 RPC 方案，对 REST 方式研究不多。

平时比较大众化的选择是：

- 服务器端 Spring MVC
- 客户端用 Jersey

RPC方案

业界的 RPC 方案众多，这里只详细介绍三种，分别代表老中新三代RPC框架

Caucho Hessian

Apache Thrift

Google gRPC

Hessian

- 基于HTTP协议传输的二进制 web service 方案，由 caucho 公司提供
- 出现的时间比较早

缺点：

1. 效率和 thrift / gRPC 等相比较差，甚至不如REST风格的Jackson
2. 基本停止发展，最后一个大版本 4.0.* 发布于2009年，之后只有bugfix版本
3. 以今天的眼光看 hessian 的特性，比较陈旧

由于历史原因，还是有不少框架在使用 hessian，比如：

- dubbo 默认采用 hessian2 序列化
- 微博新开源的 Motan 框架用的 hessian 4.0.38

Thrift

Thrift 源于 Facebook，在2007年捐献给 Apache
目前最新版本 0.9.3 (发布于 2015-10-06)

业界最经典的 RPC 框架之一
效率极高, 使用广泛，成熟稳定

缺点：

1. 项目似乎不再继续演进，看不到未来的路线图
2. 和 gRPC 相比，缺乏对HTTP/2的支持，缺乏对移动设备的支持
3. 底层通讯机制不够理想：唯品会的OSP框架干脆就放弃 Thrift 底层通讯实现, 直接基于 netty4 重写

- gRPC 是 google 最新发布的开源 RPC 框架
- 声称是"一个高性能，开源，将移动和HTTP/2放在首位的通用的RPC框架."
- 支持 android 和 iOS
- 技术栈非常的新：基于HTTP/2, netty4.1, protocol buffer 3.0
- 堪称新一代 RPC 框架的典范.

- 项目开始于2015年2月
- 1.0正式版本正式发布于2016年8月

缺点：

1. 太新，缺乏实战分享，缺乏文档，社区还不广泛
2. 遇到问题时，google也不容易找到资料 ☺

RPC的选择(个人建议)

怀旧

怀旧的同学请选择 hessian

求稳

求稳的同学请选择 thrift

最新

追新的同学请选择 gRPC

野路子: 自己定制

01 **网络通讯**
可以选择netty，mina等类库

02 **协议**
可以选择 HTTP、HTTP2、TCP等

03 **编解码**
可以选择 JSON/Proto 2或者3/thrift，甚至XML

04 **最关键的**
三种方式排列组合，结果五花八门

消息队列

RabbitMQ

Apache Kafka

NSQ

Apache ActiveMQ

消息队列的选择(个人建议)

轻量 轻量级使用 选择 RabbitMQ

重量 重量级使用 选择 Apache Kafka

高 要求非常高，考虑一下 NSQ

禁 没有特殊原因,不要选 ActiveMQ

服务注册/服务发现

Raft 协议

- Consul
- etcd

PAXOS 协议

- zookeeper

一致性要求

- 强一致性
- 弱一致性

Redis

微博 Vintage

DNS

- Spotify
- SkyDNS
- DNS by Consul

强一致性方案

	说明	编写语言
Zookeeper	来自 Apache，强一致性(CP), 使用 Zab 协议 (基于PAXOS)	Java
Doozer	doozerd 的 Go 语言客户端，强一致性，使用 PAXOS 协议。 很多年前就存在的项目，已经停滞很久.(没有特殊理由，不要选择)	Go
Etcd	据说是受 zookeeper 和 Doozer 启发，使用 Raft 协议	Go
Consul	来自 hashicorp 公司，和 etcd 一样也是基于 Raft 协议 Consul 最大的优势，是提供可以直接使用的成品	Go
SmartStack	来自 Airbnb，由 Nerve 和 Synapse 两个部分组成，依赖zookeeper和haproxy	Ruby
Eureka	来自 Netflix，服务器端和客户端都是用 Java 语言编写，因此只能用于Java和基于JVM的语言	Java
Serf	采用的是基于 gossip 的 SWIM 协议	Go

弱一致性方案

	说明	实现方式
Vintage	新浪微博内部使用(没有见到开源,可以借鉴思路) 基于 Redis 的轻量级 KV 存储系统	Redis
Spotify (DNS)		DNS
SkyDNS		DNS
DNS by Consul	可以和 Nginx 集成	DNS

内建服务注册

服务注册是任何一个服务化/微服务框架的必不可少的部分

很多框架内建了对服务注册的支持

Dubbo

- 支持注册中心扩展
- 支持多种实现: 默认zk

Motan

- 支持zookeeper
- 支持consul
- (内部用Vintage)

Spring Cloud

- Spring Cloud Consul
- Spring Cloud Zookeeper

Netflix OSS

- Eureka

个人推荐

虽然方案众多，但是对于普通用户的多数情况，建议在下面三个主流方案中选择：

Zookeeper

client尽量使用 Curator ！

etcd

Etcd

Etcd2: rest
Etcd3: gRPC, 通过proxy提供rest支持

Consul

Java client API 的通知机制用起来比较难受

负载均衡

软件

- Nginx
- HA proxy
- Apache
- LVS

硬件

- F5

实现方式

- 服务器端负载均衡
- 客户端负载均衡

单独使用

- Netflix Ribbon

框架集成

- 所有框架都内置负载均衡的实现

最新消息：
Github 即将开源他们新设计的负载均衡系统GLB(Github Load Balancer),值得期待

熔断器

这是目前找到的唯一的熔断器开源项目

第二部分

微服务框架

工欲善其事必先利其器

在推行微服务时，一个微服务框架是必须的

无论是自行打造，还是选现成的

下面我们来介绍微服务框架有哪些可选的轮子

Dubbo：一个绕不开的名字

DUBBO

Dubbo：一段辉煌的历史

"DUBBO是一个分布式服务框架，致力于提供高性能和透明化的RPC远程服务调用方案，是阿里巴巴SOA服务化治理方案的核心框架，每天为2,000+个服务提供3,000,000,000+次访问量支持，并被广泛应用于阿里巴巴集团的各成员站点。"

—— 来自 dubbo 官方的介绍

Dubbo：令人叹息的现状

- 阿里放弃了 dubbo，内部改为使用 HSF 框架
- dubbo 开发团队解散，合并到 HSF
- 阿里停止对 dubbo 的支持和后续更新
- 在2012年底之后基本就不再继续，后续也只有极少量的更新

DubboX：延续，但……

- 当当 fork 并推出了 dubbox
- 支持 REST 风格远程调用，并增加了一些新的特性
- DubboX 陆续有一些版本发布，但只是简单修复
- 最重要的：后续没有任何功能性的发展计划
- 路，貌似也是走到头了……

Dubbo：其兴也勃 其“亡”也忽





2011
开始开源

起于 2.0.7版本

2012
频繁更新

一年内发布了31个小版本，叹为观止！

2013
戛然而止

年初发布了一个小版本
然后，就再没有然后……

……
沉寂

2016
沉寂

Dubbo: 总结

个人对 dubbo 停止发展深表遗憾：这个项目本来可以成为一个伟大的项目，尤其在2015年微服务和 docker 盛行之后，本可以有更大的发挥空间，可惜了……

01 非常好

极富研究价值，即便是在废弃多年后的今天

02 如果……

能一直发展下来，前途不可限量……

03 面对现实

dubbo 已死，在生产上使用时请谨慎(仅代表个人意见)

Motan：下一个Dubbo?

新浪微博

2016年8月开源

Motan：初衷

“Dubbo 功能上比较丰富，但当时我们想要一个比较轻量的 RPC 框架，方便我们做一些适合自己业务场景的改造和功能 feature，以达到内部业务平滑改造和迁移的目的。这种情况下，在 dubbo 上改的成本可能比重新写一套更高。最终我们决定开发 motan RPC。”

—— 微博解释他们开发 Motan 而不是使用 dubbo 的原因

Motan：技术栈

- 底层通讯引擎采用了Netty网络框架
- 序列化协议支持 Hessian2 和Java序列化
- 通讯协议支持Motan、HTTP、TCP、memcached
- 服务注册支持zookeeper 和 consul

比较有意思的是：微博内部使用的是基于redis的vintage ☺

Motan: 总结

01 潜质

年轻，更新积极，有极大的发展潜力

02 现状

刚出来，功能和稳定性有待观察

03 使用

轻量级，入门门槛比 dubbo 低，但是特性少很多

04 局限

对跨语言调用支持较差，主要支持java

Netflix OSS：微服务演进的典范

NETFLIX | OSS

Netflix Open Source Software

Netflix OSS：组件众多

Netflix OSS: 总结

01 品质

非常高，可以直接用，也可以造轮子时做参考

02 社区

成熟，使用者很多，文档和资料齐全

03 口碑

有分歧，有人特别推崇，有人觉得坑多

Netflix OSS: 局限(个人观点，仅供参考)

- 先天不足: 问世太早，当时只有AWS可选
- 权衡背景：如果不是直接使用AWS EC2……
- 当前选择：docker，mesos / kubernetes

根本原因：

- OSS是5,6年前Netflix解决问题的思路和产出
- 而那时没有docker，没有微服务，没有mesos、kubernetes
- 如果现在重新设计，OSS的技术栈必然不同

建议：

- 不放弃，不盲目
- 谨慎选择，按需选择
- 避免不必要的复杂度

Spring: 一路走来

2002

- EJB 统治J2EE
- 一个人，一本书

2003

Spring Framework

- 2003，Spring团队组建
- 2003，spring@Sourceforge
- 2004，spring 1.0

2014

Spring Boot

- 开发始于2013年
- 2014年伴随 spring 4.0 发布

2015

Spring Cloud

- 2015年3月 1.0 发布

2015

Spring Cloud Netflix

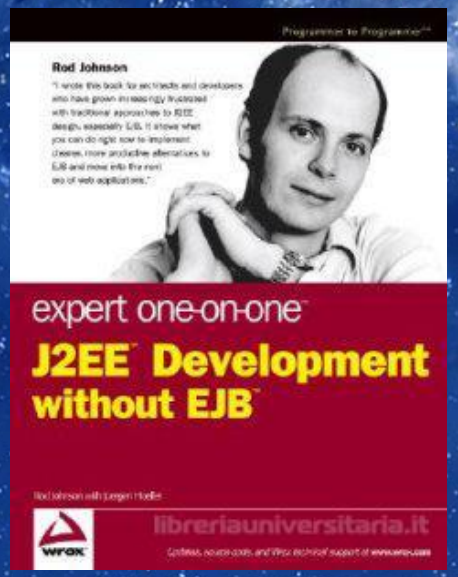
- Spring Cloud 的子项目

1 2 3 4 5

Spring 之初: 暴露年龄的时候到了 ☺

真正的大神
Rod Johnson

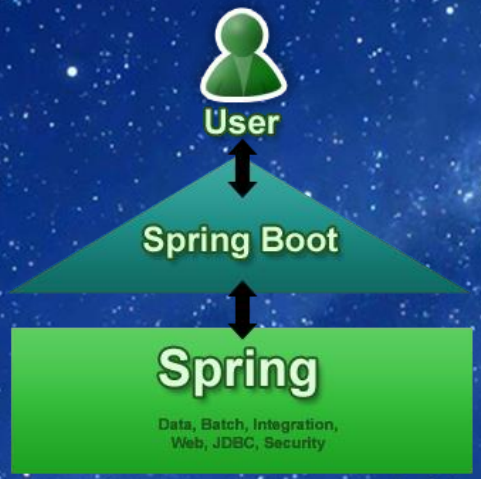
- 2002年，EJB盛行的年代
- 对EJB做技术层面的深入分析
- 质疑EJB
- 促成Spring的诞生



- 2004年
- 荡气回肠的一本书
- 向EJB的架构和观念发起冲击
- 对比分析开源架构: spring, webwork, ibatis, hibernate
- SSH从此开始大行其道

- 2005
- 抛弃EJB，只谈Spring

Spring Boot：定位



- 致力于蓬勃发展的快速应用开发领域
- 用来简化新Spring应用的初始搭建以及开发过程
- 目标不在于为已解决的问题域提供新的解决方案
- 而是为平台带来新的开发体验
- 并简化对已有技术的使用

Spring Boot：功能

- 创建独立 Spring 应用
- 直接内嵌 Tomcat， Jetty 或者 Undertow
- 提供 'starter' POM 文件来简化 maven 配置
- 尽可能的自动配置 Spring
- 提供产品级别的功能如 metrics， 健康检查和外部化配置
- 完全没有代码生成并不需要 XML 配置

Spring Boot 总结

不是微服务框架

离微服务框架还是有很大距离
最基本的服务发现都没有

超广泛的群众基础

特别适合已经熟悉
Spring 体系的开发团队

更适合作为
微服务框架的基石

推荐在 Spring Boot 的基础上，
而不是 Spring 的基础上搭建
自己的微服务框架

以Spring Boot为基础

Pivotal 团队推出解决方案
Spring Cloud

Spring Cloud：基于Spring Boot的云应用开发工具

Spring Cloud：定位

- 目标：占领企业微服务架构领域
- 帮助开发完整的微服务系统
- 弥补Spring Boot的不足(或者说补充定位)
- 完全基于Spring Boot

Spring Cloud 子项目

子项目	说明
Spring Cloud Config	中心化的外部配置管理，由 git 仓库支持
Spring Cloud Netflix	和多个 Netflix OSS 组件集成(Eureka, Hystrix, Zuul, Archaius, etc.)
Spring Cloud Bus	将服务和服务实例连接到分布式消息的 event bus。 用于在集群内传播状态变更。(如配置变更事件)
Spring Cloud Cluster	leader 选举和通用有状态模式
Spring Cloud Consul	使用 Hashicorp 公司的 Consul 来进行服务发现和配置管理
Spring Cloud Zookeeper	使用 Apache Zookeeper 做服务发现和配置管理
Spring Cloud Stream	消息中间件抽象层, 目前支持Redis, Rabbit MQ和Kafka
Spring Cloud Sleuth	用于 Spring Cloud 应用的分布式追踪 兼容 Zipkin， HTrace 和基于log(如 ELK)的追踪

Spring Cloud: 总结

01 **有潜力**

很年轻的项目，可以关注，前景看好

02 **社区**

最近使用的人越来越多，多次看到分享

03 **建议**

可以尝试，一般也能用

也可以考虑下面要介绍的Spring Cloud Netflix 套件

Spring Cloud Netflix：强强联手

微服务框架的遗憾(一家之言)

虽说选择有不少，真正满意的，没有！

☺

期待未来，能有真正压倒性优势的强力产品面世

第三部分

基础设施

这些不是微服务的范畴

但是对于微服务来说至关重要

微服务要想成功

最好能有这些配套的基础设施

基础设施

分布式配置管理

APM
应用性能监控

日志分析

服务监控/告警

分布式配置管理

分布式配置管理

	来自	说明	存储	编写语言
Disconf	百度(?)	只支持Java	zookeeper	Java
Qconf	奇虎360	支持c/c++、shell、php、python、lua、java、go、node 等语言	zookeeper	C++
Diamond	阿里	淘宝网Java中间件团队的核心产品之一 开源(但是访问地址需要输入用户名密码？) 不建议使用	mysql	
Archaius	Netflix	Netflix OSS 套件之一		Java

APM/应用性能监控

在APM领域，有非常多的优秀的商业公司，如青云，听云, OneAPM, New Relic

开源界也提供了不少项目

PinPoint

- 来自韩国团队
- 目前只支持Java
- 基于Dapper论文
- 存储 Hbase

zipkin

- Twitter开源
- 基于Dapper论文
- 存储 mysql/ Cassandra/ElasticSearch

CAT

- 大众点评开源
- 需要硬编码做”埋点”

HTrace

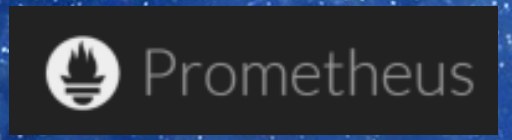
- Cloudera 开源
- 捐献给Apache
- 还在孵化中

日志分析

在日志分析领域，ELK套件是目前当之无愧的霸主地位

Elasticsrach + Logstash + Kibana

- 开源实时日志分析平台
- 应用广泛，社区成熟，文档齐全



Prometheus + Grafana

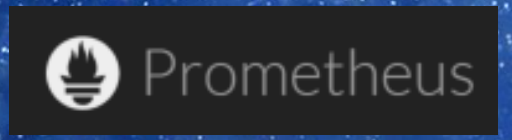
- 新秀
- 参考 Google Borgmon 的设计
- 和 ELK 相比较轻，容易扩展
- Cloud Native Computing Foundation的正式组件

日志分析

在日志分析领域，ELK套件是目前当之无愧的霸主地位

Elasticsrach + Logstash + Kibana

- 开源实时日志分析平台
- 应用广泛，社区成熟，文档齐全



Prometheus + Grafana

- 新秀
- 参考 Google Borgmon 的设计
- 和 ELK 相比较轻，容易扩展
- Cloud Native Computing Foundation的正式组件

片尾曲

仰望星空，弱水三千
立足当下，只取一瓢


T H A N K S Y O U