



## Solomon【肖老师】

海量数据 高并发 亿级用户 架构师 技术VP 大厂

产品集群 PRODUCT CLUSTER

查看更多



麦田音乐



太合麦田



海螺音乐



大石版权



千千音乐



太合乐人



秀动



Lava熔岩音乐



亚神音乐

## 讲师介绍

### Solomon(肖老师)

- 前百度环境音乐CTO&架构师、全球海量专利数据项目负责任
- 中国机械出版社签约作家,《深入理解Dubbo工业级架构设计》图书的作者
  - 目前市面上唯一一本完整的架构设计书籍
- 11年互联网大厂经验,在互联网音乐、电商、大数据、高并发、数据挖掘等领域有丰富的实战经验。
- 擅长于通信协议、微服务架构、框架设计、消息队列、服务治理等领域

# 第一天

---

## 问题探索

如果单机单体架构能够支持亿级别并发量，还需要分布式架构吗？如果需要你觉得作用的范围是什么？

分析源码有什么用？为源码而源码？还是向往一个高度？

负载均衡在什么场景中会被使用

负载均衡有哪些策略，使用啥设计模式

负载均衡解决啥问题！弊端在什么地方？

## 课程范围

1. Dubbo 画像
2. dubbo市场需求与前景
3. dubbo在企业项目中的地位
4. dubbo生态圈
5. 了解负载均衡实际使用场景
6. 负载均衡背后的本质原理
7. 负载均衡种类，以及应对业务场景
8. 负载均衡算法
9. 负载均衡涉及调用流程环节

## 课程目标

- RPC核心基本功能具备哪几个？
- 对Dubbo 分布式架构有全面的认知
- 提升获得offer概率【强项】
- RPC 负载均衡底层实现机制
- RPC 负载均衡业务的选择性

## 课程收获

1. 阿里系面试关注的点
2. 提高分布式架构负载均衡实现方式
3. 实际业务中负载均衡的选型考量
4. 了解负载均衡策略算法运行机制
5. dubbo 中如何自定义负载均衡策略

## 聚焦

---

分布式架构中负载均衡实现之治理思想

## 传统的服务调用方式

---

- http-----使用权重
- rest api

- webservice-----xfire
- rmi
- rpc -----解决什么问题

备注：跨进程通信

## Dubbo 市场需求分析





- Netflix 所有组件闭源
- Netflix 被spring cloud alibaba取代
- Dubbo 已成为apacha 顶级项目
- Dubbo 开源生态逐渐完善
  - spring-cloud-alibaba-dubbo
  - spring-cloud-alibaba-seata
  - spring-cloud-alibaba-sentinel-zuul
  - spring-cloud-alibaba-aliyun-sms
  - spring-cloud-alibaba-nacos【替代zookeeper】
  - spring-cloud-alibaba-sentinel
  - spring-cloud-stream-binder-rocketmq

## Dubbo RPC在企业中的地位

- 阿里巴巴
  - dubbo---范围广
  - hsf
  - sofa-rpc--阿里金服必备
- 当当网
  - dubbbox
- 滴滴出行
- 去哪儿
- 中国电信
- 中国工商银行
- 海尔
- 二次开发（开源自研）
- .....
- EDAS AWS（云架构师，）

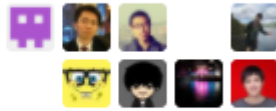
<https://github.com/apache/dubbo/issues/1012>

## Dubbo Github 基本信息【规模之大】

 Used by ▾	642	 Watch ▾	3,211	 Star	26,837	 Fork	17,747
---	-----	---	-------	--	--------	--	--------

dubbo  
tration

201 direct contributors



+ 181

201 contributors

Apache-2.0

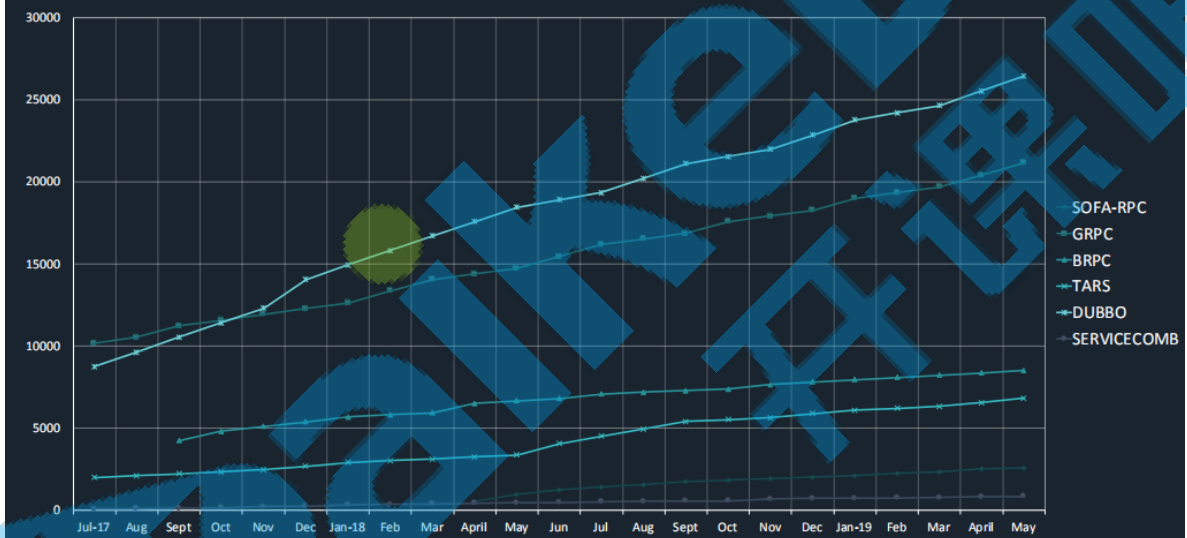
Apache Dubbo is a high-performance, java based, open source RPC framework. <http://dubbo.apache.org>

dubbo rpc rpc-framework java high-performance soa service-oriented service-registration distributed-systems service-discovery  
cluster spring service-consumer service-provider service-registry microservices

## Dubbo未来

### STAR 数增长趋势

2017/07 - 2019/05



# 今天 DUBBO 的扩展

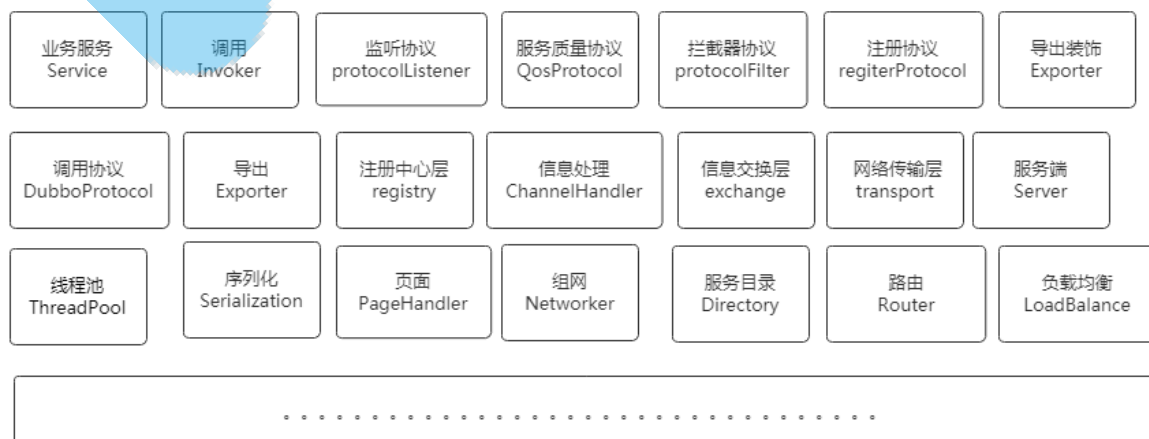


## DUBBO 3.0 路线图

定义协议

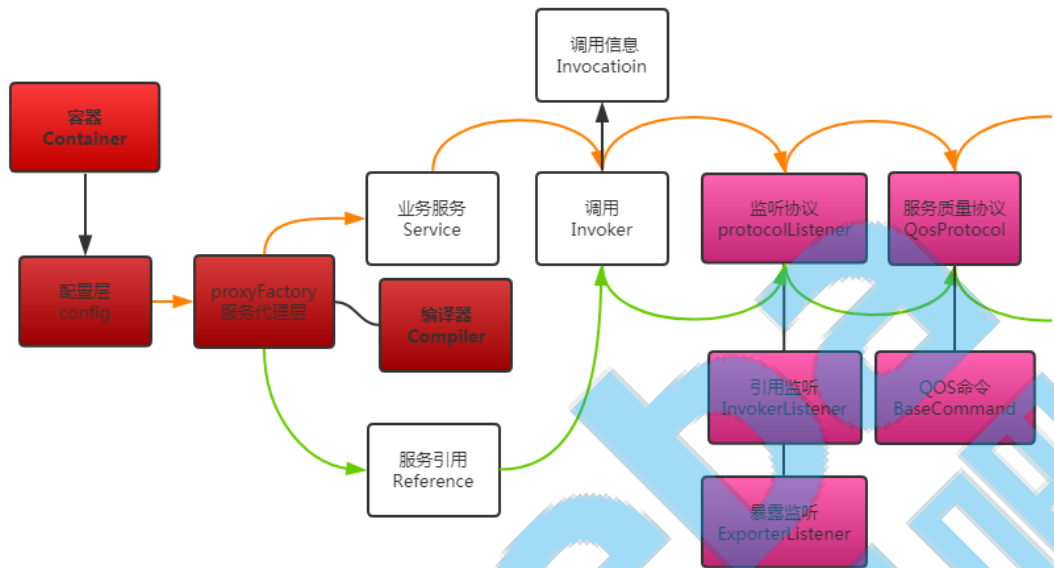


## Dubbo画像

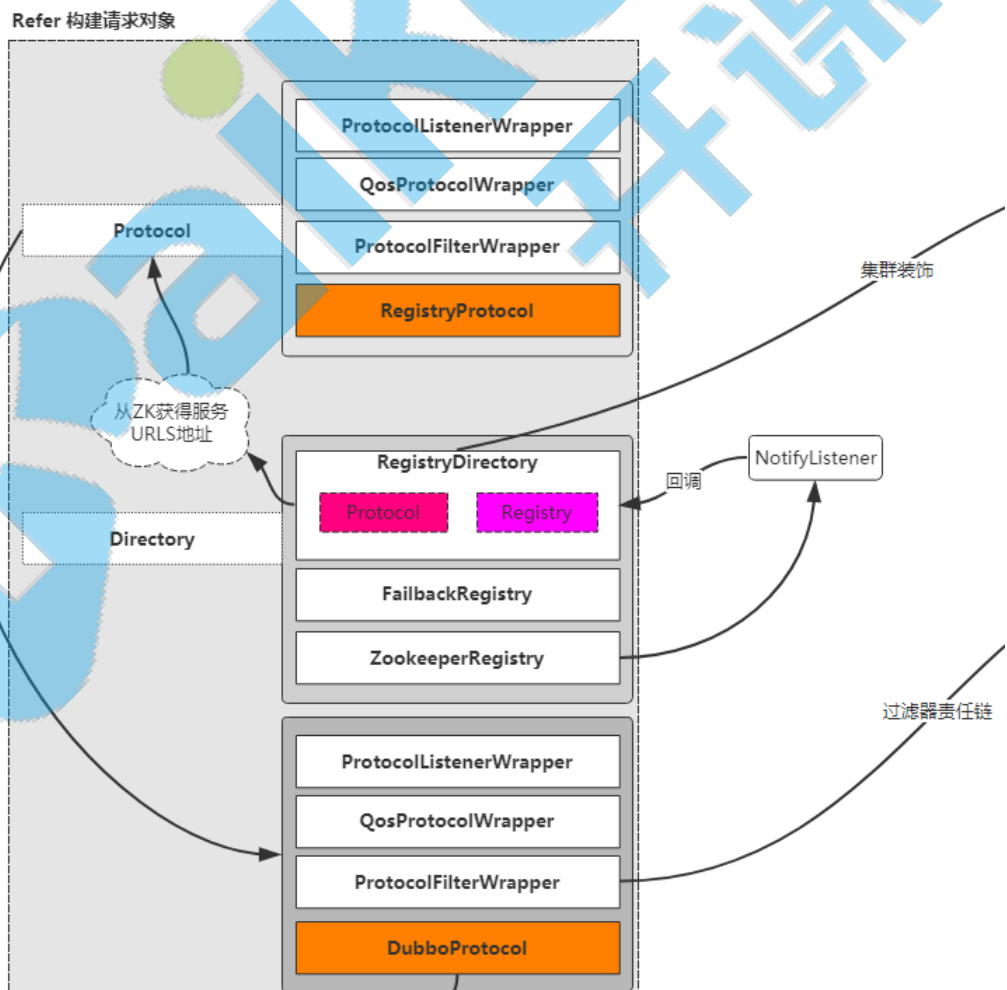


# dubbo 40+核心组件是什么&为什么如此庞大？

- 微内核+插件的设计原则
- 请求与响应流程
  - 部分流程

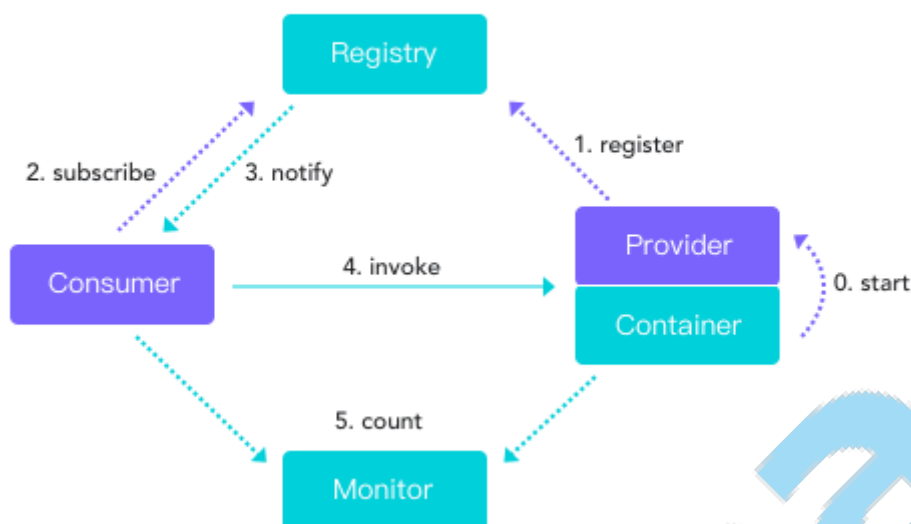


- 详细流程



## Dubbo 架构

....> init    ....> async    —> sync



## 负载均衡使用场景

### 同业务集群场景中

目标：负载均衡适用于高访问量的业务，提高应用程序的可用性和可靠性

作用：

- 分摊流量，提高响应速度【效率】
- 消除单点故障【failover-Cluster】
- 探知负载压力【智能】
- 提高节点的利用率【分发】
- 提升冗余【高可靠】
- 保证服务的稳定性【冗余】
- 横向扩张系统【弹性，统一访问入口】
- 多可用区容灾【LHA】
- 跨地域容灾【99.99999999999999%】

## 负载均衡管理划分

- **静态负载均衡**
  - 不可变（元数据:zk描述信息），XML配置文件中指定的. 这个范围是不可变的
  - 动态加载，实时更新 Class instance 更改引用（容器化技术，本地化更新）
- **动态负载均衡**
  - 通过**配置中心**来完成实例的配置管理
  - Dubbo管理平台：**override 协议**，覆盖已有服务的参数信息 配置中心
  - **zk**:长连接，有**推送模式**，节点数据发生变化，推送所有的变化数据给所有的服务
- **智能负载均衡**
  - （不需要人工干预，少人工干预），利用监控平台，监控所有服务的KPI指标，提供策略预案数据。
  - 加入服务治理平台
- **协议负载均衡**

- IP（网络层负载均衡），黑客：牛逼 在协议层

## 负载均衡级别

- 实例级别
- 接口级别
  - Order
- 方法级别
  - 查询订单
  - 生成订单

## 负载均衡种类，以及应对业务场景

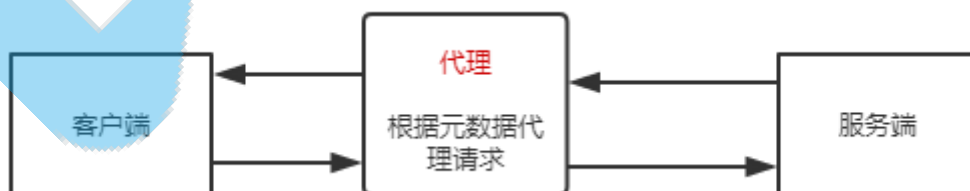
- 二层负载均衡（mac）
  - 二层负载均衡又称为数据链路层负载均衡
  - 外部对应虚拟MAC地址请求，负载均衡接收后再分配后端实际MAC地址服务响应
- 三层负载均衡（VIP）[SNAT(源地址转换，其作用是将ip数据包的源地址转换成另外一个地址)]
  - 三层负载均衡也就是网络层的负载均衡,需要用到网络层的协议，如OSPF协议，RIP协议等
  - 外部对虚拟的ip地址请求，负载均衡接收后分配后端实际的IP地址响应
- 四层负载均衡（tcp&udp）[LVS+keepalive]

三次负载均衡基础上，用vip+port接收请求，再转发到对应的机器



- 七层负载均衡（http）[nginx|Tengine|haproxy]--智能化

根据虚拟的url或IP，主机名接收请求，再转向相应的处理服务器



- HTTP重定向负载均衡

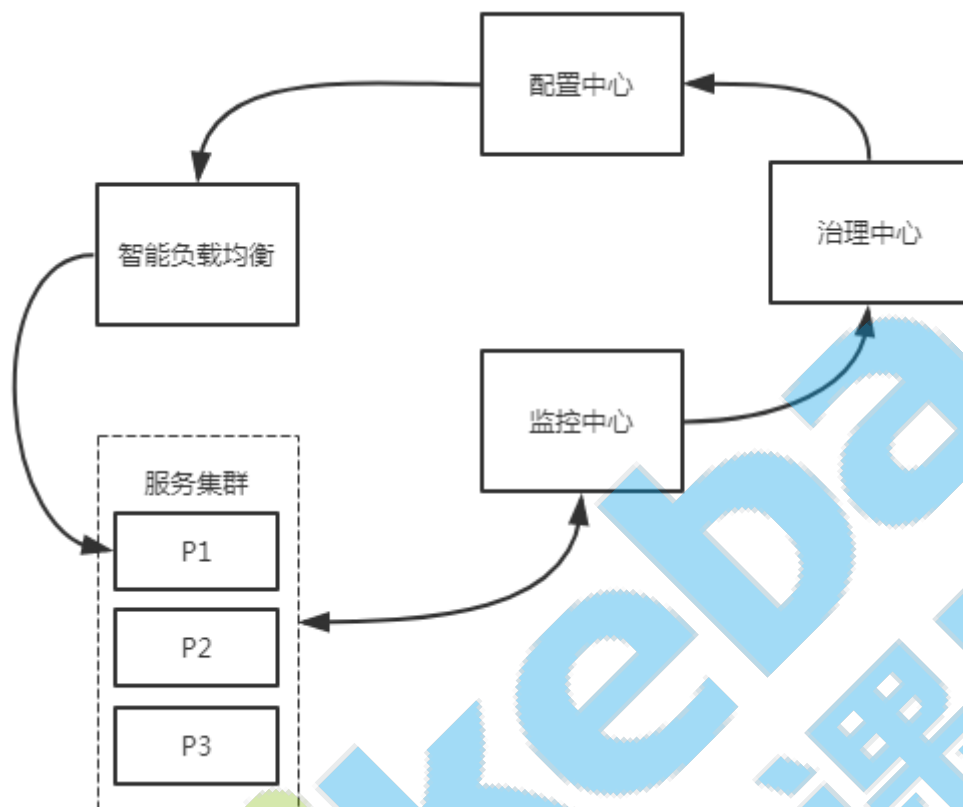
## 负载均衡背后的本质原理

在不对流量进行塑形的条件下，基于系统弹性伸缩能力来应对流量的不稳定性。有效规划服务负载承受能力。负载可用性通过周期性的健康检查【拉】或监听服务【推】来保证

负载均衡本质

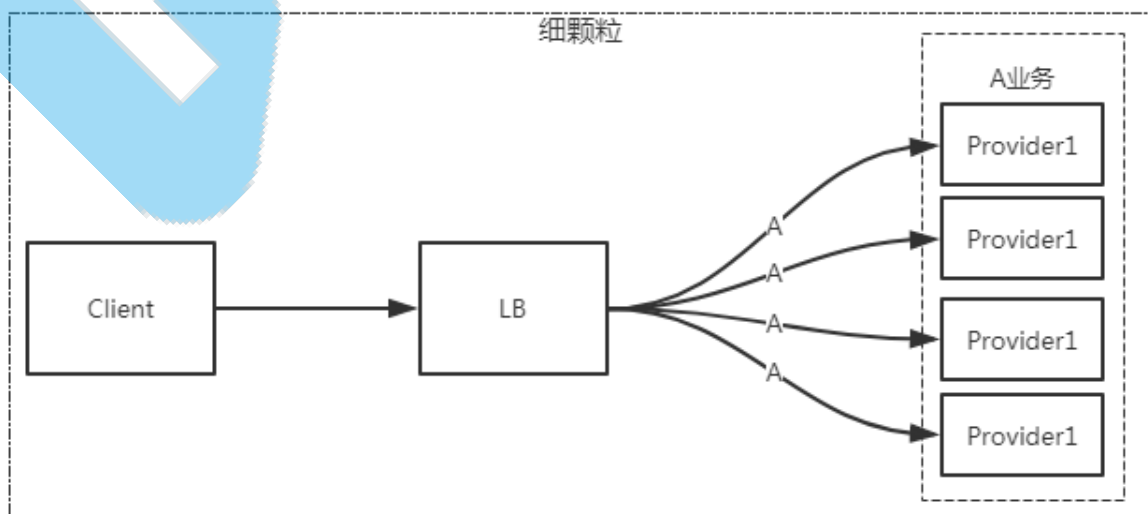


- 分配客户端的请求由哪个服务负责处理。
- 智能化的负载均衡必须配备配置中心和监控中心的支持。



#### 均衡操控原理

- 硬设施
  - 纵向扩展
    - MNDC(修改新建删除创建)
  - 横向扩展
    - 规模化硬件扩种
- 软设施
  - 基于虚拟化公众资源池扩充服务实例



注意：不同方法处理消耗不同

#### XML负载均衡配置

```
<!--服务端服务级别-->
<dubbo:service interface="..." loadbalance="roundrobin" weight="-90" />
<!--客户端服务级别-->
<dubbo:reference interface="..." loadbalance="roundrobin" />
<!--服务端方法级别-->
<dubbo:service interface="...">
    <dubbo:method name="..." loadbalance="roundrobin"/>
</dubbo:service>
<!--客户端方法级别-->
<dubbo:reference interface="...">
    <dubbo:method name="..." loadbalance="roundrobin"/>
</dubbo:reference>
```

**备注：**CSS样式的 优先级最高的是,行内样式，Dubbo最高优先级配置method