

1. 讲师介绍

马中华，2013年毕业于湖南师范大学

前Oracle数据开发技术组负责人，动批网数据运营系统负责人，阿里云/腾讯云全球第一位认证金牌讲师，阿里云栖大会NLP专场特邀演讲嘉宾，混迹大数据江湖多年，精通各大大数据技术，架构经验丰富，在机器学习，自然语言处理方向也有深入研究。有丰富的企业技术培训和技術讲座的实战经验，历史受训学员2000+。

2. 课程大纲

训练营主题：手撸Hadoop

- 1、海量数据的存储及计算方案探索分析
- 2、单线程的通用存储和计算方案设计和实现
- 3、多线程的通用存储和计算方案设计和实现
- 4、多进程的通用存储和计算方案设计
- 5、手写RPC实现

3. 海量业务需求

海量的数据需要处理？存储 计算！

3.1. 存储方面

我小时候在湘西山里边长大，家里会种水稻，种玉米，到了秋收季节，会晒干进行存储。由于外出务工的人越来越多，然后我爷爷种地也越种面积越大，最后家里的仓库不够了，我爷爷的思路，并没有把仓库扩大，还是新建其他新的仓库。因为家里是两层楼的木房子，一楼建仓库，二楼也建仓库。

大数据体系中的各种分布式存储技术组件，设计思想就是来源于我爷爷的这种**分布存储策略**。

3.2. 计算方面

1、我现在，和李老师，和玄姐三个人准备斗地主。但是只有一副旧扑克牌。现在需要确认一下，是否这一副旧扑克牌是否有54张。我的策略很简单：随手就把牌分成三堆，然后每个人数一堆，最后都汇总到我这儿。当然也可以我直接一个人数一数搞定。

2、中铁五局在长沙修地铁3号线。两种方案：

一个施工团队：从头到尾，顺序修过去

多个施工团队：每个施工团队负责一段

结论明显：第二种方案，效率很高。中国的基建狂魔的称号就是靠这些勤劳智慧的中国人民撑起来的。

你看似难理解的各种高大上的算法思维，分布式设计思想等都藏于生活当中。享受生活，善于思考，学大数据架构very easy

计算引擎的核心原理；**分布式并行计算**

3.3. 综合问题

京东自营仓库的问题：并没有在某一个地方选址建造集中式的仓库。而是全国各地建子仓库

1、分布存储（每个不同仓库有相同的商品）

2、高效集散（就近发货）（分布式计算框架：计算的数据本地性）

4. 需求背景

进入正题：海量数据来咯，怎么办怎么办？

先来几个小case热热身：

1、奈学有一个知识问答社区：<https://ask.naixuejiaoyu.com/>，现在一天的所有问题搜索中，奈学的后端运营小姐姐想知道学员们最爱提那个种类的技术问题，所以需要统计，由于这个社区刚上线，所以现在的数据量还不算很大，请帮我提供思路。但是奈学孙老板说了，将来要把这个社区做大，当用户越来越多，搜索量越来越大的时候，那么后端运营的小哥哥小姐姐又该如何处理呢？

简化：

1、现在后台有一个日志文件，每天记录大几千条搜索日志。求每天搜索中，热度最高的10个问题

2、现在后台有100w个日志文件，总大小约摸100T，依然求每天搜索中，热度最高的10个问题

解决方案：一个大问题 拆分成多个小问题来解决（分而治之）并行运行10000个任务。汇总

2、现在后端运营小姐姐又给我提需求了，是这样的：他们想知道昨天和今天都上线搜索了问题的会员有哪些？

简化：找昨天和今天都搜索了问题的共同会员。

1、现在社区后台每天有一个日志文件，每天记录大几千条搜索日志。求昨天和今天的共同会员

2、现在社区后台每天有100w个日志文件，总大小约摸100T，

解决方案：对两天的数据都进行hash散列。第一天的第一个分区，和第二天的第一个分区求交集

3、真讨厌啊，三天之后，运营小姐姐又来提需求了，说，想提供一个判断功能：搜索一个会员ID或者名称，立马返回这个用户最近一周是否登录的信息。如果没有登录，就推送一些高质量的问答。

简化：快速判断在过去7天的日志数据中，判断某个用户ID是否存在

来来来，头脑风暴！我要征服运营小姐姐！快帮我想解决方案。

提供了海量数据的计算解决方案：多个小任务并行执行运算，再执行第二个阶段的汇总通用的套路！

5. 看代码实现

上述三个需求的代码实现：见代码！

6. 为什么需要分布式

为什么需要分布式？

- 1、海量数据在单机上处理因为硬件资源限制，无法胜任
- 2、而一旦将单机版程序扩展到集群来分布式运行，将极大增加程序的复杂度和开发难度
- 3、引入分布式，开发人员可以将绝大部分工作集中在业务逻辑的开发上，而将分布式计算中的复杂性交由框架来处理

设想一个海量数据场景下的数据计算需求：

单机版的缺点：

- 1、磁盘受限
- 2、内存受限
- 3、计算能力受限

分布式版的难点：

- 1、数据存储的问题，设计一个分布式存储组件
- 2、运算逻辑至少要分为两个阶段，先并行计算（map），然后汇总（reduce）结果
- 3、这两个阶段的计算如何启动？如何协调？
- 4、运算程序到底怎么执行？数据找程序还是程序找数据？
- 5、如何分配两个阶段的多个运算任务？
- 6、如何管理任务的执行过程中间状态，如何容错？
- 7、如何监控和跟踪任务的执行？
- 8、出错如何处理？抛异常？重试？

可见在程序由单机版扩成分布式版时，会引入大量的复杂工作。为了提高开发效率，可以将分布式程序中的公共功能封装成框架，让开发人员可以将精力集中于业务逻辑。

编写一个通用的专门用来做分布式通信的框架

7. 手写RPC

- 使用JDK原生BIO（也就是ServerSocket那一套）。阻塞式IO方法，无法支撑高并发。

- 使用JDK原生NIO(Selector、SelectionKey那一套)。非阻塞式IO，可以支持高并发，但是自己实现复杂，需要处理各种网络问题。
- 使用大名鼎鼎的NIO框架Netty，天然支持高并发，封装好，API易用。
- JDK1.7提供的AIO，真正的异步编程模型

BIO编程模型两个大的缺点：

- 1、客户端和服务线程是1:1的情况，服务器的资源不够用
- 2、BIO模型中，有两处阻塞的方法实现：
`serverSocket.accept()`
`in.readLine()`

解决方案：

既然一个服务线程在完成了链接请求之后，会一直阻塞在`in.readLine()`上，等待用户发送请求过来。当前等到请求发送过来的这段时间里面，这个线程处于阻塞。可以在完成链接的时候，把这个服务线程给调度给其他的应用程序执行。

没法通过代码编写来实现！

基于此，JDK1.4提供了NIO编程模型！

NIO入门程序大概是 220行：

- 无论这个东西有多牛，你使用它的体验必定不好！
- 1、代码量非常大
 - 2、编程模型的复杂度还比较高

基于此，JDK1.7再提供一种新的编程模型AIO，异步网络通信

代码复杂要比NIO要一些
AIO的出生比较晚。没有先发优势

基于此，封装NIO提供了一个网络通信框架：netty： <https://netty.io>

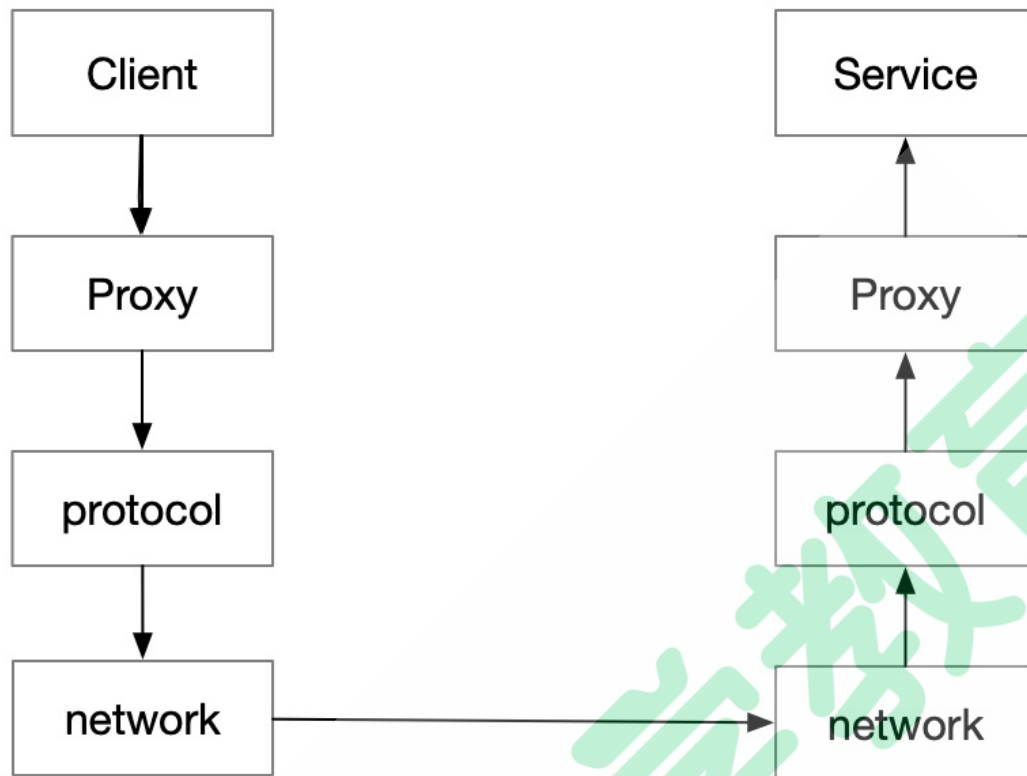
底层基于NIO实现！（Selector, Channel, Buffer）

- 服务端发布了很多的服务：
- 1、定义服务接口
 - 2、实现服务接口
 - 3、服务接口实现类的服务方法，就是客户端想要调用的方法

7.1. RPC 的调用过程

通过下图我们来了解一下 RPC 的调用过程，从宏观上来看看到底一次 RPC 调用经过些什么过程。

当一次调用开始:



完整的过程:

1. client 会调用本地动态代理 proxy
2. 这个代理会将调用通过协议转序列化字节流
3. 通过 netty 网络框架, 将字节流发送到服务端
4. 服务端在受到这个字节流后, 会根据协议, 反序列化为原始的调用, 利用反射原理调用服务方提供的方法
5. 如果请求有返回值, 又需要把结果根据协议序列化后, 再通过 netty 返回给调用方

spark hadoop flink 等等, 这些技术的底层都有一个网络通信框架

hadoop; netty (封装了netty实现了一个RPC)

spark: akka netty

flink: akka

StringDecoder

StringEncoder

使用类似于http协议的形式来实现;

发送请求: 封装一个请求对象

返回结果: 封装一个响应对象

奈学教育:

DP 大数据工程师

DE 大数据架构师

AP

AE

AM

DA

DM

奈学教育

