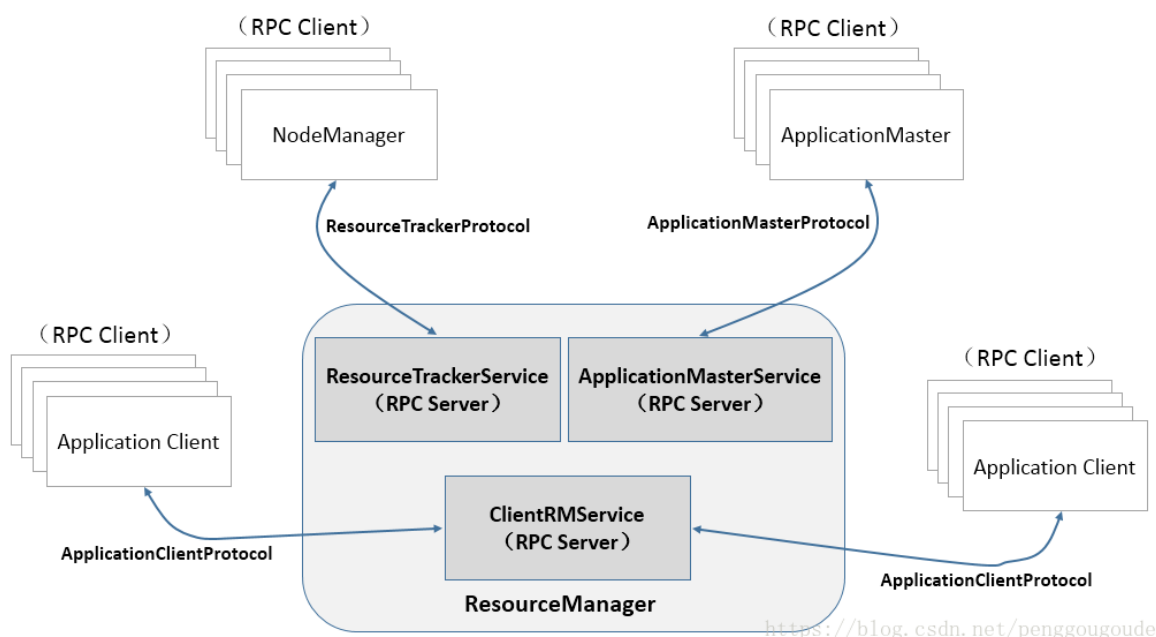


Yarn 通信协议与工作流程

Yarn 通信协议

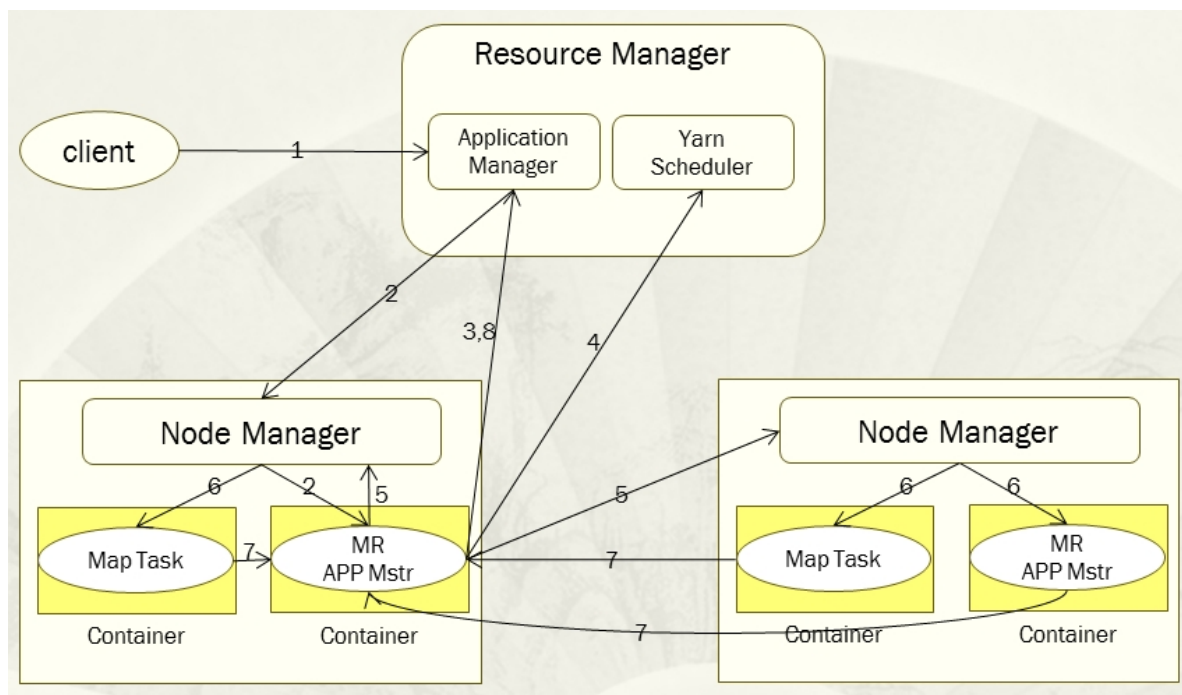
RPC协议是连接各个组件的“大动脉”，了解不同的RPC协议有助于我们更深入的学习YARN框架。在YARN中，任何两个需相互通信的组件之间仅有一个RPC协议，而对于任何一个RPC协议都有一个Client和一个Server，且Client总是主动连接Server的。



Yarn主要由以下几个RPC协议组成：

- JobClient（作业提交客户端）与RM之间的协议——ApplicationClientProtocol: JobClient 通过该RPC协议提交用户程序、查询应用程序状态。
- Admin（管理员）与RM之间的通信协议——ResourceManagerAdministrationProtocol: Admin通过该RPC协议更新系统配置文件，比如节点黑名单、用户队列权限等。
- AM与RM之间的协议——ApplicationMasterProtocol: AM通过该RPC协议向RM注册和撤销自己，并为各个任务申请资源。
- AM与NM之间的协议——ContainerManagementProtocol: AM通过该RPC要求NM启动或者停止Container，获取各个Container的使用状态等信息
- NM与RM之间的协议——ResourceTracker: NM通过该RPC协议向RM注册，并定时发送心跳信息汇报当前节点的资源和Container运行情况。

Yarn的工作流程



YARN流程主要分为一下几步：

1. 第一步：用户向YARN提交用户程序，其中包括了ApplicationMaster程序、启动ApplicationMaster的命令、用户程序等
2. 第二步：ResourceManager为该应用程序分配第一个Container，并与对应的NodeManager通信，要求它在这个Container中启动应用程序的ApplicationMaster
3. 第三步：ApplicationMaster先向ResourceManager注册，这样用户可以直接通过ResourceManage查看应用程序的运行状态，然后它将为各个任务申请资源，并监控它的运行状态，知道运行结束。
4. 第四步：ApplicationMaster采用轮询的方式通过RPC协议向ResourceManager申请和领取资源
5. 第五步：一旦一旦ApplicationMaster申请到资源后，便与对应的NodeManager通信，要求它启动任务。
6. 第六步： NodeManager为任务设置好运行环境（包括环境变量、JAR包、二进制程序等）后，将任务启动命令写到一个脚本中，并通过运行该脚本启动任务。
7. 第七步：各个任务通过某个RPC协议向ApplicationMaster汇报自己的状态和进度，以让ApplicationMaster随时掌握各个任务的运行状态，从而可以在任务失败时重新启动任务。在应用程序运行过程中，用户可随时通过RPC向ApplicationMaster查询应用程序的当前运行状态。
8. 第八步：应用程序运行完成后，ApplicationMaster向ResourceManager注销并关闭自己。