Gluster源码框架

（此文档可能有些瑕疵，如有错误，欢迎指正）

1. 目前来看glusterfsd/src/glusterfsd.c的main为glusterd守护进程Damon、glusterfsd-server、glusterfs-client的同一入口，三种模式在parse\_cmdline（argc，argv）判断处理。
2. Glusterfsd的启动是由damon 执行ret = gf\_system (cmd\_str)来启动的，也就是4.6创建监听端口那一步启动的，并与damon建立连接，它前面的流程与glusterd大体一致
3. Glusterfs的是通过mount server 上的一个卷启动，首先与对应的damon建立连接，获取卷信息，再与glusterfsd连接（glusterfs如何从main中抽离出来，mount时怎么启动还待梳理，此步骤遗留问题较多）
4. 关于translator的概念：它是一个抽象的模块对象，以damon为例我理解它下面卷、rpc、监听端口都可以成为translator
5. 简单介绍一下glusterd的启动流程

①初始化（待深入细化）

1. 创建全局glusterfs\_ctx变量并初始化
2. 初始化全局变量global\_xlator，将THIS指向此结构体
3. 获取环境变量

②开始初始化ctx（待深入细化）

1. 获取线程的数据
2. 获取uuid
3. 初始创建一个8M的iobuf\_pool,它由iobuf\_arena构成，后面可以增加，iobuf\_arena由iobuf组成（每个128k）
4. 创建event\_pool事件池，总大小为16384，拿到这个epool的句柄
5. 创建内存池，目前看有不同类型的内存池，
6. 设置cmd的缺省信息
7. 设置进程资源限制

③解析命令行参数，在此判断是glusterd守护进程Damon、glusterfsd-server、glusterfs-client的哪一种，并做相应处理

④Log初始化

⑤Daemonize，我理解此步骤是进行damon化，即转为守护进程

⑥负责创建监听端口，与监听端口建立连接（此步骤目前问题较多，流程没梳理清楚）

⑦开始监听事件池中注册的句柄，并在有事件发生时调用注册的处理函数来处理(具体待补充)