*摘要: 分布式系统很难，为什么难，其实还是数据一致性的问题。最近在看Erlang中和Mnesia相关的一些东西，想起了这个话题来了。*

万事皆有因

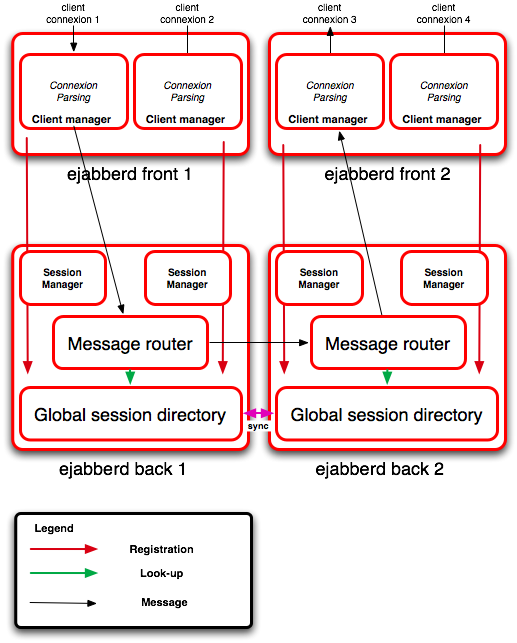
最近IM云平台也好，社交应用也好，大量的使用ejabberd的厂商涌现出来了。不过所有使用ejabberd厂商可能都会遇到Mnesia脑裂的问题。在这里打算简单的谈谈脑裂这个事情。

什么是脑裂

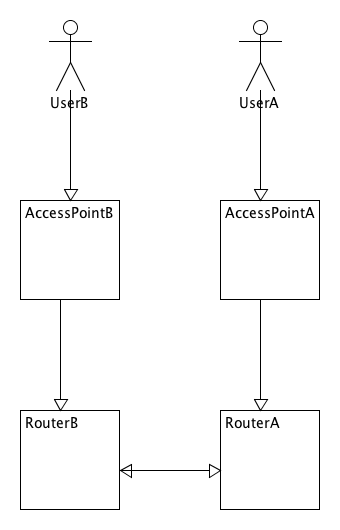
我在这里面给个非官方的定义吧。当一个集群的不同部分在同一时间都认为自己是活动的时候，我们就可以将这个现象称为脑裂症状。我们当如何理解这句话呢？

1. 首先我们需要是个集群。
2. 其次当中有业务是Master－Backup模式或双星模式。也就是说当主节点挂掉了，备用节点需要接管业务或者是两个节点直接有数据同步。

让我们举个例子（图片来自https://blog.process-one.net/angie\_introducing\_flexarch/）：



其中ejabberd front是不存在Router的，只有Client Manager。而ejabberd back是具有Message router 的。那么我们可以将这个图简单的抽象为这样的：



UserA和UserB分别将自己的信息注册在RouterA和RouterB中。RouterA和RouterB使用数据同步（2PC），来同步信息。那么当UserA想要向UserB发送一个消息的时候，需要现在RouterA中查询出UserA到UserB的消息路由路径，然后再交付给相应的路径进行路由。

当脑裂发生的时候，相当RouterA和RouterB直接的联系丢失了，RouterA认为整个系统中只有它一个Router，RouterB也是这样认为的。那么相当于RouterA中没有UserB的信息，RouterB中没有UserA的信息了，此时UserA再发送消息给UserB的时候，RouterA会认为UserB已经离线了，然后将该信息进行离线持久化。

说到这里面估计大家已经明白了脑裂是什么东西了。

怎么解决

1. paxos算法，使用奇数性质的节点来进行表决，必须选出一个说的算的老大，这个集群才能正常工作。
2. 双星模式下，使用专线直连，从硬件上保障。
3. 使用额外的探测节点，当双方直连断开之后，使用一个约定好的共同节点来探测是否是直连故障。

参考：<https://my.oschina.net/u/236698/blog/525413>

### [高可用方案之脑裂问题探讨(原创)](http://czmmiao.iteye.com/blog/1180851)

关于脑裂我们先来看看红帽的文档是如何解释的  
# What does "split-brain" mean?  
"Split brain" is a condition whereby two or more computers or groups of computers lose contact with one another but still act as if the cluster were intact（都认为自己是老大）. This is like having two governments trying to rule the same country. If multiple computers are allowed to write to the same file system without knowledge of what the other nodes are doing, it will quickly lead to data corruption and other serious problems（这将会导致数据冲撞和其他更严重的问题）.  
Split-brain is prevented by enforcing quorum rules（强制投票选举规则-强制选择一个主来决定执行的操作，不能两个或多个不相连接的节点自行决定操作） (which say that no group of nodes may operate unless they are in contact with a majority of all nodes) and fencing（排外） (which makes sure nodes outside of the quorum are prevented from interfering with the cluster).

在“双机热备”高可用（HA）系统中，当联系2个节点的“心跳线”断开时，本来为一整体、动作协调的HA系统，就分裂成为2个独立的个体。由于相互失去了联系，都以为是对方出了故障，2个节点上的HA软件像“裂脑人”一样，“本能”地争抢“共享资源”、争起“应用服务”，就会发生严重后果：或者共享资源被瓜分、2边“服务”都起不来了；或者2边“服务”都起来了，但同时读写“共享存储”，导致数据损坏（常见如数据库轮询的联机日志出错）。

运行于备用主机上的Heartbeat可以通过以太网连接检测主服务器的运行状态，一旦其无法检测到主服务器的“心跳”则自动接管主服务器的资源。通常情况下，主、备服务器间的心跳连接是一个独立的物理连接，这个连接可以是串行线缆、一个由“交叉线”实现的以太网连接。Heartbeat甚至可同时通过多个物理连接检测主服务器的工作状态，而其只要能通过其中一个连接收到主服务器处于活动状态的信息，就会认为主服务器处于正常状态。从实践经验的角度来说，建议为Heartbeat配置多条独立的物理连接，以避免Heartbeat通信线路本身存在单点故障。  
1、串行电缆：被认为是比以太网连接安全性稍好些的连接方式，因为hacker无法通过串行连接运行诸如telnet、ssh或rsh类的程序，从而可以降低其通过已劫持的服务器再次侵入备份服务器的几率。但串行线缆受限于可用长度，因此主、备服务器的距离必须非常短。  
2、以太网连接：使用此方式可以消除串行线缆的在长度方面限制，并且可以通过此连接在主备服务器间同步文件系统，从而减少了从正常通信连接带宽的占用。  
基于冗余的角度考虑，应该在主、备服务器使用两个物理连接传输heartbeat的控制信息；这样可以避免在一个网络或线缆故障时导致两个节点同时认为自已是唯一处于活动状态的服务器从而出现争用资源的情况，这种争用资源的场景即是所谓的“脑裂”（split-brain）或“partitioned cluster”。在两个节点共享同一个物理设备资源的情况下，脑裂会产生相当可怕的后果。  
为了避免出现脑裂，可采用下面的预防措施：  
1、添加冗余的心跳线，例如双线条线。尽量减少“裂脑”发生机会。  
2、启用磁盘锁。正在服务一方锁住共享磁盘，“裂脑”发生时，让对方完全“抢不走”共享磁盘资源。但使用锁磁盘也会有一个不小的问题，如果占用共享盘的一方不主动“解锁”，另一方就永远得不到共享磁盘。现实中假如服务节点突然死机或崩溃，就不可能执行解锁命令。后备节点也就接管不了共享资源和应用服务。于是有人在HA中设计了“智能”锁。即，正在服务的一方只在发现心跳线全部断开（察觉不到对端）时才启用磁盘锁。平时就不上锁了。  
3、设置仲裁机制。例如设置参考IP（如网关IP），当心跳线完全断开时，2个节点都各自ping一下 参考IP，不通则表明断点就出在本端，不仅“心跳”、还兼对外“服务”的本端网络链路断了，即使启动（或继续）应用服务也没有用了，那就主动放弃竞争，让能够ping通参考IP的一端去起服务。更保险一些，ping不通参考IP的一方干脆就自我重启，以彻底释放有可能还占用着的那些共享资源。

参考至：http://surpassdream.blog.51cto.com/1347340/284974  
            http://www1.chinaunix.com/space.php?uid=25715911&do=blog&id=261403  
本文原创，转载请注明出处、作者

参考：http://czmmiao.iteye.com/blog/1180851