# 脑裂产生以及解决办法

****2.2 脑裂****

    　　在高可用（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。两个节点上的HA软件像“裂脑人”一样，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（常见如数据库轮询着的联机日志出错）。

 　　对付HA系统“裂脑”的对策，目前达成共识的的大概有以下几条：

　　　　1）添加冗余的心跳线，例如：双线条线（心跳线也HA），尽量减少“裂脑”发生几率；

　　　　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。

****2.2.1 脑裂产生的原因****

　　一般来说，裂脑的发生有，以下几种原因：

高可用服务器对之间心跳线链路发生故障，导致无法正常通信。

因心跳线坏了（包括断了，老化）。

因网卡及相关驱动坏了，ip配置及冲突问题（网卡直连）。

因心跳线间连接的设备故障（网卡及交换机）。

因仲裁的机器出问题（采用仲裁的方案）。

高可用服务器上开启了 iptables防火墙阻挡了心跳消息传输。

高可用服务器上心跳网卡地址等信息配置不正确，导致发送心跳失败。

其他服务配置不当等原因，如心跳方式不同，心跳广插冲突、软件Bug等。

****提示：**** Keepalived配置里同一 VRRP实例如果 virtual\_router\_id两端参数配置不一致也会导致裂脑问题发生。

****2.2.2 常见的解决方案****

　　在实际生产环境中，我们可以从以下几个方面来防止裂脑问题的发生：

　　同时使用串行电缆和以太网电缆连接，同时用两条心跳线路，这样一条线路坏了，另一个还是好的，依然能传送心跳消息。

　　当检测到裂脑时强行关闭一个心跳节点（这个功能需特殊设备支持，如Stonith、feyce）。相当于备节点接收不到心跳消患，通过单独的线路发送关机命令关闭主节点的电源。

　　做好对裂脑的监控报警（如邮件及手机短信等或值班）.在问题发生时人为第一时间介入仲裁，降低损失。例如，百度的监控报警短倍就有上行和下行的区别。报警消息发送到管理员手机上，管理员可以通过手机回复对应数字或简单的字符串操作返回给服务器.让服务器根据指令自动处理相应故障，这样解决故障的时间更短.

　　当然，在实施高可用方案时，要根据业务实际需求确定是否能容忍这样的损失。对于一般的网站常规业务.这个损失是可容忍的。

****2.3 如何进行脑裂情况监控****

****2.3.1 在什么服务器上进行监控？****

　　在备服务器上进行监控，可以使用zabbix监控，参考[http://www.cnblogs.com/clsn/p/7885990.html](http://www.cnblogs.com/clsn/p/7885990.html" \t "https://blog.csdn.net/varyall/article/details/_blank)

****2.3.2 监控什么信息？****

*备上面出现vip情况：*

         　　1）脑裂情况出现

         　　2）正常主备切换也会出现

****2.3.3 编写监控脑裂脚本****

[root@lb02 scripts]# vim check\_keepalived.sh

#!/bin/bash

while true

doif [ `ip a show eth0 |grep 10.0.0.3|wc -l` -ne 0 ]

then

echo "keepalived is error!"else

echo "keepalived is OK !"

fi

done

         编写完脚本后要给脚本赋予执行权限

****2.3.4 测试 确保两台负载均衡能够正常负载****

[root@lb01 ~]# curl -H Host:www.etiantian.org 10.0.0.5web01 www

[root@lb01 ~]# curl -H Host:www.etiantian.org 10.0.0.6web01 www

[root@lb01 ~]# curl -H Host:bbs.etiantian.org 10.0.0.6web02 bbs

[root@lb01 ~]# curl -H Host:www.etiantian.org 10.0.0.5

web03 www