RHCS 安装也配置

(V1.0)

简要说明

作　者	Able li	拟制日期	2010-11-03
评审人		评审日期	
批准人		批准日期	
发布人		发布尔日期	

All rights reserved

版权所有　侵权必究

目　　录

(V1.0) i

第 1 章 RHCS 可以干些撒？ 1

第 2 章 RHCS 的原理 2

第 3 章 RHCS 编译 7

第 4 章 配置例子与意思说明 12

第 5 章 状态检查与问题确认 21

第 6 章 FAQ 6-22

6.1 启动<Shell>system-config-cluster 时错 6-22

6.2 启动 cman 时出现如下报错 6-22

6.3 GFS Mount 时出现以下问题 6-22

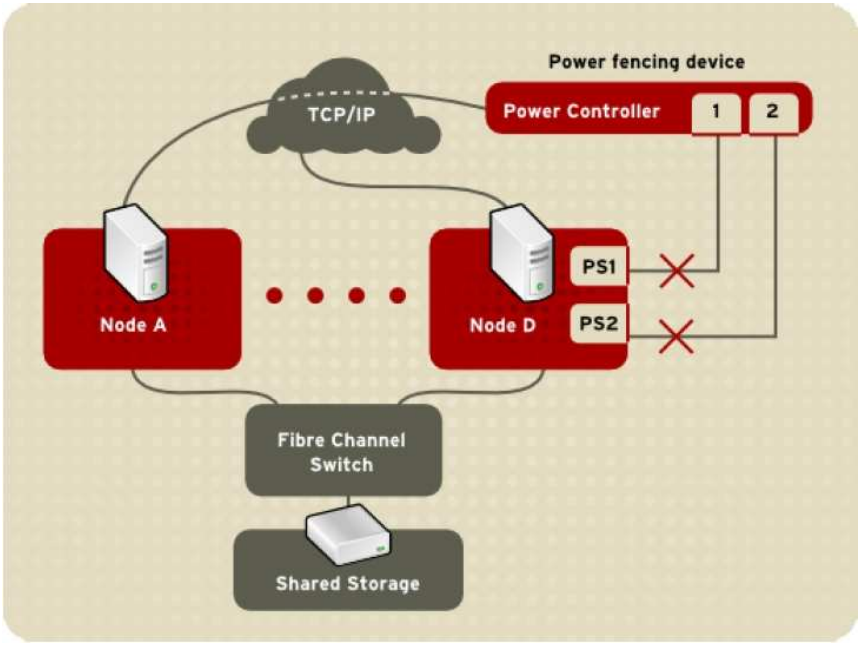
6.4 fence_domain_add: service set level failed 6-23

6.5 GFS＋RHCS 的价格与一些问题 6-23

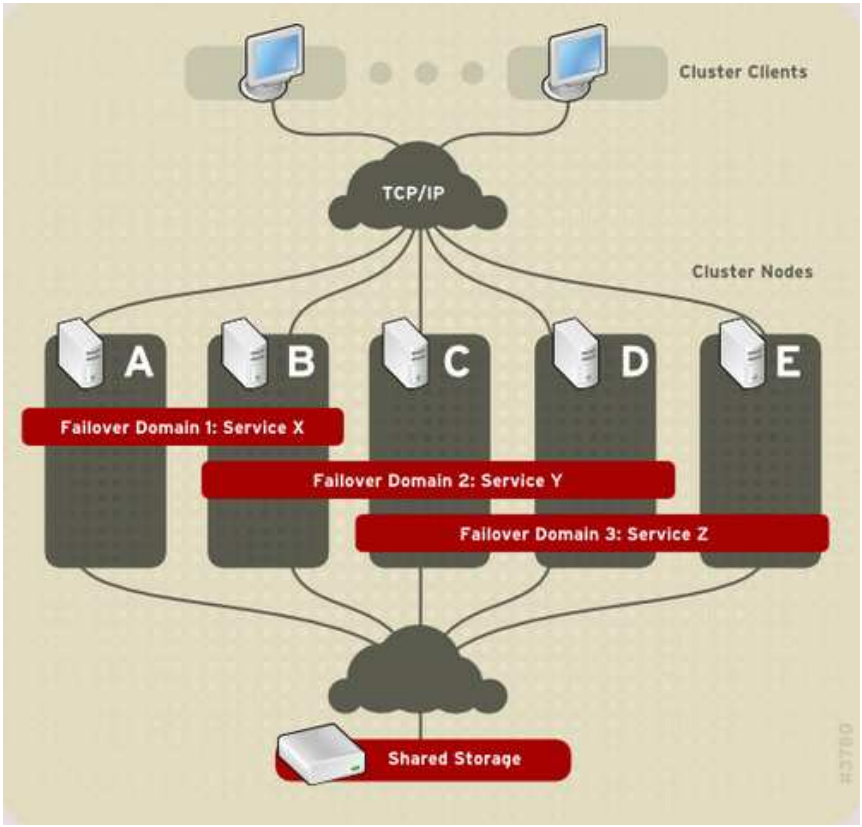
第 7 章 附件 7-23

第1章 RHCS 可以干些撒？

花了一些时间去整理一下这方面的

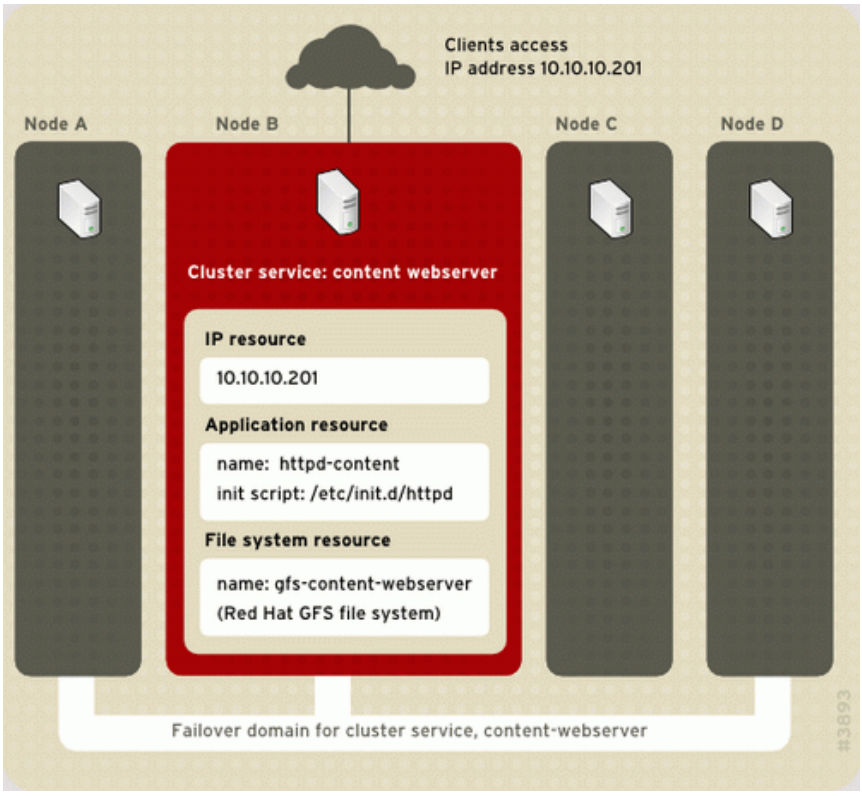
步骤	描述
1	通常企业的数据库服务器都会采用备份机制。因为数据库通常是企业内最重要的服务，一旦数据库服务器因硬件或其它原因停止服务，会造成莫大的损失。所以数据库伺服大多会采用高可用性(High Availability)架构，当原来的数据库服务器无法提供服务时，备份的服务器会马上接管(take over)数据库服务。 先上一张图  这张图整体包括了 RHCS+GFS 的硬件设备与基本的架构
2	分别说明 RHCS 与 GFS： RHCS 的作用是:A 服务器宕机了，有 B 股务器还接着替 A 服务器工作，如果 A，B 都 宕机了，还有 C 来替。就是我们所说的高可用性（HA）。 GFS 的作用：有一个磁盘，不仅 A 能访问而且 B 与 C 也能同时读写（共享存储）， GFS 需要 RHCS 的支持。
3	举例说明： Node A 有 Apache 的服务，要实现当 Node A 宕机后，服务正常提供。不影响客户的使用。就需要当 Node A 宕机后,有服务器 Node B 上的 Apache 顺利接替 Node A,并且 Apache 上的内容一致。

第2章 RHCS 的原理

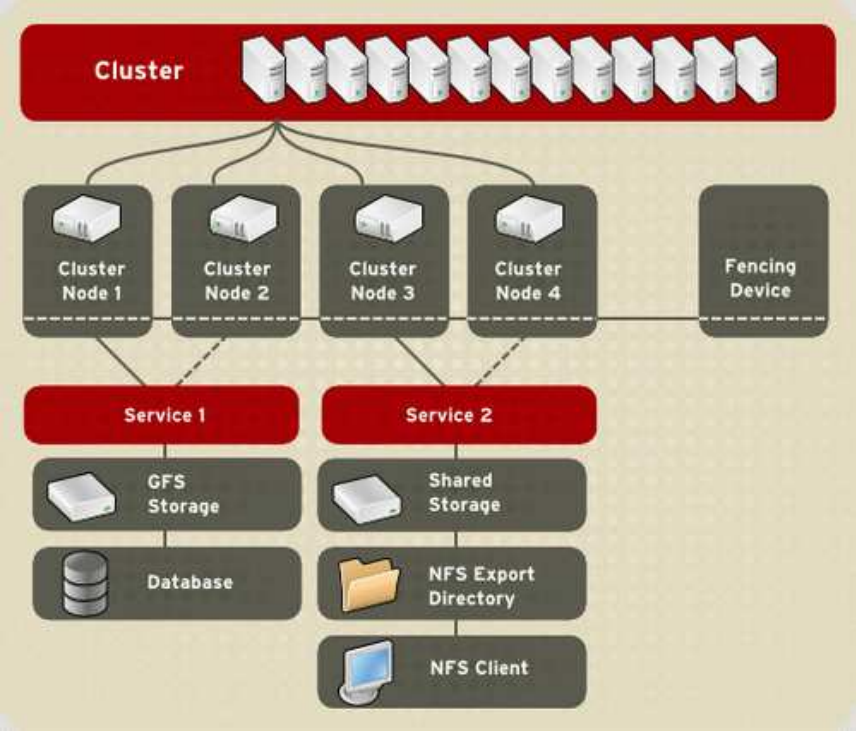
步骤	描述
1	原理这块是一台湾人写的文章，介绍的十分清楚。  图 1 Red Hat 官方文件中有三张图(图 1~图 3)可说是整个 RHCS 架构的精髓，图 1 中有 Cluster Nodes、Failover Domain、Service 三个重要名词，以下分叙之：

步骤	描述
2	Cluster Nodes： Cluster Nodes 就是 Cluster 的成员，就是一台台服务器。以前笔者也提过：Cluster 是什么？一般的定义，「所谓 Cluster 就是由一台以上的机器，为了某种特定需求所组成的架构」，根据不同的需求，可将 Cluster 分为以下三种，并对应 Red Hat 由何种软件提供相关功能。 1.High availability cluster：增加服务器和以网络为基础的应用程序的高可用性及备援性；由 Cluster Suite 中的 Cluster Manager 技术提供。 2.Load Balancing cluster：将服务需求分派给多台服务器，可视系统负载随时弹性增加服务器；由 Cluster Suite 中的 Linux Virtual Server(Piranha)技术提供。 3.High performance cluster(HPC)－提供同步运算及平行处理的能力；Cluster Suite 不提供(另外有 lam、pvm 套件，规划由 WS 担纲)。例如 IBM Cluster 1350 便是一种 HPC 解决方案。 一般的 High availability cluster，通常用两台机器组成 High availability cluster(例如 IBM HACMP 解决方案)，主要服务器停止服务时，备援服务器便会马上接手主要服务器的 service。但 Red Hat 所提供「High availability cluster」解决方案，最好用三台以上的服务器组成 High availability cluster。如果只用两台服务器组成 High availability cluster，则拔除传递 heartbeat 的网络线时，会发生 split-brain(裂脑)的问题，Red Hat 可利用 quorum disk 的技术来解决这个问题，不过这个部份理论较复杂，此篇文章先不讨论此种作法，后面实作的部份，笔者会用 3 个 nodes 来组成 Cluster，其目的是为了避免 split-brain 问题。

步骤	描述
3	Failover Domain: 「Failover Domain」这个名词比较难理解，先看看 Red Hat 官方文件的定义：「A failover domain is a subset of cluster nodes that are eligible to run a particular cluster service.」 简单来说，「Failover Domain」就是 Cluster 的子集(subset)，就是 Cluster 的部份。 由图 1 可以得知： Cluster Nodes 为 nodeA、nodeB、nodeC、nodeD 及 nodeE。 nodeA 和 nodeB 组成 Failover Domain 1。 nodeB、nodeC 及 nodeD 组成 Failover Domain 2。 nodeC、nodeD 及 nodeE 组成 Failover Domain 3。 这样照字面解释完后，我想读者一定还有疑问，那为什么要分「Failover Domain」？还记得笔者在「Single Point of Failure」文章内所提过——「"failover"机制指的是如果一台服务器停机或故障，另一台服务器可以接手(takeover)故障服务器所提供的服务。」 OK！我们再回头看图 1，Failover Domain1 后紧接着 Service X、Failover Domain2 后紧接着 Service Y、Failover Domain1 后紧接着 Service Z。 所以 Failover Domain 的目的是指定 Service X 只能在 Failover Domain1 内移转，也就是 Service X 可在 nodeA 和 nodeB 或上执行。同理 Service Y 在 Failover Domain2 内移转，也就是 Service Y 可在 nodeB、nodeC 或 nodeD 或上执行。 再提醒大家一件事，这是「High availability cluster」，最主要是提供备援机制，Service 同时间只能在某一个 node 上启动，以 Failover Domain1 而言，ServiceX 只能在 nodeA 或 nodeB 上执行，假设一开始是在 nodeA 上执行，当 nodeA 挂掉时，ServiceX 就会自动移转至 nodeB 上。
4	Service: 那什么是 Service？Service 很容易让大家误以为就是/etc/init.d/httpd 这类应用程序，在 RHCS 中的 Service 其实是一堆 Resource 的组合。各位想想，如果主要的 WWW 服务器挂掉，另一台备援的服务器要接管 WWW 服务，只需启动 /etc/init.d/httpd 吗？不对吧！应该还要把原来 Client 所连接的 IP(常称 service IP)移转到备援服务器，还有要把原来的网页档案系统(应该要在 share disk)挂载起来，最后才是执行/etc/init.d/httpd start。经过以上讲解，再看图 2，就会豁然开朗，图 2 中 content service 是由 IP resource(service ip)、Application resource(/etc/inti.d/httpd)及 File system resource(网页档案系统)所组成。

步骤	描述
5	Clients access IP address 10.10.10.201 Node A Node B Node C Node D Cluster service: content webserver IP resource 10.10.10.201 Application resource name: httpd-content init script: /etc/init.d/httpd File system resource name: gfs-content-webserver (Red Hat GFS file system) Failover domain for cluster service, content-webserver #3893 图(2)

步骤	描述
6	Fencing Device : 图 3 中，有个奇怪的装置「Fencing Device」，「Fencing Device」可能是大家最陌生的，因为其它 Unix 的 HA Solution 并不需要此装置。先不要管 Fence 中文意义是什么「栅栏、篱笆」，等一下笔者再解释为什么叫 Fencing Device。RHCS 中提到的 Fencing Device，是一种可以直接对服务器做电源 Power ON/Power OFF 的装置，注意笔者说的是「电源」，不是去执行操作系统的开关机指令。RHCS 支持的 Fencing Device 很多，不过有些并不常见，例如有种电源延长线，你可以用 telnet 指令，对某一个插座做 Power ON/Power OFF 的动作，那么接在这个位置上的服务器不管操作系统是否完全无反应(有时服务器 loading 过重，会连 ssh 也无法登入)都会重开机，这种电源延长线有时我们称作 NPS(Network Power Switch)。不过 NPS，在台湾使用的人很少，就笔者所接触的案子，客户通常是采购本身有支持 IPMI 的服务器，直接用服务器 IPMI 的功能，或是额外购买 IBM RSAII 卡或是 HP iLO 卡。 说了那么多，还是没解释为什么要「Fencing Device」，别急，也看一下原厂文件的说明——「Fencing is the disconnection of a node from the cluster's shared storage. Fencing cuts off I/O from shared storage, thus ensuring data integrity.」为什么 RHCS 需要「Fencing Device」，最主要是避免一种情况发生，什么情况呢？实务上，尤其是数据库服务器可能会遇到这种情况，就是服务器 loading 过重，完全无反应，连 heartbeat 也无法传递，但此时服务器可能不是真的挂掉，笔者开玩笑的说法是「假死」，经过一段时间，loading 没那么重时，又会活过来，heartbeat 又可以传递。但是 RHCS 的机制，只要发现主要服务器的 heartbeat 不通，读者可以把 heartbeat 想成备援服务器每隔时间就会 ping 主要服务器(实际上 RHCS 并不是用 ping 的指令)，就像是去聆听心跳，如果没有回应(Missed too many heartbeats)，就判断主要服务器死了，那么备援服务器便会着手接管 Service。但是万一主要服务器是假死，但 Service 中有包含 File system resource ，那么可能会造成两台服务器同时挂载 File system，就会造成档案系统数据不一致，严重的话可能会毁损整个档案系统。 RHCS 为了避免这种情况，想出了一个办法，就是「Fencing Device」，当 RHCS 发现主要服务器的 heartbeat 不通(Missed too many heartbeats)，第一件事不是着手接管 Service，而是利用「Fencing Device」将主要服务器重开机 Power ON/Power OFF，笔者常开玩笑说，这样一来，假死就变成真死了，也不可能发生主要服务器还挂载 File system resource 的情况。 现在，读者应该可以体会为什么要称为「Fencing Device」，Fencing 有隔离/围起来的意思，就是当服务器假死或是无法 umount File system 时，就把它隔离起来(power reboot)让它无法存取 File system resource (share storage)。

步骤	描述
7	
7	注：没有外部的 Fence Driver 千万不要用于生产

第3章 RHCS 编译

这里我的测试环境使用 Redhat4 update 7.

安装操作系统记得带上 Gcc 与 X-windows..不然在编译过程中会出现一些不明不白的问题。

源码下载

ftp://ftp.redhat.com/pub/redhat/linux/updates/enterprise/4AS/en/RHCS/SRPMS/

http://ftp.isu.edu.tw/pub/Linux/CentOS/4.7/csgfs/SRPMS/

编译好的 RPM

http://linux.web.psi.ch/dist/scientific/46/update/20080807/?C=M;O=D

（这个最好不要用,还是自己编来得实在）

步骤	描述
1	cman-kernel <Shell> rpm -ivh cman-kernel-2.6.9-55.13.el4_7.8.src.rpm <Shell>cd /usr/src/redhat/SPECS <Shell> rpmbuild -ba --nodeps cman-kernel.spec error: Architecture is not included: i386 <Shell>rpm -ivh cman-kern*
2	magma <Shell>rpm -ivh magma-1.0.7-1.src.rpm <Shell>cd /usr/src/redhat/SPECS <Shell>rpmbuild -ba --target i686 magma.spec <Shell>cd /usr/src/redhat/RPMS/i686 <Shell>ls magma-1.0.7-1.i686.rpm magma-debuginfo-1.0.7-1.i686.rpm magma-devel-1.0.7-1.i686.rpm <Shell>rpm -ivh magma-*
3	dlm-kernel <Shell>rpm -ivh dlm-kernel-2.6.9-54.11.src.rpm <Shell>cd /usr/src/redhat/SPECS <Shell> rpmbuild -ba –target i686 dlm-kernel.spec <Shell> cd /usr/src/redhat/RPMS/i686 <Shell> rpm -ivh dlm-kern*
4	dlm <Shell>rpm -ivh dlm-1.0.7-1.src.rpm <Shell>cd /usr/src/redhat/SPECS <Shell>rpmbuild -ba --target i686 dlm.spec <Shell>rpm -ivh --nodeps dlm-1.0.7-1.i686.rpm <Shell>rpm -ivh dlm-debuginfo-1.0.7-1.i686.rpm <Shell>rpm -ivh dlm-devel-1.0.7-1.i686.rpm
5	ccs <Shell>rpm -ivh ccs-1.0.7-0.src.rpm <Shell>cd /usr/src/redhat/SPECS <Shell>rpmbuild -ba --target i686 ccs.spec <Shell>cd /usr/src/redhat/RPMS/i686/ <Shell>rpm -ivh ccs-*
6	gulm <Shell>rpm -ivh gulm-1.0.7-0.src.rpm <Shell>cd /usr/src/redhat/SPECS <Shell> rpmbuild -ba --target i686 gulm.spec <Shell>cd /usr/src/redhat/RPMS/i686/ <Shell>rpm -ivh gulm-*

步骤	描述
7	magma-plugins <Shell>rpm -ivh magma-plugins-1.0.9-0.src.rpm <Shell>cd /usr/src/redhat/SPECS <Shell> rpmbuild -ba magma-plugins.spec <Shell>cd /usr/src/redhat/RPMS/i386/ <Shell>rpm –ivh magma-plugins*
8	cman <Shell>rpm -ivh cman-1.0.17-0.src.rpm <Shell>cd /usr/src/redhat/SPECS <Shell> rpmbuild -ba --target i686 cman.spec <Shell>cd /usr/src/redhat/RPMS/i686/ <Shell>rpm -ivh cman-*
9	fence <Shell>rpm -ivh fence-1.32.63-1.src.rpm <Shell> cd /usr/src/redhat/SPECS <Shell>rpmbuild -ba --target i686 fence.spec Building target platforms: i686 Building for target i686 error: Failed build dependencies: seamonkey-nss-devel is needed by fence-1.32.63-1.i686 seamonkey-nspr-devel is needed by fence-1.32.63-1.i686 redhat 4 update 7 下变为(这个错误提示的不够正确): nspr-devel nss-devel 安装这两个包就可以安装其它包了 <Shell>rpm -ivh seamonkey-* <Shell> rpmbuild -ba --target i686 fence.spec <Shell> rpm -ivh fence-* error: Failed dependencies: perl(Net::Telnet) is needed by fence-1.32.63-1.i686 pexpect is needed by fence-1.32.63-1.i686 <Shell>rpm -ivh perl-Net-Telnet-3.03-1.2.el4.rf.noarch.rpm <Shell> rpm -ivh fence-*
10	iddev <Shell>rpm -ivh iddev-2.0.0-4.src.rpm <Shell> cd /usr/src/redhat/SPECS <Shell> rpmbuild -ba --target i686 iddev.spec <Shell> cd /usr/src/redhat/RPMS/i686/ <Shell> rpm -ivh iddev-*

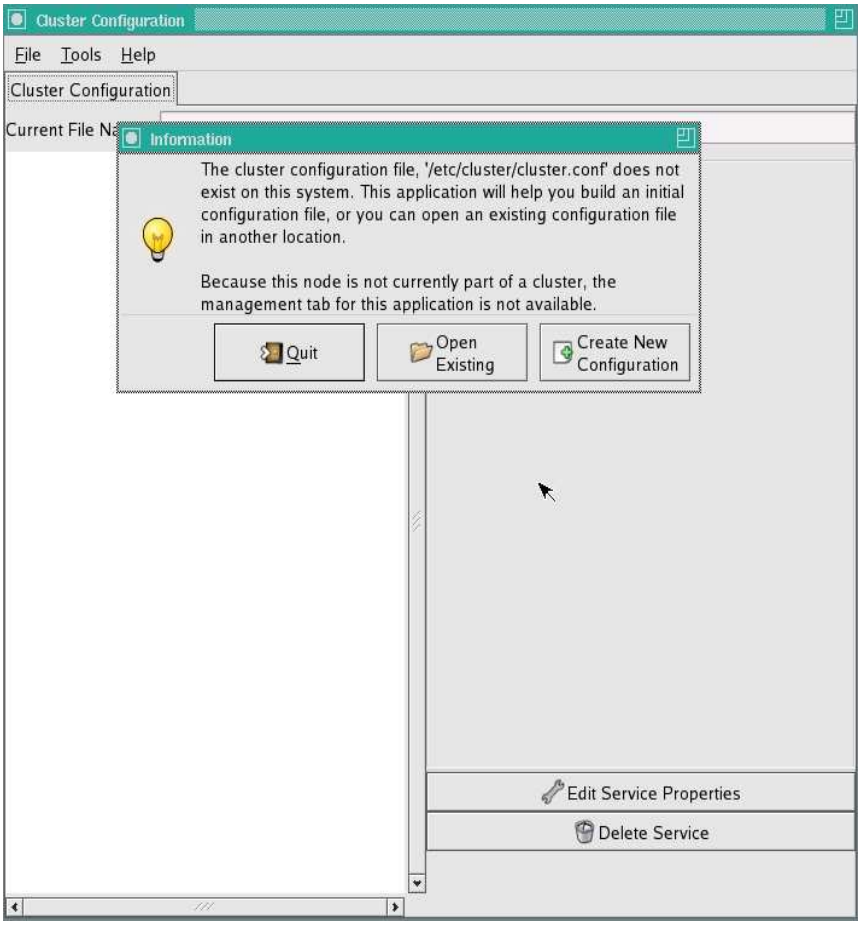
步骤	描述
11	piranha <Shell>rpm -ivh piranha-0.8.3.1-3.src.rpm <Shell>cd /usr/src/redhat/SPECS <Shell> rpmbuild -ba piranha.spec <Shell> cd /usr/src/redhat/RPMS/i386/ <Shell> rpm -ivh piranha-* --nodeps
12	rgmanager <Shell> rpm -ivh rgmanager-1.9.38-0.src.rpm <Shell> cd /usr/src/redhat/SPECS <Shell> rpmbuild -ba rgmanager.spec 版本不对时会使得编译不过 <Shell> cd /usr/src/redhat/RPMS/i386/ <Shell> rpm –ivh rgmanager*
13	system-config-cluster <Shell> rpm -ivh system-config-cluster-1.0.51-2.0.src.rpm <Shell> cd /usr/src/redhat/SPECS <Shell> rpmbuild -ba system-config-cluster.spec <Shell> cd /usr/src/redhat/RPMS/noarch <Shell> rpm -ivh system-config-cluster-1.0.51-2.0.noarch.rpm
14	GFS-kernel <Shell> rpm -ivh GFS-kernel-2.6.9-80.9.src.rpm <Shell> cd /usr/src/redhat/SPECS <Shell> rpmbuild -ba --target i686 GFS-kernel.spec Building target platforms: i686 Building for target i686 error: Failed build dependencies: dlm-kernel is needed by GFS-kernel-2.6.9-80.9.i686 dlm-kernel-hugemem is needed by GFS-kernel-2.6.9-80.9.i686 kernel-xenU-devel = 2.6.9-78.EL is needed by GFS-kernel-2.6.9-80.9.i686 dlm-kernel-xenU is needed by GFS-kernel-2.6.9-80.9.i686 <Shell> cd /usr/src/redhat/RPMS/i686/ <Shell> rpm -ivh GFS-kern*
15	GFS <Shell> rpm -ivh GFS-6.1.18-1.src.rpm <Shell> cd /usr/src/redhat/SPECS <Shell> rpmbuild -ba GFS.spec <Shell> cd /usr/src/redhat/RPMS/i386 <Shell> rpm –ivh GFS-*

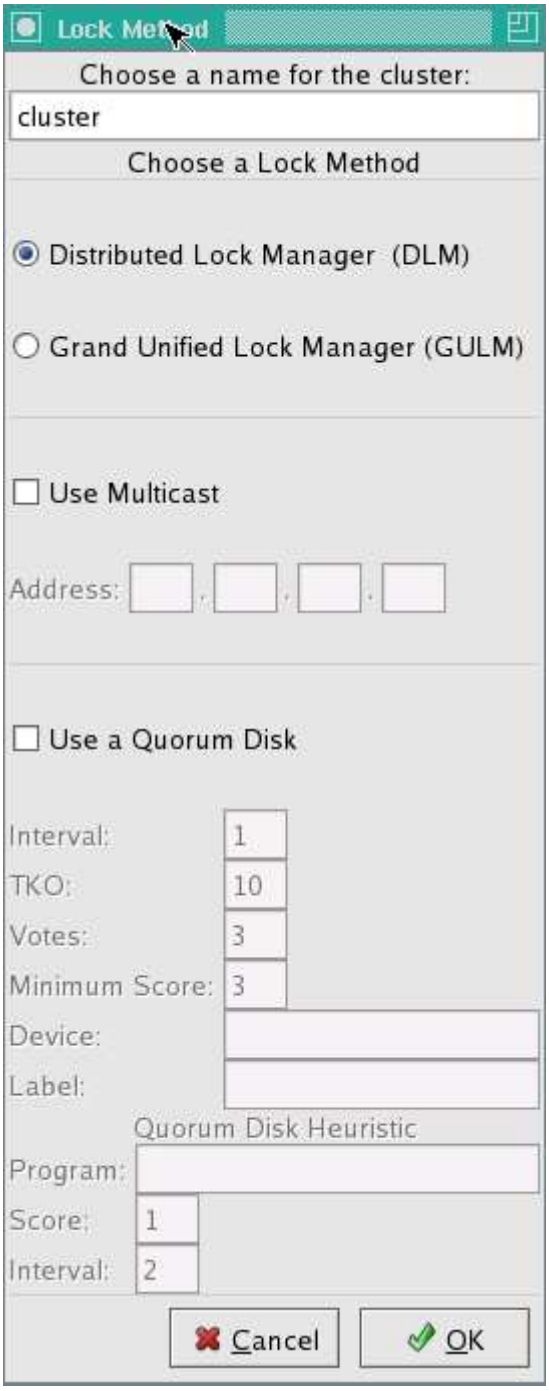
步骤	描述
16	lvm2-cluste <Shell>rpm -ivh lvm2-cluster-2.02.37-3.el4.src.rpm <Shell> cd /usr/src/redhat/SPECS <Shell>rpmbuild -ba lvm2-cluster.spec error: Failed build dependencies: libsepol-devel is needed by lvm2-cluster-2.02.37-3.el4.i386 cman-devel is needed by lvm2-cluster-2.02.37-3.el4.i386 <Shell> cd /usr/src/redhat/RPMS/i386 <Shell> rpm -ivh lvm2-cluster-*
17	ipvsadm <Shell> rpm -ivh ipvsadm-1.24-6.1.src.rpm <Shell> cd /usr/src/redhat/SPECS <Shell> rpmbuild -ba ipvsadm.spec <Shell> cd /usr/src/redhat/RPMS/i386 <Shell> rpm -ivh ipvsadm-*

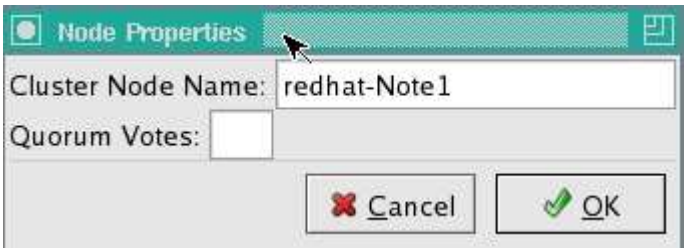
步骤	描述
18	安装顺序： rpm -ivh kernel-hugemem-2.6.9-78.EL.i686.rpm rpm -ivh kernel-xenU-2.6.9-78.EL.i686.rpm rpm -ivh cman-kern* rpm -ivh magma-1.0.7-1.i686.rpm rpm -ivh magma-debuginfo-1.0.7-1.i686.rpm rpm -ivh magma-devel-1.0.7-1.i686.rpm rpm -ivh dlm-kern* rpm -ivh --nodeps dlm-1.0.7-1.i686.rpm rpm -ivh dlm-debuginfo-1.0.7-1.i686.rpm rpm -ivh dlm-devel-1.0.7-1.i686.rpm rpm -ivh ccs-* rpm -ivh gulm-* rpm -ivh magma-plugins-* rpm -ivh cman-1.0.17-0.i686.rpm rpm -ivh cman-debuginfo-1.0.17-0.i686.rpm rpm -ivh cman-devel-1.0.17-0.i686.rpm rpm -ivh perl-Net-Telnet-3.03-1.2.el4.rf.noarch.rpm rpm -ivh pexpect-2.3-1.el4.noarch.rpm rpm -ivh fence-* rpm -ivh iddev-* rpm -ivh piranha-* --nodeps rpm -ivh rgmanager-* rpm -ivh system-config-cluster-1.0.51-2.0.noarch.rpm rpm -ivh GFS-kern* rpm -ivh GFS-6.1.18-1.i386.rpm rpm -ivh GFS-debuginfo-6.1.18-1.i386.rpm rpm -ivh lvm2-cluster-* rpm -ivh ipvsadm-*

第4章 配置例子与意思说明

这里的例子均用 Linux 系统

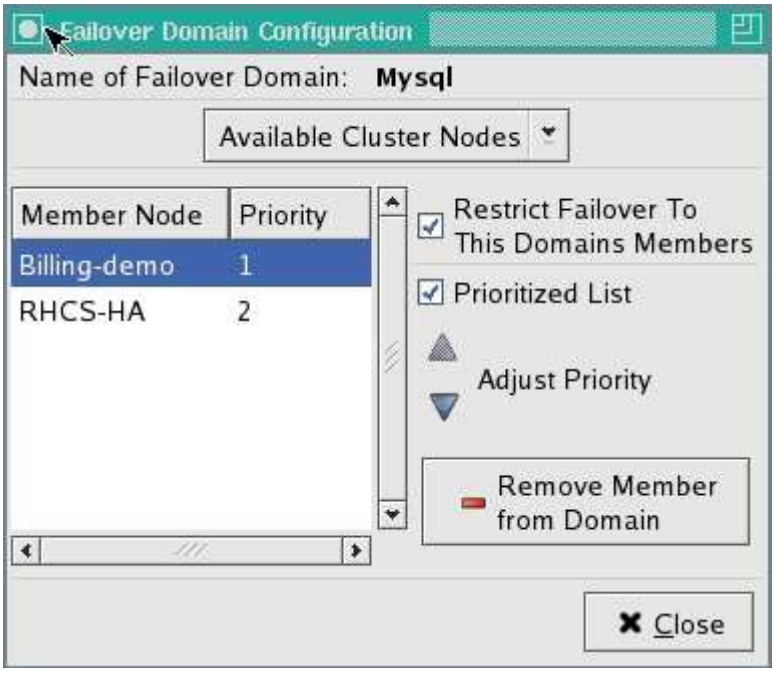
步骤	描述
1	确保相关设备的主机名和 IP 地址的对应关系都在/etc/hosts 中 例： 192.168.10.190 host1 192.168.10.192 host2 192.168.10.193 host3 这个 IP 可以是前端的 IP（服务 IP），最好设置单独的 IP 作为心跳程序。 上面是举例说明，请跟据实际情况配置
2	启动 vncserver（维护起来方便一些） <Shell>system-config-cluster  Create New Configuration.

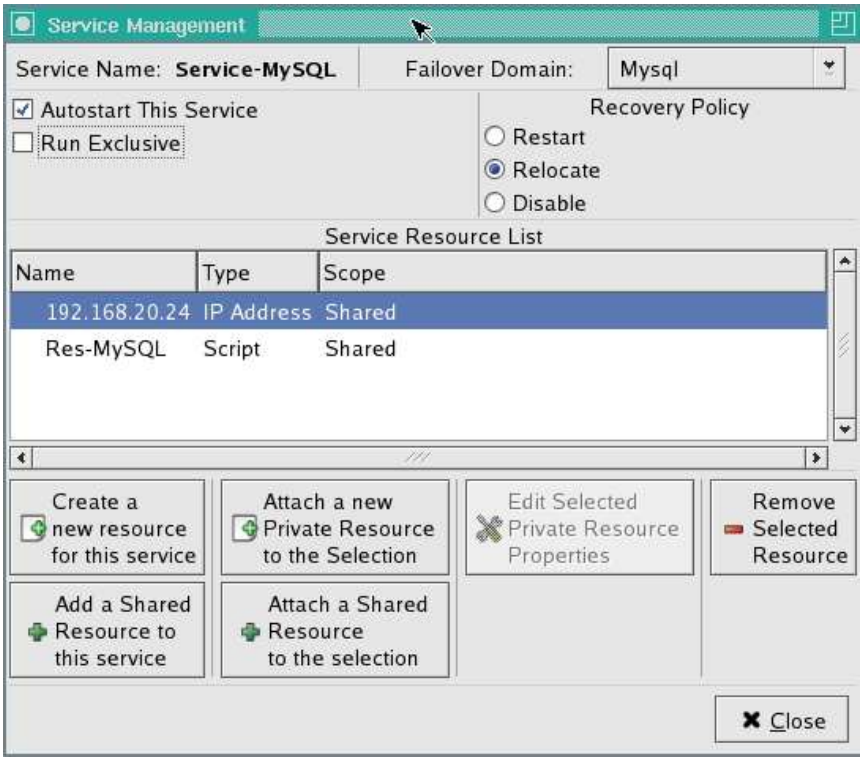
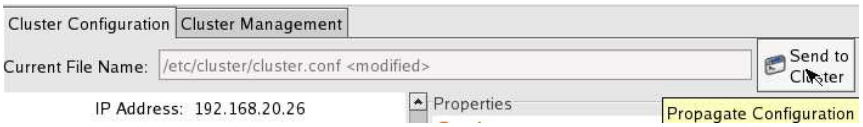
步骤	描述
3	 Choose a name for the cluster… 并选择“DLM”

步骤	描述
4	添加节点： 点击 cluster→cluster Nodes→Add a cluster node  Cluster Node Name 与 uname –n 的值一致，您有多少电脑加入集群就加多少节点吧～ 选择 OK。 依次添加这个集群中的 Node N. ^0^ 注:Node 的名称不要用下划线。在我测试的过程中遇到用下划线的 node，会出现问题。估计是 system-config-cluster 的 bug

步骤	描述
5	添加 Fence Devices(这里添加的手动的 Fence)，但实际用于生产过程中是不可能这样做的。Fence 分为外部与内部。最好用外部的与主机分开的。这样才可以实现宕机切换。  外部 Fence 表： APC Power Switch Brocade Fabric Switch Bull PAP Egenera SAN Controller GNBD IBM Blade Center McData SAN Switch QLogic SANbox2 SCSI Fencing (*See Note) Virtual Machine Fencing Vixel SAN Switch WTI Power Switch http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5/html/Cluster_Administration/s1-config-fence-devices-conga-CA.html

步骤	描述
6	分别创建 Failover Domains: Resources: Services:  如果不知道怎么弄，原理部份，多读几遍就知道了。

步骤	描述
7	Failover Domain Configuration.  这是我建的一个 MySQL 的 Failover Domain。这个失败域中包含有两个 node。分别是 Billing-demo 与 RHCS－HA。Billing-demo.的优先级为 1.RHCS－HA 为 2.就是说 Billing-Demo 宕掉后虚 IP 才会过到 RHCS－AH 上来。 Restrict Failover to This Domains Members 的意思是当 Billing-Demo 与 RHCS-HA 都宕掉后，虚 IP 不会掉到其它的 NODE。不然当这两台 NODE 关掉，虚 IP 会跳转到其它的任意 NODE 上。
8	Resources  这里我配置的 Resouces 只配置了 IP 与服务，当然你还可以配置其它的，如 GFS

步骤	描述
9	Service Management:  Recovery Policy: Restart: 优先重新启动，启动不了再切换 Relocate: 直接切换 Failover Domain 为 MySQL 虚 IP 为:192.168.20.24 Resouce 为 Res-Mysql
10	更新配置文件，并保存  Send to cluster 会自动更新/etc/cluster/cluster.conf 并保存该文件，并同时更新其它 node 的此文件。 如果无法更新请手动更新： 1. Edit /etc/cluster/cluster.conf using the editor of choice. 2. Tell ccs about the change: ccs_tool update /etc/cluster/cluster.conf 3. Find out what version your cluster.conf file currently is from cman's perspective: cman_tool status | grep "Config version" It should come back with something like this: Config version: 37 4. Tell cman your newer cluster.conf is a newer version: cman_tool version -r 38

步骤	描述
11	GFS 配置 因为我每台服务器启动后都需要访问共享存储上的资料，所以每次启动 GFS 将自动启动： 在/etc/fstab 文件中增加下面的列，并创建/gfs 目录。 /dev/sdb /gfs gfs defaults 0 0 如果不加这一行在用:service gfs restart 时看不到成功与失败的标志。 而且是不启动状态的，不知道别的版本是怎么样的。这个是看代码才知道调用的关系的。花了一些时间 并格式化 GFS 磁盘 <Shell>gfs_mkfs -p lock_dlm -t cluster:gfs -j 10 /dev/sdb 解释一下这个命令里面需要注意的： Cluster：与/etc/cluster/cluster.conf 中的创建的 cluster 要一致不然会出现只读的现象。 10：为 GFS 同时的连接数有多大。尽量写大点，以后好扩展。反正没有发现什么坏处。
12	停止与启动进程： service rgmanager stop service gfs stop service clvmd stop service fenced stop service cman stop service ccsd stop service ccsd start service cman start service fenced start service clvmd start service gfs start service rgmanager start

第5章 状态检查与问题确认

步骤	描述
1	Cluster 状态检查： [root@epg-bak etc]# clustat Member Status: Quorate, Group Member Member Name State ID ------ ---- ----- -- RHCS-MASTER Online 0x0000000000000006 Billing-demo Online 0x0000000000000005 Q411 Online 0x0000000000000004 RHCS-HA Online 0x0000000000000003 Gluster Online 0x0000000000000001 epg-bak Online 0x0000000000000002 Service Name Owner (Last) State ------- ---- ----- ------ ----- Ser-vsftpd epg-bak started Ser-ntp Gluster started Ser-upgrade Gluster started Ser-ttcp Gluster started [root@epg-bak etc]# 上面描述了 6 个 node 和 4 个 service.以及 service 所在的 owner
2	Cat /proc/cluster/nodes [root@epg-bak etc]# cat /proc/cluster/nodes Node Votes Exp Sts Name 1 1 6 M Gluster 2 1 6 M epg-bak 3 1 6 M RHCS-HA 4 1 6 M Q411 5 1 6 M Billing-demo 6 1 6 M RHCS-MASTER [root@epg-bak etc]# 文件需要一致，如果那个 node 不一致就重启那个
3	usage: clusvcadm -d <group> Disable <group> clusvcadm -e <group> Enable <group> clusvcadm -e <group> -m <member> Enable <group> on <member> clusvcadm -r <group> -m <member> Relocate <group> [to <member>] clusvcadm -q Quiet operation clusvcadm -R <group> Restart a group in place. clusvcadm -s <group> Stop <group> clusvcadm -v Display version and exit clusvcadm 管理 cluster 一般不会出撒问题
4	
5	
6	

第6章 FAQ

6.1 启动<Shell>system-config-cluster 时错

<Shell> system-config-cluster
Traceback (most recent call last):
File "/usr/sbin/system-config-cluster", line 52, in ?
from ConfigTab import ConfigTab
File "/usr/share/system-config-cluster/ConfigTab.py", line 27, in ?
from ConfigTabController import ConfigTabController
File "/usr/share/system-config-cluster/ConfigTabController.py", line 35, in ?
from FaildomController import FaildomController
File "/usr/share/system-config-cluster/FaildomController.py", line 213
if val == "Yes" or val == "yes" or val="1":
^
SyntaxError: invalid syntax
++
To work around this issue, please open the file /usr/share/system-config-cluster/FaildomController.py with an editing tool(such as vi, gedit, or similar), and find line 213 of this file. This line should be read as follows:
if val == "Yes" or val == "yes" or val="1":
Modify the line to:
if val == "Yes" or val == "yes" or val == "1":
Save the modified file and re-execute the system-config-cluster and it will work.

6.2 启动 cman 时出现如下报错

cman_tool: Can't find broadcast address for node RHCS-MASTER failed
解答:/etc/hosts 里面的 127.0.0.1 和下面的文件电脑名称重复定义了。去掉 127.0.0.1 即可

6.3 GFS Mount 时出现以下问题

[root@RHCS-MASTER cluster]# mount -t gfs /dev/gnbd/gfs /gfs
mount: permission denied
<Shell>dmesg
lock_dlm: cman cluster name "cluster" does not match file system cluster name "cluster1"
lock_dlm: init_cluster error -1
GFS: can't mount proto = lock_dlm, table = cluster1:gfs, hostdata =
GFS: Trying to join cluster "lock_dlm", "cluster1:gfs"
解答:
gfs_mkfs -p lock_dlm -t cluster1:gfs -j 5 /dev/gnbd/gfs
与 cluster 配置的 Clusterl 值不一致。
改成 gfs_mkfs –p lock_dlm –t cluster:gfs –j 5 /dev/gnbd/gfs

6.4 fence_domain_add: service set level failed

如果主机没有启动，在启动 fence 时也会有问题。

```
[root@epg-bak ~]# cat /proc/cluster/nodes
Node  Votes Exp Sts  Name
   1   1    6   M   epg-bak
   2   1    6   M   Billing_demo
   3   1    6   M   Q411
   4   1    6   M   RHCS-HA
   5   1    6   X   RHCS-MASTER
   6   1    6   M   Gluster
```

所有的这个文件出现不一致的情况也会出现这个问题。

6.5 GFS＋RHCS 的价格与一些问题

1.GFS 一年订阅的价格是 15000 元人民币，三年的订阅价格是 43000 元人民币（仅 供参考）
2.RHCS 价格有三个版本，简单来说支持 2 个节点做 HA 的叫基础版： 价格 15000 元 人民币，支持 4 个节点的版本叫标准版 价格 30000 元人民币，高级版最大支持 128 个节点 价格 45000 元人民币（以上全部是一年订阅服务价格，仅供参考）
3.红帽专家服务：10000 元人民币 / 人天（仅供参考）

其它的问题

4.GFS: 最大支持多少个 NODE 的连接？分别描述 32 位与 64 位?
Answer: 红帽 GFS 不分 32 位和 64 位的，最大支持 300 个 Node 的连接（理论上）
5.GFS: 最大支持多大的存储？
Answer：最大支持 2PB ， 也部分 32 位和 64 位的
6.RHCS 最多支持 128 个 Node

第7章 附件