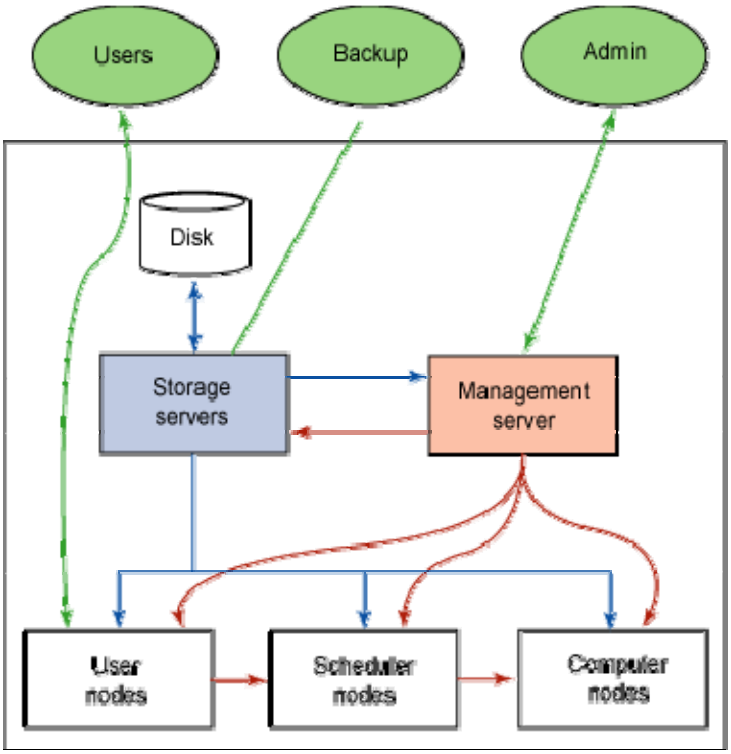
火龙果•整理
uml.org.cn

图 1. 集群架构图



示例集群的一些额外设计和布局细节包括：

- **管理服务器**——管理服务器的功能可以放在一台服务器或多台服务器上。在单台服务器环境中，管理服务器以独立模式运行。也可以设置高可用性管理服务器。可以使用 CSM 高可用性（HA）软件在两台服务器之间进行“心跳测试”，在发生故障情况时管理服务器之间的动态故障屏蔽。引入额外管理服务器的另一种可行方法是：当 HA 在环境中不重要的时候，使用复制设置。在这种情况下，可以把管理服务器的数据备份到其他活动系统，可以通过手动设置将备份系统联机以便接管管理工作（如果有必要的话）。在 图 1 中，管理网络连接用红色显示。管理服务器是 CSM 服务器，它使用 CSM 功能专门控制集群：负责系统安装、监视、维护和其他任务。在这个集群中，只有一台管理服务器。

- **存储服务器和磁盘** —— 可以用多种机制把多台存储服务器连接到基于磁盘的后端。可以用光纤、铜缆或结合使用二者，直接把存储器连接到集群，或者通过存储区域网络（SAN）交换机连接（请参阅 图 1）。这些服务器为集群中的其他服务器提供共享的存储访问。如果需要数据备份，请用额外的铜缆或光纤链路把备份设备连接到存储服务器。对于示例集群，存储的备份端是个单一实体，提供了跨集群的共享文件系统访问。本系列的下一篇文章介绍存储硬件和集群文件系统设置、配置和实现的细节。

- **用户节点** —— 理想情况下，集群的计算机节点不应该接受外部连接，只应当由管理员通过管理服务器访问。系统用户可以登录到用户节点（或登录节点），在集群上运行他们的工作。每个用户节点都包含带有完整编辑功能的镜像、必要的开发工具、编译器和开发支持集群的应用程序和检索结果所必需的所有其他内容。

- **调度器节点** —— 为了在集群上运行工作负荷，用户应当把自己的工作提交到调度器节点。在一个或多个调度器节点上运行的调度器守护程序使用预定的策略在集群上运行工作负荷。与计算机点一样，调度器节点也不应当接受来自用户的外部连接。系统管理员应当从管理服务器管理它们。

- **计算节点** —— 这些节点运行集群的工作负荷，接受来自调度器的作业。计算机节点是集群中最常使用的部分。系统管理员可以轻易地使用管理服务器重新安装或配置它们。

火龙果•整理
uml.org.cn

外部连接 —— 示例外部连接在 图 1 中用绿色显示。这些连接被看作在集群之外，所以本文将不多做说明。

硬件配置

装配好机架并把它们放在适当位置上，连接好全部电缆之后，仍要做大量的硬件配置工作。本文不介绍特定于某种具体集群的特定电缆连接细节。本文使用了上面列出的示例集群设计作为具体示例，说明了在安装集群前所需的硬件配置

逻辑网络设计

在安装集群时最常被忽略的任务就是逻辑网络设计。理想情况下，在实现集群前，应该在纸上进行逻辑设计。有了逻辑网络设计之后，就用它创建主机文件。在小型集群中，如果网络上没有太多设备，可以手工编写主机文件。但是，通常最好是生成一个命名规范，然后编写一个定制脚本，自动生成主机文件。

确保网络上的所有设备都在主机文件中表示出来。请看如下示例（包含示例名称）：

- 管理服务器 (mgmt001 - mgmtXXX)
- 存储服务器 (stor001 - storXXX)
- 计算节点 (node001 - nodeXXX)
- 调度器节点 (schd001 - schdXXX)
- 用户节点 (user001 - userXXX)

这个命名规范只涉及了网络中的五类计算机系统，而且只有一个网络，所以还不够好。还要考虑存储网络和计算机网络，以及设备管理网络。所以这个文件还需要扩展。需要访问集群文件系统的每个节点都需要存储网络上的一个地址。每个节点在计算网络上需要两个地址：一个用于计算地址，另一个用于基板管理控制器（BMC），BMC 用来进行硬件监视和电源控制。表 1 用示例 IP 地址范围描述了更全面的命名规范。

表 1. 主机文件命名规范

Device	Compute 192.168.0.0/24	BMC 192.168.0.0/24	Storage 192.168.1.0/24	Device 192.168.2.0/24	External ext n/w
Management server	mgmt001	mgmt001_d	mgmt001_s	mgmt001_m	mgmt001_
Storage server	stor001	stor001_d	stor001_s	stor001_m	stor001_e
用户节点	user001	user001_d	user001_s	无	无
Scheduler nodes	schd001	schd001_d	schd001_s/	无	无
User nodes	node001	node001_d	node001_s	无	无
Compute switches	无	无	无	gigb01a	无
Storage switches	无	无	无	gigb01b	无
Terminal servers	无	无	无	term001	无
Storage controller A/B	无	无	无	disk01a/b	无

火龙果•整理
uml.org.cn

LCM/KVM/RCM	无	无	无	cons001	无

在实现的时候，这个方案生成与 下载 中能访问的示例类似的主机文件。这个小的示例集群包括 16 个计算节点、一个管理服务器、一个存储服务器、一个用户节点和一个调度器节点，放在两个机架内，并附加了相关的设备。虽然没有提供一个大型集群，但对这个示例集群来说已经足够了，如果需要，可以很容易地把它扩展成更大的集群。

以太网交换机

有两个物理网络：一个用于计算通信量，一个用于存储。每个机架中，标准的 32 个节点需要有两个 48 口交换机，一个交换机负责一个网络。在更小型的集群中，管理机架还需要两个同样的交换机。对于更大的集群，48 端口可能还不够，所以可能需要更大的中心交换机。

两个主网络（不考虑设备管理网络）的每个交换机需要的配置各不相同，因为（就像在示例中那样），千兆以太网间的连接在存储网络中使用巨型帧，而在计算机网络上使用标准大小的帧。设备管理网络的设置通常非常简单：在 10/100 兆交换机上平层的两类网络可以满足设备管理要求，所以无需多做说明。

示例 A： Extreme Networks（极限网络）交换机

下面是 Extreme Networks Summit 400-48t 48 口千兆以太网交换机的配置步骤。

首先，用串口直连电缆连接到每台交换机的串行控制端口（9600、8-N-1、无流控制），使用默认用户 ID admin，无密码。（只需在提示符上按下回车键。）

对于所有交换机，请执行以下步骤：

1. 输入 unconfig switch all —— 如需要，清除现有配置
2. 输入 configure vlan mgmt ipaddress 192.168.2.XXX/24 —— 设置管理 IP 地址。
3. 输入 configure snmp sysname gigbXXX.cluster.com —— 设置交换机名称。
4. 输入 configure sntp-client primary server 192.168.2.XXX —— 把 NTP 服务器设置为管理服务器。
5. 输入 configure sntp-client update-interval 3600 —— 设置时间同步为每小时同步一次。
6. 输入 configure timezone 0 —— 设置时区。
7. 输入 enable sntp-client —— 启动 NTP。
8. 如果需要，输入 configure ports 1-4 preferred-medium copper —— 在端口 1-4 上将默认的首选介质光纤改为铜缆。

现在在存储网络交换机上配置巨型帧，请执行以下步骤：

9. 输入 create vlan jumbo —— 创建巨型帧 vlan。
10. 输入 configure "mgmt" delete ports 1-48 —— 从 mgmt vlan 删除端口。
11. 输入 configure "jumbo" add ports 1-48 —— 把端口添加到 jumbo vlan。
12. 输入 configure jumbo-frame size 9216—— 设置最大传输单元（MTU）的大小。
13. 输入 enable jumbo-frame ports 1-48 —— 开启巨型帧支持。

要在 2 个端口的连接上启用中继，使用 enable sharing 47 grouping 47-48 （把端口 47 和 48 组合起来，47 作为主端口）。

要完成配置，请完成以下操作：

14. 输入 save configuration primary —— 把交换机配置写入 flash，这样重启也不丢失。
15. 输入 use configuration primary

示例 B：Force 10 Networks 交换机

下面是 Force 10 Networks e600 multi- blade Gigabit Ethernet 交换机（有两个 48 端口面）配置步骤，用于路由网络（中央 48 端口交换机还不够大）。

执行以下步骤配置初始层配置的底盘、线卡和端口：

火龙果•整理
uml.org.cn

1. 用串口直连电缆连接到每台交换机的串行控制端口（9600、8-N-1、无流控制），默认下不需要用户 ID 和密码。
2. 输入 enable —— 进入超级用户模式，默认下不需要口令。
3. 输入 chassis chassis-mode TeraScale —— 把交换机初始化为 tera-scale 模式。
4. 交换机提示的时候，重新启动交换机。这要花上几分钟时间。
5. 重启之后，连接到交换机，再次输入 enable 进入超级用户模式。
6. 输入 configure —— 进入配置模式。提示符类似于 Force10(conf)#) 。
7. 输入 Interface Range GigabitEthernet 0/0 - 47 （配置线卡 0 端口 0 到 47，提示符类似于 Force10(conf-if-range-ge0/1-47)#) 。
8. 如果需要，输入 mtu 9252 —— 设置巨型帧。
9. 输入 no shutdown—— 允许激活端口。
10. 输入 exit —— 回到配置模式。
11. 千兆以太网的接口范围是 1/0 - 47。 （提示符类似于 Force10(conf-if-range-ge0/1-47)#)。
12. 对每个线卡重复步骤 7-10。

执行以下操作为第 3 层（Vlan 路由层）配置线卡和端口：

13. 连接到交换机，输入 enable 进入超级用户配置模式。
14. 输入 int port channel 1 —— 配置端口通道 1。
15. 输入 channel-member gig 0/46-47 —— 把线卡 0 的端口 46 和 47 添加到 vlan.
16. 输入 no shutdown —— 允许端口通道激活；这个选项覆盖不活动/活动端口的端口配置。
17. 输入 ip add 192.168.x.x/24 —— 设置端口通道的 IP 地址；这是子网的网关。
18. 如果需要，输入 mtu 9252 —— 设置巨型帧。

现在，打开 DHCP 助手，执行以下操作，跨越子网转发 DHCP 广播：

19. 输入 int range po 1-X —— 把配置应用到刚才配置了的所有通道端口。
20. 输入 ip helper 192.168.0.253 —— 把 DHCP 转发到管理服务器的 IP 地址。

接下来，执行以下操作配置交换机实现远程管理（用 telnet 或 SSH）：

21. 输入 interface managementethernet 0 —— 在配置提示符上配置管理端口。
22. 输入 ip add 192.168.2.x/24 —— 设置设备管理网络上的 IP 地址，并把管理端口连接到设备管理交换机。
23. 设置允许远程连接的用户 ID 和密码。

最后，输入 write mem 保存交换机配置。
交换机配置完成之后，可以在配置上运行完备性检查。请在网络的不同点上插入一台设备，例如笔记本电脑，检查连接性。多数交换机能导出它们的配置。请考虑在网络设置正确之后立即对生效的交换机配置做一个备份拷贝。
之所以描述这两个示例交换机，是因为它们提供了 100% 无阻塞以及高性能千兆以太网交换。Cisco Systems 的交换机不能提供 100% 无阻塞的吞吐，但也能使用。

终端服务器

在使用 CSM 1.4 之前版本的大型集群安装中，终端服务器发挥着重要作用。使用早期版本的集群要靠终端服务器来搜集安装使用的 MAC 地址。由于 CSM 和系统 UUID 的兼容性，终端服务器对于更新的 IBM 集群来说不那么重要。但是 ，如果在大型集群中使用了稍微有些陈旧的硬件，那么在系统设置期间终端服务器仍然至关重要。确保终端服务器本身设置正确，将在后面的安装过程中节省大量时间。除了搜集 MAC 地址，终端服务器还能用来在一个点查看终端从启动自检一直到进入操作系统的情况。

火龙果•整理
uml.org.cn

确保每个端口的终端服务器波特速度与连接计算机的速率匹配。多数计算机默认设置为 9600 波特，所以这可能不是个问题。还要确保终端服务器和连接的每个系统之间的连接设置和流控制相同。如果终端服务器需要验证过的连接，请在 CSM 中设置验证或把验证都关闭。

示例 C: MRV 交换机

下面是 MRV InReach LX 系列交换机的示例配置（请参阅 参考资料 获得关于这个交换机的更多信息。）。请执行以下操作配置 MRV 卡：

1. 用串口直连电缆连接到交换机串行控制端口（9600、8-N-1、无流控制）。
2. 按提示登录。默认用户名是 InReach，自带口令。
3. 输入 enable —— 用默认口令 system 进入超级用户模式。在设备初次配置时会看到配置屏幕。否则，输入 setup 打开配置屏幕。.
4. 输入并保存必要的各种网络参数。
5. 输入 config —— 进入配置模式。
6. 输入 port async 1 48 —— 配置端口 1 到 48。
7. 输入 no authentication outbound —— 关闭内部验证。
8. 输入 no authentication inbound —— 关闭外部验证。
9. 输入 no autobaud —— 选定波特率。
10. 输入 access remote —— 允许远程连接。
11. 输入 flowcontrol cts —— 把硬件流控制设为 CTS，在多数 IBM 计算机上是默认设置。
12. 输入 exit —— 回到配置模式。
13. 输入 exit —— 回到默认超级用户模式。
14. 输入 save config flash —— 保存配置，使配置在重启之后还存在。

初始配置后，就没什么要做的了。同样，要保证所做的设置与连接的计算机上的配置匹配。现在应当能远程登录到终端服务器并对它们进行管理。同处理以太网交换机一样，可以查看运行的配置，以便在终端服务器上对配置进行完备性检查（如果需要的话）。 例如，show port async all char 命令返回终端服务器上每个端口的详细信息。

固件升级和设置 BMC 地址

如果合适，请检查和更新整个集群的固件。请考虑以下元素：

- 计算机 BIOS
- 基板管理控制器（BMC）固件
- 设备固件
- 网络适配器固件

可以在 IBM 支持的 Web 站点上得到 IBM 系统的升级>，特定于厂商的硬件升级通常可以直接从厂商的 Web 站点得到（请参阅 参考资料）。

在 IBM 系统上更新固件

注意：以下更新固件的方法在您所在的地区或硬件上可能不受支持。建议在升级之前与本地的 IBM 代理确认。这个信息只是作为示例之用。

用来远程刷新固件的 CSM 代码仍处在开发之中。目前，如果需要对许多计算机进行 BIOS、BMC 或其他固件更新，就会遇到一个大问题。用当前的方法不能合理地刷新大型集群，需要写入一张软盘或 CD 镜像，并需要对每台计算机分别执行刷新；所以需要另一种方法。如果没有硬件电源控制（没有设置 BMC IP 地址），请先从刷新 BMC 固件开始，使您能够同时设置 IP 地址。只需要一次性按下全部电源按钮。对于其他固件刷新，可以远程打开或关闭系统的电源。

下面的示例针对的是 IBM Systems 325 或基于 AMD 处理器的 326 系统。但是，只要稍做修改，就能把它应用到 System x 计算机。其原理是获得一个默认固件更新镜像，对其进行修改，这样就可以把当用作 PXE 启动镜像。然后，可以通过网络启动一个系统，然后解开镜像，刷新相关固件。一旦把系统设置成 PXE 启动，只需要开启系统就可以进行刷新。

火龙果•整理
uml.org.cn

设置 PXE 启动服务器

在网络上需要一台运行 DHCP 和 TFTP 服务器的计算机。安装并运行 CSM 管理节点的计算机是一个合适的替代方案。但是，如果目前在网络上还没有安装计算机，可以使用一台连接到网络并运行 Linux 的笔记本。确保 PXE 服务器在网络的正确位置（在同一子网内），或者确保交换机能把 DHPC 请求跨子网转发到正确的服务器。然后，完成以下步骤：

1. 用 IP 地址 192.168.0.1 启动 PXE 服务器。
2. 在同一台计算机上安装、配置、启动简单的 DHCP 服务器。下面是个示例配置：

```
ddns-update-style ad-hoc;
subnet 192.168.0.0 netmask 255.255.255.0 {
  range 192.168.0.2 192.168.255.254;
  filename "/pxelinux.0";
  next-server 192.168.0.1;
}
```

3. 安装、配置和启动 TFTP 服务器，在 /tftpboot/ 上运行。安装 syslinux，在 Suse 和 Red Hat Linux 上都以 RPM 包形式提供。
4. 把随 syslinux 包一起安装的 memdisk 和 pxelinux.0 文件拷贝到 /tftpboot/。
5. 创建目录 /tftpboot/pxelinux.cfg/，存放配置文件，创建 /tftpboot/firmware/ 存放固件镜像。
6. 将包含要更新的固件的内容的默认 PXE 配置写入 /tftpboot/pxelinux.cfg/default， 就像下面这样：

```
serial 0 9600
default local
#default bmc
#default bios
#default broadcom

label local
        localboot 0

label bmc
        kernel memdisk
        append initrd=firmware/bmc.img

label bios
        kernel memdisk
        append initrd=firmware/bios.img

label broadcom
        kernel memdisk
        append initrd=firmware/broadcom.img
```

作为参考，当计算机在 PXE 期间接收到 DHCP 地址时，就用特定顺序搜索 /tftpboot/pxelinux.cfg 下的配置文件，第一个找到的文件被用作请求计算机的启动配置。搜索顺序的确定方法是：把请求 DHCP 地址转换成 8 个 16 进制数字，通过扩展子网对配置目录搜索第一个匹配的文件名（每一次搜索后从右到左删除一位数字)。

火龙果•整理
uml.org.cn

例如，假设一台客户计算机在 PXE 启动期间从服务器得到地址 192.168.0.2。第一次文件搜索是搜索这个 IP 地址的 16 进制形式 /tftpboot/pxelinux.cfg/C0A80002。如果这个配置文件不存在，第二个搜索就查找 C0A8000，依次类推。如果没有找到匹配，就用名为 default 的文件。所以，把以上 PXE 配置放在名为 default 的文件可以为所有计算机工作，而不论 DHCP 的配置如何。但是，如果把配置写入 C0A800（192.168.0.0/24 子网）将减少搜索的次数。

更新基板管理控制器（BMC）固件和设置 IP 地址

*注意：*这里描述的产品用于基于 AMD 的集群节点。但是，对基于 Intel 的产品也可以采用类似过程。Intel BMC 的更新由 bmc_cfg.exe 程序（不是 lancfg.exe）提供，用来设置 BMC 地址。可以用终端服务器，使用与 下载 部分提供的脚本类似的脚本驱动该过程。而且，对于基于 Intel 的计算机，通常可以在系统 BIOS 中设置 BMC 地址。

在节点上设置了 BMC 地址之后，就可以进行远程电源控制，这样的话就可以更加轻松地配置集群了。但是，更新 BMC 的这种方法依赖网络启动，所以如果计算机在 BIOS 中没有设置为 PXE 启动，只能先更新 BIOS，然后再回来对 BMC 进行更新。

下载最新的 BMC 固件更新 DOS 镜像，按照说明创建软盘启动镜像。这个镜像包含叫做 lancfg.exe 的程序，用它可以设置 BMC 的 IP 地址。执行的一般过程是插入软盘，并从软盘启动，用软盘进行更新。但是，首先在 PXE 启动服务器计算机上使用以下命令从软盘创建 PXE 启动镜像：

```
dd if=/dev/fd0 of=/tftpboot/firmware/bmc.img bs=1024
```

现在可以根据需要编辑 DOS 镜像。对于 BMC 更新来说，基本镜像本身不需要做修改，只要把 DOS 关机程序拷贝到镜像。从高层次来看，在计算机加电时，计算机通过 PXE 启动来刷新 BMC 固件，然后让计算机在 DOS 镜像中运行。可以用一个脚本来通过终端服务器设置 BMC 地址，然后关闭计算机。这样，就可以知道所有开着的计算机正在刷新 BMC 固件，或者等待设置 IP 地址。在所有关闭的计算机上，将结束该过程。请下载合适的基于 DOS 的关机命令，例如 atxoff.com 实用工具。有了关机实用工具之后，请按照下面的步骤把它拷贝到镜像中：

```
mount -o loop /tftpboot/firmware/bmc.img /mnt
cp /path/to/poweroff.exe /mnt
umount /mnt
```

现在修改适当的注释，在前面创建的 /tftpboot/pxelinux.cfg/default 文件中设置 BMC 为默认项，确保 PXE 启动配置文件能发送正确的镜像。测试了一个节点之后，从关机状态启动所有计算机，这样所有需要的节点就都能得到更新。所有节点都启动了 PXE 镜像之后，请把配置修改回本地启动，从而最小化意外刷新计算机的情况（如果其中一个被重新启动的话）。

现在可以调用 lancfg 程序，并通过终端服务器操作它（假设 BIOS 设置通过串口导出了终端，与终端服务器配置相同）。可以在 Perl 脚本中用 lancfg 配置 BMC IP 地址，例如 下载 部分中可用的不支持的示例脚本。例如，要在网关地址 192.168.10.254、子网掩码 255.255.255.0 的名为 Rack1 的节点组中设置所有计算机的 BMC 地址，请从 PXE 启动服务器计算机上运行以下命令：

```
perl set-bmc-address.pl -N Rack1 -g 192.168.10.254 -m 255.255.255.0
```

可以根据自己的设置定制这个脚本。脚本完成时，计算机在设置完 BMC IP 地址之后，用拷贝到启动镜像的 DOS 关机程序自动关闭。

更新 BIOS

如果将默认 BIOS 设置应用到所有计算机，那么可以在上面的 BMC 更新之前实现此步骤。刷新 BIOS 的过程包含两个阶段，如果执行没有被修改的话，将应用出厂的默认设置。所以，需要刷新并应用带有集群所需的修改的新配置。请下载最新的 BIOS 更新 DOS 镜像，并按说明创建软盘启动镜像。

需要一份针对自己需要的 BIOS 等级和设置的保存好的配置。为此，请手动更新一台计算机。用软盘镜像启动一台计算机（如果计算机没有软驱，可以使用 USB 软驱）。请根据自述文件应用更新，并等候更新正常完成。重新启动计算机，在 BIOS 中对需要的设置做修改。需要考虑的选项有：要关闭数字锁定键（如果在

火龙果•整理
uml.org.cn

键盘上没有数字键盘）、启用串口、使用匹配终端服务器的设置通过串口设置控制台重定向、设置启动顺序（保证 Network 在 Hard Disk 之前）。完成修改之后，保存修改，关闭计算机。
在另一台计算机上（例如设置了 PXE 启动的计算机），装入包含 BIOS 更新的软盘。重新命名 autoexec.bat 文件，将其作为软盘上的备份，供以后使用。这可以防止再次使用这个磁盘启动系统时，再次刷新系统。把磁盘插回刚才更新完并设置好 BIOS 配置选项的计算机，从修改后的软盘镜像启动。
出现 DOS 提示符时，请确保当前工作目录是驱动器 a: 。在软盘上有个程序叫做 cmosram.exe，用它可以把 BIOS 的配置保存到磁盘。运行这个程序把 BIOS 设置保存到软盘，如下所示：

```
cmosram /load:cmos.dat
```

设置放入 autoexec.bat 文件之后，就可以应用更新了。作为完备性测试，请在一台计算机上测试软盘镜像，查看刷新是否自动进行，是否应用了正确的配置。可以注意到系统在 BIOS 刷新之后仍然开着。可以采用 BMC 更新节 中类似的方法，通过使用从 autoexec.bat 文件中调用的 DOS 关机工具，在 BIOS 刷新后自动关闭系统。
对修改后的 BIOS 更新镜像满意之后，就可以用以下命令用软盘创建 PXE 启动镜像：

```
dd if=/dev/fd0 of=/tftpboot/firmware/bios.img bs=1024
```

修改默认的 PXE 启动配置文件 /tftpboot/pxelinux.cfg/default，让它在系统进行 PXE 启动时提供 BIOS 镜像。现在，开启连接到网络的系统，系统就会自动刷新 BIOS 而不需要用户输入，应用正确的 BIOS 设置，并自动关机。更新完成后，把默认 PXE 启动配置改为从本地磁盘启动，防止计算机再次发出 PXE 请求时发生意外。

更新 Broadcom 固件
更新完 BMC 固件和 BIOS 后，更新 Broadcom 固件只是相同概念的简单重复。请按以下步骤操作：

1. 下载 Broadcom 固件 （请参阅 参考资料），并按说明创建软盘启动镜像。
2. 用以下命令从软盘创建 PXE 启动镜像： dd if=/dev/fd0 of=/tftpboot/firmware/broadcom.img bs=1024″
3. 用以下命令循环装入镜像文件：mount -o loop /tftpboot/firmware/broadcom.img /mnt
4. 把基于 DOS 的关机程序拷贝到镜像目录。
5. 修改 autoexec.bat 文件，用无人参与模式自动更新 Broadcom 固件，并在更新完成时自动关闭计算机。例如，对于 IBM Systems 326 ，机器类型为 8848，autoexec.bat 文件可能像下面这样：

```
@echo off
call sramdrv.bat
echo.
echo Extracting files...
call a:\bin.exe -d -o %ramdrv%\update >NULL
copy a:\command.com %ramdrv%\command.com
copy a:\atxoff.com %ramdrv%\atxoff.com
set COMSPEC=%ramdrv%\command.com
if exist NULL del NULL
%ramdrv%
cd \update
call update.bat 8848
cd \
atxoff
```

火龙果•整理
uml.org.cn

6. 卸载镜像。
7. 检查默认配置 /tftpboot/pxelinux.cfg/default，确保计算机能为 Broadcom 适配器启动固件更新。
8. 启动需要更新的计算机。
9. 把配置恢复到本地磁盘 PXE 配置。

跨集群更新了固件之后，可以继续硬件的设置，因为有了最新的固件代码，很少会出现问题。但是，如果需要其他更新，随时可以重复更新过程。而且，这类固件更新使用的原则也可以应用到需要刷新的其他类型的固件，只要能从 PXE 启动得到固件更新。

第 2 部分：配置管理服务器和安装节点

配置管理服务器

设置集群的软件需要两个阶段：首先，按照本系列第 1 部分所述安装集群管理服务器，然后，安装集群的剩余部分，如 "安装节点" 一节所示。遵循这样的步骤进行操作让您可以使用管理服务器来帮助配置集群的剩余部分，而且会非常有助于安装后的维护和操作。

安装 Linux

在管理服务器上安装全新的操作系统。确定管理服务器具有如下配置。本例使用的是 System x 346 机器，一款典型的 IBM 管理服务器，其上运行 Red Hat Enterprise Linux (RHEL) 3，当然您也可以使用运行其他 Linux 版本，比如 Suse Linux Enterprise Server (SLES)，的其他计算机。System x 346 具有 64 位处理能力，所以安装的操作系统是 RHEL 3 的 x86_64 架构，并带 ServeRAID 7k 卡和双磁盘镜像。当然，您的环境可以与之不同，但安装 CSM 管理服务器的基本配置应该大体与之类似。

用最新的 IBM ServeRAID 支持 CD 引导服务器来为 RAID 1（镜像）配置板载磁盘，前提是服务器内至少有两个磁盘，并且需要为操作系统启用磁盘故障保护。

将磁盘配置为单一镜像之后，用第一个 RHEL CD 引导服务器来安装 RHEL 操作系统。根据控制台的不同，您可能需要更改安装的外观。例如，对于一个低分辨率的控制台，您可能需要通过在引导提示符后键入 linux vga=normal 来引导 CD。当看到 Linux 安装 GUI 时，请按照以下指导正常进行安装：

1. 选择语言、键盘图、鼠标类型等。

2. 配置磁盘分区，如下所示：
   - 128Mb /boot 主分区。
   - 2 GB 交换分区。
   - 将剩余空间分配给 LVM 分区且不进行格式化
3. 执行逻辑卷 (LVM) 设置，如下所示：
   - 命名卷组系统。
   - 添加逻辑卷，如 表 1 所示。
4. 设置网络接口，如下所示：
   - 根据以上的示例主机文件，激活 boot 上具有固定 IP 地址 192.168.0.253/24 的 eth0 。
   - 将主机名设置为 mgmt001.cluster.com.。
   - 在此阶段，不需要设置网关/DNS ，如果有外部 IP 信息，可以在安装过程中再对其进行配置。
5. 将防火墙设置为 no firewall 以允许所有连接。如果需要 IP 表，可以在后面对之进行配置。

6. 应用本地设置，选择合适的时区。

7. 设置 root 密码；本例中的密码是 cluster。

8. 定制安装包使之包括如下内容：
   - X Window 系统
   - KDE（即，K 桌面环境）

火龙果•整理
uml.org.cn

- Graphical internet
- 服务器配置工具
- FTP 服务器
- 网络服务器
- 遗留软件开发
- 管理工具

9. 开始安装。

表 1. 逻辑卷布局

逻辑卷	挂载点	大小
Root	/	8192 MB
Var	/var	8192 MB
Usr	/usr	8192 MB
Opt	/opt	4096 MB
Tmp	/tmp	2048 MB
Csminstall	/csminstall	10240 MB

安装完成后，需要逐一处理所有安装后的设置屏幕。针对您自己的环境对管理服务器进行安装后的定制。比如，您可能需要配置 X 服务器使之适合您自己的 KVM（键盘、视频和鼠标）设置。

安装 CSM

在支持系统上安装 Cluster Systems Management (CSM) 软件总的来说多少比较简单。在 IBM Linux 集群文档库（参见 参考资料）可以找到 HTML 格式和 PDF 格式的相关文档。

第一步是将软件复制到管理服务器。因为必须要以 root 用户身份执行安装，所以可以将其存储到 root 根目录。表 2 给出了一种合适的目录结构。

表 2. CSM 软件

目录	描述
/root/manuals/csm/	PDF 格式的 CSM 文档
/root/manuals/gpfs/	PDF 格式的 GPFS 文档
/root/manuals/rsct/	PDF 格式的 RSCT 文档
/root/csm/	CSM 软件（CSM 压缩包的内容）
/root/csm/downloads/	开源 RPMS for CSM 下载（例如 autorpm）

要安装 CSM，需安装 csm.core i386 RPM 包。该包也适用于 the x86_64 架构。安装完该包之后，就可以安装 CSM 管理服务器了。首先，将 /etc/profile.d/Csm.sh 引入到当前的 shell 来获得新的路径设置。然后，运行 installms 命令并将 CSM 许可证应用到系统。以下是需要输入的命令：

```
rpm -ivh /root/csm/csm.core*.i386.rpm
. /etc/profile.d/Csm.sh
installms -p /root/csm/downloads:/root/csm
csmconfig -L <Your License File>
```

注意：如果没有 CSM 许可证文件，同样可以运行 csmconfig -L 命令接受 60 天的试用许可证。60 天的试用期过后，您必须要应用正式的 CSM 许可证以继续 CSM 功能。

针对大型集群进行优化

火龙果•整理
uml.org.cn

CSM 本身就具有可伸缩性。Red Hat Linux 在许多标准条件下也能很好地正常工作。但为了让大型的集群环境能顺利运行，还是需要对管理服务器进行一些优化。以下列举了优化方法的一些样例：

侦听在特定接口上的 DHCP 请求。

编辑 /etc/sysconfig/dhcpd DHCPD 配置文件以使 DHCPDARGS 设置为合适的接口。变量 DHCPDARGS 存在于 Red Hat Linux 的 /etc/init.d/dhcpd DHCPD 启动脚本中，用来启动具有特定参数的 DHCP 守护程序。为了能够在 eth0 侦听，请确保多个参数均已包含在引号之内，如下所示：

```
DHCPDARGS="eth0"
```

增加 ARP 表大小和超时设置。

如果一个很大的网络的同一个子网上具有很多或全部集群，ARP 表就可能会过载，就会让人觉得 CSM 和网络请求的响应速度太慢。为了避免上述现象，需要对运行中的系统做如下更改，并将这些更改添加到 /etc/sysctl.conf 文件，以使更改能够持久有效：

```
net.ipv4.conf.all.arp_filter = 1
net.ipv4.conf.all.rp_filter = 1
net.ipv4.neigh.default.gc_thresh1 = 512
net.ipv4.neigh.default.gc_thresh2 = 2048
net.ipv4.neigh.default.gc_thresh3 = 4096
net.ipv4.neigh.default.gc_stale_time = 240
```

增加 NFS 守护程序的数量。

默认地，标准的 CSM 扇出值为 16。这意味着跨集群运行的命令会一次运行 16 个节点，节点安装也是如此。Red Hat Linux 的标准 NFS 设置是 8 个可同时运行的守护程序。您可以扩展 NFS，方法是增加 NFSD 线程数到 16 以匹配默认的 CSM 扇出值。如果增加扇出值，您可能也会想增加 NFS 的线程数。一般地，扇出值为 32、NFS 线程数为 32 对于保证速度和可靠性已经足够，而且还能允许同时安装具 32 节点的单个 rack。要实现这一目的，创建配置文件 /etc/sysconfig/nfs 并添加如下行：

```
RPCNFSDCOUNT=16
```

设置 NTP 服务器。

默认的 Red Hat Linux 配置应该也适用于 NTP 服务器。向 /etc/ntp.conf NTP 配置文件添加一个配置行以允许集群网络上的节点能够与管理服务器的时钟同步，如下所示：

```
restrict 192.168.0.253 mask 255.255.255.0 notrust nomodify notrap
```

如果管理服务器能够连到一个外部的时钟服务器，通过添加如下一行来将该管理服务器的时钟与这个外部时钟服务器同步：

```
server server.full.name
```

通过如下指令确保 NTP 服务器在运行并会在引导时自动启动：

```
chkconfig ntpd on
service ntpd start
```

安装节点

完成所有的设置和配置步骤之后，CSM 管理服务器就安装完毕了。在安装节点之前，需要在 CSM 管理服务器上完成一些额外的配置来定义节点安装的方式。在 CSM 管理服务器上执行本节所示的安装步骤。

火龙果•整理
uml.org.cn

定义节点

可以参考有关定义节点手册页来定义节点。但定义大量节点的一种更为简单的方法是通过节点定义文件。具体做法是先创建一个 stanza 文件，然后将它作为参数传递给 CSM 来定义所有列出的节点。创建 stanza 文件很容易。

清单 1 给出了一个简短的示例节点定义文件。如果其他节点具有与之相同的属性，可以在默认 stanza 文件的顶部定义它们。之后，每个 stanza 都应该代表一个节点名，下跟特定于节点的属性。这个例子显示了应该如何定义示例集群中的这三个机器 —— 两个计算节点，一个存储服务器。

清单 1. 示例节点定义文件

```
  default:
  ConsoleMethod = mrv
  ConsoleSerialDevice = ttyS0
  ConsoleSerialSpeed = 9600
  InstallAdapterName = eth0
  InstallCSMVersion = 1.4.1
  InstallMethod = kickstart
  InstallOSName = Linux
  InstallPkgArchitecture = x86_64
  ManagementServer = mgmt001.cluster.com
  PowerMethod = bmc
node001.cluster.com:
  ConsolePortNum = 1
  ConsoleServerName = term002
  HWControlNodeId = node001
  HWControlPoint = node001_d.cluster.com
  InstallDistributionName = RedHatEL-WS
  InstallDistributionVersion = 4
  InstallServiceLevel = QU1
node002.cluster.com:
  ConsolePortNum = 2
  ConsoleServerName = term002
  HWControlNodeId = node002
  HWControlPoint = node002_d.cluster.com
  InstallDistributionName = RedHatEL-WS
  InstallDistributionVersion = 4
  InstallServiceLevel = QU1
stor001.cluster.com:
  ConsolePortNum = 2
  ConsoleServerName = term001
  HWControlNodeId = stor001
  HWControlPoint = stor001_d.cluster.com
  InstallDistributionName = RedHatEL-AS
  InstallDistributionVersion = 3
  InstallServiceLevel = QU5
```

对于大型集群来说，节点定义文件的脚本要比本例中的这个脚本长很多。当如果将以下命令传递给 CSM ，它可以非常迅速地创建节点：

```
definenode -f <node-def-filename>
```

火龙果•整理
uml.org.cn

注意 node-def-filename 可被改为您在其中存储节点定义文件的文件名，例如，definenode -f //tmp/my_nodes.def。

CSM 节点数据库现在应该包含所有节点。对于本例中这个小型的集群来说，节点数据库应该包含 16 个计算节点、1 个用户节点、1 个调度器节点和 1 个存储服务器。CSM 管理服务器并不出现在 CSM 数据库内。可以通过 lsnodes 命令查看节点列表。还可以使用 lsnode -F 命令来查看更加详细的列表，该列表可用来备份 CSM 节点定义。如果将该命令的输出重定向到一个文件，您就可以使用 definenode -f 命令来重新定义节点了。

定义节点组

CSM 允许通过一些条件来对节点进行分组，这样，以后就可以针对某一组节点应用 CSM 命令了。对于需要引用具有相同属性的同类节点的情况来说，这一点将非常有用。

CSM 既支持动态节点组又支持静态节点组。静态节点组包含节点名的一个固定列表，该列表可由管理员手动维护。比如，使用静态节点组，您必须手动将新定义的节点添加到相应的节点组中。在大型集群中，动态节点组更为常见，而且如果能被很好地设置还能节省大量时间和减少在命令行内的输入。动态节点组定义了这样的一个节点列表，该列表中的成员由特定的条件定义，而且如果一个节点（包括新定义的节点）满足这个给定的条件，它就会被自动放到相应的节点组。表 3 给出了一些动态节点组的定义。

表 3: 动态节点组

定义命令	注释
Nodegrp -w "Hostname like 'node%'" ComputeNodes	创建一个 ComputeNodes 节点组
Nodegrp -w "Hostname like 'schd%'" SchedulerNodes	创建一个 SchedulerNodes 节点组
Nodegrp -w "Hostname like 'stor%'" StorageNodes	创建一个 StorageNodes 节点组
Nodegrp -w "Hostname like 'user%'" UserNodes	创建一个 UserNodes 节点组
Nodegrp -w "Hostname like 'node%' && ConsoleServerName==='term002'" Rack02 nodegrp -w "Hostname like 'node%' && ConsoleServerName==='term003'" Rack03 nodegrp -w "Hostname like 'node%' && ConsoleServerName==='term...'" Rack...	基于 Hostname 和 ConsoleServerName 为每个 rack 创建一个节点组。假设每个 rack autorpm 有一个控制台服务器。

准备 Linux 发布版

CSM 管理服务器应该包含需要跨集群安装的所有 Linux 发布版的 CD 上的全部内容。它也应该为在客户机上进行 CSM 安装做好准备，这是在任何安装进行之前都必须要做的。CSM 为此提供了两个命令，您必须为每个要安装的 Linux 发布版都要运行这两个命令。

要用所需的 CSM 数据准备 /csminstall/Linux 树，可以运行 copycsmpkgs 命令。例如：

```
copycsmpkgs -p /path/to/csm:/path/to/downloads
InstallDistributionName=RedHatEL-WS
       InstallDistributionVersion=4 InstallServiceLevel=QU1
copycsmpkgs -p /path/to/csm:/path/to/downloads
InstallDistributionName=RedHatEL-AS
       InstallDistributionVersion=3 InstallServiceLevel=QU5
```

要用所需的 Linux 发布版 CD 准备 /csminstall/Linux 树，可以运行 copycds 命令，例如：

```
copycds InstallDistributionName=RedHatEL-WS InstallDistributionVersion=4
       InstallServiceLevel=QU1
copycds InstallDistributionName=RedHatEL-AS InstallDistributionVersion=3
```

火龙果•整理
uml.org.cn

```
InstallServiceLevel=QU5
```

一旦为这些 CD 设置好目录结构，您就能添加任意的定制包来在系统安装过程中进行安装或更新，例如：

- Copy to /csminstall/Linux/.../x86_64/install to ensure they are installed.
- Copy to /csminstall/Linux/.../x86_64/updates to install only if an existing version is present.

可以用节点组名创建子目录以便在特定节点组上只安装或更新 RPMS（如果需要这么做的话）。

设置 CFM

CSM 具有称为 Configuration File Manager (CFM) 的机制，可以用来跨集群分布文件。使用 CFM 可以跨集群发送相同的文件。如果在节点安装之前就做好设置，文件就会在安装过程中被分发。

CFM 可以包含到管理服务器上的其他目录中的文件的链接。这些链接在被发送到节点时是跟随过来的而不是被复制过来的。这一点对于 hosts 之类的文件非常有用，如下所示：

```
mkdir /cfmroot/etc
ln -s /etc/hosts /cfmroot/etc/hosts
```

除了链接文件外，还可以复制文件到 CFM，例如：

- 复制默认的 NTP 配置文件到 /cfmroot/ntp.conf
- 为管理服务器添加服务器行，如下所示：

```
/cfmroot/etc/ntp
echo "management.server.full.name" gt; /cfmroot/etc/ntp/step-tickers
```

文件将会跨集群分发。

在需要将少量文件发送到集群上的特定地点时，可以使用 CFM。但如果是要分发大型集群上的大量文件，最好不要使用 CFM 。例如，不要使用 CFM 来从一个 tar 归档文件安装额外的软件。如果在大型集群硬要使用 CFM，CFM 会花很长的时间运行，使之异常难用。请务必坚持用受支持的安装机制来安装软件。例如，使用 RPM 而非 tar 文件来安装软件并只复制配置文件（即那些很可能会随时间而更改的文件）到 CFM。

定制节点构建

对于您计划安装在每个节点的操作系统来说，CSM 都有到标准网络安装机制的接口，比如 NIM on AIX®、autoYaST on Suse Linux 和 kickstart on Red Hat Linux。这里以 Red Hat 为例说明如何使用 kickstart 和 kickstart 配置文件进行节点安装。

在开始设置 kickstart 之前，先要检查您是否拥有对所有节点的 rpower 控制权。这有助于 CSM 获得计算机的 UUID，尤其是在更新的 CSM 版本中。如果 UUID 不可用，或者 CSM 版本低于 1.4.1.3，CSM 就会尝试从节点的第一个 Ethernet 设备获得 MAC 地址。为了使 CSM MAC 地址收集可用，终端服务器配置必须匹配节点 BIOS 中的设置。使用 rconsole 命令可检查终端服务器连接。当成功建立起 rpower 控制和终端服务器连接（如果合适的话 ）之后，就可以继续 kickstart 配置了。

CSM 在文件 /opt/csm/install/kscfg.tmpl.* 中提供了默认的 kickstart 模板。如果需要的话，可以将这些模板复制到不同的文件并加以定制以更好地满足您自己的环境要求。这些模板是一个很好的起点，一般来说，您都应该采取对这些模板文件进行定制的方式而不是使用任何其他标准 kickstart 文件。原因是这些模板包含各种 CSM 函数（例如运行一个安装后脚本）的宏。CSM 通常在生成最终的 kickstart 文件供节点使用之前会先分析 kickstart 模板文件，这对于 kickstart 过程非常有帮助。最终的文件包含所有解析过的宏并包含针对模板内定义的一切内容的全部脚本。

通常，可以通过如下方式对模板进行更改：

- 更改磁盘分区，可能需要包括 LVM
- 改变默认密码

火龙果•整理
uml.org.cn

- 编辑要被安装的包列表

编辑完 kickstart 模板之后，运行 CSM setup 命令来生成最终的 kickstart 文件，并按如下所示进行最初的 UUID 或 MAC 地址收集：

```
csmsetupks -n node001 -k /opt/csm/install/your.kickstart.file -x
```

注意：需要使用 -x 开关，因为先前运行了 copycds 命令。

更新驱动程序

即使在集群中安装了操作系统不能直接支持的硬件，也仍然可能会有相应的驱动程序可以使用。在必要的时候，这个过程也适用于驱动程序更新。如果最终安装和安装操作系统时所用的 RAM 磁盘需要更新驱动程序，CSM 可以自动将附加或替换的驱动程序包括进来。

在使用 System x 硬件的例子时，您通常会希望获得板载 Broadcom Ethernethard 适配器的 Broadcom 网络驱动程序所提供的性能和稳定性。要实现这一目的，可按如下步骤操作，这些步骤使用的是 Broadcom bcm5700 驱动程序而不是 Red Hat Linux 提供的标准的 tg3 网络驱动程序：

1. 因为处理的是一个内核模块，所以要确保针对目标系统安装的内核源匹配内核等级和类型（UP 或 SMP）。
2. 从 Broadcom （参看 参考资料）下载最新的 bcm57xx 驱动程序并解包驱动程序源代码。
3. 从解包后的 bcm 驱动程序的 src 目录运行 make 来针对当前的运行内核进行构建。
4. 将构建驱动程序 （bcm5700.ko for 2.6 kernel 或 bcm5700.o for 2.4 kernels）复制到管理服务器上的 /csminstall/csm/drivers/lt;kernel versiongt;/x86_64。
5. 如果想针对其他内核版本进行构建，可以运行 make clean 来清除当前的构建，然后再运行 make LINUX=/path/to/your/kernel/source。

CSM 在构建 RAM 磁盘映像时，使用目录结构 /csminstall/csm/drivers/lt;kernel versiongt;/lt;architecturegt 之下的驱动程序；当内核版本匹配 RAM 磁盘内核版本时，这些映像用来在安装过程中引导系统。注意当为安装映像创建驱动程序时一定要小心：安装内核的内核版本号可能会不一样。例如 Red Hat 通常会在版本字符串尾部追加单词 BOOT。如果内核版本与所安装系统的运行内核相匹配，驱动程序对当前的操作系统也可用。如果您不能确认内核的版本，可以在 RAM 磁盘映像内进一步深入研究，如以下一节所述。

修改 RAM 磁盘

通常，这一步骤不推荐使用。但有时候又不得不这么做，比如不能确认内核版本的情况。在由于更新应用程序或其他情况而需要深入研究 RAM 磁盘映像时，以下命令会很有帮助。

当使用主机总线适配器 (HBA) 将存储直接被连到 Red Hat Linux 系统时，存储驱动程序（例如 qlogic qla2300 驱动程序）可能会在 ServeRAID 驱动程序（用于内部系统磁盘即操作系统磁盘）之前加载。如果是这样，就意味着安装在错误的磁盘上了。/dev/sda 代表的是附加存储介质而非本地磁盘上的 LUN。在这种情况下，在安装新的操作系统时，对重写 SAN 上而非本地磁盘上的数据要格外小心。为避免其发生，请从 CSM 用来创建引导映像的默认 Red Hat RAM 磁盘上移除 qlogic 驱动程序。当然，系统运行时也需要驱动程序，所以请使用另外一种机制，例如 post 安装脚本来为运行中的操作系统安装驱动程序。由于默认 Red Hat qlogic 驱动程序通常都不是故障转移驱动程序，所以推荐采用上述做法。

例如，从 Red Hat Enterprise Linux Advanced Server Version 3 的默认 RAM 磁盘映像移除 qla2300 驱动程序。表 4 给出了能实现此目的的命令。

表 4: RAM 磁盘命令

命令	目的

火龙果•整理
uml.org.cn

cd /csminstall/Linux/RedHatEL-AS/3/x86_64/RedHatEL-AS3-QU5/images/pxeboot	转到包含所需更改的 RAM 磁盘映像的那个目录。
cp initrd.img initrd.img.orig	备份原始映像。
mkdir mnt	创建一个挂载点。
gunzip -S .img initrd.img	解包这个映像。
mount -o loop initrd.img /mnt	将这个映像挂载到挂载点。
手动步骤	手动移除所有到 mnt/modules/* 中的 qla[23]00 的引用。
cp mnt/modules/modules.cgz	将模块归档文件从映像复制到当前目录。
gunzip -c modules.cgz | cpio -ivd	解包这个模块归档文件。
rm modules.cgz	删除这个模块归档文件。
rm 2.4.21-32.EL/ia32e/qla2*	从解包后的模块归档文件中删除 qlogic 模块。
find 2.4.21-32.EL -type f | cpio -–o -H crc | gzip -c -9 > modules.cgz	删除 qlogic 模块后，打包剩余模块。
rm -rf 2.4.21-32.EL	删除解包后的模块归档文件。
mv -f modules.cgz mnt/modules	用新的模块归档文件代替旧的模块归档文件。
umount mnt	卸载这个 RAM 磁盘映像。
rmdir mnt	移除挂载点。
gzip -S .img initrd	再次打包这个 RAM 磁盘映像。

注意：要修改 Suse 或 SLES 的 RAM 磁盘，必须确保 ips（ServeRAID 驱动程序）出现在 INITRD_MODULES stanza 之下的 /etc/sysconfig/kernel 文件内的所有 HBA 驱动程序之前。用于创建 RAM 磁盘映像的 Suse 或 SLES 机制可确保驱动程序被顺序加载。

安装重新引导前和重新引导后脚本

由于每个环境和集群都是不相同的，您可能需要应用一些安装后脚本来根据您自己的特定需求定制操作系统安装。可以在重新引导到新安装的系统之前或之后进行这些操作。对于配置备用网络适配器来说，这十分有用，CSM 针对此目的提供了一个示例脚本。由于双网络设置（到每个节点都有一个计算网络和一个存储网络）的原因，这个示例集群将需要使用备用适配器配置。具体的步骤如下所示：

火龙果•整理
uml.org.cn

1. 将 CSM 提供的默认适配器配置脚本复制到 installprereboot 脚本执行目录以使脚本可以在安装时运行，确保它能运行，如下所示：

```
cp /csminstall/csm/scripts/adaptor_config_Linux /csminstall/csm/scripts/
       installprereboot/100_adapter_config._LinuxNodes
chmod 755
/csminstall/csm/scripts/installprereboot/100_adaptor_config._LinuxNodes
```

2. 生成适配器 stanza 文件 /csminstall/csm/scripts/data/Linux_adapter_stanza_file，可如下所示编写头部：

```
default:
machine_type=secondary
network_type=eth
interface_name=eth1
DEVICE=eth1
STARTMODE=onboot
ONBOOT=yes
BROADCAST=192.168.1.255
NETMASK=255.255.255.0
MTU=9000
```

3. 这会配置所有备用 (eth1) 适配器在引导时启动，并采用针对广播地址、网络掩码和 MTU 大小的默认设置。在附加的 stanza 行内还可以以一种与节点定义文件相同的方式配置特定于计算机的网络细节，如下所示：

```
for node in $(lsnodes)
do
  ip=$(grep $node /etc/hosts | head -n 1 | awk '{print $1}')
  echo -e "$node:\n  IPADDR=$ip" gt;gt; Linux_adaptor_stanza_file
done
```

4. 这会把输出追加到适配器 stanza 文件以为每个计算机配置不同的 IP 地址，如下所示：

```
node001.cluster.com:
  IPADDR: 192.168.1.1
node002.cluster.com:
  IPADDR: 192.168.1.2
node003.cluster.com:
  IPADDR: 192.168.1.3
```

安装

在节点安装过程中有两个主要的 shell 环境变量：CSM_FANOUT 和 CSM_FANOUT_DELAY。前一个变量控制向多少节点同时发送 CSM 指令，比如有多少节点从管理服务器被重新引导。后一个变量控制在重新引导下一组要安装的节点时 CSM 要等待多长时间（以秒为单位）。这两个变量的设置分别为 16 个节点用来扇出和在

火龙果•整理
uml.org.cn

重新引导下一组节点之前要等待 20 分钟。这些默认值对于大多数安装来说是可以接受的，但也可以针对大型集群有所增加。

要以传统方式安装集群，请完成以下步骤：

1. 配置此安装并按如下所示安装计算节点：

```
csmsetupks -N ComputeNodes -k
/opt/csm/install/your.kickstart.file.nodes -x
installnode -N ComputeNodes
```

2. 配置此安装并按如下所示安装用户节点：

```
csmsetupks -N UserNodes -k /opt/csm/install/your.kickstart.file.user -x
installnode -N UserNodes
```

3. 配置此安装并按如下所示安装调度器节点：

```
csmsetupks -N SchedulerNodes -k
/opt/csm/install/your.kickstart.file.schd -x
installnode -N SchedulerNodes
```

4. 配置此安装并按如下所示安装存储节点：

```
csmsetupks -N StorageNodes -k
/opt/csm/install/your.kickstart.file.stor -x
installnode -N StorageNodes
```

对于大型集群的安装，需要使用安装服务器来进行分阶段安装，并按如下所示使安装过程并行进行：

1. 在 CSM 内设置 InstallServer 属性。对于每一个想从安装服务器安装的节点，将 InstallServer 属性设置为该节点所使用的安装服务器的主机名。如果没有设置这一属性，就会使用其默认值，即从中心管理服务器安装。在大型集群环境，比方说，每个 rack 有 32 个节点，可以选择每个 rack 底部的节点作为此集群的安装服务器。在本例中，要将 rack 1 中的 node002 到 node032 配置成从 node001 安装并让 node001 从管理服务器安装，可以使用如下命令：

```
chnode -n node002-node032 InstallServer=node001
```

2. 按如下所示创建一个包含所有安装服务器的动态节点组和一个包含客户机的动态节点组：

```
nodegrp -w "InstallServer like '_%'" InstallServers
nodegrp -w "InstallServer not like '_%'" InstallClients
```

3. 配置此安装并按如下所示安装安装服务器：

火龙果•整理
uml.org.cn

```
csmsetupks -N InstallServers -x
installnode -N InstallServers
```

4. 要充分利用安装服务器提供的带宽，可增加 CSM 扇出值来同时重新引导更多节点。在这个每个 rack 32 节点的例子中，CSM 的最佳扇出值应该是 32 乘以安装服务器数量（或者在每个 rack 一个节点的情况下，就是 rack 的数量）。在本例中，您也可以将每个安装服务器上的 NFS 线程数增加到 32 来更好地扩展 NFS。使用这种方法，您可以同时安装成百上千台机器。

5. 配置此安装并按如下所示安装安装客户机：

```
csmsetupks -N InstallClients -x
installnode -N InstallClients
```

第 3 部分：存储和共享文件系统

存储架构

在继续之前，先阅读一下本系列 第 1 部分 中有关总体架构的章节将会很有益处。

图 1 给出了本系列中描述的示例集群的存储配置概览。此配置在本文中将进一步加以解释。该设置基于 GPFS 版本 2.3。包括一个大型的 GPFS 集群，该集群分成两个逻辑部分，共用一个大型文件系统。示例设计还提供了应付灾难的恢复能力，如果存储后端的一半丢失，另一半仍可继续操作。

图 1. 存储架构概览

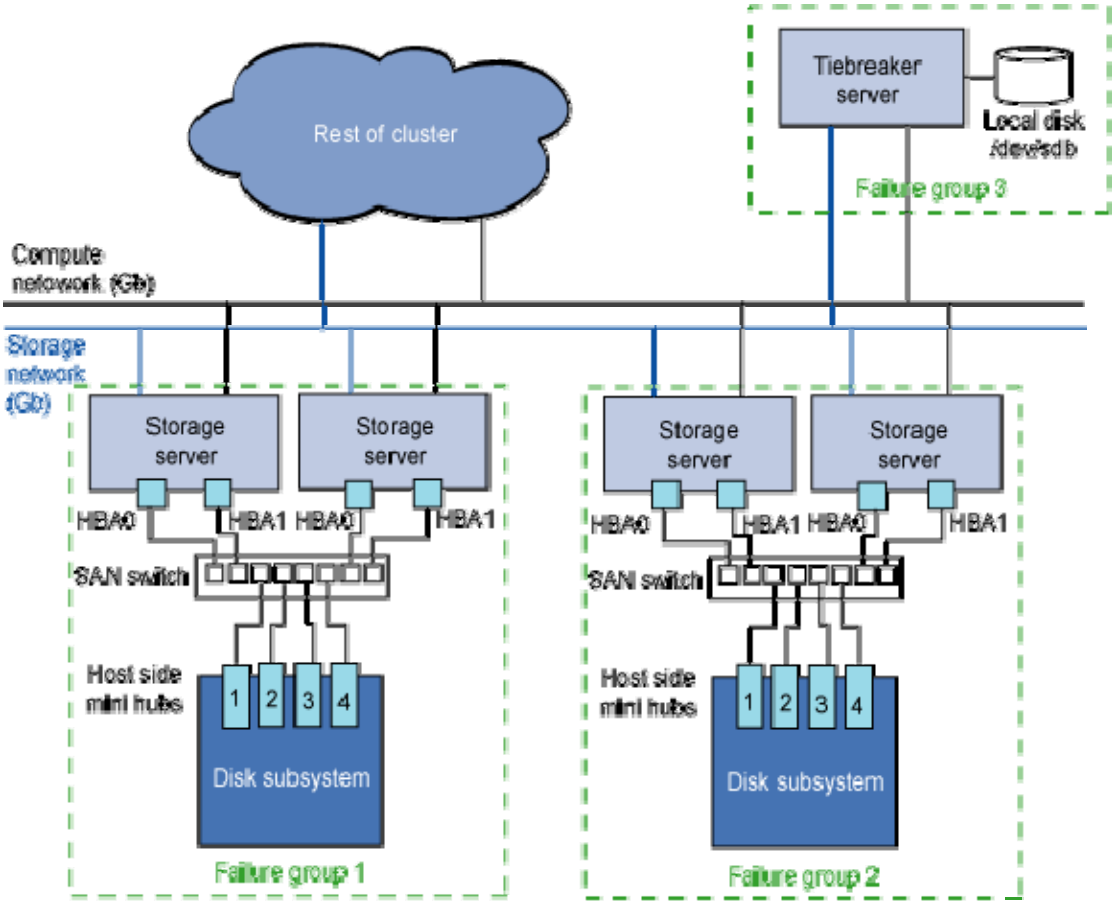


图 1 显示了四个存储服务器，可管理由两个磁盘子系统提供的存储。在右上角，可以看到一个 tie-breaker 服务器。网络连接和光纤连接也显示出来以供参考。所有这些内容会在本文的后续章节详细加以介绍。集群的其他部分显示为云状，本文不会涉及。更多关于该集群其他部分的信息，请参看本系列的 第 1 部分 和 第 2 部分 。

节点

火龙果•整理
uml.org.cn

GPFS 集群的大部分节点都运行 Red Hat Enterprise Linux 3。本例使用服务器/客户机架构，其中，少数几个服务器通过光纤通道拥有对存储的可见性。它们充当集群其他部分的网络共享磁盘 (NSD) 服务器。这意味着大多数 GPFS 集群的成员都将使用 NSD 服务器经由 IP 访问存储。NSD 节点（即存储节点）一共有四个：GPFS 集群的逻辑分区各两个。它们均按对分组，每一对管理其中的一个存储子系统。

Tiebreaker

集群的每一半都包含完全相同数目的节点，若其中的一半丢失，就会带来仲裁问题。通过 GPFS，若要使文件系统保持可用，就需要有一个有效的节点仲裁。Quorum 由 Quorum = (number of quorum nodes / 2) + 1 定义。

在本例所示的这种配置中，集群由完全相同的两个部分组成，若其中任何一个丢失，GPFS 文件系统都会不可用。为了避免这种情况，系统采用了一个 tie-breaker 节点。此节点的物理位置与主集群不在一起。这意味着若任何一部分不可用，另一部分都能继续访问 GPFS 文件系统。这是通过使用三个故障组得以实现的，我们将在 数据复制 一节详细对之进行介绍。这意味着数据的两个副本都可用：集群的每一半各一个。

架构

如图 1 所示，每个节点均连接至两个网络。其中的第一个网络用于计算流量和总的集群通信。第二个网络则专门针对 GPFS，为那些不具有 Storage Area Network (SAN) 存储系统直观视图的节点提供经由 IP 的存储访问。这个网络使用巨大帧来获得好的性能。有关存储网络的更详细的信息，请参阅本系列第 4 部分的 GPFS 网络调优一节。

存储区域网络

本解决方案的存储后端包括两个磁盘子系统，均为 IBM TotalStorage DS4500（之前为 FAStT 900）磁盘系统，每一个都附带完全填充过的 EXP710 扩展磁盘柜。每个 DS4500 均被配置成 RAID 5 4+P 阵列外加热后备磁盘。

每个 DS4500 均由一对存储服务器*所有*。在这两个服务器间拆分 4+P 阵列，以便每个服务器都可充当阵列其中一半的主服务器、阵列另一半的备用服务器。这样一来，若任何一个服务器故障，另一个服务器都能立即作为主服务器从故障的服务器接管磁盘。

数据复制

本例中，GPFS 会复制 GPFS 文件系统中的数据或元数据。存储被拆分成三个故障组。每一个故障组都包括一组共享共同故障点的逻辑磁盘（从操作系统不难发现，这里，一个磁盘对应于一个 LUN，这个 LUN 是 DS4500 上的一个磁盘阵列）。此系统中的故障组由如下部分组成：

- 故障组 1 中的一个 DS4500 系统
- 故障组 2 中一个 DS4500 系统
- 属于 tie-breaker 节点的本地磁盘

在创建 GPFS 文件系统时，应该先将数据和元数据的副本数量指定为 2。所以，对于如上定义的故障组，每半个部分都会包含文件系统的一个副本。第 3 个故障组是解决磁盘仲裁问题所必需的，以便如果存储的其中一半离线之后，磁盘仲裁能立即发挥作用且文件系统也能保持总可以被访问。

硬件准备

如所提到的，此集群包含两个 IBM TotalStorage DS4500 设备，二者组成了此解决方案的存储后端。有关此硬件的更多信息，可以在 参考资料 一节找到。

IBM 使用 IBM TotalStorage DS4000 EXP710 光纤通道（FC）存储扩展单元将两个 DS4500 系统连接起来。其中的每个单元都是一个 14-bay、2 GBps、可机架安装的 FC 封装。 有关此硬件的更多信息，可以在 参考资料 一节找到。

如下的章节涵盖了这个示例解决方案中用到的 DS4500 和 EXP710 单元的配置细节。

开关的顺序

注意，您需要以一种特定的顺序来开关 SAN 系统，以便所有存储都可被正确发现。请以如下顺序执行系统开启：

1. SAN 交换机（允许其完全初始化）
2. EXP 710 柜

火龙果•整理
uml.org.cn

3. DS4500 （允许其完全初始化）
4. 存储服务器

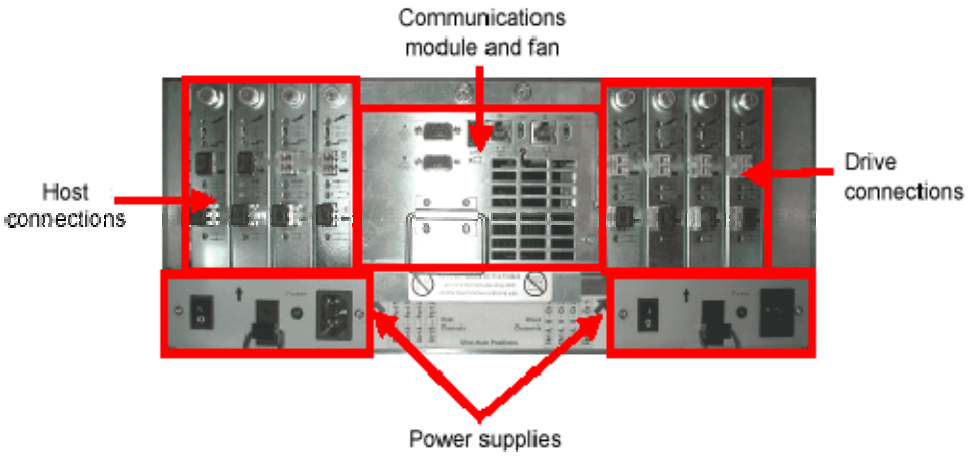
关闭的顺序则与之相反，如下所示：

1. 存储服务器
2. DS4500
3. EXP 710
4. SAN 交换机

连接

图 2 显示了 DS4500 单元的背面。在左侧有四个 mini-hub 端口，可用来进行主机连接。在本文，将这四个端口由左向右依次表示为 slot 1 至 slot 4，如图 1 所示。Slot 1 和 3 对应于顶部控制器，即控制器 A。Slot 2 和 4 分别对应于底部的控制器，即控制器 B。在右侧，也有四个 mini-hub 端口，可用于进行扩展柜 (EXP710) 的连接。

图 2：DS4500 的背板图

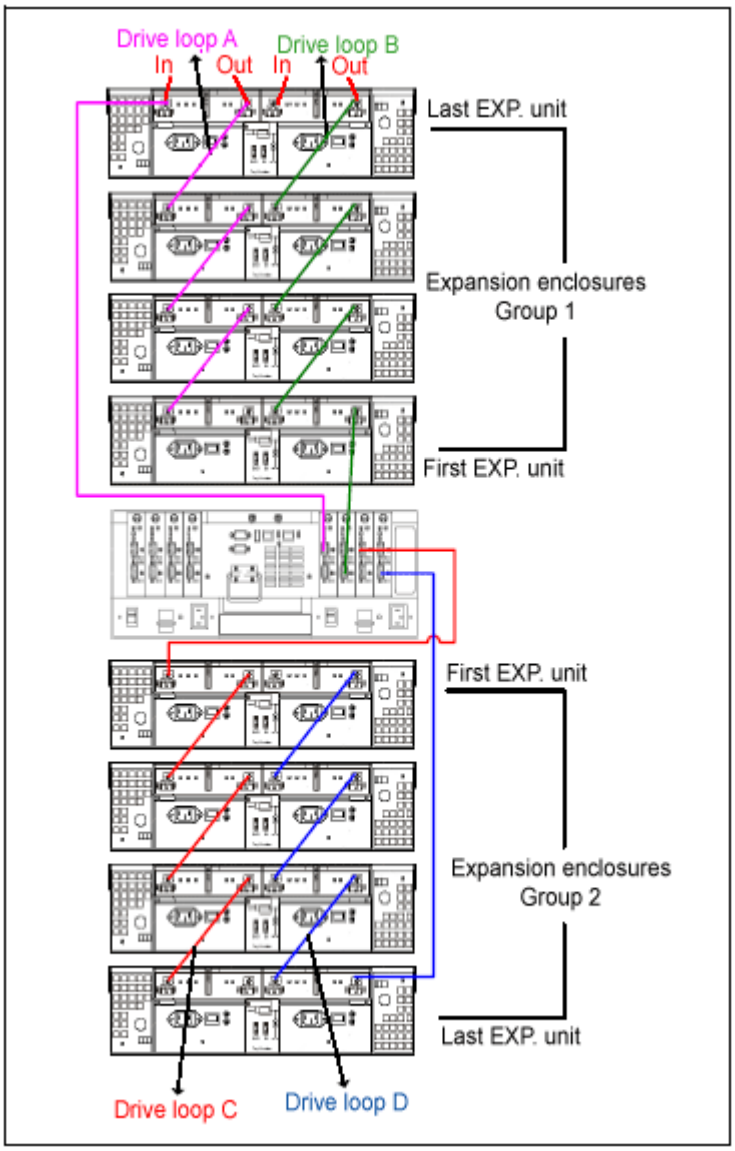


Cabling

每个 DS4500 都会连至两个环，如图 3 所示。

火龙果•整理
uml.org.cn

图 3：DS4500 和 EXP 柜间的示例布线



设置 EXP 封装 ID

每个 EXP 710 柜都必须有惟一的 ID。这可通过每个封装背部的面板进行设置。

为 DS4500 控制器配置 IP 地址

使用每个封装背面的串行口设置每个控制器的 IP 地址。可以使用 Windows® 中的应用程序超级终端或 Linux 上的小型计算机进行这里设置。本例使用了如下设置：

- 波特 38400
- 比特 8
- 奇偶校验 无
- 结束比特 1
- 流量 xon/xoff

通过发送 Break（如果使用超级终端，则为 Ctrl-Break）来建立连接，然后再按下空格键来设置速度。之后，发送另一个 break 并使用退出键进入内核。默认的密码是 infiniti。

使用命令 netCfgShow 显示控制器的当前 IP 地址设置，使用命令 netCfgSet 来设置所需连接的 IP 地址、子网掩码和网关。

从 Storage Manager 发现 DS4500

这之后，可通过 Storage Manager (SM)软件对 DS4500 进行管理。对新的硬件使用最新的版本（9.1 或更高）。

火龙果•整理
uml.org.cn

您可以使用 Storage Manager 来：

- 配置阵列和逻辑驱动器
- 将逻辑驱动器分配给存储分区
- 替换和重构失效的磁盘驱动器
- 扩展阵列的规模
- 从一个 RAID 级别转换到另一个级别

您也可以进行故障排除或执行管理任务，例如检查 TotalStorage 子系统的状态及更新 RAID 控制器的固件。要获得适合您的具体硬件的最新版 Storage Manager，请参看 参考资料。
SM 客户机可以安装在许多操作系统上。在本文所描述的示例中，SM 客户机安装于管理服务器。使用左侧的第一个按钮从 SM 客户机发现新配置好的 DS4500。要在可通过接口看到的 DS4500 上执行操作，可双击计算机名来打开新窗口。

DS4500 控制器总的配置步骤

首选，重命名 DS4500，方法是转到 Storage Subsystem > Rename...，并输入新的名称。接下来，检查时钟是否同步，方法是转到 Storage Subsystem > Set Controller Clock。在这里，确保所有时钟均已同步。现在，转到 Storage Subsystem > Change > Password，设置系统密码。

更新 DS4500 和 EXP 710 柜的固件

要从 Storage Manager 检查系统固件的等级，可转到 Advanced > Maintenance > Download > Firmware。当前的等级列于窗口的最顶部。您可以从这里向计算机下载更新的版本，但务必对此模型使用正确的固件且以特定的顺序对级别进行升级，此顺序应该是在固件代码中的注释中指定的。磁盘和 ESM 的固件也可以从 Download 菜单进行检查。

手动配置还是脚本配置

下面的章节详细介绍了 DS4500 的手动配置。请遵循本方案中的 DS4500 初始配置的操作步骤，并保存第一个 DS4500 的配置。这个操作会生成脚本，一旦 DS4500 被重置或被替换了硬件，您就可以使用此脚本来重新配置相同的 DS4500。
为了简单和准确地重新生成同样的设置，您也可以复制此脚本并加以编辑以便在其他 DS4500 上也可以使用。在此过程中，需要更改包含 DS4500 名称、磁盘位置、阵列名称和主机映射细节的域。注意这些脚本会将 Access LUN 留在主机组定义。这可手动从每个 DS4500 删除。

创建热后备磁盘

本例在每个 DS4500 上都使用了大量磁盘作为热后备。添加的方式是右键单击要分配用作热后备的磁盘，选择手动选项并输入 DS4500 的密码（在 DS4500 控制器总的配置步骤 一节设置）。

创建磁盘阵列

1. 右键单击一个要添加到阵列的尚未分配的磁盘，选择 Create Logical Drive。
2. 在所出现的向导中单击 Next。
3. 选择 RAID level 5。原始驱动器已经被选中。
4. 向阵列再添加其他四个驱动器，使其总数为 5。
5. 在 Array Success 窗口，单击 OK，在此阵列上创建逻辑驱动器。
6. 选择默认选项，其中整个 LUN 都被用于此逻辑驱动器。而这个逻辑驱动器的命名约定是 <ds4500 name>_array<number>。在 Advanced Parameters 之下，选择 Customize Settings。
7. 在 I/O Characteristics 类型，选择默认，即 File System，并选择首选槽以便阵列可以在 A 和 B 间交替。在本例中，slot A 中有奇数个阵列，slot B 中有偶数个阵列。
8. 选择 Map Later 以稍候再返回到映射。

在创建此阵列时，您会看到一个绿色的圆筒，旁边还有一个时钟。可以通过在这个逻辑磁盘名上右键单击并选择 Properties 来查看进展。

火龙果•整理
uml.org.cn

注意，此后的步骤需要您必须已经配置好 SAN 交换机并已安装和运行存储服务器，且主机总线适配器 (HBA) 也已配置，以便 HBA 的 WWPN 均可对 SAN 交换机可见，并进而对 DS4500 可见。有关这些步骤的详细信息，请参看第 4 部分的 SAN 架构 一节和 HBA 配置的相关章节。

存储分区和磁盘映射

LUN 一旦创建，就必须被分配到主机。在本例中，使用存储分区。通过创建逻辑驱动器到 LUN 的映射可以定义存储分区。这就将对特定逻辑驱动器的访问权授予了主机或主机组。在定义存储分区时，请按顺序执行这些步骤。需要首先定义拓扑结构然后再定义存储分区：

1. 定义主机组。
2. 定义组内的主机。
3. 定义主机端口。
4. 定义存储分区。

如前所述，在本例的设置中，每个 DS4500 只有一个组，包含两个存储节点，两节点间的所有在 DS4500 上的磁盘都将被 twin tailed。除了 Access LUN 之外，所有其他的 LUN 均被分配到这个组。Access LUN 用于 DS4500 的带内管理。然而，Linux 并不提供对它的支持，必须从所创建的所有节点组中删除。

创建一个新的主机组，方法是右键单击 Default Group 段并选择 Define New Host Group。输入主机组的名称。之后，创建一个新的主机，方法是右键单击所选的主机组并选择 Define Host Port。在下拉菜单，选择对应于所添加 HBA 的 WWPN。注意，若要使 WWPN 出现在此菜单，您必须在此 SAN 中已经正确对该主机进行了配置和专区划分。随后，Storage Manager 将会在 Show All Host Port Information 下看到此主机。Linux Host Type 已被选中，随后，主机端口名称也会被输入到最后的那个框内。

重复此步骤，以便每个主机都会有两个已经定义好的端口，接下来，创建存储分区，方法右键单击新创建主机组并选择 Define Storage Partition，以打开存储分区向导。单击 Next 启动该向导。选择刚刚创建的主机组，单击 Next。选择之前定义的那些 LUN 以将其包括到这里。注意 Access LUN 也必须包括进来。单击 Finish 来完成此集合。

SAN 架构

本节将介绍设置集群中的 SAN 架构所需的步骤。本例的配置中使用的 SAN 交换机是 IBM TotalStorage SAN Switch H16 交换机 (2005-H16)。有关此硬件的更多信息，请参考 参考资料。

在本节中，详细介绍了配置 SAN 交换机的步骤，参考了 H16 交换机的某些命令和接口。

为 H16 SAN 交换机配置 IP 地址和主机名

要在 H16 SAN 交换机上进行 IP 地址的初始配置，可以将此交换机随带的串行电缆（黑头、非 null modem）连接至计算机后端的端口，并使用如下的连接设置：

- 9600 波特
- 8 数据位
- 无奇偶校验
- 1 终止位
- 无流量控制

使用默认的登录信息：用户名为 admin ，密码为 password。可使用命令 ipAddrSet 更改主机名和 IP 地址。也可使用命令 ipAddrShow 验证设置。

IP 地址一旦配置，就可以用 Web 界面来管理 SAN 交换机了。用 IP 地址连接至 SAN 交换机，浏览器必须带 Java™ 插件。要访问 Admin 界面，单击 Admin 按钮并输入用户名和密码。这里，您可以将交换机的新名字输入到相应的框中并应用这些更改。

网络中每个域的域 ID 必须惟一。本例中的所有交换机均包含在其自身的网络中，但考虑到未来合并的可能，可以改变它们的 ID。注意，在更改域 ID 之前，交换机需要被禁用。

还有一点可供参考，网络一旦可以访问此交换机，您就可以使用 Admin 界面的 Network Config 选项卡更改此 SAN 交换机的 IP 地址。这种方法是串行连接的一种替代方法。

火龙果•整理
uml.org.cn

SAN 交换机分区（zoning）

示例集群使用如下的分区原则：

- 所有主机上的 HBA0（PCI slot 3 中的 Qlogic 光卡）划分为一区，对此 DS4500 的控制器 A（slot 1 和 3）公开
- 所有主机上的 HBA1（PCI slot 4 中的 Qlogic 光卡) 划分为一区，对此 DS4500 的控制器 B（slot 2 和 4）公开

设置 SAN 交换机的分区可以通过前一节所介绍的交换机上的 Web 界面完成。使用窗口左下角最右侧的按钮可以访问到分区页面。欲简化分区的管理，可以为每个 WWPN 分配别名来识别附在此端口的设备。

以下介绍了如何创建别名并将其分配给主机。首先，通过单击 Create 并输入别名的名称来添加别名，然后，选择一个 WWPN 以将它分配给新建的别名。在每个端口都会有三个级别，即：

1. 主机 WWN
2. WWPN
3. 注释

选择第二个级别以及 Add member，可以将第二个级别添加到别名。

创建了别名之后，下一步是通过对别名进行分组来创建各个专区（zone）。在本例的配置中，曾用到过专区，在这些专区中，每个主机上的每个 HBA 只能看到相关 DS4500 上的一个控制器。如上一节所介绍的，在本例的设置中，每个 DS4500 只将它的磁盘展示给两个主机。每个主机使用不同的连接连至控制器以分散负荷及最大化吞吐量。这种分区称为*单 HBA 分区*。在 SAN 级，所有主机之间相互隔离。这种分区消除了主机间不必要的 PLOGI 动作，也消除了故障的 HBA 对其他 HBA 产生影响的风险。其结果是，由于对单个专区的修改并不会影响到其他主机，所以交换机的管理更加安全。当需要添加新的主机时，请尽量创建新的专区，而不是将该主机添加到现有的一个专区内。

最后一步是将定义添加到配置中的专区，这些专区可被保存，然后激活。此外，很有必要生成一个交换机报告，方法是单击 Admin 按钮并选择 Switch Report。此报告是 html 格式的，包含手动重建交换机配置所需的全部信息。

将配置保存到另一个服务器

SAN 交换机一旦配置完毕，就可以通过 ftp 将配置上载到另一个服务器。如果需要自动重置交换机，也可以再次这么做。将配置文件保存到另一个服务器的步骤如下：

1. 在要接收此文件的服务器上设置并启动 ftp。
2. 以 admin（默认密码为 password）通过 telnet 登录到 SAN 交换机。
3. 输入 configupload 命令。
4. 输入所需信息：IP 地址、帐号和密码、将被创建的文件的名称和位置。

更新固件

可使用某个 FTP 服务器上的下载更新固件。具体步骤如下：

1. 设置 FTP 服务器并将固件 tar 压缩包解压缩到 ftp 目录。
2. 使用 firmwareshow 命令检查当前的固件级别。
3. 使用 firmwaredownload 命令启动下载进程。
4. 输入所需信息：IP 地址、帐号和密码、存有固件的当前目录（后跟 release.plist，例如 /pub/v4.4.0b/release.plist）。若 release.plist 文件没有出现，也不要惊慌，此交换机会下载和安装软件，然后重新引导。
5. 以 admin 登录并通过命令 firmwaredownloadstatus 检查更新的状态。

第 4 部分：节点安装和 GPFS 集群配置

火龙果•整理
uml.org.cn

详细介绍节点安装细节

本节详细介绍了与示例集群的存储后端相关的集群服务器管理（CSM）细节。内容包括在每个节点上安装 GPFS 代码，以及为存储节点配置 Qlogic 适配器。请注意这种配置并不一定需要使用 CSM 执行；也可以手工完成。本文中的例子使用 CSM 几乎实现了新服务器安装的自动化，包括存储服务器。

回顾架构问题

在阅读本节之前，先阅读一下本系列 第 1 部分 中有关总体集群架构的章节将会很有益处。阅读 第 3 部分 中有关存储架构的章节将会获益良多。理解整个架构可以使您更好的利用本文中的信息。

按照正确的顺序安装存储节点

按照正确的顺序进行安装是解决后面介绍的 ROM 溢出问题所必须的，因为该配置中使用的 xSeries™ 346 系统没有使用 RAID 7K 卡。请按照以下顺序完成这些步骤：

1. 在管理服务器上运行 csmsetupks -vxn <nodename>。
2. 断开存储服务器与 SAN 上的连接，从而避免在 SAN 磁盘上安装操作系统，这些磁盘会被首先找到。
3. 在管理服务器上运行 installnode -vn <nodename>。
4. 当存储节点重新启动时在控制台上按 F1 键进入 BIOS。
5. 进入 Start Options，并将 PXEboot 从 disabled 修改为 enabled for planar ethernet 1。
6. 让节点重新启动，开始安装。
7. 从终端服务器上监视安装过程，让节点全部引导。
8. 登录到节点中，监视 csm 安装 log /var/log/csm/install.log。
9. 在 post－reboot 任务完成时重新启动节点。
10. 当节点重新启动时，按 F1 键进入 BIOS。
11. 进入 Start Options，将 PXEboot 修改为 disabled。
12. 插上 SAN 线，让节点完全引导。
13. 按照 配置磁盘路径和负载均衡 中介绍的方法使用 MSJ 配置磁盘路径。

提供节点间无密码的根访问权限

GPFS 需要 GPFS 集群中的所有节点都能够使用根用户 ID 访问其他节点，而不需要提供密码。GPFS 使用这个中间节点的访问允许 GPFS 集群中的任何节点在其他节点上运行相关命令。例如，此处就使用了安全 shell（ssh）来提供这种访问能力，不过您也可以使用远程 shell（rsh）。为此，请创建一个集群范围内的密钥和相关配置文件，然后按照下面的步骤使用 CSM 分发这些文件：

1. 创建两个新目录 /cfmroot/root/.ssh 和 /cfmroot/etc/ssh。
2. 输入

```
ssh-keygen -b 1024 -t rsa -f
/cfmroot/etc/ssh/ssh_host_rsa_key -N "" -C "RSA_Key"
```

，创建一个 RSA 密钥对，即验证使用的公钥和私钥

火龙果•整理
uml.org.cn

3.

```
ssh-keygen -b 1024 -t dsa -f
/cfmroot/etc/ssh/ssh_host_dsa_key -N "" -C "DSA_Key"
```

，创建一个 DSA 密钥对，即验证使用的公钥和私钥

4. 创建一个包含公钥的授权文件，如下所示。SSH 使用这个文件来确定是否提示输入密码。

```
cat /root/.ssh/id_rsa.pub >
/cfmroot/root/.ssh/authorized_keys2
cat /root/.ssh/id_dsa.pub >>
/cfmroot/root/.ssh/authorized_keys2
cat /cfmroot/etc/ssh/ssh_host_rsa_key.pub >>
/cfmroot/root/.ssh/authorized_keys2
cat /cfmroot/etc/ssh/ssh_host_dsa_key.pub >>
/cfmroot/root/.ssh/authorized_keys2
```

5. 停止 CSM 维护 known_hosts 文件，如下所示。这个文件中包含了很多主机名。如果一个主机已经出现在这个文件中，那么 SSH 就不会提示用户确认连接。CSM 会试图维护这个文件，但是在一个不需要根密码访问的混合集群环境中，这可能会成为一种障碍。

```
stopcondresp NodeFullInstallComplete SetupSSHAndRunCFM
startcondresp NodeFullInstallComplete RunCFMToNode
perl -pe 's!(.*update_known_hosts.*)!#$1!' -i
/opt/csm/csmbin/RunCFMToNode
```

6. 生成一个系统范围的 known_hosts 文件。这最好通过创建一个脚本来实现，如下所示。运行这个脚本并将输出重定向到 /cfmroot/root/.ssh/known_hosts 中。

```
#!/bin/bash
RSA_PUB=$(cat "/cfmroot/etc/ssh/ssh_host_rsa_key.pub")
DSA_PUB=$(cat "/cfmroot/etc/ssh/ssh_host_dsa_key.pub")
for node in $(lsnodes); do
  ip=$(grep $node /etc/hosts | head -n 1 | awk '{print $1}')
  short=$(grep $node /etc/hosts | head -n 1 | awk '{print
$3}')
  echo $ip,$node,$short $RSA_PUB
  echo $ip,$node,$short $DSA_PUB
done
```

7. 该示例脚本只能用于单个接口。您可以对其稍加修改，从而允许跨多个接口的无密码连接。known_hosts 文件的格式已经超出了本文的范围，但是对每一行都使用一个以逗号分隔开的主机名会非常有用。

8. 通过链接所生成的密钥可以允许无密码的根用户访问，如下所示。

火龙果•整理
uml.org.cn

```
cd /cfmroot/root/.ssh
ln -s ../../etc/ssh/ssh_host_dsa_key id_dsa
ln -s ../../etc/ssh/ssh_host_dsa_key.pub id_dsa.pub
ln -s ../../etc/ssh/ssh_host_rsa_key id_rsa
ln -s ../../etc/ssh/ssh_host_rsa_key.pub id_rsa.pub
```

9. 您可能希望确保这种配置在安装时（操作系统重新启动之前）就安装到了每个系统上。CSM 并不能确保后安装过程的执行顺序，因此如果有任何后安装任务依赖于这种配置，就可能会失败。但也可能会成功，因此安装成功与否无法预测。例如，您可能有一个 GPFS 的后安装脚本，并且需要将一个节点添加到 GPFS 集群中，然后挂载 GPFS 文件系统。实现这种功能的一种方法是为此处创建的所有文件创建一个 tar 包，并使用一个 CSM 的 post-installation pre-reboot 脚本解包。

定义与 GPFS 有关的 CSM 组

对于本例来说，我们定义了两个主要的 CSM 组在配置 GPFS 过程中使用，如下所示。

- StorageNodes，其中只包括那些直接连接到 SAN 上的节点，例如 nodegrp -w "Hostname like 'stor%'" StorageNodes。
- NonStorageNodes，其中包括了构成 GPFS 集群的所有其他节点，例如 nodegrp -w "Hostname not like 'stor%'" NonStorageNodes。

这两个组在安装过程中用来确保执行存储节点角色的服务器可以接收特定的二进制文件和配置文件，这在下面会详细进行介绍。注意本节没有介绍 CSM 所执行的详细安装步骤。有关对这个过程的介绍，请参看本系列文章的 第 1 部分 和 第 2 部分。

安装过程被总结为以下阶段：

1. 从安装服务器上进行 PXE/DHCP 引导
2. 从安装服务器上开始 NFS 安装
3. 执行 Pre-reboot 脚本
4. 重新启动
5. 执行 Post-reboot 脚本
6. CFM 文件传输
7. CSM 后安装配置

本文中的配置修改都是在 pre-reboot 和 CFM 文件传输阶段进行的。

安装 GPFS RPM

GPFS 需要每个集群成员都安装一个基本的 GPFS RPM 集。样例安装所使用的 GPFS 版本是 2.3.0-3。这些 RPM 的安装是一个两阶段的过程：安装 2.3.0-1 基本版本，然后更新到 2.3.0-3。这个安装使用的 RPM 有：

- gpfs.base
- gpfs.docs
- gpfs.gpl
- gpfs.msg.en_US

注意：由于本文中的例子使用的是 GPFS Version 2.3，因此并不需要安装 Reliable Scalable Cluster Technology（RSCT）并创建一个对应的域。Version 2.3 之前版本的 GPFS 需要这些手工设置这些步骤。CSM 可以采用多种方法来安装 GPFS RPM。本文推荐在安装基本操作系统时安装 RPM。CSM 提供了一个包含定制的 RPM 的安装和更新目录结构，然而，对于最初的 RPM 安装，以及之后升级到 GPFS 2.3.0-3 所需的 RPM 安装，CSM 提供的结构不一定能够工作。

火龙果•整理
uml.org.cn

另一种方法是为 CSM 编写一些 pre-reboot post-installation 脚本来根据需要安装 RPM。在这种情况中，要将所有的 GPFS RPM（包括更新 RPM）全部拷贝到管理服务器上的 /csminstall/Linux 目录中。CSM 通常为脚本数据预留的目录是 /csminstall/csm/scripts/data，在安装过程中它会被挂载到节点上，这样就可以通过 NFS 来使用所需要的 RPM。

编写安装脚本 /csminstall/csm/scripts/installprereboot/install-gpfs.sh 来安装 GPFS。下面是一个安装脚本的示例：

```
#! /bin/bash
# Installs GPFS filesets and updates to latest levels
# CSMMOUNTPOINT environment variable is set by CSM
DATA_DIR=$CSMMOUNTPOINT/csm/scripts/data
cd $DATA_DIR
rpm -ivh gpfs.*2.3.0-1*.rpm
rpm -Uvh gpfs.*2.3.0-3*.rpm
echo 'export PATH=$PATH:/usr/lpp/mmfs/bin' >
/etc/profile.d/gpfs.sh
```

一旦在存储服务器上安装 GPFS 之后，您可能还会希望自动安装 FAStT MSJ 工具，这可以以静寂（非交互）模式来完成。MSJ 用于 Qlogic 适配器、故障恢复和多路径等配置，这将在 HBA 配置 一节中详细进行介绍。安装并不是基于 RPM 的，因此默认情况下无法简单地集成到 CSM 中。要安装它，需要在 GPFS 安装程序的末尾添加一个脚本来检查这个节点是否是一个存储服务器，然后再安装 MSJ。要以静寂模式完成安装，请使用 FAStTMSJ*_install.bin -i silent 命令。

配置 Qlogic 故障恢复

示例集群使用的是 Qlogic qla2300 驱动程序，版本为 7.01.01，用于 Qlogic QLA 2342 适配器。存储节点组中的每个节点都有两个这种 PCI 适配器。qla2300 驱动程序已经成为 Red Hat Enterprise Linux 3 update 4 发行版中的标准组件了。然而，为了满足这个示例集群的需要，您需要进行以下修改：

- 修改 qla2300 驱动程序以执行故障恢复功能。这使您能够利用多个磁盘路径，如果首选路径失败，可以进行故障恢复。默认情况下没有设置这个功能。

  使用脚本进行第一次修改（这个脚本在使用 CSM 安装过程中重启之前运行）。实现这种功能的脚本在 /csminstall/csm/scripts/installprereboot/ 目录中。这个脚本包含了以下命令：

  ```
  #! /bin/bash
  # Adds lines to /etc/modules.conf to enable failover for
  the qla2300 drivers
  echo "options qla2300 ql2xfailover=1 ql2xmaxsectors=512
  ql2xmaxsgs=128" >>
  /etc/modules.conf
  echo "Updating initrd with new modules.conf set up"
  mkinitrd -f /boot/initrd-`uname -r`.img `uname -r`
  ```

- 设置每台主机上到磁盘的首选路径，使其可以匹配每个 DS4500 上的设置。使用通过 HBA0 看到的奇数编号的数组，以及通过 HBA1 看到的偶数编号的数组。

  第二个修改需要在对存储节点重新安装时手动执行。详细信息在 在存储服务器上定义 HBA 配置 一节中介绍。

火龙果•整理
uml.org.cn

GPFS 网络调优

将该示例中提供的几行代码添加到每个节点上的 /etc/sysctl.conf 文件中，用来对 GPFS 网络进行调优。这是通过 CSM 使用的 post-reboot 安装脚本完成的。这个脚本位于 /csminstall/csm/scripts/installpostreboot 目录中，包含以下内容：

```
FILE=/etc/sysctl.conf

# Adds lines to /etc/sysctl.conf for GPFS network tuning
echo "# CSM added the next 8 lines to the post installation
script for
         GPFS network tuning" >> $FILE
echo "# increase Linux TCP buffer limits" >> $FILE
echo "net.core.rmem_max = 8388608" >> $FILE
echo "net.core.wmem_max = 8388608" >> $FILE
echo "# increase default and maximum Linux TCP buffer
sizes" >> $FILE
echo "net.ipv4.tcp_rmem = 4096 262144 8388608" >> $FILE
echo "net.ipv4.tcp_wmem = 4096 262144 8388608" >> $FILE
echo "# increase max backlog to avoid dropped packets" >>
$FILE
echo "net.core.netdev_max_backlog=2500" >> $FILE

# Following lines are not related to GPFS tuning
echo "# Allow Alt-SysRq" >> $FILE
echo "kernel.sysrq = 1" >> $FILE
echo "# Increase ARP cache size" >> $FILE
echo "net.ipv4.neigh.default.gc_thresh1 = 512" >> $FILE
echo "net.ipv4.neigh.default.gc_thresh2 = 2048" >> $FILE
echo "net.ipv4.neigh.default.gc_thresh3 = 4096" >> $FILE
echo "net.ipv4.neigh.default.gc_stale_time = 240" >>
$FILE

# Reset the current kernel parameters
sysctl -p /etc/sysctl.conf
```

分发 GPFS 可移植层

GPFS 可移植层（PL）是内核特有的，必须为集群中的每个操作系统级别单独创建。PL 的作用和为示例集群创建它的过程将在 生成并安装可移植层 一节中详述。CSM 使用 CFM 文件传输工具来管理 PL 二进制文件的分发。将 PL 二进制文件拷贝到管理服务器上的 /cfmroot/usr/lpp/mmfs/bin 目录中，并对文件进行命名，使它们只会分发到相关组中具有特定内核版本的节点上。例如：

```
/cfmroot/usr/lpp/mmfs/bin/dumpconv._<nodegroup>
/cfmroot/usr/lpp/mmfs/bin/lxtrace._<nodegroup>
/cfmroot/usr/lpp/mmfs/bin/mmfslinux._<nodegroup>
/cfmroot/usr/lpp/mmfs/bin/tracedev._<nodegroup>
```

注意在大型集群中，为了减轻 CFM 的负载，可以将这 4 个文件添加到定制 RPM 中，并使用上文介绍的 GPFS RPM 安装方法与 GPFS 一起安装。

自动将新节点添加到 GPFS 集群中

火龙果•整理
uml.org.cn

简单地安装 GPFS RPMS 和可移植层无法在新安装的节点上挂载并配置 GPFS 集群中的文件系统。在手工集群中，这可以手工进行管理。然而，扩展到更大规模的集群之后，就需要自动化完成这个步骤。这可以通过使用 CSM 的监视功能来完成：监视整个新节点的安装过程，并启动一个脚本在集群中的新节点上配置和挂载 GPFS。

清单 1 给出了一个示例脚本，可以用来配置 GPFS。您可能需要针对自己的配置对这个脚本稍加修改。清单 1 提供了一些基本内容。这个脚本会接收节点名（由 CSM 监视器传入），将其添加到 GPFS 集群中，并尝试在这个节点上启动 GPFS，这个过程会提供一点错误处理功能。

清单 1. 配置 GPFS 使用的示例脚本

```
#!/bin/bash

# CSM condition/response script to be used as a response
to the InstallComplete
# condition.  This will attempt to add the node to the GPFS
cluster, dealing
# with some common failure conditions along the way.  Only
trivial attempts are
# made at problem resolution, advanced problems are left
for manual
# intervention.

# Note requires the GPFS gpfs-nodes.list file.  This file
should contain a list
# of all nodes in the GPFS cluster with client/manager and
# quorum/non-quorum details suitable for passing to the
mmcrcluster command.

# Output is sent to /var/log/csm/

# Returned error codes:
# 1 - GPFS is already active
# 2 - unable to read the gpfs-nodes.list file
# 3 - node name not present in the gpfs-nodes.list file
# 4 - node is a quorum node
# 5 - unable to add node to cluster (mmaddnode failed)
# 6 - unable to start GPFS on the node (mmstartup failed)

# set this to the location of your node list file
gpfs_node_list=/etc/gpfs-nodes.list
# set this to the interface GPFS is using for communication
gpfs_interface=eth1

PATH=$PATH:/usr/lpp/mmfs/bin # ensure GPFS binaries are
in the PATH
log_file=/var/log/csm/`basename $0`.log # log to
/var/log/csm/
touch $log_file
```

火龙果•整理
uml.org.cn

```
# Get the node short name as set by RSCT condition ENV var
ERRM_RSRC_NAME
node=`echo $ERRM_RSRC_NAME | cut -d. -f1`

(

[ ! -r "$gpfs_node_list" ] && echo " ** error: cannot read
GPFS
node list $gpfs_node_list" && exit 2

echo
echo "--- Starting run of `basename $0` for $node at `date`"

# Is the node a quorum node? If so exit.
quorum_status=`grep $node $gpfs_node_list | cut -d: -f2
| cut -d- -f2`
if [ -n "$quorum_status" ]; then
        if [ "$quorum_status" = "quorum" ]; then
                echo "** error: this is a quorum node,
stopping..."
                exit 4
        else
                node_s=`grep $node $gpfs_node_list | cut
-d: -f1`
        fi
else
        echo "** error: could not find node $node in GPFS
node list $gpfs_node_list"
        exit 3
fi

# Find out if the node is already part of the cluster
if mmlscluster | grep $node &gt;/dev/null; then

        # check the node status
        status=`mmgetstate -w $node | grep $node | awk
'{print $3}'`
        if [ "$status" = "active" ]; then
                echo "** error: this node already appears
to have GPFS active!"
                exit 1
        fi

        # attempt to remove node from cluster
        echo "Node $node is already defined to cluster,
removing it"
```

火龙果•整理
uml.org.cn

```
        # attempt to disable storage interface on node
        if ssh $node $ifdown $gpfs_interface; then
                mmdelnode $node
                ssh $node ifup $gpfs_interface
        else
                echo "** error: could not ssh to $node, or
ifdown $gpfs_interface failed"
        fi
fi

# try to add node to GPFS cluster
if mmaddnode $node; then
        echo "Successfully added $node to GPFS cluster,
starting GPFS on $node"
        if mmstartup -w $node; then
                echo "mmstartup -w $node succeeded"
        else
                echo "** error: cannot start GPFS on $node,
please investigate"
                exit 6
        fi
else
        echo "** error: could not add $node to GPFS cluster"
        exit 5
fi

) &gt;&gt;$log_file 2&gt;&1
```

当节点完成基本操作系统的安装时，您可以使用 CSM 自动运行如清单 1 所示的脚本，这样在节点引导时，就可以自动挂载 GPFS 文件系统了。首先需要将这个脚本定义为 CSM 监视器中的一种响应机制。例如：mkresponse -n SetupGPFS -s /path/to/script/SetupGPFS.sh SetupGPFS。
现在就有一个名为 SetupGPFS 的响应了，它会运行这个脚本。接下来应该将这个响应关联到默认的 CSM 条件 NodeFullInstallComplete，方法如下： startcondresp NodeFullInstallComplete SetupGPFS。
现在每当安装一个新节点时，CSM 都会在管理服务器上自动运行这个脚本。在 CSM 管理服务器上，现在当运行 lscondresp 命令时，应该可以看到 NodeFullInstallComplete 条件与 SetupGPFS 响应关联。这个条件或响应应该被列为 Active。

解决 ROM 溢出问题

xSeries 346 上可用 ROM 的空间数量存在一个问题，在引导时可能会出现 PCI 分配错误。消息说明系统 ROM 空间已满，没有空间来为其他使用 ROM 空间的适配器使用了（更详细内容请参看 参考资料）。
如果启用了 PXE 引导，这个问题就会影响存储节点，其中 Qlogic PCI 适配器没有足够的空间正确进行初始化。针对这个问题的一个解决方案如下；

1. 禁用 GPFS 网络使用的 Broadcom PCI 卡上的 PXE 引导功能。使用下载的诊断工具 b57udiag -cmd，选择设备，然后禁用 PXE 引导。

2. 通过 CSM 利用 PXE 引导来安装节点，然后使用 BIOS 禁用两个 onboard 适配器中的 PXE 引导功能（详细步骤在 按照正确顺序安装存储节点 一节中介绍）。

火龙果•整理
uml.org.cn

避免这个问题的另外一种解决方案是在每台 xSeries 346 上都使用一个 RAID 7K 卡。这可以减少 Qlogic BIOS 所使用的 ROM 数量，并允许 Qlogic BIOS 成功加载，即使启用了 PXE 也没有问题。

在存储服务上定义 HBA 配置

示例集群中 xSeries 346 存储服务器上使用的 HBA 是 IBM DS4000 FC2-133 Host Bus Adapter（HBA）型号的适配器。它也称为 Qlogic 2342 适配器。本例使用了 1.43 版本的固件，以及上一节提到的 v7.01.01-fo qla2300 驱动程序。驱动程序中的 -fo 表示具有故障恢复功能，这不是驱动程序的默认选项。可以通过修改每个存储节点上的 /etc/modules.conf 的设置启用。这是在安装过程中使用 CSM 实现的，将在 配置 Qlogic 故障恢复 一节中介绍。

下一节介绍了更新固件和每个存储服务器上 HBA 设置所需要的步骤，以及重新安装驱动以在两个 HBA 之间启用负载均衡所需要的手工步骤。

下载 HBA 固件

您可以从 IBM System x 支持 Web 站点（请参看 参考资料）上下载 FC2-133 HBA 的固件。固件可以使用 IBM Management Suite Java 或可引导的磁盘和 flasutil 程序进行更新。

配置 HBA 设置

对于这个示例集群来说，需要对 HBA 的默认设置进行以下修改。这些值位于所下载的驱动程序提供的 README 中。您可以使用 Qlogic BIOS 完成这些修改，在引导过程中出现提示符时使用 <ctrl>-q 可进入 Qlogic BIOS，或者使用 MSJ 工具 修改。设置如下：

- 主机适配器设置
  - Loop reset delay：8
- 高级适配器设置
  - LUNs per target：0
  - Enable target reset：Yes
  - Port down retry count：12

安装 IBM Management Suite Java

IBM FAStT Management Suite Java（MSJ）是一个基于 Java 的 GUI 应用程序，可以管理存储服务器中的 HBA。这个工具可以用来进行配置和诊断。有关下载软件的链接，请参看 参考资料。

示例设置使用 CSM 将 MSJ 作为 GPFS 安装的一部分在每个存储节点上进行安装。这个二进制文件是包含 GPFS RPM 的 tar 文件的一部分，这个文件是在 CSM 节点安装过程中由 CFS 进行分发的。由一个 post 脚本解压这个 tar 文件，接着运行 tar 文件里面包含的安装脚本。本文的示例在这次安装中使用了 32 位的 FAStT MSJ 来防止在安装 64 位版本时可能出现的问题。示例脚本使用了下面的命令安装 MSJ：FAStTMSJ*_install.bin -i silent。

这会同时安装应用程序和代理。注意由于这是一个 32 位版本的 MSJ，因此即使示例使用了静寂安装模式，代码也会查找并加载 32 位版本的库。因此，使用已经安装好的 32 位版本的 XFree86-libs，以及基本 64 位安装附带的 64 位版本。32 位库包含在 XFree86-libs-4.3.0-78.EL.i386.rpm 文件中，该文件包含在 tar 文件中。这个 rpm 的安装是通过 install.sh 脚本进行的，然后会安装 MSJ。

配置到磁盘的路径和负载均衡

每个存储节点上都需要使用 MSJ 手工配置到 DS4500s 上的磁盘阵列的路径，以及在每台计算机上两个 HBA 卡之间实现负载均衡。如果没有执行这种配置，默认情况下磁盘阵列只能通过每台计算机上的第一个适配器 HBA0 以及每个 DS4500s 上的控制器 A 进行访问。通过将磁盘分布在 HBA 之间，以及 DS4500s 的控制器之间，就可以实现负载均衡，并提高后端的性能。

注意负载均衡的配置是一个手工执行的步骤，每次重新安装存储节点时，必须在每个存储节点上都重新执行。对于这个示例集群来说，配置负载均衡的步骤如下所示：

1. 从新安装的具有 xforwarding 设置（ssh <nodename> -X）的服务器的本地计算机上打开一个新窗口。

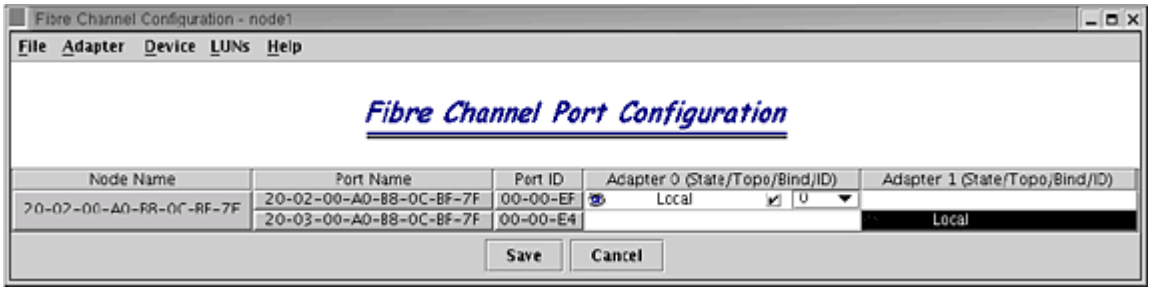
2. 在一个会话中输入 # qlremote。

火龙果•整理
uml.org.cn

3. 在另外一个会话中运行 # /usr/FAStT MSJ & 以启动 MSJ GUI。

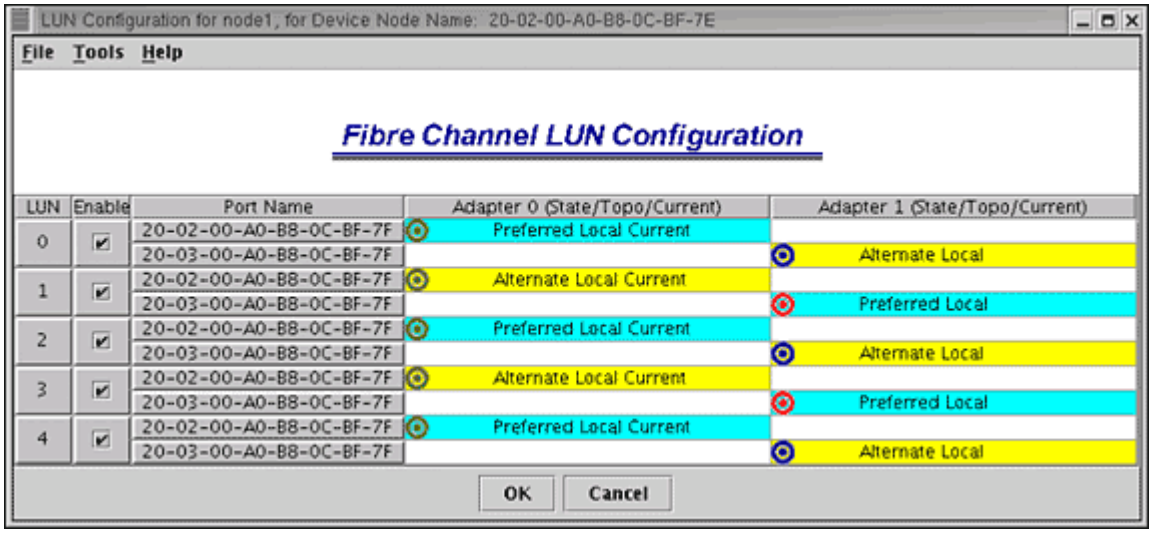
4. 在 MSJ GUI 中，高亮显示 HBA 选项卡下面的一个适配器，并选择 Configure。此时会出现一个类似于图 1 所示的窗口。

图 1. 选择 DS4500 时 MSJ 的视图



5. 要启用负载均衡，请右键单击节点名高亮显示存储子系统，然后从菜单中选择下面的内容：LUNs > Configure LUNs。这样就会出现 LUN 配置窗口。您可以通过选择 Tools > Load Balance 自动配置负载均衡。现在会看到一个类似于图 2 所示的窗口。

图 2. 在配置故障恢复时 MSJ 的视图



6. 在配置好逻辑驱动器之后，LUN 配置窗口就会关闭，并将配置保存到 Port Configuration 窗口中的主机系统（默认密码是 config）。如果配置成功保存，就会看到一条确认消息。配置会保存到一个名为 /etc/qla2300.conf 的文件中。有几个新的选项应该已经被添加到 /etc/modules.conf 的 qla2300 驱动程序行中，说明这个文件已经存在且应该能够使用。

7. 切换回启动 qlremote 进程的窗口，并使用 <ctrl>-c 命令停止这个进程。这是非常重要的一步。

8. 要启用新配置，请重新加载 qla2300 驱动模块。如果磁盘被挂载在一个连接到使用该驱动程序的适配器的 Fibre Channel 子系统上，那就无法执行这种操作。配置主机适配器驱动程序，通过初始 RAM 磁盘进行加载，初始 RAM 磁盘将在引导过程中加载适配器模块时对冗余磁盘应用配置数据。注意：只要逻辑磁盘的配置发生变化，都必须执行这个过程将有效配置保存到磁盘上。

在一个具有多个存储节点并且需要进行负载均衡的配置中，使用 MSJ 最有效的方法是保持 MSJ 在一个节点上打开，然后在其他每个节点上运行 qlremote，并使用一个 MSJ 会话连接到其他相应的会话。

火龙果•整理
uml.org.cn

配置 GPFS 集群

本节将详细介绍在创建 GPFS 集群过程中所需要的步骤。此处假设所有节点都已经按照本文前面的介绍进行了安装和配置，或者已经手工执行了下面的配置：

- 每台计算机上已经安装了 GPFS RPM。
- PATH 已经修改为包括 GPFS 二进制文件目录。
- 已经配置好存储接口。
- 不用输入密码就可以以根用户身份在节点之间以 ssh 连接。
- 已经在 sysctl 中完成了网络调优设置。
- NSD 服务器可以看到一块 SAN 磁盘。

在本系列文章的 第 3 部分 的“存储架构”一节中可以看到对这个示例集群的 GPFS 架构的详细介绍。请结合 GPFS 文档阅读本节内容（请参看 参考资料），尤其是以下内容：

- *GPFS V2.3 Administration and Programming Reference*，其中包含了对很多管理任务和 GPFS 命令的详细介绍。
- *GPFS V2.3 Concepts, Planning, and Installation Guide*，详细介绍了对 GPFS 集群的规划考虑，以及在安装新集群过程中所执行的步骤。
- *GPFS V2.3 Problem Determination Guide*，其中包含了进行故障诊断时所包含的步骤，以及 GPFS 的错误消息。

生成并安装可移植层

GPFS 可移植层（PL）是一组二进制文件，需要从源代码开始进行本地编译，以便匹配作为 GPFS 集群一部分的计算机上的 Linux 内核和配置。对于这个示例集群来说，这个过程是在一个存储节点上进行的。所生成的文件会使用 CSM 和 CFM 拷贝到每个节点上。（详细信息请参看 分发 GPFS 可移植层 一节的内容）。这是一个有效的方法，因为所有计算机都使用相同的架构，并且使用的是相同的内核。编译 GPFS PL 的命令可以在 /usr/lpp/mmfs/src/README 中找到。这个示例集群使用的步骤如下所示：

1. 导出 SHARKCLONEROOT=/usr/lpp/mmfs/src。
2. 输入 cd /usr/lpp/mmfs/src/config, cp site.mcr.proto site.mcr。
3. 编辑新文件 site.mcr 来匹配要使用的配置。确保以下行没有被注释掉：
   - #define GPFS_LINUX
   - #define GPFS_ARCH_X86_64
   - LINUX_DISTRIBUTION = REDHAT_AS_LINUX
   - #define LINUX_DISTRIBUTION_LEVEL 34
   - #define LINUX_KERNEL_VERSION 2042127

   （注意 a # 不代表注释。）
4. 输入 cd /usr/lpp/mmfs/src。

火龙果•整理
uml.org.cn

5. 使用 make World 创建 GPFS PL。

6. 使用 make InstallImages 将 GPFS PL 拷贝到 /usr/lpp/mmfs/bin 目录中。GPFS PL 中包含以下 4 个文件：
   - tracedev
   - mmfslinux
   - lxtrace
   - dumpconv

7. 拷贝一组这些文件（每个对应内核使用一个）到 CSM 结构中，以便使用 CFM 进行分发。

创建 GPFS 集群

您可以使用几个独特的步骤为这个例子创建 GPFS 集群。尽管所有这些步骤都不是必须的，但是处理集群中不同类型的节点（存储节点或其他节点）是一种好方法。

第一个步骤是创建一个只包含存储节点和 quorum 节点的集群：一共有 5 个节点。在创建包含要包括的所有节点的存储接口的主机简写名时，请使用描述符文件，描述符文件之后再加上以下信息：

- 管理节点或客户机：定义节点是否构成从中提取配置和文件系统管理器的一个资源池的一部分。示例集群中只包括了这个资源池中的存储节点。
- Quorum 或 nonquorum：定义节点是否应该算做一个 quorum 节点。示例集群中的 quorum 节点是存储节点和 tiebreaker 节点 quor001。

创建集群的命令如下：

```
mmcrcluster -n stor.nodes -C gpfs1 -p stor001_s -s
stor002_s -r /usr/bin/ssh -R
/usr/bin/scp
```

- -C 标志设置了集群名。
- -p 设置主配置服务器节点。
- -s 设置次配置服务器节点。
- -r 为 GPFS 使用的远程 shell 程序设置完整路径。
- -R 设置 GPFS 使用的远程文件拷贝程序。

下面是示例集群中使用的 stor.nodes 节点描述符文件：

```
stor001_s:manager-quorum
stor002_s:manager-quorum
stor003_s:manager-quorum
stor004_s:manager-quorum
quor001_s:client-quorum
```

火龙果•整理
uml.org.cn

对要加入集群中的所有其他节点，例如计算节点、用户节点和管理节点，请在以后步骤中添加类似于 <nodename>_s:client-nonquorum 的项。

在 quorum 节点上启用 unmountOnDiskFail

下一个步骤是在 tiebreaker 节点上使用 mmchconfig unmountOnDiskFail-yes quor001_s 启用 unmountOnDiskFail 选项。这可以防止 SAN 配置中的虚假磁盘错误被误报给文件系统管理器。

定义网络共享磁盘

下一个步骤是使用 mmcrnsd -F disc#.desc 命令创建 GPFS 所使用的磁盘。运行这个命令为每个磁盘创建一个全局名，这是一个必要的步骤，因为磁盘在 GPFS 集群中的每个节点上可能有不同的 /dev 名。对 GPFS 文件系统要使用的所有磁盘运行该命令。现在，为每个磁盘定义主 NSD 服务器和次 NSD 服务器；它们用于代表 NSD 客户机的 I/O 操作，NSD 客户机不能够本地访问 SAN 存储。

定义为 NSD 的磁盘的磁盘描述符包含在一个文件里，-F 标志就用来指向这个文件。为了保证示例集群的可管理能力，请在每个 DS4500 的 LUN 上执行一次该过程，并在 tiebreaker 磁盘上执行一次该过程。每个磁盘阵列或 DS4500 上的每个 LUN 在所使用的文件中都有一个描述符。下面是从 disk1.desc 中摘录出来的一行：

```
sdc:stor001_s:stor002_s:dataAndMetadata:1:disk01_array01S
```

下面是该行中的几个字段，按顺序依次显示：

- 主 NSD 服务器上的本地磁盘名
- 主 NSD 服务器
- 次 NSD 服务器
- 数据类型
- 故障组
- 所生成的 NSD 的名称

通过使用上面的描述符文件，可以在这个配置中定义下面的 3 个故障组：

- 第一个 DS4500 中的磁盘，即 disk01。
- 第二个 DS4500 中的磁盘，即 disk02。
- quorum 节点上的 tiebreaker 磁盘。

启动 GPFS

下一个步骤是使用下面的命令在集群范围内启动 GPFS：

1. 在所有 NSD 服务器上启动 GPFS，同时防止 NSD 被标记为 down 状态。请使用下面的命令：mmstartup -w stor001_s,stor002_s,stor003_s,stor004_s。
2. 在所有不是 NSD 服务器的其他节点（包括 tiebreaker 节点）上启动 GPFS。请使用下面的命令：mmstartup -w quor001_s,mgmt001_s,...
3. 从管理节点上在所有计算节点上启动 GPFS。请使用下面的命令： dsh -N ComputeNodes /usr/lpp/mmfs/bin/mmstartup。

火龙果•整理
uml.org.cn

4. 检查所有节点的状态，方法为：监视当前文件系统管理器上的 /var/adm/log/mmfs.log.latest 文件（使用 mmlsmgr <filesystem> 命令时可以找到），并监视下面这个命令的输出结果： mmgetstate -w <nodenames> dsh -N ComputeNodes /usr/lpp/mmfs/bin/mmgetstate。

这个方法看起来可能过分谨慎了，但是它已经被作为一种可扩充的方法，可在规模巨大的集群中使用。上面这些步骤另外一种方法是使用命令 mmstartup -a。这对于小型集群来说可以正常工作，但是对于大型集群来说，可能会需要花费很长一段时间来返回，在大型集群中，可能有些节点会由于各种原因而变得不可访问，例如网络问题。

创建 GPFS 文件系统

对于本例来说，使用所有定义为 GPFS 的 NSD 创建了一个大型的 GPFS 文件系统。注意所使用的命令使用上面的 mmcrnsd 命令中的不同磁盘描述符文件作为参数。这需要将创建 NSD 的每个步骤中的输出结果连接到一个文件中。

示例集群使用了以下设置：

- 所有 NSD（使用 -F 设置）
- Mountpoint：/gpfs
- Automount：是（使用 -A 设置）
- Blocksize：256KB（使用 -B 设置）
- 复制：数据和元数据都有两个拷贝（使用 -m, -M, -r, -R 设置）
- 挂载文件系统的节点估计数：1200（使用 -n 设置）
- 启用配额（使用 -Q 设置）

下面是完整的命令：

```
 mmcrfs /gpfs /dev/gpfs -F disk_all.desc -A yes -B 256K
-m 2 -M 2

-r 2 -R 2 -n
                                          1200 -Q yes
```

在创建 /gpfs 之后，要手工进行第一次挂载。然后，如果启用了 automount，就可以在节点启动 GPFS 时自动挂载。

启用配额

上面 mmcrfs 命令的 -Q 标志在 /gpfs 文件系统上启用配额。可以为每个用户或组用户定义配额。默认配额已经进行了设置，可以适用于任何新用户或组。可以使用命令 mmdefquotaon 启用默认配额。使用 mmdefedquota 命令编辑默认配额。这个命令打开了一个编辑窗口，其中可以指定配额的范围。下面的示例演示了配额范围的设置：

```
gpfs: blocks in use: 0K, limits (soft = 1048576K, hard =
2097152K)
                                  inodes in use: 0,
limits (soft = 0, hard = 0)
```

使用 mmedquota -u <username> 命令可以为一个用户或组编辑特定配额。用户可以使用命令 mmlsquota 显示自己的配额。超级用户可以使用 mmrepquota gpfs 命令显示文件系统的配额状态。

调优

火龙果•整理
uml.org.cn

这个集群经过配置后，当服务器引导时，使用 mmchconfig autoload=yes 命令在 /etc/inittab 中添加一项，GPFS 就可以自动启动。

使用 GPFS pagepool 可以缓存用户数据和文件系统元数据。pagepool 机制允许 GPFS 将读请求与写请求异步实现。增加 pagepool 的大小可以增大 GPFS 可以缓存的数据或元数据量，而不需要执行同步 I/O。pagepool 的默认值是 64 MB。最大 GPFS pagepool 大小是 8 GB。所允许的最小值是 4 MB。在 Linux 系统上， pagepool 的最大值是计算机物理内存的一半

pagepool 的理想大小取决于应用程序以及对其重新访问的数据的有效缓存的需要。如果系统中的应用程序可以访问大型文件、重用数据、可以从 GPFS 数据预取获益或具有随机 I/O 模式，增加 pagepool 的值可能会进一步获益。然而，如果该值设置得太大，那么 GPFS 就无法启动。

对于示例集群来说，集群中所有节点的 pagepool 所使用的值都是 512 MB。

优化网络设置

要对网络性能进行优化，GPFS 通过将存储网络适配器的 MTU 的大小设置为 9000 来启用巨帧。始终启用 /proc/sys/net/ipv4/tcp_window_scaling，因为这是默认设置。TCP 窗口设置在安装时使用 CSM 脚本进行了调优，方法是将下面几行添加到 NSD 服务器和 NSD 客户机的 /etc/sysctl.conf 文件中：

```
# increase Linux TCP buffer limits
net.core.rmem_max = 8388608
net.core.wmem_max = 8388608
# increase default and maximum Linux TCP buffer sizes
net.ipv4.tcp_rmem = 4096 262144 8388608
net.ipv4.tcp_wmem = 4096 262144 8388608
# increase max backlog to avoid dropped packets
net.core.netdev_max_backlog=2500
```

配置 DS4500 设置

存储服务器缓存设置如果设置错误，可能会影响 GPFS 性能。本例在 DS4500s 上使用了以下设置，这也是 GPFS 文档的推荐设置：

- Read cache：启用
- Read ahead multiplier：0
- Write cache：禁用
- Write cache mirroring：禁用
- Cache block size：16K