



Red Hat Enterprise Linux 5

群集套件概述

红帽企业版 Linux 5 的红帽群集套件

版 3

Red Hat Enterprise Linux 5 群集套件概述

红帽企业版 Linux 5 的红帽群集套件
版 3

Landmann
rlandmann@redhat.com

法律通告

Copyright © 2009 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

摘要

Red Hat Cluster Suite概述提供了对用于Red Hat Enterprise Linux 5 的Red Hat Cluster Suite的概述。

目录

简介	2
1. 反馈	2
第 1 章 RED HAT CLUSTER SUITE概述	4
1.1. 群集基础	4
1.2. RED HAT CLUSTER SUITE INTRODUCTION	5
1.3. CLUSTER INFRASTRUCTURE	6
1.3.1. 群集管理	6
1.3.2. 锁管理	7
1.3.3. Fencing	7
1.3.4. 群集配置系统	11
1.4. 高可用性服务管理	13
1.5. RED HAT GFS	15
1.5.1. 出众的性能和灵活性	16
1.5.2. 性能、灵活性、合适的价格	17
1.5.3. 经济性和性能	18
1.6. 群集逻辑卷管理器	18
1.7. 全球网络块设备	21
1.8. LINUX 虚拟服务器	22
1.8.1. Two-Tier LVS Topology	24
1.8.2. Three-Tier LVS Topology	25
1.8.3. 路由方法	26
1.8.3.1. NAT 路由	26
1.8.3.2. 直接路由	27
1.8.4. 持久性和防火墙标记	29
1.8.4.1. Persistence	29
1.8.4.2. 防火墙标记	29
1.9. 群集管理工具	29
1.9.1. Conga	29
1.9.2. 群集管理图形界面	32
1.9.2.1. Cluster Configuration Tool	32
1.9.2.2. Cluster Status Tool	34
1.9.3. 命令行管理工具	35
1.10. LINUX 虚拟服务器管理图形用户界面	36
1.10.1. CONTROL/MONITORING	37
1.10.2. GLOBAL SETTINGS	38
1.10.3. REDUNDANCY	40
1.10.4. VIRTUAL SERVERS	41
1.10.4.1. 「VIRTUAL SERVER」分支	41
1.10.4.2. 「REAL SERVER」分支	43
1.10.4.3. EDIT MONITORING SCRIPTS Subsection	45
第 2 章 RED HAT CLUSTER SUITE 组件摘要	48
2.1. 群集组件	48
2.2. MAN PAGE	53
2.3. 兼容的硬件	55
附录 A. 修订记录	56
索引	57

简介

本文档提供了对 Red Hat Enterprise Linux 5 中 Red Hat Cluster Suite 的高级概述，其结构如下：

- [第 1 章 Red Hat Cluster Suite 概述](#)
- [第 2 章 Red Hat Cluster Suite 组件摘要](#)

虽然本文档的信息只是一个概述，但您应该具有深入的 Red Hat Enterprise Linux 的相关知识并理解服务器计算的概念，这样才能充分地理解这些信息。

关于使用 Red Hat Enterprise Linux 的更多信息，请参考以下资源：

- 《*Red Hat Enterprise Linux 安装指南*》— 提供安装 Red Hat Enterprise Linux 5 的信息。
- 《*Red Hat Enterprise Linux 部署指南*》— 提供关于 Red Hat Enterprise Linux 5 的部署、配置和管理的信息。

有关 Red Hat Enterprise Linux 5 中 Red Hat Cluster Suite 的更多信息，请参考以下资源：

- 《*配置和管理红帽群集系统*》— 提供安装、配置和管理红帽群集组件的信息。
- *Logical Volume Manager Administration* — Provides a description of the Logical Volume Manager (LVM), including information on running LVM in a clustered environment.
- 《*全球文件系统：配置和管理*》— 提供安装、配置和维护红帽 GFS（红帽全球文件系统）的信息。
- 《*全球文件系统 2：配置和管理*》— 提供安装、配置和维护红帽 GFS2（红帽全球文件系统 2）的信息。
- 《*使用设备映射器多路径*》— 提供关于使用 Red Hat Enterprise Linux 5 设备映射器多路径信息。
- 《*在全球文件系统里使用 GNBD*》— 一个关于在红帽全球文件系统（GFS）里使用全球网路块设备（Global Network Block Device, GNBD）的概述。
- 《*Linux 虚拟服务器管理*》— 提供用 Linux 虚拟服务器（Linux Virtual Server, LVS）配置高性能的系统和服务的信息。
- 《*红帽群集套件发行注记*》— 关于当前的红帽群集套件（Red Hat Cluster Suite）的发行注记。

在 Red Hat Enterprise Linux 文档 CD 和在线网站 <http://www.redhat.com/docs/> 里，您可以找到 Red Hat Cluster Suite 文档和其他 Red Hat 文档的 HTML、PDF 和 PRM 版本。

1. 反馈

如果您发现了排印错误或者有改进本文档的建议，我们希望能听取您的意见。请在 Bugzilla (<http://bugzilla.redhat.com/bugzilla/>) 里提交关于组件 **Documentation-cluster** 的报告。

Be sure to mention the document's identifier:

Cluster_Suite_Overview(EN)-5 (2009-08-18T15:49)

By mentioning this document's identifier, we know exactly which version of the guide you have.

如果您有改进本文档的建议，请尽量将其明确化。如果您发现了错误，请指出章节号以及其周围的相关文字，以便我们尽快找到并更正该错误。

第 1 章 RED HAT CLUSTER SUITE概述

群集系统为关键产品服务提供可靠性、灵活性和可用性。使用 Red Hat Cluster Suite，您可以创建群集来满足对性能、高可用性、负载平衡、灵活性、文件共享以及经济性的需要。本章提供 Red Hat Cluster Suite 组件和功能的概述，它由下列章节组成：

- 第 1.1 节 “群集基础”
- 第 1.2 节 “Red Hat Cluster Suite Introduction”
- 第 1.3 节 “Cluster Infrastructure”
- 第 1.4 节 “高可用性服务管理”
- 第 1.5 节 “Red Hat GFS”
- 第 1.6 节 “群集逻辑卷管理器”
- 第 1.7 节 “全球网络块设备”
- 第 1.8 节 “Linux 虚拟服务器 ”
- 第 1.9 节 “群集管理工具”
- 第 1.10 节 “Linux 虚拟服务器管理图形用户界面”

1.1. 群集基础

群集是由两个或多个计算机（称为 *节点*或*成员*）组成的一起执行任务的系统。主要的群集类型有 4 种：

- 存储型（Storage）
- 高可用性型（High availability）
- 负载平衡型（Load balancing）
- 高性能型（High performance）

存储型群集跨服务器提供一致的文件系统映像，它允许服务器同步地读取和写入单个的共享文件系统。存储型群集将应用程序的安装和修补限制到一个文件系统，简化了存储的管理。而且，采用群集范围的文件系统，存储型群集消除了应用程序数据的冗余拷贝并简化了备份和恢复。Red Hat Cluster Suite 是通过 Red Hat GFS 提供存储群集功能的。

高可用性型群集通过消除单点失效以及在一个节点停止运作时将服务从一个群集节点切换到另外一个节点，提供服务的持续可用性。通常，在高可用性型群集中的服务读写数据（通过以读写方式挂载的文件系统）。因此，高可用性型群集必须维护数据的完整性，因为一个群集节点是从另一个群集节点接管某个服务的控制。客户端中高可用性型群集中产生故障的节点在群集外并不可见。（高可用性型群集有时候被称作故障切换群集。）Red Hat Cluster Suite 通过其高可用性服务管理（High-availability Service Management）组件提供高可用性的群集功能。

负载平衡型群集将网络服务请求分摊在多个群集节点中来平衡请求负载。因为您可以按照负载来调整节点数目，这样就提供了有性价比的灵活性。如果负载平衡型群集中的某个节点不可操作了，负载平衡软件检测到这个故障并将请求重定向到其他群集节点。负载平衡型群集中的节点故障在群集外是不可见的。Red Hat Cluster Suite 通过 Linux 虚拟服务器（Linux Virtual Server, LVS）提供了负载平衡功能。

高性能型群集使用群集节点来执行并行运算。高性能型群集允许应用程序以并行的方式工作，因此提高了应用程序的性能。（高性能型群集也被称为计算群集或网格计算。）



注意

前面总结的群集类型仅反映了基本的配置；实际应用时您可能需要将它们进行合并。

1.2. RED HAT CLUSTER SUITE INTRODUCTION

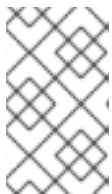
Red Hat Cluster Suite (RHCS) 是软件组件的集成，它可以不同的配置进行部署来满足您对性能、高可用性、负载平衡、灵活性、文件共享和经济性的需要。

RHCS consists of the following major components (refer to [图 1.1 “Red Hat Cluster Suite Introduction”](#)):

- 群集基础结构 — 提供节点以群集方式一起工作所需的基本功能、配置文件管理、成员资格管理、锁管理和安全管理。
- 高可用性服务管理 — 提供在某个节点不可操作时，服务从一个群集节点到另外一个节点的故障切换。
- 群集管理工具 — 用于设置、配置和管理Red Hat群集的配置和管理工具。这些工具和群集基础结构组件、高可用性和服务管理组件、存储组件一起使用。
- Linux 虚拟服务器 (Linux Virtual Server, LVS) — 提供 IP 负载平衡的路由软件。LVS 在一对冗余的服务器中运行，它将客户请求均匀地分发到它后面的真实服务器。

您可为Red Hat Cluster Suite提供以下组件，它们是可选软件包的一部分（但不是Red Hat Cluster Suite的一部分）：

- Red Hat GFS (全球文件系统, Global File System) — 提供和 Red Hat Cluster Suite 一起使用的群集文件系统。GFS 允许多个节点进行块级别的存储共享，就好像每个群集节点都连接至本地存储空间一样。
- 群集逻辑卷管理器 (Cluster Logical Volume Manager, CLVM) — 提供群集存储的卷管理。



注意

When you create or modify a CLVM volume for a clustered environment, you must ensure that you are running the `clvmd` daemon. For further information, refer to [第 1.6 节 “群集逻辑卷管理器”](#).

- 全球网络块设备 (Global Network Block Device, GNBDD) — GFS 的辅助组件，它将块级别的存储设备导出至以太网。这是在 Red Hat GFS 中使用块级别存储的一个经济的方法。

For a lower level summary of Red Hat Cluster Suite components and optional software, refer to [第 2 章 Red Hat Cluster Suite 组件摘要](#).

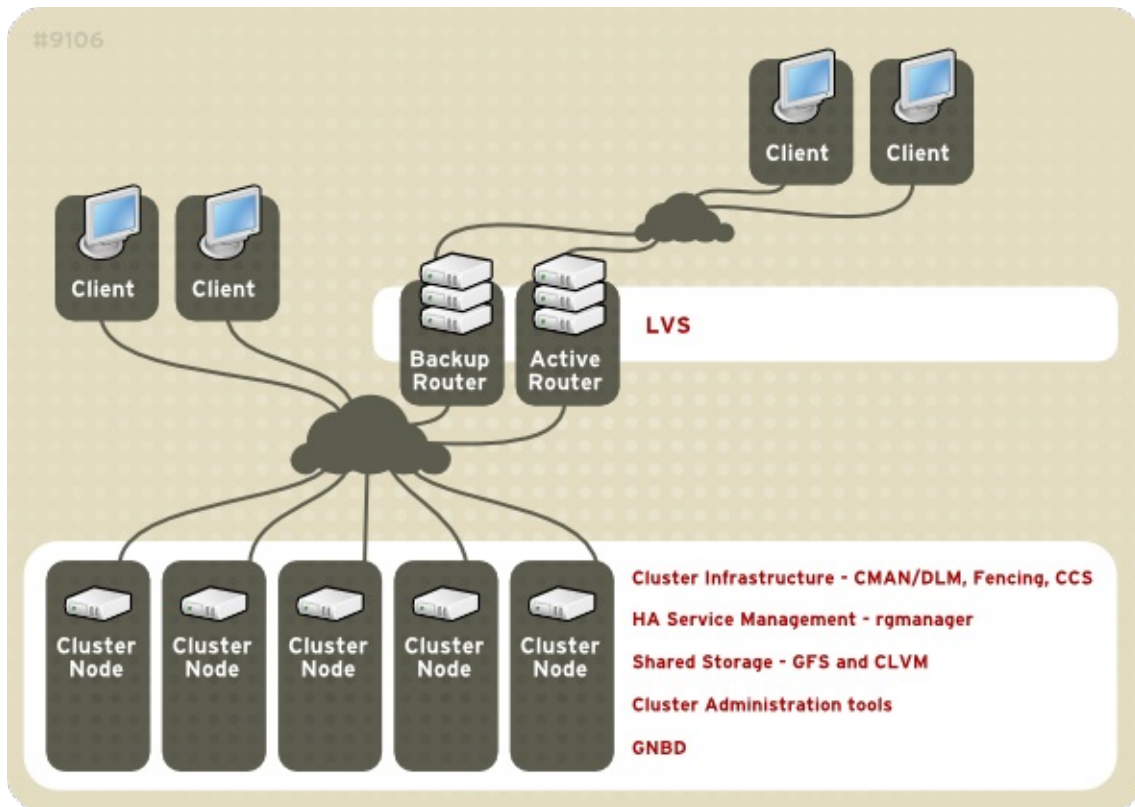


图 1.1. Red Hat Cluster Suite Introduction



注意

图 1.1 “Red Hat Cluster Suite Introduction” includes GFS, CLVM, and GNBD, which are components that are part of an optional package and *not* part of Red Hat Cluster Suite.

1.3. CLUSTER INFRASTRUCTURE

Red Hat Cluster Suite 群集基础结构为一组计算机（称为节点或成员）以群集方式一起工作提供基本的功能。一旦使用群集基础结构组成了群集，您就可以使用其他 Red Hat Cluster Suite 的组件来满足群集需求（例如，设立在 GFS 文件系统上进行文件共享的群集，或服务的故障切换）。群集基础结构提供下面的功能：

- 群集管理
- 锁管理
- Fencing
- 群集配置管理

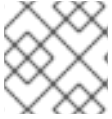
1.3.1. 群集管理

Cluster management manages cluster quorum and cluster membership. CMAN (an abbreviation for cluster manager) performs cluster management in Red Hat Cluster Suite for Red Hat Enterprise Linux 5. CMAN is a distributed cluster manager and runs in each cluster node; cluster management is distributed across all nodes in the cluster (refer to 图 1.2 “CMAN/DLM Overview”).

CMAN keeps track of cluster quorum by monitoring the count of cluster nodes. If more than half the nodes are active, the cluster has quorum. If half the nodes (or fewer) are active, the cluster does not

have quorum, and all cluster activity is stopped. Cluster quorum prevents the occurrence of a "split-brain" condition — a condition where two instances of the same cluster are running. A split-brain condition would allow each cluster instance to access cluster resources without knowledge of the other cluster instance, resulting in corrupted cluster integrity.

Quorum 由群集节点间经由以太网的消息通信决定的。Quorum 也可以由通过以太网和 quorum 磁盘的消息的组合来决定。对于经由以太网的 quorum, quorum 由节点 vote 的 50% 加 1 组成的。而对于经由 quorum 磁盘的 quorum, quorum 由用户指定的条件组成。



注意

在默认情况下, 每个节点都有一个 quorum vote。您也可以配置每个节点拥有多个 vote。

CMAN 通过监控其他群集节点的信息来跟踪成员资格。当群集的成员资格发生变化时, 群集管理器通知其他基础结构组件, 它们再采取合适的行动。例如, 如果节点 A 加入了群集并挂载了节点 B 和 C 已经挂载的 GFS 文件系统, 那么节点 A 就需要额外的 journal 和锁的管理来使用这个 GFS 文件系统。如果群集节点在规定的时间内没有传送消息, 群集管理器将将这个节点从群集里删除, 并通知其他基础结构组件这个节点已经不再是成员了。其他群集基础结构组件在获得这个通知后将决定采取什么样的行动。例如, Fencing 将保护不再是成员的节点。

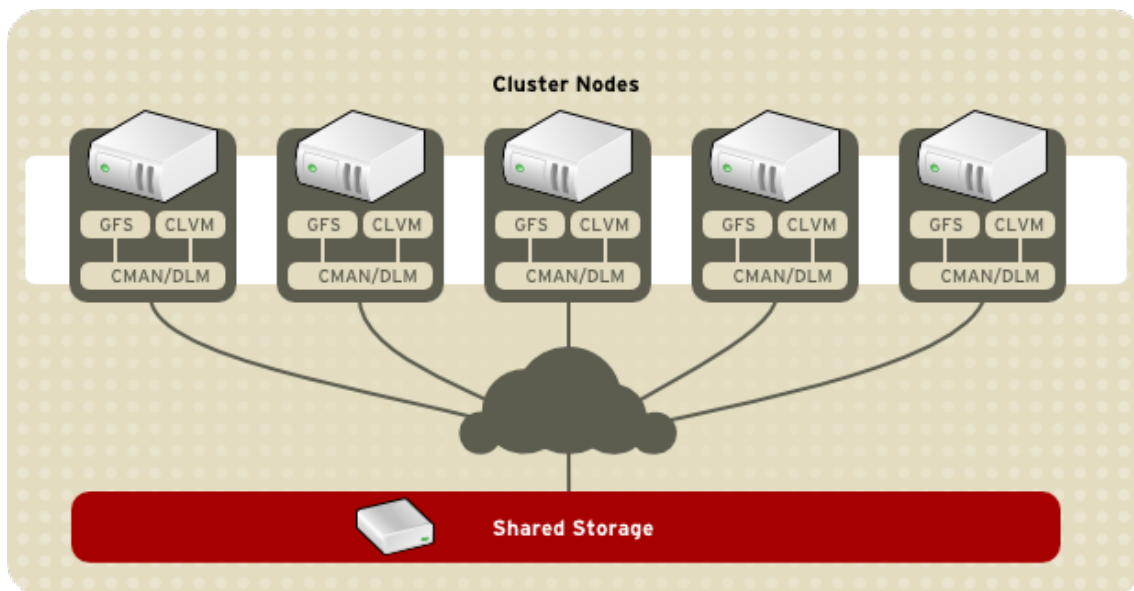


图 1.2. CMAN/DLM Overview

1.3.2. 锁管理

Lock management is a common cluster-infrastructure service that provides a mechanism for other cluster infrastructure components to synchronize their access to shared resources. In a Red Hat cluster, DLM (Distributed Lock Manager) is the lock manager. As implied in its name, DLM is a distributed lock manager and runs in each cluster node; lock management is distributed across all nodes in the cluster (refer to [图 1.2 “CMAN/DLM Overview”](#)). GFS and CLVM use locks from the lock manager. GFS uses locks from the lock manager to synchronize access to file system metadata (on shared storage). CLVM uses locks from the lock manager to synchronize updates to LVM volumes and volume groups (also on shared storage).

1.3.3. Fencing

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. The cluster infrastructure performs fencing through the fence daemon, **fenced**.

当 CMAN 决定某个节点已经发生故障，它将告知其他群集基础结构组件。在收到通知后，**fenced** 将保护（fence）故障节点。其他群集基础结构组件将决定采取什么行动 — 也就是说，它们执行任何必需的恢复。例如，当 DLM 和 GFS 被告知节点故障时，将暂停活动，直到它们检测到 **fenced** 已经完成对故障节点的保护。当确认故障节点已经被保护时，DLM 和 GFS 会执行恢复。DLM 释放对失败节点的锁定；GFS 恢复故障节点的日志。

保护程序根据群集配置文件决定使用哪种保护方法。群集配置文件用两个关键元素定义了保护方法：保护代理（fencing agent）和保护设备（fencing device）。保护程序调用这里定义的保护代理。而保护代理则通过保护设备来保护节点。当保护过程结束时，保护程序将通知群集管理器。

Red Hat Cluster Suite 提供各种保护方法：

- 电源保护（Power fencing）— 该方法使用电源控制器来关闭不可操作的节点。
- 光线通道开关保护（Fibre Channel switch fencing）— 该方法禁用连接至不可操作节点的光线通道端口。
- GNBD fencing — A fencing method that disables an inoperable node's access to a GNBD server.
- 其他保护方法 — 如禁用 I/O 或不可操作节点的电源，包括 IBM Bladecenters, PAP, DRAC/MC, HP ILO, IPMI, IBM RSA II 等。

图 1.3 “Power Fencing Example” shows an example of power fencing. In the example, the fencing program in node A causes the power controller to power off node D. 图 1.4 “Fibre Channel Switch Fencing Example” shows an example of Fibre Channel switch fencing. In the example, the fencing program in node A causes the Fibre Channel switch to disable the port for node D, disconnecting node D from storage.

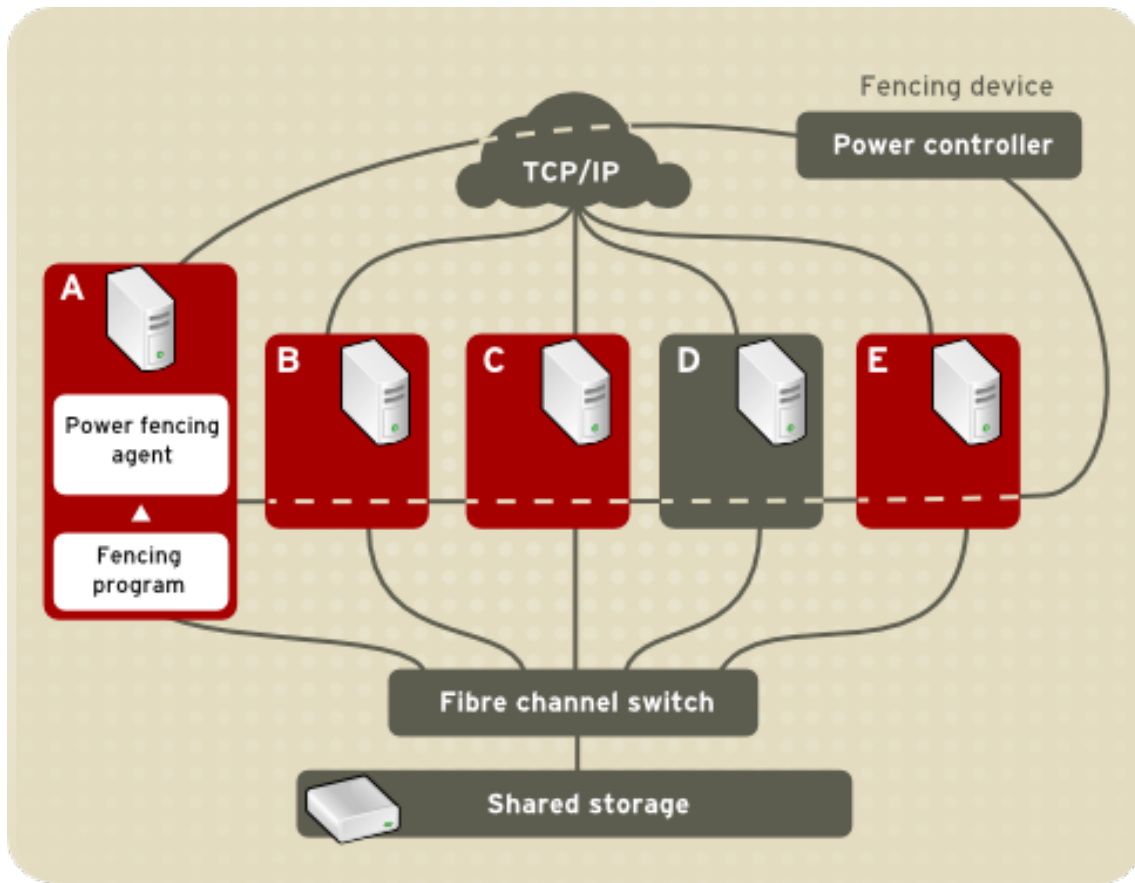


图 1.3. Power Fencing Example

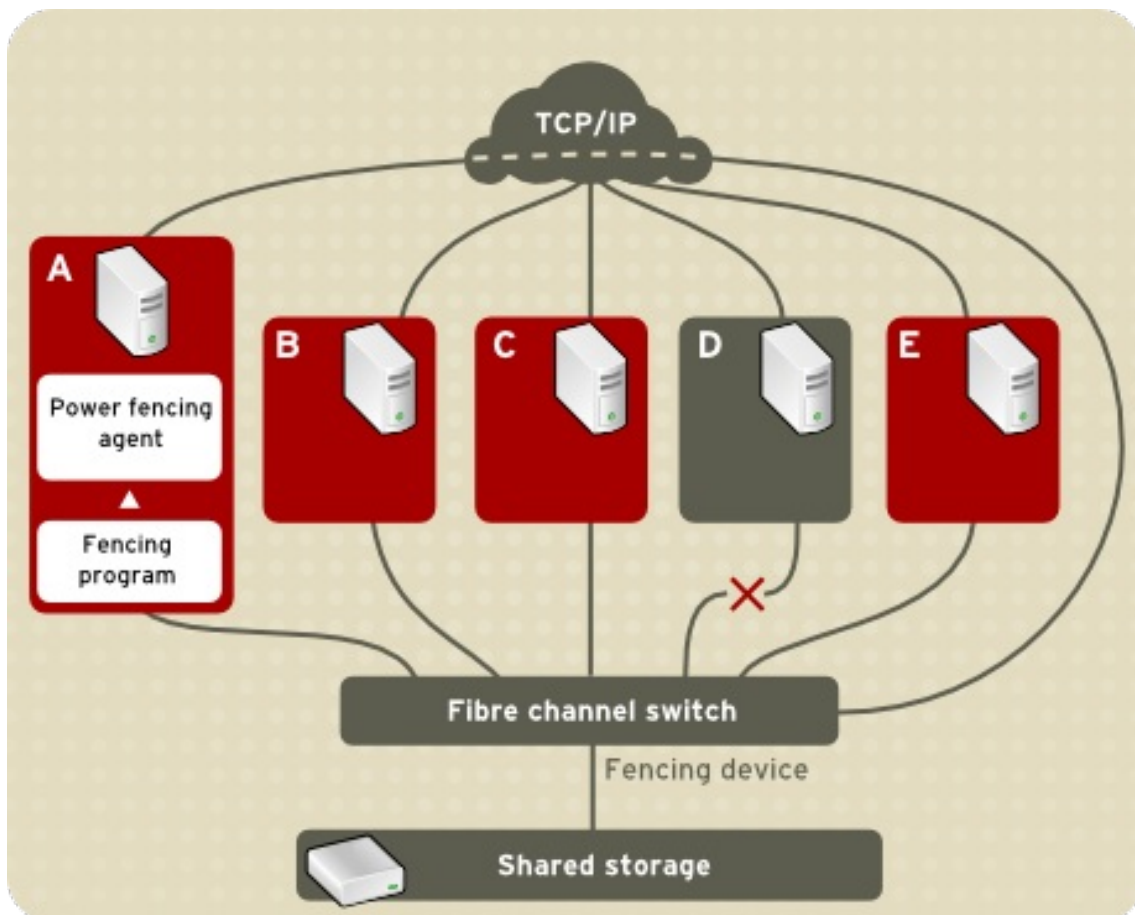


图 1.4. Fibre Channel Switch Fencing Example

指定保护方法通过编辑群集配置文件完成，如分配保护方法名、保护代理（fencing agent）、以及用于群集里每个节点的保护设备（fencing device）。

The way in which a fencing method is specified depends on if a node has either dual power supplies or multiple paths to storage. If a node has dual power supplies, then the fencing method for the node must specify at least two fencing devices — one fencing device for each power supply (refer to [图 1.5 “Fencing a Node with Dual Power Supplies”](#)). Similarly, if a node has multiple paths to Fibre Channel storage, then the fencing method for the node must specify one fencing device for each path to Fibre Channel storage. For example, if a node has two paths to Fibre Channel storage, the fencing method should specify two fencing devices — one for each path to Fibre Channel storage (refer to [图 1.6 “Fencing a Node with Dual Fibre Channel Connections”](#)).

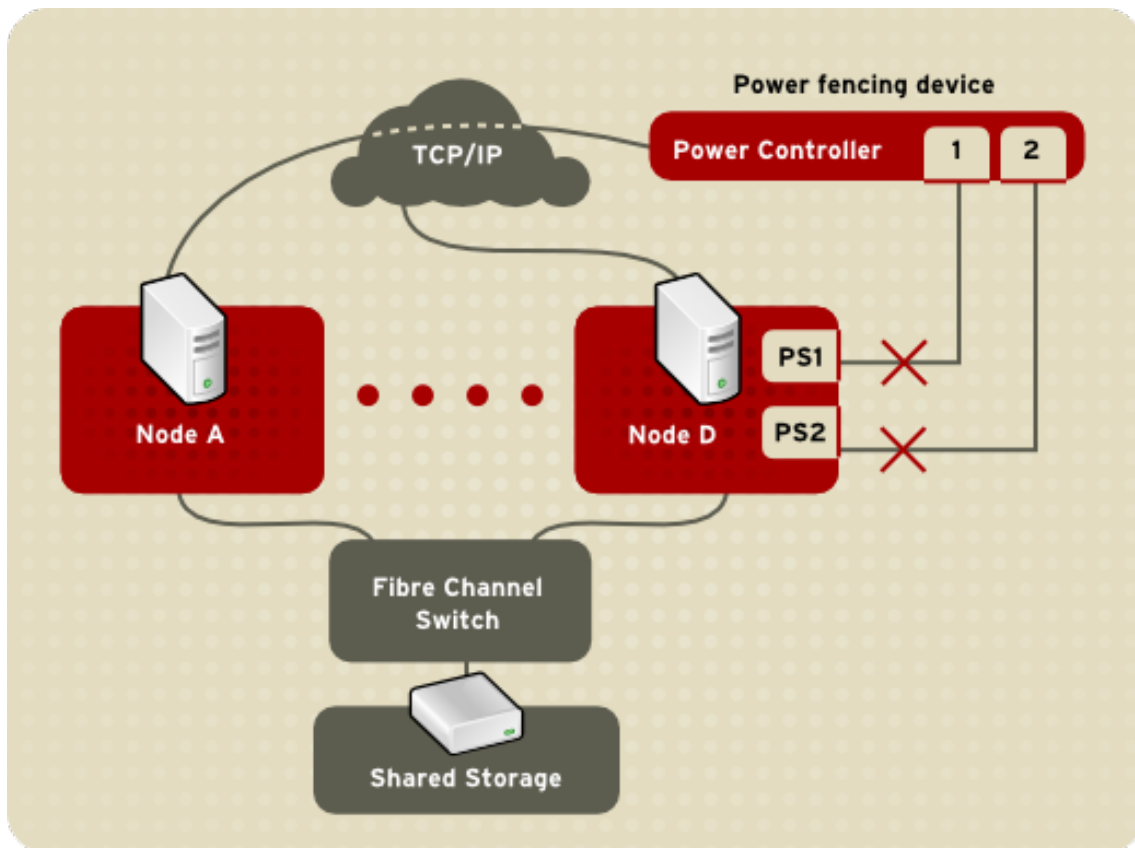


图 1.5. Fencing a Node with Dual Power Supplies

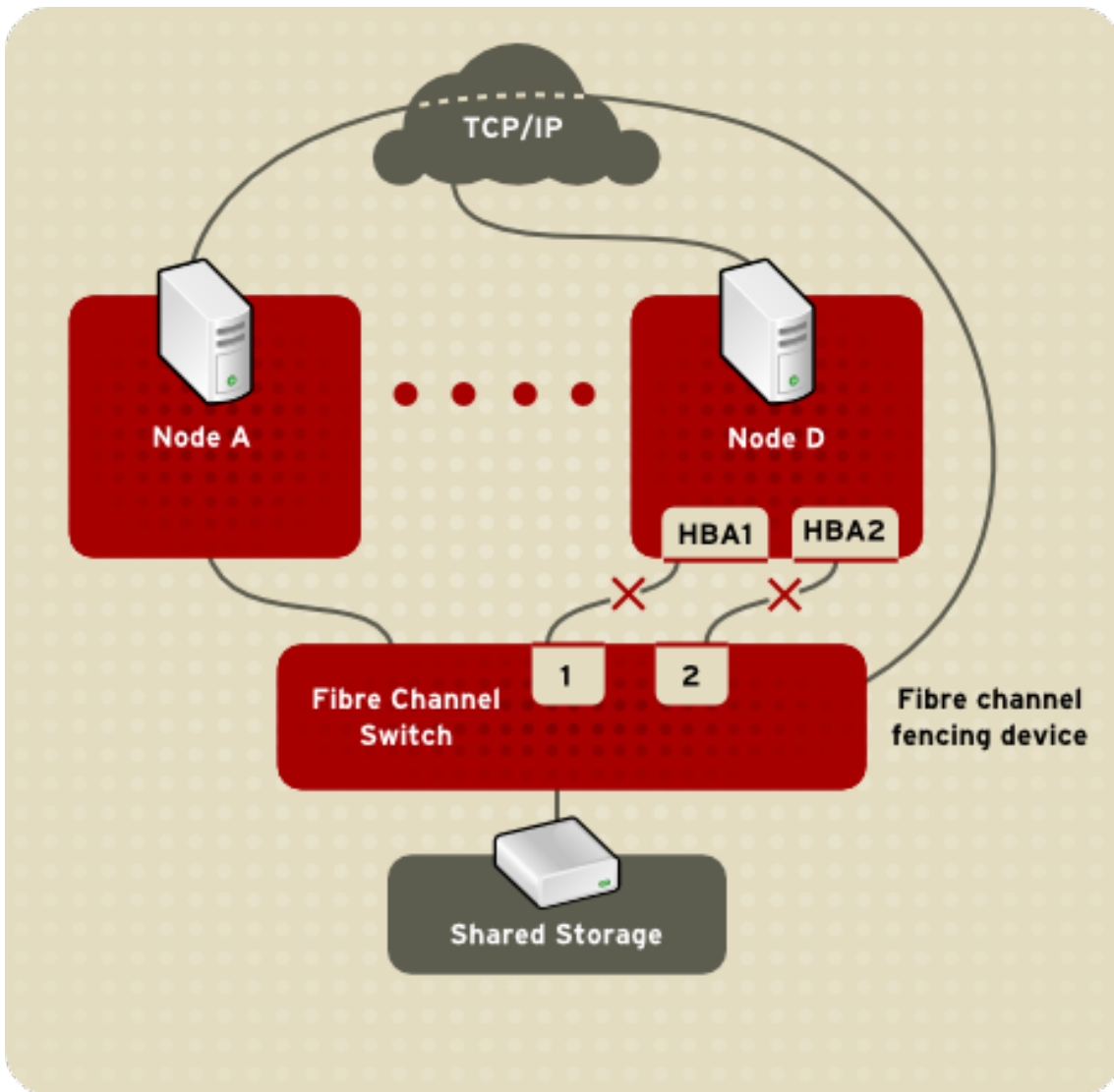


图 1.6. Fencing a Node with Dual Fibre Channel Connections

您可以配置节点具有一个或多个保护方法。当您配置某个节点具有一个保护方法时，这个方法就是保护该节点的唯一可用保护方法。而当您配置某个节点具有多个保护方法时，保护方法将按照群集配置文件里指定的顺序挨个进行层叠 (*cascade*)。如果某个节点出现故障，配置文件里指定的第一个保护方法将对它进行保护。如果第一个保护方法没有成功，指定的第二个保护方法将被使用。如果所有方法都没有成功，保护将从第一个保护方法重新开始，且安装群集配置文件里指定的顺序进行循环，直到这个节点被成功地保护为止。

1.3.4. 群集配置系统

The Cluster Configuration System (CCS) manages the cluster configuration and provides configuration information to other cluster components in a Red Hat cluster. CCS runs in each cluster node and makes sure that the cluster configuration file in each cluster node is up to date. For example, if a cluster system administrator updates the configuration file in Node A, CCS propagates the update from Node A to the other nodes in the cluster (refer to [图 1.7 “CCS Overview”](#)).

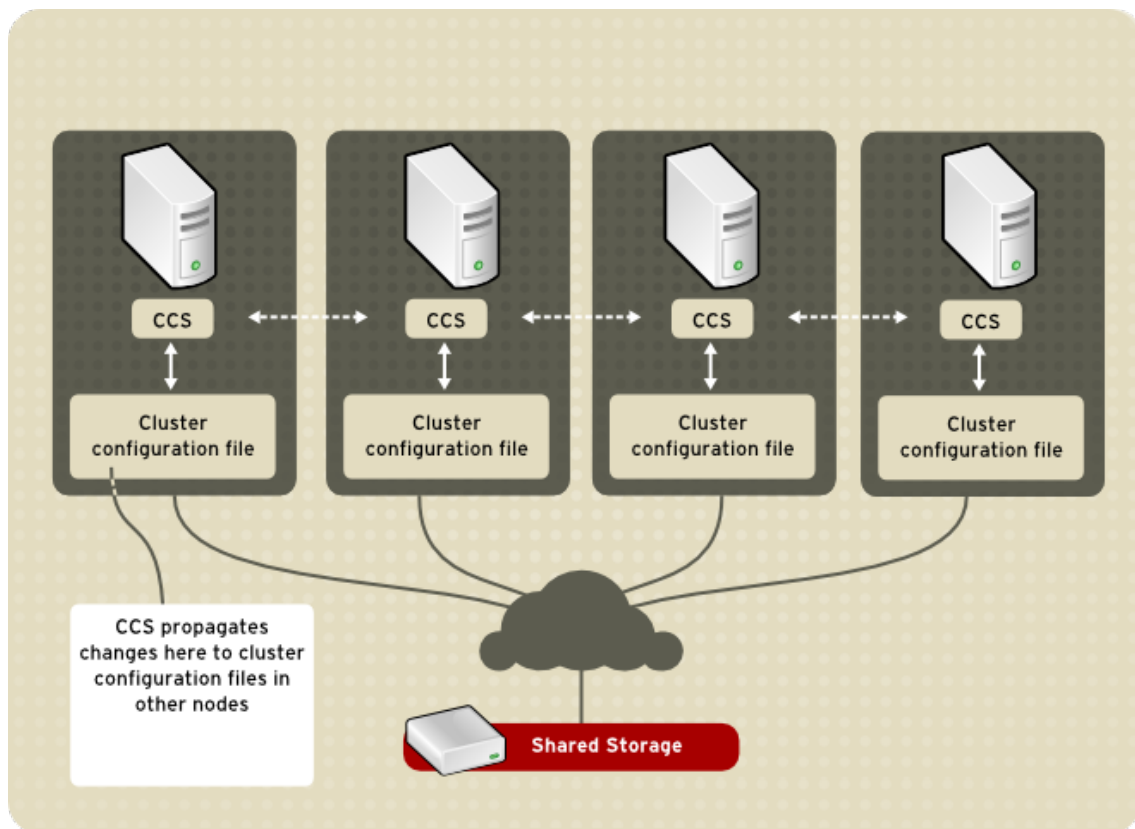


图 1.7. CCS Overview

Other cluster components (for example, CMAN) access configuration information from the configuration file through CCS (refer to [图 1.7 “CCS Overview”](#)).

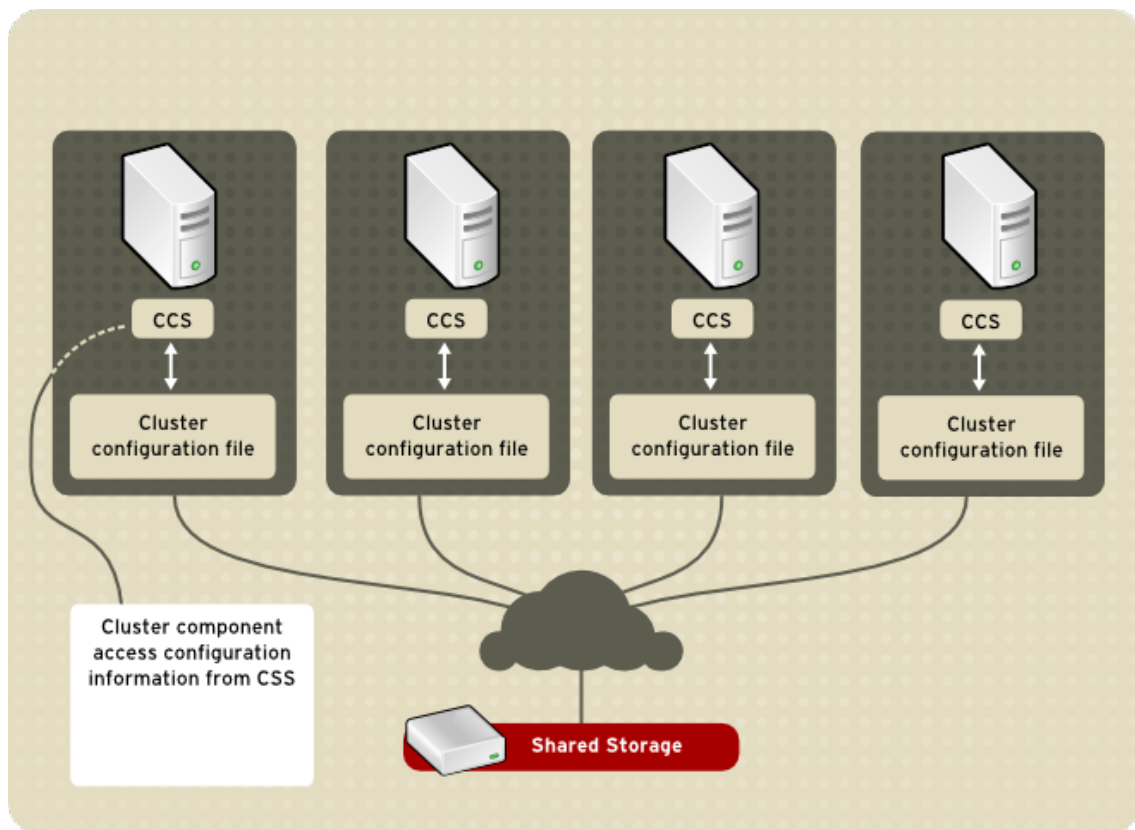


图 1.8. Accessing Configuration Information

群集配置文件 (`/etc/cluster/cluster.conf`) 是一个 XML 文件，它描述了下列群集属性：

- 群集名 — 显示群集的名字、群集配置文件版本、当节点入群集或被保护所使用的基本保护时间属性。
- 群集 — 显示群集的每个节点，指定节点名、节点 ID、quorum vote 的数量和其保护方法。
- 保护设备 — 显示群集里的保护设备。其参数根据保护设备的类型而有所不同。例如，对于用作保护设备的电源控制器，群集配置定义了电源控制器的名字、IP 地址、登录帐号和密码。
- 被管理的资源 — 显示创建群集服务所必需的资源。被管理的资源包括故障切换域的定义、资源（如 IP 地址）和服务。它也定义了群集服务和群集服务的故障切换行为。

1.4. 高可用性服务管理

高可用性服务管理提供在 Red Hat 群集中创建和管理高可用性**群集服务** (*cluster service*) 的能力。Red Hat 群集里的高可用性服务管理的关键组件是 **rgmanager**，它实现了 off-the-shelf 应用程序的冷失效切换 (cold failover)。在 Red Hat 群集里，应用程序和其他群集资源进行配置来组成高可用性的群集服务。高可用性群集服务可以从一个节点故障切换到另外一个节点，而不会对群集客户有明显的影晌。当某个群集节点发生故障或群集系统管理员将服务迁移（例如，需要对这个节点进行预定的维护时）到另外一个节点时，群集服务的故障切换都可以发生。

要创建高可用性服务，您必须在群集配置文件里进行配置。群集服务由群集**资源**组成。群集资源是您在群集配置文件里创建和管理的构建块 (building block) — 例如，IP 地址、应用程序初始化脚本或者 Red Hat GFS 共享分区。

You can associate a cluster service with a *failover domain*. A failover domain is a subset of cluster nodes that are eligible to run a particular cluster service (refer to [图 1.9 “故障切换域”](#)).



注意

故障切换域不是必需的。

群集服务在某个时间只能在一个节点中运行，这样可以维护数据的完整性。您可以在故障切换域里指定故障切换的优先级。指定故障切换的优先级是通过为故障切换域里的每个节点分配优先级别实现的。优先级决定故障切换的顺序 — 决定群集服务应该切换到哪个节点。如果您没有指定优先级，群集服务将可能切换至故障切换域里的任意节点。而且，您可以指定是否限制群集服务在其相关的故障切换域里的节点中运行。（如果与非限制性故障切换域相关联，在故障切换域里的成员都不可用的时候，群集服务可以在任意节点中启动。）

In [图 1.9 “故障切换域”](#), Failover Domain 1 is configured to restrict failover within that domain; therefore, Cluster Service X can only fail over between Node A and Node B. Failover Domain 2 is also configured to restrict failover with its domain; additionally, it is configured for failover priority. Failover Domain 2 priority is configured with Node C as priority 1, Node B as priority 2, and Node D as priority 3. If Node C fails, Cluster Service Y fails over to Node B next. If it cannot fail over to Node B, it tries failing over to Node D. Failover Domain 3 is configured with no priority and no restrictions. If the node that Cluster Service Z is running on fails, Cluster Service Z tries failing over to one of the nodes in Failover Domain 3. However, if none of those nodes is available, Cluster Service Z can fail over to any node in the cluster.

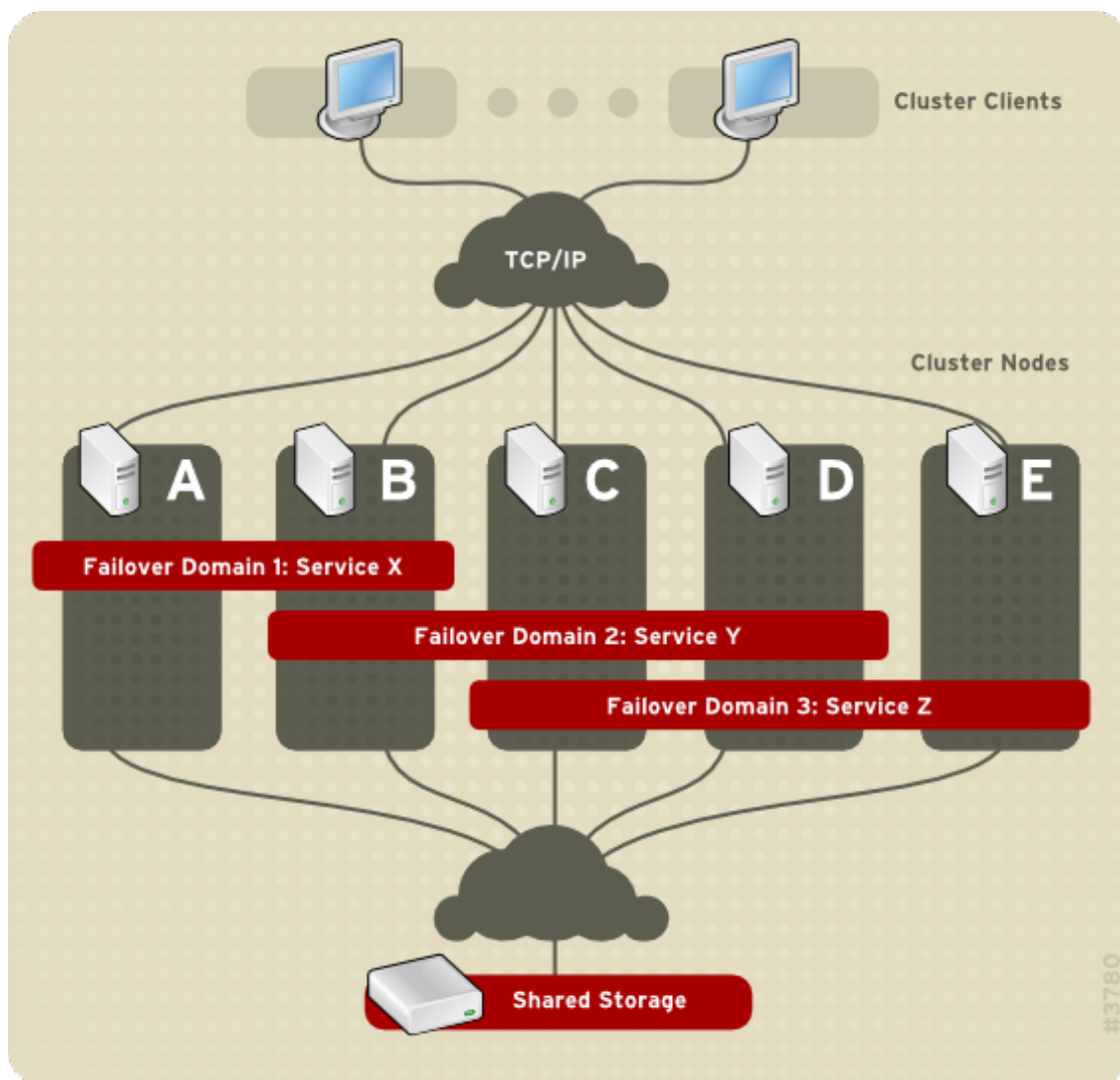


图 1.9. 故障切换域

图 1.10 “Web Server Cluster Service Example” shows an example of a high-availability cluster service that is a web server named “content-webserver”. It is running in cluster node B and is in a failover domain that consists of nodes A, B, and D. In addition, the failover domain is configured with a failover priority to fail over to node D before node A and to restrict failover to nodes only in that failover domain. The cluster service comprises these cluster resources:

- IP 地址资源 — 10.10.10.201。
- An application resource named “httpd-content” — a web server application init script `/etc/init.d/httpd` (specifying `httpd`).
- A file system resource — Red Hat GFS named “gfs-content-webserver”.

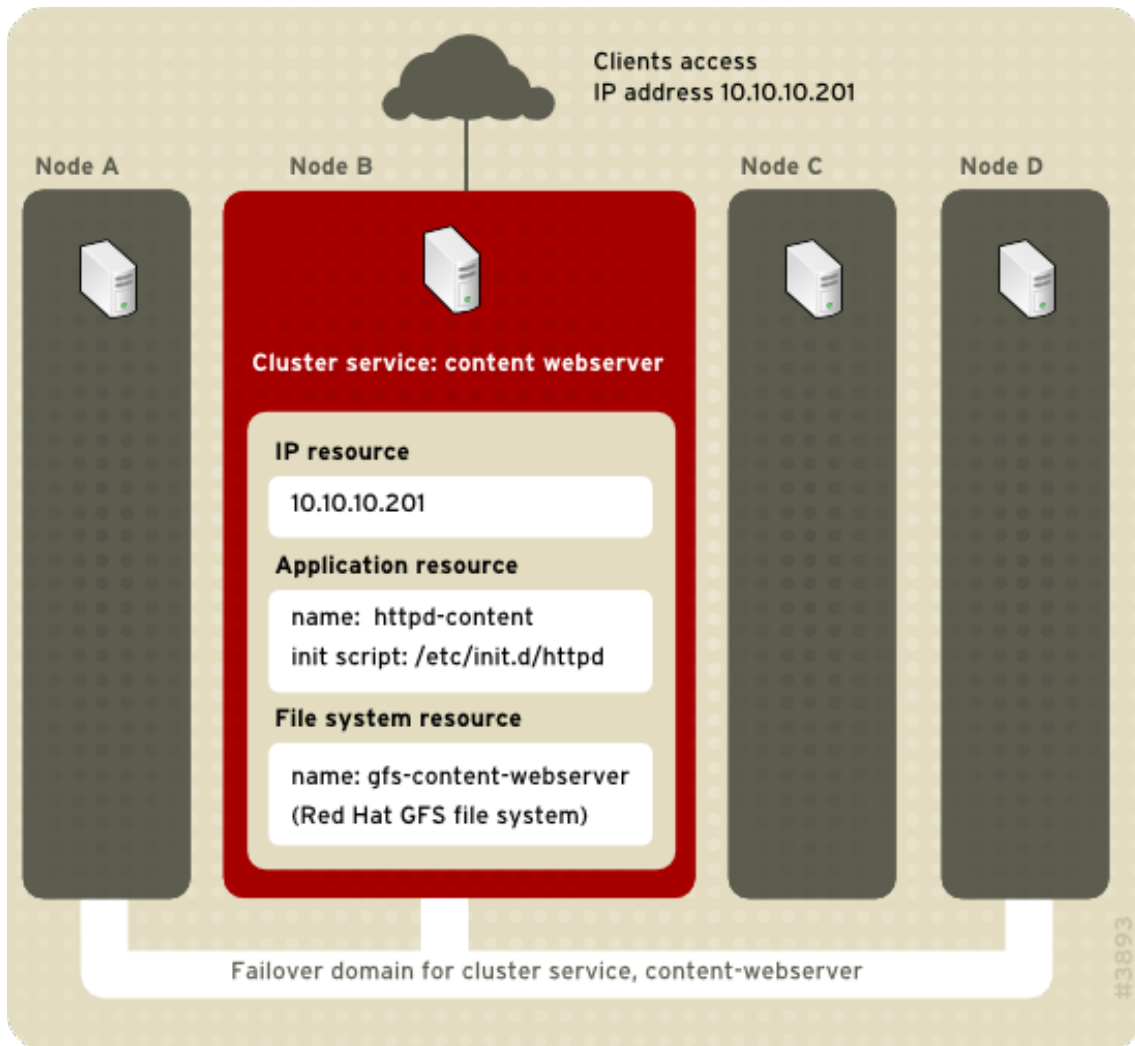


图 1.10. Web Server Cluster Service Example

客户通过 IP 地址 10.10.10.201 访问群集服务，和 web 服务器应用程序 httpd-content 进行交互。httpd-content 使用 gfs-content-webserver 文件系统。如果节点 B 发生故障，content-webserver 群集服务将切换到节点 D。如果节点 D 不可用或者也发生了故障，服务将切换至节点 A。故障切换将对群集客户不产生明显的影响。和切换前一样，群集服务将可以用相同的 IP 地址从另外一个群集节点进行访问。

1.5. RED HAT GFS

Red Hat GFS 是允许群集节点同时访问某个共享的块设备的群集文件系统。GFS 是一个原生的 (native) 文件系统，它直接和 Linux 内核文件系统接口的 VFS 层进行通信。GFS 使用分布式元数据和多个日志来优化群集操作。要维护文件系统的完整性，GFS 使用一个锁管理器来协调 I/O。当节点修改了 GFS 文件系统上的数据，这个改动马上为使用这个文件系统的其他群集节点可见。

使用 Red Hat GFS，您可以从如下几方面受益并实现应用程序的最大正常运行时间：

- 简化数据基础结构
 - 整个群集只需安装一次应用程序或补丁
 - 消除对应用程序数据的冗余拷贝（复制）的需要。
 - 启用多个客户对数据的并行读/写操作。
 - 简化备份和恢复（只需要备份或恢复一个文件系统）。

- 最大化对存储资源的使用；最小化存储的管理开销。
 - 将存储作为一个整体而不是根据分区管理。
 - 通过消除数据复制的需要从而减少总体的存储需求。
- 通过即时添加服务器或存储空间来无缝地扩展群集。
 - 不需要复杂的技术来对存储空间分区。
 - 通过挂载至普通文件系统来实时在群集里添加服务器。

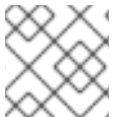
运行 Red Hat GFS 的节点都用 Red Hat Cluster Suite 配置和管理工具进行配置和管理。卷管理通过群集逻辑卷管理器 (Cluster Logical Volume Manager, CLVM) 进行管理。Red Hat GFS 在 Red Hat 群集的 GFS 节点间提供数据共享。GFS 提供 Red Hat 群集的 GFS 节点的单一的、一致的文件系统命名空间的视图。GFS 允许安装和运行应用程序而无需了解底层的存储基础结构。而且，GFS 提供通常在企业环境才要求的功能，如配额 (quota)、多重日志和多路径 (multipath) 支持。

GFS 根据存储环境的性能、可伸缩性和经济性需要来提供网络存储的通用方法。本章提供一些非常基础的、简略的信息来帮助您理解 GFS。

You can deploy GFS in a variety of configurations to suit your needs for performance, scalability, and economy. For superior performance and scalability, you can deploy GFS in a cluster that is connected directly to a SAN. For more economical needs, you can deploy GFS in a cluster that is connected to a LAN with servers that use *GNBD* (Global Network Block Device) or to *iSCSI* (Internet Small Computer System Interface) devices. (For more information about GNBD, refer to [第 1.7 节 “全球网络块设备”](#).)

下面的章节提供的例子，展示了怎样部署 GFS 来满足您对性能、灵活性和经济性的需要：

- [第 1.5.1 节 “出众的性能和灵活性”](#)
- [第 1.5.2 节 “性能、灵活性、合适的价格”](#)
- [第 1.5.3 节 “经济性和性能”](#)



注意

GFS 部署示例只是演示了基本的配置；在实践中，您可能需要将这些配置进行组合。

1.5.1. 出众的性能和灵活性

You can obtain the highest shared-file performance when applications access storage directly. The GFS SAN configuration in [图 1.11 “GFS with a SAN”](#) provides superior file performance for shared files and file systems. Linux applications run directly on cluster nodes using GFS. Without file protocols or storage servers to slow data access, performance is similar to individual Linux servers with directly connected storage; yet, each GFS application node has equal access to all data files. GFS supports over 300 GFS nodes.

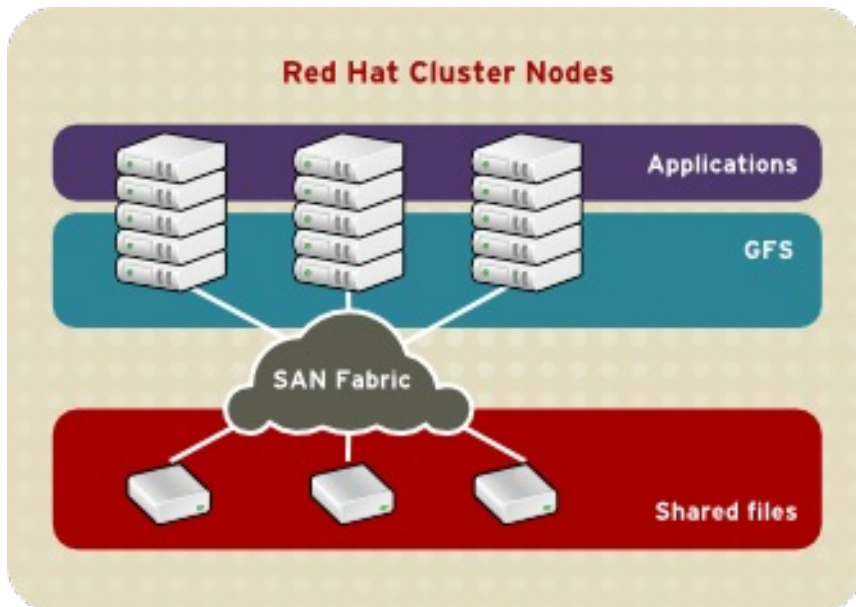


图 1.11. GFS with a SAN

1.5.2. 性能、灵活性、合适的价格

Multiple Linux client applications on a LAN can share the same SAN-based data as shown in [图 1.12](#) “GFS and GNBD with a SAN”. SAN block storage is presented to network clients as block storage devices by GNBD servers. From the perspective of a client application, storage is accessed as if it were directly attached to the server in which the application is running. Stored data is actually on the SAN. Storage devices and data can be equally shared by network client applications. File locking and sharing functions are handled by GFS for each network client.

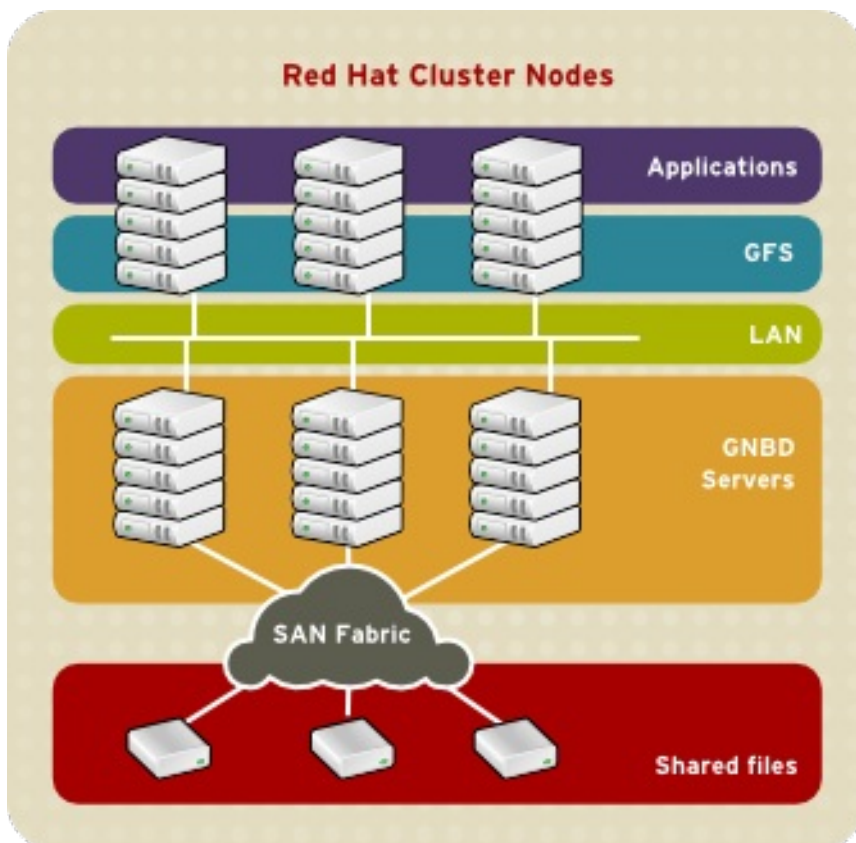


图 1.12. GFS and GNBD with a SAN

1.5.3. 经济性和性能

图 1.13 “使用直连存储设备的 GFS 和 GNBD” shows how Linux client applications can take advantage of an existing Ethernet topology to gain shared access to all block storage devices. Client data files and file systems can be shared with GFS on each client. Application failover can be fully automated with Red Hat Cluster Suite.

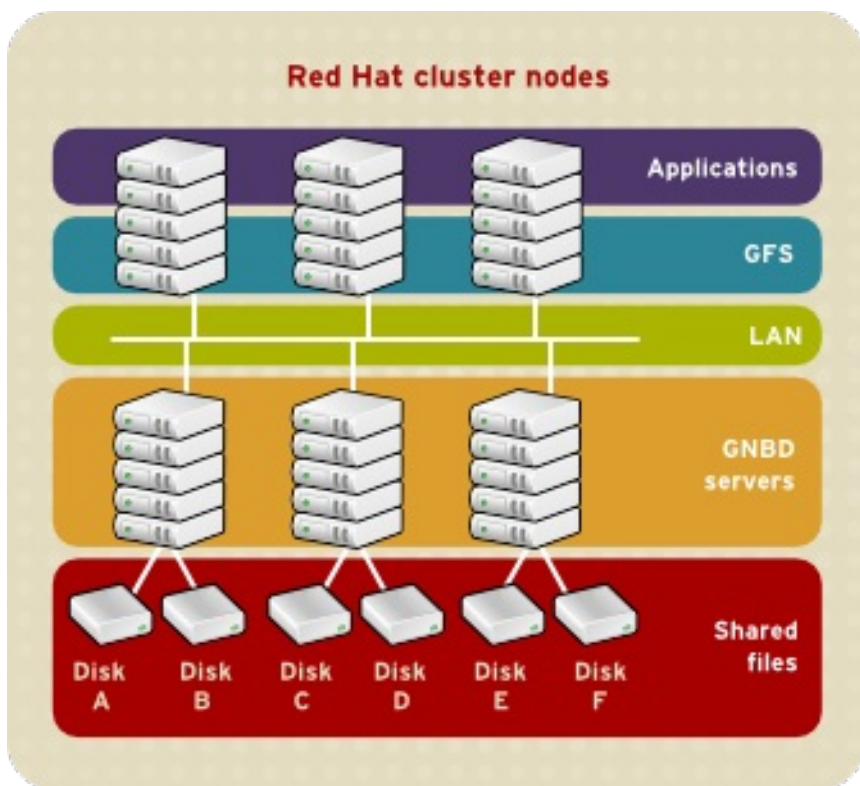


图 1.13. 使用直连存储设备的 GFS 和 GNBD

1.6. 群集逻辑卷管理器

群集逻辑卷管理器（Cluster Logical Volume Manager, CLVM）提供了 LVM2 的群集版本。CLVM 具备和单节点上的 LVM2 相同的能力，但它可以使逻辑卷为 Red Hat 群集里的所有节点所用。用 CLVM 创建的逻辑卷可为群集里的所有节点所用。

The key component in CLVM is **clvmd**. **clvmd** is a daemon that provides clustering extensions to the standard LVM2 tool set and allows LVM2 commands to manage shared storage. **clvmd** runs in each cluster node and distributes LVM metadata updates in a cluster, thereby presenting each cluster node with the same view of the logical volumes (refer to 图 1.14 “CLVM Overview”). Logical volumes created with CLVM on shared storage are visible to all nodes that have access to the shared storage. CLVM allows a user to configure logical volumes on shared storage by locking access to physical storage while a logical volume is being configured. CLVM uses the lock-management service provided by the cluster infrastructure (refer to 第 1.3 节 “Cluster Infrastructure”).



注意

在红帽群集套件中使用的共享存储要求您运行群集逻辑卷管理器守护进程（**clvmd**）或者高可用性逻辑卷管理代理（HA-LVM）。如果您无法使用 **clvmd** 守护进程或者 HA-LVM，无论是因为操作原因还是您没有正确的权利，您就一定不能在共享磁盘中使用单一事件 LVM，因为这将导致数据崩溃。如果您有任何疑问，请联络您的红帽服务代表。



注意

使用 CLVM 需要对 `/etc/lvm/lvm.conf` 进行群集范围的锁定方面的少许修改。

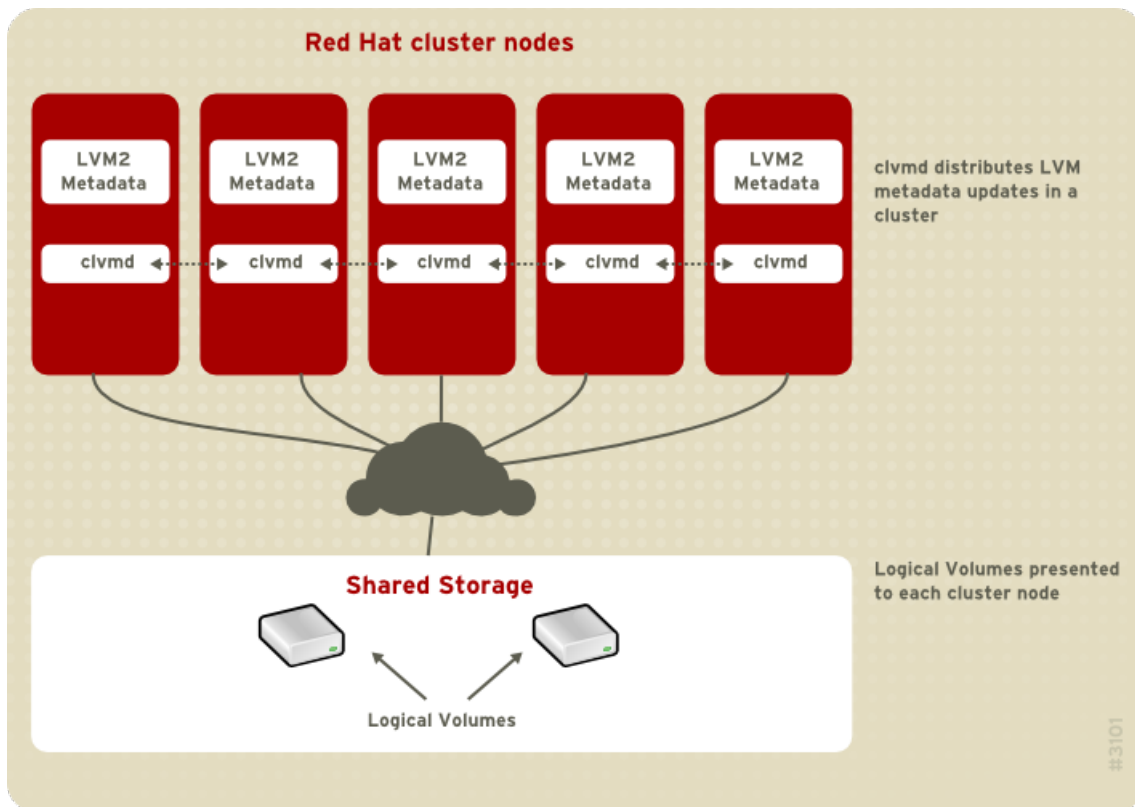


图 1.14. CLVM Overview

You can configure CLVM using the same commands as LVM2, using the LVM graphical user interface (refer to [图 1.15 “LVM Graphical User Interface”](#)), or using the storage configuration function of the **Conga** cluster configuration graphical user interface (refer to [图 1.16 “Conga LVM Graphical User Interface”](#)). [图 1.17 “Creating Logical Volumes”](#) shows the basic concept of creating logical volumes from Linux partitions and shows the commands used to create logical volumes.

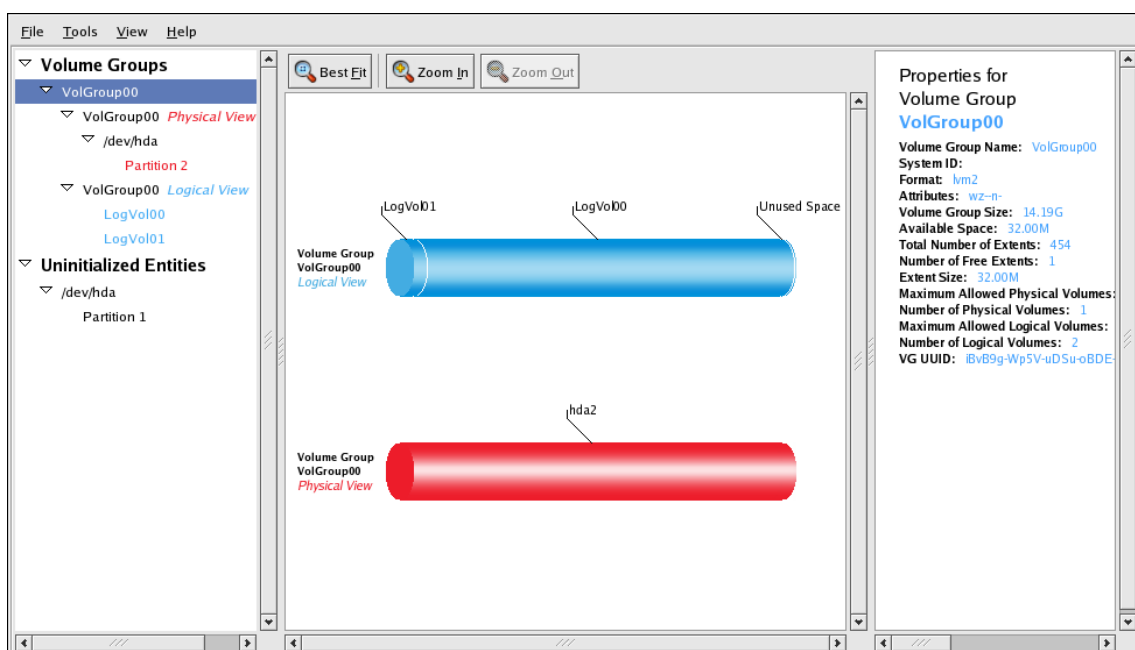


图 1.15. LVM Graphical User Interface

The screenshot displays the Conga LVM GUI. At the top, there is a navigation bar with 'homebase', 'cluster', and 'storage' tabs. The 'storage' tab is active, and the URL 'tng3-1.lab.msp.redhat.com' is shown. On the left, a sidebar menu includes 'System List', 'tng3-1.lab.msp.redhat.com', 'Hard Drives', 'Partition Tables', 'Volume Groups', 'New Volume Group', and 'myvg'. The main area shows 'Volume Group myvg' with a 'Graphical View' checkbox checked. Below this, there are two cylinders: a blue one for 'Logical Volumes' and a red one for 'Physical Volumes'. A note says 'Click cylinders to view properties, unselect all to view Volume Group's properties'. Below the cylinders is a detailed configuration window for 'Volume Group 'myvg'' with the following data:

Property	Value
Volume Group Name	myvg
Extent Size	4.0 MB
Total Extents	13170
Free Extents	0
Size	51.44 GB
Used Extents	13170
Maximum Physical Volumes	256
Maximum Logical Volumes	256
Attributes	wz--n-
Clustered	false
UUID	yEdO77-SF1T-uQHU-ENpR-Zow4-Dd3m-2Q2HZu

At the bottom of the configuration window are buttons for 'Remove', 'Add Physical Volumes', 'Reset', and 'Apply'. At the very bottom of the page, a copyright notice reads: 'The Conga Cluster and Storage Management System is Copyright © 2000-2007 by Red Hat, Inc. Distributed under the GNU GPL license.'

图 1.16. Conga LVM Graphical User Interface

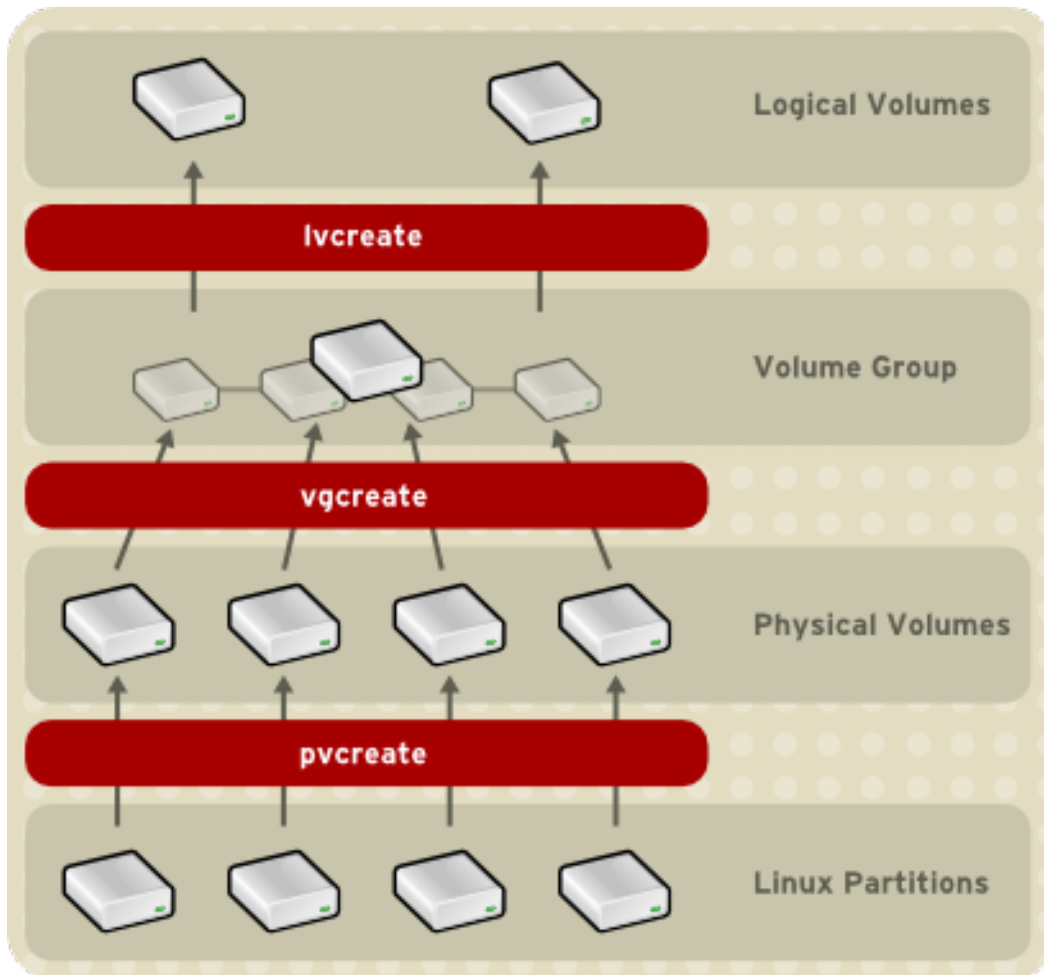


图 1.17. Creating Logical Volumes

1.7. 全球网络块设备

全球网络块设备（Global Network Block Device, GNBD）提供通过 TCP/IP 访问 Red Hat GFS 块设备的功能。GNBD 从概念上和 NBD 相似；然而，GNBD 是和 GFS 相关的且为 GFS 单独进行了调整。当需要更健壮的技术时，GNBD 是有用处的 — 光纤通道或 single-initiator SCSI — 不是必需的或成本过高。

GNBD consists of two major components: a GNBD client and a GNBD server. A GNBD client runs in a node with GFS and imports a block device exported by a GNBD server. A GNBD server runs in another node and exports block-level storage from its local storage (either directly attached storage or SAN storage). Refer to [图 1.18 “GNBD 概述”](#). Multiple GNBD clients can access a device exported by a GNBD server, thus making a GNBD suitable for use by a group of nodes running GFS.

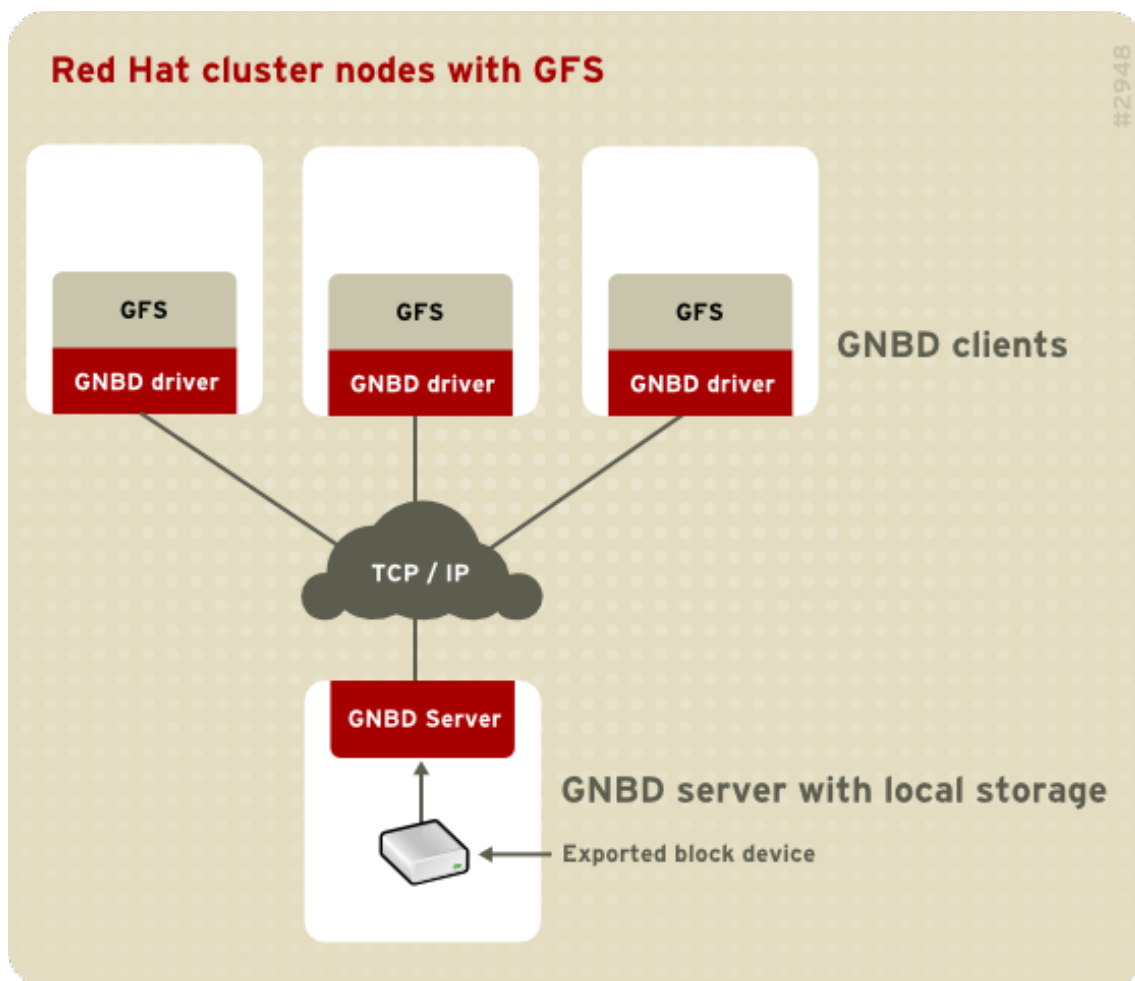


图 1.18. GNBD 概述

1.8. LINUX 虚拟服务器

Linux 虚拟服务器 (LVS) 是一组集成的软件组件，它用于在真实服务器之间平衡 IP 负载。LVS 运行在一对配置相同的机器上：一个作为活动的 LVS 路由器而另外一个作为备份 LVS 路由器。活动的 LVS 路由器扮演两个角色：

- 在真实服务器之间平衡负载。
- 检查每个真实服务器上的服务的完整性。

备份 LVS 路由器监控活动的 LVS 路由器并在它发生故障时进行接管。

图 1.19 “Components of a Running LVS Cluster” provides an overview of the LVS components and their interrelationship.

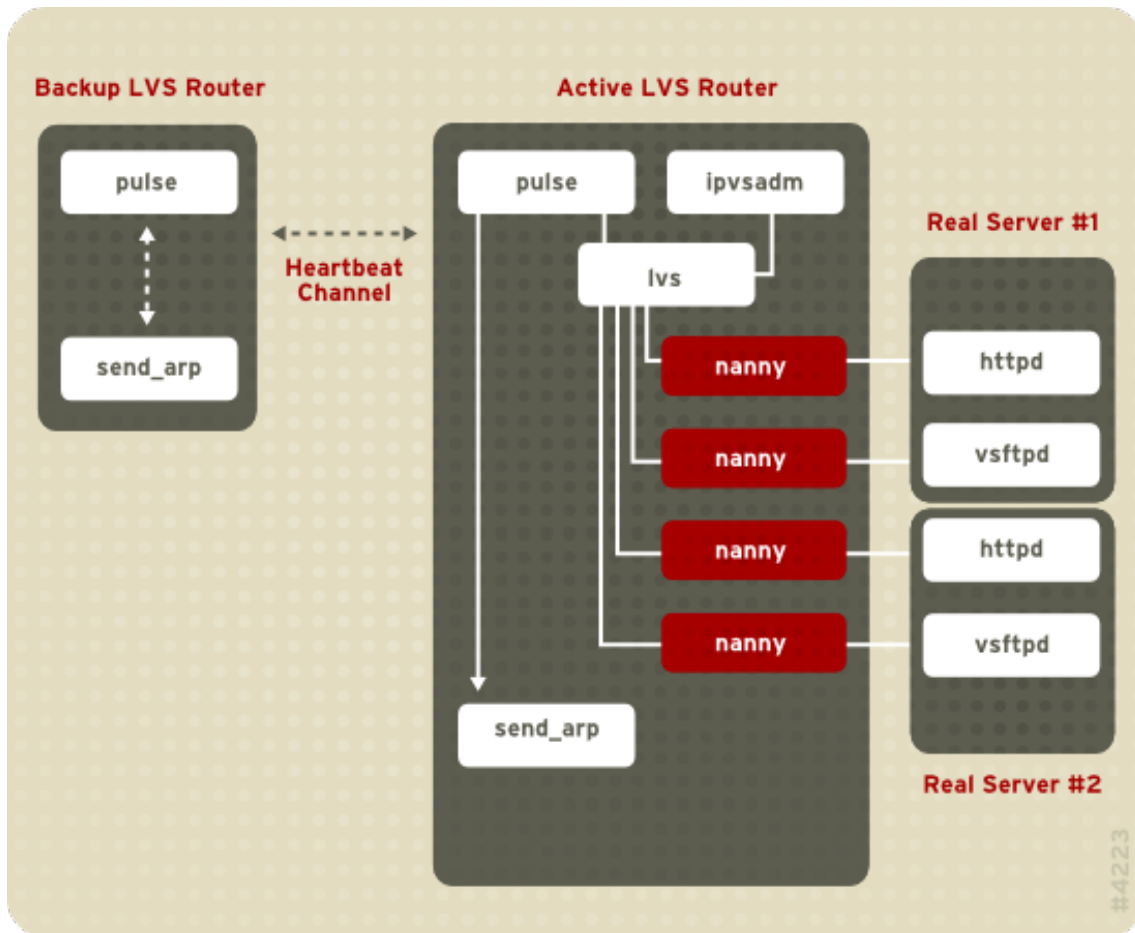


图 1.19. Components of a Running LVS Cluster

`pulse` 守护进程运行在活动的以及备份 LVS 路由器上。在备份 LVS 路由器上，`pulse` 发送 *heartbeat* 到活动路由器的公用接口，确保活动路由器正常工作。在活动的 LVS 路由器上，`pulse` 启动 `lvs` 守护进程并响应备份 LVS 路由器的 *heartbeat* 查询。

一旦启动，`lvs` 守护进程将调用 `ipvsadm` 工具来配置和维护内核里的 IPVS（IP 虚拟服务器）路由表，并为每台真实服务器上配置的每个虚拟服务器启动一个 `nanny` 进程。每个 `nanny` 进程检查一台真实服务器上的一个配置好的服务的状态，并告诉 `lvs` 守护进程这个服务是否发生了故障。如果检测到了故障，`lvs` 守护进程将指引 `ipvsadm` 从 IPVS 路由表中删除那台真实服务器的条目。

如果备份 LVS 路由器没有从活动的 LVS 路由器收到响应，它将初始化故障切换，调用 `send_arp` 将所有虚拟 IP 地址重新分配给备份 LVS 路由器的 NIC 硬件地址（MAC 地址），并通过公用以及专用网络接口向活动的 LVS 路由器发送命令来关闭其 `lvs` 守护进程，然后在备份 LVS 路由器上启动 `lvs` 守护进程来接受虚拟服务器的请求。

对于访问宿主服务（如网站或数据库应用程序）的外部用户，LVS 就象是一个服务器。然而，用户实际上访问了 LVS 路由器后面的多个真实服务器。

因为在 LVS 里没有内置组件在真实服务器间共享数据，您有两个基本选择：

- 在真实服务器间对数据进行同步。
- 在拓扑结构中添加用于共享数据访问的第三层。

对于不允许大量用户在真实服务器上上载或修改数据的服务器来说，第一个选择是首选项。而如果真实服务器允许大量用户修改数据，如电子商务网站，添加第三层则更为可取。

有许多方法可以在真实服务器间进行数据同步。例如，您可以使用 shell 脚本同时将更新的 web 页面登记

到真实服务器。或者，您可以使用 `rsync` 等程序以设定的时间间隔在所有节点间复制改动的数据。然而，在用户频繁地上载文件或执行数据库事务的环境里，使用脚本或 `rsync` 命令来进行同步并不是最理想的方法。因此，对于执行大量上载、数据库事务或具有类似流量的真实服务器，*三层的拓扑结构 (three-tiered topology)* 更适合于数据同步。

1.8.1. Two-Tier LVS Topology

图 1.20 “Two-Tier LVS Topology” shows a simple LVS configuration consisting of two tiers: LVS routers and real servers. The LVS-router tier consists of one active LVS router and one backup LVS router. The real-server tier consists of real servers connected to the private network. Each LVS router has two network interfaces: one connected to a public network (Internet) and one connected to a private network. A network interface connected to each network allows the LVS routers to regulate traffic between clients on the public network and the real servers on the private network. In 图 1.20 “Two-Tier LVS Topology”, the active LVS router uses *Network Address Translation (NAT)* to direct traffic from the public network to real servers on the private network, which in turn provide services as requested. The real servers pass all public traffic through the active LVS router. From the perspective of clients on the public network, the LVS router appears as one entity.

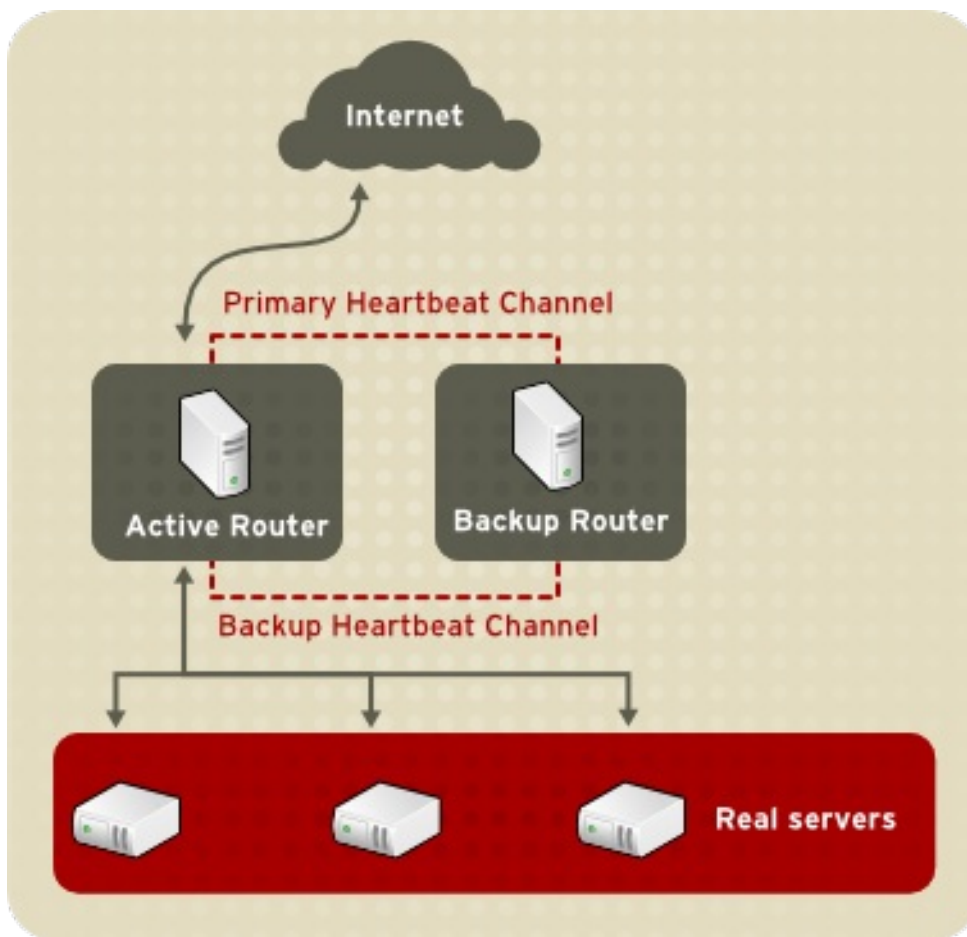


图 1.20. Two-Tier LVS Topology

到达 LVS 路由器的服务请求被指引到 *虚拟 IP* 地址或 *VIP*。这是一个公用的地址，站点管理员将它和全限定域名相关联，如 `www.example.com`，并分配给一个或多个 *虚拟服务器*^[1]。注意，进行故障切换时，*VIP* 地址将从一个 LVS 路由器移植到另外一个 LVS 路由器，并保持这个 IP 地址，这被称为 *浮动 IP 地址 (floating IP address)*。

VIP 地址可以以连接 LVS 路由器到公用网络的接口为别名。例如，如果 `eth0` 连接至 Internet，那么多个虚拟服务器可以使用别名 `eth0:1`。或者，每个虚拟服务器可以按照服务和单独的设备相关联。例如，HTTP 通信在 `eth0:1` 上处理，而 FTP 通信在 `eth0:2` 上进行处理。

在某一时刻只能有一个 LVS 路由器处于活动状态。活动的 LVS 路由器的角色是将服务请求从虚拟 IP 地址重定向到实际的服务器。这种重定向是基于下面 8 个负载均衡算法中的一个：

- 轮询调度 (Round-Robin Scheduling) — 将每个请求按顺序地在一个服务器池里进行分发。使用这个算法，所有的真实服务器都被一视同仁，而不考虑其能力或负载。
- 加权轮询调度 (Weighted Round-Robin Scheduling) — 将每个请求按顺序地在一个服务器池里进行分发，但赋予能力更强的服务器更多的负载。能力用用户分配的权重来表示，然后根据负载信息动态地进行上下调整。如果服务器池里的服务器的能力区别很大，这是首选的办法。然而，如果请求负载突然发生变化，权重更大的服务器可能会处理超过其份额的请求。
- 最小连接调度 (Least-Connection Scheduling) — 给活动连接较少的服务器分配更多的请求。这是一种动态调度算法，如果请求负载变化较快，这是一个更好的选择。它最适合于每个节点都具备基本相同的能力的服务器池。如果服务器的能力不一样，权重最小连接调度 (weighted least-connection scheduling) 就是更好的选择。
- 权重最小连接调度 (Weighted Least-Connections) — 这是缺省的算法。它将更多的请求分配给相对其能力具有较少连接的服务器。能力以用户分配的权重来表示，并根据动态负载信息上下调整。当服务器池包含具有不同能力的硬件时，附加的权重使其成为理想的算法。
- 基于局部性的最少连接调度 (Locality-Based Least-Connection Scheduling) — 将更多的请求分配给相对其目的 IP 地址具有较少连接的服务器。这个算法用于代理-缓存 (proxy-cache) 的服务器群集。它根据目标 IP 地址将数据报文指引到这个地址的服务器，除非该服务器超载并有一服务器处于一半的工作负荷，此时它将分配给负荷最小的服务器。
- 带复制的基于局部性最小连接调度 (Locality-Based Least-Connection Scheduling with Replication Scheduling) — 将更多的请求分配给相对其目的 IP 地址具有较少连接的服务器。这个算法也用于代理-缓存 (proxy-cache) 的服务器群集。它和基于局部性最小连接调度的区别是：它将目标 IP 地址映射到服务器节点的一个子集。请求将被发送到这个子集里具有最低连接数的服务器上。如果所有目的 IP 地址的节点都超载，它将整个服务器池里的具有最少连接数的服务器添加到用于该目的 IP 地址的服务器子集里，这样就复制了用于该地址的一个新的服务器。然后负载最重的节点将从服务器子集里删除来防止过度复制。
- 源地址散列调度 (Source Hash Scheduling) — 通过在静态散列表里查找源 IP 地址来将请求发送到服务器池。这个算法用于带有多重防火墙的 LVS 路由器。

而且，活动的 LVS 路由器通过简单的 *send/expect scripts* 动态地监控服务器上的特定服务的健康状况。为了帮助检测需要动态数据的服务（如 HTTPS 或 SSL）的健康状况，您也可以调用外部的执行程序。如果服务器上的服务发生了故障，活动的 LVS 路由器将停止发送任务到这个服务器，直到它恢复正常操作。

备份 LVS 路由器扮演备用系统的角色。LVS 路由器通过主要的外部公用接口定期地交换 heartbeat 信息，在故障切换的情况下则通过专用接口。如果备份 LVS 路由器在预期的时间间隔里没有接收到 heartbeat 信息，它将初始化故障切换并开始执行活动 LVS 路由器的任务。在故障切换过程中，备份 LVS 路由器用 *ARP spoofing* 技术接管失效路由器的 VIP 地址 — 备份 LVS 路由器将自己作为发送到失效节点的 IP 数据包的目的地址。当失效节点恢复正常服务时，备份 LVS 路由器将再次行使其备份功能。

The simple, two-tier configuration in [图 1.20 “Two-Tier LVS Topology”](#) is suited best for clusters serving data that does not change very frequently — such as static web pages — because the individual real servers do not automatically synchronize data among themselves.

1.8.2. Three-Tier LVS Topology

[图 1.21 “Three-Tier LVS Topology”](#) shows a typical three-tier LVS configuration. In the example, the active LVS router routes the requests from the public network (Internet) to the second tier — real servers. Each real server then accesses a shared data source of a Red Hat cluster in the third tier over the private network.

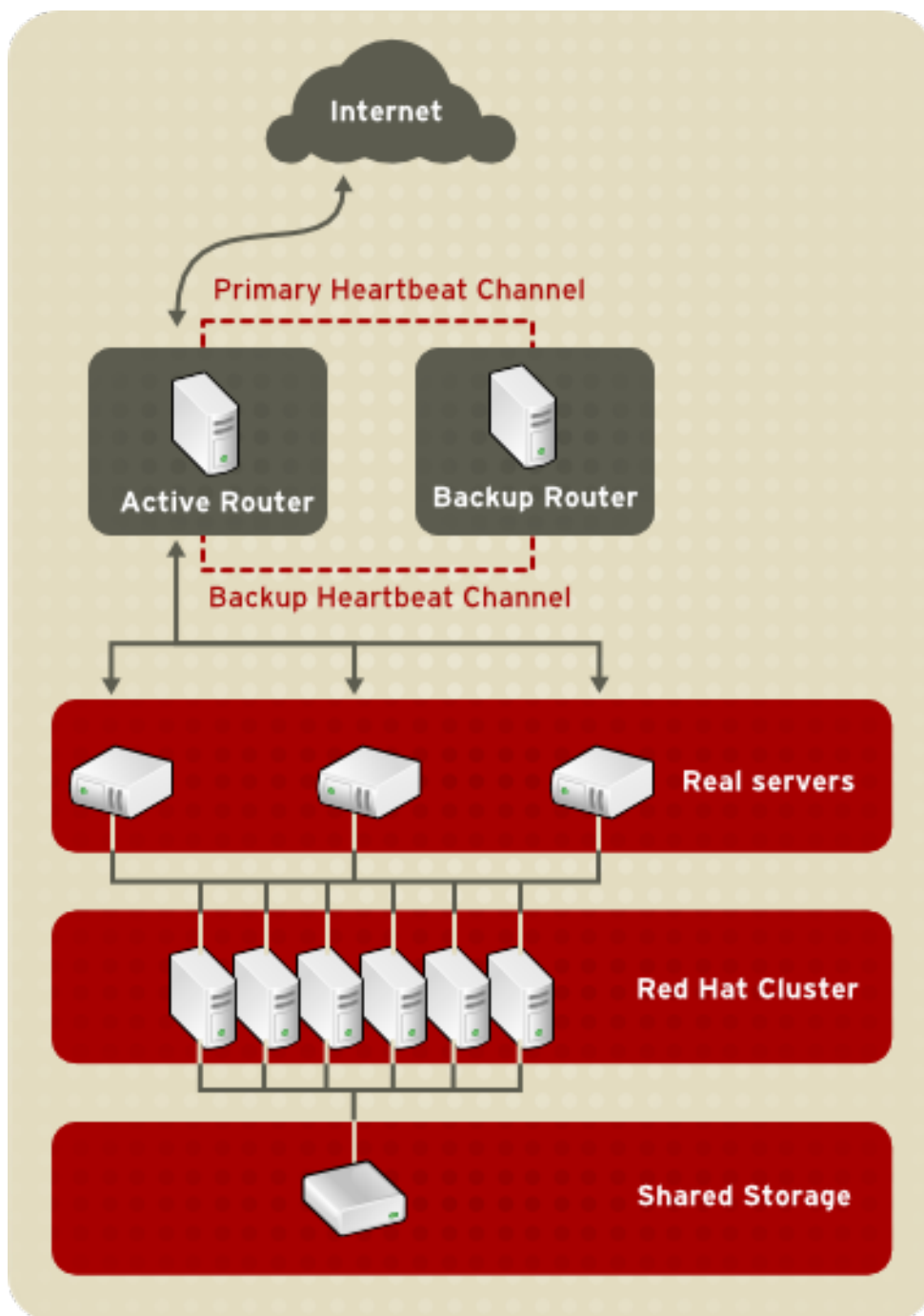


图 1.21. Three-Tier LVS Topology

这种拓扑结构适合于繁忙的 FTP 服务器，其可访问的数据存储在中央的、高可用的服务器上，而每个真实服务器都通过导出的 NFS 目录或 Samba 共享访问该服务器。我们也推荐在访问用于事务的中央的、高可用性的数据库的网站上应用这种拓扑结构。此外，在 Red Hat 群集中使用 active-active 配置，您可以配置高可用性的群集来同时行使这个两种功能。

1.8.3. 路由方法

您可以在 LVS 里使用网络地址转换（Network Address Translation, NAT）或直接路由。下面的内容简单地描述了在 LVS 里使用 NAT 路由和直接路由。

1.8.3.1. NAT 路由

图 1.22 “LVS Implemented with NAT Routing”，illustrates LVS using NAT routing to move requests between the Internet and a private network.

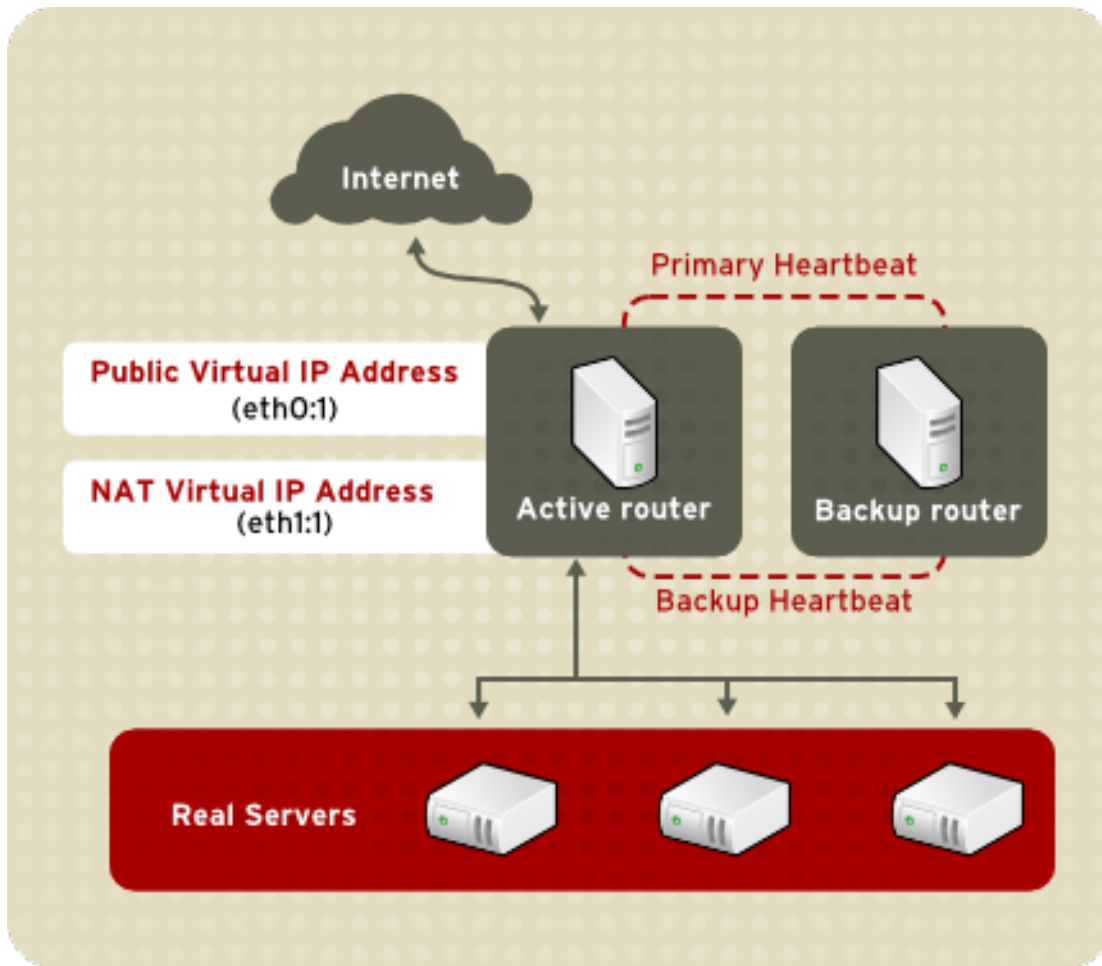


图 1.22. LVS Implemented with NAT Routing

在这个例子中，活跃 LVS 路由器装有两块网络接口卡（NIC）。Internet 使用的 NIC 在 eth0 上有一个真实的 IP 地址，它也有一个别名为 eth0:1 的浮动 IP 地址。专用网络接口的 NIC 在 eth1 上有一个真实的 IP 地址以及别名为 eth1:1 的浮动 IP 地址。在故障切换发生时，面向 Internet 的虚拟接口以及专用虚拟接口由备份 LVS 路由器同时接管。专用网络中的所有真实服务器都使用用于 NAT 路由器的浮动 IP 地址来作为与活跃 LVS 路由器通信的默认路由，这样它们响应来自 Internet 的请求的能力就不会被削弱。

In the example, the LVS router's public LVS floating IP address and private NAT floating IP address are aliased to two physical NICs. While it is possible to associate each floating IP address to its physical device on the LVS router nodes, having more than two NICs is not a requirement.

使用这种拓扑结构，活动的 LVS 路由器接收请求并将其发送到合适的服务器上。让后真实服务器处理这个请求并将数据包返回给 LVS 路由器。LVS 路由器使网络地址转换来代替带有 LVS 路由器挂公用 VIP 地址的数据包里的真实服务器。因为对客户隐藏了事实服务器的真正的 IP 地址，所以这个过程被称作 IP 伪装 (masquerading)。

使用 NAT 路由，真实服务器可以是运行不同操作系统的任何类型的机器。NAT 的主要缺点是 LVS 路由器可能成为大型部署中的瓶颈，因为它既得处理转入又得处理转出的请求。

1.8.3.2. 直接路由

和 NAT 相比，直接路由可以提供增强的性能。直接路由允许真实服务器处理和直接发送数据包给请求用户而不是通过 LVS 路由器传送转出数据包。让 LVS 路由器只处理转入的数据包，直接路由降低了网络性能出现问题的可能性。

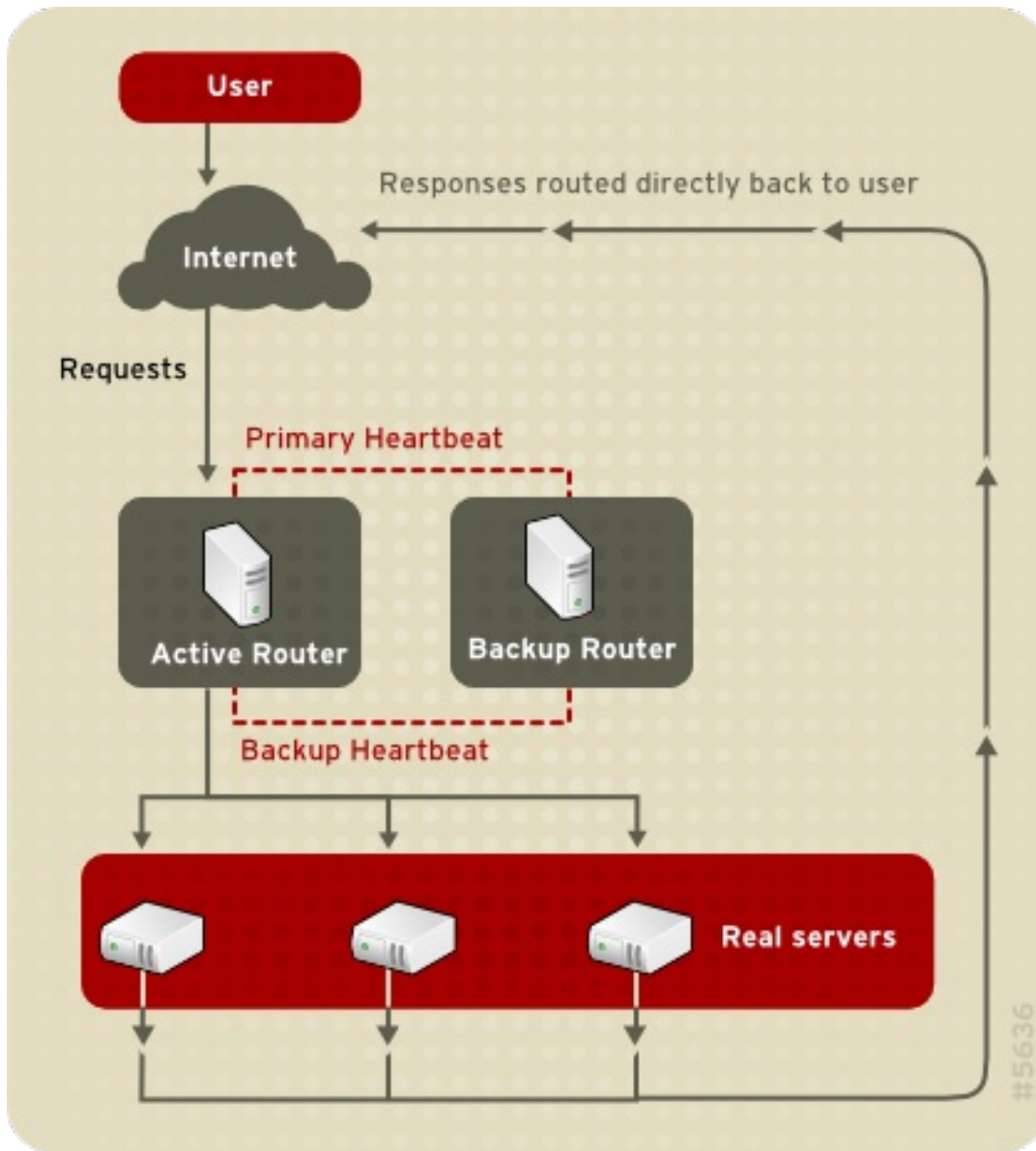


图 1.23. LVS Implemented with Direct Routing

在典型的直接路由 LVS 配置里，LVS 路由器通过虚拟 IP（VIP）接收转入的服务器请求并使用一个调度算法来将请求发送到真实服务器上。每个真实服务器都处理请求，并将响应直接发送给客户而不通过 LVS 路由器。直接路由提供了可伸缩性，因为可以添加真实服务器而不增加 LVS 路由器将转出数据包从真实服务器发送给客户的负担，而这在网络负载很重时可能是瓶颈。

虽然在 LVS 里使用直接路由有很多优点，但它也有缺陷。直接路由最常见的问题是和 *地址解析协议*（*Address Resolution Protocol*）（ARP）相关的。

In typical situations, a client on the Internet sends a request to an IP address. Network routers typically send requests to their destination by relating IP addresses to a machine's MAC address with ARP. ARP requests are broadcast to all connected machines on a network, and the machine with the correct IP/MAC address combination receives the packet. The IP/MAC associations are stored in an ARP cache, which is cleared periodically (usually every 15 minutes) and refilled with IP/MAC associations.

使用直接路由的 LVS 配置里和 ARP 请求相关的问题是：因为到一个 IP 地址的客户请求必须和一个 MAC 地址相关联才能被处理，LVS 路由器的虚拟 IP 地址也必须关联到一个 MAC。然而，因为 LVS 路由器和真实服务器具有相同的 VIP，ARP 请求被广播至和这个 VIP 相关联的所有节点。这会导致几个问题，如直接和某个真实服务器相关联或直接处理请求的 VIP，将完全绕过 LVS 路由器并使 LVS 配置不起作用。

即使是带有强大的 CPU、可以快速响应客户请求的 LVS 路由器也未必能解决这个问题。如果 LVS 路由器负载很重，它对 ARP 的响应可能会比较空闲的真实服务器更慢，后者分配了请求客户的 ARP 缓存里的 VIP 且响应更快。

要解决这个问题，转入的请求应该只和 LVS 路由器的 VIP 相关联，这样就会正确地处理请求并将它们发送到真实服务器池。这可以用 **iptables** 数据包过滤工具来实现。

1.8.4. 持久性和防火墙标记

在某些情况下，对于客户来说，重新连接系统的真实服务器、而不是用 LVS 负载均衡算法将请求发送到最合适的服务器是可取的。这样的例子包括多屏幕的 web 表单、cookies、SSL 和 FTP 连接。在这些情况下，除非保留了上下文的相同服务器来处理事务用，否则客户端可能不能正确工作。LVS 提供两个不同的功能来处理这种情况：*持久性 (persistence)* 和 *防火墙标记 (firewall mark)*。

1.8.4.1. Persistence

启用后，持久性起到定时器的作用。当客户连接一个服务时，LVS 会在指定的时间内记住最后的连接。如果具有相同 IP 地址的客户在这段时间内再次进行连接，它将被送往和上次连接相同的服务器 — 忽略负载均衡机制。而当连接在这段时间外发生，它会按照适当的调度规则进行处理。

持久性也允许您指定适用于客户 IP 地址的子网掩码，它将连接按子网分组，控制哪些地址具有更高的持久性级别。

对于使用多个端口通信的协议（如 FTP）来说，根据端口进行连接分组是很重要的。然而，持久性不是处理根据端口进行连接分组的效率最高的办法。对于这样的情况，您最好使用 *防火墙标记 (firewall mark)*。

1.8.4.2. 防火墙标记

根据协议或相关协议组使用不同的端口，防火墙标记是容易且高效率的方法。例如，如果 LVS 部署在一个电子商务网站里，防火墙标记可以将 HTTP 连接绑定在端口 80、安全 HTTP 连接绑定在端口 443。通过根据每个协议为虚拟服务器分配相同的防火墙标记，事务的状态信息可以保留，因为 LVS 路由器在连接打开后将所有的请求转发到相同的真实服务器。

因为它效率高且易于使用，对于连接分组，LVS 管理员应该尽可能用防火墙标记代替持久性。然而，您应该仍然将持久性和防火墙标记一起加入到虚拟服务器里，这样才能确保客户在足够长的时间内重新连接到相同的服务器。

1.9. 群集管理工具

Red Hat Cluster Suite 提供不同的工具来配置和管理您的 Red Hat 群集。本节是对 Red Hat Cluster Suite 里可用的管理工具的概述：

- [第 1.9.1 节 “Conga”](#)
- [第 1.9.2 节 “群集管理图形界面”](#)
- [第 1.9.3 节 “命令行管理工具”](#)

1.9.1. Conga

Conga 是为 Red Hat 群集和存储提供集中配置和管理的软件组件集合。**Conga** 具有下面的主要特征：

- 管理群集和存储的 web 界面

- 自动部署群集数据和支持软件包
- 易于集成现有群集
- 不需要重验证
- 集成群集状态和日志
- 用户权限的细颗粒度控制

Conga 里的主要组件是 **luci** 和 **ricci**，它们都是可单独安装的。**luci** 是一个服务器组件，它运行在一台机器上并通过 **ricci** 与多个群集和机器通信。**ricci** 是运行在 **Conga** 管理的每台机器（群集成员或独立的机器）上的代理。

luci 可以通过 Web 浏览器访问，它提供 3 个可以从下面的标签页访问的主要功能：

- **homebase** — 提供添加和删除机器、添加和删除用户和配置用户权限的工具。只有系统管理员才被允许访问此标签页。
- **cluster** — 提供创建和配置群集的工具。**luci** 的每个实例都列出用这个 **luci** 设立的群集。系统管理员可以管理此标签页上列出的所有群集。其他用户只能管理其具有管理权限（由管理员赋予）的群集。
- **storage** — 提供远程管理存储设备的工具。您可以在属于或不属于群集的机器上使用这个标签页上的工具。

要管理群集或存储设备，管理员要将群集或机器添加（或注册）到 **luci** 服务器。当群集或机器注册了 **luci**，每台机器的全限定主机名或 IP 地址都保存在 **luci** 数据库里。

您可以将一个 **luci** 实例的数据库传递给另外一个 **luci** 实例。这种能力提供了复制 **luci** 服务器实例的方法以及高效的升级和测试的途径。当 **luci** 实例刚被安装时，它的数据库是空的。然而，当部署新的 **luci** 服务器时，您可以从现有的 **luci** 服务器导入部分或全部的 **luci** 数据库。

在初始安装时，每个 **luci** 实例都有一个用户 — **admin**。只有 **admin** 可以将系统添加到 **luci** 服务器里。而且，**admin** 用户可以创建其他的用户帐号并决定哪些用户可以访问在 **luci** 里注册的群集和机器。您也可以将用户批量地导入到新的 **luci** 服务器里，就如同导入群集和节点一样。

当机器被加入 **luci** 服务器进行管理时，将进行一次验证。此后不需要再进行验证（除非使用的证书被 CA 调用）。然后，您可以通过 **luci** 用户界面远程地配置和管理群集和存储设备。**luci** 和 **ricci** 通过 XML 进行相互通信。

下面的图例展示了 3 个主要的 **luci** 标签页：**homebase**、**cluster** 和 **storage**。

关于 **Conga** 的更多信息，请参考《配置和管理红帽群集》以及 **luci** 服务器的在线帮助。

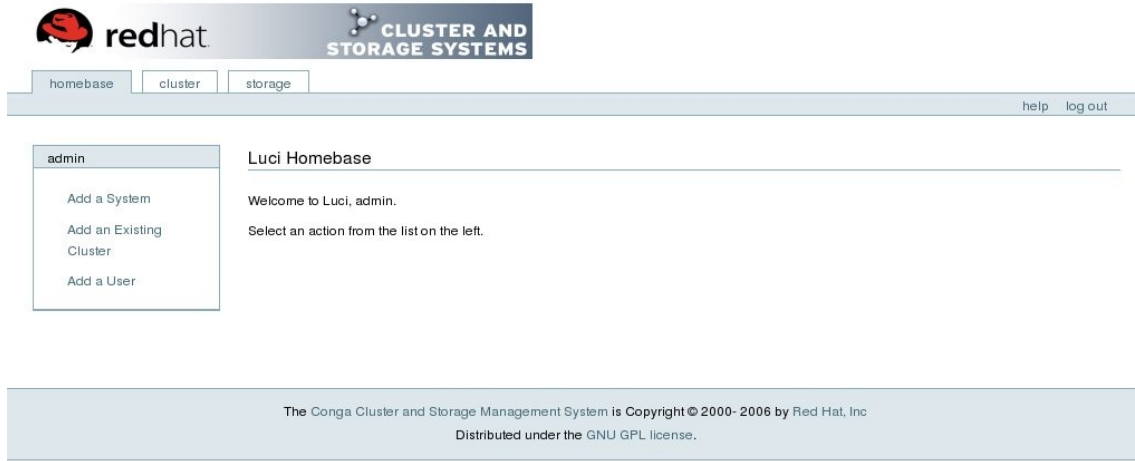


图 1.24. luci 的 homebase 标签页

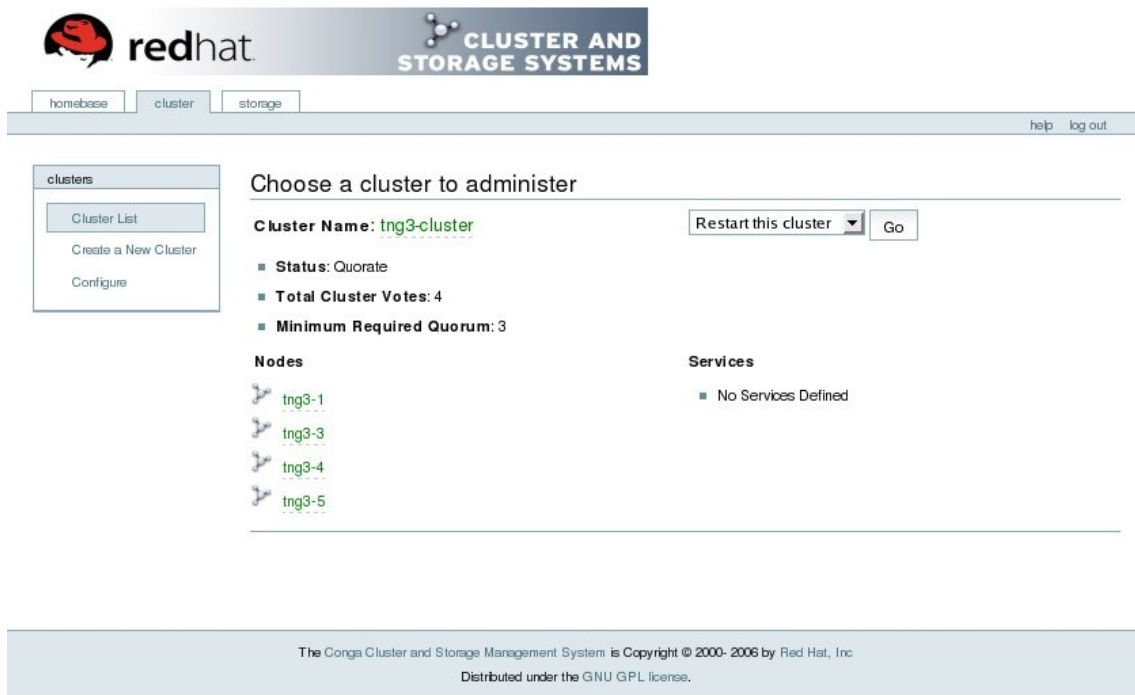


图 1.25. luci 的 cluster 标签页

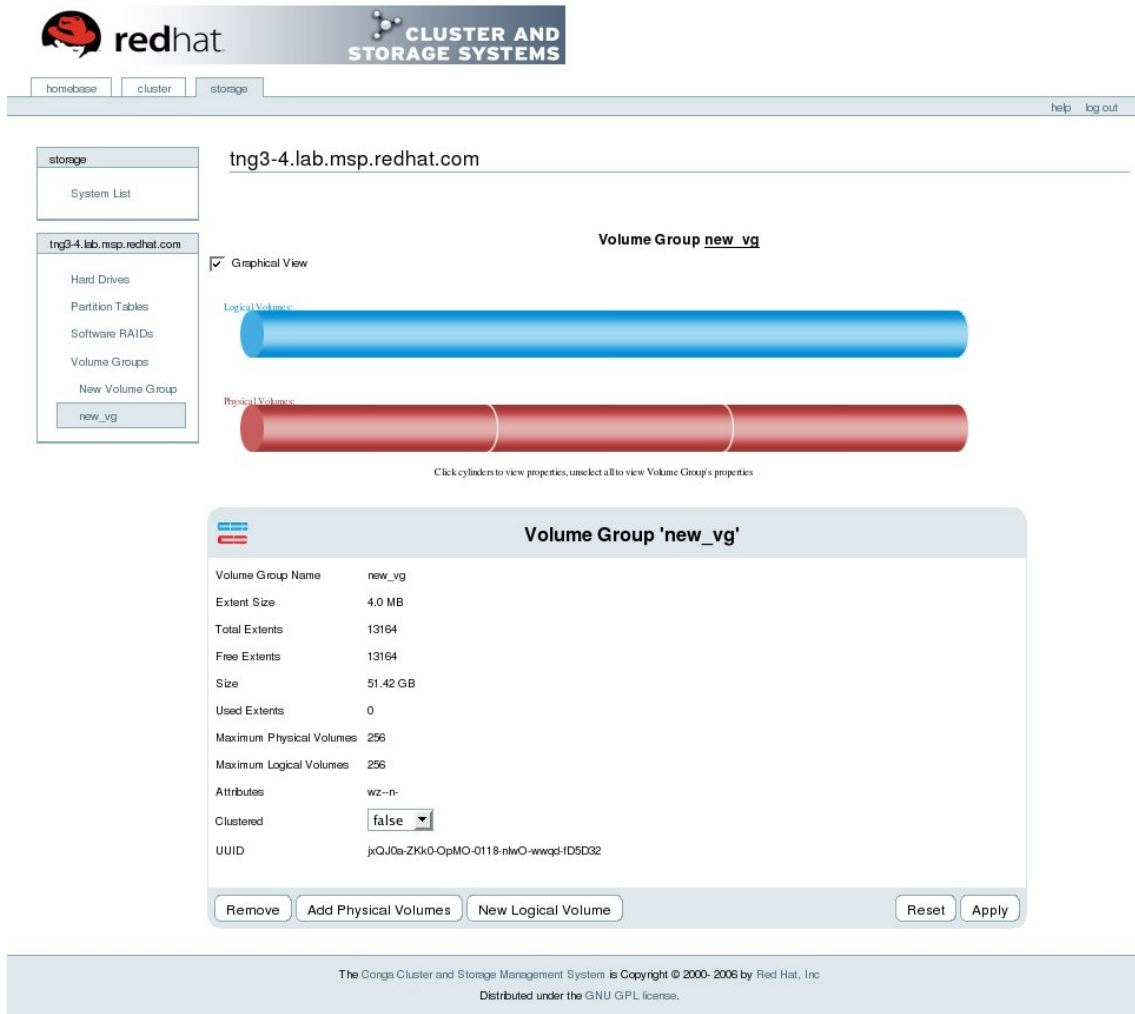


图 1.26. luci 的 storage 标签页

1.9.2. 群集管理图形界面

This section provides an overview of the **system-config-cluster** cluster administration graphical user interface (GUI) available with Red Hat Cluster Suite. The GUI is for use with the cluster infrastructure and the high-availability service management components (refer to 第 1.3 节 “Cluster Infrastructure” and 第 1.4 节 “高可用性服务管理”). The GUI consists of two major functions: the **Cluster Configuration Tool** and the **Cluster Status Tool**. The **Cluster Configuration Tool** provides the capability to create, edit, and propagate the cluster configuration file (`/etc/cluster/cluster.conf`). The **Cluster Status Tool** provides the capability to manage high-availability services. The following sections summarize those functions.

- 第 1.9.2.1 节 “Cluster Configuration Tool”
- 第 1.9.2.2 节 “Cluster Status Tool”

1.9.2.1. Cluster Configuration Tool

You can access the **Cluster Configuration Tool** (图 1.27 “Cluster Configuration Tool”) through the **Cluster Configuration** tab in the Cluster Administration GUI.

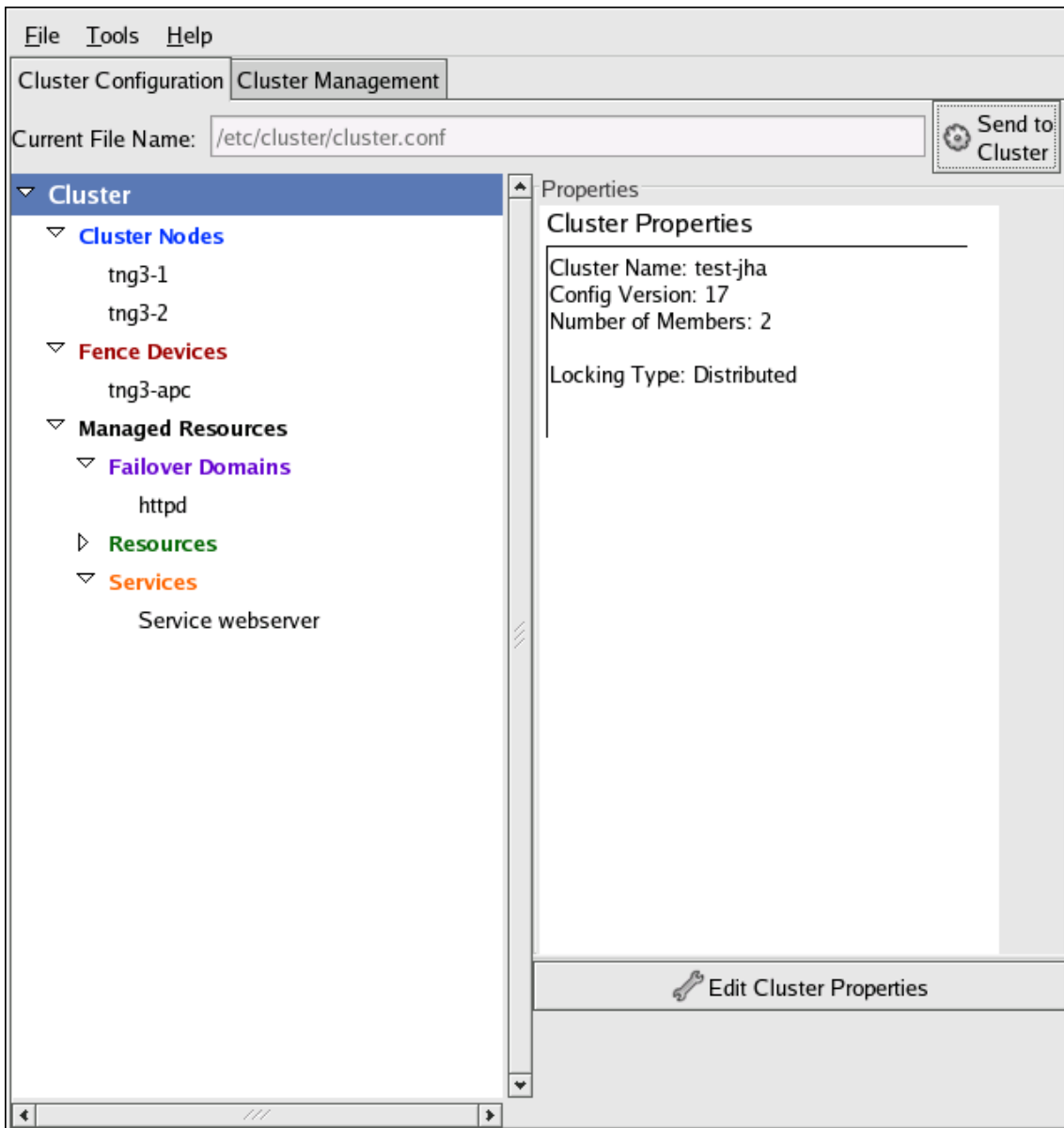


图 1.27. Cluster Configuration Tool

Cluster Configuration Tool 以分层结构的形式在面板的左边显示配置文件

(`/etc/cluster/cluster.conf`) 里的群集配置组件。组件名左边的三角形图标表示这个组件分配有一个或多个下级组件。点击这个三角图标可以展开组件下的目录树。这个图形界面显示如下的组件：

- 「Cluster Nodes」 — 显示群集节点。节点名出现在「Cluster Nodes」的下级列表里。使用右栏底部的配置按钮（在「Properties」下面），您可以添加节点、删除节点、编辑节点属性以及为每个节点配置 fencing 方法。
- 「Fence Devices」 — 显示保护设备。保护设备出现在「Fence Devices」的下级列表里。使用右栏底部的配置按钮（在「Properties」下面），您可以添加、删除保护设备、编辑其属性。在配置保护之前，您必须先为每个节点定义保护设备（用「Manage Fencing For This Node」按钮）。
- 「Managed Resources」 — 显示故障切换域、资源和服务。
 - 「Failover Domains」 — 配置一个或多个群集节点子集，用来在节点发生故障时运行高可用性服务。故障切换域出现在「Failover Domains」的下级列表里。使用右栏底部的配置按钮（在「Properties」下面），您可以创建故障切换域（当选定了「Failover Domains」项时）或编辑故障切换域的属性（当选择其中一个故障切换域时）。

- **「Resources」** — 配置用于高可用性服务的共享资源。共享资源由可用于群集里的任何高可用性服务的文件系统、IP 地址、NFS 挂载和导出以及用户创建的脚本组成。资源出现在 **「Resources」** 的下级列表里。使用右栏底部的配置按钮（在 **「Properties」** 下面），您可以创建资源（当选住了 **「Resources」**）或编辑资源属性（但选中了其中一个资源）。



注意

Cluster Configuration Tool 也提供配置专用资源的能力。专用资源是仅用于一个服务的资源。您可以在图形用户界面里的 **Service** 组件里配置专用资源。

- **「Services」** — 创建和配置高可用性服务。服务通过分配资源（共享的或专用的）、分配故障切换域、定义故障恢复策略来进行配置。服务出现在 **「Services」** 的下级列表里。使用右栏底部的配置按钮（在 **「Properties」** 下面），您可以创建服务（当选住 **「Services」**）或者编辑服务属性（当选中了某个服务）。

1.9.2.2. Cluster Status Tool

You can access the **Cluster Status Tool** (图 1.28 “**Cluster Status Tool**”) through the **Cluster Management** tab in Cluster Administration GUI.

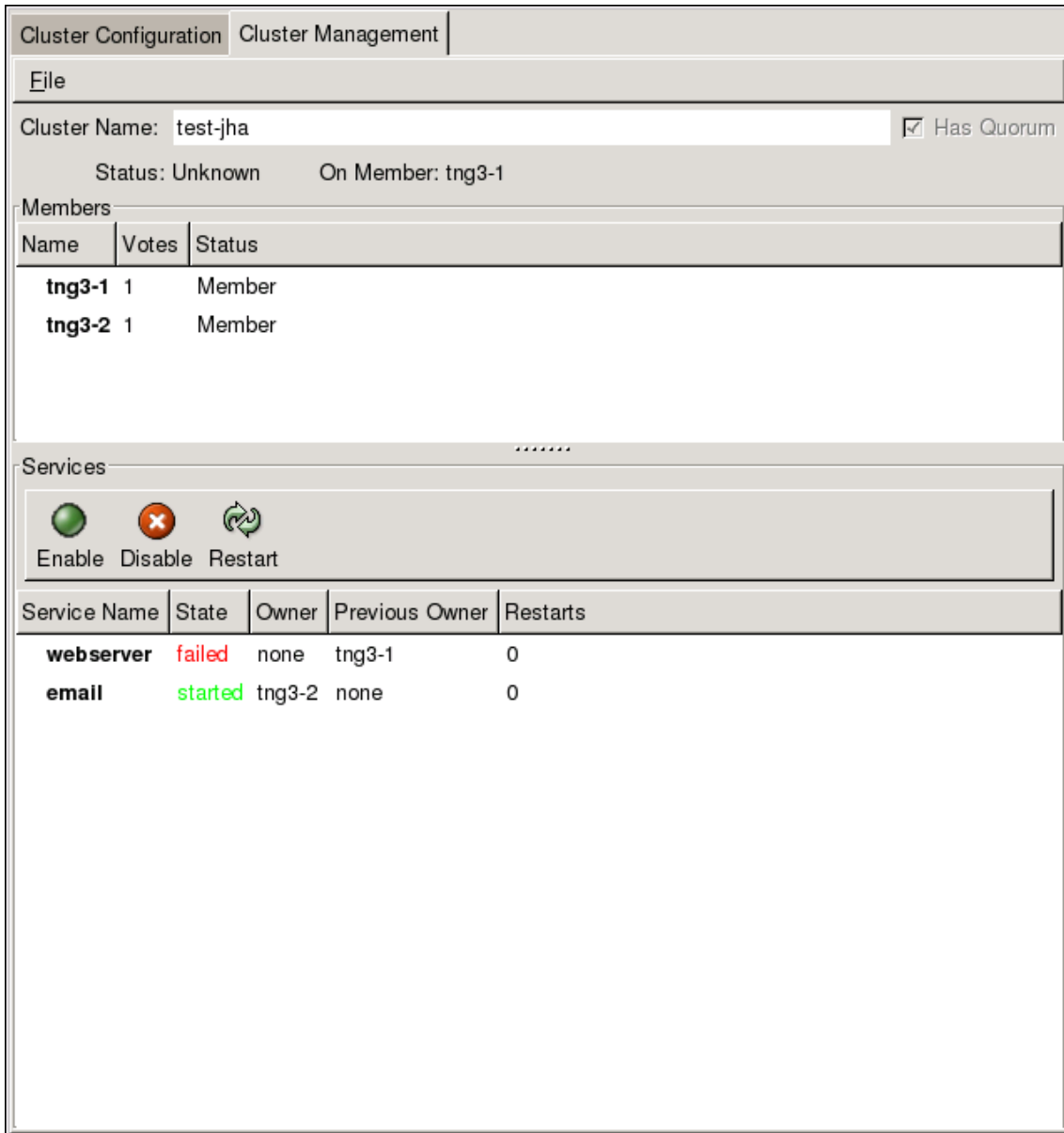


图 1.28. Cluster Status Tool

Cluster Status Tool 里显示的节点和服务由群集配置文件（`/etc/cluster/cluster.conf`）决定。您可以使用 **Cluster Status Tool** 来启用、禁用、重启或迁移高可用性服务。

1.9.3. 命令行管理工具

In addition to **Conga** and the **system-config-cluster** Cluster Administration GUI, command line tools are available for administering the cluster infrastructure and the high-availability service management components. The command line tools are used by the Cluster Administration GUI and init scripts supplied by Red Hat. 表 1.1 “命令行工具” summarizes the command line tools.

表 1.1. 命令行工具

命令行工具	用于	描述
ccs_tool — 群集配置系统工具	Cluster Infrastructure	ccs_tool 是一个制作群集配置文件的在线更新的程序。它提供创建和修改群集基础结构组件（如：创建群集、添加和删除节点）的能力。关于这个工具的更多信息，请参考 <code>ccs_tool(8)</code> 帮助页。

命令行工具	用于	描述
cman_tool — 群集管理工具	Cluster Infrastructure	cman_tool 是一个管理 CMAN 群集管理器的程序。它提供加入群集、离开群集、杀死节点或修改群集里节点的预期 quorum votes 的功能。关于这个工具的更多信息，请参考 <code>cman_tool(8)</code> 帮助页。
fence_tool — 保护工具	Cluster Infrastructure	fence_tool 是一个用来加入和离开缺省保护域的程序。具体来说，它启动保护守护进程 (fenced) 来加入域以及终止 fenced 进程来离开域。关于这个工具的更多信息，请参考 <code>fence_tool(8)</code> 帮助页。
clustat — 群集状态工具	高可用性服务管理组件	clustat 命令显示群集的状态。它显示成员信息、quorum 视图以及所有配置的用户服务的状态。关于这个工具的更多信息，请参考 <code>clustat(8)</code> 帮助页。
clusvcadm — 群集用户服务管理工具	高可用性服务管理组件	clusvcadm 命令允许您在群集里启用、禁用、迁移和重启高可用性服务。关于这个工具的更多信息，请参考 <code>clusvcadm(8)</code> 帮助页。

1.10. LINUX 虚拟服务器管理图形用户界面

本节提供对 Red Hat Cluster Suite 里可用的 LVS 配置工具的概述 — **Piranha Configuration Tool**。 **Piranha Configuration Tool** 是一个 web 浏览器图形用户界面 (GUI)，它提供结构化的方法来创建 LVS 配置文件 — `/etc/sysconfig/ha/lvs.cf`。

要访问 **Piranha Configuration Tool**，您需要在活动的 LVS 路由器上运行 **piranha-gui** 服务。您可以在本地访问 **Piranha Configuration Tool** 或通过 web 浏览器进行远程访问。本地访问您可以使用这个链接：`http://localhost:3636`。如果要从远程访问，您可以用主机名或者真实的 IP 地址后面接着 `:3636`。如果您从远程访问 **Piranha Configuration Tool**，您需要以根用户身份用 **ssh** 连接活动 LVS 路由器。

Starting the **Piranha Configuration Tool** causes the **Piranha Configuration Tool** welcome page to be displayed (refer to 图 1.29 “The Welcome Panel”). Logging in to the welcome page provides access to the four main screens or *panels*: **CONTROL/MONITORING**, **GLOBAL SETTINGS**, **REDUNDANCY**, and **VIRTUAL SERVERS**. In addition, the **VIRTUAL SERVERS** panel contains four *subsections*. The **CONTROL/MONITORING** panel is the first panel displayed after you log in at the welcome screen.

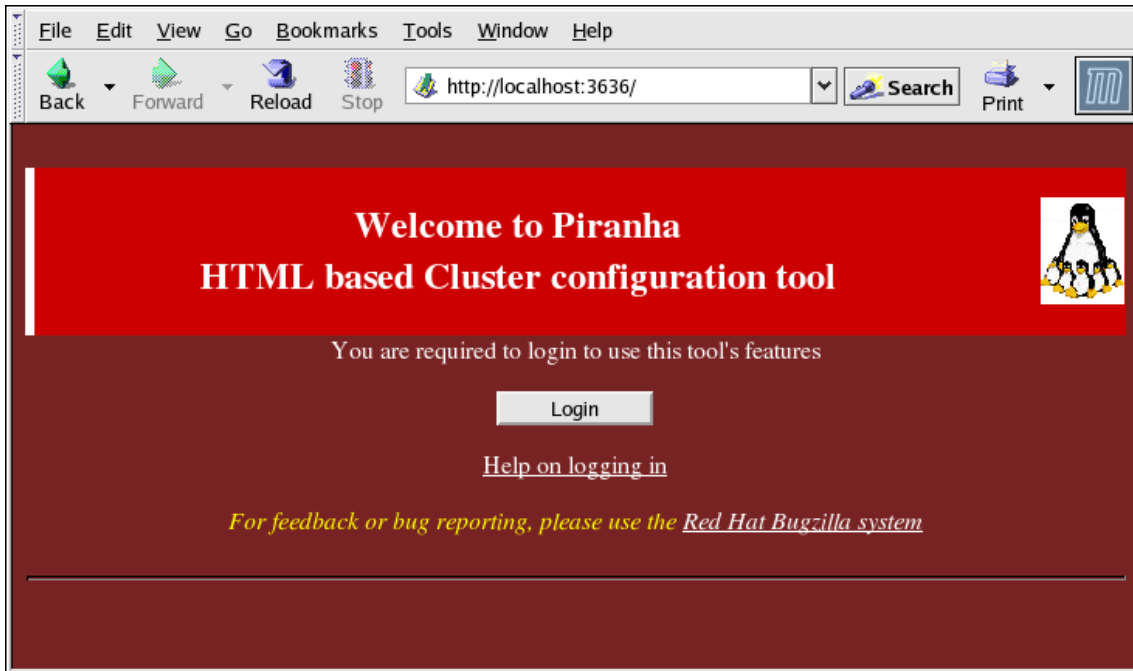


图 1.29. The Welcome Panel

下面的内容提供了对 **Piranha Configuration Tool** 配置页面的简短描述。

1.10.1. CONTROL/MONITORING

「CONTROL/MONITORING」面板显示运行状态。它显示 **pulse** 守护进程、LVS 路由表以及 LVS 所产生的 **nanny** 进程的状态。

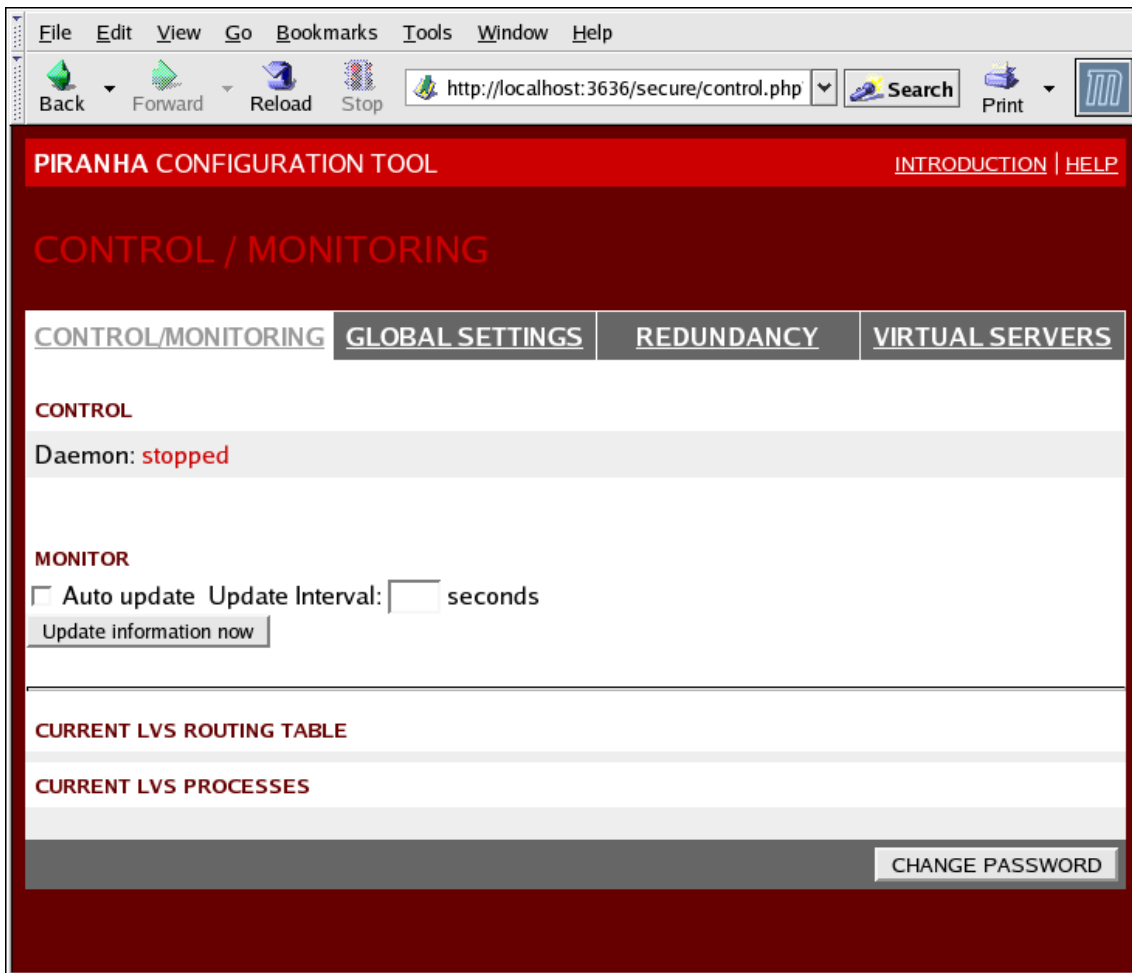


图 1.30. The CONTROL/MONITORING Panel

Auto update

让状态显示按照「Update frequency in seconds」文本框里设置时间间隔（缺省为 10 秒）自动进行更新。

我们不推荐将自动更新的时间间隔设置为少于 10 秒。这样使重新配置「Auto update」时间间隔变得困难，因为页面会过于频繁地更新。如果您遇到这个问题，只要简单地点击另外一个面板并回到「CONTROL/MONITORING」就可以了。

Update information now

提供对状态信息的手工更新。

CHANGE PASSWORD

点击这个按钮会使您进入一个帮助屏幕，上面有如何修改 Piranha Configuration Tool 的管理密码的信息。

1.10.2. GLOBAL SETTINGS

The **GLOBAL SETTINGS** panel is where the LVS administrator defines the networking details for the primary LVS router's public and private network interfaces.

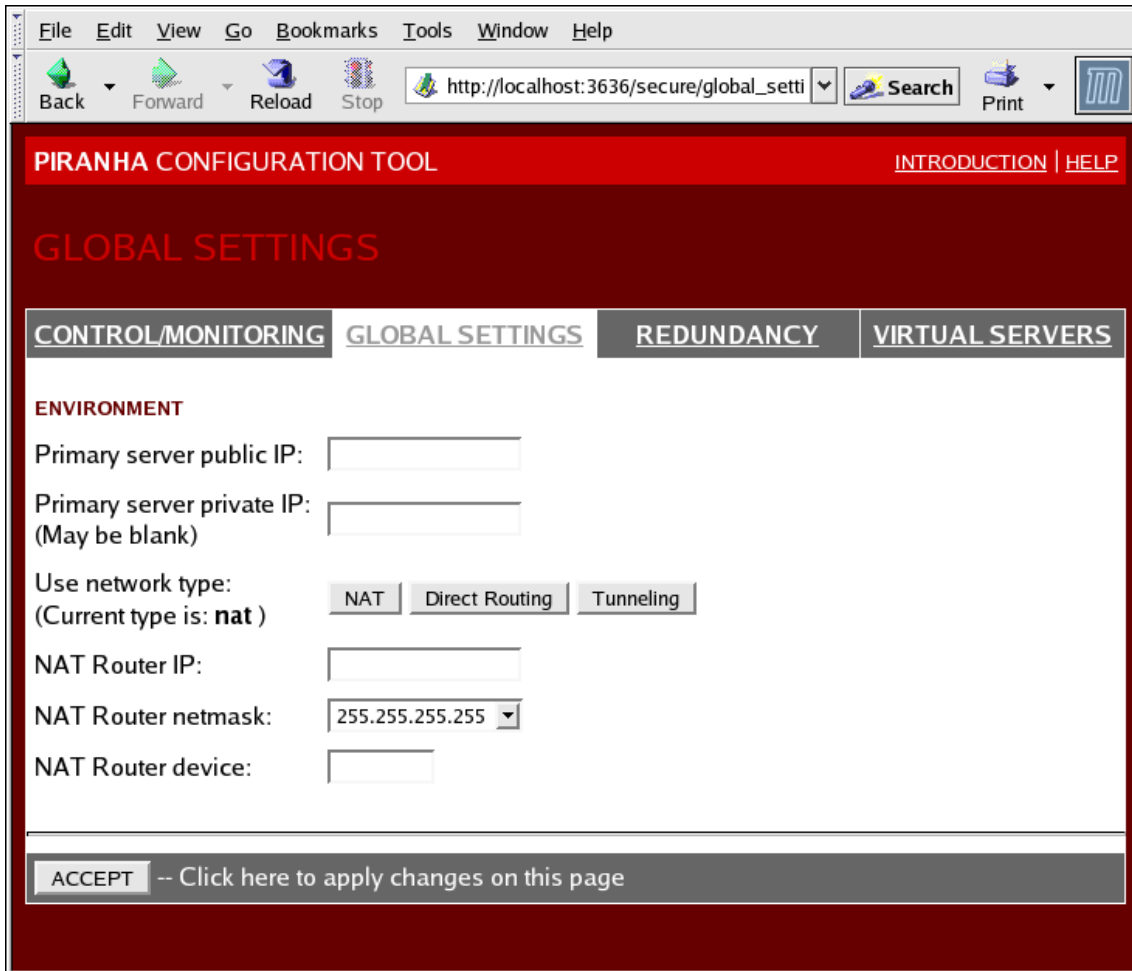


图 1.31. The GLOBAL SETTINGS Panel

The top half of this panel sets up the primary LVS router's public and private network interfaces.

Primary server public IP

主 LVS 节点的公用真实 IP 地址。

Primary server private IP

主 LVS 节点的替代网络接口的真实 IP 地址。这个地址仅用于与备份路由器之间的 heartbeat 通道。

Use network type

选择 NAT 路由

The next three fields are specifically for the NAT router's virtual network interface connected the private network with the real servers.

NAT Router IP

这个文本字段里是专用浮动 IP 地址。这个浮动 IP 地址应该作为真实服务器的网关。

NAT Router netmask

If the NAT router's floating IP needs a particular netmask, select it from drop-down list.

NAT Router device

定义用于浮动 IP 地址的网络接口的设备名称，如 `eth1:1`。

1.10.3. REDUNDANCY

「REDUNDANCY」面板允许您配置备份 LVS 路由器节点并设置不同的 heartbeat 监控选项。

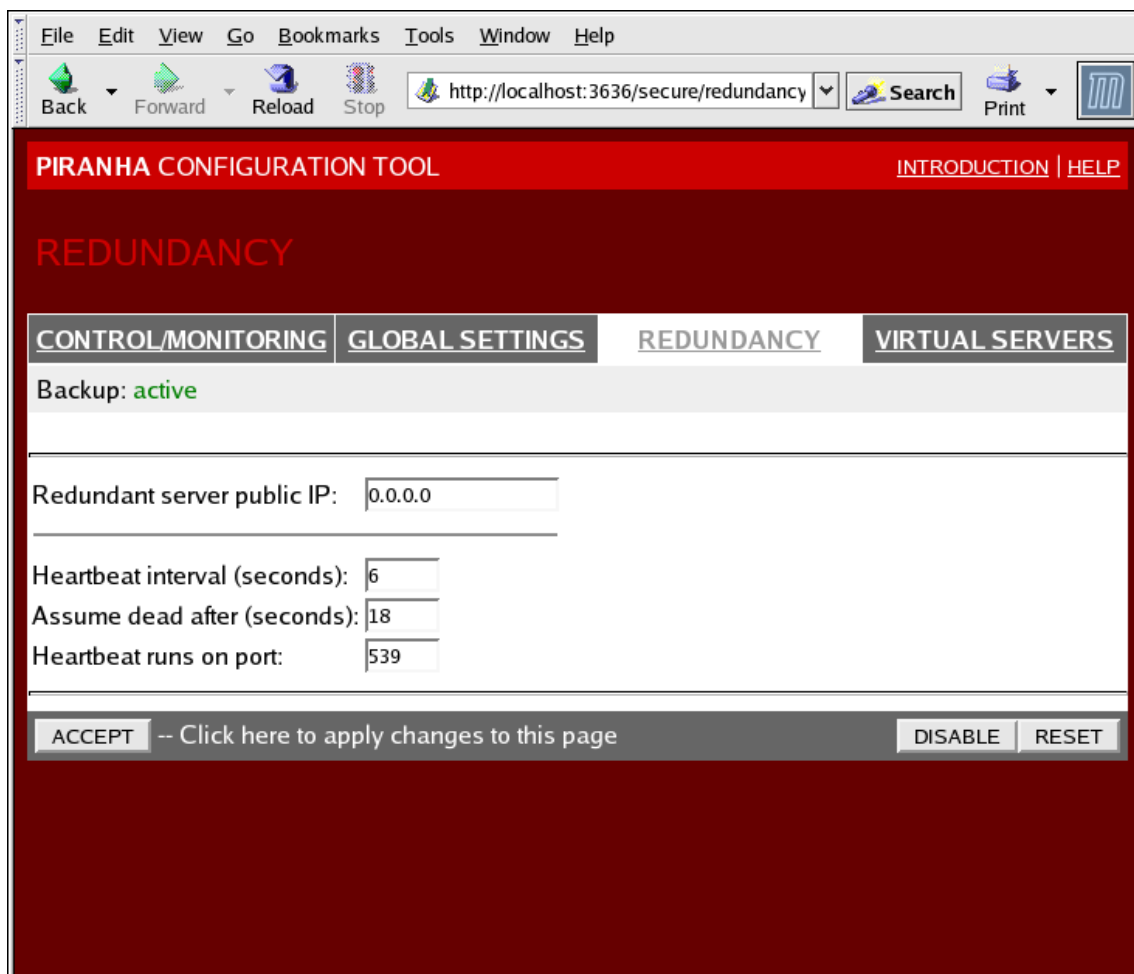


图 1.32. The REDUNDANCY Panel

Redundant server public IP

备份 LVS 路由器的公用的真实 IP 地址。

Redundant server private IP

The backup router's private real IP address.

面板的其他部分包括配置 heartbeat 通道，备份节点用它来监控主节点的故障情况。

Heartbeat Interval (seconds)

设置 heartbeat 之间间隔的秒数 — 备份节点检查主 LVS 节点功能性状态的时间间隔。

Assume dead after (seconds)

如果主 LVS 节点在这段时间内没有响应，备份 LVS 路由器节点将初始化故障切换。

Heartbeat runs on port

设置 heartbeat 与主 LVS 节点通信的端口。如果不填写这个字段，其省省值是 539。

1.10.4. VIRTUAL SERVERS

「VIRTUAL SERVERS」面板显示每个当前定义的虚拟服务器的信息。表里的每个条目都显示了虚拟服务器的状态、服务器名、分配的 IP 地址、虚拟 IP 的掩码、服务端口、使用的协议以及虚拟设备接口。

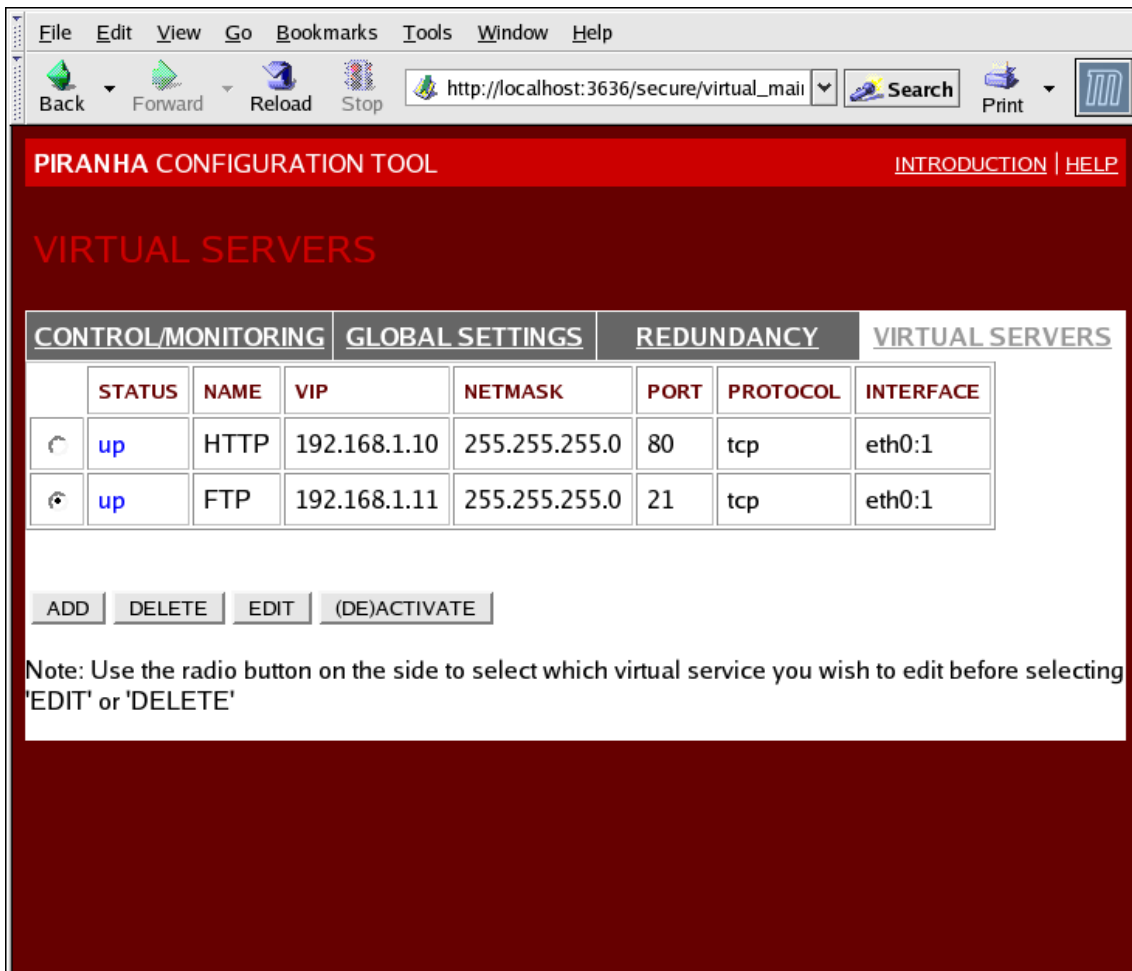


图 1.33. The VIRTUAL SERVERS Panel

「VIRTUAL SERVERS」面板里显示的每个服务器都可以在接下来的屏幕或分支 (*subsections*) 里进行配置。

点击「ADD」按钮可以添加一个服务。要删除某个服务，可以选中虚拟服务器旁的单选按钮并点击「DELETE」按钮。

要启用或禁用表里的虚拟服务器，选中单选按钮并点击「(DE)ACTIVATE」按钮。

添加了虚拟服务器后，您可以选中其左边的单选按钮并点击「EDIT」按钮进入「VIRTUAL SERVER」分支来进行配置。

1.10.4.1. 「VIRTUAL SERVER」分支

The VIRTUAL SERVER subsection panel shown in 图 1.34 “The VIRTUAL SERVERS Subsection” allows you to configure an individual virtual server. Links to subsections related specifically to this virtual server are located along the top of the page. But before configuring any of the subsections related to this virtual server, complete this page and click on the **ACCEPT** button.

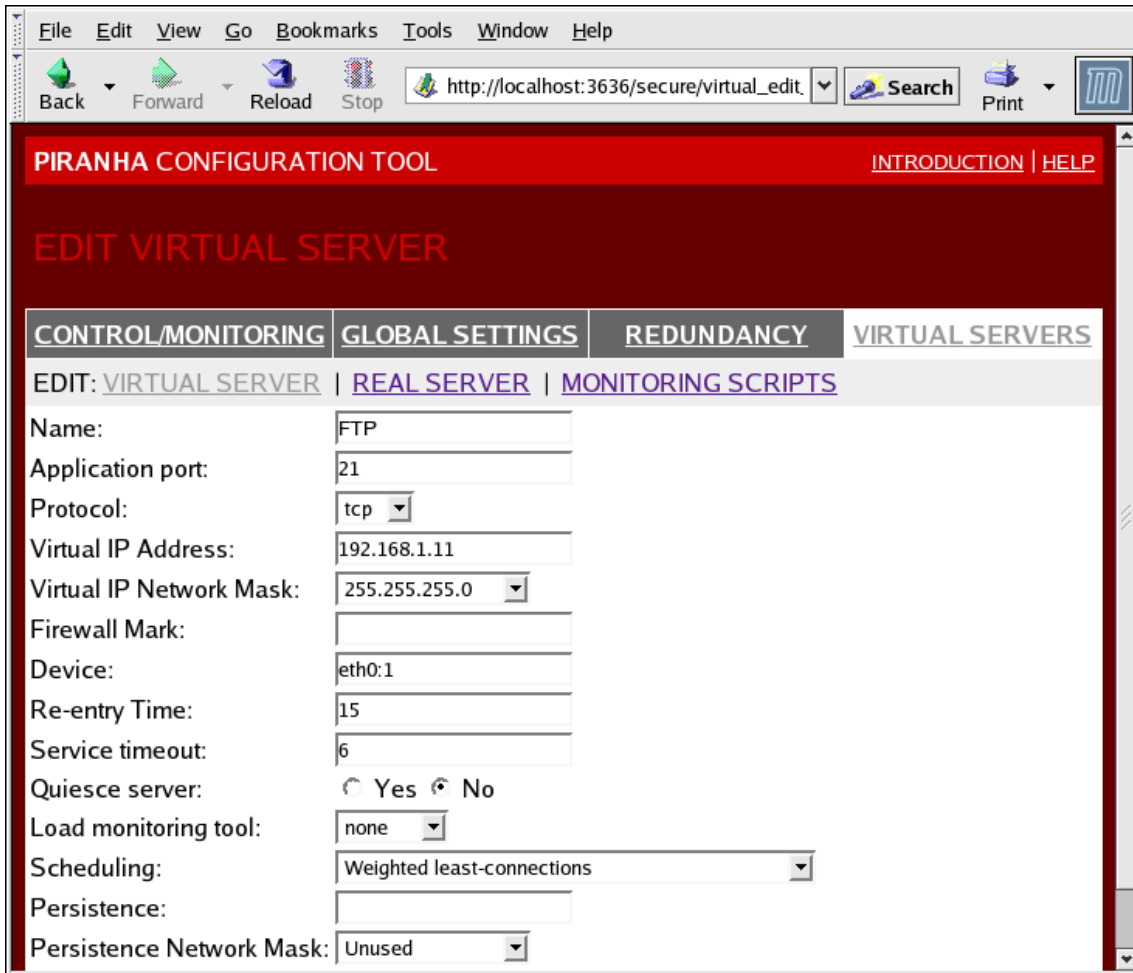


图 1.34. The VIRTUAL SERVERS Subsection

Name

标识虚拟服务器的描述性名称。这个名称不是机器的主机名，所以可以使用描述性且易于辨别的文字。您甚至可以引用虚拟服务器使用的协议，如 HTTP。

Application port

应用程序将侦听的端口。

Protocol

在下拉菜单里提供 UDP 或 TCP 选项。

Virtual IP Address

The virtual server's floating IP address.

Virtual IP Network Mask

可在下拉菜单里选择的虚拟服务器的掩码。

Firewall Mark

捆绑多端口的协议或创建使用独立但相关的协议的多端口虚拟服务器时，输入的防火墙掩码整型数值。

Device

「**Virtual IP Address**」里定义的浮动 IP 地址绑定的网络设备名称。

您应该为连接至公用网络的以太接口的公用浮动 IP 地址起别名。

Re-entry Time

服务器发生故障后，活动 LVS 路由器试图使用这个服务器前等待的时间（以秒为单位，整型数值）。

Service Timeout

定义服务器被认为发生了故障或不可用之前的秒数的整型数值。

Quiesce server

当「**Quiesce server**」单选按钮被选中时，新的服务器节点上线时，最少连接表被重置为 0，活动 LVS 路由器将发送请求，就好像所有的服务器都是刚加入群集一样。这个选项可以避免当大量的连接进入群集时，新加入的服务器超载。

Load monitoring tool

LVS 路由器可以用 **rup** 或 **ruptime** 监控不同服务器上的负载。如果您从下拉菜单里选择了 **rup**，那么每个服务器都必须运行 **rstatd** 服务。如果您选择了 **ruptime**，每个服务器都必须运行 **rwhod** 服务。

Scheduling

下拉菜单里的首选调度算法。缺省是权重最少连接 (**Weighted least-connection**) 算法。

Persistence

在处理客户事务时，如果您需要与虚拟服务器的持久的连接，可以使用这个属性。在这个文本字段里指定连接超时前允许的不活动时间（秒）。

Persistence Network Mask

要将持久性限制到特定的子网，您可以从下拉菜单里选择合适的网络掩码。

1.10.4.2. 「REAL SERVER」分支

点击面板顶部的「**REAL SERVER**」分支链接将显示「**EDIT REAL SERVER**」分支。

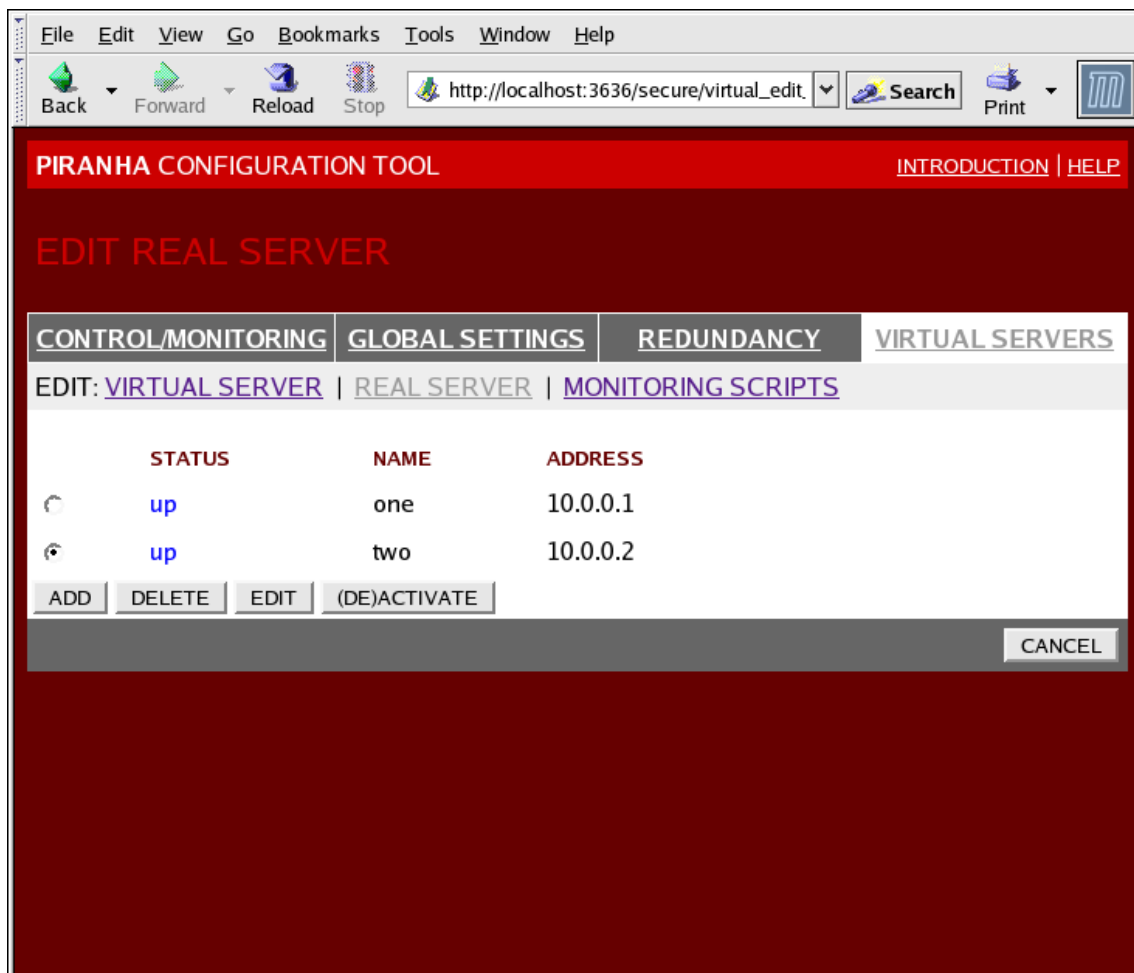


图 1.35. The REAL SERVER Subsection

Click the **ADD** button to add a new server. To delete an existing server, select the radio button beside it and click the **DELETE** button. Click the **EDIT** button to load the **EDIT REAL SERVER** panel, as seen in 图 1.36 “The REAL SERVER Configuration Panel”.

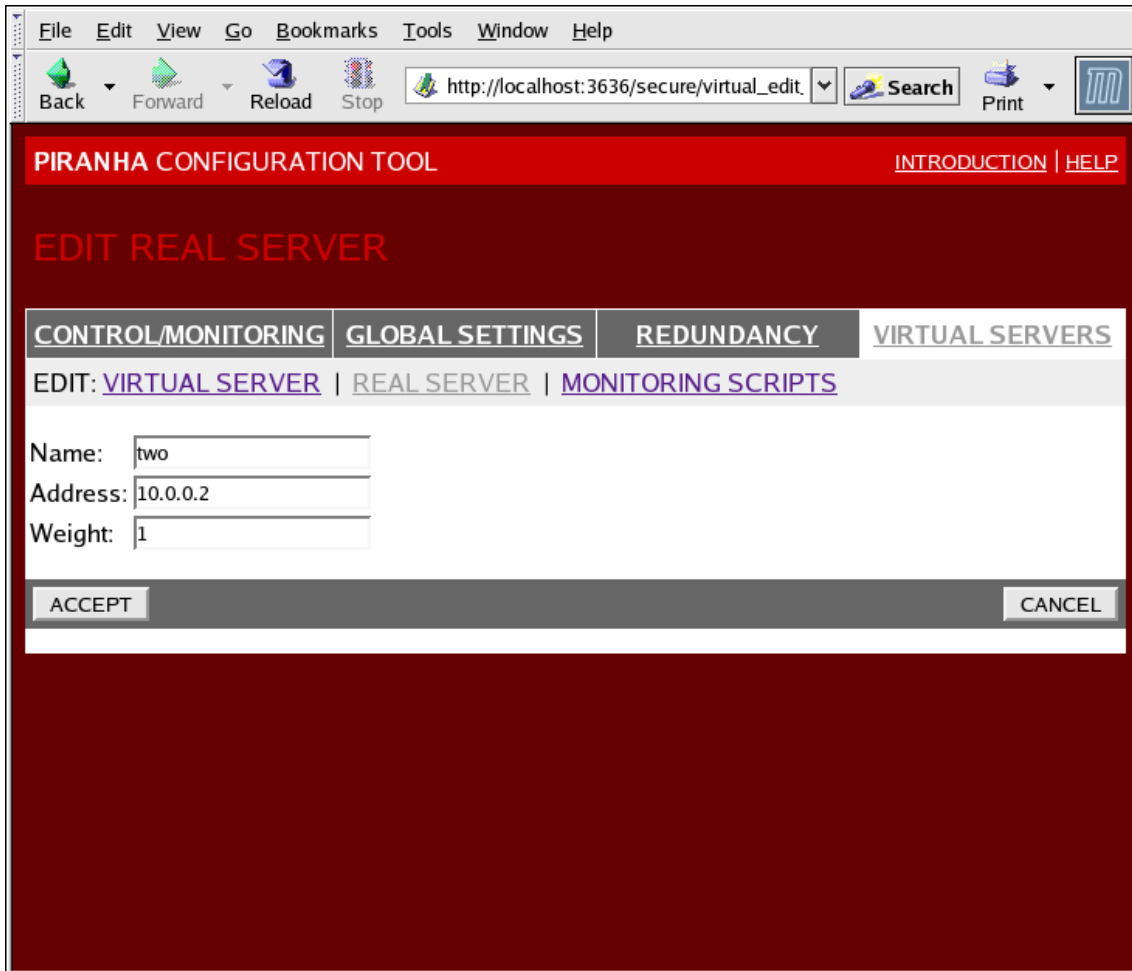
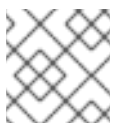


图 1.36. The REAL SERVER Configuration Panel

这个面板由 3 个字段组成：

Name

真实服务器的描述性名称。



注意

这个名称不是机器的主机名，它应该具有描述性且易于辨认。

Address

The real server's IP address. Since the listening port is already specified for the associated virtual server, do not add a port number.

Weight

An integer value indicating this host's capacity relative to that of other hosts in the pool. The value can be arbitrary, but treat it as a ratio in relation to other real servers.

1.10.4.3. EDIT MONITORING SCRIPTS Subsection

点击页面顶部的「MONITORING SCRIPTS」链接。「EDIT MONITORING SCRIPTS」分支允许管理员指定一个 send/expect 字符串序列来验证虚拟服务器服务在每个真实服务器上正常运行。管理员也可以在这里定义检查需要动态数据更新的服务的自定义脚本。

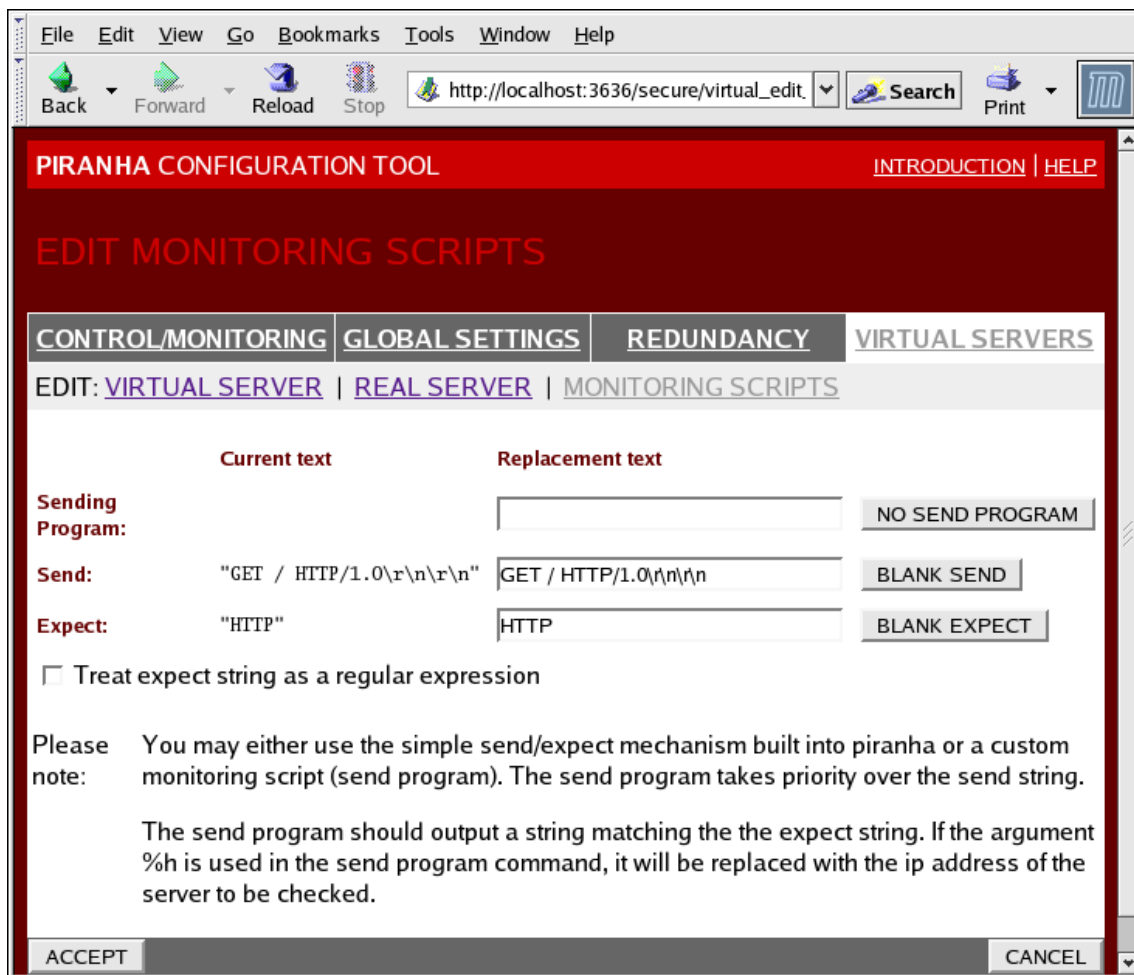
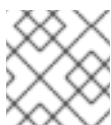


图 1.37. The EDIT MONITORING SCRIPTS Subsection

Sending Program

对于更高级的服务验证，您可以使用这个字段来指定服务检查脚本的路径。这个功能对于需要动态更新数据的服务来说尤其有用，如 HTTPS 或 SSL。

要使用这个功能，您必须编写一个返回文本型响应的脚本，将它设置可执行文件，并在「Sending Program」里输入其路径。



注意

如果在「Sending Program」字段里输入了外部程序，「Send」将被忽略。

Send

nanny 守护进程将这个字段里的字符串发送到每个真实服务器。这个字段里缺省是 HTTP 信息。您可以根据需要修改它。如果您让这个字段为空，nanny 守护进程将试图打开服务端口，如果成功则假设服务正在运行。

这个字段里只允许输入一个发送序列，且它只能包含可打印的、ASCII 字符串以及下面的转义符：

- \n 新行。
- \r 回车。
- \t 制表符。

- \将下一个字符转义。

Expect

如果服务器正常运行，它应该返回的文本型响应。如果您编写了自己的发送程序，在此输入您设定的服务器返回的响应。

[1] 虚拟服务器是一个侦听特定的虚拟 IP 地址的服务。

第 2 章 RED HAT CLUSTER SUITE 组件摘要

本章提供了 Red Hat Cluster Suite 组件的摘要，它由下面的章节组成：

- 第 2.1 节 “群集组件”
- 第 2.2 节 “Man Page”
- 第 2.3 节 “兼容的硬件”

2.1. 群集组件

表 2.1 “Red Hat Cluster Suite 软件子系统组件” summarizes Red Hat Cluster Suite components.

表 2.1. Red Hat Cluster Suite 软件子系统组件

功能	组件	描述
Conga	luci	远程管理系统 - 管理工作站。
	ricci	远程管理系统 - 被管理的工作站。
Cluster Configuration Tool	system-config-cluster	以图形方式管理群集配置的命令
群集逻辑卷管理器 (Cluster Logical Volume Manager, CLVM)	clvmd	在群集中发布 LVM 元数据更新的守护进程。它必须在群集中的所有节点中运行，如果群集中的某个节点没有运行此时间进程则会提示错误。
	lvm	LVM2 工具。为 LVM2 提供命令行工具。
	system-config-lvm	为 LVM2 提供图形化用户界面。
	lvm.conf	LVM 配置文件。完整路径是 /etc/lvm/lvm.conf
群集配置系统 (Cluster Configuration System, CCS)	ccs_tool	ccs_tool 是群集配置系统 (CCS) 的一部分。它被用来制作 CSS 配置文件的在线更新。此外，可用它将群集配置文件从使用 GFS 6.0 (或更早版本) 创建的 CCS 归档升级为 Red Hat Cluster Suite 发行版本中使用的 XML 格式。
	ccs_test	通过 ccsd 从配置文件获取信息的诊断和测试命令。
	ccsd	在所有群集节点中运行并为群集软件提供配置文件数据的 CCS 守护进程。

功能	组件	描述
	cluster.conf	这个群集配置文件。完整的路径是 /etc/cluster/cluster.conf 。
群集管理器 (Cluster Manager, CMAN)	cman.ko	CMAN 的内核模块。
	cman_tool	这是 CMAN 的管理前台。它启动和停止 CMAN 并能修改某些内部参数, 如 vote 。
	dlm_controld	由 cman 初始脚本启动的守护进程, 它管理内核中的 dlm ; 用户不可直接使用。
	gfs_controld	cman 初始脚本启动的守护进程, 它管理内核中的 gfs ; 用户不可直接使用。
	group_tool	用来获取与保护 (fencing)、DLM、GFS 相关的组群列表, 并获取调试信息, 包括 cman_tool services 在 RHEL 4 中提供的信息。
	groupd	cman 初始脚本启动的守护进程, 它是 openais/cman 和 dlm_controld/gfs_controld/fenced 之间的接口; 用户不可直接使用。
	libcman.so.<version number>	需要和 cman.ko 交互的程序所使用的库。
资源组管理器 (Resource Group Manager, rgmanager)	clusvcadm	手动启用、禁用、迁移和重启群集中用户服务的命令。
	clustat	用来显示群集状态 (包括节点成员资格和运行的服务) 的命令。
	clurgmgrd	用来处理用户服务请求, 如服务启动、服务禁用、服务迁移和服务重启的守护进程。
	clurmtabd	处理群集的 NFS 挂载表的守护进程。
保护 (Fence)	fence_apc	用于 APC 电源开关的保护代理。
	fence_bladecenter	用于带有 Telnet 接口的 IBM Bladecenter 的保护代理。
	fence_bullpap	用于 Bull Novascale 平台管理处理器 (Platform Administration Processor, PAP) 接口的保护代理。

功能	组件	描述
	fence_drac	用于 Dell 远程访问卡的保护代理。
	fence_ipmilan	用于通过 LAN 使用智能平台管理界面 (Intelligent Platform Management Interface, IPMI) 进行控制的保护代理。
	fence_wti	用于 WTI 电源开关的保护代理。
	fence_brocade	用于 Brocade 光线通道开关的保护代理。
	fence_mcddata	用于 McData 光线通道开关的保护代理。
	fence_vixel	用于 Vixel 光线通道开关的保护代理。
	fence_sanbox2	用于 SANBox2 光线通道开关的保护代理。
	fence_ilo	用于 HP ILO 接口的保护代理 (以前是 fence_rib)。
	fence_rsa	用于 IBM RSA II 的 I/O 保护代理。
	fence_gnbd	用于 GNBD 存储的保护代理。
	fence_scsi	用于 SCSI 持久保留的 I/O 保护代理。
	fence_egenera	用于 Egenera BladeFrame 系统的保护代理。
	fence_manual	手动交互的保护代理。注意：产品环境不支持这个组件。
	fence_ack_manual	fence_manual 代理的用户界面。
	fence_node	在单个节点上执行 I/O 保护的程序。
	fence_xvm	用于 Xen 虚拟机的 I/O 保护代理。
	fence_xvmd	用于 Xen 虚拟机的 I/O 保护代理宿主。
	fence_tool	加入和离开保护域的程序。
	fenced	I/O 保护守护进程。

功能	组件	描述
DLM	libdlm.so.<version number>	支持分布式锁管理器（Distributed Lock Manager, DLM）的库。
GFS	gfs.ko	在 GFS 文件系统使用的内核模块，并在 GFS 群集节点中载入。
	gfs_fsck	修复未挂载 GFS 文件系统的命令。
	gfs_grow	扩展挂载的 GFS 文件系统的命令。
	gfs_jadd	在已挂载的 GFS 文件系统里添加日志的命令。
	gfs_mkfs	在存储设备中创建 GFS 文件系统的命令。
	gfs_quota	在已挂载的 GFS 文件系统中管理配额的命令。
	gfs_tool	配置或调整 GFS 文件系统的命令。这个命令也可以收集文件系统的各种信息。
	mount.gfs	mount (8) 所调用的挂载辅助工具；用户不可直接使用。
GNBD	gnbd.ko	在客户端中使用 GNBD 设备驱动程序的内核模块。
	gnbd_export	在 GNBD 服务器中创建、导出和管理 GNBD 的命令。
	gnbd_import	在 GNBD 客户中导入和管理 GNBD 的命令。
	gnbd_serv	允许节点通过网络导出本地存储服务器守护进程。

功能	组件	描述
LVS	pulse	This is the controlling process which starts all other daemons related to LVS routers. At boot time, the daemon is started by the /etc/rc.d/init.d/pulse script. It then reads the configuration file /etc/sysconfig/ha/lvs.cf . On the active LVS router, pulse starts the LVS daemon. On the backup router, pulse determines the health of the active router by executing a simple heartbeat at a user-configurable interval. If the active LVS router fails to respond after a user-configurable interval, it initiates failover. During failover, pulse on the backup LVS router instructs the pulse daemon on the active LVS router to shut down all LVS services, starts the send_arp program to reassign the floating IP addresses to the backup LVS router's MAC address, and starts the lvs daemon.
	lvsd	一旦被 pulse 调用, lvs 守护进程就会在活跃的 LVS 路由器中运行。它读取配置文件 /etc/sysconfig/ha/lvs.cf , 调用 ipvsadm 工具来构建和维护 IPVS 路由表, 并为每个配置的 LVS 服务分配 nanny 过程。如果 nanny 报告某个服务器关闭了, lvs 将指引 ipvsadm 工具从 IPVS 路由表中删除这个服务器。
	ipvsadm	这个服务更新内核中的 IPVS 路由表。 lvs 守护进程通过调用 ipvsadm 添加、修改或删除 IPVS 路由表里的条目来设立并管理 LVS。
	nanny	nanny 监控守护进程在活跃的 LVS 路由器中运行。通过这个进程, 活跃的 LVS 路由器可以决定每个服务器的健康状况, 或者监控其工作负载。单独的进程将用于每个服务器上定义的每个服务。
	lvs.cf	这是 LVS 的配置文件。其完整路径是 /etc/sysconfig/ha/lvs.cf 。所有守护进程都直接或间接地从这个文件获取配置信息。
	Piranha Configuration Tool	这个用于监控、配置和管理 LVS 的基于 web 的工具。它是维护 /etc/sysconfig/ha/lvs.cf LVS 配置文件的默认工具。

功能	组件	描述
	send_arp	在故障切换过程中，当浮动 IP 地址从一个节点更改为另一个节点时，这个程序发送 ARP 广播。
Quorum 磁盘	qdisk	用于 CMAN / Linux-Cluster 的基于磁盘的 quorum 守护进程。
	mkqdisk	群集 Quorum 磁盘工具。
	qdiskd	群集 Quorum 磁盘守护进程。

2.2. MAN PAGE

下面的内容列出了和 Red Hat Cluster Suite 相关的 man page 以及其它资源。

- 群集基础结构
 - ccs_tool (8) - 制作 CCS 配置文件的在线更新的工具
 - ccs_test (8) - 诊断运行的群集配置系统的工具
 - ccsd (8) - 访问 CCS 群集配置文件的守护进程
 - ccs (7) - 群集配置系统
 - cman_tool (8) - 群集管理工具
 - cluster.conf [cluster] (5) - 群集产品的配置文件
 - qdisk (5) - 用于 CMAN / Linux-Cluster 的基于磁盘的 quorum 守护进程
 - mkqdisk (8) - 群集 Quorum 磁盘工具
 - qdiskd (8) - 群集 Quorum 磁盘守护进程
 - fence_ack_manual (8) - 作为手工 I/O 保护的一部分运行的程序
 - fence_apc (8) - APC MasterSwitch 的 I/O 保护代理
 - fence_bladecenter (8) - IBM Bladecenter 的 I/O 保护代理
 - fence_brocade (8) - IBrocade FC switch 的 I/O 保护代理
 - fence_bullpap (8) - PAP 管理控制台控制的 Bull FAME 架构的 I/O 保护代理
 - fence_drac (8) - Dell 远程访问卡的保护代理
 - fence_egenera (8) - Egenera BladeFrame 的 I/O 保护代理
 - fence_gnbd (8) - 基于 GNBD 的 GFS 群集的 I/O 保护代理
 - fence_ilo (8) - HP Integrated Lights Out card 的 I/O 保护代理

- fence_ipmilan (8) - IPMI 通过局域网控制的机器的 I/O 保护代理
- fence_manual (8) - 作为手工 I/O 保护的一部分运行的程序
- fence_mcddata (8) - 用于 McData FC switch 的 I/O 保护代理
- fence_node (8) - 在单个节点上执行 I/O 保护的程序
- fence_rib (8) - 用于 Compaq Remote Insight Lights Out card 的 I/O 保护代理
- fence_rsa (8) - IBM RSA II 的 I/O 保护代理
- fence_sanbox2 (8) - 用于 QLogic SANBox2 FC switch 的 I/O 保护代理
- fence_scsi (8) - 用于 SCSI persistent reservation 的 I/O 保护代理
- fence_tool (8) - 加入和离开保护域的程序
- fence_vixel (8) - 用于 Vixel FC switch 的 I/O 保护代理
- fence_wti (8) - 用于 WTI Network Power Switch 的 I/O 保护代理
- fence_xvm (8) - 用于 Xen 虚拟机的 I/O 保护代理
- fence_xvmd (8) - 用于 Xen 虚拟机的 I/O 保护代理守护进程
- fenced (8) - I/O 保护守护进程
- 高可用性服务管理
 - clusvcadm (8) - 群集用户服务管理工具
 - clustat (8) - 检查群集状态的工具
 - Clurgmgrd [clurgmgrd] (8) - 资源组（群集服务）管理者守护进程
 - clurmtabd (8) - 群集 NFS 远程挂载表守护进程
- GFS
 - gfs_fsck (8) - 脱机 GFS 文件系统检查程序
 - gfs_grow (8) - 扩展 GFS 文件系统
 - gfs_jadd (8) - 在 GFS 文件系统里添加日志 (journal)
 - gfs_mount (8) - GFS 挂载选项
 - gfs_quota (8) - 控制 GFS 磁盘配额 (quota)
 - gfs_tool (8) - GFS ioctl 调用接口
- 群集逻辑卷管理者 (Cluster Logical Volume Manager)
 - clvmd (8) - 群集 LVM 守护进程
 - lvm (8) - LVM2 工具

- lvm.conf [lvm] (5) - LVM2 的配置文件
- lvmchange (8) - 修改逻辑卷管理者的属性
- pvcreate (8) - 初始化 LVM 使用的磁盘或分区
- lvs (8) - 报告逻辑卷的信息
- 全球网络块设备 (Global Network Block Device)
 - gnbd_export (8) - 导出 GNBD 的接口
 - gnbd_import (8) - 操纵客户机上的 GNBD 块设备
 - gnbd_serv (8) - gnbd 服务器守护进程
- LVS
 - pulse (8) - 监控群集节点健康状况的 heartbeating 守护进程
 - lvs.cf [lvs] (5) - lvs 的配置文件
 - lvscan (8) - 扫描 (所有磁盘) 的逻辑卷。
 - lvsd (8) - 控制红帽群集服务的守护进程
 - ipvsadm (8) - Linux 虚拟服务器管理
 - ipvsadm-restore (8) - 从标准输入恢复 IPVS 表
 - ipvsadm-save (8) - 把 IPVS 保存至标准输出
 - nanny (8) - 监控群集里服务的状态的工具
 - send_arp (8) - 将新的 IP 地址/MAC 地址映射通知网络的工具

2.3. 兼容的硬件

有关与Red Hat Cluster Suite组件兼容的硬件的信息 (例如: 支持保护设备、存储设备和光纤交换机), 请参考硬件配置指南, 地址为 http://www.redhat.com/cluster_suite/hardware/。

附录 A. 修订记录

修订 3-8.400 Rebuild with publican 4.0.0	2013-10-31	Rüdiger Landmann
修订 3-8 Rebuild for Publican 3.0	2012-07-18	Anthony Towns
修订 1.0-0 统一发布重点	Tue Jan 20 2008	Paul Kennedy

索引

C

cluster

displaying status, [Cluster Status Tool](#)

cluster administration

displaying cluster and service status, [Cluster Status Tool](#)

cluster component compatible hardware, [兼容的硬件](#)

cluster component man pages, [Man Page](#)

cluster components table, [群集组件](#)

Cluster Configuration Tool

accessing, [Cluster Configuration Tool](#)

cluster service

displaying status, [Cluster Status Tool](#)

command line tools table, [命令行管理工具](#)

compatible hardware

cluster components, [兼容的硬件](#)

Conga

overview, [Conga](#)

Conga overview, [Conga](#)

F

feedback, [反馈](#)

I

introduction, [简介](#)

other Red Hat Enterprise Linux documents, [简介](#)

L

LVS

direct routing

requirements, hardware, [直接路由](#)

requirements, network, [直接路由](#)

requirements, software, [直接路由](#)

routing methods

NAT, [路由方法](#)

three tiered

high-availability cluster, [Three-Tier LVS Topology](#)

M

man pages

cluster components, [Man Page](#)

N

NAT

routing methods, LVS, [路由方法](#)

network address translation (见 [NAT](#))

O

overview

economy, [Red Hat GFS](#)

performance, [Red Hat GFS](#)

scalability, [Red Hat GFS](#)

P

Piranha Configuration Tool

CONTROL/MONITORING, [CONTROL/MONITORING](#)

EDIT MONITORING SCRIPTS Subsection, [EDIT MONITORING SCRIPTS Subsection](#)

GLOBAL SETTINGS, [GLOBAL SETTINGS](#)

login panel, [Linux 虚拟服务器管理图形用户界面](#)

necessary software, [Linux 虚拟服务器管理图形用户界面](#)

REAL SERVER subsection, [「REAL SERVER」分支](#)

REDUNDANCY, [REDUNDANCY](#)

VIRTUAL SERVER subsection, [VIRTUAL SERVERS](#)

Firewall Mark, [「VIRTUAL SERVER」分支](#)

Persistence, [「VIRTUAL SERVER」分支](#)

Scheduling, [「VIRTUAL SERVER」分支](#)

Virtual IP Address, [「VIRTUAL SERVER」分支](#)

VIRTUAL SERVERS, [VIRTUAL SERVERS](#)

R

Red Hat Cluster Suite

components, [群集组件](#)

T

table

cluster components, [群集组件](#)

command line tools, [命令行管理工具](#)