



Linux 系统运维之运维监控

UNIXHOT 运维社区

<http://www.unixhot.com>

版权信息:

Copyright (c) 2010 Zhao Shundong. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

使用说明:

1. 为保证本文的完整性和可用性, 本文遵循 GFDL 协议。
2. 可以在 <http://www.unixhot.com/pdf/monitor.pdf> 找到本文的最新版本。
3. 本文仅供参考使用, 不承担任何因文档错误而造成的任何损失。
4. 有任何问题可以在 UnixHot 运维社区讨论交流。
5. 有相关问题或业务合作。请邮件至 admin@unixhot.com。

相关文档:

- | | |
|-----------------------------|---|
| 1. 《Linux 系统运维之系统架构》 | http://www.unixhot.com/pdf/cluster.pdf |
| 2. 《Linux 系统运维之系统管理》 | http://www.unixhot.com/pdf/admin.pdf |
| 3. 《Linux 系统运维之运维监控》 | http://www.unixhot.com/pdf/monitor.pdf |
| 4. 《Linux 系统运维之性能优化》 | http://www.unixhot.com/pdf/tunning.pdf |
| 5. 《Linux 系统运维之运维规范》 | http://www.unixhot.com/pdf/guifan.pdf |
| 6. 《Linux 系统运维之 MySQL DBA》 | http://www.unixhot.com/pdf/mysql.pdf |
| 7. 《Linux 系统运维之 Oracle DBA》 | http://www.unixhot.com/pdf/oracle.pdf |

修订历史记录

日 期	版 本	说 明
2009-08-01	V1.0	创建本文原始版本以网页形式发布
2010-07-26	V1.1	添加 第八章 自动化批量部署
2010-07-27	V1.2	添加 GFDL 协议和使用说明
2010-07-29	V1.3	进行详细的修改工作
2010-07-30	V1.4	更名为 Linux 系统运维之监控，添加 Cacti 监控
2010-08-01	V1.5	添加 Smokeping 监控
2010-11-25	V1.6	添加网站流量统计分析
2010-12-06	V1.7	正式网络发布
2011-01-01	V1.8	提升各软件至最新版本，增加有关 Nagios 最新信息
2011-01-10	V1.9	增加详细 rrdtool 工具的使用

内容简介

UNIXHOT 运维社区致力于为想成为系统运维工程师、系统集成工程师、系统架构师、MySQL DBA 和 Oracle DBA 的互联网朋友们创建一个开源的、共享的、完整的、创新的、一站式的学习和交流平台。运维社区现在有运维导航、运维商城、运维知道、运维 Wiki 四个频道。欢迎大家加入，让我们成为一个圈子。

本文是系统运维工程师的实用手册，主要讲解基于 Linux 平台运维工作中需要掌握的监控知识。本文全部基于实际应用，部署步骤详细，可以直接复制用于生产环境的安装配置，帮助刚刚接触 Linux 运维的朋友们，迅速搭建企业级的监控平台。

本文共分为五个部分：

第一部分简单的介绍了监控的重要性，需要监控的对象，和部署监控平台前的规划，让大家可以在前期有个条理化的思路，并对整体架构有良好的认识。

第二部分通过目前最流行的监控软件 Nagios 来介绍如何实现 7x24 小时的监控、预警和报警系统，并通过介绍如何编写 Nagios 插件，来让工程师们可以自定义监控对象和方法，拓展并满足工作需求。最后也讲解了我们实际中会用到的权限的设置和 Nagios 的优化工作。

第三部分讲的是目前图形化监控比较优秀的 Cacti 软件，它通过 SNMP 协议获取监控数据，通过 rrdtool 工具完美的展现系统或者网络设备的流量和性能趋势图，便于工程师及时发现问题，并通过数据的叠加，分析系统负载的周期性，为调优工作获得最直接也最准确的性能报表。

第四部分讲的 Smokeping 这个网络监控软件，它也是 rrdtool 作者的佳作，天生的分布式设计，可以更准确的展现目前的网络状况，出现网络故障时可以快速的分析出现故障的节点，是企业网络监控中一款很优秀的开源软件。

第五部分简单介绍了网站流量的统计分析，由于目前 Google 和百度都提供了很优秀的流量分析统计工具，朋友们可以自己了解开源的日志分析工具 AWStat 的配置与使用。

目录 (Contents)

第一部分 监控概述

- 第 1 章 监控的重要性
- 第 2 章 监控的对象
- 第 3 章 监控的方法

第二部分 Nagios 监控平台

- 第 1 章 Nagios 简介
- 第 2 章 Nagios 服务器端部署
- 第 3 章 Nagios 配置文件简介
- 第 4 章 Nagios 外部监控
- 第 5 章 Nagios 监控远程 Linux 服务器
- 第 6 章 Nagios 监控远程 Windows 服务器
- 第 7 章 Nagios 插件开发
- 第 8 章 Nagios 自动化批量部署
- 第 9 章 Nagios 用户和权限管理
- 第 10 章 Nagios 使用杂谈

第三部分：Cacti 监控平台

- 第 1 章 Cacti 简介
- 第 2 章 SNMP 协议
- 第 3 章 rrdtool 工具
- 第 4 章 Cacti 安装
- 第 5 章 Cacti 配置管理
- 第 6 章 Spine (Cactid)

第四部分：Smokeping 网络监控平台

- 第 1 章 Smokeping 简介

第 2 章 Smokeping 安装配置

第 3 章 Smokeping 分布式

第五部分：网站流量统计分析

第 1 章 名词解释

第 2 章 Google Analytics

第 3 章 百度统计

前言 (Preface)

我无意去写一本书，我称本文为实用手册或者工作文档，只是按出刊物的编排格式来整理而已，主要是方便大家参考查阅，喜欢的朋友也可以直接打印出来，参考本文你可以快速的学习和掌握 Linux 系统运维中的监控工作，在运维社区有很多朋友参考本文构建自己企业的监控平台，也欢迎大家来运维社区一起讨论学习。

UNIXHOT 运维社区的同志们在钻研技术方面都有很好的积极性，并没有在工作若干年后成为一名“文档哥”。对于技术细节喜欢刨根问底，这让很多新加入的朋友们都感觉到浓厚的学习氛围，我也时刻被这种气氛感染着，从而去不停的探索，不停的测试、不停的学习。在这里希望更多的朋友们加入，请用 Google 搜索“运维社区”，我们始终站在搜索结果的第一位！

在您开始阅读本文之前，我愿意把自己的学习方法和大家分享一下，仅供参考：

我学习知识的方式是一般分为四步，总结起来就是从哪里开始，从哪里结束，起点同样是终点。我称自己的学习方法为“圆形学习法”：

第一步、先了解该程序或者方案的工作原理和一些基础知识。

第二步、搜索一篇文档自己来做测试，遇到问题要自己解决，自己确实解决不了时请教他人。

第三步、超越基本，做一些高级应用的测试，如分布式部署、性能优化等。

第四步、回到起点，深入研究其工作原理，并归纳总结，分享自己的成果。

第一部分 监控概述

在这一部分会简单介绍下监控的重要性，为什么要做监控，需要监控哪些对象，以及如何规划一个监控平台等。要知道我们需要做什么，怎么做，做完后的结果是什么。对于监控后报警信息的处理不在本文探讨范围呢，相关问题请在运维论坛或者运维知道讨论。

第 1 章 监控的重要性

单单从“监控”两个字来谈，范围之广可以涵盖我们生活的方方面面，我们生活和工作中处处可见视频监控的摄像机；机房中的电压电流监控、干湿计、温度计；值班室的网络监控，网站流量监控等等都是监控的范畴。但本文仅仅讲解和计算机网络有关的监控，主要针对我们生产应用的服务器性能和应用服务状态的监控，以及服务器或者网络设备上的网络流量监控。

监控的重要性？大家想想以下情况的发生：

从应用层角度来看：你公司业务的 WEB 服务器拒绝连接了？客户无法给你们的邮箱发邮件了？论坛的大量用户无法登陆？DNS 无法正常解析？机房流量超过购买的带宽限制了？数据库服务器宕机了，用户无法写入数据了？我想上面的任何一个问题出现的话，对我们的业务造成的影响都是巨大的。

从服务器、应用和网络设备的角度来看：目前服务器的 CPU 负载多少，一天中那个时刻负载高，一个月中哪一天服务器 CPU 的负载比较高？磁盘的使用率是多少？网卡的进口和出口流量？内存的使用率是多少？目前的磁盘 I/O, 是否考虑更换转速更快的磁盘？Apache 的连接数目前是多少，最大是多少？缓存服务器的缓存命中率是多少？MySQL Replication 是否同步？即使同步了，主从库数据是否一致？Oracle 的 Redo log 切换是否频繁，是否需要增加日志组？DBWR 进程默认一个够用吗？数据库表空间的使用率？RMAN 备份是否成功？路由器或者交换机有什么错误日志.....？如果继续写下去，我想可以出一本书，专门来写监控对象，因为太多太多了，不同的公司，不同的应用，就会增加更多需要监控的对象。

通过上面列举的比较常见的问题，我想，监控的重要性就不言而喻了。我们如何及时的获得这些应用的运行状态信息，在出现问题时能够及时知道，这就是监控要做的事情。

第 2 章 监控的对象

监控的对象？似乎没有确切的答案，在上一节讲解监控重要性所列举的内容，就是需要监控的对象。简单的说，监控的对象就是一切你想知道的状态的一个集合。

通常情况下，我们可以将监控对象这么来分：

1. 服务器监控，主要监控服务器如：CPU 负载、内存使用率、磁盘使用率、登陆用户数、进程状态、网卡状态等。
2. 应用程序监控，主要监控该应用程序的服务状态，吞吐率和响应时间，因为不同应用需要监控的对象不同，这里不一一列举。
3. 数据库监控，之所以把数据库监控单独列出来，足以说明它的重要性，一般监控数据库状态，数据库表或者表空间的使用情况，是否有死锁，错误日志，性能信息等等。
4. 网络监控，主要监控当前的网络状况，网络流量等。

以上四条应该算是最基本的，也是保证网站正常运行必须要知道的几点内容，这样才能实现我们常说的“运筹帷幄之中，决胜千里之外”。

第 3 章 监控的方法

前两个章节，介绍了监控的重要性，并了解了我们需要监控哪些内容，当然在开始之前，我们要知道有哪些工具和方法可以进行监控工作。

目前监控工具多如牛毛，针对不同的行业，针对不同的应用，针对不同的服务都有很多监控工具。有客户端的监控和浏览器的监控，本书只讲解基于 Linux 平台的开源监控工具。

Linux 平台下的监控方法基本上可以分为两种，第一种是通过外部检测命令或者代理程序获得被监控主机的相关状态，代表工具：Nagios、Munin、Ganglia。第二种是通过 SNMP 协议来实现对服务器和网络设备的监控工作，代表工具：Cacti。其中 Nagios 可以使用 `check_snmp` 命令，或者自己写脚本来使用 SNMP 协议获取数据。

细心的朋友可以发现，从开始我们就用“平台”两个字，之所以把一个软件称之为“平台”足以说明其强大性，通过软件本身功能再加上自己的扩展，我们可以通过该软件监控我们日常工作中所有需要监控的对象。

第二部分 Nagios 监控

第 1 章 Nagios 简介

1.1 什么是 Nagios

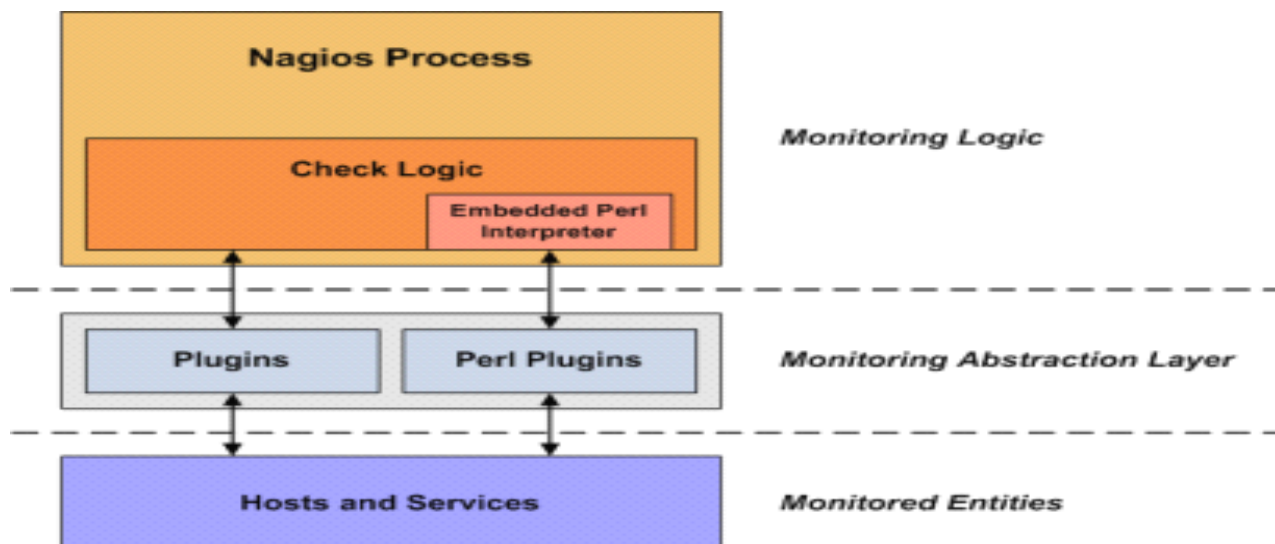
Nagios 是一款用于系统和网络监控的应用程序。它可以在你设定的条件下对主机和服务进行监控，并根据管理员所设置的报警阈值和报警方式在监控对象报警时和恢复时通知管理员。

Nagios 强大之处在于灵活的插件式监控和报警功能，如果你想获得系统性能方面的图形信息可以使用 Cacti，也可以把 Nagios 和 Cacti 有效的集成起来，这要看你实际的应用需求了。

1.2 Nagios 的特性

- 1> 监控网络服务（SMTP、POP3、HTTP、NNTP、PING 等）。
- 2> 监控主机资源（处理器负载、磁盘利用率等）。
- 3> 简单地插件设计使得用户可以方便地扩展自己服务的检测方法。
- 4> 并行服务检查机制。
- 5> 具备定义网络分层结构的能力，用“parent”主机定义来表达网络主机间的关系，这种关系被用来发现和明晰主机宕机或不可达状态。
- 6> 当服务或主机问题产生与解决时将告警发送给联系人（通过 Email、短信、用户定义方式）。
- 7> 具备定义事件句柄功能，它可以在主机或服务的事件发生时获取更多问题定位。
- 8> 自动的日志回滚。
- 9> 可以支持并实现对主机的冗余监控。
- 10> 可选的 WEB 界面用于查看当前的网络状态、通知和故障历史、日志文件等。

1.3 Nagios 体系结构



1.4 Nagios 相关资源

- 1> Nagios 官方网站: <http://www.nagios.org/>
- 2> Nagios 官方文档: <http://support.nagios.com/knowledgebase/officialdocs>
- 3> Nagios 中文文档: <http://nagios-cn.sourceforge.net/nagios-cn/>
- 4> Nagios 实用文档: <http://www.unixhot.com/pdf/monitor.pdf>

1.5 其它开源监控软件

- 1> <http://www.zenoss.com/>
- 2> <http://www.zabbix.com/>
- 3> <http://www.cacti.net/>
- 4> <http://ganglia.sourceforge.net/>
- 5> <http://munin-monitoring.org/>

第 2 章 Nagios 服务器端部署

2.1 下载所需的软件包

以下是 Nagios 部署需要的软件包，可以使用 wget 直接下载，建议将源码包存放在 /usr/local/src 目录下。

```
[root@Nagios-Server ~]# cd /usr/local/src
[root@Nagios-Server src]# wget (Nagios 主程序)
http://prdownloads.sourceforge.net/sourceforge/nagios/nagios-3.2.3.tar.gz
[root@Nagios-Server src]# wget (Nagios 插件)
http://prdownloads.sourceforge.net/sourceforge/nagiosplug/nagios-plugins-1.4.15.tar.gz
[root@Nagios-Server src]# wget (代理检测程序用来执行远程的 Nagios 插件)
http://prdownloads.sourceforge.net/sourceforge/nagios/nrpe-2.12.tar.gz
[root@Nagios-Server src]# wget (部署冗余和分布式 Nagios 的软件包)
http://prdownloads.sourceforge.net/sourceforge/nagios/nsca-2.7.2.tar.gz
[root@Nagios-Server src]# wget (导入当前和历史的监控数据到 MySQL 数据库)
http://prdownloads.sourceforge.net/sourceforge/nagios/ndoutils-1.4b9.tar.gz
```

2.2 安装环境准备

注意：建议配置 Yum 仓库来安装缺失的软件包。可以参 <http://www.unixhot.com/unixhot/1.htm> 来快速创建 Yum 仓库，也可使用网络 yum 仓库。

```
[root@Nagios-Server src]# yum install -y httpd php gcc glibc glibc-common gd gd-devel
[root@Nagios-Server src]# useradd nagios
#创建一个用户组名为 nagcmd 用于从 Web 接口执行外部命令。
[root@Nagios-Server src]# groupadd nagcmd
#添加 nagios 和 apache 用户到该组
[root@Nagios-Server src]# usermod -a -G nagcmd nagios
[root@Nagios-Server src]# usermod -a -G nagcmd apache
[root@Nagios-Server src]# passwd nagios
```

2.3 安装 Nagios 主程序

和普通源码包的安装类似，更多编译选项可以通过 `./configure --help` 查看。在安装过程中，每执行完一个操作，Nagios 会在末尾告诉你，下一步将要执行什么操作。

```
[root@Nagios-Server src]# tar zxvf nagios-3.2.3.tar.gz
[root@Nagios-Server src]# cd nagios-3.2.3
[root@Nagios-Server nagios-3.2.3]# ./configure --with-command-group=nagcmd \
--with-nagios-user=nagios \
--with-nagios-group=nagios
[root@Nagios-Server nagios-3.2.3]# make all
[root@Nagios-Server nagios-3.2.3]# make install
#生成 init 启动脚本
[root@Nagios-Server nagios-3.2.3]# make install-init
#设置相应的目录权限
[root@Nagios-Server nagios-3.2.3]# make install-commandmode
#生成模板配置文件
[root@Nagios-Server nagios-3.2.3]# make install-config
#生成 apache 配置文件
[root@Nagios-Server nagios-3.2.3]# make install-webconf
#为 apache 创建一个登陆用户，注意用户名是 nagiosadmin，该用户名是在 cgi.cfg 设置的默认用户名，关于 CGI 权限的设置。会在第 9 章给大家介绍。
注意：第一次添加用户用 -c 选项，以后再添加千万别在用这个选项了，会覆盖以前的所有用户。
[root@Nagios-Server nagios-3.2.3]# htpasswd -c /usr/local/nagios/etc/htpasswd.users
nagiosadmin
```

2.4 添加开机自动启动

为了让 Nagios 和 Apache 可以开机自动启动，我们把 Nagios 和 Apache 添加到启动管理程序中。

```
[root@Nagios-Server ~]# chkconfig --add nagios
[root@Nagios-Server ~]# chkconfig --level 3 nagios on
```

```
[root@Nagios-Server ~]# chkconfig --level 3 httpd on
```

2.5 修改 SELinux

Nagios 的访问页面作为 HTTP 类型被安装在服务器上，如果你打开了 SELinux，就需要修改相关设置，有两种方法可以让 Nagios 在 SELinux 安全模型下正常运行。

第一种最直接，关闭 SELinux，对于 SELinux 不是很熟悉的用户，请选择此。

```
[root@Nagios-Server ~]# cat /etc/sysconfig/selinux
```

SELINUX=disabled #确保是关闭的

第二种给打上 HTTP 的安全标签。

```
chcon -R -t httpd_sys_content_t /usr/local/nagios/sbin/
```

```
chcon -R -t httpd_sys_content_t /usr/local/nagios/share/
```

2.6 安装 Nagios 插件 nagios-plugins

在启动 Nagios 之前，我们还需要安装的是 Nagios 的 Plugins，这个包里面包含了很多 Nagios 用来执行检测的命令和脚本，是 Nagios 必须的软件包。

```
[root@Nagios-Server ~]# cd /usr/local/src
```

```
[root@Nagios-Server src]# tar zxvf nagios-plugins-1.4.15.tar.gz
```

```
[root@Nagios-Server src]# cd nagios-plugins-1.4.14
```

```
[root@Nagios-Server nagios-plugins-1.4.15]# ./configure --prefix=/usr/local/nagios
```

```
--with-nagios-user=nagios --with-nagios-group=nagios
```

```
[root@Nagios-Server nagios-plugins-1.4.14]# make && make install
```

2.7 启动 Nagios

如果在安装过程中，没有出现错误，我们就可以正常启动 Nagios 和 Apache 了，如果出现了错误，请根据错误提示，检测安装，一般错误时由于依赖的库文件没有安装。

```
[root@Nagios-Server ~]# /etc/init.d/httpd start
```

Starting httpd:

[OK]

```
[root@Nagios-Server ~]# /etc/init.d/nagios start
```

Starting nagios: done.

2.8 登陆 Nagios

接着可以使用 `http://Nagios 服务器 IP/nagios/`

并输入你设置的账号 `nagiosadmin` 和密码登陆 Nagios, Nagios 会默认添加本机的监控, 在这里不再截图。

第 3 章 Nagios 配置文件简介

在开始使用 Nagios 之前, 我们一定要了解 Nagios 的各个配置文件, 网上有很多网友称 Nagios 为 (难够死), 正是因为它众多的配置文件, 而且配置文件和配置文件之间的关系密切, 只要其中一个配置文件出现错误, 就会导致 Nagios 无法正常工作。

3.1 配置文件概述

如果按照本文的步骤, 安装完成后, 配置文件在安装时放在了 `/usr/local/nagios/etc/` 目录下。

```
[root@Nagios-Server ~]# ls -l /usr/local/nagios/etc/
```

```
total 88
```

<code>-rw-rw-r-- 1 nagios nagios 11408 May 29 02:09 cgi.cfg</code>	#CGI 配置文件
<code>-rw-r--r-- 1 root root 22 May 29 02:13 htpasswd.users</code>	#Apache 验证密码文件
<code>-rw-rw-r-- 1 nagios nagios 43776 May 29 02:09 nagios.cfg</code>	#Nagios 主配置文件
<code>drwxrwxr-x 2 nagios nagios 4096 May 29 02:09 objects</code>	#监控对象定义文件目录
<code>-rw-rw---- 1 nagios nagios 1340 May 29 02:09 resource.cfg</code>	#Nagios 资源配置文件

3.2 主配置文件 `nagios.cfg`

主配置文件包括了一系列的设置, 它们会影响 Nagios 守护进程。不仅是 Nagios 守护进程要使用主配置文件, CGI 程序组模块也需要, 在主配置文件里, 我们可以指定主机、主机组、服务、服务组、命令、通知人、通知人组和时间周期等配置文件的位置。

主配置文件参数众多，在这里只修改小部分，确保 Nagios 可以正常运行。

```
[root@Nagios-Server ~]# vi /usr/local/nagios/etc/nagios.cfg
# You can specify individual object config files as shown below:
cfg_file=/usr/local/nagios/etc/objects/commands.cfg    #命令配置文件
cfg_file=/usr/local/nagios/etc/objects/contacts.cfg    #联系人配置文件
cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg #时间周期配置文件
cfg_file=/usr/local/nagios/etc/objects/templates.cfg   #模板配置文件
cfg_file=/usr/local/nagios/etc/objects/contactgroups.cfg #增加联系组定义文件
cfg_file=/usr/local/nagios/etc/objects/hosts.cfg       #增加主机配置文件
cfg_file=/usr/local/nagios/etc/objects/hostgroups.cfg  #增加主机组配置文件
cfg_file=/usr/local/nagios/etc/objects/services.cfg   #增加服务配置文件
cfg_file=/usr/local/nagios/etc/objects/servicegroups.cfg #增加服务组配置文件
# Definitions for monitoring the local (Linux) host
#cfg_file=/usr/local/nagios/etc/objects/localhost.cfg #注释掉（默认的监控本机的配置）
```

3.3 资源配置文件 resource.cfg

资源文件可以保存用户自定义的宏。资源文件的一个主要用处是用于保存一些敏感的配置信息如系统口令等不能让 CGI 程序模块获取到的东西。

3.4 对象定义文件

对象定义文件是管理员经常需要修改的配置文件，用于定义主机、服务、主机组、服务组、联系人、联系人组、命令等等，也就是要定义监控的对象和监控的方法。

对象定义文件有两种基本写法，例如监控一台 Linux 服务器。

方法一：将该主机定义写入主机定义文件，将要监控的服务写到服务定义文件中，前提是在 Nagios 的主配置文件里需要定义这两个配置文件。

方法二：将该主机的配置写入到一个如【主机名】.cfg 文件里，然后再 Nagios 的主配置文件里定义这个配置文件。

这两种方法的选择要根据实际情况，方法一由于主机数的增加会增加配置文件的管理难度，文件内容多，容易出现配置错误等问题。

方法二由于是每个监控主机一个配置文件，这样虽然清晰明了，但对于想结构化管理的用户不太方便。

3.5 CGI 配置文件 `cgi.cfg`

CGI 配置文件包含了一系列的设置，它们会影响 CGI 程序模块。还有一些保存在主配置文件中，因此 CGI 程序会知道你是如何配置的 Nagios 并且在哪里保存了对象定义。最实际的例子就是，如果你想建立一个只有查看报警权限的用户，或者只有查看其中一些服务器或者服务状态的权限，通过修改 `cgi.cfg` 可以灵活的控制 web 访问端的权限。

3.6 通过实例介绍配置文件

单纯的讲解配置文件，是让人很头疼的一件事，下面就通过监控一台服务器是否存活的实例来介绍相关配置文件的用法，注意 `nagios.cfg` 配置文件在 3.2 章节已经修改好了，下面不再详述。

3.6.1 主机定义文件

定义你要监控的对象，这里定义的“`host_name`”被应用到其它的所有配置文件中，这个是我们配置 Nagios 必须修改的配置文件。

```
[root@Nagios-Server ~]# vi /usr/local/nagios/etc/objects/hosts.cfg
```

```
define host{
    host_name                Nagios-Server  #设置主机的名字，该名字会出现在
hostgroups.cfg 和 services.cfg 中。注意，这个名字可以不是该服务器的主机名。
    alias                    Nagios Server  #别名
    address                  192.168.140.141  #主机的 IP 地址
    check_command             check-host-alive  #检查使用的命令，需要在命令定
义文件定义，默认是定义好的。
    check_interval            5                #检测的时间间隔
    retry_interval            1                #检测失败后重试的时间间隔
```

```

max_check_attempts      5      #最大重试次数
check_period             24x7   #检测的时段
process_perf_data        0
retain_nonstatus_information 0
contact_groups           sagroup #需要通知的联系组
notification_interval     30     #通知的时间间隔
notification_period       24x7   #通知的时间段
notification_options      d,u,r   #通知的选项
                                #w—报警(warning), u—未知(unknown)
                                #c—严重(critical), r—从异常情况恢复正常
}

define host{
    host_name             Nagios-Client
    alias                  Nagios Client
    address                192.168.140.143
    check_command           check-host-alive
    check_interval         5
    retry_interval         1
    max_check_attempts     5
    check_period           24x7
    process_perf_data       0
    retain_nonstatus_information 0
    contact_groups         sagroup
    notification_interval   30
    notification_period     24x7
    notification_options    d,u,r
}

```

3.6.2 主机组定义文件

主机组定义文件，可以方便的将相同功能或者在应用上相同的服务器添加到一个主机组里，在WEB 界面可以通过 HOST Group 方便的查看该组主机的状态信息。

将刚才定义的两个主机加入到主机组中，针对生产环境就像把所有的 MySQL 服务器加到一个 MySQL 主机组里，将 Oracle 服务器加到一个 Oracle 主机组里，方便管理和查看，可以配置多个组。

```
[root@Nagios-Server ~]# vi /usr/local/nagios/etc/objects/hostgroups.cfg

define hostgroup {
    hostgroup_name Nagios-Example #主机组名字
    alias           Nagios Example #主机组别名
    members         Nagios-Server,Nagios-Client #主机组成员，用逗号隔开
}
```

3.6.3 服务定义文件

服务定义文件定义你需要监控的对象的服务，比如本例为检测主机是否存活，在后面会讲到如何监控其它服务，比如服务器负载、内存、磁盘等。

```
[root@Nagios-Server ~]# vi /usr/local/nagios/etc/objects/services.cfg

define service {
    host_name           Nagios-Server #hosts.cfg 定义的主机名称
    service_description check-host-alive #服务描述
    check_period        24x7 #检测的时间段
    max_check_attempts  4 #最大检测次数
    normal_check_interval 3
    retry_check_interval 2
    contact_groups      sagroup #发生故障通知的联系人组
    notification_interval 10
    notification_period 24x7 #通知的时间段
    notification_options w,u,c,r
    check_command        check-host-alive
}
```

```
define service {
```

```

host_name          Nagios-Client
service_description check-host-alive
check_period        24x7
max_check_attempts  4
normal_check_interval 3
retry_check_interval 2
contact_groups      sagroup
notification_interval 10
notification_period 24x7
notification_options w, u, c, r
check_command        check-host-alive
}

```

3.6.4 服务组定义文件

和主机组一样，我们可以按需将相同的服务放入一个服务组，这样有规律的分类，便于我们在 WEB 端查看。

```

[root@Nagios-Server ~]# vi /usr/local/nagios/etc/objects/servicegroups.cfg

define servicegroup{
    servicegroup_name    Host-Alive #组名
    alias                 Host Alive #别名设置
    members               Nagios-Server, check-host-alive, Nagios-Client, check-host-alive
}

```

#组员的设置，主机名，服务名的方式表示，中间用逗号分隔。

3.6.5 联系人定义文件

定义发生故障时，需要通知的联系人信息。默认安装完成后，该配置文件已经存在，而且该文件不仅定义了联系人，也定义了联系人组，为了条理化的规划，我们把联系人定义放在 `contacts.cfg` 文件里，把联系人组放在 `contactgroups.cfg` 文件中。

```

[root@Nagios-Server ~]# mv /usr/local/nagios/etc/objects/contacts.cfg

```

/usr/local/nagios/etc/objects/contacts.cfg.bak #先将该文件备份。

```
[root@Nagios-Server ~]# vi /usr/local/nagios/etc/objects/contacts.cfg
```

```
define contact{
    contact_name nagiosadmin    #联系人的名字
    alias Nagios Admin        #别名
    service_notification_period 24x7    #服务报警的时间段
    host_notification_period 24x7        #主机报警的时间段
    service_notification_options w,u,c,r #就是在这四种情况下报警。
    host_notification_options d,u,r     #同上。
```

#服务报警发消息的命令，在 command.cfg 中定义。

```
service_notification_commands notify-service-by-email
```

#服务报警发消息的命令，在 command.cfg 中定义。

```
host_notification_commands notify-host-by-email
```

```
email 13717514\*\*\*@139.com #定义邮件地址，也就是接收报警邮件地址。
```

```
}
```

3.6.6 联系人组定义文件

联系人组定义文件在实际应用中很有好处，我们可以把报警信息分级别，报联系人分级别存放在联系人组里面。例如：当发生一些警告信息的情况下，只发邮件给系统工程师联系人组即可，但是当发生重大问题，比如主机宕机了，可以发给领导联系人组。

```
[root@Nagios-Server ~]# vi /usr/local/nagios/etc/objects/contactgroups.cfg
```

```
define contactgroup{
    contactgroup_name    sagroup            #组名
    alias                Nagios Administrators #别名
    members              nagiosadmin        #联系人组成员
}
```

3.6.7 命令定义文件

命令定义文件是 Nagios 中很重要的配置文件，所有在 hosts.cfg 还是 services.cfg 使用的命令都必须在命令定义文件中定义才能使用。默认情况下，范例配置文件已经配置好了日常需要使用

的命令，所以一般不做修改。

```
[root@Nagios-Server ~]# vim /usr/local/nagios/etc/objects/commands.cfg

define command{

    command_name    notify-host-by-email

    command_line    /usr/bin/printf "%b" "***** Nagios *****\n\nNotification
Type: $NOTIFICATIONTYPE$\nHost: $HOSTNAME$\nState: $HOSTSTATE$\nAddress:
$HOSTADDRESS$\nInfo: $HOSTOUTPUT$\n\nDate/Time: $LONGDATETIME$\n" | /bin/mail -s "**
$NOTIFICATIONTYPE$ Host Alert: $HOSTNAME$ is $HOSTSTATE$ **" $CONTACTEMAIL$

}

# 'notify-service-by-email' command definition

define command{

    command_name    notify-service-by-email

    command_line    /usr/bin/printf "%b" "***** Nagios *****\n\nNotification
Type: $NOTIFICATIONTYPE$\n\nService: $SERVICEDESC$\nHost: $HOSTALIAS$\nAddress:
$HOSTADDRESS$\nState: $SERVICESTATE$\n\nDate/Time: $LONGDATETIME$\n\nAdditional
Info:\n\n$SERVICEOUTPUT$" | /bin/mail -s "** $NOTIFICATIONTYPE$ Service Alert:
$HOSTALIAS$/$SERVICEDESC$ is $SERVICESTATE$ **" $CONTACTEMAIL$

}
```

(省略) 以上定义了，我们在联系人定义文件 (contacts.cfg) 用到的两个命令，也就是 Nagios 用来发报警邮件的命令。

3.6.8 时间段定义文件 (timeperiods.cfg)

我们在检测、通知、报警的时候都需要定义时间段，默认都是使用 7x24，这也是默认配置文件里配置好的，如果你需要周六日不做检测，或者在制定的维护时间不做检测，都可以在该时间段定义文件定义好，这样固定维护的时候，就不会为大量的报警邮件或者短信烦恼。

```
[root@Nagios-Server ~]# vim /usr/local/nagios/etc/objects/timeperiods.cfg

define timeperiod{

    timeperiod_name 24x7

    alias            24 Hours A Day, 7 Days A Week

    sunday           00:00-24:00
```

```

monday      00:00-24:00
tuesday     00:00-24:00
wednesday   00:00-24:00
thursday    00:00-24:00
friday      00:00-24:00
saturday    00:00-24:00
}

```

（省略）以上为 7x24 的配置，读者可以根据业务需求来更改。

3.6.9 启动 Nagios

1> 修改配置文件所有者

```
[root@Nagios-Server ~]# chown -R nagios:nagios /usr/local/nagios/etc/objects/
```

2> 检测配置是否正确

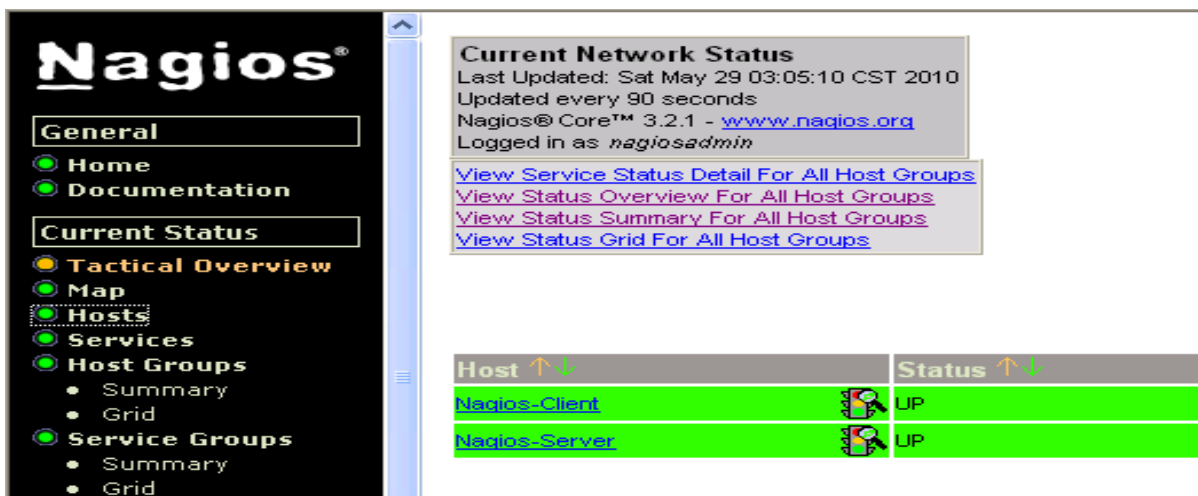
```
[root@Nagios-Server ~]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

如果配置错误，会给出相应的报错信息，可以根据信息查找，注意，如果配置文件中不可见字符也可以导致配置错误。

3> 重载 Nagios

```
[root@Nagios-Server ~]# /etc/init.d/nagios reload
```

3.6.10 登陆验证配置



The screenshot shows the Nagios web interface. On the left is a navigation menu with sections: General (Home, Documentation), Current Status (Tactical Overview, Map, Hosts, Services, Host Groups, Service Groups), and a sidebar with icons. The main content area displays the 'Current Network Status' with the following information:

- Current Network Status**
- Last Updated: Sat May 29 03:05:10 CST 2010
- Updated every 90 seconds
- Nagios® Core™ 3.2.1 - www.nagios.org
- Logged in as *nagiosadmin*

Below this status box are four links:

- [View Service Status Detail For All Host Groups](#)
- [View Status Overview For All Host Groups](#)
- [View Status Summary For All Host Groups](#)
- [View Status Grid For All Host Groups](#)

At the bottom, there is a table showing the status of hosts:

Host ↑↓	Status ↑↓
Nagios-Client	UP
Nagios-Server	UP

第 4 章 Nagios 外部监控

4.1 Nagios 监控的方式:

可以把 Nagios 的服务器监控分为两个部分:

第一部分是主机外监控, 比如: 主机是否存活, WEB 服务是否正常, MySQL 服务是否正常等内容, 在主机外通过访问其端口即可得知。这些监控命令在安装 **nagios-plugins** 时已经生成了, 在 `/usr/local/nagios/libexec` 目录下。

第二部分是主机内监控, 比如: 要监控服务器的进程、磁盘使用等功能。这些功能的实现要依靠 **nrpe** 了, **nrpe** 的工作模式是 C/S 模式, 在被监控主机中, 开启 **nrpe** 监听, 当收到监控服务器上所发出的检测命令, 让它检查该服务器上的硬盘使用信息时, 它就会执行, 并把信息传回监控服务器, 用一个不太恰当的比喻, 就是木马的工作模式。所以要监控远程的 Linux 服务器, 需要在监控端和被监控端都安全 NRPE 这个外部构建包。

4.2 Nagios 插件的使用。

外部监控最简单的理解就是不通过代理程序, Nagios 主程序直接执行相关的 **plugin** 来获得数据, 比如监控 WEB 服务器, 获得该服务器的响应时间。

当我们安装完成 **nagios-plugins** 后在下面目录可以看到安装的各种检测插件:

```
[root@Nagios-Server ~]# cd /usr/local/nagios/libexec/
```

```
[root@Nagios-Server libexec]# ls
```

```
check_apd      check_dhcp      check_dummy     check_icmp      check_ircd
check_log      check_nntp      check_ntp_time  check_pop       check_simap    check_swap
check_users    utils.sh ..... (省略)
```

想要获得插件的使用说明, 最简单的方法就是执行它, 你会发现, 有详细的选项可以选择。

```
[root@Nagios-Server libexec]# ./check_http
```

```
check_http: Could not parse arguments
```

```
Usage: check_http -H <vhost> [-I <IP-address> [-u <uri>] [-p <port>]
```

```
[-w <warn time>] [-c <critical time>] [-t <timeout>] [-L]
```

```
[-a auth] [-f <ok | warn | critical | follow | sticky | stickyport>]
```



```
[-e <expect>] [-s string] [-l] [-r <regex> | -R <case-insensitive regex>]  
[-P string] [-m <min_pg_size>:<max_pg_size>] [-4|-6] [-N] [-M <age>]  
[-A string] [-k string] [-S] [-C <age>] [-T <content-type>] [-j method]
```

4.3 Nagios 监控 Web 服务

Web 服务在我们工作中最常见的需要监控的对象了,我们需要监控一个 Web 服务器的响应时间,例如: 如果超过5秒就报警, 超过10秒就报严重错误。并发送报警邮件给管理员。

在上一小节我们已经执行了 `check_http` 这个插件并获得了使用方法, 在这里先简单介绍下该插件常用的几个参数:

- H 用于虚拟主机, 并可以增加端口。例如, `unixhot.com:10000`
- I IP 地址
- S 基于 SSL 连接。默认端口为443
- u URL 地址
- w 警告响应时间, 响应时间超过指定时间则显示 **WARNING**
- c 临界报警响应时间, 响应时间超过设定时间则显示 **CRITICAL**
- p 端口设定
- e 监测服务器反馈的第一行(状态)信息是否包含匹配字符
- N 不读取 **body** 部分, 在读取 **header** 信息
- t timeout 时间值
- v 显示服务器反馈的所有信息
- r 正则表达式匹配反馈页信息
- R 同上, 但可以不考虑大小写问题
- invert-regex 反馈信息中包含字符则报警
- L 反馈信息包含 URL link
- a 对需要基础认证的, 提供用户名和密码, 获取反馈值

以下是我监控 `unixhot.com` 的输出:

```
[root@Nagios-Server ~]# cd /usr/local/nagios/libexec/  
[root@Nagios-Server libexec]# ./check_http -H www.unixhot.com
```

HTTP OK: HTTP/1.1 200 OK - 24039 bytes in 0.470 second response time |time=0.470371s;;;0.000000
size=24039B;;;0

如果要做到开始提到的：如果超过 5 秒就报警，超过 10 秒就报严重错误。并发送报警邮件给管理员，大概分为三步。

4.3.1 第一步 添加命令

确保 `commands.cfg` 文件里定义了该命令，注意，所有需要 Nagios 执行的命令必须在 `commands` 里面定义，该文件默认已经定义了 `check_http` 命令，如下：

```
define command{  
    command_name    check_http  
    command_line     $USER1$/check_http -I $HOSTADDRESS$ $ARG1$  
}
```

4.3.2 第二步 添加主机

`[root@Nagios-Server ~]# vim /usr/local/nagios/etc/objects/hosts.cfg`

在末行添加以下主机定义：

```
define host{  
    host_name          UnixHot  
    alias              UnixHot  
    address            192.168.140.141  
    check_command      check-host-alive  
    check_interval     5  
    retry_interval     1  
    max_check_attempts 5  
    check_period       24x7  
    process_perf_data  0  
    retain_nonstatus_information 0  
    contact_groups     sagroup
```

```
notification_interval      30
notification_period        24x7
notification_options       d,u,r
}
```

你也可以把该主机加入到某个分组，这里不再添加。

4.3.3 第三步 添加服务

```
[root@Nagios-Server ~]# vim /usr/local/nagios/etc/objects/hosts.cfg
```

在末行添加以下服务定义：

```
define service {
    host_name          UnixHot
    service_description Check Http
    check_period        24x7
    max_check_attempts  4
    normal_check_interval 3
    retry_check_interval 2
    contact_groups      sagroup
    notification_interval 10
    notification_period  24x7
    notification_options w,u,c,r
    check_command        check_http!-H www.unixhot.com!5!10
}
```

你可以把该 Web 服务添加到 Web 服务组，这里不再添加。

4.3.4 第四步 重载配置并登陆测试

```
[root@Nagios-Server ~]# /etc/init.d/nagios reload
```

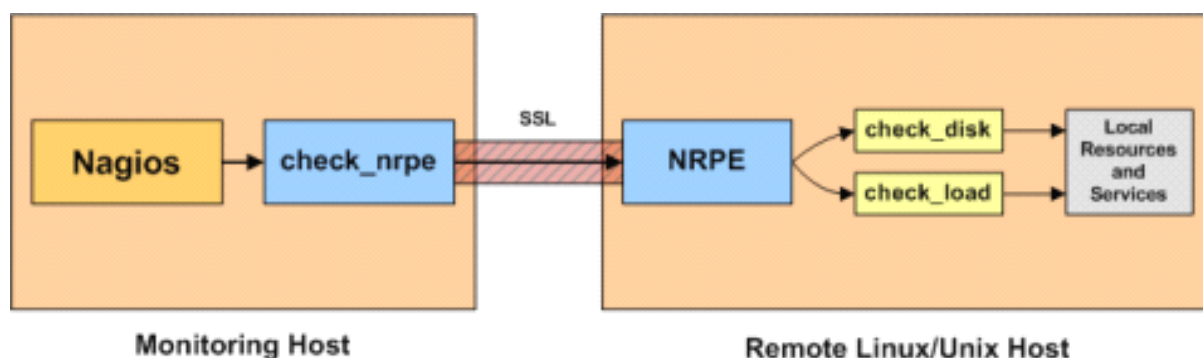
现在可以登陆测试，注意，刚添加的服务需要等待一段时间，才可以监控。

第 5 章 Nagios 监控远程 Linux 服务器

在上一章,我们使用 Nagios 的 Plugins 直接从被监控主机的外部获得相关监控对象的状态信息,这一章我们将通过 NRPE, 实现主机内部的监控, 通过 NRPE 将主机内部监控的数据发送给 Nagios 监控端。

NRPE 是一个可在远程 Linux/Unix 主机上执行的插件的外部构件包。如果你需要监控远程的主机上的本地资源或属性, 如磁盘利用率、CPU 负荷、内存利用率等时是很有用的。

注意: NRPE 是使用的 ssl 进行通信的, 所以安全性是可以保证的, 系统工程师不必为这种代理模式的安全性有太多的顾虑。



5.1 Nagios 服务器的部署

5.1.1 安装 NRPE

```
[root@Nagios-Server ~]# cd /usr/local/src/
[root@Nagios-Server src]# tar zxvf nrpe-2.12.tar.gz
[root@Nagios-Server src]# cd nrpe-2.12
[root@Nagios-Server nrpe-2.12]# ./configure && make all
[root@Nagios-Server nrpe-2.12]# make install-plugin
[root@Nagios-Server nrpe-2.12]# make install-daemon
[root@Nagios-Server nrpe-2.12]# make install-daemon-config
[root@Nagios-Server nrpe-2.12]# make install-xinetd
```

5.1.2 配置 NRPE

```
[root@Nagios-Server nrpe-2.12]# vi /etc/xinetd.d/nrpe
# default: on
# description: NRPE (Nagios Remote Plugin Executor)
service nrpe
{
    flags                = REUSE
    socket_type          = stream
    port                 = 5666
    wait                 = no
    user                  = nagios
    group                 = nagios
    server                = /usr/local/nagios/bin/nrpe
    server_args           = -c /usr/local/nagios/etc/nrpe.cfg --inetd
    log_on_failure        += USERID
    disable               = no #注意修改为 no
    only_from             = 127.0.0.1 192.168.140.143
#注意：在 only_from 添加要监控的主机的 IP 地址，中间以空格隔开。
}
```

5.1.3 添加服务端口

```
[root@Nagios-Server nrpe-2.12]# vi /etc/services
#在文件末尾添加，如果不添加端口，nrpe 将不能正常运行
nrpe      5666/tcp                                #nrpe
```

5.1.4 重启服务并验证安装

```
[root@Nagios-Server ~]# /etc/init.d/xinetd restart
[root@Nagios-Server ~]# netstat -na | grep 5666
```

```
tcp          0          0 0.0.0.0:5666      0.0.0.0:*
```

5.1.5 修改命令定义文件

由于命令定义文件里默认没有 check_nrpe 命令，所以需要先定义后使用。

```
[root@Nagios-Server ~]# vim /usr/local/nagios/etc/objects/commands.cfg
```

#nrpe set 在文件末尾添加下面命令定义

```
define command{  
    command_name    check_nrpe  
    command_line    /usr/local/nagios/libexec/check_nrpe -H $HOSTADDRESS$ -c $ARG1$  
}
```

5.2 受监控 Linux 服务器部署

5.2.1 添加用户名

```
[root@Nagios-Client ~]# useradd -s /sbin/nologin nagios
```

5.2.2 安装 Nagios 的插件 nagios-plugin

```
[root@Nagios-Client ~]# cd /usr/local/src
```

```
[root@Nagios-Client src]# wget
```

<http://prdownloads.sourceforge.net/sourceforge/nagiosplug/nagios-plugins-1.4.15.tar.gz>

```
[root@Nagios-Client src]# wget
```

<http://prdownloads.sourceforge.net/sourceforge/nagios/nrpe-2.12.tar.gz>

```
[root@Nagios-Client src]# tar zxvf nagios-plugins-1.4.15.tar.gz
```

```
[root@Nagios-Client src]# cd nagios-plugins-1.4.15
```

```
[root@Nagios-Client nagios-plugins-1.4.15]# ./configure
```

注意：如果你 RedHat AS4 版本的服务器要添加--enable-redhat-pthread-workaround 选项。

```
[root@Nagios-Client nagios-plugins-1.4.15]# make && make install
```

5.2.3 安装 NRPE 代理检测程序

同服务器端的安装，不同的是修改 xinetd 配置文件的时候，权限的设置。

1> 安装 nrpe

```
[root@Nagios-Client ~]# cd /usr/local/src/
[root@Nagios-Client ~]# tar zxvf nrpe-2.12.tar.gz
[root@Nagios-Client ~]# cd nrpe-2.12
[root@Nagios-Client nrpe-2.12]# ./configure && make all
[root@Nagios-Client nrpe-2.12]# make install-plugin
[root@Nagios-Client nrpe-2.12]# make install-daemon
[root@Nagios-Client nrpe-2.12]# make install-daemon-config
[root@Nagios-Client nrpe-2.12]# make install-xinetd
```

2> 配置 NRPE

```
[root@Nagios-Client nrpe-2.12]# vi /etc/xinetd.d/nrpe
# default: on
# description: NRPE (Nagios Remote Plugin Executor)
service nrpe
{
    flags                = REUSE
    socket_type          = stream
    port                 = 5666
    wait                 = no
    user                  = nagios
    group                = nagios
    server                = /usr/local/nagios/bin/nrpe
    server_args           = -c /usr/local/nagios/etc/nrpe.cfg --inetd
    log_on_failure       += USERID
```

```
disable          = no
only_from        = 127.0.0.1 192.168.140.141
```

#注意：在 only_from 添加 Nagios 监控服务器的 IP 地址。

```
}
```

3> 添加服务端口

```
[root@Nagios-Client nrpe-2.12]# vi /etc/services
```

#在文件末尾添加

```
nrpe      5666/tcp                                #nrpe
```

4> 重启服务并验证安装

```
[root@Nagios-Client ~]# /etc/init.d/xinetd restart
```

```
[root@Nagios-Client ~]# netstat -na | grep 5666
```

```
tcp        0      0 0.0.0.0:5666        0.0.0.0:*
```

5.2.4 修改权限

```
[root@Nagios-Client ~]# chown -R nagios:nagios /usr/local/nagios/
```

5.3 修改受控端 NRPE 配置文件

修改受控端的 NRPE 配置文件时确定监控的内容和命令，然后再修改 Nagios 服务器上的服务定义文件中引用来实现对客户端的监控。

```
[root@Nagios-Client ~]# vi /usr/local/nagios/etc/nrpe.cfg
command[check_load]=/usr/local/nagios/libexec/check_load -w 15,10,5 -c 30,25,20
command[check_]=/usr/local/nagios/libexec/check_disk -w 20% -c 10%
-p /dev/mapper/VolGroup00-LogVol00 #我的Client端根分区时LVM逻辑卷
command[check_zombie_procs]=/usr/local/nagios/libexec/check_procs -w 5 -c 10 -s Z
command[check_total_procs]=/usr/local/nagios/libexec/check_procs -w 150 -c 200
command[check_swap]=/usr/local/nagios/libexec/check_swap -w 20% -c 10%
```


5.4 修改 Nagios 服务器服务定义文件

```
[root@Nagios-Server ~]# vi /usr/local/nagios/etc/objects/services.cfg
```

添加以下内容:

```
define service {
    host_name          Nagios-Client
    service_description check-users
    check_period        24x7
    max_check_attempts 4
    normal_check_interval 3
    retry_check_interval 2
    contact_groups      sagroup
    notification_interval 10
    notification_period  24x7
    notification_options w,u,c,r
    check_command         check_nrpe!check_users
}

define service {
    host_name          Nagios-Client
    service_description check-load
    check_period        24x7
    max_check_attempts 4
    normal_check_interval 3
    retry_check_interval 2
    contact_groups      sagroup
    notification_interval 10
    notification_period  24x7
    notification_options w,u,c,r
    check_command         check_nrpe!check_load
}
```

```
define service {  
    host_name            Nagios-Client  
    service_description  check-zombie-procs  
    check_period         24x7  
    max_check_attempts   4  
    normal_check_interval 3  
    retry_check_interval 2  
    contact_groups       sagroup  
    notification_interval 10  
    notification_period   24x7  
    notification_options  w, u, c, r  
    check_command         check_nrpe!check_zombie_procs  
}  
  
define service {  
    host_name            Nagios-Client  
    service_description  check-/  
    check_period         24x7  
    max_check_attempts   4  
    normal_check_interval 3  
    retry_check_interval 2  
    contact_groups       sagroup  
    notification_interval 10  
    notification_period   24x7  
    notification_options  w, u, c, r  
    check_command         check_nrpe!check_  
}
```

```
define service {
```

```

host_name Nagios-Client
service_description check-swap
check_period 24x7
max_check_attempts 4
normal_check_interval 3
retry_check_interval 2
contact_groups sagroup
notification_interval 10
notification_period 24x7
notification_options w, u, c, r
check_command check_nrpe!check_swap
}

```

5.5 登陆验证配置

5.5.1 查看配置文件是否正确

```
[root@Nagios-Server ~]# /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

5.5.2 重新加载配置文件

```
[root@Nagios-Server ~]# /etc/init.d/nagios reload
```

5.5.3 用 IE 浏览器登陆验证

Host ↑↓	Service ↑↓	Status ↑↓	Last Check ↑↓	Duration ↑↓	Attempt ↑↓	Status Information
Nagios-Client	check-/	OK	05-29-2010 04:26:36	0d 0h 1m 12s	1/4	DISK OK - free space: / 7017 MB (79% inode=96%);
	check-host-alive	OK	05-29-2010 04:26:31	0d 1h 28m 17s	1/4	PING OK - Packet loss = 0%, RTA = 0.34 ms
	check-load	OK	05-29-2010 04:27:27	0d 0h 0m 21s	1/4	OK - load average: 0.05, 0.14, 0.11
	check-swap	OK	05-29-2010 04:25:19	0d 0h 2m 29s	1/4	SWAP OK - 100% free (511 MB out of 511 MB)
	check-users	OK	05-29-2010 04:26:10	0d 0h 1m 38s	1/4	USERS OK - 2 users currently logged in
	check-zombie-procs	OK	05-29-2010 04:27:02	0d 0h 0m 46s	1/4	PROCS OK: 0 processes with STATE = Z
Nagios-Server	check-host-alive	OK	05-29-2010 04:25:01	0d 1h 26m 47s	1/4	PING OK - Packet loss = 0%, RTA = 0.09 ms

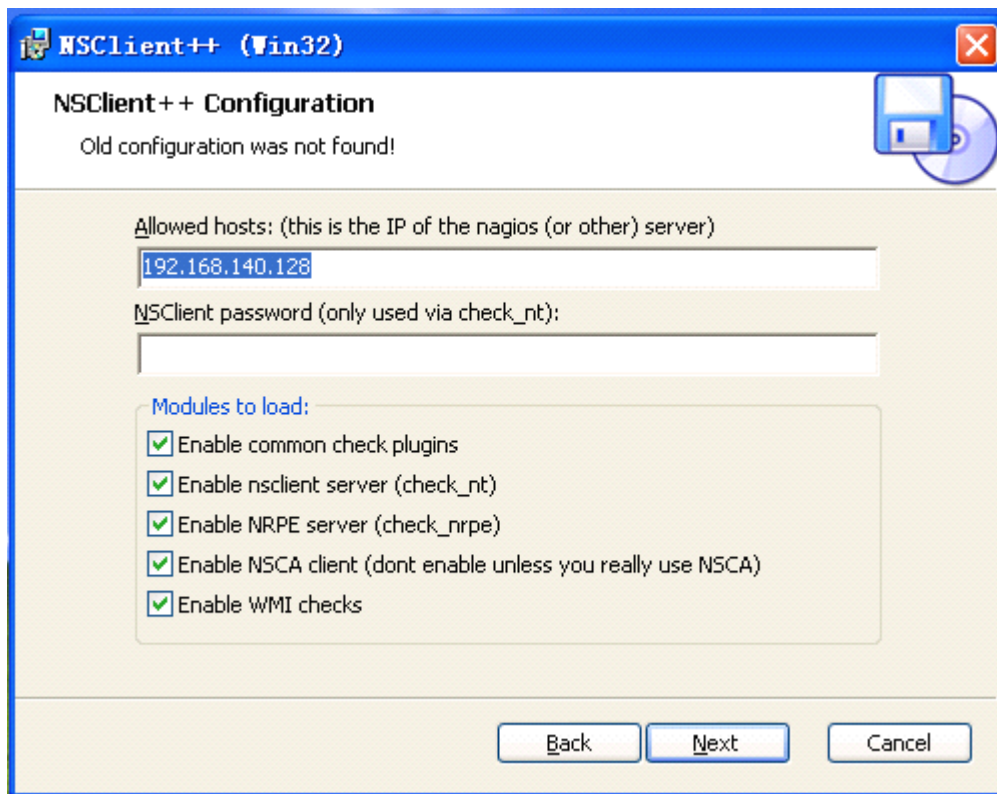
第 6 章 Nagios 监控远程 Windows 服务器

Nagios 监控 Windows 服务器使用的是 NSClient++ 程序，该程序完成的功能和 nrpe 类似。这里不过多的介绍。

6.1 在受监控 Windows 服务器部署

6.2 下载 NSClient++

<http://sourceforge.net/projects/nscplus/>



6.3 修改 Nagios 服务器配置文件

在这里使用一个配置文件来做，也就是把主机，服务写到一个配置文件里。

```
[root@Nagios-Server ~]# vi /usr/local/nagios/etc/nagios.cfg
```

添加:

```
cfg_file=/usr/local/nagios/etc/objects/Windows-Client.cfg
```

```
[root@Nagios-Server ~]# vi /usr/local/nagios/etc/objects/windows-client.cfg
```

```
define host{  
    host_name                Windows-Client  
    alias                    Nagios Server  
    address                  172.16.0.168  
    check_command             check-host-alive  
    check_interval           5  
    retry_interval           1  
    max_check_attempts       5  
    check_period             24x7  
    process_perf_data        0  
    retain_nonstatus_information 0  
    contact_groups           sagroup  
    notification_interval    30  
    notification_period      24x7  
    notification_options     d,u,r  
}
```

```
define service {  
    host_name                Windows-Client  
    service_description      Check-Memory  
    check_period             24x7  
    max_check_attempts       4  
    normal_check_interval    3  
    retry_check_interval     2  
    contact_groups           sagroup  
    notification_interval    10  
    notification_period      24x7  
    notification_options     w,u,c,r
```

```

        check_command          check_nt!MEMUSE!-w 80 -c 90
    }

define service {
    host_name                  Windows-Client
    service_description        Check-CPU-Load
    check_period                24x7
    max_check_attempts         4
    normal_check_interval      3
    retry_check_interval       2
    contact_groups              sagroup
    notification_interval      10
    notification_period         24x7
    notification_options        w,u,c,r
    check_command               check_nt!CPULOAD!-l 5,80,90
}

define service {
    host_name                  Windows-Client
    service_description        Check-Disk-C
    check_period                24x7
    max_check_attempts         4
    normal_check_interval      3
    retry_check_interval       2
    contact_groups              sagroup
    notification_interval      10
    notification_period         24x7
    notification_options        w,u,c,r
    check_command               check_nt!USEDISKSPACE!-l c -w 80 -c 90
}

define service {
    host_name                  Windows-Client

```

```

service_description      Check-Disk-D
check_period              24x7
max_check_attempts       4
normal_check_interval     3
retry_check_interval      2
contact_groups            sagroup
notification_interval     10
notification_period       24x7
notification_options      w,u,c,r
check_command             check_nt!USEDISKSPACE!-l d -w 80 -c 90
}

```

6.4 登陆验证配置

```
[root@Nagios-Server ~]# /etc/init.d/nagios restart
```

第 7 章 Nagios 插件开发

在以上的章节中我们通过使用 Nagios 自带的检测插件，实现了一些基本的监控功能，这个插件对于我们应用或许远远不够，那么就需要运维工程师有 Nagios 插件的编写能力，至于用什么脚本语言，至少我认为 SHELL 是一名运维工程师必须精通的语言，推荐“高级 BASH SHELL 编程”一书。

在本章，将通过编写一个监控 Linux 系统内存的检测插件和 Nagios 监控 MySQL 复制的一个实例来介绍，系统管理员如何根据业务应用编写 Nagios 支持的插件检测检测脚本。

7.1 如何编写检测脚本

编写 Nagios 的插件可以使用 PERL、SHELL、Python 等脚本语言，管理员可以根据自身情况选择一种来使用。

7.2 Nagios 插件编写条件

Nagios 插件的脚本或执行程序必须(至少)要做两件事:

一是退出时给出几种可能的返回值中的一个。

Nagios 用插件的返回值来生成主机或服务的状态。下表里列出了合法的返回值以及对应的服务或主机状态。

插件返回值	服务状态	主机状态
0	正常(OK)	运行(UP)
1	告警(WARNING)	运行(UP)或宕机(DOWN)/不可达(UNREACHABLE)*
2	紧急(CRITICAL)	宕机(DOWN)/不可达(UNREACHABLE)
3	未知(UNKNOWN)	宕机(DOWN)/不可达(UNREACHABLE)

二是至少要给出一条输出内容到标准输出设备(STDOUT)。

例如: DISK OK - free space: / 3326 MB (56%);

7.3 检测脚本存放目录

```
[root@Nagios-Client ~]# cd /usr/local/nagios/libexec/
```

所有的检测脚本都放在该目录下。

7.4 规划编写一个监控 Linux 内存的插件

```
[root@Nagios-Client ~]# free -m
```

	total	used	free	shared	buffers	cached
Mem:	249	235	14	0	45	143
-/+ buffers/cache:		46	203			
Swap:	511	0	511			

第一行的 total 值: 系统内存总值

第二行的 free 值：系统可用内存值

系统可用内存值 / 系统内存总值，所得的百分比为目前系统可用的内存百分比。

```
[root@Nagios-Client ~]# vi /usr/local/nagios/libexec/check_mem

#!/bin/bash

#Written by ZhaoShundong

#This Nagios plugin can be check you system memory status

USAGE_Method="$ (basename $0) [-w|--warning] <Free Percent> [-c|--critical] <Free Percent>"

USAGE_Value="WARNING value must be large than CRITICAL value: `basename $0` $"

STATE_OK=0

STATE_WARNING=1

STATE_CRITICAL=2

STATE_UNKNOWN=3

#设置 Nagios 要求的状态

if [ $# -lt 4 ];then

    echo

    echo "Usage: $USAGE_Method"

    echo

    exit 0

fi

while [ $# -gt 0 ];do

    case "$1" in

        -w|--warning)

            shift

            WARNING=$1

            ;;

        -c|--critical)

            shift

            CRITICAL=$1

            ;;

    esac

done
```

```
        esac

        shift

done

if [[ $WARNING -eq $CRITICAL || $WARNING -lt $CRITICAL ]]
then
    echo

    echo "$USAGE_Value"

    echo

    echo "Usage: $USAGE_Method"

    echo

    exit 0

fi

FREE_MEM=$(free -m | grep - | awk -F ' ' '{print $4}') #取当前未使用的内存，注意是未使用！
TOTAL_MEM=$(free -m | grep Mem | awk -F ' ' '{print $2}') #取当前系统总内存
PERCENT=$(bc <<< "scale=2;$FREE_MEM/$TOTAL_MEM" | tr '^.' '' ) #使用 bc 输出百分比
if [ "$PERCENT" -le "$CRITICAL" ] #如果监测百分比小于等于用户设置的严重错误值，返回 2
    then

        echo "CRITICAL - $FREE_MEM MB ($PERCENT%) Free Memory"

        exit 2

    fi

if [ "$PERCENT" -le "$WARNING" ] #如果百分比小于等于用户设置的警告值就警告并返回 1
    then

        echo "WARNING - $FREE_MEM MB ($PERCENT%) Free Memory"

        exit 1

    fi

if [ "$PERCENT" -gt "$WARNING" ] #如果百分比大于用户设置的警告值就输出 OK 返回 0
    then

        echo "OK - $FREE_MEM MB ($PERCENT%) Free Memory"

        exit 0

    fi

fi
```

7.5 测试插件

7.5.1 本地测试

```
[root@Nagios-Client ~]# cd /usr/local/nagios/libexec/
[root@Nagios-Client libexec]# chmod +x check_mem
[root@Nagios-Client libexec]# ./check_mem
```

Usage: check_mem [-w|--warning] <Free Percent> [-c|--critical] <Free Percent>

```
[root@Nagios-Client libexec]# ./check_mem -w 30 -c 20
OK - 203 MB ( 81%) Free Memory
```

7.5.2 服务器端测试

1> 修改客户端的 **nrpe.cfg**

```
[root@Nagios-Client ~]# vi /usr/local/nagios/etc/nrpe.cfg
```

添加:

```
command[check_mem]=/usr/local/nagios/libexec/check_mem -w 20 -c 10
```

2> 修改服务器端的服务定义文件

```
[root@Nagios-Server ~]# vi /usr/local/nagios/etc/objects/services.cfg
```

添加:

```
define service {
    host_name          Nagios-Client
    service_description check-mem
    check_period        24x7
    max_check_attempts 4
```

```

normal_check_interval    3
retry_check_interval     2
contact_groups           sagroup
notification_interval    10
notification_period      24x7
notification_options     w,u,c,r
check_command            check_nrpe!check_mem
}

```

3> 重新加载 Nagios

```
[root@Nagios-Server ~]# /etc/init.d/nagios reload
```

Running configuration check...done.

Reloading nagios configuration...done

4> 登陆验证

Host ↑↓	Service ↑↓	Status ↑↓	Last Check ↑↓	Duration ↑↓	Attempt ↑↓	Status Information
Nagios-Client	check-/	OK	05-29-2010 13:05:36	0d 8h 40m 31s	1/4	DISK OK - free space: / 7017 MB (79% inode=96%)
	check-host-alive	OK	05-29-2010 13:04:16	0d 10h 7m 36s	1/4	PING OK - Packet loss = 0%, RTA = 0.31 ms
	check-load	OK	05-29-2010 13:05:05	0d 8h 39m 40s	1/4	OK - load average: 0.00, 0.00, 0.00
	check-mem	OK	05-29-2010 13:06:29	0d 0h 3m 38s	1/4	OK - 201 MB (80%) Free Memory
	check-swap	OK	05-29-2010 13:04:19	0d 8h 41m 48s	1/4	SWAP OK - 100% free (511 MB out of 511 MB)
	check-users	OK	05-29-2010 13:06:00	0d 8h 40m 57s	1/4	USERS OK - 2 users currently logged in
	check-zombie-procs	OK	05-29-2010 13:06:55	0d 8h 40m 5s	1/4	PROCS OK: 0 processes with STATE = Z

7.6 编写一个 MySQL Replication 插件

在上一章已经简单介绍了，如果编写 Nagios 的检测插件，本章针对 MySQL Replication 做监控插件的介绍，如果还不清楚 MySQL Replication 的朋友请在网站查找 <http://www.unixhot.com>。

7.7 建立 MySQL 检测用户

```
mysql> GRANT REPLICATION CLIENT ON *.* TO nagios@localhost identified by 'nagiosre';
```

7.8 编写插件

```
[root@Nagios-Client ~]# vi /usr/local/nagios/libexec/check_slave
```

```
#!/bin/bash

MYSQLUSER=nagios
MYSQLPS=nagiosre
MYSQLBIN=/usr/local/mysql/bin/mysql

STATE_OK=0
STATE_WARNING=1
STATE_CRITICAL=2
STATE_UNKNOWN=3

STATE_NUM=$(($MYSQLBIN -u$MYSQLUSER -p$MYSQLPS -e "show slave status\G" | grep Running |
grep Yes | wc -l)

if [ $? -ne 0 ];then
    echo "Please Check the Plugins"
    exit $STATE_UNKNOWN
fi

if [ "${STATE_NUM}" -eq 2 ];then
    echo "Check OK,MySQL Replication is running"
    exit $STATE_OK
else
    echo "Check Critical,MySQL Replication is error"
    exit $STATE_CRITICAL
fi
```

7.9 测试插件

7.9.1 本地测试

```
[root@Nagios-Client ~]# cd /usr/local/nagios/libexec/
[root@Nagios-Client libexec]# chmod +x check_slave
[root@Nagios-Client libexec]# ./check_slave
Check OK, MySQL Replication is running
```

7.9.2 服务器端测试

1> 修改客户端的 **nrpe.cfg**

```
[root@Nagios-Client ~]# vi /usr/local/nagios/etc/nrpe.cfg
添加:
command[check_slave]=/usr/local/nagios/libexec/check_slave
```

2> 修改服务器端的服务定义文件

```
[root@Nagios-Server ~]# vi /usr/local/nagios/etc/objects/services.cfg
添加:
define service {
    host_name            Nagios-Client
    service_description  check-slave
    check_period          24x7
    max_check_attempts    4
    normal_check_interval 3
    retry_check_interval  2
    contact_groups        sagroup
    notification_interval 10
    notification_period    24x7
}
```

```
notification_options    w, u, c, r
check_command           check_nrpe!check_slave
}
```

3> 重新加载 Nagios

```
[root@Nagios-Server ~]# /etc/init.d/nagios reload
```

```
Running configuration check...done.
```

```
Reloading nagios configuration...done
```

4> 登陆验证

check-/	OK	05-30-2010 01:33:19	0d 21h 8m 30s	1/4	DISK OK - free space: / 6259 MB (70% inode=95%);
check-host-alive	OK	05-30-2010 01:33:17	0d 22h 35m 35s	1/4	PING OK - Packet loss = 0%, RTA = 0.39 ms
check-load	OK	05-30-2010 01:34:02	0d 21h 7m 39s	1/4	OK - load average: 0.00, 0.00, 0.00
check-mem	OK	05-30-2010 01:34:47	0d 6h 25m 37s	1/4	OK - 197 MB (79%) Free Memory
check-slave	OK	05-30-2010 01:34:36	0d 0h 3m 30s	1/4	Check OK,MySQL Replication is running
check-swap	OK	05-30-2010 01:32:32	0d 21h 9m 47s	1/4	SWAP OK - 100% free (511 MB out of 511 MB)
check-users	OK	05-30-2010 01:33:32	0d 6h 27m 6s	1/4	USERS OK - 2 users currently logged in
check-zombie-procs	OK	05-30-2010 01:34:17	0d 6h 25m 11s	1/4	PROCS OK: 0 processes with STATE = Z

该章通过两个实例，介绍了如何编写 Nagios 插件，当然实例中的功能是远远不够的，至少我生产应用的脚本要比这个复杂的多，该文并不是讲 Shell 编程的，所以通过简单的两个脚本把功能实现了而已。如果你在脚本上遇到了难题，欢迎到“运维社区”讨论。

第 8 章 Nagios 自动化批量部署

8.1 自动化部署简介

由于服务器数量的不断增加，面对数以千计的需要做 Nagios 监控的客户端，自动化部署就会被提上日程了,自动化部署最简单的是把安装时的命令通过脚本组合在一起，再加上一些成功与否的判断，在这里粘贴，本人写的一个简单的自动化部署脚本，此脚本简单易懂，且在不断更新中，需要

最新脚本信息，请关注 <http://www.unixhot.com>。

脚本下载地址：<http://www.unixhot.com/nagios.zip>

8.2 Nagios Server 端部署脚本

（略）暂不推荐使用

8.3 Nagios Client 端部署脚本

```
#/bin/bash

clear

echo "=====
echo "Welcome to UnixHot Linux Monitor"
echo ""
echo ""
echo "UHLM v1.0 client by UnixHot "
echo "=====
echo ""
echo "For more information please visit http://www.unixhot.com/"
echo ""

SOFTWARE_PATH=/usr/local/src
PACKAGE1=gcc
PACKAGE2=glibc
PACKAGE3=xinetd
#PACKAGE4=gd

download() {
    echo "=====Start download Nagios Client package=====
    cd $SOFTWARE_PATH
```



```
wget
http://prdownloads.sourceforge.net/sourceforge/nagiosplug/nagios-plugins-1.4.14.tar.gz
wget http://prdownloads.sourceforge.net/sourceforge/nagios/nrpe-2.12.tar.gz
chmod +x nagios-plugins-1.4.14.tar.gz
chmod +x nrpe-2.12.tar.gz
echo "Download packages finished!"
echo "=====
}

nagios-plugins() {
    echo "=====install Nagios-plugins=====
    cd $SOFTWARE_PATH
    useradd -s /sbin/nologin nagios
    tar zxvf nagios-plugins-1.4.14.tar.gz
    cd nagios-plugins-1.4.14
    ./configure
    make && make install
}

nrpe() {
    echo "=====install nrpe=====
    cd $SOFTWARE_PATH
    tar zxvf nrpe-2.12.tar.gz
    cd nrpe-2.12
    ./configure && make all
    make install-plugin
    make install-daemon
    make install-daemon-config
    make install-xinetd
}
```

```

config() {
    echo "=====Config=====
sed -i 's/yes/no/g' /etc/xinetd.d/nrpe
sed -i 's/127.0.0.1/127.0.0.1 '$Nagios_Server'/g' /etc/xinetd.d/nrpe
echo "nrpe 5666/tcp #nrpe" >> /etc/services
chown -R nagios:nagios /usr/local/nagios
/etc/init.d/xinetd restart
sleep 5
netstat -na | grep 5666
echo "*****
echo "If you look the LISTEN 5666 port, Congratulations, Nagios-Client Install are
successful"
    echo "*****
}

main() {
    rpm -q $PACKAGE1 $PACKAGE2 $PACKAGE3 > /tmp/rpm.log
    SOFTWARE_STATUS=`grep 'not installed' /tmp/rpm.log`
    if [ -z "$SOFTWARE_STATUS" ];then
        echo "Please input the Nagios Server IP Address"
        read
        echo "The Nagios server ip address is $REPLY"
        Nagios_Server=$REPLY
        download
        nagios-plugins
        nrpe
        config
    else
        echo "Please install the following software first!"

```

```
        echo ""
        echo ""
        grep 'not installed' /tmp/rpm.log
        echo ""
    fi
}
```

Main

第 9 章 Nagios 用户和权限管理

在我们的实际应用中，用户和权限是必须掌握的，因为在一个运维 team 中，不可能每个人都有权限去改变 Nagios 的设置，如果你想让其它人仅仅有查看某些服务器、甚至某些服务的权限，就需要使用 Nagios 的权限设置了。

通过前面几个章节的介绍，大家应该会认识到，我们对 Nagios 的权限控制，实际上就是对 Web 页面的权限控制，也就是对 CGI 程序模块的权限控制。在创建 nagiosadmin 的时候，我特意强调了，用户名必须为 nagiosadmin，是因为，这涉及到我们这一章节要讲到的权限控制。

9.1 添加认证用户

这是设置权限的第一步，和添加 nagiosadmin 这个默认用户一样，我们可以使用 htpasswd 命令来添加可以登陆 Nagios 的用户。

```
[root@Nagios-Server ~]# htpasswd /usr/local/nagios/etc/htpasswd.users tech
New password:
Re-type new password:
Adding password for user tech
```

9.2 打开用户验证功能

在用户权限配置，我们都会围绕在一个配置文件周围，这个文件就是 cgi.cfg 文件。所有的修

改均在此文件中。再次强调一个权限问题，该文件的所有者和所有组都是 nagios 用户。

```
[root@Nagios-Server ~]# vim /usr/local/nagios/etc/cgi.cfg
```

首先需要打开用户验证选项。

use_authentication=1（默认是打开的）

9.3 许可权限

针对许可权限的详细内容，可以查看 Nagios 官方网站，在这里，简单介绍在实际生产环境中需要用到的权限。你现在可以使用刚才添加的 tech 用户登陆，你会发现，无法查看主机、服务等信息。因为我们还没有授权。

查看默认的权限设置：

```
[root@Nagios-Server ~]# grep "authorized" /usr/local/nagios/etc/cgi.cfg

authorized_for_system_information=nagiosadmin

authorized_for_configuration_information=nagiosadmin

authorized_for_system_commands=nagiosadmin

authorized_for_all_services=nagiosadmin

authorized_for_all_hosts=nagiosadmin

authorized_for_all_service_commands=nagiosadmin

authorized_for_all_host_commands=nagiosadmin

#authorized_for_read_only=user1,user2
```

你会发现，系统默认给 nagiosadmin 的权限是如何设置的，如果我们给 tech 授权，就是很好的参考物，例如，我们是 tech 可以查看所有主机的监控信息。可以在 cgi.cfg 添加如下内容：

```
[root@Nagios-Server ~]# vim /usr/local/nagios/etc/cgi.cfg
```

```
authorized_for_all_services=tech
```

```
[root@Nagios-Server ~]# /etc/init.d/nagios reload
```

重载完后刷新页面，可以发现，目前可以查看所有 service 的监控信息了，但是没有主机监控信息，也不能执行外部命令，关闭某个服务的通知功能。

第 10 章 Nagios 使用杂谈

Nagios 的 Web 界面的使用非常简单，本文不做过多的介绍。包括如何添加 cgi 组件，来实现各

种功能都可以参考 Nagios 的中文文档，来获得帮助，本章简单的讲讲日常经常使用的功能。

10.1 报警方式

Nagios 的报警功能一直是它最强大的地方，Nagios 默认使用 mail 命令发送报警邮件，RedHat 默认安装并启动了 sendmail，为了安全你可以使用其他 SMTP 服务，如 postfix, qmail 等。也有很多使用第三方的如飞信来发送报警短信，并不推荐使用，因为这是正规化运营中必须禁止的。当然，这不代表你不能使用，很多小公司都是使用这种方式。有条件可以单独买一台短信猫，和人事公用，用来发送报警短信和一些日常工作的通知短信。还有很多第三方的短信平台，可以提供 http、JDK、软甲+数据库等各种各样的接口使用，而且价钱很便宜，通常充值 3000 元，短信可以按 0.07 元的价格发送，在这里就不推荐具体的公司了。

如果你选择使用邮件报警，可以使用 139 邮箱的短信通知功能，测试你的 Nagios 是否可以正常发送邮件：`[root@Nagios-Server ~]# echo "hehe" | mail -s "test" 574592**@qq.com`

如果测试没问题，你的 Nagios 配置也没有问题，就可以正常使用了。注意：发送的邮件一般会被当做垃圾邮件，存放在垃圾邮件箱。

10.2 关闭报警

使用率最高的功能了，如果某些服务器需要送修，如果预期进行某些主机和服务的维护工作，我们可以提前关闭该主机或者服务的报警功能，不然你的邮箱会不停的接受报警邮件，默认 5 分钟发送一封报警邮件，直到恢复为止（Nagios 誓不罢休的精神值得我们应用到学习中）。

在 Services 标签下，点击相应的服务名称，在右侧点击 **Disable notifications for this service**，直接提交即可关闭，完成后，返回 Services 标签，你会发现，在该服务前面有个小喇叭的符号，上面有个红色的 X。

10.3 Nagios Web 界面简介

名称	说明
General	
Home	显示 Nagios 最新版本、版权等信息。

Documentation	Nagios 的文档
Current Status	
Tactical Overview	概述信息，主机，服务状态的状态总览
Map	Statusmap.cgi 提供的功能
Hosts	监控主机的状态列表，status.cgi 提供
Services	监控服务的状态列表，status.cgi 提供
Host Groups	监控主机组的状态列表，status.cgi 提供
Summary	摘要信息
Grid	详细信息
Service Groups	监控服务组的状态列表，status.cgi 提供
Summary	摘要信息
Grid	详细信息
Problems	
Services(Unhandled)	出现报警故障的服务
Hosts(Unhandled)	出现报警故障的主机
Reports	
Availability	查看主机或者服务单位时间内的可用性
Trends	主机或者服务可用性的趋势图
Alerts	
History	历史信息查询
Summary	主机和服务的摘要信息
Histogram	主机或者服务的柱状图信息
Notifications	故障通知列表
Event Log	Nagios 日志信息

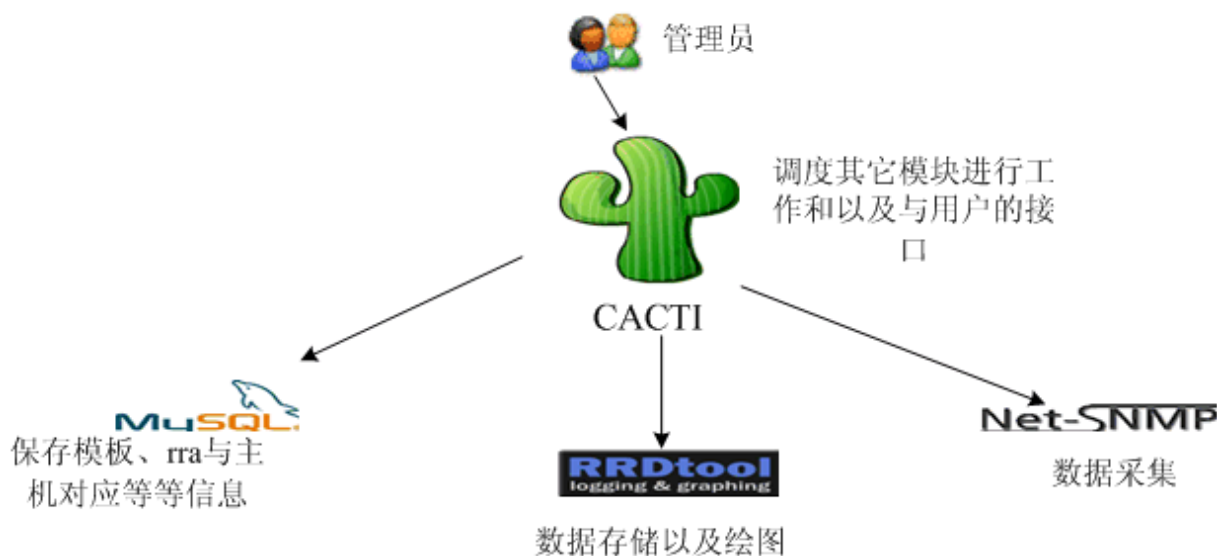
第三部分 Cacti 监控

第 1 章 Cacti 简介

1.1 什么是 Cacti

简单的说 Cacti 就是一个 PHP 程序。它通过使用 SNMP 协议获取远端网络设备和相关信息，（其实就是使用 Net-SNMP 软件包的 snmpget 和 snmpwalk 命令获取）并通过 RRDTOOL 工具绘图，通过 PHP 程序展现出来。我们使用它可以展现出监控对象一段时间内的状态或者性能趋势图。

1.2 Cacti 架构概述



通过上图我们可以发现 Cacti 有三大组件：MySQL 数据库、RRDtool 工具、Net-Snmp 程序。MySQL 并不负责监控数据的存储，只保存模板和 rra 文件与主机对应等信息，而真正的监控数据存储和图像展现都是 rrdtool 来做的。而 Net-SNMP 负责数据的采集。

1.3 Cacti 相关资源

1> Cacti 官网: <http://www.cacti.net/>

2> RRDTool 官网: <http://www.mrtg.org/rrdtool/>

3> Cacti 文档: <http://www.mrtg.org/rrdtool/doc/>

4> CactiEZ OS: <http://cactiez.cactiusers.org/>

CactiEZ 是一个可安装的 Linux OS (其实就是基于 CentOS 的集成了 kickstart 脚本来进行自动安装) 可以直接安装成为一个包括很多扩展组件的 Cacti, 对于急于部署 Cacti 程序, 又不想进行繁琐安装的朋友们是一个很不错的选择。

第 2 章 SNMP 协议

2.1 SNMP 协议简介

SNMP(Simple Network Management Protocol)简单网络管理协议。是由互联网工程任务组(IETF: Internet Engineering Task Force)定义的一套网络管理协议。该协议基于简单网关监视协议(SGMP: Simple Gateway Monitor Protocol)。利用 SNMP, 一个管理工作站可以远程管理所有支持这种协议的网络设备, 包括监视网络状态、修改网络设备配置、接收网络事件警告等。

SNMP 采用了 Client/Server 模型的特殊形式: 代理/管理站模型。对网络的管理与维护是通过管理工作站与 SNMP 代理间的交互工作完成的。每个 SNMP 从代理负责回答 SNMP 管理工作站(主代理)关于 MIB 定义信息的各种查询。

2.1.1 管理信息库 MIB

IETF 规定的管理信息库 MIB (其中定义了可访问的网络设备及其属性, 由对象识别符(OID: Object Identifier)唯一指定。MIB 是一个树形结构, SNMP 协议消息通过遍历 MIB 树形目录中的节点来访问网络中的设备, MIB 的树形结构类似于我们常见到的 DNS 树形结构。

以下链接分别为 TCP 和 UDP 的 MIB 树形结构

<http://www.oidview.com/mibs/0/UDP-MIB.html>

<http://www.oidview.com/mibs/0/TCP-MIB.html>

2.2.2 SNMP 的五种消息类型

SNMP 中定义了五种消息类型:

- 1) get-request 操作: 从代理进程处提取一个或多个参数值, snmpget 命令实现该类型。
- 2) get-next-request 操作: 从代理进程处提取一个或多个参数的下一个参数值, snmpwalk 命令实现。
- 3) set-request 操作: 设置代理进程的一个或多个参数值
- 4) get-response 操作: 返回的一个或多个参数值, 返回值是由代理进程发出的。它是前面3中操作的响应操作。
- 5) trap 操作: 代理进程主动发出的报文, 通知管理进程有某些事情发生。

2.2 Net-SNMP 安装配置

2.2.1 Net-SNMP 安装

Net-snmp 的安装如下, 前两个包是基本的 net-snmp 的包, net-snmp-utils 是一些工具。

```
[root@UnixHot ~]# yum install -y net-snmp net-snmp-libs net-snmp-utils
[root@UnixHot ~]# cd /etc/snmp/
[root@UnixHot snmp]# mv snmpd.conf snmpd.conf.bak
```

2.2.2 Net-SNMP 的配置

```
[root@UnixHot snmp]# vim /etc/snmp/snmpd.conf
#把 admin 这个 COMMUNITY 串和 admin 这个名字做好一对映射
com2sec admin default admin
#把 admin 放在 admin 组里, 使用 v2c 协议
group admin v2c admin
#设置 admin 这个 view 能够查看.1 下面所有节点的权限
view admin included .1 80
#权限设置, 格式为:  access NAME CONTEXT MODEL LEVEL PREFIX READ WRITE NOTIFY"
access admin "" any noauth exact admin admin none
启动 SNMP。
```

```
[root@Nagios-Server snmp]# /etc/init.d/snmpd start
```

查看端口是否在监听, 默认监听 UDP 161 端口。

```
[root@Nagios-Server snmp]# netstat -nulp | grep 161
```

```
udp          0      0 0.0.0.0:161          0.0.0.0:*
```

2.3 snmpget 和 snmpwalk 使用

如果你安装了 net-snmp-util, 就会发现安装了很多以 snmp 开头的工具, 在这里重点介绍和我们监控密切相关的 snmpget 和 snmpwalk 工具。Snmpget 和 snmpwalk 的区别很多朋友总是搞不清楚, 其实很简单, SNMP 在管理进程和代理进程之间交互信息定义了 5 种报文。Snmpget 是使用 get-request 的查询请求。而 Snmpwalk 是使用 get-next-request 的查询请求。

2.4 使用 tcpdump 查看 SNMP 通信

为了体现 snmpget 和 snmpwalk 的工作机制, 现在我们用 tcpdump 来获取 SNMP 协议的通信情况, 我的环境是两台测试机, 一台为 SNMP Server, 使用 snmpget 和 snmpwalk 获取 SNMP Client 的信息, 以便我们分析:

Server 执行:

```
[root@SNMP-S ~]# snmpget -v2c -c admin 192.168.140.13 .1.3.6.1.4.1.2021.10.1.3.1
```

```
UCD-SNMP-MIB::laLoad.1 = STRING: 0.01
```

Client 端显示:

```
[root@SNMP-C ~]# tcpdump 'port 161'
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
```

```
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
```

```
04:06:58.414520 IP localhost.54379 > localhost.snmp: C=admin GetRequest(31) E:2021.10.1.3.1
```

```
04:06:58.418808 IP localhost.snmp > localhost.54379: C=admin GetResponse(35)
```

```
E:2021.10.1.3.1="0.01"
```

Server 端执行:

```
[root@SNMP-S ~]# snmpwalk -v2c -c admin 192.168.140.137 .1.3.6.1.4.1.2021.10.1.3
```

```
UCD-SNMP-MIB::laLoad.1 = STRING: 0.00
```

```
UCD-SNMP-MIB::laLoad.2 = STRING: 0.06
```

```
UCD-SNMP-MIB::laLoad.3 = STRING: 0.03
```

Client 端显示:

```
[root@SNMP-C ~]# tcpdump 'port 161'
```

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
```

```
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
```

```
04:07:40.407935 IP localhost.42184 > 192.168.140.137.snmp: C=admin GetNextRequest(30)
```

```
E:2021.10.1.3
```

```
04:07:40.411517 IP 192.168.140.137.snmp > localhost.42184: C=admin GetResponse(35)
```

```
E:2021.10.1.3.1="0.00"
```

```
04:07:40.413132 IP localhost.42184 > 192.168.140.137.snmp: C=admin GetNextRequest(31)
```

```
E:2021.10.1.3.1
```

```
04:07:40.413492 IP 192.168.140.137.snmp > localhost.42184: C=admin GetResponse(35)
```

```
E:2021.10.1.3.2="0.06"
```

```
04:07:40.414521 IP localhost.42184 > 192.168.140.137.snmp: C=admin GetNextRequest(31)
```

```
E:2021.10.1.3.2
```

```
04:07:40.414736 IP 192.168.140.137.snmp > localhost.42184: C=admin GetResponse(35)
```

```
E:2021.10.1.3.3="0.03"
```

```
04:07:40.417153 IP localhost.42184 > 192.168.140.137.snmp: C=admin GetNextRequest(31)
```

```
E:2021.10.1.3.3
```

```
04:07:40.417462 IP 192.168.140.137.snmp > localhost.42184: C=admin GetResponse(36)
```

```
E:2021.10.1.4.1="12.00"
```

2.5 常用例子

CPU 负载统计:

```
1 minute Load: .1.3.6.1.4.1.2021.10.1.3.1
```

5 minute Load: .1.3.6.1.4.1.2021.10.1.3.2

15 minute Load: .1.3.6.1.4.1.2021.10.1.3.3

CPU 使用率:

percentage of user CPU time: .1.3.6.1.4.1.2021.11.9.0

raw user cpu time: .1.3.6.1.4.1.2021.11.50.0

percentages of system CPU time: .1.3.6.1.4.1.2021.11.10.0

raw system cpu time: .1.3.6.1.4.1.2021.11.52.0

percentages of idle CPU time: .1.3.6.1.4.1.2021.11.11.0

raw idle cpu time: .1.3.6.1.4.1.2021.11.53.0

raw nice cpu time: .1.3.6.1.4.1.2021.11.51.0

Memory 使用统计:

Total Swap Size: .1.3.6.1.4.1.2021.4.3.0

Available Swap Space: .1.3.6.1.4.1.2021.4.4.0

Total RAM in machine: .1.3.6.1.4.1.2021.4.5.0

Total RAM used: .1.3.6.1.4.1.2021.4.6.0

Total RAM Free: .1.3.6.1.4.1.2021.4.11.0

Total RAM Shared: .1.3.6.1.4.1.2021.4.13.0

Total RAM Buffered: .1.3.6.1.4.1.2021.4.14.0

Total Cached Memory: .1.3.6.1.4.1.2021.4.15.0

2.6 Windows 下开启 SNMP 代理程序

Windows 下开启 SNMP 代理程序需要安装组建后进行简单的设计即可，这里以 Windows Server 2003 为例，其它 Windows 系列系统与其类似：

第一步、确认当前的 Windows 组件中是否已经安装了 SNMP 协议，在控制面板的“添加或删除程序”中，查看已安装的 Windows 组件。

点击在“管理和监控工具”的详细信息，你会发现“简单网络管理协议（SNMP）”，如果没有安装，

则打勾进行安装，期间你可能需要定位 Windows 安装盘。



SNMP 组件安装成功后，接下来打开“管理”菜单中的“服务”。

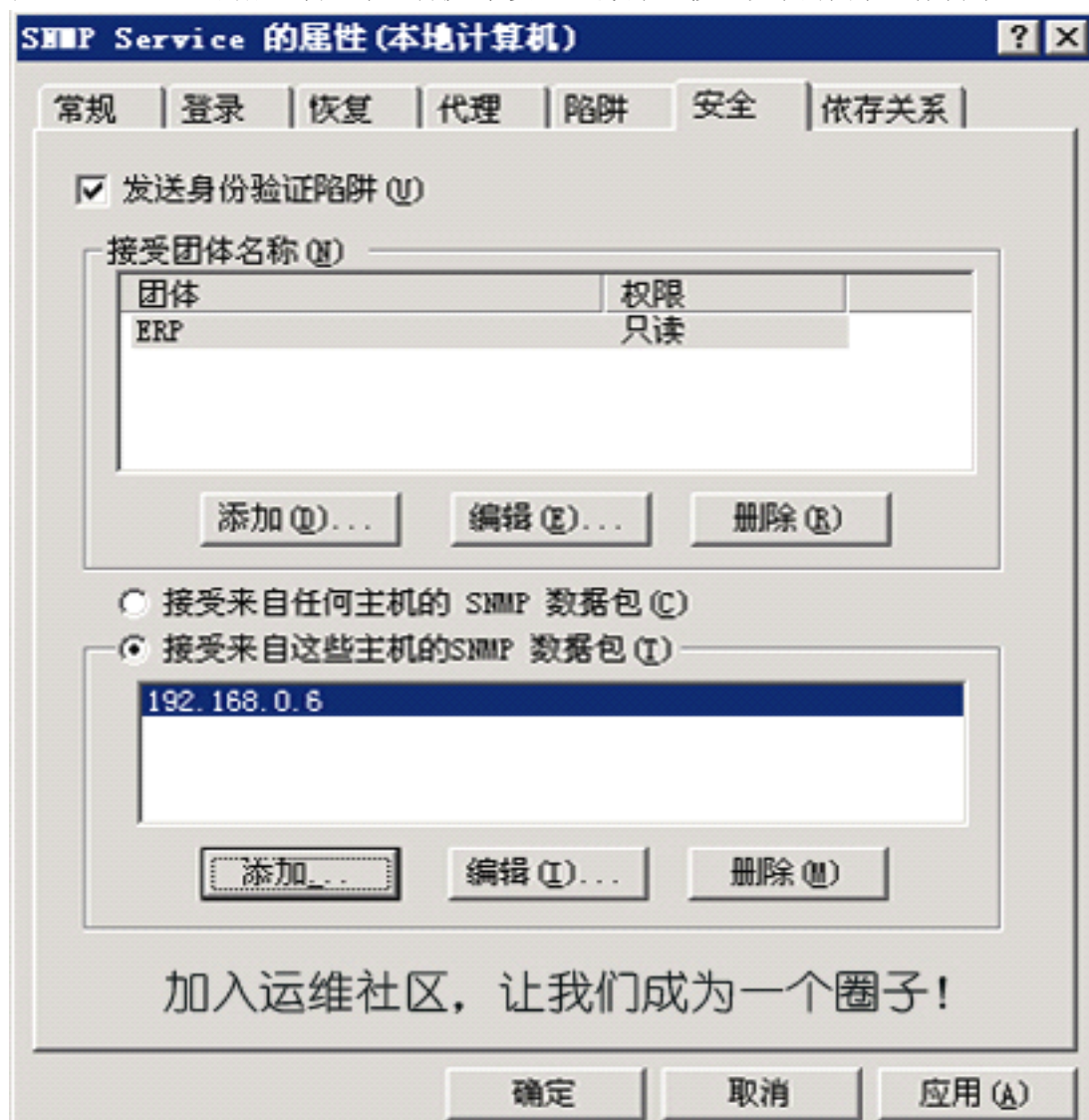


找到 SNMP Service，双击它进行配置。这里需要注意的是，在前边安装 SNMP 协议后，SNMP Service

会默认启动，而它下边的 SNMP Trap Service 是不需要开启的，如果它已被开启，请关闭它。



在 SNMP Service 的配置窗口中，切换到“安全”选项卡，按照以下的图示进行填写。



一切配置好后，保证 SNMP Service 已经开启，并且如果以后重启了服务器，也请来检查是否开启了 SNMP Service。

另外，还要保证你的服务器已经打开了 SNMP Service 的端口，默认为 UDP 161。如果你启用了系统防火墙，那么请添加这个例外端口，打开本地连接->属性->高级->防火墙->例外，添加端口如下：本地连接->属性->高级->防火墙->例外->添加端口。

第 3 章 rrdtool 工具

3.1 rrdtool 简介

RRDTool 是由 Tobias Oetiker 开发的开源软件，它使用 RRD（Round Robin Databases）作为存储格式，Round robin 是一种处理定量数据以及当前元素指针的技术，RRDTool 主要用来跟踪对象的变化情况，生成对象变化的趋势图。

在这一章，我们从 rrdtool 入手，来学习 rrdtool 工具的实用方法，只有深刻理解了 rrdtool 工具的实用才可以解决以后工作中出现的各种问题，并可以实现自定义监控的图形展示，如果把 FLEX 比作网站开发者的福音，那么 rrdtool 便是运维工程师手中漂亮的徽章。

3.2 rrdtool 安装

3.2.1 安装前的依赖

安装 rrdtool 需要依赖很多库文件，建议使用 YUM 安装，避免漫长的编译过程。

```
[root@Cacti-Server ~]# yum install -y cairo-devel libxml2-devel pango-devel pango-libpng-devel freetype freetype-devel libart_lgpl-devel
```

3.2.2 安装 rrdtool 软件包

```
[root@Cacti-Server ~]# cd /usr/local/src
```



```
[root@Cacti-Server src]# wget http://oss.oetiker.ch/rrdtool/pub/rrdtool-1.4.4.tar.gz
[root@Cacti-Server src]# tar zxvf rrdtool-1.4.4.tar.gz
[root@Cacti-Server src]# cd rrdtool-1.4.4
[root@Cacti-Server rrdtool-1.4.4]# ./configure --prefix=/usr/local/rrdtool
[root@Cacti-Server rrdtool-1.4.4]# make && make install
```

3.2.3 检测安装是否成功

```
[root@Cacti-Server ~]# /usr/local/rrdtool/bin/rrdtool
```

执行 rrdtool 命令，如果出现以下输出，表示安装成功，并列出了该命令的使用帮助。

```
RRDtool 1.4.4 Copyright 1997-2010 by Tobias Oetiker <tobi@oetiker.ch>
Compiled Nov 15 2010 09:13:34
```

```
Usage: rrdtool [options] command command_options
```

```
Valid commands: create, update, updateev, graph, graphv, dump, restore,
                last, lastupdate, first, info, fetch, tune,
                resize, xport, flushcached
```

```
RRDtool is distributed under the Terms of the GNU General
Public License Version 2. (www.gnu.org/copyleft/gpl.html)
```

```
For more information read the RRD manpages
```

在安装完毕，开始动手之前，有必要了解下 rrdtool 是怎么生成我们最终看到的监控图表，总体来说可以分为四步：

第一步：建立 rrd 数据文件（或者说叫数据库，或者叫 log）

第二步：更新 rrd 数据文件。

第三步：创建图形。

第四步：返回第一步，继续执行。

在下面我会根据生成环境中一个实际应用来讲解 rrdtool 工具的使用：

场景：笔者目前在一家电子商务公司任职，

3.3 建立 rrd 数据库文件

官方文档：<http://oss.oetiker.ch/rrdtool/doc/rrdcreate.en.html>

语法：

```
rrdtool create filename [--start|-b start time]
                        [--step|-s step]
                        [--no-overwrite|-0]
                        [DS:ds-name:DST:heartbeat:min:max]
                        [RRA:CF:xff:steps:rows]
```

例子：监控 CPU1 分钟、5 分钟、15 分钟内的平均负载。

```
/usr/local/rrdtool/bin/rrdtool create ${_HOME}/${_SNMP_HOST}.rrd \
--start $(date -d '1 day ago' +%s) \
--step 300 \
DS:sys1minute:GAUGE:600:0:500 \
DS:sys5minute:GAUGE:600:0:500 \
DS:sys15minute:GAUGE:600:0:500 \
RRA:AVERAGE:0.5:1:600          RRA:AVERAGE:0.5:6:700          RRA:AVERAGE:0.5:24:775
RRA:AVERAGE:0.5:288:797 \
RRA:MAX:0.5:1:600 RRA:MAX:0.5:6:700 RRA:MAX:0.5:24:775 RRA:MAX:0.5:288:797
```

Filename:建立的文件名称，一般以 rrd 结尾。

--step: 时间间隔，就是 RRDtool “期望” 每隔多长时间就收到一个值，默认是 5 分钟。可以调整，但是如果间隔太短，会加重监控服务器的负载。

--start: 给出 RRDtool 的第一个记录的起始时间。RRDtool 不会接受任何采样时间小于或者等于指定时间的数据。也就是说 --start 指定了数据库最早的那个记录是从什么时候开始的。如果 update 操作中给出的时间在 --start 之前，则 RRDtool 拒绝接受。--start 选项的值必须是 timestamp 的格式。

DS:*ds-name:GAUGE | COUNTER | DERIVE | ABSOLUTE:heartbeat:min:max*

DS : (Date Source) 用于定义数据源名称。也就是用于存放脚本的结果的变量名 (DSN)。就是例子中的 sys1minute、sys5minute、sys15minute。DSN 长度是 1-19 个字符，必须是 0-9, a-z, A-Z 。

DST: (Data Source Type) 用于定义数据源的类型。有 COUNTER、GUAGE、DERIVE、ABSOLUTE、COMPUTE 5 种。

Hearbeat: 用于定义有效期，简单的理解，就是，如果在某一时间点，比如中午 12 点，应该产生的数值，由于某种原因，没有产生，那么就把 12:00 前后各 300 秒的平均值用 12:00 代替：

Min 和 Max: 两个代表 DSN 这个变量的范围，最小时多少，最大是多少。这个根据实际应用来定义，之后是可以更改的，如果接收到的值超过定义的最小或者最大的值，被 rrdtool 认为是“未知”。

RRA:*AVERAGE | MIN | MAX | LAST:xff:steps:rows*

RRA: (Round Robin Archive) 用于定义数据是如何记录、如何归档的。

CF: (consolidation function) 合并方式，包含四类：AVERAGE、MIN、MAX、LAST。

经过一个刷新周期 (5 分钟)，会获得一个主数据点 (PDP)，将若干个 PDPs 使用合并方式 (CF) 合并后会产生一个合并数据点 CDP (consolidated data point)。

xff: xfiles factor 和 unkown 数据有关，很多资料都取 0.5

step: 把"step"条 PDP 合并形成一条 CDP

row: 记录的合并数据点 CDP 条数

3.4 更新 rrd 数据库文件

上小节，我们建立了一个 rrd file，在这小节，需要去更新 rrd file，然后才能根据 rrd 文件来绘图。更新 rrd file，简单的说就是给上面定义的 DSN (Date Soure Name) 赋值。

首先给 DSN 赋值。

```
sys1minute=$(snmpwalk -v2c -c ${SNMPCNAME} ${IPADDRESS} .1.3.6.1.4.1.2021.10.1.3.1 | awk '{ print $4 }')
```

```
sys5minute=$(snmpwalk -v2c -c ${SNMPCNAME} ${IPADDRESS} .1.3.6.1.4.1.2021.10.1.3.2 | awk '{ print $4 }')
```

```
sys15minute=$(snmpwalk -v2c -c ${SNMPCNAME} ${IPADDRESS} .1.3.6.1.4.1.2021.10.1.3.3 | awk '{ print $4 }')
```

然后执行 `rrdtool update`

语法:

```
rrdtool update filename
    [--template|-t ds-name:ds-name:...]
    [--daemon <address>]
    time|N:value[:value...]
    at-time@value[:value...]
    [ time:value[:value...] ..]
```

3.5 使用 **rrd** 画图

语法:

`rrdtool graph filename`

`[-s|--start seconds]` #绘图开始时间，默认是一天前

`[-e|--end seconds]` #绘图结束时间，默认是现在。

```
[-x|--x-grid x-axis grid and label]
[-Y|--alt-y-grid] [--full-size-mode]
[-y|--y-grid y-axis grid and label]
[-v|--vertical-label string] #Y 轴上的说明字符串
[-w|--width pixels] #绘图区域的宽度
[--right-axis scale:shift] [--right-axis-label label]
[--right-axis-format format]
[-h|--height pixels] #绘图区域的高度
[-o|--logarithmic]
[-u|--upper-limit value] [-z|--lazy]
[-l|--lower-limit value] [-r|--rigid]
[-g|--no-legend] [--daemon <address>]
[-F|--force-rules-legend]
[-j|--only-graph]
```

[-n|--font FONTTAG:size:font]
[-m|--zoom factor]
[-A|--alt-autoscale]
[-M|--alt-autoscale-max]
[-G|--graph-render-mode {normal,mono}]
[-R|--font-render-mode {normal,light,mono}]
[-B|--font-smoothing-threshold size]
[-T|--tabwidth width]
[-E|--slope-mode]
[-P|--pango-markup]
[-N|--no-gridfit]
[-X|--units-exponent value]
[-L|--units-length value]
[-S|--step seconds] #绘图起始时间，默认是一天前
[-f|--imginfo printfstr]
[-a|--imgformat PNG] #设置图像格式
[-c|--color COLORTAG#rrgbbb[aa]]
[--border width]
[-t|--title string] #图上显示的标题字符串
[-W|--watermark string]
[DEF:vname=rrd:ds-name:CF]

[CDEF:vname=rpn-expression]
[VDEF:vdefname=rpn-expression]
[PRINT:vdefname:format]
[GPRINT:vdefname:format]
[COMMENT:text]
[SHIFT:vname:offset]
[TEXTALIGN:{left|right|justified|center}]

```
[TICK:vname#rrgbb[aa][:[fraction][:legend]]]
[HRULE:value#rrgbb[aa][:legend]]
[VRULE:value#rrgbb[aa][:legend]]
[LINE[width]:vname[#rrgbb[aa][:[legend][:STACK]]]]
[AREA:vname[#rrgbb[aa][:[legend][:STACK]]]]
[PRINT:vname:CF:format] (deprecated)
[GPRINT:vname:CF:format] (deprecated)
[STACK:vname[#rrgbb[aa][:legend]]] (deprecated)
```

第 4 章 Cacti 安装

4.1 安装前的准备工作

Cacti 要依赖一下软件包，所以在安装前先检查，并安装。

```
[root@Cacti-Server ~]# yum install -y httpd php php-mysql php-snmp mysql mysql-server
net-snmp gd gd-devel php-gd
```

4.2 安装 Cacti

```
[root@Cacti-Server ~]# cd /usr/local/src
[root@Cacti-Server src]# wget http://www.cacti.net/downloads/cacti-0.8.7g.tar.gz
[root@Cacti-Server src]# tar zxvf cacti-0.8.7g.tar.gz
[root@Cacti-Server src]# mv cacti-0.8.7g /var/www/html/cacti
```

4.3 配置 MySQL

```
[root@Cacti-Server ~]# /etc/init.d/mysqld start
[root@Cacti-Server ~]# chkconfig --level 3 mysqld on
[root@Cacti-Server ~]# mysqladmin -u root password unixhot
[root@Cacti-Server ~]# mysqladmin -u root -p create cacti
```

```
[root@Cacti-Server ~]# mysql -u root -p cacti < /var/www/html/cacti/cacti.sql
[root@Cacti-Server ~]# mysql -u root -p
mysql> grant all on cacti.* to cactiuser@localhost identified by 'cactiuser';
```

4.4 配置 Apache 和 PHP

设置 MySQL 的链接信息。

```
[root@Cacti-Server ~]# vim /var/www/html/cacti/include/config.php
$database_type = "mysql";
$database_default = "cacti";
$database_hostname = "localhost";
$database_username = "cactiuser";
$database_password = "cactiuser";
$database_port = "3306";
```

启动 Apache。

```
[root@Cacti-Server ~]# /etc/init.d/httpd start
[root@Cacti-Server ~]# chkconfig --levels 3 httpd on
```

4.5 Cacti 安装

```
[root@Cacti-Server ~]# useradd cacti    (添加 Cacti 用户)
[root@Cacti-Server ~]# passwd cacti    (设置密码)
[root@Cacti-Server ~]# chown -R cacti:apache /var/www/html/cacti/
[root@Cacti-Server ~]# echo '*/* * * * * cacti php /var/www/html/cacti/poller.php >
/dev/null 2>&1' >> /etc/crontab
[root@Cacti-Server ~]#/etc/init.d/crond reload
```

浏览器访问 <http://IP/cacti/install> 设置相关软件的路径。

第 5 章 Cacti 配置管理

5.1 Cacti 登陆

Cacti 的默认用户名密码都为：admin。注意：第一次登陆后，首先强制要求你修改 admin 用户的密码，注意设置安全密码，并牢记。

5.2 Cacti 界面介绍

侧栏名称	说 明
Create	
New Graphs	创建图像的快捷方式。
Management	管理菜单
Graph Management	图像管理，可以复制、删除、合并图形。
Graph Trees	图像树，可以设置图像树行结构，方便分类和查看
Data Sources	数据源管理，也就是管理生成的 rrd 文件
Devices	设备管理，我们添加监控主机或者网络设备都在这里
Collection Methods	获收集数据方式
Data Queries	数据查询
Data Input Methods	数据导入方式
Templates	模 板
Graph Templates	图像模板
Host Templates	主机模板
Data Templates	数据模板
Import/Export	导入\导出
Import Templates	导入模板
Export Templates	导出模板
Configuration	配 置
Settings	主要配置菜单，如使用程序的路径、Snmp 版本等等。
Utilities	其它功能

System Utilities	显示系统的日志和缓存信息
User Management	用户管理，用户权限设置
Logout User	注销用户

5.3 添加服务器监控实例

默认安装完成后，Cacti 会默认创建监控服务器 LocalHost。在这里，我们先删除这个服务器监控，然后重新一步一步建立监控对象。

在 Console 标签下，Management 下的 Devices。选中 localhost 后，点击 go，删除。会提示你删除选项，选择默认选项 Delete all associated **Graph(s)** and **Data Source(s)**。后确定。默认的监控对象就删除了。

5.3.1 添加一台需要监控的服务器

1>新建设备：点击 Devices，点击右上方的 Add 按钮，打开新建设备页面

2>设备信息：填写相关信息，Description（描述信息）：Cacti-Client

Host（服务器 IP 地址）：192.168.140.141

Host Template（主机模板）：ucd/net SNMP Host

Downed Device Detection:SNMP

SNMP Version:Version 2

SNMP Community:admin

其他值默认即可，点击 Create 创建设备。

注意：如果 SNMP 协议设置错误，回报红色的 SNMP error。

3>创建图形：点击 [Create Graphs for this Host](#)

默认会列出 **Graph Templates** 和 **Data Query** [SNMP - Interface Statistics]

将这些模板都选上，点击 Creat 创建。

4>查看图形：点击左侧的 Graph Management 菜单，即可查看刚才选择的监控模板，但是需要等五分钟才会有图形。

5.3.2 创建图形树

现在我们已经添加了设备，也创建了监控对象，并成功的显示出了图像，但是目前的情况不方便我们查看，我们可以建立树形的查看模式。便于我们将相同功能的设备，或者相同区域的设备分类。

点击 Management 标签下的 Graph Trees。会发现，Cacti 默认为我们创建了一个 Default Tree。点击右侧的红 X，将其删除，我们来一步一步创建新的 Trees。

1>点击右侧上方 Add 按钮，打开进行节点页面。

输入图形 tree 的名字：Cacti Test，点击 Create 创建。

2>在 **Tree Items** 右侧点击 Add 按钮，为该项目添加成员。

Tree Item Type: Host

Host: 选择已经加入的设备，Cacti-Client (192.168.140.141)

点击 Create 创建。点击 Save 保存。

3>点击 Graphs 标签，便可产看刚才添加的图形树。

注意：在创建图形树的时候，我们可以常见的不仅仅是主机，还可以是多级主机，或者图形，主要是看 Tree Item Type 你选择的什么，上面的例子，我选择的是 Host。

5.4 Cacti 用户和权限

Cacti 用户和权限的设置是我们工作中必然遇到的问题，因为我需要一个监控工程师来监控日常的应用和网络状况，我只想他有产看图像的权限，也就是登陆后只有 graphs 标签显示，Cacti 已经可以完成相对简单的权限设置。

用户和权限的设置在 Utilities 下的 User Management 下。系统默认创建了两个用户，admin 和 guest。但是 guest 默认是不允许登陆的。

Cacti 的权限控制非常的简单，权限的分类也比较详细，这里不再详细讲解。

第 6 章 Spine (Cactid)

6.1 Spine 是什么

Spine 前身是 Cactid，是纯 C 编写的，是 cmd.php 的替代品，目的是为了加快 SNMP 轮训。

6.2 何时使用 Spine

如果你发现你的 cmd.php 运行超过 300 秒的话，推荐使用 Spine。

6.3 Spine 安装

Spine 依赖以下软件包：

```
[root@Cacti-Server ~]# yum install -y net-snmp-devel openssl-devel mysql-devel
[root@Cacti-Server ~]# cd /usr/local/src
[root@Cacti-Server src]# wget http://www.cacti.net/downloads/spine/cacti-spine-0.8.7g.tar.gz
[root@Cacti-Server src]# tar zxvf cacti-spine-0.8.7g.tar.gz
[root@Cacti-Server src]# cd cacti-spine-0.8.7g
[root@Cacti-Server cacti-spine-0.8.7g]# ./configure
[root@Cacti-Server cacti-spine-0.8.7g]# make && make install
```

6.4 Spine 配置

在 Cacti Web 界面，点击 Configuration 下的 settings。

1> 在 Paths 标签下，填写 Spine 的路径

Spine Poller File Path: /usr/local/spine/bin/spine

2> 在 Poller 标签下，修改 Poller Type 类型为 Spine。

第 7 章 Cacti 监控 Nginx

在完全掌握 Cacti 的基本使用之后，在这里通过第三方插件来监控 Nginx 的状态。是通过 Perl 脚本获得 nginx status 模块的信息，并进行绘图。需要你编译 Nginx 时加入 http_stub_status_module 选项。

7.1 Nginx 配置文件修改

```
location /nginx_status {  
    stub_status on;  
    access_log off;  
    allow 172.16.1.10;  
    deny all;  
}
```

7.2 添加监控模板

```
[root@Monitor-Server src]# wget http://www.unixhot.com/code/cacti-nginx.tar.gz  
[root@Monitor-Server src]# tar zxvf cacti-nginx.tar.gz  
[root@Monitor-Server src]# cd cacti-nginx  
[root@Monitor-Server cacti-nginx]# cp get_nginx_clients_status.pl /var/www/html/cacti/scripts/  
[root@Monitor-Server cacti-nginx]# cp get_nginx_socket_status.pl /var/www/html/cacti/scripts/  
[root@Monitor-Server cacti-nginx]# chmod 0755 /var/www/html/cacti/scripts/get_nginx_clients_status.pl  
[root@Monitor-Server cacti-nginx]# chmod 0755 /var/www/html/cacti/scripts/get_nginx_socket_status.pl
```

7.3 测试脚本

```
[root@Monitor-Server scripts]# ./get_nginx_clients_status.pl http://shop.awc618.com/nginx_status  
nginx_active:1 nginx_reading:0 nginx_writing:1 nginx_waiting:0  
[root@Monitor-Server scripts]# ./get_nginx_socket_status.pl http://shop.awc618.com/nginx_status  
nginx_accepts:12 nginx_handled:12 nginx_requests:27
```

7.4 导入到 Cacti

在测试没有问题的時候，可以直接导入到 Cacti，在 Import/Export 标签下，点击 Import Templates 进行导入操作。

第四部分 Smokeping 网络监控

第 1 章 Smokeping 简介

1.1 什么是 Smokeping

Smokeping 是 rrdtool 的作者 Tobi Oetiker 的作品，是用 Perl 写的，主要是监视网络性能，包括常规的 ping，用 echoping 监 www 服务器性能，监视 dns 查询性能，监视 ssh 性能等。底层也是 rrdtool 做支持，特点是画的图非常漂亮，网络丢包和延迟用颜色和阴影来表示。

1.2 Smokeping 架构组件

Smokeping 有以下组件组成：RRDtool、Fping、Echoping、Curl、Dig、SSh、Perl 模块，Perl、SpeedyCGI、Apache 等。

1.3 Smokeping 相关资源

1> Smokeping 官方网站：<http://oss.oetiker.ch/smokeping/>

2> Smokeping 官方文档：<http://oss.oetiker.ch/smokeping/doc/index.en.html>

第 2 章 Smokeping 安装配置

Smokeping 的安装分为 5 步，第一步、安装 rrdtool 工具。第二步、安装 Fping 和 Echoping 组件。第三步、安装 SpeedyCGI。第四步、安装依赖的 Perl 模块。第五步、下载解压 Smokeping。下面详细介绍具体的安装步骤。

2.1 安装 rrdtool

请参考 《第三部分 Cacti 监控》第三章 rrdtool 安装

2.2 安装 Fping 和 Echoping

2.2.1 下载软件包

```
[root@Smokeping-S src]# wget http://fping.sourceforge.net/download/fping.tar.gz
[root@Smokeping-S src]# wget
http://cdnetworks-kr-1.dl.sourceforge.net/project/echoping/echoping/6.0.2/echoping-6.0.2.tar.gz
[root@Smokeping-S src]# wget
http://cdnetworks-kr-2.dl.sourceforge.net/project/speedycgi/SpeedyCGI/2.22/CGI-SpeedyCGI-2.22.tar.gz
[root@Smokeping-S src]# wget http://oss.oetiker.ch/smokeping/pub/smokeping-2.4.2.tar.gz
```

2.2.2 安装 Fping

```
[root@Smokeping-S src]# tar zxvf fping.tar.gz
[root@Smokeping-S src]# cd fping-2.4b2_to/
[root@Smokeping-S fping-2.4b2_to]# ./configure && make && make install
```

2.2.3 安装 Echoping

```
[root@Smokeping-S src]# tar zxvf echoping-6.0.2.tar.gz
[root@Smokeping-S src]# cd echoping-6.0.2
[root@Smokeping-S echoping-6.0.2]# ./configure （注意：时间较长，请耐心等待。）
[root@Smokeping-S echoping-6.0.2]# make && make install
```

2.3 安装 SpeedyCGI

```
[root@Smokeping-S src]# tar zxvf CGI-SpeedyCGI-2.22.tar.gz
[root@Smokeping-S src]# cd CGI-SpeedyCGI-2.22
[root@Smokeping-S CGI-SpeedyCGI-2.22]# perl Makefile.PL
[root@Smokeping-S CGI-SpeedyCGI-2.22]# make
[root@Smokeping-S CGI-SpeedyCGI-2.22]# make install
```

2.4 安装 Perl 模块

以下全是 Smokeping 需要的 perl 模块。

```
[root@Smokeping-S ~]# yum install -y
perl-libwww-perl perl-IO-Socket-SSL perl-Net-Telnet perl-Socket6 perl-Net-DNS perl-LDAP
perl-libwww-perl perl-IO-Socket-SSL
```

2.5 安装 Smokeping

2.5.1 Smokeping 安装

```
[root@Smokeping-S src]# tar zxvf smokeping-2.4.2.tar.gz
[root@Smokeping-S src]# mv smokeping-2.4.2 /usr/local/smokeping
```

2.5.2 增加 Apache 配置文件。

```
[root@Smokeping-S ~]# vim /etc/httpd/conf.d/smokeping.conf
Alias /smokeping "/usr/local/smokeping/htdocs"
Alias /smokepingcache "/usr/local/smokeping/cache"
<Directory "/usr/local/smokeping/htdocs">
    AddHandler cgi-script cgi
    Options ExecCGI
    AllowOverride None
    Order allow,deny
```

```
Allow from all  
</Directory>
```

2.5.3 增加 Smokeping 认证用户

```
[root@Smokeping-S ~]# htpasswd -c /usr/local/smokeping/etc/htpasswd.users tech  
New password:  
Re-type new password:  
Adding password for user tech
```

2.6 Smokeping 在 FreeBSD 上的安装

如果你监控的平台是 FreeBSD 系统，那么恭喜你，因为 Smokeping 在 FreeBSD 上的安装很简单，只需要一个命令即可：

```
[root@freebsd /root]# pkg_add -r smokeping  
[root@freebsd /root]# whereis smokeping  
smokeping:          /usr/local/bin/smokeping          /usr/local/man/man1/smokeping.1.gz  
/usr/ports/net-mgmt/smokeping
```

2.7 Smokeping 配置

在安装完 Smokeping 之后，默认配置是不能运行的，需要做一下修改。我们需要修改三个配置文件：

```
[root@Smokeping-S ~]# cd /usr/local/smokeping/  
[root@Smokeping-S smokeping]# cp -a bin/smokeping.dist bin/smokeping #Smokeping 主程序  
[root@Smokeping-S smokeping]# cp -ahtdocs/smokeping.cgi.disthtdocs/smokeping.cgi #显示图像，和作为 Slave 端的数据传输接口，因为在启动 Slave 时需要制定该文件。  
[root@Smokeping-S smokeping]# cp -aetc/config.distetc/config #Smokeping 配置文件  
下面具体介绍如何来修改这三个配置文件，来让 Smokeping 成功运行。
```

2.7.1 /usr/local/smokeping/bin/smokeping

```
[root@Smokeping-S ~]# vim /usr/local/smokeping/bin/smokeping
```

修改后的结果如下，修改了三个地方，如下：

#rrdtool 的库文件位置，注意，如果你是 32 的系统，lib 的位置不同。你可以根据报错提示找到正确的位置

```
use lib qw(/usr/local/rrdtool/lib/perl/5.8.8/x86_64-linux-thread-multi);
```

```
use lib qw(/usr/local/smokeping/lib); #Smokeping 的库文件位置
```

```
use strict;
```

```
use warnings;
```

```
use Smokeping 2.004002;
```

```
Smokeping::main("/usr/local/smokeping/etc/config"); #Smokeping 配置文件的位置
```

修改完毕后，可以使用以下命令检测，也就是直接启动看是否报错，如果只出现配置文件的错误标示该文件配置正常，因为配置文件还未修改：

```
[root@Nagios-Server ~]# /usr/local/smokeping/bin/smokeping
```

2.7.2 /usr/local/smokeping/htdocs/smokeping.cgi

```
[root@Smokeping-S ~]# vim /usr/local/smokeping/htdocs/smokeping.cgi
```

该文件修改和内容同 bin/smokeping，也是修改三个位置

```
use lib qw(/usr/local/rrdtool/lib/perl/5.8.8/x86_64-linux-thread-multi); #rrdtool 的 lib 库
```

```
use lib qw(/usr/local/smokeping/lib); #Smokeping 的 lib 库
```

```
Smokeping::cgi("/usr/local/smokeping/etc/config"); #Smokeping 的配置文件
```

修改完毕后，可以使用以下命令检测，如果可以正常显示 HTML 代码，代表配置正常。

```
[root@Nagios-Server ~]# perl /usr/local/smokeping/htdocs/smokeping.cgi
```

2.8 主配置文件 /usr/local/smokeping/etc/config

```
[root@Smokeping-S ~]# cd /usr/local/smokeping/
```

```
[root@Smokeping-S smokeping]# mkdir cache data var
```

由于这个是 Smokeping 的主配置文件，将该配置文件贴出做出详细介绍：


```
[root@Smokeping-S ~]# cat /usr/local/smokeping/etc/config
#####配置文件开始#####

*** General ***    #基本配置

owner      = unixhot

contact    = admin@unixhot.com  #联系人邮箱

mailhost = localhost

sendmail = /usr/lib/sendmail

# NOTE: do not put the Image Cache below cgi-bin

# since all files under cgi-bin will be executed ... this is not
# good for images.

imgcache = /usr/local/smokeping/cache  #rrdtool 生成图片的存放地址

imgurl    = http://192.168.140.141/smokepingcache  注意：这里的设置和 Apache 的配置要一致

datadir   = /usr/local/smokeping/data    #rrd 文件的存放地址

piddir    = /usr/local/smokeping/var      #pid 的存放地址

cgiurl     = http://192.168.140.141/smokeping/smokeping.cgi

smokemail = /usr/local/smokeping/etc/smokemail.dist

tmail     = /usr/local/smokeping/etc/tmail.dist

# specify this to get syslog logging

syslogfacility = local0

# each probe is now run in its own process

# disable this to revert to the old behaviour

# concurrentprobes = no

*** Alerts ***    #报警的配置

to = admin@unixhot.com

from = smokeping@unixhot.com

+some loss

type = loss
```

in percent

pattern = >0%,*12*,>0%,*12*,>0%

comment = loss 3 times in a row

*** Database *** #rrd 数据文件配置

step = 60 #每 60 秒时间，ping5 次

pings = 5

consfn mrhb steps total

AVERAGE 0.5 1 1008

AVERAGE 0.5 12 4320

MIN 0.5 12 4320

MAX 0.5 12 4320

AVERAGE 0.5 144 720

MAX 0.5 144 720

MIN 0.5 144 720

*** Presentation *** #网络状态的一些配置，无需修改

template = /usr/local/smokeping/etc/basepage.html.dist

+ charts

menu = Charts

title = The most interesting destinations

++ stddev

sorter = StdDev(entries=>4)

title = Top Standard Deviation

menu = Std Deviation

format = Standard Deviation %f

++ max

sorter = Max(entries=>5)

title = Top Max Roundtrip Time

menu = by Max

format = Max Roundtrip Time %f seconds

++ loss

sorter = Loss(entries=>5)

title = Top Packet Loss

menu = Loss

format = Packets Lost %f

++ median

sorter = Median(entries=>5)

title = Top Median Roundtrip Time

menu = by Median

format = Median RTT %f seconds

+ overview #overview 图片大小的设置

width = 600

height = 100

range = 1h

+ detail #detail 图片大小的设置

width = 600

height = 200

unison_tolerance = 2

"Last 3 Hours" 3h

"Last 30 Hours" 30h

"Last 10 Days" 10d

"Last 400 Days" 400d

#+ hierarchies

#+ owner

#title = Host Owner

#+ location

#title = Location

*** Probes ***

+ FPing

binary = /usr/local/sbin/fping

*** Slaves *** #从服务器设置，下章介绍

secrets=/usr/local/smokeping/etc/smokeping_secrets.dist

*** Targets *** #目标配置

probe = FPing

menu = Top

title = Network Latency Grapher

```
remark = Welcome to the SmokePing website of xxx Company. \
```

```
    Here you will learn all about the latency of our network.
```

```
+ Test                #使用+、++、+++可以配置多个菜单级别
```

```
menu= Targets
```

```
++ UnixHot
```

```
menu = UnixHot
```

```
title = UnixHot
```

```
alerts = someloss
```

```
host = www.unixhot.com
```

```
#####配置文件结束#####
```

2.9 启动 Smokeping

```
[root@Smokeping-S ~]# chown -R apache:apache /usr/local/smokeping/
```

```
[root@Smokeping-S ~]# 启动并制定日志文件
```

```
/usr/local/smokeping/bin/smokeping --logfile=/usr/local/smokeping/smokeping.log
```

第 3 章 Smokeping 分布式

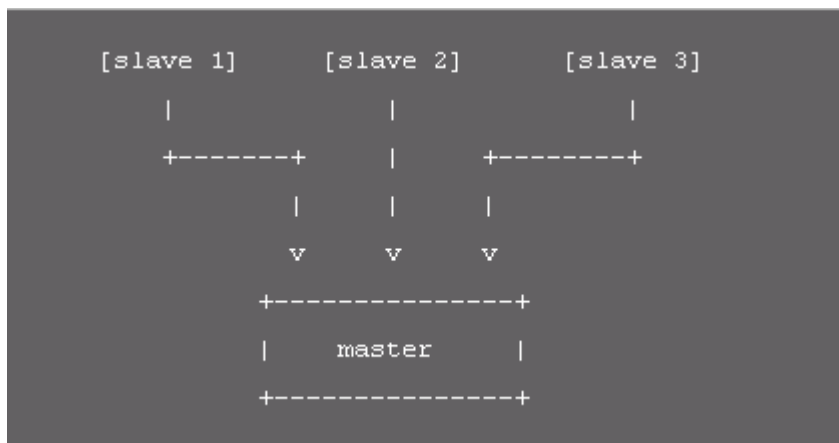
3.1 Smokeping 分布式介绍

我们在使用 Smokeping 的过程中，很容易就发现，如果从单个节点去探测网络性能，往往是不够的，Smokeping 提供了多节点的分布式部署功能，也就是可以从多个节点去探测到某个地区的网络状态，这样才是我们监控网络希望看到的情况。

Smokeping 称其为主从模式，主从模式的配置很简单，从服务器的安装和主服务器相同，只是从服务器不用自己的 config 文件，而是在启动的过程中请求主服务器的 config 文件，这样你只需要维护主服务器上的配置文件即可。

Smokeying 检测分布式的检测方式是被动模式，由从节点启动时获得主节点的 config 文件，然后进行测试，测试完毕后直接将数据提交给主节点。主从通信验证是通过

我们可以设置多个从服务器，主从服务器通过指定`--shared-secret=filename`来和主进行密码验证。



3.2 Smokeping 主从通信验证

Smokeping 主从验证时通过主节点和从节点/usr/local/smokeping/etc/smokeping_secrets 文件进行的，主节点和从节点的验证文件书写方式是不同的。

3.2.1 主节点 smokeping_secrets 文件的格式为：

从节点名称：密码

从节点名称是在从节点启动的时候制定的，smokeping_secrets 文件在 etc/config 文件中指定。

```
[root@Smokeping-S ~]# cd /usr/local/smokeping/etc/
[root@Smokeping-S etc]# vim smokeping_secrets
Smokeping-node1:unixhot:www:node1
[root@Smokeping-S etc]# chmod 600 smokeping_secrets
```

3.2.2 从节点 secrets 文件的格式为：

密码

从节点的 secrets 文件时在启动时指定，主节点和从节点设置的密码必须一致。并且这两个文件的权限必须为 600。

```
[root@Smokeping-node1 ~]# vim /usr/local/smokeping/secret
Unixhot:www:node1
[root@Smokeping-node1 ~]# chmod 600 /usr/local/smokeping/secret
```

3.3 主节点配置

主节点的配置就两步，一是指定并修改 smokeping_secrets 文件，添加 slave 用户和密码。二是修改 etc/config 文件，来进行检测。3.2.1 章节已经修改并指定好了验证文件，这里开始修改配置文件。

```
[root@Smokeping-S ~]# vim /usr/local/smokeping/etc/config
```

修改 SLave 如下：

*** Slaves ***

secrets=/usr/local/smokeping/etc/smokeping_secrets

+Smokeping-node1

display_name=node1

location=node1

color=00ff00

修改 Targets 如下:

*** Targets ***

probe = FPing

menu = Top

title = Network Latency Grapher

remark = Welcome to the SmokePing website of xxx Company. \

Here you will learn all about the latency of our network.

+ Test

menu= Targets

slaves = Smokeping-node1

++ UnixHot

menu = UnixHot

title = UnixHot

alerts = someloss

host = www.unixhot.com

slaves = Smokeping-node1

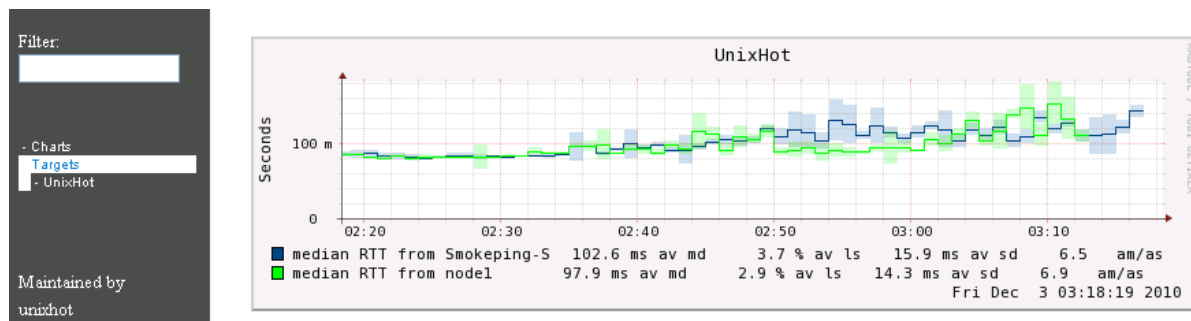
[root@Smokeping-S ~]# /usr/local/smokeping/bin/smokeping --restart

3.4 Slave 端部署

Slave 端部署和主服务器部署相同，都需要 rrdtool, fping, echoping 等组件，也需要修改 /usr/local/smokeping/htdocs/smokeping.cgi 文件和 /usr/local/smokeping/bin/smokeping 文件中两个库文件的位置。

```
[root@Smokeping-node1 ~]# mkdir /usr/local/smokeping/cache
[root@Smokeping-node1 ~]# /usr/local/smokeping/bin/smokeping \
--master-url=http://192.168.140.141/smokeping/smokeping.cgi \
--cache-dir=/usr/local/smokeping/cache \
--shared-secret=/usr/local/smokeping/secret \
--slave-name=Smokeping-node1
```

3.5 结果图



3.6 常见部署错误

在部署 Smokeping 的过程中，经常会出现各种各样的问题，现将曾经遇到的一些常见错误列出来。

- 1>库文件的位置配置不正确
- 2>smokeping 目录的拥有者需要和 apache 启动的用户是一致的。
- 3>主从节点的密码一致，权限必须是 600。
- 4>主机在主节点配置 slave 时指定的是 slave 的主机名，但是 slave 的主机名可以在启动从节

点的时候，用--slave-name 指定。

3.7 Smokeping 报警

Smokeping 自身也具备了报警功能，当网络出现故障时，根据你的配置进行报警。

第五部分 网站流量统计分析

网站流量分析对于运维人员来说，更是一门必须掌握的知识了，尤其是对于运维方向是门户网站、SNS、CMS、BBS 等，对于网站流量的统计和分析更显得及其重要，最直接的一个例子，通过对来路流量的统计和分析，可以了解我们在某个网站上的广告投入有没有收到预期的效果。

第 1 章 名词解释

提到网站流量，很多朋友都会被如下名词搞的头晕：SEO、PR 值，PV，UV 等等，如果你用搜索引擎搜索你会发现，很多人都在讲，而且讲的都不同，甚至有很多错误的解释误导读者。在本章节，就来探讨探讨它们的具体含义。

1.1 SEO

SEO (Search Engine Optimization) 搜索引擎优化, 是一种利用搜索引擎的搜索规则来提高目的网站在有关搜索引擎内的排名的方式。

SEO 就是通过总结搜索引擎的排名规律, 对网站进行合理优化, 使你的网站在百度和 Google 的排名提高, 如你在 Google 搜索“运维社区”四个字, 你会发现 UNIXHOT 运维社区是排在第一位的, 因为我进行了简单的 SEO 优化。

SEO 也是一种营销思路, 也是一个工作职位即: SEO 工程师, 主要负责公司网站在搜索引擎的排名优化, 来达到网站推广的目的。要想在网站推广中取得成功, 搜索引擎优化是最为关键的一项任务。

如果你想深入了解 SEO, 可以查找相关的资料, 也需要了解更多诸如: 内链、外链、关键字、关键字密度等等, 可以使用 Google 的网站管理员工具来进行网站的管理, 例如: 网站的外链, 搜索引擎抓取错误, 关键字查询等。

1.2 PR 值

PR 值 (PageRank), 网页的级别技术。取自 Google 的创始人 Larry Page, 它是 Google 排名运算法则 (排名公式) 的一部分, 用来标识网页的等级/重要性。级别从 0 到 10 级, 10 级为满分。PR 值越高说明该网页越受欢迎。例如: 一个 PR 值为 1 的网站表明这个网站不太具有流行度, 而 PR 值为 7 到 10 则表明这个网站非常受欢迎。经常一些网站的友情链接就要求你的 PR 值不低于 5。

1.3 PV 值

PV (page view) 即页面浏览量, 或点击量, 通常是衡量一个网站, 或者一条网络新闻的主要指标。一个访问者在 24 小时 (0 点到 24 点) 内到底看了你网站几个页面。需要注意的是: 同一个人浏览你网站同一个页面, 不重复计算 pv 量。pv 就是一个访问者打开了你网站的几个页面。PV 对于网站, 就像收视率之于电视, 从某种程度上已成为投资者衡量商业网站表现的最重要尺度。

下面截取了《开心网》介绍自己的一段文字, 大家可以看到 PV 的重要性:

“开心网提供的优质产品和服务深受用户的欢迎, 截止到 2010 年 9 月, 开心网注册用户已超越 9000 万, 页面周浏览量 (PageViews) 超过 80 亿, 月活跃用户量达到 5000 万, 居中国 SNS 网站第一名。”

而对于运维工程师，PV 值更是我们最关心的，因为每一个 PV 的产生就代表一次 HTTP 连接，在 Web 性能调优时必须关心的。

1.4 IP 值

IP 值，指访问某个站点或点击某条新闻不同的 IP 地址的人数，注意是不同的 IP 地址，如果你在网吧里面，共用一个出口也就是一个公网 IP 的话，那么网吧的所有人都访问该站点，访问的 IP 值只加 1。

1.5 UV 值

UV(unique visitor)，指访问某个站点或点击某条新闻的不同访客的人数。注意，是不同的访客，也就是客户端。UV 的引入就更准确的判断了站点的访客数量，这样上述的网吧例子，每一个访问者 IP 值加 1，但 UV 值就是最准确的访客数量值了。

第 2 章 Google Analytics

Google Analytics（分析）是企业级的网站分析解决方案。此工具不但可让您进一步了解网站流量和营销效果，现在还提供了富有灵活性又易于使用的强大功能，让您可以通过全新的方式查看并分析流量数据。有了 Google Analytics（分析），您可以更胸有成竹地撰写定位准确的广告、强化营销计划并提高网站的转化率。更重要的是该服务是免费的，赶快注册吧。（摘自官网）

目前已经有很多的公司采用 Google Analytics 而并非自己做流量监控分析。

详情见官方网站：<http://www.google.com/intl/zh-CN/analytics/index.html>

官方安装指南：

<http://www.google.com/support/analytics/bin/answer.py?hl=cn&answer=74932>

第 3 章 百度统计

百度统计是百度推出的一款专业网站流量分析工具，能够告诉您访客是如何找到并浏览您的网站，

以及如何改善访客在您网站上的使用体验，帮助您让更多的访客成为客户，不断提升网站的投资回报率。

百度统计提供了几十种图形化报告，全程跟踪访客的行为路径，并且帮助监控各种网络媒介推广效果，让您及时了解哪些关键词、哪些创意的效果最好。同时，百度统计集成百度推广数据，帮助您及时了解百度推广效果并优化推广方案。

基于百度强大的技术实力，百度统计提供了丰富的数据指标，系统稳定，功能强大但操作简易。登陆系统后按照系统说明完成代码添加，百度统计便可马上收集数据，为您提高投资回报率提供决策依据。（摘自官网）

详情见官方网站：<http://tongji.baidu.com/hm-web/welcome/login>

```
[root@Monitor-Server ~]# cd /usr/local/src
[root@Monitor-Server src]# wget
http://www.splitbrain.org/\_media/projects/dokuwiki/dokuwiki-2010-11-07a.tgz
[root@Monitor-Server src]# tar zxvf dokuwiki-2010-11-07a.tgz
```

第六部分 运维社区开源运维平台

附录：GFDL 协议

(注：可以参考本协议的中文翻译版本 <http://www.thebigfly.com/gnu/FDLv1.3/>)

GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein.

The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image

formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version

gives permission.

- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the

Modified Version.

- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same

adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their

copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (C) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with ... Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

实验答疑: <http://www.unixhot.com> <http://www.bosshot.com>