Linux 运维趋势

第十期

2011年7月号

关键字：log、安全审计、数据库、防火墙

日志分析技巧分享

内容目录

杂志策划：51CTO 系统频道

本期主编：李晶

责任编辑：杨赛

封面制作：徐泽琼

交流圈子：
http://g.51cto.com/linuxops

专题页面邮件订阅入口：
http://os.51cto.com/art/201011/233915.htm

下载汇总页：
http://down.51cto.com/zt/71

投稿信箱：
yangsai@51cto.com

我觉得要在运维方面做的比较出色的话，网络，系统，数据库必须都要会。否则的话，想做一个管理人员是比较有难度的。

饭统网运维主管葛海龙：从 IDC 机房到高并发网站

采访/杨赛

人物简介：

葛海龙（龙哥），饭统网运维主管,曾任职于北京铜牛机房，瑞盛 IDC 运维主管。技术特长：网络监控、Linux 系统管理、高并发网站运维、负载均衡、Cisco/H3C 设备等，目前关注 Oracle 数据库。

个人博客：http://gehailong.blog.51cto.com/

饭统网是典型的餐饮服务类网站，目前日均 pv 在 500 万左右。在本期《趋势》访谈中，我们邀请到了饭统网运维主管葛海龙（龙哥）来分享一下他的运维经验。

51CTO：首先，简单的介绍一下您自己吧。您是什么时候开始从事运维的工作的？

葛海龙：我在山东师范大学的计算机系毕业，之后在北京 ITET 学了一段时间的网络和 Linux，这才开始从事运维的工作。

一开始是在 IDC 工作，期间帮朋友做过一些高并发网站的搭建和运维。到今天差不多已经有 5 年的时间了。

51CTO：那您在 IDC 和网站都做过，也都做到了运维主管这个级别。您做到运维主管用了几年时间？

葛海龙：今年算是正式的运维主管吧，因为原来虽然称为运维主管，但没做什么事情。

51CTO：可以理解为在网站的运维工作比 IDC 复杂么？可不可以简单说一下您在瑞盛 IDC，相比现在在饭统网的工作，都分别关注哪些方面呢？

葛海龙：可以这样认为。

网站运维有很多东西需要考虑，比如网站的负载，缓存的处理，日志的分析，数据的备份/恢复等等。

在 IDC 主要是保证网络的畅通，监控是非常重要的一部分。因为现在的机房一般都是双线，甚至多线机房。要保证每条链路都不能出现问题。

再就是流量的监控，如果某条链路带宽突然增大，要迅速找出原因，否则会影响其他客户。

再就是链路的冗余备份。

总之，IDC 主要关注网络和硬件，而网站需要关注的东西就更多一些。

51CTO：那就聊聊网站的事吧。您搭建维护过这些高并发网站，包括现在的饭统网，日均 pv 应该也是百万量级的。能否介绍下饭统网当前大概的架构，以及用到了哪些技术？

葛海龙：饭统网现在访问量在每天 500 万左右的 pv。

大体架构可以分为三层：第一层，也就是前端是缓存服务器，用 squid 做的；第二层是 web，第三层是存储。

用的技术也比较丰富，比如 web 主要是 nginx，只有一个服务器还在使用 apache；缓存的技术有 squid 做静态缓存，memcached 做数据库内容的缓存；负载均衡用的是 lvs 的 DR 方式；数据库 mysql，oracle 都有，它们也都做的负载均衡；dns 也是用的我们自己搭建的 dns，也做了 master 和 slave；监控是用的 cacti 和 nagios，用飞信来报警。

51CTO：dns 是自己写的？还是用 bind 修改的？

葛海龙：bind 做的，没有实力自己写 dns 软件，呵呵。因为我们域名比较多，要是让公网上专门做 dns 厂商做的话，解析的时间是个问题。用自己的 dns，只要本地没有该记录的缓存，几分钟内就可以在全球生效。

51CTO：原来如此。多域名是指地方分站多吗？

葛海龙：对。有几百个城市的分站。服务器都在北京，然后有 cdn 加速。再就是前端有缓存服务器。

51CTO：大致了解了，十分感谢。最后，再谈谈您自己的成长经历吧。我看您的博客，主要有两个方面，一个是 Linux 系统和应用服务，一个就是网络设备。您自己制定过什么学习/进阶计划么？

葛海龙：首先，系统和网络是肯定要学的，而且必须要学好。最近这段时间我在学习 oracle。

我觉得要在运维方面做的比较出色的话，网络，系统，数据库必须都要会。否则的话，想做一个管理人员是比较有难度的。

51CTO：对其他 Linux 运维有什么建议吗？比如您现在招聘 Linux 运维的时候，最关注他们的哪些素质？

葛海龙：我在招聘 linux 的时候主要关心以下几个方面：1.基本功必须好，2.不浮躁，遇事必须沉着，不慌张，3.最好能有 2 年左右的运维经验。

51CTO：感谢海龙的分享！本次内容到此结束。如果您有什么问题想要沟通，或者希望听到某位运维进行分享，欢迎留言交流。▌※

本文为节选，原文：
http://os.51cto.com/art/201106/272523.htm

推荐阅读：

专访田逸：运维这十年 手边的那些工具

对于企业来说 免费的 DNS 合适吗？

Linux 下的 DNS 安全保障十大技巧

专题：可怕的服务器并发

图解：VNC Server 攻击实战案例

系统管理员需要掌握哪些软技能？

面向系统管理员的 iPad 应用推荐

有高效批量上传大量数据的需求？我们还是得找点有对应功能的产品。大多数产品都无法胜任，因为它们不支持批量操作。

NoSQL 数据库选型：35 个参考实现

文/Todd Hoff
编译/核子可乐

我们曾经讨论过“到底 NoSQL 能在我们的工作中发挥什么作用？”

现在我们改变目标，转而思索哪些具体应用因素会对选择产生影响以及哪种系统在应对此类因素时更加适用。

你有什么意见？

首先，我们先来聊聊各类数据模型。

文档类数据库

传承:受 Lotus Notes 启发而来。

数据模型:文档汇总，包括键-值汇总。

实例: CouchDB, MongoDB

优势: 数据建模自然、程序员易于上手、开发流程短、兼容网页模式、便于达成 CRUD（即添加、查询、更新及删除的简称）。

图形类数据库

传承:来自 Euler 及图形理论。

数据模型:节点及关系，二者结合能够保持键-值间的成对状态

实例: AllegroGraph, InfoGrid, Neo4j

优势:轻松玩转复杂图形问题、处理速度快

关系类数据库

传承:源自 E. F. Codd 在大型共享数据库中所提出的数据关系模型理论

数据模型:以关系组为基础

实例: VoltDB, Clustrix, MySQL

优势:性能强大、联机事务处理系统扩展性好、支持 SQL 访问、视图直观、擅长处理交易关系、与程序员间的交互效果优异

面向对象类数据库

传承:源自图形数据库方面的研究成果

数据模型: 对象

实例: Objectivity, Gemstone

优势:擅长处理复杂的对象模型、快速的键-值访问及键-功能访问并且兼具图形数据库的各类功能

键-值存储

传承: Amazon Dynamo 中的 paper 概念及分布式 hash 表

数据模型:对成对键-值的全局化汇总

实例: Membase, Riak

优势:尺寸掌控得当、擅长处理持续的小规模读写需求、速度快、程序员易于上手

BigTable Clones

传承自:谷歌 BigTable 中的 paper 概念

数据模型:纵列群，即在某个表格模型中，每行在理论上至少可以有一套单独的纵列配置

实例: HBase, Hypertable, Cassandra

优势:尺寸掌控得当、擅长应对大规模写入负载、可用性高、支持多数据中心、支持映射简化

数据结构类服务

传承: 不明

实例: Redis

数据模型: 执行过程基于索引、列表、集合及字符串值

优势:为数据库应用引入前所未有的新鲜血液

网格类数据库

传承:源自数据网格及元组空间研究

数据模型:基于空间的构架

实例: GigaSpaces, Coherence

优势:优良的性能表现及上佳的交易处理扩展性

我们该为自己的应用程序选择哪套方案？

选择的关键在于重新思考我们的应用程序如何依据不同数据模型及不同产品进行有针对性的协同工作。即用正确的数据模型处理对应的现实任务、用正确的产品解决对应的现实问题。

将应用实例中的客观需求与我们的选择联系起来。这样大家就能够逆向分析出我们的基础架构中适合引入哪些产品。至于具体结论是 NoSQL 还是 SQL，这已经不重要了。

关注数据模型、产品特性以及自身需要。产品总是将各种不同的功能集中起来，因此我们很难单纯从某一类数据模型构成方式的角度直接找到最合用的那款。

如果我们的应用程序需要...

复杂的交易：因为没人愿意承受数据丢失，或者大家更倾向于一套简单易用的交易编程模式，那么请考虑使用关系类或网格类数据库。

若是以扩展性为优先，那么 NoSQL 或 SQL 都能应对自如。这种情况下我们需要关注那些支持向外扩展、分类处理、实时添加及移除设备、负载平衡、自动分类及整理并且容错率较高的系统。

要求持续保有数据库写入功能，则需要较高的可用性。在这种情况下不妨关注 BigTable 类产品，其在一致性方面表现出众。

如有大量的小规模持续读写要求，也就是说工作负载处于波动状态，可以关注文档类、键-值类或是那些提供快速内存访问功能的数据库。引入固态硬盘作为存储媒介也是不错的选择。

以社交网络为实施重点的话，我们首先想到的就是图形类数据库；其次则是 Riak 这种关系类数据库。Redis 的集合及列表操作也能发挥作用。

如果我们的应用程序需要...

在访问模式及数据类型多种多样的情况下，文档类数据库比较值得考虑。这类数据库不仅灵活性好，性能表现也可圈可点。

需要完备的脱机报告与大型数据集的话，首选产品是 Hadoop，其次则是支持映射简化的其它产品。不过仅仅支持映射简化还不足以提供如 Hadoop 一样上佳的处理能力。

如果业务跨越数个数据中心，Bigtable Clone 及其它提供分布式选项的产品能够应对由地域距离引起的延迟现象，并具备较好的分区兼容性。

要建立 CRUD 应用程序，首选文档类数据库。这类产品简化了从外部访问复杂数据的过程。需要内置搜索功能的话，推荐 Riak。

要对数据结构中的诸如列表、集合、队列及发布/订阅信息进行操作，Redis 是不二之选。其具备的分布式锁定、覆盖式日志及其它各种功能都会在这类应用状态下大放异彩。

将数据以便于处理的形式反馈给程序员（例如以 JSON、HTTP、REST、Javascript 这类形式），文档类数据库能够满足这类诉求，键-值类数据库效果次之。

如果我们的应用程序需要...

支持辅助索引，以便通过不同的关键词查找数据，这要由关系类数据库及 Cassandra 推出的新辅助索引系统共同支持才能实现。

创建一套处于不断增长中的数据集合（真正天文数量级的数据）然而访问量却并不大，那么 Bigtable Clone 是最佳选择，因为它会将数据妥善安排在分布式文件系统当中。

需要整合其它类型的服务并确保数据库提供延后写入同步功能？那最好的实现方式是捕捉数据库的各种变化并将其反馈到其它系统中以保障运作的一致性。

通过容错性检查了解系统对供电中断、隔离及其它故障情况的适应程度。

尝试在移动平台上工作并关注 CouchDB 及移动版 couchbase。

哪种方案更好？

25%的状态改善尚不足以让我们下决心选择 NoSQL。

选择标准是否恰当取决于实际情况。这类标准对你的方案有指导意义吗？

如果你的公司尚处于起步阶段，并且需要尽快推出自己的产品，这时不要再犹豫不决了。无论是 SQL 还是 NoSQL 都可以作为参考。

性能表现在一台主机上来说也许差别并不大，但如果我们要将其部署在 N 多台主机上呢？

世上万物皆不完美，如果大家常逛 Amazon 论坛就会发现上面的 EBS 响应缓慢，当然没准我这属于特例。不过 GAE 的数据存储体系响应也很缓慢，有时甚至干脆显示红叉。每种我们使用着的产品都存在诸多问题，但对于自己亲手选择的方案，你能接受它所存在的问题吗？▌※

本文为节选，完整内容请参考英文原文：
http://highscalability.com/blog/2011/6/20/35-use-cases-for-choosing-your-next-nosql-database.html

译文全文：
http://database.51cto.com/art/201107/274036.htm

编辑推荐：

NoSQL 内战：MongoDB 与 CouchDB 对比

关于 NoSQL 数据库你应该知道的 10 件事

八卦，趣闻与数字 2011.06 – 2011.07

Linux 在华 12 年 基于 ARM 的 Linux 一团糟

收集整理/51CTO 系统频道

【12】12 年前，国内首次最高级别的“Linux 技术研讨会“(1999 年 7 月 15 日)召开，随后国家正式下达政策文件，开始了中国式的 Linux 长征。

http://os.51cto.com/art/201106/270023.htm

【Red Hat Enterprise MRG】红帽 7 月 1 日宣布正式提供 Red Hat Enterprise MRG 2.0，包括该产品在消息、实时和网格组件等领域的扩展能力。

http://os.51cto.com/art/201107/272709.htm

【Ubuntu 软件中心】新版的 Ubuntu software center 5.0 将有新的用户界面，设计灵感来源于目前流行的各类 App Store 。

http://os.51cto.com/art/201106/269401.htm

【Debian】6 月 25 日，Debian6.0 发布它的第二个稳定版本。此更新主要增加了对安全问题的修正，以稳定版本发布，调整和修复了伴随着的严重的问题。建议尽快进行升级。

http://os.51cto.com/art/201106/271408.htm

【Linux Mint】Linux Mint 团队 7 月 1 日发布了 Linux Mint 11 LXDE RC 版，预计正式版将会在一个月后的时间发布。

http://os.51cto.com/art/201107/272698.htm

【LibreOffice】7 月 4 日，LibreOffice 发布最新版本 3.4.1。

http://os.51cto.com/art/201107/273006.htm

【Linux Deepin 11.06】Linux Deepin 11.06 于 7 月 4 日正式发布。Linux Deepin 11.06 基于 Ubuntu 11.04，开发了自己的软件中心。

http://os.51cto.com/art/201106/266912.htm

【亚太证交所 x5】新加坡证券交易所(SGX)、菲律宾证券交易所(PSE)、巴基斯坦拉合尔证券交易所(LSE)、印度国家商品及衍生品交易所(NCDEX)、香港证券交易所以及东京证券交易所(TSE)现在均使用红帽企业级 Linux 支持其交易平台。

http://os.51cto.com/art/201107/272801.htm

【一团糟】基于 ARM 芯片的 Linux 系统已经彻底分化了，并且只有一部分专业的开发者和制造商明白到底发生了什么。

http://os.51cto.com/art/201107/272978.htm

【IT 这十年】 时间总是匆匆的从我们身边流过，在这里我们回看一下近十年间操作系统与浏览器的格局变化，本专题将会为大家展示系统、浏览器、运维工具、办公软件以及即时通讯工具的演变。

http://os.51cto.com/exp/ittenyears/

本期专题：日志分析技巧分享

对于一个 Linux 的系统管理员来说，系统日志可谓是相当重要，因为你可以从日志中看到系统做的任何一件事情。

这是一个从其他主机收集日志，并将它们存放在同一个地方的系统，很容易使来自多个主机的日志条目关联起来，对其进行统一管理、分析，甚至配合自动化工具进行实时的监控。

日志主机系统概述：部署与监控

文/李晨光

操作系统的日志主要具有审计与监测的功能，通过对日志信息的分析，可以检查错误发生的原因，监测追踪入侵者及受到攻击时留下的痕迹，甚至还能实时的进行系统状态的监控。有效利用日志信息并对其进行分析与实时的监控管理，对于系统的安全性具有极为重要的作用。

对于日志信息的管理通常采用两种方法，一种方法是不同服务器的日志信息都存放在各自系统内，系统管理员对各服务器进行分散管理。另一种方法则是使用日志主机系统，这是一个从其他主机收集日志，并将它们存放在同一个地方的系统，很容易使来自多个主机的日志条目关联起来，对其进行统一管理、分析，甚至配合自动化工具进行实时的监控，有效提高管理的效率。

第一种方法往往是大多数系统管理员的常用的方法，这种传统的管理方法在服务器数量较少时还能勉强应付，但在处理多主机状况时却并非一种有效的方法。本文主要讲述二种日志管理方法，探寻一种提高系统管理效率的途径。

一、日志主机系统的部署

日志主机系统包括日志主机及各主机系统两个部分，其中日志主机相当于服务器端，而各主机系统相当于客户端，将日志信息实时的传送到日志主机上来。

1. 日志主机的部署

日志主机采用一台 RHEL 5 的服务器(假设其主机名为 loghost),日志收集软件采用 Linux 平台上的 Syslog,Syslog 一般都随 Linux 系统安装时已经安装，对于我们部署整个系统提供了极大的便利性，因此在此不对其安装步骤进行阐述，仅讲述其配置方法。

Syslog 既可作为客户端，也可作为服务器端，并且支持远程的日志收集。其配置文件为 /etc/sysconfig/syslog，要配置其作为服务器端，需对此配置文件相应部分改为如下所示：

```
SYSLOGD_OPTIONS="-r-m 0"
```

“-r”选项表示使 syslog 接收客户端的远程日志信息。

接下来重启 Syslog 服务器端使配置生效：

```
#service syslogd restart
```

由于 Syslog 采用 514 端口监听来自各客户端的日志信息，因此需要在日志主机的防火墙上开放 514 端口，以 iptables 为例，对特定网段开放 514 端口：

```
/sbin/iptables -A INPUT -i eth0 -p tcp
 -s 192.168.0.0/16 -dport 514 -syn -j
 ACCEPT
```

2.客户端的部署

2.1 Linux 平台下客户端的部署

在 Linux 平台下依然选择 syslog 作为客户端进行部署，此时此配置文件为 /etc/syslog.conf, 其默认配置为（仅以/var/log/message 日志为例）：

```
*.info;mail.none;authpriv.none;cron.no
 ne /var/log/messages
```

/var/log/message 即 Sysolg 存放系统日志的绝对路径，将此值替换为日志主机名即可。例子如下：

```
*.info;mail.none;authpriv.none;cron.no
 ne @loghost
```

根据上述配置，当 Syslog 重启使用配置生效后，客户端服务器的日志信息将会实时的传送到日志主机的 /var/log/message 文件里，对各服务器的日志信息进行统一的管理。

使用如下命令重启 Syslog 服务使配置生效：

```
#service syslogd restart
```

依上述方法将其他系统日志信息(如/var/log/secure)导入到日志主机上。

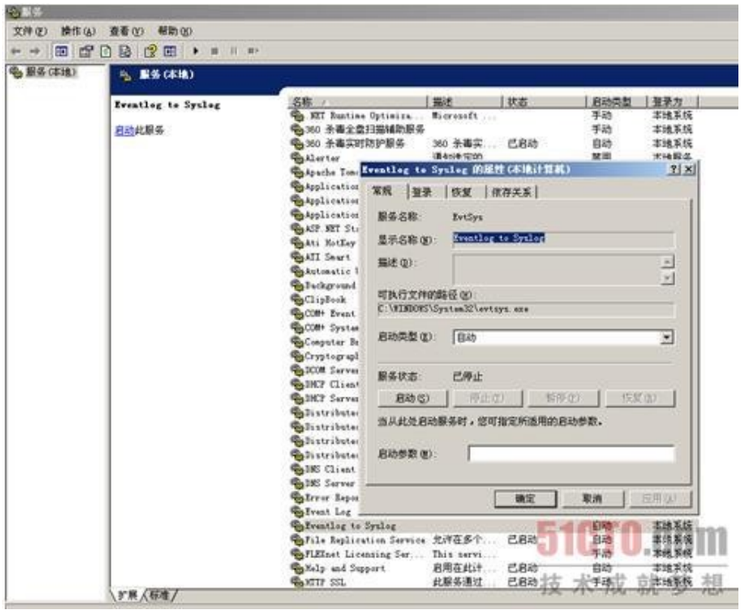

图 1 evtsys 在服务列表中

笔者建议，采用添加配置而非修改的方法，同时在本地及日志主机上保存系统日志。

2.2 Windows 平台下客户端的部署

在 windows 平台下采用软件 evtsys 进行客户端的部署，其下载链接为

http://code.google.com/p/eventlog-to-syslog/downloads/list

解开后得到两个文件：evtsys.ext 和 evtsys.dll。将这两个文件放到

C：\WINDOWS\system32 目录下，在命令行状态下运行如下命令进行安装：

```
%systemroot%\system32\evtsys -i -h
 loghost
```

当安装成功后，可查看服务列表看到相应的信息，如图 1 所示。

卸载 evtsys：

```
%systemroot%\system32\evtsys -u
```

更改日志主机名：

```
Net stop evtsys
evtsys -u
evtsys -l -h newloghost
net start evtsys
```

二、日志主机的自动日志分析与监控

当整个系统部署好后，可以从日志主机里验证各服务器是否将日志信息发送到了日志主机上。以/var/log/message 为例，打开此文件，节选部分日志如下：

```
Jan 9 08:39:38 dog crond(pam_unix)
 [4528]:ses-sion opened for user root
 by (uid=0)
Jan 9 08:39:36 dog crond(pam_unix)
 [4528]:session closed for user root
Jan 9 08:39:40 panda crond(pam_unix)
 [20296]:ses-sion opened for user root
 by(uid=0)
Jan 9 08:39:40 panda crond(pam_unix)
 [20296]:ses-sion closed for user root
```

```
Jan 9 08:39:53 app last message
 repeated 8 times
Jan 9 08:40:11 apple net-
 snmp[657]:Connection from
 udp:92.168.1.11:4298
Jan 9 08:40:11apple net-
 snmp[657]:Received SNMP packet(s)
 from udp:159.226.2.144:42988
Jan 9 08:41:15orangesshd(pam_unix)
 [28389]:ses-sion opened for user tom
 by(uid=2011)
Jan 9 08:41:28 orange sshd(pam_unix)
 [28389]:ses-sion opened for user tom
 by (uid=2011)
Jan 9 08:41:28 orange 1月9 08:41:28
 su` (pam_unix)[28425]:session opened
 for user root by tom (uid=2011)
```

对于如此庞大的日志信息，大部分并没多大的用处，但在跟踪某一具体问题或者安全漏洞时却可能很有用。那么我们如何对其进行有效地分析与监测，发挥其真正作用呢?在此推荐两款比较常用的日志分析与监控软件，对这些日志信息进行自动地分析与监控。

1. 利用 Logwatch 进行日志监控

linux 中，已经默认安装了 Logwatch，配合 Sendmail 的邮件发送功能，向系统管理员发送前一天的日志分析结果邮件。其配置文件为/etc/log.d/logwatch.conf，下面是省略注释后的配置文件，一般只需将 MailTo 部分改为系统管理员邮箱地址即可。

```
LogDir = /var/log
MailTo = admin@local.com
Print = No
Range = yesterday
Detail = High
Service = All
```

2. 利用 Swatch 进行日志的实时监控

Swatch 下载链接为
http://sourceforge.net/projects/swatch/

要安装 Swatch，需要先安装两个 perl 模块包：Date-Calc-5.4.tar.gz 和 TimeDate-1.16.tar.gz 接着安装 Swatch,安装步骤如下：

```
#tar-zxvf swatch-3.2.1.tar.gz
#cd swatch-3.2.1
#perl Makefile.pl
#make
#make test
#make install
```

配置 Swatch 使其作，需建立配置文件~/.swatchrc,按照其语法规则添加监测的相关内容，可使用“man swatch”命令查看具体配置内容及含义。

使用“swatch--help”查看 Swatch 运行时的具体选项。

当出现监控到的信息时，Swatch 即会实时地发送邮件给系统管理员，及时杜绝入侵者的各种入侵尝试，保护系统的安全。

3. 总结

日志主机系统的建立能够有效提高日志管理、分析及监测的效率，同时它也对于日志信息的安全保护起到了极为重要的作用。▌※

原文：
http://os.51cto.com/art/201101/243450.htm
作者近期文章推荐：
VNC Server 模拟攻击实战
暴力破解 FTP 服务器技术探讨与防范措施
Fedora 14 炫酷应用新体验
提高 IIS 的 FTP 安全性 管理员的九阴真经
国内云存储产品应用简介
Linux 下搭建 JSP 环境

虽然 Ubuntu 的官方软件源中已经加入了 Puppet，但是官方源中的 Puppet 版本太老，不但很多新功能没有，而且旧版本还可能有 bug

善用脚本 让你的 Nagios 记录系统监控日志

文/抚琴煮酒

Nagios 报警功能很强大，但有时我们有这个需求，特别是系统繁忙时希望能留下日志，以供分析：到底是受到了攻击，还是开发人员设置不当，亦或是运维人员改动了系统配置等。机器少时可能问题不大，但公司的 CDN 服务器集群是一百多台，目前看形势还在增长，所以我想设计一个 shell 脚本来作 Nagios 的补充，在系统繁忙时分离出日志，供同事们一起分析问题，得出问题的症结所在。

这里介绍下以 vmstat 为基础的系统监控脚本/root/monitor.sh

此脚本设计思想与功能实现：

① 此脚本设计为 Nagios 监控补遗，Nagios 是即时监控服务器状态并即时报警，但美中不足的不能记录其状态及日志，所以设计此脚本；

②此脚本已在 FreeBSD 上成功调试运行，亦适用于 RHEL/Centos 系统；

③这里以常用生产服务器 HPDL380G6(英特尔至强 E5540@2.53GHz 双四核)为依据，r 的阀值为 4。

脚本内容如下

```
#!/bin/bash
while :
do
vmr=`vmstat | tail -1 \
| awk '{print $1}'`
if [ ${vmr} -gt 4 ]
then
date    >> /root/monitor.txt
vmstat >> /root/monitor.txt
netstat -anp >> /root/monitor.txt
ps -aux >> /root/monitor.txt
last    >> /root/monitor.txt
tail -10 /var/log/messages >> \
 /root/monitor.txt
fi
sleep 60
done
```

此脚本可放至后台运行

```
sh /root/monitor.sh &
```

如遇 CPU 繁忙的情况，它会自动记载系统日志等以供分析。

用 vmstat 监视内存使用情况

vmstat 是对系统的整体情况进行统计，不足之处是无法对某个进程进行深入分析。

```
vmstat [-V] [-n] [delay [count]]
```

其中，-V 表示打印出版本信息；-n 表示在周期性循环输出时，输出的头部信息仅显示一次；delay 是两次输出之间的延迟时间；count 是指按照这个时间间隔统计的次数。对于 vmstat 输出各字段的含义，可运行 man vmstat 查看。

vmstat 命令有四个可选标志可供使用。如果机器有虚拟地址缓存-c 标志就改变输出报

告缓存刷新统计数据。报告包括自从系统启动后每种缓存刷新全部总量。六个缓存类型是用户，上下文，区域，段，页，部分页。

-i 标志 使输出变为报告中断的数量。如果给出设备名，如 d1,d2 等，监控将在设备级*执行，并报告每个给定设备的统计信息。

修改"普通"报告来显示交换而非页面调度活动的信息。这选项改变显示的两个字段：si(换入)和 so(换出)替代了 re 和 mf 字段。

通过 VMSTAT 识别 RAM 瓶颈

数据库服务器都只有有限的 RAM，出现内存争用现象是 Oracle 的常见问题。

首先察看 RAM 的数量，命令如下：

```
[root@brucelau root]#free
             total       used       free
 shared     buffers     cached
Mem:       1027348     873312
 154036      185736     187496
 293964
-/+ buffers/cache:        391852
 635496
Swap:      2096440          0
 2096440
```

当然可以使用 top 等其他命令来显示 RAM。

当内存的需求大于 RAM 的数量，服务器启动了虚拟内存机制，通过虚拟内存，可以将 RAM 段移到 SWAP DISK 的特殊磁盘段上，这样会出现虚拟内存的页导出和页导入现象，页导出并不能说明 RAM 瓶颈，虚拟内存系统经常会对内存段进行页导出，但页导入操作就表明了服务器需要更多的内存了，页导入需要从 SWAP DISK 上将内存段复制回 RAM，导致服务器速度变慢。

解决的办法有几种：

1. 最简单的，加大 RAM
2. 改小 SGA，使得对 RAM 需求减少
3. 减少 RAM 的需求（如：减少 PGA）

▌※

本文有删节，原文：
http://os.51cto.com/art/201006/203215.htm

编辑推荐：

Linux 系统监控工具之 vmstat 详解

users 用单独的一行打印出当前登录的用户，每个显示的用户名对应一个登录会话。如果一个用户有不止一个登录会话，那他的用户名将显示相同的次数。

Linux 日志管理五大命令详解

文/李洋

从 Ext 到 Ext2，从 Ext2 再到 Ext3，乃至 Ext4 或者更高版本，Linux 系统历来以强大、丰富和完整的日志系统著称。通过管理日志，可以清晰地了解系统的运行状况，也能从各种蛛丝马迹中发现入侵和快速地阻止入侵。

Linux 中提供了异常日志，并且日志的细节是可配置的。Linux 日志都以明文形式存储，所以用户不需要特殊的工具就可以搜索和阅读它们。还可以编写脚本，来扫描这些日志，并基于它们的内容去自动执行某些功能。Linux 日志存储在/var/log 目录中。这里有几个由系统维护的日志文件，但其他服务和程序也可能会把它们的日志放在这里。大多数日志只有 root 账户才可以读，不过修改文件的访问权限就可以让其他人可读。在 Linux 系统中，有四类主要的日志：

◆连接时间日志：由多个程序执行，把记录写入到/var/log/wtmp 和/var/run/utmp，login 等程序更新 wtmp 和 utmp 文件，使系统管理员能够跟踪谁在何时登录到系统。

◆进程统计：由系统内核执行。当一个进程终止时，为每个进程往进程统计文件（pacct 或 acct）中写一个记录。进程统计的目的是为系统中的基本服务提供命令使用统计。

◆错误日志：由 syslogd（8）守护程序执行。各种系统守护进程、用户程序和内核通过 syslogd（3）守护程序向文件

/var/log/messages

报告值得注意的事件。另外有许多 UNIX 程序创建日志。像 HTTP 和 FTP 这样提供网络服务的服务器也保持详细的日志。

◆实用程序日志：许多程序通过维护日志来反映系统的安全状态。su 命令允许用户获得另一个用户的权限，所以它的安全很重要，它的文件为 sulog。同样的还有 sudolog。另外，诸如 Apache 等 Http 的服务器都有两个日志：access_log（客户端访问日志）以及 error_log（服务出错日志）。 FTP 服务的日志记录在 xferlog 文件当中，Linux 下邮件传送服务（sendmail）的日志一般存放在 maillog 文件当中。

utmp、wtmp 日志文件是多数 Linux 日志子系统的关键，它保存了用户登录进入和退出的记录。有关当前登录用户的信息记录在文件 utmp 中；登录进入和退出记录在文件 wtmp 中；数据交换、关机以及重启的机器信息也都记录在 wtmp 文件中。所有的记录都包含时间戳。时间戳对于日志来说非常重要，因为很多攻击行为分析都与时间有极大的关系。这些文件在具有大量用户的系统中增长十分迅速。

例如 wtmp 文件可以无限增长，除非定期截取。许多系统以一天或者一周为单位把 wtmp 配置成循环使用。它通常由 cron 运行的脚本来修改。这些脚本重新命名并循环使用 wtmp 文件。通常，wtmp 在第一天结束后命名为 wtmp.1；第二天后 wtmp.1 变为 wtmp.2 等等，用户可以根据实际情况来对这些文件进行命名和配置使用。

utmp 文件被各种命令文件使用，包括 who、w、users 和 finger。而 wtmp 文件被程序 last 和 ac 使用。

wtmp 和 utmp 文件都是二进制文件，他们不能被诸如 tail 命令剪贴或合并（使用 cat 命令）。

用户需要使用 who、w、users、last 和 ac 来使用这两个文件包含的信息。

1．who 命令

who 命令查询 utmp 文件并报告当前登录的每个用户。Who 的缺省输出包括用户名、终端类型、登录日期及远程主机。使用该命令，系统管理员可以查看当前系统存在哪些不法用户，从而对其进行审计和处理。例如：运行 who 命令显示如下所示：

```
# who
root     pts/1        2010-02-22 13:02
 (:0.0)
root     pts/2        2010-02-22 15:57
 (:0.0)
root     pts/3        2010-02-22 15:57
 (:0.0)
```

如果指明了 wtmp 文件名，则 who 命令查询所有以前的记录。命令 who /var/log/wtmp 将报告自从 wtmp 文件创建或删改以来的每一次登录。例如：运行该命令如下所示：

```
# who /var/log/wtmp
root     :0           2010-01-24 21:47
root     pts/1        2010-01-24 21:47
 (:0.0)
root     :0           2010-02-20 19:36
root     pts/1        2010-02-20 19:36
 (:0.0)
root     :0           2010-02-21 15:21
root     pts/1        2010-02-21 15:56
 (:0.0)
root     pts/2        2010-02-21 16:03
 (:0.0)
root     :0           2010-02-22 13:01
root     pts/1        2010-02-22 13:02
 (:0.0)
root     pts/2        2010-02-22 15:57
 (:0.0)
root     pts/3        2010-02-22 15:57
 (:0.0)
```

2．users 命令

users 用单独的一行打印出当前登录的用户，每个显示的用户名对应一个登录会话。如果一个用户有不止一个登录会话，那他的用户名将显示相同的次数。运行该命令将如下所示：

```
# users
root root root
```

3．last 命令

last 命令往回搜索 wtmp 来显示自从文件第一次创建以来登录过的用户。系统管理员可以周期性地对这些用户的登录情况进行审计和考核，从而发现起中存在的问题，确定不法用户，并进行处理。运行该命令，如下所示：

```
# last
root     pts/3          :0.0
 Mon Feb 22 15:57   still logged in
root     pts/2          :0.0
 Mon Feb 22 15:57   still logged in
root     pts/1          :0.0
```

```
 Mon Feb 22 13:02   still logged in
root     :0
 Mon Feb 22 13:01   still logged in
reboot   system boot  2.6.18-8.el5
 Mon Feb 22 12:56          (03:02)
root     pts/2         :0.0
 Sun Feb 21 16:03 - down   (02:37)
root     pts/1         :0.0
 Sun Feb 21 15:56 - down   (02:45)
root     :0
 Sun Feb 21 15:21 - down   (03:20)
reboot   system boot  2.6.18-8.el5
 Sun Feb 21 15:19          (03:22)
root     pts/1         :0.0
 Sat Feb 20 19:36 - down   (01:50)
root     :0
 Sat Feb 20 19:36 - down   (01:51)
reboot   system boot  2.6.18-8.el5
 Sat Feb 20 19:34          (01:53)
root     pts/1         :0.0
 Sun Jan 24 21:47 - down   (00:02)
root     :0
 Sun Jan 24 21:47 - down   (00:02)
reboot   system boot  2.6.18-8.el5
 Sun Jan 24 21:45          (00:05)
reboot   system boot  2.6.18-8.el5
 Sun Jan 24 21:41          (00:02)

wtmp begins Sun Jan 24 21:41:03 2010
```

读者可以看到，使用上述命令显示的信息太多，区分度很小。所以，可以通过指明用户来显示其登录信息即可。例如：使用 last reoot 来显示 reboot 的历史登录信息，则如下所示：

```
# last reboot
reboot   system boot  2.6.18-8.el5
 Mon Feb 22 12:56          (03:07)
reboot   system boot  2.6.18-8.el5
 Sun Feb 21 15:19          (03:22)
reboot   system boot  2.6.18-8.el5
 Sat Feb 20 19:34          (01:53)
reboot   system boot  2.6.18-8.el5
 Sun Jan 24 21:45          (00:05)
reboot   system boot  2.6.18-8.el5
 Sun Jan 24 21:41          (00:02)

wtmp begins Sun Jan 24 21:41:03 2010
```

4．ac 命令

ac 命令根据当前的/var/log/wtmp 文件中的登录进入和退出来报告用户连结的时间（小时），如果不使用标志，则报告总的时间。

```
# ac
total       18.47
```

另外，可加一些参数，例如，last -u 102 将报告 UID 为 102 的用户；last -t 7 表示限制上一周的报告。

5．lastlog 命令

lastlog 文件在每次有用户登录时被查询。可以使用 lastlog 命令检查某特定用户上次登录的时间，并格式化输出上次登录日志/var/log/lastlog 的内容。它根据 UID 排序显示登录名、端口号（tty）和上次登录时间。如果一个用户从未登录过，lastlog 显示 **Never logged**。注意需要以 root 身份运行该命令。▌※

本文为节选，完整内容和代码下载见原文：
http://os.51cto.com/art/201009/228771.htm

作者近期文章：

深入浅出 Netfilter/iptables（基础篇）

如何正确地部署防火墙？

内网安全部署应选择不同品牌的产品

从韩国农协银行事件谈信息安全工作的要点

活动目录在构建核心过程中的八个关键点

你的 Postfix 邮件服务器安全么？

日志文件分析工作中的第一个挑战是把异常活动从正常活动中识别出来。完成这项挑战的前提是你必须知道系统和网络上的正常活动在日志文件里是什么样子。

Linux 服务器日志文件查找技巧精粹

文/曹江华

用来在日志文件里搜索特定活动事件的工具不下几十种，本文将介绍搜索日志文件时应该采取的策略。

1、查找日志文件简单方法

一般来说，系统日志文件几乎都保存在/var/子目录（该路径由 syslog.conf 文件定义）。如果想让所有的应用程序都把日志文件集中存放到/var/子目录下，需要依次对每一个应用程序的配置文件进行编辑。把日志集中到/var/子目录下是个很好的主意。首先，当需要查看它们、修改它们的权限或者对它们进行备份的时候，只要到一个地方就可以找到所有的日志文件。

其次，/var/子目录通常是在一个独立于根目录（/）的文件系统里，这有助于防止日志文件迅速变大并占满可用空间，避免操作系统和应用程序受到影响。可以利用 find 命令把你不知道的日志文件查找出来具体做法是：先切换到根目录下，然后以根用户（root）身份执行下面这条命令把最近被修改过的文件全部找出来：

```
find . -type f -mtime -5 -print \
 | grep -v proc | grep -v lock
```

2、搜索日志文件时的策略

日志文件分析工作中的第一个挑战是把异常活动从正常活动中识别出来。完成这项挑战的前提是你必须知道系统和网络上的正常活动在日志文件里是什么样子。如果没有事先积累的经验，很难知道按规律发生的事件在日志文件里的表现。熟悉正常的日志文件活动要有一个时间过程，要求大家一上来就把日志文件看明白显然不现实，这是个需要时间积累的过程。

随着网络上的应用程序和用户的数量不断增减和变化，日志文件的内容肯定会发生相应的改变。把异常情况孤立出来之后，接下来的步骤是正确地判断这种异常情况是否属于警报条件。要想正确地做出判断，只能通过加深对公司日常活动的了解才能做到。往往需要在系统的可用性与防范风险之间把握一种平衡。

3、以手动方式搜索日志文件

grep 是 Unix 系统上功能最强大的 shell 命令之一。利用 grep 命令在日志文件里搜索各种可疑线索是这个文本文件搜索命令的绝佳用途。grep 命令的用法很简单——在命令行上输入：

```
grep "failed" /var/log/messages
```

上面这条 grep 命令将把 /var/log/messages 文件里包含单词

“failed”的文本行全部找出来。在默认情况下，grep 命令是大小写敏感的，你可能需要根据具体情况使用 grep 命令和它的“-i”选项来进行对大小写不敏感的搜索。搜索日志文件的挑战之一是你必须先知道自己想找什么东西，然后才可以把可能存在的这个东西找出来。有几种办法可以帮助解决这一问题。

如果你知道自己想找的事件或活动——比如，用户试图使用 su 命令切换为根用户——可以先自己进行一次这样的活动，然后去日志文件里看看它是什么样子。比如，在 SUSE Linux 系统上，执行失败的 su 命令将在日志文件里留下这样一条记录：

```
Apr  1 11:15:54 chim su: FAILED SU (to
 root) rreck on /dev/pts/1
```

于是，如果想把所有这类活动都查出来，应该使用下面这样的命令：

```
grep "FAILED SU" /var/log/messages
```

上面示例里的活动是黑客攻击的一种标志。如果 grep 命令只在日志文件里零零散散地找到了几个这样的失败事件，那很可能是有人忘记了口令字或者是打字时出现了错误。反之，如果 grep 命令在日志文件里找到几十个这样的失败事件，很可能是有人在试图闯入你的系统，应立刻采取措施在网络级拒绝他们的访问。

4、使用 logsurfer 工具搜索日志文件

logsurfer 是一个日志文件搜索工具。与 swatch 等日志搜索工具相比，logsurfer 允许人们做出更细致的决定。与其他日志搜索程序相似，logsurfer 工具也是把日志文件里的每一行与一些规则表达式进行匹配，每找到一个匹配就相应地执行某个动作，而那些动作被表达为“规则”。logsurfer 工具在某些方面比 swatch 工具做得还要好。

首先，logsurfer 工具使用两组规则表达式匹配文本行；日志文件中的文本行必须匹配第一组表达式，但不需要匹配可选的第二组表达式。这种安排在某些场合非常有用，它可以让人们方便地排除异常情况。logsurfer 工具的另外一个突出优点是它可以结合上下文进行检查而不是只检查单个的文本行。这很方便，因为日志文件里的单独一行文本往往不足以包含做出某个决定所需要的信息。

不少人认为 logsurfer 工具比较难以配置，因为对它进行配置必须精通规则表达式，而且它本身提供的配置示例不够多。

5、使用 swatch 工具搜索日志文件

swatch 工具不是 Red Hat Enterprise Linux 发行版本的标准组成部分，但因为 swatch 工具很容易安装、配置和使用，所以你应该考虑下载并安装它。在开始使用它之前，还需要一个配置文件。在默认情况下，swatch 工具将在启动它的那个用户的登录子目录里寻找 .swatchrc 文件（例如/home/rreck/.swatchrc），但你可以用“-f”选项为它另外指定一个配置文件，如下所示：

```
swatch -f /etc/.swatchrc
```

接下来，看看如何使用 swatch 工具从日志文件里发现黑客活动的踪迹以及发现之后该做些什么。▌※

本文为节选，原文地址：
http://os.51cto.com/art/200808/84036.htm

它最重要的特点是容错性好。当后端的存储系统 crash 时，scribe 会将数据写到本地磁盘上，当存储系统恢复正常后，scribe 将日志重新加载到存储系统中。

开源日志系统比较

文/董西成

1. 背景介绍

许多公司的平台每天会产生大量的日志（一般为流式数据，如，搜索引擎的 pv，查询等），处理这些日志需要特定的日志系统，一般而言，这些系统需要具有以下特征：

（1） 构建应用系统和分析系统的桥梁，并将它们之间的关联解耦；

（2） 支持近实时的在线分析系统和类似于 Hadoop 之类的离线分析系统；

（3） 具有高可扩展性。即：当数据量增加时，可以通过增加节点进行水平扩展。

本文从设计架构，负载均衡，可扩展性和容错性等方面对比了当今开源的日志系统，包括 facebook 的 scribe，apache 的 chukwa，linkedin 的 kafka 和 cloudera 的 flume 等。

2. FaceBook 的 Scribe

Scribe 是 facebook 开源的日志收集系统，在 facebook 内部已经得到大量的应用。它能够从各种日志源上收集日志，存储到一个中央存储系统（可以是 NFS，分布式文件系统等）上，以便于进行集中统计分析处理。它为日志的“分布式收集，统一处理”提供了一个可扩展的，高容错的方案。

它最重要的特点是容错性好。当后端的存储系统 crash 时，scribe 会将数据写到本地磁盘上，当存储系统恢复正常后，scribe 将日志重新加载到存储系统中。

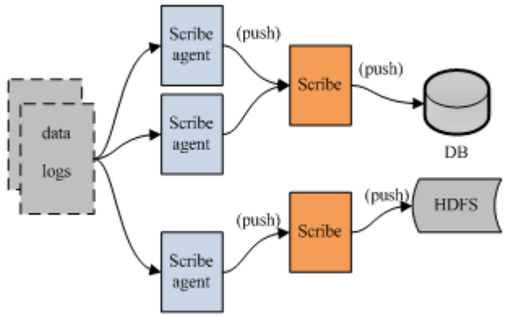

图 1 Scribe 架构

架构：

scribe 的架构比较简单，主要包括三部分，分别为 scribe agent， scribe 和存储系统。

(1) scribe agent

scribe agent 实际上是一个 thrift client。 向 scribe 发送数据的唯一方法是使用 thrift client， scribe 内部定义了一个 thrift 接口，用户使用该接口将数据发送给 server。

(2) scribe

scribe 接收到 thrift client 发送过来的数

据，根据配置文件，将不同 topic 的数据发送给不同的对象。scribe 提供了各种各样的 store，如 file， HDFS 等，scribe 可将数据加载到这些 store 中。

(3) 存储系统

存储系统实际上就是 scribe 中的 store，当前 scribe 支持非常多的 store，包括 file（文件），buffer（双层存储，一个主储存，一个副存储），network（另一个 scribe 服务器），bucket（包含多个 store，通过 hash 的将数据存到不同 store 中），null(忽略数据)，thriftfile（写到一个 Thrift TFileTransport 文件中）和 multi（把数据同时存放到不同 store 中）。

3. Apache 的 Chukwa

chukwa 是一个非常新的开源项目，由于其属于 hadoop 系列产品，因而使用了很多 hadoop 的组件（用 HDFS 存储，用 mapreduce 处理数据），它提供了很多模块以支持 hadoop 集群日志分析。

需求：

(1) 灵活的，动态可控的数据源

(2) 高性能，高可扩展的存储系统

(3) 合适的框架，用于对收集到的大规模数据进行分析

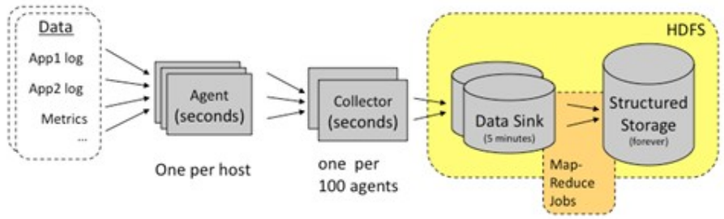

图 2 Chukwa 架构

架构：

Chukwa 中主要有 3 种角色，分别为：adaptor，agent，collector。

(1) Adaptor 数据源

可封装其他数据源，如 file，unix 命令行工具等

目前可用的数据源有：hadoop logs，应用程序度量数据，系统参数数据（如 linux cpu 使用流率）。

(2) HDFS 存储系统

Chukwa 采用了 HDFS 作为存储系统。HDFS 的设计初衷是支持大文件存储和小并发高速写的应用场景，而日志系统的特点恰好相反，它需支持高并发低速率的写和大量小文件的存储。需要注意的是，直接写到 HDFS 上的小文件是不可见的，直到关闭文件，另外，HDFS 不支持文件重新打开。

(3) Collector 和 Agent

为了克服(2)中的问题，增加了 agent 和 collector 阶段。

Agent 的作用：给 adaptor 提供各种服务，包括：启动和关闭 adaptor，将数据通过 HTTP 传递给 Collector；定期记录 adaptor 状态，以便 crash 后恢复。

Collector 的作用：对多个数据源发过来的数据进行合并，然后加载到 HDFS 中；隐藏 HDFS 实现的细节，如，HDFS 版本更换后，只需修改 collector 即可。

(4) Demux 和 achieving

直接支持利用 MapReduce 处理数据。它内置了两个 mapreduce 作业，分别用于获取 data 和将 data 转化为结构化的 log。存储到

data store（可以是数据库或者 HDFS 等）中。

4. LinkedIn 的 Kafka

Kafka 是 2010 年 12 月份开源的项目，采用 scala 语言编写，使用了多种效率优化机制，整体架构比较新颖（push/pull），更适合异构集群。

设计目标：

(1) 数据在磁盘上的存取代价为 O(1)

(2) 高吞吐率，在普通的服务器上每秒也能处理几十万条消息

(3) 分布式架构，能够对消息分区

(4) 支持将数据并行的加载到 hadoop

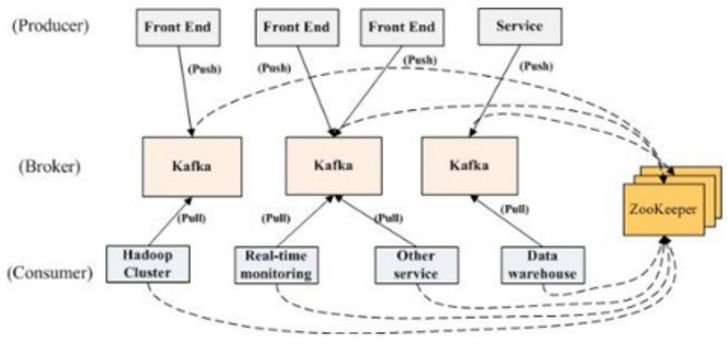

图 3 Kafka 架构

架构：

Kafka 实际上是一个消息发布订阅系统。producer 向某个 topic 发布消息，而 consumer 订阅某个 topic 的消息，进而一旦有新的关于某个 topic 的消息，broker 会传递给订阅它的所有 consumer。 在 kafka 中，消息是按 topic 组织的，而每个 topic 又会分为多个 partition，这样便于管理数据和进行负载均衡。同时，它也使用了 zookeeper 进行负载均衡。

Kafka 中主要有三种角色，分别为 producer，broker 和 consumer。

(1) Producer

Producer 的任务是向 broker 发送数据。Kafka 提供了两种 producer 接口，一种是 low_level 接口，使用该接口会向特定的 broker 的某个 topic 下的某个 partition 发送数据；另一种那个是 high level 接口，该接口支持同步/异步发送数据，基于 zookeeper 的 broker 自动识别和负载均衡（基于 Partitioner）。

其中，基于 zookeeper 的 broker 自动识别值得一说。producer 可以通过 zookeeper 获取可用的 broker 列表，也可以在 zookeeper 中注册 listener，该 listener 在以下情况下会被唤醒：

a．添加一个 broker

b．删除一个 broker

c．注册新的 topic

d．broker 注册已存在的 topic

当 producer 得知以上时间时，可根据需要采取一定的行动。

(2) Broker

Broker 采取了多种策略提高数据处理效率，包括 sendfile 和 zero copy 等技术。

(3) Consumer

consumer 的作用是将日志信息加载到中央存储系统上。kafka 提供了两种 consumer 接口，一种是 low level 的，它维护到某一个 broker 的连接，并且这个连接是无状态的，即，每次从 broker 上 pull 数据时，都要告诉 broker 数据的偏移量。另一种是 high-level 接口，它隐藏了 broker 的细节，允许 consumer 从 broker 上 push 数据而不必关

心网络拓扑结构。更重要的是，对于大部分日志系统而言，consumer 已经获取的数据信息都由 broker 保存，而在 kafka 中，由 consumer 自己维护所取数据信息。

5. Cloudera 的 Flume

Flume 是 cloudera 于 2009 年 7 月开源的日志系统。它内置的各种组件非常齐全，用户几乎不必进行任何额外开发即可使用。

设计目标：

(1) 可靠性

当节点出现故障时，日志能够被传送到其他节点上而不会丢失。Flume 提供了三种级别的可靠性保障，从强到弱依次分别为：end-to-end，Store on failure（这也是 scribe 采用的策略，当数据接收方 crash 时，将数据写到本地，待恢复后，继续发送），Best effort（数据发送到接收方后，不会进行确认）。

(2) 可扩展性

Flume 采用了三层架构，分别问 agent，collector 和 storage，每一层均可以水平扩展。其中，所有 agent 和 collector 由 master 统一管理，这使得系统容易监控和维护，且 master 允许有多个（使用 ZooKeeper 进行管理和负载均衡），这就避免了单点故障问题。

(3) 可管理性

所有 agent 和 colletor 由 master 统一管理，这使得系统便于维护。用户可以在 master 上查看各个数据源或者数据流执行情况，且可以对各个数据源配置和动态加载。Flume 提供了 web 和 shell script command 两种形式对数据流进行管理。

(4) 功能可扩展性

用户可以根据需要添加自己的 agent，colletor 或者 storage。此外，Flume 自带了很多组件，包括各种 agent，collector 和 storage。

架构：

正如前面提到的，Flume 采用了分层架构，由三层组成，分别为 agent，collector 和 storage。其中，agent 和 collector 均由两部分组成：source 和 sink，source 是数据来源，sink 是数据去向。

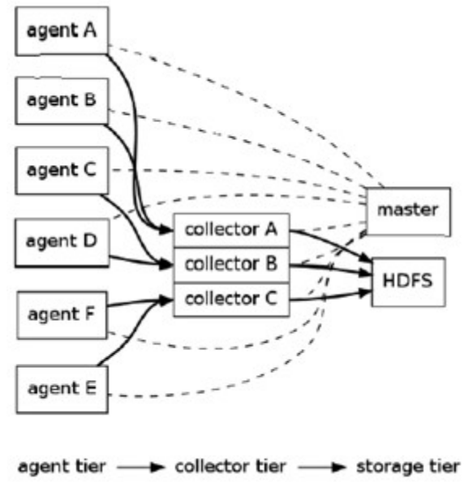

图 4 Flume 架构

(1) agent

agent 的作用是将数据源的数据发送给 collector，Flume 自带了很多直接可用的数据源（source），如：

text(“filename”)：将文件 filename 作为数据源，按行发送

tail(“filename”)：探测 filename 新产生的数据，按行发送出去

fsyslogTcp(5140)：监听 TCP 的 5140 端口，并且接收到的数据发送出去

同时提供了很多 sink，如：

console[("format")] ：直接将将数据显示在桌面上

text(“txtfile”)：将数据写到文件 txtfile 中

dfs(“dfsfile”)：将数据写到 HDFS 上的 dfsfile 文件中

syslogTcp(“host”,port)：将数据通过 TCP 传递给 host 节点

(2) collector

collector 的作用是将多个 agent 的数据汇总后，加载到 storage 中。它的 source 和 sink 与 agent 类似。

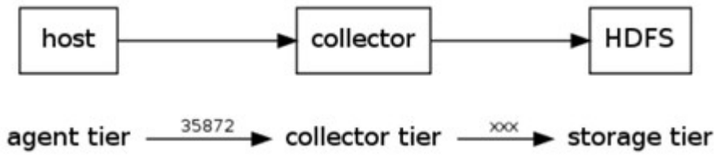

图 5 collector

下面例子中，agent 监听 TCP 的 5140 端口接收到的数据，并发送给 collector，由 collector 将数据加载到 HDFS 上。

```
host : syslogTcp(5140) |
 agentSink("localhost",35853) ;
```

```
collector : collectorSource(35853) |
 collectorSink("hdfs://namenode/user/flume/ ","syslog");
```

此外，使用 autoE2EChain，当某个 collector 出现故障时，Flume 会自动探测一个可用 collector，并将数据定向到这个新的可用 collector 上。

(3) storage

storage 是存储系统，可以是一个普通 file，也可以是 HDFS，HIVE，HBase 等。

6. 总结

根据这四个系统的架构设计，可以总结出典型的日志系统需具备三个基本组件，分别为 agent（封装数据源，将数据源中的数据发送给 collector），collector（接收多个 agent 的数据，并进行汇总后导入后端的 store 中），store（中央存储系统，应该具有可扩展性和可靠性，应该支持当前非常流行的 HDFS）。

下面表格对比了这四个系统：

	scribe	Chukwa	Kafka	Flume
公司	facebook	apache/yahoo	LinkedIn	Cloudera
开源时间	2008 年 10 月	2009 年 11 月	2010 年 12 月	2009 年 7 月
实现语言	C/C++	JAVA	SCALA	JAVA
框架	push/push	push/push	push/pull	push/push
容错性	collector 和 store 之间有容错机制，而 agent 和 collector 之间的容错需用户自己实现	Agent 定期记录已送给 collector 的数据偏移量，一旦出现故障后，可根据偏移量继续发送数据。	Agent 可用通过 collector 自动识别机制获取可用 collector。store 自己保存已经获取数据的偏移量，一旦 collector 出现故障，可根据偏移量继续获取数据。	Agent 和 collector，collector 和 store 之间均有容错机制，且提供了三种级别的可靠性保证。
负载均衡	无	无	使用 zookeeper	使用 zookeeper
可扩展性	好	好	好	好
agent	Thrift client，需自己实现	自带一些 agent，如获取 hadoop logs 的 agent	用户需根据 Kafka 提供的 low-level 和 high-level API 自己实现。	提供了各种非常丰富的 agent
collector	实际上是一个 thrift server	--	使用了 sendfile，zero-copy 等技术提高性能	系统提供了很多 collector，直接可以使用。
store	直接支持 HDFS	直接支持 HDFS	直接支持 HDFS	直接支持 HDFS
总体评价	设计简单，易于使用，但容错和负载均衡方面不够好，且资料较少。	属于 hadoop 系列产品，直接支持 Hadoop，目前版本升级比较快，但还有待完善。	设计架构（push/pull）非常巧妙，适合异构集群，但产品较新，其稳定性有待验证。	非常优秀

▌※

本文为节选，原文见：
http://dongxicheng.org/search-engine/log-systems/

chattr 命令可以修改文件属性，达到保护文件和目录的作用。相比改变文件读写、执行权限的 chmod 命令，chattr 命令可以控制更底层的文件属性。

Linux 服务器安全初始化 Shell 脚本

脚本提供/晓辉
内容整理/抚琴煮酒

本脚本用于 Linux 系统的安全初始化，可以在服务器系统安装完毕之后立即执行以快速建立起服务器的安全防护。最初的脚本由晓辉撰写，在数次修改之后已经大量应用在某大型媒体网站体系中。该脚本由抚琴煮酒推荐至 51CTO 系统频道，通过和原作者交流之后在原有的脚本上又做了些小改动，修改了一些 bug，已经在 CentOS 5.5 x86_64 下通过，目前在一些没有硬件防火墙的服务器上使用。

使用方法：将其复制，保存为一个 shell 文件，比如 security.sh。将其上传到 linux 服务器上，执行 sh security.sh，就可以使用该脚本了。建议大家在系统初始化后立即执行，然后创建了用户帐号和密码后就不要再改动了，以免影响重要文件的初始 md5 值。

脚本内容：

```
#!/bin/sh
# desc: setup linux system security
# author:coralzd
# powered by www.freebsdsystem.org
# version 0.1.2 written by 2011.05.03

#设置账号
passwd -l xfs
passwd -l news
passwd -l nscd
passwd -l dbus
passwd -l vcsa
passwd -l games
passwd -l nobody
passwd -l avahi
passwd -l haldaemon
passwd -l gopher
passwd -l ftp
passwd -l mailnull
passwd -l pcap
passwd -l mail
passwd -l shutdown
passwd -l halt
passwd -l uucp
passwd -l operator
passwd -l sync
passwd -l adm
passwd -l lp

# 用 chattr 给用户路径更改属性。
# chattr 命令用法参考文末说明[1]
chattr +i /etc/passwd
chattr +i /etc/shadow
chattr +i /etc/group
chattr +i /etc/gshadow

# 设置密码连续输错 3 次后锁定 5 分钟
sed -i 's#auth required\
 pam_env.so#auth required\
 pam_env.so nauth required\
 pam_tally.so onerr=fail deny=3\
 unlock_time=300\
 nauth required /lib/security/ \
 $ISA/pam_tally.so onerr=fail \
 deny=3 unlock_time=300#'\
```

```
 /etc/pam.d/system-auth

# 5 分钟后自动登出，原因参考文末说明[2]
echo "TMOUT=300" >> /etc/profile

# 历史命令记录数设定为 10 条
sed -i
 "s/HISTSIZE=1000/HISTSIZE=10/"
 /etc/profile

# 让以上针对 /etc/profile 的改动立即生效
source /etc/profile

# 在 /etc/sysctl.conf 中启用 syncookie
echo "net.ipv4.tcp_syncookies=1" >>
 /etc/sysctl.conf
# exec sysctl.conf enable
sysctl -p

# 优化 sshd_config
sed -i "s/#MaxAuthTries 6/MaxAuthTries
 6/" /etc/ssh/sshd_config
sed -i  "s/#UseDNS yes/UseDNS no/"
 /etc/ssh/sshd_config

# 限制重要命令的权限
chmod 700 /bin/ping
chmod 700 /usr/bin/finger
chmod 700 /usr/bin/who
chmod 700 /usr/bin/w
chmod 700 /usr/bin/locate
chmod 700 /usr/bin/whereis
chmod 700 /sbin/ifconfig
chmod 700 /usr/bin/pico
chmod 700 /bin/vi
chmod 700 /usr/bin/which
chmod 700 /usr/bin/gcc
chmod 700 /usr/bin/make
chmod 700 /bin/rpm

# 历史安全
chattr +a /root/.bash_history
chattr +i /root/.bash_history

# 给重要命令写 md5
cat > list << "EOF" &&
/bin/ping
/usr/bin/finger
/usr/bin/who
/usr/bin/w
/usr/bin/locate
/usr/bin/whereis
/sbin/ifconfig
/bin/vi
/usr/bin/vim
/usr/bin/which
/usr/bin/gcc
/usr/bin/make
/bin/rpm
EOF

for i in `cat list`
do
   if [ ! -x $i ];then
   echo "$i not found,no md5sum!"
  else
   md5sum $i >>
 /var/log/`hostname`.log
  fi
done
rm -f list
```

知识点[1]：有关 chattr 命令

chattr 命令可以修改文件属性，达到保护文件和目录的作用。相比改变文件读写、执行权限的 chmod 命令，chattr 命令可以控制更底层的文件属性。该命令十分强大，其中一些功能是由 Linux 内核版本来支持的，如果 Linux 内核版本低于 2.2，那么许多功能不能实现。同样-D 检查压缩文件中的错误的功能，需要 2.5.19 以上内核才能支持。另外，通过 chattr 命令修改属性能够提高系统的安全性，但是它并不适合所有的目录。chattr 命令不能保护/、/dev、/tmp、/var 目录。

此类属性的查看可以通过 lsattr 命令完

成。

chattr 命令的用法：chattr [-RV] [-v version] [mode] files...

最关键的是在[mode]部分，，即文件属性部分。[mode]部分是由+-=和[ASacDdIijsTtu]这些字符组合的。

+ ：在原有参数设定基础上，追加参数。

- ：在原有参数设定基础上，移除参数。

= ：更新为指定参数设定。

A：文件或目录的 atime (access time)不可被修改(modified), 可以有效预防例如手提电脑磁盘 I/O 错误的发生。

S：硬盘 I/O 同步选项，功能类似 sync。

a：即 append，设定该参数后，只能向文件中添加数据，而不能删除，多用于服务器日志文 件安全，只有 root 才能设定这个属性。

c：即 compresse，设定文件是否经压缩后再存储。读取时需要经过自动解压操作。

d：即 no dump，设定文件不能成为 dump 程序的备份目标。

i：设定文件不能被删除、改名、设定链接关系，同时不能写入或新增内容。i 参数对于文件 系统的安全设置有很大帮助。

j：即 journal，设定此参数使得当通过 mount 参数：data=ordered 或者 data=writeback 挂 载的文件系统，文件在写入时会先被记录(在 journal 中)。如果 filesystem 被设定参数为 data=journal，则该参数自动失效。

s：保密性地删除文件或目录，即硬盘空间被全部收回。

u：与 s 相反，当设定为 u 时，数据内容其实还存在磁盘中，可以用于 undeletion.

各参数选项中常用到的是 a 和 i。a 选项强制只可添加不可删除，多用于日志系统的安全设定。而 i 是更为严格的安全设定，只有 superuser (root) 或具有 CAP_LINUX_IMMUTABLE 处理能力（标识）的进程能够施加该选项。

应用实例：

1、用 chattr 命令防止系统中某个关键文件被修改

```
# chattr +i /etc/fstab
```

然后试一下 rm mv rename 等命令操作于该文件，都是得到 Operation not permitted 的结果

2、让某个文件只能往里面追加内容，不能删除，一些日志文件适用于这种操作

```
# chattr +a /data1/user_act.log
```

知识点[2]:为何要设置 5 分钟后自动登出

由于客户的维护人员常常登陆上去后通过直接关闭 TERM 端口非法退出 telnet，造成系统的 pts 进程越来越多，一个月下来竟然近百，当进程过多的时候系统就会产生报警。规范操作应该是用 exit 或者 ctrl+D，但是其他人并不这样操作，所以我们定义了 echo "TMOUT=300" >>/etc/profile 这一项内容，是让服务器自动剔除 300 秒没有任何动作的客户端。当然了这一项大家可以根据实际需求而决定是否添加。▌※

本文为节选，原文地址：
http://os.51cto.com/art/201107/273839.htm

> 注意尽量不用-9，数据库服务器上更不能轻易用 kill，否则造成重要数据丢失后果将不堪设想。

实用推荐：MogileFS 排错小技巧

文/扶凯

MogileFS 内部提供了很强大的功能，对排错和调节也设计的非常好，只是普通的时候用不大。现在我就来教大家一下，这些常用的 Mogilefsd 的命令如下：

◇!version 服务器的版本

◇!recent 最新的查询和花费的时间

◇!queue 队列中正在执行的查询

◇!stats 全局的状态和统计

◇!watch 显示子程序的出错和相关的信息

◇!jobs 没完成的任务的 counts, desired level 和 pids.

◇!shutdown 直播 kill 掉 mogilefsd.

比如有一次服务器出现无法显示硬盘空间的问题，我就使用 Mogilefsd 的!watch 命令来排错：

```
$ telnet 192.168.1.11 7001
Trying 192.168.1.11...
Connected to 192.168.1.11
 (192.168.1.11).
Escape character is '^]'.
!stats
uptime 6088
pending_queries 0
processing_queries 0
bored_queryworkers 5
queries 36
work_queue_for_replicate 0
work_sent_to_replicate 9
.
!watch
```

```
Added you to watcher list.
.
:: [monitor(23829)] Port 7500 not
 listening on 192.168.1.12
 (http://192.168.1.12:7500/dev4/usage)
 ? Error was: 500 Cant connect to
 192.168.1.12:7500 (No route to host)
:: [monitor(23829)] Port 7500 not
 listening on 192.168.1.12
 (http://192.168.1.12:7500/dev2/usage)
 ? Error was: 500 Cant connect to
 192.168.1.12:7500 (No route to host)
:: [monitor(23829)] Port 7500 not
 listening on 192.168.1.12
 (http://192.168.1.12:7500/dev4/usage)
 ? Error was: 500 Cant connect to
 192.168.1.12:7500 (No route to host)
:: [monitor(23829)] Port 7500 not
 listening on 192.168.1.12
 (http://192.168.1.12:7500/dev2/usage)
 ? Error was: 500 Cant connect to
 192.168.1.12:7500 (No route to host)
```

这样一看，原来是 iptables 默认打开所引起的问题。MogileFS 本身非常稳定，有问题的话建议还是多从自身来查原因。▌※

本文为节选，原文：
http://www.php-oa.com/2011/06/30/mogilefs-troubleshooting-request-failure-fetching.html

> 如果你重复执行同样两条或三条命令的不同排列组合，你还可以选择通过识别其位移负指数来执行出现在上一命令出现处的命令。

强有力的 Linux 历史命令 你还记得几个

文/ Jason Gilmore
编译/Mark

“忘记历史的 Linux 用户注定要输入很多信息。”

这也让强有力的历史命令不仅在援引之前执行命令而不需重新输入它们时有用，在调用其它很少用到的命令时也有用，这省去了必须重新使用它们的麻烦。

该命令的输出示例如下：

```
$ history
...
62 rm 092210.sql
63 mysqldump
64 mysqldump -u root -p
 dev_gamenomad_com > 092210.sql
65 more 092210.sql
66 rm 092210.sql
...
9991 mkdir chapter05
9992 cd chapter05
9993 dir
9994 npm install websocket-server
9995 node hello.js
9996 exit
9997 history
```

与每条命令相关的顺序号服务于重要目的，允许用户通过提供直接跟着感叹号的顺序号来重新执行相关命令，如下：

```
$ !10000
sudo /etc/init.d/apache2 start
* Starting web server apache2
```

掌控历史扩展

敲击向上箭头键会显示之前执行的命令，敲击 Enter 键会再次执行该命令。但是还可以使用另一个可能更快的包括历史扩展功能的选择：

```
$ !!
```

如果你重复执行同样两条或三条命令的不同排列组合，你还可以选择通过识别其位移负指数来执行出现在上一命令出现处的命令。例如，执行之前命令的前一命令(回退两条命令)，运用以下序列：

```
$ !-2
```

执行早期命令的另一方式是输入紧跟着感叹号的命令序列号的开头。符合字符对象的第一条命令会执行。举例来说，假设最后三条命令如下：

```
$ history
...
9876 build-book
 /home/wjgilmore/easy_php
9877 mkdir chapter05
9878 cd chapter05
9879 touch chapter05.md
```

你可以只通过运行以下命令来再次执行构书脚本：

```
$ !b
```

用两步过程创建一个确认它的新目录和导航和以运用历史扩展缩短。在本例中，我创建了一个叫做 easy_bash 的新目录，位置

在/home/wjgilmore/books。要确认该目录，运用!$来获得在前一命令中发现在最后“字符”：

```
$ mkdir
 /home/wjgilmore/books/easy_bash
$ cd !$
```

搜索命令历史

虽然可以翻阅命令历史，但在使用 Ctrl+R 开放选择搜索它之后也可以调用该命令的片段，提示的命令行将如下所示：

```
(reverse-i-search)`':
```

开始输入命令片段，命令行会实时更新反映出最符合的一条。输入 apa 后会出现类似于这样的命令：

```
(reverse-i-search)`apa': sudo
 /etc/init.d/apache2 start
```

当你看到想要的命令，敲击 Enter 执行它，或者敲击向上箭头键在重新执行前修改它。

调整历史行为

你可以做很多有趣的设置更改来控制命令历史的行为方式。例如，你可以通过设置在 Bash 配置文件(.bashrc，位于主目录中)上发现的 HISTSIZE 变量提高限制数。

```
HISTSIZE=10000
```

你在命名为 HISTCONTROL 的.bashrc 中通常还会遇到另一个与历史有关的变量。该变量帮助明确规定历史文件中的存放内容(位于.bash_history，也可见于主目录中)。例如，你也许在检查最新的日志文件附件时重复执行了尾命令，就在运行 tail 的几分钟里。通过重复来集群历史文件不太实际，通过设置它为 ignoredups 来让 HISTCONTROL 变量忽视复制行。

```
HISTCONTROL=ignoredups
```

可疑效用的另一个 HISTCONTROL 设置是忽略空格。该 HISTCONTROL 设置的结果是所有前面有空格的执行命令都从历史文件中删除。这对于不跟踪每条命令似乎起了反作用，但是如果你除了 ignoredups 之外还想实现该功能，你可以将两者设置成：

```
HISTCONTROL=ignoreboth
```

我最后要谈及的一个功能是命令替换。假设你想基于一个虚拟主机配置文件模板快速配置一群虚拟主机。这些命令序列可能非常长，就像下面所示：

```
$ cp vhost.template
 /etc/apache2/sites-
 available/dev.example.com
$ cp vhost.template
 /etc/apache2/sites-
 available/forum.example.com
$ cp vhost.template
 /etc/apache2/sites-
 available/staging.example.com
```

通常你执行该一列命令的方法会是首先输入和执行，接着向上滚动、删除尾行片段(dev.example.com)，然后输入下一片段(forum.example.com)，然后重新执行。另一种方法，你可以运用命令替换来快速地用一行字符串替换另一相关字符串，如下所示：

```
$ !!:s/dev/forum
```

该命令行在指向-点击界面中已为用户带来了大量好处，让任务执行和操作系统导航可以快速且优雅地完成。▌※

本文为节选，原文地址：
http://www.searchsv.com.cn/showcontent_49775.htm

英文原文：
Mastering the Linux history command

招募启事

《Linux 运维趋势》的建设需要您的加入！

您可以通过如下方式参与我们杂志的建设：

1、推荐文章

无论是您在互联网上看到的好文章，还是您自己总结/整理的资料；无论是英文还是中文；无论是入门的还是高端的，都欢迎推荐！推荐方式包括：

a）在技术圈中分享：http://g.51cto.com/linuxops

b）发邮件给编辑：yangsai@51cto.com

2、投稿

如果您认为自己在 Linux 方面具有专家级别的能力，并且有与大家分享您技术经验的热诚，同时也有兴趣挣点稿费花花，那么欢迎您的投稿！

如果您在 IT 技术方面的翻译有很高的能力，能够快速、高质量的完成译文，并且也经常浏览到一些 Linux 方面的优秀外文，那么也欢迎您的投稿，或加入我们 51CTO 的翻译团队！

投稿信箱：yangsai@51cto.com

3、推广与意见

如果您喜欢我们的杂志，认为这本杂志对于您的工作有所帮助，请向您的 Linux 好友、同事们推荐它！

如果您觉得这份杂志还有什么地方需要改进或补充，也希望您能够提出您的宝贵意见！

联系人：yangsai@51cto.com

下期预告

Linux 内核 20 周年特刊，敬请期待！

本刊为月刊，预定每月发布日期为：

每个月的第二个星期五

您可以通过如下方式检查新刊发布：

1、通过电子邮件订阅：

订阅方式请参考本杂志的专题页：
http://os.51cto.com/art/201011/233915.htm

2、经常光顾 51CTO Linux 频道：
http://os.51cto.com/linux/