Broadview
www.broadview.com.cn

网站运维技术与实践

——饶琛琳 编著——

Web Operations Technology and Practice

大型网站一线运维技巧与经验总结
全面解析运维相关技术

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

网站运维技术与实践

——饶琛琳 编著——

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

网站运维工作，一向以内容繁杂、覆盖面广著称。本书选取日常工作涉及的监测调优、日志分析、集群规划、自动化部署、存储和数据库等方面，力图深入阐述各项工作的技术要点及协议原理，并介绍相关开源产品的实践经验。在技术之外，作者也分享了一些关于高效工作及个人成长方面的心得。

本书适合 Linux 系统管理员、中大型网站运维工程师及技术负责人、DevOps 爱好者阅读。同时也适于刚踏上或有兴趣踏上运维岗位的年轻朋友，了解运维职业的工作和发展。

图书在版编目（CIP）数据

网站运维技术与实践 / 饶琛琳编著. —北京：电子工业出版社，2014.3
ISBN 978-7-121-22433-1

Ⅰ. ①网… Ⅱ. ①饶… Ⅲ. ①网站－开发 Ⅳ. ①TP393.092

中国版本图书馆 CIP 数据核字（2014）第 019500 号

策划编辑：董 英
责任编辑：徐津平
印 刷：北京丰源印刷厂
装 订：三河市鹏成印业有限公司
出版发行：电子工业出版社
北京市海淀区万寿路 173 信箱 邮编 100036
开 本：787×980 1/16 印张：28.5 字数：535 千字
印 次：2014 年 3 月第 1 次印刷
定 价：69.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前　言

运维是一个古老但愈发新奇的职位。在不同时代、不同公司，都有不同的称呼。在万维网到来之前漫长的几十年中，运维工作大都由系统本身的开发人员来完成，他们很自豪地给自己加上了系统管理者（System Administrator）的头衔。随着万维网的出现和发展，计算机系统管理者中也就出现了专注于网站管理的人群，这些人自称为网站管理者，至今我们依然可以在一些历史悠久的软件（比如 Apache、Squid）的配置中，看到专门的指令来设定这个身份。与此同时，在互联网的另一端（接入端）——为普及上网而大量出现的网吧和网城中，另一批专注于网络接入、局域网共享和桌面应用管理的人群，则被称为“网吧管理员”。接入端牢牢占据了绝大多数人对互联网的第一印象，并将他们所能触及的方面认定为互联网从业人员的全部，即软件开发者和网络管理者。

毫不讳言，笔者在五年前（大学毕业时），同样以这种眼光看待自己“很熟悉”的这个互联网。

那么，除去接入端的网管，在互联网的另一端的管理者们到底是什么状态呢？

先说看得见的一面：每次当你发现网页变样了，这说明网站管理者完成了一次应用发布；每次你投诉访问有问题并附上截图，这意味着网站管理者要开始一次故障排查和修复；每次你觉得比上次访问快一点了，这说明网站管理者已经悄悄结束了一次后台优化……

再说看不见的一面：管理者尽力为你提供优质的访问体验，也带来指数级增长的新访问者。一百万、一千万乃至更多，大家的访问体验都要一样好，数据都要一样可靠，甚至业大招贼后还要保护大家的信息不被窃取……这些问题的背后，都是网站管理者的工作。

正是由于网站管理者与访问者之间的频繁交流，以及网站访问数据对业务发展的支撑，慢慢地将管理者的职责从系统维护扩展到了运营相关的广泛领域，最终合二为一成为了“运维”，而这也是现代的专职的“网站运维”与“古代的”模糊化的“系统管理员”最重要的区别。在业内公推为经典著作的 *Web Operations* 一书中，甚至专门有第 8 章“Community Management and Web Operations”来讲述运维和用户交流、社区管理相关的内容。

从上一代互联网巨头引申出网站运维这个独立的职位到现在，运维的职能依然在不断细化和变化——网络运维、系统运维、应用运维、数据库运维，甚至更细分的 CDN 运维和业务变更运维，都有专门的人员和团队来负责。运维团队甚至不再仅仅是网站服务的支持方，还越来越以网站内部的技术需求方的角色出现，进行广泛而细心的考察，采取更激进的方法，从而提升网站的单位成本效益。

从大概一两年前开始，另一个新的概念“敏捷运维”（DevOps）跟随云计算的浪潮出现。从思想和理论上，目前对其依然没有准确的定义，但从技术实质上，无非是在保证产品质量和访问性能的前提下提高产品发布的频率，“自动化一切可自动化的工作”加上“充分了解业务流程”——而这本来就是一个优秀的运维人员应该去实现的事情！

了解业务流程是一件取决于个人偏好和公司文化的事情，虽然笔者在过去的工作经历中见过不少比相应的开发负责人还了解业务的老运维人员，但是这方面确实很难说出太多可以循序渐进的道理，还是让我们先掌握那些可以帮助我们“Laziness, Impatience and Hubris”（程序员的三大美德——懒惰、急躁和傲慢——出自 Larry Wall 的 *Programming Perl*）地完成网站运维工作的技术吧！

饶琛琳

十载耕耘奠定专业地位

博文视点诚邀精锐作者加盟

以书为证彰显卓越品质

《C++Primer（中文版）（第5版）》、《淘宝技术这十年》、《代码大全》、《Windows内核情景分析》、《加密与解密》、《编程之美》、《VC++深入详解》、《SEO实战密码》、《PPT演义》……

“圣经”级图书光耀夺目，被无数读者朋友奉为案头手册传世经典。

潘爱民、毛德操、张亚勤、张宏江、咎辉Zac、李刚、曹江华……

“明星”级作者济济一堂，他们的名字熠熠生辉，与IT业的蓬勃发展紧密相连。

十年的开拓、探索和励精图治，成就**博**古通今、**文**圆质方、**视**角独特、**点**石成金之计算机图书的风向标杆：博文视点。

“风翱翔于千仞兮，非梧不栖”，博文视点欢迎更多才华横溢、锐意创新的作者朋友加盟，与大师并列于IT专业出版之巅。

英雄帖

江湖风云起，代有才人出。
IT界群雄并起，逐鹿中原。
博文视点诚邀天下技术英豪加入，
指点江山，激扬文字
传播信息技术，分享IT心得

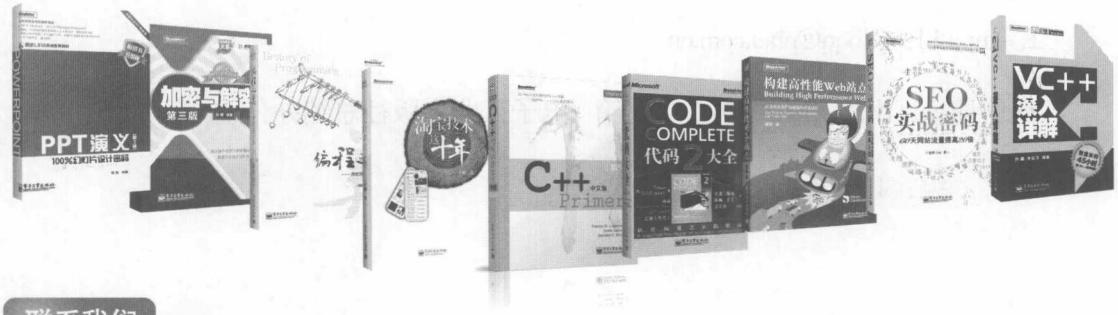
• 专业的作者服务 •

博文视点自成立以来一直专注于IT专业技术图书的出版，拥有丰富的与技术图书作者合作的经验，并参照IT技术图书的特点，打造了一支高效运转、富有服务意识的编辑出版团队。我们始终坚持：

善待作者——我们会把出版流程整理得清晰简明，为作者提供优厚的稿酬服务，解除作者的顾虑，安心写作，展现出最好的作品。

尊重作者——我们尊重每一位作者的技术实力和生活习惯，并会参照作者实际的工作、生活节奏，量身制定写作计划，确保合作顺利进行。

提升作者——我们打造精品图书，更要打造知名作者。博文视点致力于通过图书提升作者的个人品牌和技术影响力，为作者的事业开拓带来更多的机会。



联系我们

博文视点官网：http://www.broadview.com.cn　　CSDN官方博客：http://blog.csdn.net/broadview2006/
投稿电话：010-51260888　88254368　　投稿邮箱：jsj@phei.com.cn

新浪微博 weibo.com　@博文视点Broadview　　微信公众账号　博文视点Broadview

反侵权盗版声明

目　录

第1章 服务器监测

1.1 理解监测的意义

不论网站运维还是系统管理，服务器本身的运行状况都是我们需要掌控的基础资料。在《打造 Facebook》一书中，王淮介绍 Facebook 的工程师文化中有一句“Move Fast and Monitor Closely”。这个“Closely”有两层意思：其一是“即时”，要从系统开发的初期，就有意识地设计好配套的监测，并且逐步改进完善；其二是“深入”，监控不能停留在只监测主机负载、网卡流量的表面层次，而要尽可能地细化，以贴近系统的业务特性。

在系统的运行和发展过程中，监测的重要性得到更进一步的提高。运维人员总是倾向于为了保证稳定性而不轻易对系统做任何改动，所以，运维人员对稳定运行中的系统的一举一动都必须要有监测数据的支持。所有的大规模集群运作，包括近年来流行的自动化管理、弹性控制等理念，都要求我们首先对自己系统的细节有足够的了解——这都需要我们先熟悉监控。

1.2 通过命令了解系统的性能概况

Linux 操作系统有一些命令是初学者必须首先掌握的，而这些命令的输出结果，正是 Linux 精挑细选出来的关键信息。不论是对了解新上线的机器、收集应用测试数据，还是排除线上故障，这些基础信息都具有极大的帮助。

下列命令能够分别从网络、磁盘、CPU、内存、进程以及内核方面，提取出运维监控和排障时所需要的数据。

1.2.1 ifconfig

在服务器上运行命令的输出如下。

```
eth0    Link encap:Ethernet  HWaddr 44:37:e6:84:79:99
        inet addr:10.2.6.1  Bcast:10.2.6.255  Mask:255.255.255.0
        inet6 addr: fe80::4637:e6ff:fe84:7999/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:343813 errors:0 dropped:0 overruns:0 frame:0
        TX packets:30544 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:111460818 (111.4 MB)  TX bytes:5282490 (5.2 MB)
        Interrupt:17
lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:16436  Metric:1
        RX packets:20 errors:0 dropped:0 overruns:0 frame:0
        TX packets:20 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:1000 (1000.0 B)  TX bytes:1000 (1000.0 B)
```

结果中展示了服务器的网卡数目、IP 地址、MAC 地址、MTU 的大小、网卡收发包的情况（包括丢包和错误包等），这些都是服务器排障时首先要检查的数据。

与网卡有关的另一个重要命令是 ethtool，通过该命令可以监测网卡速度及工作模式。笔者就曾经碰到过千兆网卡工作模式变成百兆半双工导致服务性能严重下降的情况。

```
$ sudo ethtool eth0|egrep 'Speed|Duplex'
```

```
        Speed: 1000Mb/s
        Duplex: Full
```

1.2.2　w

命令输出如下。

```
15:09:49 up  4:53,  5 users,  load average: 0.99, 0.83, 0.69
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
chenryn  pts/2    :0               10:21    1:30m  0.20s  0.12s ssh
chenryn  pts/5    com8-31.opi.com  15:00 0.00s  0.10s  0.00s w
```

结果中包括了服务器的运行时间、当前用户及其运行的程序，以及 1 分钟、5 分钟和 10 分钟的平均负载。

平均负载是反映服务器当前运行状态最直观和简洁的数据。关于平均负载，proc 的 man 文档中是这样解释的：

```
The load average numbers give the number of jobs in the run queue (state R) or waiting for disk I/O (state D) averaged over 1, 5, and 15 minutes.
```

简单地说，Linux 会每 5 秒钟统计一次当前正在运行的任务（kernel 代码中的 TASK_RUNNING）数和正在等待磁盘 I/O 的任务（TASK_UNINTERRUPTIBLE）数，然后每 1、5 和 15 分钟计算一次平均值，这就是平均负载。

如果是多 CPU 的服务器，那么 Linux 会先对每个 CPU 进行平均负载计算，然后求和。

在单核时代，关于平均负载，有以下三条著名的经验准则。

◎ “敬请关注”法则（Need to Look into it）：0.70

如果平均负载大于 0.70 了，那么趁事情还没变得更糟，赶紧开始查原因吧。

◎ “立刻修复”法则（Fix this now）：1.00

要是负载已经高过 1.00 了，立马就扔掉其他事情先找这个问题并修复好。否则，没准今晚你就被半夜报警喊起来干活了。

◎ “靠，半夜 3 点了！”法则（Arrgh, it's 3AM WTF?）：5.0

如果负载超过 5.00 了，你的机器随时可能挂掉，而且就是在你半夜睡觉或者在盛大聚

会上正 happy 的时候！尽自己所能别让这种事情发生吧。

到了多核时代，这三个数据自然可以乘以你的 CPU 个数。注意，Linux 计算时，不会区分服务器上是几处理器还是几核心，所以又有两条新准则。

◎ “核心数=最大负载”法则（number of cores = max load）

多核系统上，负载不要高过设备的核心数。

◎ “啥核心都是核心”法则（cores is cores）

核心如何分布在 CPU 上不重要。两个四核心，四个双核心，八个单核心，效果是一样的。对于计算平均负载来说，它们都是八核心。

当然，以上准则都只是在普通情况下的经验总结，实际情况还需要大家自己跟踪分析。

比如当核心数多到好几十时，Linux 轮询各核心来统计单核负载的耗时长到足以让某些任务状态变化，这时候平均负载会普遍比实际情况偏低。针对这方面，Linux 内核社区已经有些补丁尽量调整算法。

比如在虚拟机环境下，因为时钟中断是由软件模拟的，在实际负载较高的时候，时钟中断会被高优先级抢占，导致平均负载计算时长比实际更长，计算结果也就偏高。笔者在虚拟机上见到过高达 2000+的平均负载，而此时服务器还可以 SSH 登录。

又比如某些单核时代的软件，像 Squid，本身无法运用多核的优势。这时候，其他的空余资源无法帮助服务提高性能，我们也只能关注单核负载的影响。

1.2.3 df

该命令常见的用法有两种，罗列如下。

```
$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda8       121G   14G  103G  12% /
udev            981M  4.0K  981M   1% /dev
tmpfs           397M  856K  396M   1% /run
$ df -Ti
Filesystem     Type      Inodes IUsed   IFree IUse% Mounted on
```

```
/dev/sda8      btrfs          0     0       0    - /
udev           devtmpfs   214792   513  214279    1% /dev
tmpfs          tmpfs      220462   445  220017    1% /run
```

如此分别查看挂载盘的目录、总容量和使用量、Inode 的总量和使用量，以及磁盘文件系统的类型。

在使用 Ext3 等非动态调节 inode 数目的文件系统的时候，关注 inode 的余额是一件很容易被遗忘的事情。但如果业务类型正好是以小文件为主的情况，发生磁盘容量未满却无法使用时，要记得查看 inode 数目。

一般来说，在 mkfs 格式化的时候，inode 数目就被自动分配完成了。Linux 会根据文件系统的 blocksize 和 bytes/inode 来自动计算。当然也可以在格式化的时候通过-N 参数自己指定这个数值。如何格式化磁盘的问题，大多数 Linux 命令手册上都有详细讲述，这里不详加讨论。

面对这种问题，更好的办法是提前规划，针对业务需求在上线前就准备好使用 reriserfs 等会自动动态调整 inodes 的文件系统来面对 inodes 的骤增。

此时还有一些特殊情况，我们可以特殊处理。比如 inodes 被大量不占空间的软连接使用，这时候我们可以用一个变通的办法，不格盘获取大量 inodes。

```
$ dd if=/dev/zero of=disk.img count=1024 bs=1024KB
$ mkfs.ext2 -N 5000000 -b 1024 -I 128 disk.img
$ mount -o loop disk.img /mnt/dd4inode
$ df -i | grep dd4inode
/dev/loop0      5003712      11  5003701       1% /mnt/dd4inode
```

其中“-b”和“-I”指定的都已经是 Linux 可接受的最小值，以获取尽可能大的 inodes。

又比如，采用 tmpfs 的缓存盘 inodes 不足的时候，没法用 reiserfs 来挂载内存，但是可以单独指定让 tmpfs 本身也放开这个 inodes 的限制。

在 Linux 源码文档中，对 tmpfs 的 inodes 有如下描述。

```
    nr_inodes: The maximum number of inodes for this instance. The default
               is half of the number of your physical RAM pages, or (on a
               machine with highmem) the number of lowmem RAM pages,
               whichever is the lower.
    These parameters accept a suffix k, m or g for kilo, mega and giga and
```

```
can be changed on remount. The size parameter also accepts a suffix % to limit
this tmpfs instance to that percentage of your physical RAM:
    the default, when neither size nor nr_blocks is specified, is size=50%
    if nr_inodes=0, inodes will not be limited.
```

所以，我们只要在 mount 的时候指定 nr_inodes 参数为 0，就可以不受 inodes 限制了，如下。

```
$ mount -t tmpfs -o size=2000M,mode=777,nr_inodes=0 tmpfs /tmpfs
$ df -i|grep tmpfs
tmpfs                    0      0      0    - /tmpfs
```

1.2.4 ps

ps 命令也有多种用法。最常见的是 ps auxfww，其输出如下。

```
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root     28332  0.0  0.1   8292  3196 pts/2    Ss   07:18   0:00 -bash
root     28834  0.0  0.0   6628  1168 pts/2    R+   07:20   0:00  \_ ps uafw
root      1046  0.6  1.1  72564 24040 tty7     Ss+ Mar06 196:54 /usr/bin/X :0
vt7 -br -nolisten tcp -auth /var/run/xauth/A:0-dRgv8a
root      1320  0.0  0.0   3756   588 tty1     Ss   Mar06   0:00 /bin/login --
root     11759  0.0  0.0   8192   372 tty1     S+   Mar06   0:00  \_ -bash
root       979  0.0  0.0   4632   300 tty6     Ss+  Mar06   0:00 /sbin/getty
-8 38400 tty1
root       975  0.0  0.0   4632   300 tty3     Ss+  Mar06   0:00 /sbin/getty
-8 38400 tty2
```

这样，能以树的形式显示进程间的父子关系，又能比 pstree 命令多出 CPU 和 MEM 等其他性能数据。

在这里，需要注意的是 VSZ 和 RSS 两列。

- VSZ(Virtual Memory Size)：指进程可以占用的内存地址空间的大小。
- RSS(Resident Set Size)：指进程实际占用的内存地址空间的大小。

但是要注意：RSS 中包括了共享库占用的内存大小，比如“libc”等。我们可以通过 pmap 命令看到进程调用的各种库占用的内存如下。