

# 欧阳归来

[博客园](#) [园子](#) [首页](#) [新随笔](#) [联系](#) [管理](#) [订阅](#) [XML](#)

随笔- 109 文章- 0 评论- 1

## 从苦逼到牛逼，详解Linux运维工程师的打怪升级之路

昵称: cosomsap

园龄: 5年

粉丝: 20

关注: 1

[+加关注](#)

2019年6月						
日	一	二	三	四	五	六
26	27	28	29	30	31	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	1	2	3	4	5	6

### 搜索

### 常用链接

[我的随笔](#)[我的评论](#)[我的参与](#)[最新评论](#)[我的标签](#)[更多链接](#)

### 我的标签

[Linux\(44\)](#)[LNMP\(6\)](#)[LAMP\(5\)](#)[监控\(5\)](#)[Zabbix\(4\)](#)[suse\(3\)](#)[工具\(3\)](#)[命令\(3\)](#)[CentOS\(3\)](#)[Mysql\(3\)](#)[更多](#)

### 随笔分类

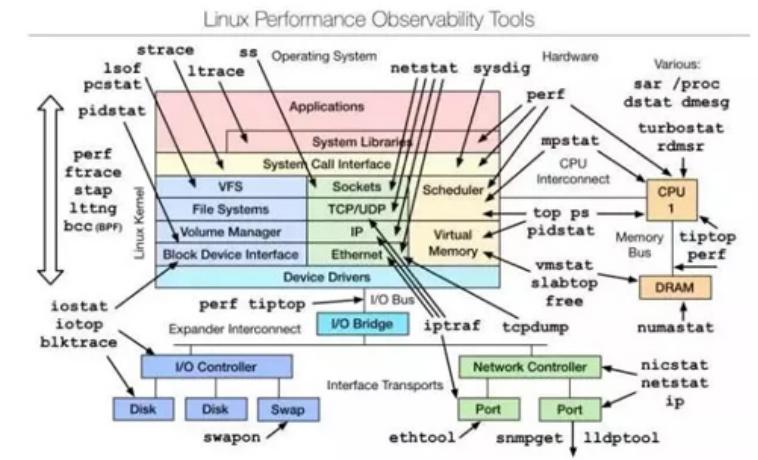
[ERP](#)[Linux\(51\)](#)[MYSQL\(7\)](#)[OA](#)[python\(2\)](#)[SAP\(3\)](#)[Zabbix\(5\)](#)[安全\(3\)](#)[操作系统\(6\)](#)[管理\(7\)](#)[软件工程\(7\)](#)[网络\(12\)](#)[网站优化\(1\)](#)

做运维也快四年多了，就像游戏打怪升级，升级后知识体系和运维体系也相对变化挺大，学习了很多新的知识点。

之前写过运维基础篇，发现对很多人收益挺大，接下来也写下关于这4年多的运维实践经验，从事了2年多游戏运维，1年多安全运维，1年大数据运维，相关行业信息不能算非常精通，但是熟悉和熟练还是相对可以的。

### 初级篇

8张Linux知识图谱，帮你读懂相关技术和性能



邮件服务器(9)  
自动化运维(26)

## 随笔档案

- 2018年5月 (1)
- 2018年1月 (1)
- 2017年12月 (1)
- 2017年11月 (1)
- 2017年10月 (1)
- 2017年9月 (1)
- 2017年8月 (3)
- 2017年4月 (1)
- 2016年12月 (1)
- 2016年11月 (3)
- 2016年10月 (2)
- 2016年8月 (1)
- 2016年6月 (2)
- 2016年5月 (1)
- 2016年3月 (3)
- 2016年1月 (1)
- 2015年12月 (2)
- 2015年11月 (1)
- 2015年10月 (9)
- 2015年9月 (2)
- 2015年8月 (4)
- 2015年7月 (6)
- 2015年6月 (15)
- 2015年5月 (7)
- 2015年4月 (9)
- 2015年3月 (8)
- 2014年8月 (7)
- 2014年7月 (11)
- 2014年6月 (4)

## 最新评论

1. Re:国内无线频谱资源分配图集  
谢谢分享

--J\_Y

## 阅读排行榜

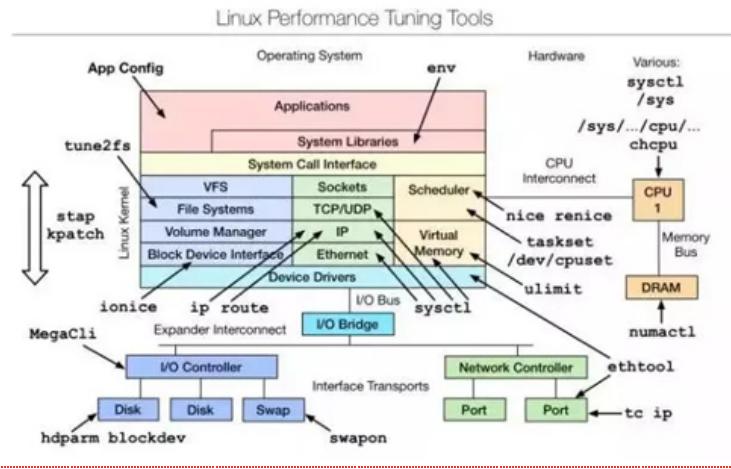
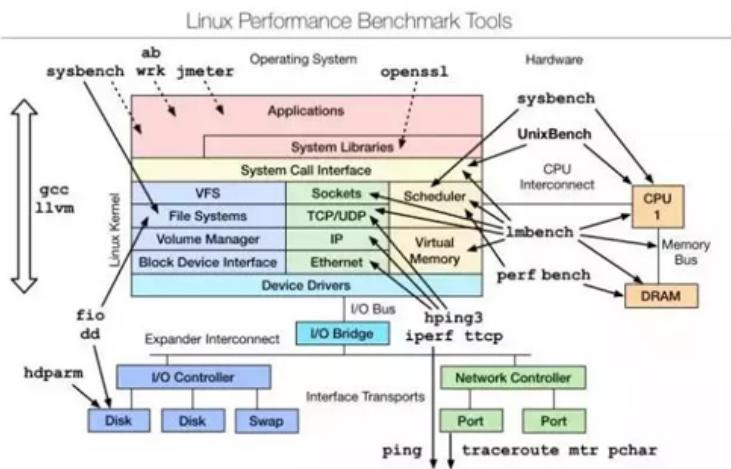
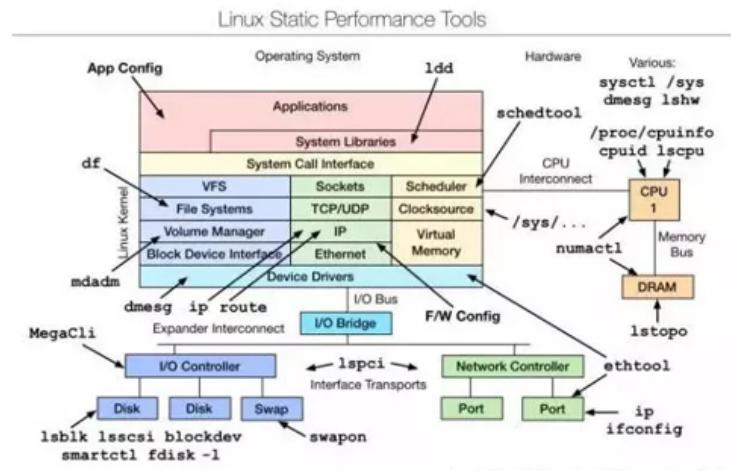
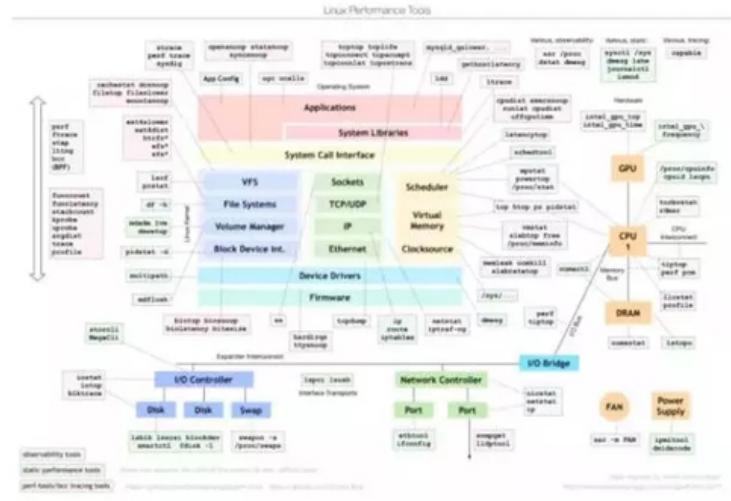
1. Windows server 2012 AD DS 搭建步骤(10968)
2. CentOS 6.5中linux grub修复(10371)
3. 走进Linux之systemd启动过程(7915)
4. Zabbix3.x安装图解教程(7233)
5. 互联网扫描器 ZMap 完全手册(6212)

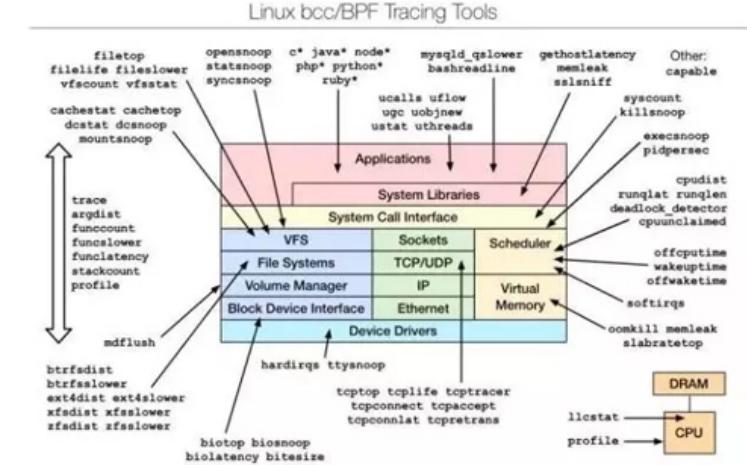
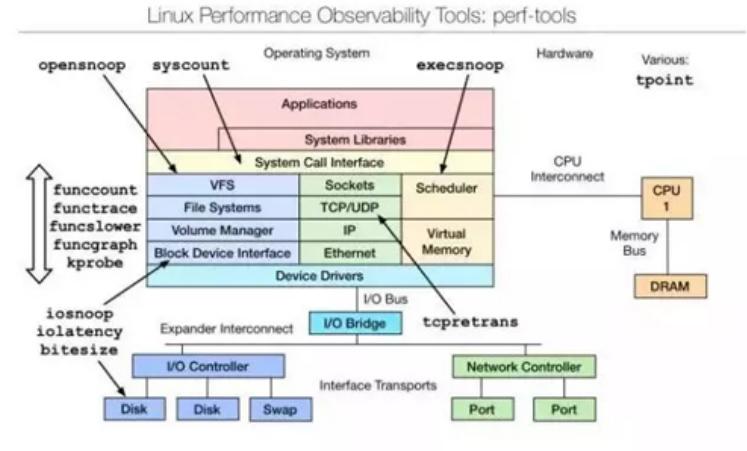
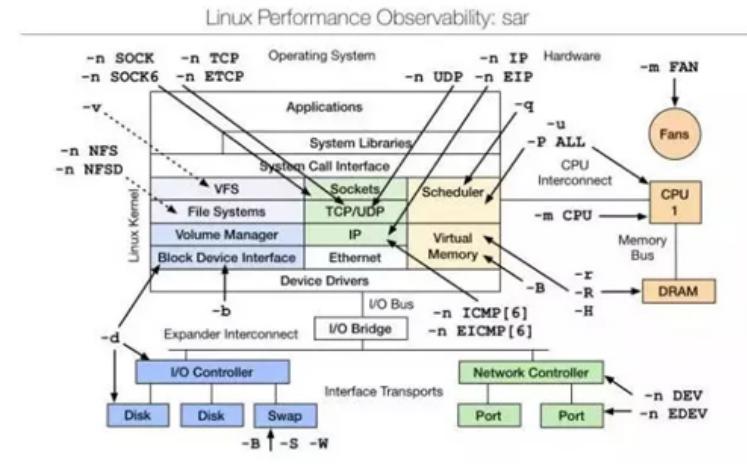
## 评论排行榜

1. 国内无线频谱资源分配图集(1)

## 推荐排行榜

1. 走进Linux之systemd启动过程(1)
2. 互联网扫描器 ZMap 完全手册(1)
3. 架构设计之如何评审架构设计说明书(1)
4. 自定义SAP用户密码规则(1)





详解Linux运维工程师入门级必备技能：<http://chenhao6.blog.51cto.com/6228054/1341458>

## 01rsync工具

很多地方经常会用到 rsync 工具，实施几台服务器的同步效果。我们公司就是使用这个工具完成服务器的游戏的服务端和客户端同步。

有几个文章例子：

rsync 强化技术(手动修改端口开启防火墙的情况下)并且通过脚本只同步需要的服务器：  
<http://chenhao6.blog.51cto.com/6228054/1322579>

inotify+rsync+mutt+msmtp 实现 Linux 文件或者目录自动更新并且实现发邮件给管理员：  
<http://chenhao6.blog.51cto.com/6228054/1298375>

## 02网络服务

服务有很多种，每间公司都会用到不同的，但基础的服务肯定要掌握，如 FTP、DNS、SAMBA、邮件，这几个大概学一下就行，LAMP 和 LNMP 是必须要熟练。

我所指的不是光会搭建，而是要很熟悉里面的相当配置才行，因为公司最关键的绝对是 Web 服务器，所以 Nginx 和 Apache 要熟悉，特别是 Nginx 一定要很熟悉才行，至少有些公司还会用 Tomcat，这个也最好学一下。

其实网络服务方面不用太担心，一般公司的环境都已经搭建好，就算有新服务器或让你整改，公司会有相应的文档让你参照来弄，不会让你乱来的，但至少相关的配置一定要学熟，而且肯定是编译安装多，那些模块要熟悉一下它的作用，特别是 PHP 那些模块。

上面两点只是基础，也是必要条件，不能说是工具，以下才是真正的要掌握的工具：

Samba 文件共享服务(共享脚本让你工作更轻松)：

<http://chenhao6.blog.51cto.com/6228054/1218028>

Linux Web 服务安装 Apache 思路(源码编译，自己定义服务)：

<http://chenhao6.blog.51cto.com/6228054/1223484>

FTP(持虚拟用户，并且每个虚拟用户可以具有独立的属性配置)：

<http://chenhao6.blog.51cto.com/6228054/1219713>

Linux 下构建 DHCP 服务器：

<http://chenhao6.blog.51cto.com/6228054/1217232>

### 03脚本语言

Shell 脚本和另一个脚本语言，Shell 是运维人员必须具备的，不懂这个连入职都不行，至少也要写出一些系统管理脚本，最简单也得写个监控 CPU，内存比率的脚本吧，这是最最基本了。

别以为会写那些猜数字和计算什么数的，这些没什么作用，只作学习意义，写系统脚本才是最有意义，而另一个脚本语言是可选的，一般是 3P，即 Python、Perl 和 PHP。

PHP 就不需要考虑了，除非你要做开发，我个人建议学 Python 会比较好，可实现自动化运维，Perl 是文本处理很强大，这两个学一个就行了。

Shell(一) 入门到复杂 自己做的各种脚本实例与解释：

<http://chenhao6.blog.51cto.com/6228054/1230337>

Shell(二)入门到复杂 脚本实例(计算器)：

<http://chenhao6.blog.51cto.com/6228054/1232070>

### 04sed 和 awk 工具

这两个工具必须要掌握，同时还要掌握正则表达式，这个就痛苦了，正则是最难学的表达式，但结合到 sed 和 awk 中会很强大。

在处理文本内容和过滤 Web 内容时十分有用，不过在学 Shell 的同时一般会经常结合用到的，所以学第 3 点就会顺便学第 4 点。

sed 简明教程：

<https://coolshell.cn/articles/9104.html>

### 05文本处理命令

sort 、 tr、 cut、 paste、 uniq、 tee 等必学，也是结合第 3 点脚本语言时一并学习的。

### 06数据库

首选 MySQL，别问我为什么不学 SQL Server 和 Oracle，因为 Linux 用得最多绝对是 MySQL，增删改查必学，特别要学熟练，其它方面可能不太需要，因为运维人员使用最多还是查，哪些优化和开发语句不会让你弄的。

MySQL(手动编译详细思路，以及增删改查、授权、备份还原)：

<http://chenhao6.blog.51cto.com/6228054/1225129>

### 07防火墙

防火墙也算是个难点，说难不难，说易不易，最重要弄懂规则，如果学过 CCNA 的朋友可能会比较好学，因为 iptables 也有 NAT 表，原理是一样的，而 FILTER 表用得最多，反正不学就肯定不合格。

防火墙(一)主机型防火墙

<http://chenhao6.blog.51cto.com/6228054/1239306>

防火墙(二)SNAT和DNAT

<http://chenhao6.blog.51cto.com/6228054/1240714>

### 08监控工具

我个人建议，最好学这 3 个：Cacti，Nagios，Zabbix，企业用得最多应该是 Nagios 和 Zabbix，反正都学吧，但 Nagios 会有点难，因为会涉及到用脚本写自动监控，那个地方很难。

CentOS 6.2+Nginx+Nagios，手机短信和QQ邮箱提醒：

<http://chenhao6.blog.51cto.com/6228054/1323192>

服务器集中检测Cacti：

<http://chenhao6.blog.51cto.com/6228054/1249302>

## 09集群和热备

这个很重要，肯定要懂的，但到了公司就不会让你去弄，因为新手基本不让你碰，集群工具有很多，最好学是 LVS，这是必学，最好也学 Nginx 集群、反向代理，还有热备，这个就有更多工具能实现了，像我公司是自己开发热备工具的。

MySQL 热备也要学，就是主从复制，这个要学懂整个流程一点也不容易，只照着做根本没意思。

MySQL主从同步，双主同步，如果服务器意外挂机，不同步怎么办：

<http://chenhao6.blog.51cto.com/6228054/1325247>

MySQL高性能压力测试(总结了好久)

：<http://chenhao6.blog.51cto.com/6228054/1314418>

Nginx 缓存配置及报错解决：

<http://chenhao6.blog.51cto.com/6228054/1329106>

## 10数据备份

工具有很多，但至少要把 RAID 的原理弄懂，特别是企业最常用的 1+0 或 0+1，自己做实验也要弄出来，备份工具有很多，如 tar、dump，最好多了解一下。

学会以上 10 点，应该可以入门了，有些技术会比较难学，例如 Apache 和 Nginx 中还有些很重要的技术，如系统调优、服务优化、程序优化，这些在没接触工作前很难学习到的。

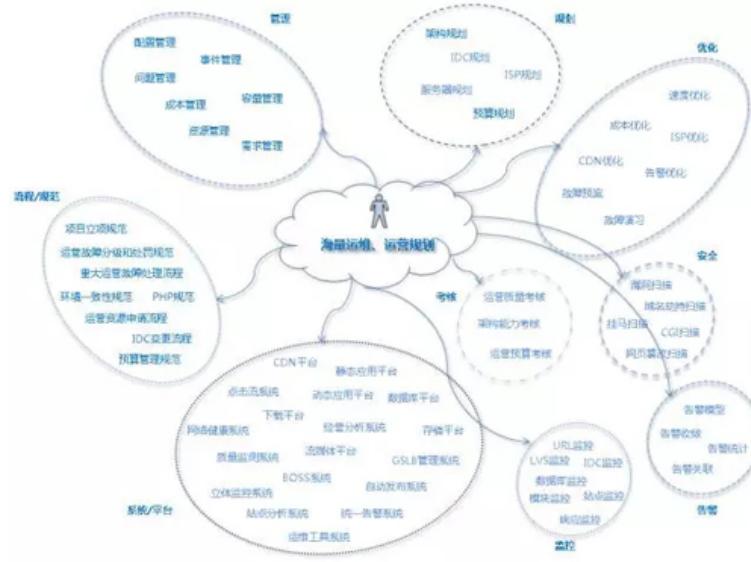
所以先把这 10 点学了吧，估计要学熟至少 3 个月不止，脚本部分会觉得很吃力了，我建议是先学熟 Shell，等工作后再学另一门脚本语言，这样会比较好。

以上就是踏入 Linux 运维工程师需要掌握的工具，还有很多工具要掌握的，但在学习环境中是很难学到。

最后我再提醒一下，这里所指的工具相当于技能，而不是像 Windows 或 Ubuntu 的图形化工具，还有学 Linux 就别装图形界面，这样虚拟机就不用吃太多内存，而且绝对不建议在真机上装 Linux，根本达不到学习效果。

## 中级篇

这部分来自我自己的面试经历和面试别人的经验总结。先附上运维思路拓扑图：



有些人认为，运维就是部署某个软件，设置些基础功能，就算会运维了。

举个例子：安装 LAMP，LNMP，就感觉部署方法我都掌握了。其实网上大多数都有一键安装脚本啥的根本没有啥技术含量，在面试官眼里，这些都不是你的亮点。

基本到了公司一般环境架构都是部署好的，很少需要你去变动环境架构。就算你安装好 LNMP 架构，你熟悉里面的原理吗？熟悉 Nginx 优化吗？熟悉 MySQL 优化吗？

再举个例子：我面试遇到的问题，面试官问你既然熟悉 LNMP 架构，那么 Nginx 反向代理的作用呢。

你应该不是说出懂这个软件和配置，你尽可能的说怎么优化，怎么深入提高网站性能：

- 使用反向代理可以理解为 7 层应用层的负载均衡，使用负载均衡之后可以非常便捷的横向扩展服务器集群，实现集群整体并发能力、抗压能力的提高。
- 通常反向代理服务器会带有本地 Cache 功能，通过静态资源的 Cache，有效的减少后端服务器所承载的压力，从而提高性能。

下面说说运维在工作中需要掌握的核心技术。需要注意的是，这是在工作中掌握的，在学习中很难掌握。

## 01第一条最主要的排错

- 分析部分程序不能运行或没有按预想结果运行的原因，对程序运行跟踪，查看系统调用的过程。
- 较深入的系统瓶颈点分析。

查看剩余内存：

1. free -m
2. # -/+ buffers/cache: 6458 1649
3. # 6458M 为真实使用内存 1649M 为真实剩余内存(剩余内存+缓存+缓冲器)
4. # linux 会利用所有的剩余内存作为缓存，所以要保证linux运行速度，就需要保证内存的缓存大小

系统信息：

1. uname -a # 查看Linux内核版本信息
2. cat /proc/version # 查看内核版本
3. cat /etc/issue # 查看系统版本
4. lsb\_release -a # 查看系统版本 需安装 centos-release
5. locale -a # 列出所有语系
6. locale # 当前环境变量中所有编码
7. hwclock # 查看时间
8. who # 当前在线用户
9. w # 当前在线用户
10. whoami # 查看当前用户名
11. logname # 查看初始登陆用户名
12. uptime # 查看服务器启动时间
13. sar -n DEV 1 10 # 查看网卡网速流量
14. dmesg # 显示开机信息
15. lsmod # 查看内核模块

硬件信息：

1. more /proc/cpuinfo # 查看cpu信息
2. lscpu # 查看cpu信息
3. cat /proc/cpuinfo | grep name | cut -f2 -d: | uniq -c # 查看cpu型号和逻辑核心数
4. getconf LONG\_BIT # cpu运行的位数
5. cat /proc/cpuinfo | grep 'physical id' | sort | uniq -c # 物理cpu个数
6. cat /proc/cpuinfo | grep flags | grep ' lm ' | wc -l # 结果大于0支持64位
7. cat /proc/cpuinfo | grep flags # 查看cpu是否支持虚拟化 pae支持半虚拟化 IntelVT 支持全虚拟化
8. more /proc/meminfo # 查看内存信息
9. dmidecode # 查看全面硬件信息
10. dmidecode | grep "Product Name" # 查看服务器型号

```

11. dmidecode | grep -P -A5 "Memory\s+Device" | grep Size | grep -v Range      # 查看内存插槽

12. cat /proc/mdstat                                # 查看软raid信息

13. cat /proc/scsi/scsi                            # 查看Dell硬raid信息(IBM、HP需要官方检测工具)

14. lspci                                         # 查看硬件信息

15. lspci|grep RAID                               # 查看是否支持raid

16. lspci -vvv |grep Ethernet                     # 查看网卡型号

17. lspci -vvv |grep Kernel|grep driver          # 查看驱动模块

18. modinfo tg2                                  # 查看驱动版本(驱动模块)

19. ethtool -i em1                             # 查看网卡驱动版本

20. ethtool em1

• 使用分析系统分析 Web 日志(如逆火软件)

• 分析系统性能瓶颈点(IO/Memory/CPU，常用工具，top 命令中 shift 组合键的特殊用 Sar/vmstat/iostat/ipcs)

```

日志管理常用命令：

1. history # 历时命令默认1000条
2. HISTTIMEFORMAT="%Y-%m-%d %H:%M:%S" # 让history命令显示具体时间
3. history -c # 清除记录命令
4. cat \$HOME/.bash\_history # 历史命令记录文件
5. lastb -a # 列出登录系统失败的用户相关信息 清空二进制日志记录文件 echo > /var/log/btmp
6. last # 查看登陆过的用户信息 清空二进制日志记录文件 echo > /var/log/wtmp 默认打开乱码
7. who /var/log/wtmp # 查看登陆过的用户信息
8. lastlog # 用户最后登录的时间
9. tail -f /var/log/messages # 系统日志
10. tail -f /var/log/secure # ssh日志

02优化

优化可以说是运维最吃香的技能，基本会优化的运维普遍工资很高，而且优化是要承担风险的，并不是网上搜个文章改一下配置文件或者参数就叫优化了，这样很容易造成宕机。

优化是根据实际的现场环境硬件各个参数进行部分优化，提高软件性能和网站性能。这个我只能讲半知半解，当时优化 MySQL 和 Tomcat 参数也是根据网上文章和官网文档查找参数在虚拟机上测试然后查看性能。

成本优化，性能优化，这里我给出 Tomcat 优化 JVM 参数(做过相应测试才放到现场环境的)：(记住无监控不调优)

1. -标准参数，所有jvm都应该支持
2. -X 非标，每个jvm实现都不同
3. -XX 不稳定参数，下一版本可能会取消
4. serial collector 单线程 序列化
5. parallel collector 多线程

启动 jvisualvm.exe 监控 dump 内存溢出：

1. -Xms:初始堆大小
2. -Xmx:最大堆大小

3. -Xss:线程栈大小

4. -XX:NewSize=n:设置年轻代大小

5. -XX:NewRatio=n:设置年轻代和年老代的比值，如3，表示年轻代：年老代比值1:3,年轻代占整个年轻代年老代和的1/4

6. -XX:SurvivorRatio=n:年轻代中的eden区与2个Survivor区的比值。

7. -XX:MaxPermSize=n:设置持久代大小

收集器设置：

1. -XX:+UseSerialGC: 设置串行收集器

2. -XX:+UseParallelGC:设置并行收集器

3. -XX:+UseConcMarkSweepGC:设置并发收集器

回收统计信息：

1. -XX:+PrintGC

2. -XX:+PrintGCDetails

3. -Xloggc:filename

Tomcat 优化，确认有几个 JVM 虚拟机：

1. set JAVA\_OPTS=

2. -Xms4g

3. -Xmx4g

4. -Xss512k

5. -XX:+AggressiveOpts 进攻型的优化选项,所有优化项都加上

6. -XX:+UseBiasedLocking 优化锁，基本都要选上，偏执锁

7. -XX:permSize=64m 原始区大小，最大300m 类多就设置大一点

8. -XX:MaxPermSize=300m

9. -XX:+DisableExplicitGC //System.gc() 不显示调用gc

10. -XX:+UseConcMarkSweepGC 使用cms缩短相应时间，并发收集，低停顿

11. -XX:+UseParNewGC 并行收集新生代的垃圾

12. -XX:+CMSParallelRemarkEnabled 在使用UseParNewGC的情况下，尽量减少mark的时间

13. -XX:+UseCMSCompactAtFullCollection 使用并发收集器时，开启对年老代的压缩，使碎片减少

14. -XX:LargePageSizeInBytes=128m 内存分页大小对性能的提升

15. -XX:+UseFastAccessorMethods get/set方法转成本地代码

16. -Djava.awt.headless=true 修复linux下tomcat处理图标时可能产生的bug

内存调优：

```
1. "C:\Program Files\Java\jdk1.8.0_31\bin\java" -XX:+DoEscapeAnalysis -  
XX:+EliminateAllocations -XX:+UseTLAB -XX:+PrintGCDetails -Didea.launcher.port=7540 "-  
Didea.launcher.bin.path=E:\java\IntelliJ IDEA 2016.3\bin" -Dfile.encoding=UTF-8 -  
classpath "C:\Program Files\Java\jdk1.8.0_31\jre\lib\charsets.jar;C:\Program Files\Java\jdk1.  
8.0_31\jre\lib\deploy.jar;C:\Program Files\Java\jdk1.8.0_31\jre\lib\ext\access-bridge-  
64.jar;C:\Program Files\Java\jdk1.8.0_31\jre\lib\ext\clldrdata.jar;C:\Program Files\Java\jdk1.8.  
0_31\jre\lib\ext\dnsns.jar;C:\Program Files\Java\jdk1.8.0_31\jre\lib\ext\jaccess.jar;C:\Progra  
m Files\Java\jdk1.8.0_31\jre\lib\ext\jfxrt.jar;C:\Program Files\Java\jdk1.8.0_31\jre\lib\ext\loc  
aledata.jar;C:\Program Files\Java\jdk1.8.0_31\jre\lib\ext\nashorn.jar;C:\Program Files\Java\j  
dk1.8.0_31\jre\lib\ext\sunec.jar;C:\Program Files\Java\jdk1.8.0_31\jre\lib\ext\sunfce_provide  
r.jar;C:\Program Files\Java\jdk1.8.0_31\jre\lib\ext\sunmscapi.jar;C:\Program Files\Java\jdk1.  
8.0_31\jre\lib\ext\sunpkcs11.jar;C:\Program Files\Java\jdk1.8.0_31\jre\lib\ext\zipfs.jar;C:\Pro  
gram Files\Java\jdk1.8.0_31\jre\lib\javaws.jar;C:\Program Files\Java\jdk1.8.0_31\jre\lib\jce.ja  
r;C:\Program Files\Java\jdk1.8.0_31\jre\lib\jfr.jar;C:\Program Files\Java\jdk1.8.0_31\jre\lib\jfx
```

```
swt.jar;C:\Program Files\Java\jdk1.8.0_31\jre\lib\jsse.jar;C:\Program Files\Java\jdk1.8.0_31\jre\lib\management-agent.jar;C:\Program Files\Java\jdk1.8.0_31\jre\lib\plugin.jar;C:\Program Files\Java\jdk1.8.0_31\jre\lib\resources.jar;C:\Program Files\Java\jdk1.8.0_31\jre\lib\rt.jar;E:\java\new\out\production\new;E:\java\IntelliJ IDEA 2016.3\lib\idea_rt.jar" com.intellij.rt.execution.application.AppMain aa.T02
```

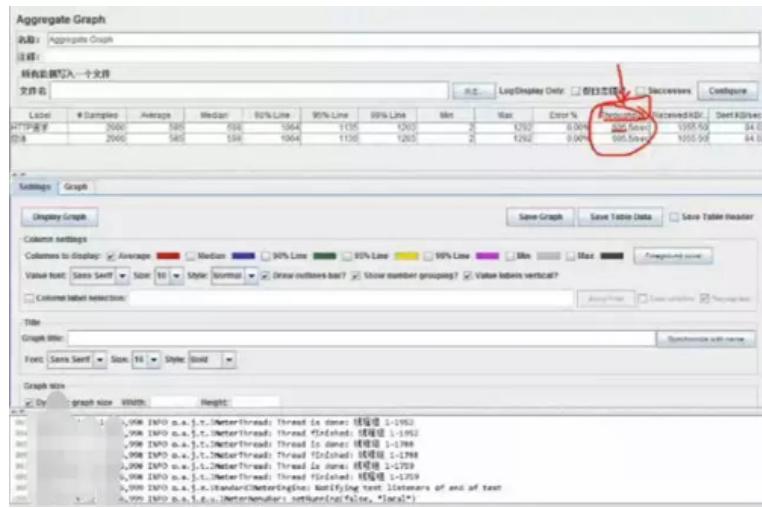
## 2. Heap

3. PSYoungGen total 38400K, used 3994K [0x00000000d5d80000, 0x00000000d8800000, 0x0000000100000000)
4. eden space 33280K, 12% used [0x00000000d5d80000, 0x00000000d61668b8, 0x00000000d7e00000)
5. from space 5120K, 0% used [0x00000000d8300000, 0x00000000d8300000, 0x00000000d8800000)
6. to space 5120K, 0% used [0x00000000d7e00000, 0x00000000d7e00000, 0x00000000d8300000)
7. ParOldGen total 87552K, used 0K [0x0000000081800000, 0x0000000086d80000, 0x0000000d5d80000)
8. object space 87552K, 0% used [0x0000000081800000, 0x0000000081800000, 0x0000000086d80000)
9. Metaspace used 3072K, capacity 4494K, committed 4864K, reserved 1056768K
10. class space used 329K, capacity 386K, committed 512K, reserved 1048576K

## 11. Heap

12. PSYoungGen total 38400K, used 1147K [0x00000000d5d80000, 0x00000000d8800000, 0x0000000100000000)
13. eden space 33280K, 3% used [0x00000000d5d80000, 0x00000000d5e9ecb8, 0x00000000d7e00000)
14. from space 5120K, 0% used [0x00000000d8300000, 0x00000000d8300000, 0x00000000d8800000)
15. to space 5120K, 0% used [0x00000000d7e00000, 0x00000000d7e00000, 0x00000000d8300000)
16. ParOldGen total 87552K, used 0K [0x0000000081800000, 0x0000000086d80000, 0x0000000d5d80000)
17. object space 87552K, 0% used [0x0000000081800000, 0x0000000081800000, 0x0000000086d80000)
18. Metaspace used 3072K, capacity 4494K, committed 4864K, reserved 1056768K
19. class space used 330K, capacity 386K, committed 512K, reserved 1048576K
20. 线程本地缓存使用eden的，开启就会使用更多

Tomcat 前任何参数没参加大概每秒 605，调优后大概每秒 435，接近 3 倍的结果。



### 03开发技能

优选 Shell 和 Python，现在 Shell 无法满足你的需求或者效率很低，那么选择自动化 Python 是最好的选择。现在普遍招聘需求要求，会写 Shell 或者 Python、Perl 脚本，个人选择还是选 Python。

Python 这门语言上手比较快，容易理解。在服务器管理工具上非常丰富，配置管理(Saltstack) 批量执行(Fabric、Saltstack) 监控(Zenoss、Nagios 插件) 虚拟化管理( python-libvirt ) 进程管理 (Supervisor) 云计算(OpenStack) .....

还有大部分系统 C 库都有 Python 绑定。

对于流程确定的事情，最终一定是纳入系统管理的体系，写成程序，成为系统的一部分。而不是无法复用游离与整体的各种脚本。

随着云计算时代的来临，中小型公司不需要运维了，大型公司没有工程开发能力的运维，是没有竞争力的。

最重要的学好 Python 可以涨工资，可以涨工资，可以涨工资。(重要的事情说三遍~)目前本人也是在学 Python，正在把以前 Shell 脚本的实例转换成 Python 脚本。

推荐《Python 实例手册》下载链接：

<http://down.51cto.com/data/2329173>

### 意识篇

#### 01安全意识

运维人员的权限很大，所以一定要保证帐号/私钥的安全：

- 最好使用加密工具存储。比如 Truecrypt、lpassword。
- 基于本地存储。切勿用网盘，也不建议用 lastpass 等。
- SSH 私钥添加密码。

#### 02磨刀意识

关于任何操作配置，最好先搞明白操作或配置的原理，然后再去操作。应一句话叫做“磨刀不误砍柴功”，而且对于类似的操作可以举一反三。

#### 03计划意识

复杂的变更操作比如多台主机以及牵涉到 san 存储，最好先作操作计划，写计划文档，详细到每条命令，然后请高手帮忙审核。

这样能最大程度使整个操作过程安全。如果是重要的客户业务系统，操作最好有回退方案，而一旦变更失败，客户可以在短时间内将业务回退。

#### 04记录分享意识

遇到自己认为较特殊的案例时，记得要写案例过程及分析的文档。也方便自己以后翻看，或者其他兄弟分享，作知识的传播以便于大家以后都能少走弯路。

#### 05监控意识

对运维来说，监控是非常重要的，监控是发现系统各种异常的眼睛，所以运维应该和监控紧密配合。

#### 06业务意识

尽量了解维护的各主机上的业务类型，以及各主机业务之间的关联性，因为任何维护工作都是为主机能提供业务服务的。

当某业务中断，能最快知道与此业务相关的主机群，从而缩小故障排查范围，最快定位故障。

并不是你技术很牛，学的技术很多很熟，就不代表你不需要运维意识，其实领导很看重运维意识的。

例如有没有做好备份、权限分配问题、平台测试情况、故障响应时间等，这些都是意识，而不是你学了很多技术自认大牛了，平台发现故障你以为很简单的问题喜欢处理就处理，不需要向其它部门反馈等。

领导不是看你的技术如何，而是看你的运维意识如何，你没运维意识，技术再牛也没用，只会让其它部门的人跟你不协调。

要知道做IT这行是苦逼的，需要无尽的学习，不学习只会被淘汰，不想被年轻的淘汰，就只能不断增值自己，不然不是你工资无法提升，而是你无法再从事这行。

这个世界，在悄悄惩罚不改变的人.....

分类: [自动化运维](#)

标签: [Linux](#), [运维经验](#)

[好文要顶](#)

[关注我](#)

[收藏该文](#)



cosomsap

关注 - 1

粉丝 - 20

[+加关注](#)

0

0

« 上一篇: [Linux文件系统概览](#)

» 下一篇: [ITSS相关的名词解释](#)

posted @ 2017-10-10 09:05 cosomsap 阅读(3177) 评论(0) 编辑 收藏

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请[登录](#)或[注册](#)，[访问](#)网站首页。

[【推荐】超50万C++/C#源码: 大型实时仿真组态图形源码](#)

[【前端】SpreadJS表格控件，可嵌入系统开发的在线Excel](#)

[【推荐】程序员问答平台，解决您开发中遇到的技术难题](#)

#### 相关博文：

- [· 详解Linux运维工程师应具备的十大技能](#)
- [· 详解Linux运维工程师必备技能](#)
- [· 详解Linux运维工程师](#)
- [· linux运维工程师](#)
- [· Linux 运维工程师的十个基本技能点](#)

#### 最新新闻：

- [· 太阳系的真正强者：木星](#)
  - [· 一线 | 京东618手机战报：荣耀数项第一 OPPO系冲击华为](#)
  - [· 老虎为什么是显眼的橙色？在猎物眼中其实是绿色的](#)
  - [· 25天！我国率先完成草地贪夜蛾染色体基因组测序和组装](#)
  - [· 苏姿丰谈接任AMD CEO职务后的顺利转型：专注于做正确的事情](#)
- [» 更多新闻...](#)

