psacct 工具

内容提要

1. 了解 psacct 软件包的组成
2. 掌握 lastcomm 和 sa 命令使用之前的配置方法
3. 掌握 lastcomm 和 sa 命令的使用
4. 掌握 ac 命令的使用
5. 熟悉日志文件 /var/account/pacct 和 /var/log/wtmp 的滚动配置

psacct 简介

管理员可以使用 psacct 软件包提供的工具监视所有用户执行的命令，包括 CPU 时间和内存占用，实现进程记帐功能。

psacct 软件包提供了三个进程活动监视工具: ac, lastcomm 和 sa。

accton 用于打开或关闭进程记帐功能，它是运行 lastcomm 和 sa 命令的前提。accton 命令是一个开关，运行一次将打开进程记帐功能，再运行一次将关闭进程记帐功能。为了方便管理员使用，CentOS 提供一个 init 脚本执行 accton 命令，您可以使用如下的命令打开或关闭进程记帐功能。

```
# service psacct start
# service psacct stop
```

- 为避免进程记帐日志文件过大，通常在使用进程记帐功能时才打开，不用时就将其关闭。
- 默认情况下，psacct 服务没有打开。若您想在系统启动时就打开进程记帐功能，请执行如下命令：

```
chkconfig acct on
```

lastcomm 和 sa 命令默认从进程记帐文件 /var/account/pacct 中读取数据，此文件为二进程文件。同时系统提供了 /etc/logrotate.d/psacct 脚本用 cron 实现进程记帐文件的滚动。

ac 命令从/var/log/wtmp 文件中读取数据，此文件也为二进程文件。

lastcomm

lastcomm 命令格式

lastcomm [http://www.proxyserve.net/index.php?q=aHR0cDovL21hbi5jeC9sYXN0Y29tbQ%3D%3D] 命令用于从 /var/account/pacct 搜索并显示以前执行过的命令信息。

命令格式为：

```
lastcomm [ Command ] [ Username ] [ Terminal ]
```

lastcomm 命令显示的列表被限制在以下范围：

- command 参数指定命令。
- Username 参数指定用户执行的命令。
- Terminal 参数指定由终端发出的命令。

当三者易发生混淆时，可以使用如下的参数格式：

```
lastcomm [ --command name ] [ --user name ] [ --tty name ]
```

默认情况下，各个参数之间是"或"的关系，可以使用 - - strict-match 参数实现参数表的精确匹配，即各个参数之间是"与"的关系。

lastcomm 使用举例

1、显示所有记录在 /var/account/pacct 文件中以前执行过的命令信息

```
# lastcomm
bash     S     root     tty1       0.03 secs Tue Jan 22 12:17
ls               crq      pts/0      0.00 secs Tue Jan 22 11:49
ac               osmond   pts/1      0.00 secs Tue Jan 22 10:59
sh       F     root     pts/0      0.00 secs Tue Jan 22 09:50
crond    SF    root     __         0.01 secs Tue Jan 22 09:50
sadc     S     root     __         0.01 secs Tue Jan 22 09:50
.....................
```

2、显示 tty1 终端上的 root 用户执行的所有命令的信息

```
# lastcomm tty1 root --strict-match
bash     S     root     tty1       0.03 secs Tue Jan 22 12:17
clear          root     tty1       0.00 secs Tue Jan 22 12:17
ls             root     tty1       0.00 secs Tue Jan 22 12:17
bash     F     root     tty1       0.00 secs Tue Jan 22 12:17
id             root     tty1       0.00 secs Tue Jan 22 12:17
.....................
```

3、显示 pts/1 终端上的 osmond 用户执行的名为 ac 的命令的信息

```
# lastcomm ac osmond pts/1 --strict-match
ac               osmond   pts/1      0.00 secs Tue Jan 22 10:59
ac               osmond   pts/1      0.00 secs Tue Jan 22 09:53
```

lastcomm 输出字段说明

1. 命令名
2. 当命令执行时记帐设备收集的标志：
   - S -- 命令由超级用户执行
   - F -- 命令由 fork 产生，但是没有 exec （执行）
   - D -- 命令终止并生成一个core（核心）文件
   - X -- 命令被 SIGTERM 信号终止
3. 运行进程的用户名
4. 运行进程的终端
5. 这个进程用的 CPU 时间
6. 进程开始的时间

sa

sa 的命令格式及输出

sa [http://www.proxyserve.net/index.php?q=aHR0cDovL21hbi5jeC9zYSg4KQ%3D%3D] 命令从 /var/account/pacct 原始记帐数据文件中读取信息并进行信息汇总。

sa 命令的显示结果可以包含以下字段：

1. calls -- 命令的调用次数。
2. 记帐设备收集的信息：
   - re -- 实际使用时间（单位为分钟）。
   - cpu -- 通常简写为 cp，表示用户和系统时间之和（单位为分钟）。
   - k -- 平均 CPU 时间（单元的大小为1K）。
   - u -- 用户 CPU 时间（单位为分钟）。
   - s -- 系统 CPU 时间（单位为分钟）。
3. command -- 调用的命令。

不带任何参数的 sa 命令，将显示 calls、re、cpu、k 和 command 字段，以 CPU 字段排序。并将只用过一次的命令放在 ***other 类别中显示。

命令格式如下：

```
sa [ 参数 ]
```

常用参数：

- **-a** ：显示所有命令的名称（包括那些带有不可打印字符的和只用过一次的命令）。
- **-c** ：同时显示百分比字段。
- **-l** ：将 CPU 时间字段拆分为系统时间和用户时间两个字段显示。
- **-t** ：同时显示实际使用时间与 CPU 总时间之比，即 re/cp。
- **-u** ：忽略所有其它参数并显示每个命令执行时的 CPU 时间。
- **-m** ：显示每个用户的汇总信息。
- **--sort-real-time** ：将输出按实际使用时间进行排序。
- **-b** ：将输出按用户和系统时间的总和除以调用次数来进行排序，即 cp/calls。
- **-k** ：将输出按平均 CPU 时间进行排序。
- **-n** ：将输出按命令的调用次数进行排序。
- **-r** ：将输出按逆序排列，可与其他排序参数一起使用。
- -b、-k、-n 和 --sort-real-time 参数确定如何进行输出排序。
- 如果在命令行中指定了不止一个排序参数，则只有最后那个参数生效。

sa 使用举例

1、要对 /var/account/pacct 文件命令进行记帐汇总

```
# sa
    1355     3870.84re       0.79cp      778k   # 所有命令的汇总行
       2      718.17re       0.67cp      542k   top
     134        0.05re       0.03cp      421k   sadc
     169       13.12re       0.02cp     1318k   crond*
       4        0.23re       0.02cp     1082k   find
      12        0.71re       0.01cp      636k   netstat
.....................
       1        0.00re       0.00cp      925k   ***other
```

2、改变汇总信息的输出列

```
# 同时显示百分比字段
# sa -c
    1356  100.00%     3870.84re  100.00%       0.79cp  100.00%      777k
       2    0.15%      718.17re   18.55%       0.67cp   85.45%      542k   top
     134    9.88%        0.05re    0.00%       0.03cp    3.22%      421k   sadc
     169   12.46%       13.12re    0.34%       0.02cp    2.95%     1318k   crond*
       4    0.29%        0.23re    0.01%       0.02cp    2.63%     1082k   find
      12    0.88%        0.71re    0.02%       0.01cp    0.68%      636k   netstat
.....................
# 将 CPU 时间字段拆分为系统时间和用户时间两个字段显示
# sa -l
    1357     3870.84re       0.19u       0.60s      777k
       2      718.17re       0.19u       0.49s      542k   top
     134        0.05re       0.00u       0.03s      421k   sadc
     169       13.12re       0.00u       0.02s     1318k   crond*
       4        0.23re       0.00u       0.02s     1082k   find
      12        0.71re       0.00u       0.00s      636k   netstat
```

```
.....
# 同时显示实际使用时间与 CPU 总时间之比
# sa -t
    1358     3870.84re       0.79cp  4923.7re/cp      777k
       2      718.17re       0.67cp  1069.2re/cp      542k   top
     134        0.05re       0.03cp     1.8re/cp      421k   sadc
     169       13.12re       0.02cp   566.3re/cp     1318k   crond*
       4        0.23re       0.02cp    11.0re/cp     1082k   find
      12        0.71re       0.01cp   133.3re/cp      636k   netstat
.....
```

3、记帐汇总信息进行排序输出

```
# 按实际使用时间排序
# sa --sort-real-time
    1362     3870.85re       0.79cp      776k
       2     2960.04re       0.00cp      507k   login
       2      718.17re       0.67cp      542k   top
       4      145.72re       0.00cp     1037k   info
     170       13.12re       0.02cp     1318k   crond*
.....

# 按CPU时间除以调用次数排序
# sa -b
    1361     3870.85re       0.79cp      777k
       2      718.17re       0.67cp      542k   top
       4        0.23re       0.02cp     1082k   find
       2        0.45re       0.00cp     1240k   ntsysv
       4        0.65re       0.00cp     1142k   bash
.....

# 按平均 CPU 时间排序
# sa -k
    1363     3870.85re       0.79cp      776k
      10        0.02re       0.00cp     2221k   rpmq
      12       12.28re       0.00cp     2008k   sendmail
      12        0.01re       0.00cp     1465k   troff
     170       13.12re       0.02cp     1318k   crond*
      12        0.01re       0.00cp     1269k   grotty
.....

# 按命令的调用次数排序
# sa -n
    1364     3870.85re       0.79cp      776k
     170       13.12re       0.02cp     1318k   crond*
     135        0.05re       0.03cp      421k   sadc
     118        0.37re       0.00cp      414k   sa
      56        0.00re       0.00cp      855k   bash*
      48        0.00re       0.00cp      407k   runlevel
.....

# 按命令的调用次数排序（逆序）
# sa -nr
    1366     3870.85re       0.79cp      775k
       1        0.00re       0.00cp      925k   ***other
       2        0.00re       0.00cp      441k   free
       2        0.00re       0.00cp      501k   gzip
       2        0.00re       0.00cp      919k   kbd_mode
.....
```

4、显示包含用户名的进程汇总信息

```
# 显示每个用户执行的每个命令的信息
# sa -u
root       0.01 cpu     1125k mem psacct
root       0.01 cpu     1112k mem service
osmond     0.01 cpu      932k mem sar
osmond     0.00 cpu      926k mem iostat
osmond     0.00 cpu      927k mem mpstat
osmond     0.00 cpu      441k mem free
osmond     0.00 cpu      439k mem uptime
osmond     0.01 cpu     1047k mem ps
crq        0.00 cpu      518k mem id
crq        0.00 cpu      605k mem bash             *
.....

# 显示每个用户执行命令的汇总信息
# sa -m
           1369     3870.85re       0.79cp      775k
root       1091     3693.33re       0.77cp      746k
osmond      224      177.00re       0.02cp      928k
crq          54        0.52re       0.00cp      724k
```

可以通过查看 re, k, cp/cpu 等字段的值找出可疑的活动。例如：

1. 某个用户或某个命令占用了所有的 CPU时间
2. 如果某个命令的 CPU 时间和内存使用在不断增加，说明命令存在问题

ac

ac 命令及格式

ac 命令从 /var/log/wtmp 文件中的登录和退出时间记录计算并输出用户的连接时间和总计连接时间。

记帐文件 /var/log/wtmp 由 init 和 login 维护。ac 和 login 均不生成 /var/log/wtmp 文件。若记帐文件不存在，则不做记帐工作。

请使用 info accounting 命令了解 GNU ac 和其他系统上的 ac 在输出上的不同。

文件 /var/log/wtmp 可能很快就变得非常大，所以默认情况下 CentOS 在 /etc/logrotate.conf 中 配置了此文件的日志滚动。配置片段如下：

```
/var/log/wtmp {
    monthly                 # 指定日志滚动周期为每月
    create 0664 root utmp   # 使用指定的文件模式创建新的日志文件
    rotate 1                # 只保留一个滚动日志备份，即只保留 /var/log/wtmp.1
}
```

命令格式：

```
ac［参数］
```

常用参数：

- **-d**：为每天输出一个总计时间。
- **-p**：为每个用户输出总计时间，并在最后追加一个所有用户的总计时间。
- **-a**：输出每天的记录，而不忽略没有登录活动的日子。
- **-y**：在显示日期时输出年份。
- **-z**：显示值为0的类别总计（除了全部总计）。默认禁止输出值为0的总计。
- **userlist**：显示指定用户的连接时间。多个用户之间用空格间隔，不允许有通配符。

ac 命令使用举例

```
# 显示所有用户的总计登录时间
$ ac
        total       66.53

# 显示用户 crq 的总计登录时间
$ ac crq
        total        0.04

# 显示用户 crq 和 osmond 的总计登录时间
$ ac crq osmond
        total       26.19

# 显示每个用户的总计登录时间
$ ac -p
        crq                                0.06
        osmond                            26.15
        root                              40.38
        total       66.58

# 显示指定用户的总计登录时间
$ ac -p crq osmond
        crq                                0.04
        osmond                            26.15
        total       26.19

# 为每天输出一个所有用户的总计登录时间
$ ac -d
Jan 21          total       30.82
Today           total       35.76

# d 和 y 参数的结合
$ ac -dyp
                osmond                            12.95
                root                              17.87
Jan 21 2008     total       30.82
                crq                                0.06
                osmond                            13.20
                root                              22.51
Today           total       35.76
$
```

为了读取日志文件 /var/log/wtmp，CentOS 在 SysVinit 软件包中还提供了 last 和 lastb 两个命令。

参考

- http://www.cyberciti.biz/tips/howto-log-user-activity-using-process-accounting.html [http://www.proxyserve.net/index.php?q=aHR0cDovL3d3dy5jeWJlcmNpdGkuYml6L3RpcHMvaG93dG8tbG9nLXVzZXItYWN0aXZpdHktdXNpbmctcHJvY2Vzcy1hY2NvdW50aW5nLmh0bWw%3D]
- 显示源文件
- 登录