

July 2014

ParnassusData is a software company

ORACLE DBA 必备技能 - 使用 OSWatcher(OSWBB)工具监控系统性能负载

Parnassus
诗檀

Creation Date: July 08, 2014



Document Control

Author

Biot Wang

Change Logs

Date	Author	Version	Change Log
------	--------	---------	------------

Reviewers

Name	Position
------	----------

ZhangYang Hu

HanJue Xu

Approvals

<Approver 1> Zhang Yang Hu

<Approver 2> _____

Distribution

Copy No.	Name	Location
----------	------	----------

Document Control	2
Author.....	2
Change Logs	2
Reviewers	2
Approvals.....	2
Distribution	2
为什么使用 OSWatcher?.....	4
本文目的	4
OSWatcher 概述	4
OSWatcher 安装 (Unix 平台)	5
OSWatcher - 在 Unix 平台上的启动/停止/卸载	6
OSWatcher Windows (OSFW) 概述.....	9
OSWatcher - shell 脚本概述.....	10
在解决节点重启问题中 OSWatcher 的作用	26
OSWatcher - 用于进程间通信超时(IPC Send Timeout)问题的分析使用	29
OSWatcher - 对 RAC 性能问题的分析解决	32
OSWatcher 的图形化输出 (OSWg, 后更名为 oswbba: OSWatcher Analyzer)	33
IPD/OS 工具 (后名为 CHM: Cluster Health Monitor 集群健康监视器)- OSWatcher 工具扩展	34
OSWatcher - 常见问题:.....	34
Find More	36
Conclusion	36

为什么使用 OSWatcher?

- OSWatcher 是 Oracle 开发并推荐的一种系统工具。它能用于辅助监控系统的资源使用情况。
- 如果没有安装 OSWatcher 工具或其 OSW 数据不可用的情况下,我们就不得不自己去尽可能多地收集相关信息以完成诊断工作。
- 如果客户有通过其他工具获取类似系统统计信息,那么我们就需要通过那些信息以完成诊断。
- 由于 OSWatcher 是针对运行 Oracle 关系型数据库服务器,用于侦查系统资源问题而量身订做的一种工具,且 Oracle 支持工程师也对分析 OSW 的统计数据更为熟练。因此 OSWatcher 的使用也更受欢迎及推荐。

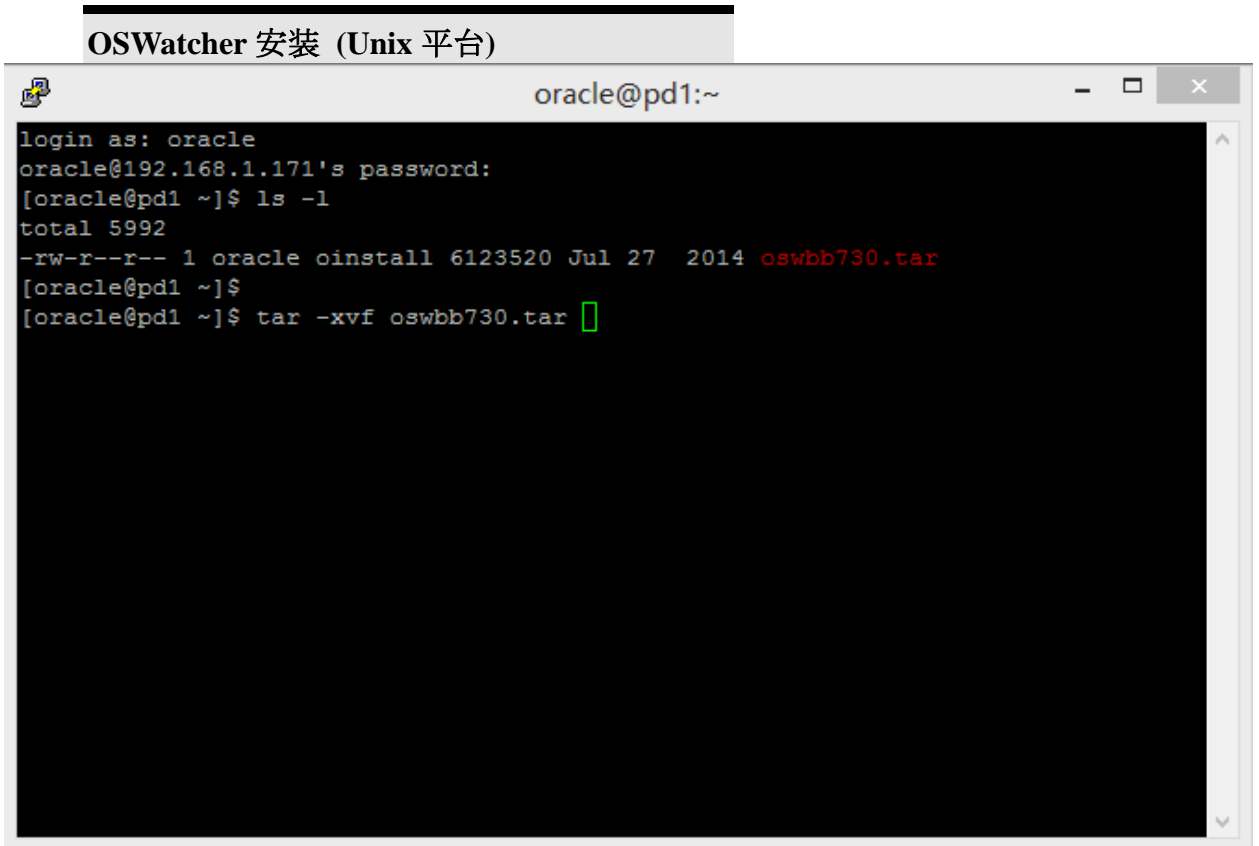
本文目的

1. 对于如何使用 OSWatcher 来诊断系统资源及调度问题进行概述。
2. 对于如何使用 OSWatcher 来解决节点重启/ORACLE 集群 (RAC) 性能问题进行概述。

OSWatcher 概述

- OS Watcher (OSW) 工具实际是由一系列相关 Unix shell 脚本和 Windows 批处理文件组成。它们被用来收集归档 系统和网络计量信息,以此来辅助支持诊断性能问题。
- OSW 的操作会开启一些服务器后台进程用以定期收集与各个功能相关的操作系统数据。
- 可以从“OS Watcher Users’ Guide”中找到基于 Unix 平台的 OSW 工具下载地址 - 请看文档: **301137.1**
- 可以从“OS Watcher For Windows User Guide”中找到基于 Windows 平台的 OSW 工具下载地址 - 请看文档: **433472.1**
- OSW 工具也包含在 RAC-DDT 工具中,但作为可选组件并不会被 RAC-DDT 直接安装 - 请看文档“RACDDT User Guide”: **301138.1**

- OSW 需要被安装在每个需要进行数据收集的节点。
- 支持平台:
- OSW 已被认证可运行于以下平台:
- Unix 操作系统 (OSwatcher 版本号 2.1.2)
- AIX
- Tru64
- Solaris
- HP-UX
- Linux
- Windows (XP, 2003)

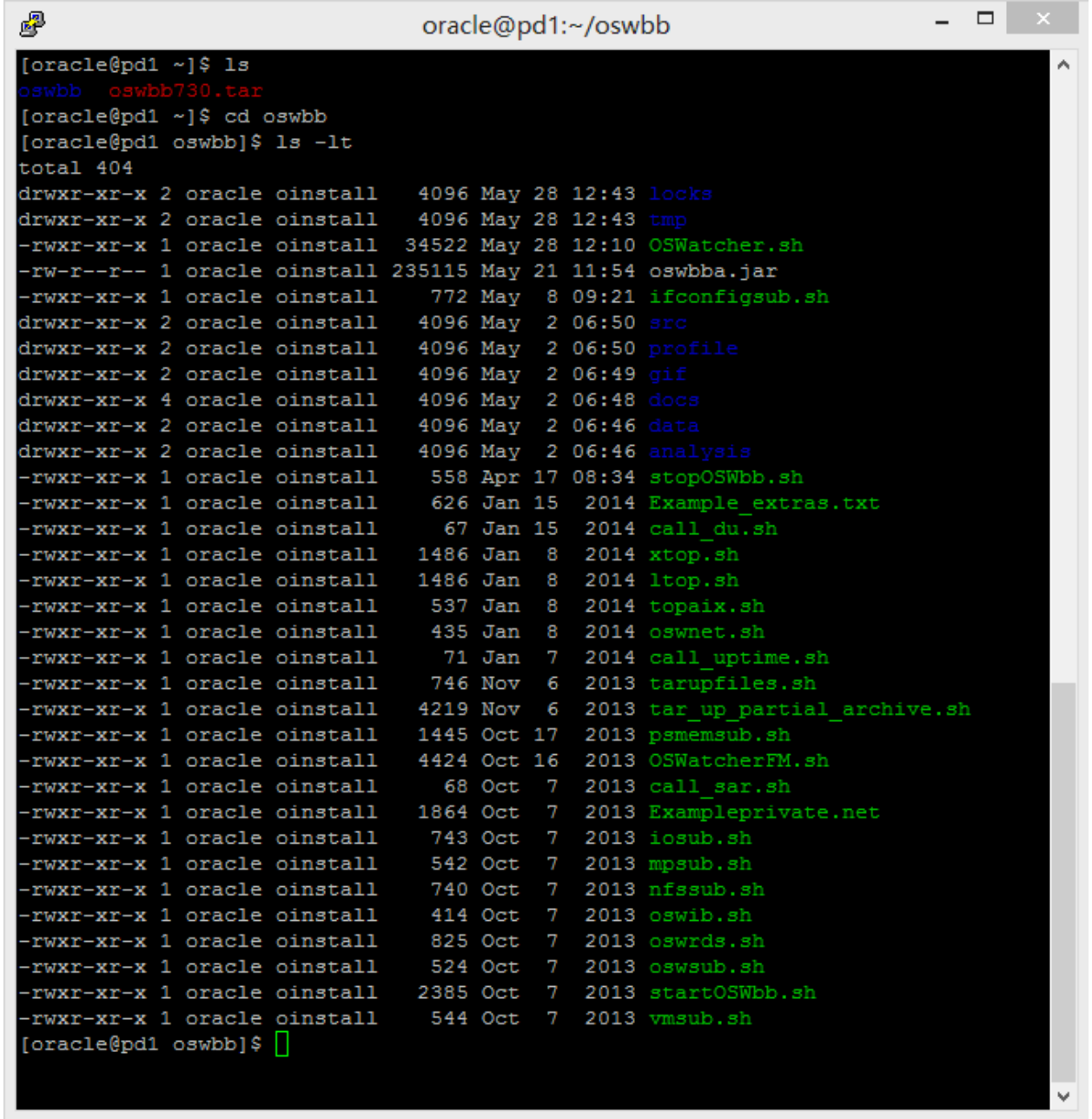


The image shows a terminal window titled "OSWatcher 安装 (Unix 平台)" with the user "oracle@pd1:~". The terminal output is as follows:

```
login as: oracle
oracle@192.168.1.171's password:
[oracle@pd1 ~]$ ls -l
total 5992
-rw-r--r-- 1 oracle oinstall 6123520 Jul 27  2014 oswbb730.tar
[oracle@pd1 ~]$
[oracle@pd1 ~]$ tar -xvf oswbb730.tar
```

- 从 OS Watcher Users' Guide (文档: 301137.1)下载 oswbb<xxx>.tar 文件
- 拷贝文件到需要安装 OSW 的每个节点下的一个目录中。
- 运行"tar -xvf oswbb<xxx>.tar"命令以提取/安装 OSW.

OSWatcher: 以下列出的解压后 oswbb 文件夹下的脚本文件



```
oracle@pd1:~/oswbb
[oracle@pd1 ~]$ ls
oswbb  oswbb730.tar
[oracle@pd1 ~]$ cd oswbb
[oracle@pd1 oswbb]$ ls -lt
total 404
drwxr-xr-x 2 oracle oinstall  4096 May 28 12:43 locks
drwxr-xr-x 2 oracle oinstall  4096 May 28 12:43 tmp
-rwxr-xr-x 1 oracle oinstall  34522 May 28 12:10 OSWatcher.sh
-rw-r--r-- 1 oracle oinstall 235115 May 21 11:54 oswbba.jar
-rwxr-xr-x 1 oracle oinstall   772 May  8 09:21 ifconfigsub.sh
drwxr-xr-x 2 oracle oinstall  4096 May  2 06:50 src
drwxr-xr-x 2 oracle oinstall  4096 May  2 06:50 profile
drwxr-xr-x 2 oracle oinstall  4096 May  2 06:49 gif
drwxr-xr-x 4 oracle oinstall  4096 May  2 06:48 docs
drwxr-xr-x 2 oracle oinstall  4096 May  2 06:46 data
drwxr-xr-x 2 oracle oinstall  4096 May  2 06:46 analysis
-rwxr-xr-x 1 oracle oinstall   558 Apr 17 08:34 stopOSWbb.sh
-rwxr-xr-x 1 oracle oinstall   626 Jan 15 2014 Example_extras.txt
-rwxr-xr-x 1 oracle oinstall    67 Jan 15 2014 call_du.sh
-rwxr-xr-x 1 oracle oinstall  1486 Jan  8 2014 xtop.sh
-rwxr-xr-x 1 oracle oinstall  1486 Jan  8 2014 ltop.sh
-rwxr-xr-x 1 oracle oinstall   537 Jan  8 2014 topaix.sh
-rwxr-xr-x 1 oracle oinstall   435 Jan  8 2014 oswnet.sh
-rwxr-xr-x 1 oracle oinstall    71 Jan  7 2014 call_uptime.sh
-rwxr-xr-x 1 oracle oinstall   746 Nov  6 2013 tarupfiles.sh
-rwxr-xr-x 1 oracle oinstall  4219 Nov  6 2013 tar_up_partial_archive.sh
-rwxr-xr-x 1 oracle oinstall  1445 Oct 17 2013 psmemsub.sh
-rwxr-xr-x 1 oracle oinstall  4424 Oct 16 2013 OSWatcherFM.sh
-rwxr-xr-x 1 oracle oinstall    68 Oct  7 2013 call_sar.sh
-rwxr-xr-x 1 oracle oinstall  1864 Oct  7 2013 Exampleprivate.net
-rwxr-xr-x 1 oracle oinstall   743 Oct  7 2013 iosub.sh
-rwxr-xr-x 1 oracle oinstall   542 Oct  7 2013 mpsub.sh
-rwxr-xr-x 1 oracle oinstall   740 Oct  7 2013 nfssub.sh
-rwxr-xr-x 1 oracle oinstall   414 Oct  7 2013 oswib.sh
-rwxr-xr-x 1 oracle oinstall   825 Oct  7 2013 oswrds.sh
-rwxr-xr-x 1 oracle oinstall   524 Oct  7 2013 oswsub.sh
-rwxr-xr-x 1 oracle oinstall  2385 Oct  7 2013 startOSWbb.sh
-rwxr-xr-x 1 oracle oinstall   544 Oct  7 2013 vmsub.sh
[oracle@pd1 oswbb]$
```

OSWatcher - 在 Unix 平台上的启动/停止/卸载

- 启动 OSW -

`./startOSWbb.sh 60 10`

启动 osw 工具以每 60 秒间隔收集一次数据并将最近 10 小时的数据记录入归档文件中。

./startOSWbb.sh

如果对此脚本运行不设参数，则默认设定为 30 48。也就是说以每 30 秒进行一次数据收集并在归档文件中保存最近 48 个小时的相关数据。

- **nohup ./startOSWbb.sh 60 10 &**

使用以上命令可让 OSW 能够在后台持续运行并在当前会话终止后不会被挂断。

- **停止 OSW -**

./ stopOSWbb.sh

以上命令是 Oracle 唯一支持的用于停止 OSW 的方法。

- **卸载 OSW -**

rm -fr oswbb

- OSWatcher 输出文件格式-<节点名>_<操作系统工具名>_YY.MM.DD.HH24.dat

如 - pd1.oracle.com_vmstat_14.07.27.1000.dat

在 Unix 平台上启动 OSWatcher

```
oracle@pd1:~/oswbb
oracle@192.168.1.171's password:
Last login: Sun Jul 27 03:39:50 2014 from 192.168.1.15
[oracle@pd1 ~]$ cd oswbb
[oracle@pd1 oswbb]$ ./startOSWbb.sh 30 2
[oracle@pd1 oswbb]$ Setting the archive log directory to /home/oracle/oswbb/archive

Testing for discovery of OS Utilities...
VMSTAT found on your system.
IOSTAT found on your system.
MPSTAT found on your system.
IFCONFIG found on your system.
[oracle@pd1 oswbb]$ NETSTAT found on your system.
TOP found on your system.

Testing for discovery of OS CPU COUNT
oswbb is looking for the CPU COUNT on your system
CPU COUNT will be used by oswbba to automatically look for cpu problems

CPU COUNT found on your system.
CPU COUNT = 2

Discovery completed.

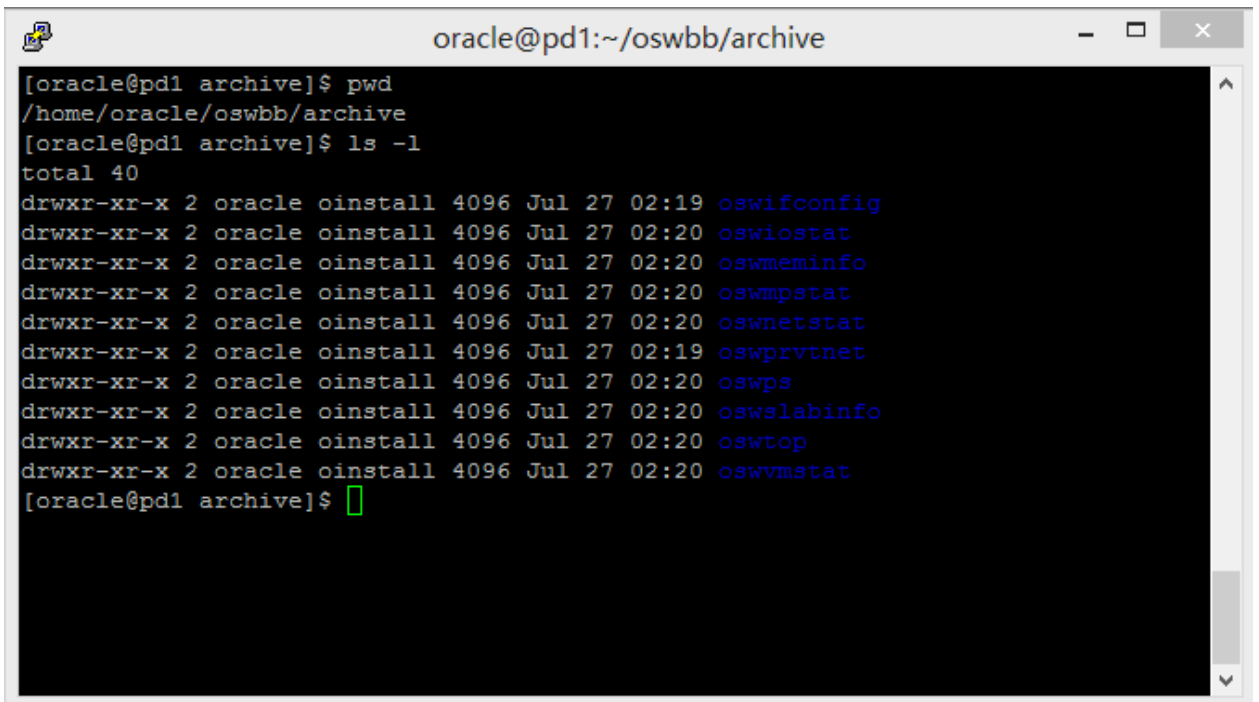
Starting OSWatcher Black Box v7.3.0 on Sun Jul 27 03:49:30 EDT 2014
With SnapshotInterval = 30
With ArchiveInterval = 2

OSWatcher Black Box - Written by Carl Davis, Center of Expertise,
Oracle Corporation
For questions on install/usage please go to MOS (Note:301137.1)
If you need further assistance or have comments or enhancement
requests you can email me Carl.Davis@Oracle.com

Data is stored in directory: /home/oracle/oswbb/archive

Starting Data Collection...

oswbb heartbeat:Sun Jul 27 03:49:35 EDT 2014
oswbb heartbeat:Sun Jul 27 03:50:05 EDT 2014
oswbb heartbeat:Sun Jul 27 03:50:35 EDT 2014
```

```
oracle@pd1:~/oswbb/archive
[oracle@pd1 archive]$ pwd
/home/oracle/oswbb/archive
[oracle@pd1 archive]$ ls -l
total 40
drwxr-xr-x 2 oracle oinstall 4096 Jul 27 02:19 oswifconfig
drwxr-xr-x 2 oracle oinstall 4096 Jul 27 02:20 oswiostat
drwxr-xr-x 2 oracle oinstall 4096 Jul 27 02:20 oswmeminfo
drwxr-xr-x 2 oracle oinstall 4096 Jul 27 02:20 oswmpstat
drwxr-xr-x 2 oracle oinstall 4096 Jul 27 02:20 oswnetstat
drwxr-xr-x 2 oracle oinstall 4096 Jul 27 02:19 oswprvtnet
drwxr-xr-x 2 oracle oinstall 4096 Jul 27 02:20 oswps
drwxr-xr-x 2 oracle oinstall 4096 Jul 27 02:20 oswslabinfo
drwxr-xr-x 2 oracle oinstall 4096 Jul 27 02:20 oswtop
drwxr-xr-x 2 oracle oinstall 4096 Jul 27 02:20 oswvmstat
[oracle@pd1 archive]$
```

- 当 OSWatcher 被首次启动时，其归档文件夹和 osw<工具名> 子文件夹会被建立。
- 除了 Linux 还会产生 oswslabinfo 和 oswmeminfo 子目录外，其它所有 Unix 平台的子目录结构都几乎相同。

OSWatcher Windows (OSFW) 概述

- 在 Windows 操作系统中，运行 OSW 实际上是执行一个带有各种计数器的 logman 工具命令的批处理文件。
- OSWatcher for Windows (OSFW) 已经对以下运行平台进行了认证:

Windows XP

Windows 2003

- 主要的控制执行文件是"OSWATCHER.BAT"文件，它会使用 Windows logman 工具建立并调度独立的计数器以收集特定类型数据。
- Logman 能够用于管理 "性能日志和警告"服务以建立和管理事件追踪会话日志及性能日志。

如: logman create counter perf_log -c "%Processor(_Total)% Processor Time"

PRM For Oracle 数据库灾难拯救工具下载: <http://www.parnassusdata.com/>

OSWatcher Windows (OSFW) 命令概述

- 可通过以下命令来安装 OSFW 相关文件

```
C:¥> unzip oswfw.zip
```

- 启动 OSWatcher -

```
C:¥> oswatcher 60 10
```

- 停止 Oswatcher

```
C:¥> oswatcher stop
```

- 使用以下命令以卸载 OSW 及其相关目录:

```
C:¥> oswatcher stop
```

```
C:¥> del /s osw
```

- 输出文件名为:

```
%计算机名%_OSW<性能对象>_MMDDHHMM_nnn.csv
```

- 发送文件给技术支持

压缩打包归档目录并上传给 Oracle 技术支持。

OSWatcher - shell 脚本概述

< OSWatcher.sh >

为主要控制执行脚本并负责产生数据收集进程。

可以通过运行 `startOSWbb.sh` 来启动此程序

< startOSWbb.sh >

```
./OSWatcher.sh $1 $2 $3 $4 &
```

\$1 = 以秒为单位的快照间隔时间

\$2 = 保存多少小时的归档数据

\$3 = (可选) 你希望使用的打包压缩工具, 在完成收集后 OSW 将使用其来打包压缩归档文件

\$4 = (可选) archive 目录地址, 程序会首先查找系统变量 OSWBB_ARCHIVE_DEST, 如果未设再看\$4 是否有设置, 如果仍为设置则保存文件至 oswbb 目录下默认路径。

< stopOSWbb.sh >

```
PLATFORM=`/bin/uname`
case $PLATFORM in
  AIX)
    kill -15 `ps -ef | grep OSWatch | awk '{print $2}'`
    ;;
  *)
    kill -15 `ps -e | grep OSWatch | awk '{print $1}'`
    ;;
esac
```

< OSWatcherFM.sh >

OSWatcher.sh 会调用此文件管理脚本程序。每 1 分钟，程序会醒来并查看当前小时时间是否已改变。如果我们进入新的一小时，程序会查看有多少文件被归档并移除那些超出（归档间隔时间）设定的归档文件。

以下是其代码片断=

```
numberOfFiles=`ls -t archive/oswvstat | wc -l`
numberToDelete=$((numberOfFiles-$archiveInterval)
if [ $numberOfFiles -gt $archiveInterval ]
  then
    ls -t archive/oswvstat/* | tail -$numberToDelete | xargs rm
```

例如 -

```
ls -l archive/oswvstat | wc -l = 50
```

我们希望归档最近 48 小时数据= 48 (归档间隔)

删除 50-48 = 最后 2 个文件.

< oswnet.sh >

脚本会被 OSWatcher.sh 调用。此脚本会连着运行 2 个 netstat 命令。

- netstat -a -i -n >> \$1
- netstat -s >> \$1

< oswsb.sh >

脚本会被 OSWatcher.sh 调用。此脚本是一个通用数据收集 shell 脚本。第一个参数(\$1)是设置数据收集器的输出文件名。第二个参数(\$2)是所要执行的操作系统工具。

- echo "zzz ***" `date` >> \$1
- \$2 >> \$1

- 例如: `./oswsub.sh archive/oswiostat/mbs1_iostat_09.01.04.0200.dat iostat 13`

< tarupfiles.sh >

- 此脚本生成 tarball 格式打包文件, 压缩并上传 Oracle 技术支持.

```
tar cvf osw_archive.tar archive
compress osw_archive.tar
hour=`date +%m%d%y%H%M.tar.Z`
mv osw_archive.tar.Z osw_archive_${hour}
```

- 不同平台之间, 操作系统工具命令 (如 top, ps, vmstat, iostat, mpstat) 输出可能不同.
- 针对在客户操作系统下 OSW 使用的操作系统命令, 我们需要很谨慎地查询 (man) 下相关命令手册, 这是因为你总可能发现一些小区别 (系统偏好), 但这往往会导致你做出错误的结论.

例如:

Linux 平台上的 **vmstat** 输出:

```
procs -----memory----- --swap-- -----io----- --system-- -----cpu-----
r  b  swpd  free  buff  cache  si  so  bi  bo  in  cs  us  sy  id  wa
st
0  0      0 254384 346096 958304  0  0   4  19  45  36  0  1 99  0
0
1  0      0 254384 346096 958304  0  0   6  36 1146 1578  0  0 100  0
0
0  0      0 254432 346096 958304  0  0   2  18 1117 1439  0  0 100  0
0
```

Solaris 平台上的 **vmstat** 输出:

```
procs  memory          page          disk          faults          cpu
r  b  w  swap  free  re  mf  pi  p  fr  de  sr  s0  s1  s2  s3  in  sy  cs  us  sy  id
0  0  0 11456 4120 1 41 19 1 3 0 2 0 4 0 0 48 112 130 4 14 82
0  0  1 10132 4280 0 4 44 0 0 0 0 0 23 0 0 211 230 144 3 35 62
```

OSWatcher - 对应操作平台不同输出有所不同

eg: iostat

```

oracle@pd1:~/oswbb/archive/oswiostat
[oracle@pd1 oswiostat]$ ls
pd1.oracle.com_iostat_14.07.27.0200.dat
pd1.oracle.com_iostat_14.07.27.0300.dat
[oracle@pd1 oswiostat]$ cat pd1.oracle.com_iostat_14.07.27.0200.dat | more
Linux OSWbb v7.3.0
zzz ***Sun Jul 27 02:20:25 EDT 2014
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.00    0.00   1.00    0.00    0.00   99.00

Device:            rrqm/s   wrqm/s   r/s     w/s     kB/s     kB/s avgrq-sz avgqu-sz
  await  svctm   %util
sda      0.00    0.00    0.00    0.00    0.00    0.00  0.00    0.00
  0.00    0.00    0.00
sda1     0.00    0.00    0.00    0.00    0.00    0.00  0.00    0.00
  0.00    0.00    0.00
sda2     0.00    0.00    0.00    0.00    0.00    0.00  0.00    0.00
  0.00    0.00    0.00
dm-0     0.00    0.00    0.00    0.00    0.00    0.00  0.00    0.00
  0.00    0.00    0.00
dm-1     0.00    0.00    0.00    0.00    0.00    0.00  0.00    0.00
  0.00    0.00    0.00

zzz ***Sun Jul 27 02:20:55 EDT 2014
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           1.01    0.00   1.51    0.00    0.00   97.49

Device:            rrqm/s   wrqm/s   r/s     w/s     kB/s     kB/s avgrq-sz avgqu-sz
  await  svctm   %util
sda      0.00    0.00    0.00    0.00    0.00    0.00  0.00    0.00
  0.00    0.00    0.00
sda1     0.00    0.00    0.00    0.00    0.00    0.00  0.00    0.00
  0.00    0.00    0.00
sda2     0.00    0.00    0.00    0.00    0.00    0.00  0.00    0.00
  0.00    0.00    0.00
dm-0     0.00    0.00    0.00    0.00    0.00    0.00  0.00    0.00
  0.00    0.00    0.00
dm-1     0.00    0.00    0.00    0.00    0.00    0.00  0.00    0.00
  0.00    0.00    0.00

zzz ***Sun Jul 27 02:21:25 EDT 2014
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0.50    0.00   1.01    0.00    0.00   98.49

Device:            rrqm/s   wrqm/s   r/s     w/s     kB/s     kB/s avgrq-sz avgqu-sz
  await  svctm   %util
    
```

iostat 命令被用于检测统计系统 i/o。

以该信息为分析依据, 改变系统设置以更好得平衡在物理磁盘和适配器之间的输入/输出负载.

- **iostat -xtc 5 2 (Solaris)**

```

extended disk statistics          tty   cpu
disk  r/s   w/s  Kr/s  Kw/s  wait  actv svc_t  %w  %b tin tout us sy wt id
    
```

PRM For Oracle 数据库灾难拯救工具下载: <http://www.parnassusdata.com/>

```
sd0  2.6  3.0  20.7  22.7  0.1  0.2  59.2  6  9  0
84  3  85  11  0
sd1  4.2  1.0  33.5  8.0  0.0  0.2  47.2  2  23
sd3  10.2  1.6  51.4  12.8  0.1  0.3  31.2  3  31
```

- 在输出中，我们需要关注什么？

在 iostat 他他输出中我们需要关注的值有:

- 每秒完成的读/写 I/O 设备次数 (r/s , w/s)
 - 设备平均繁忙度 (%b)
 - 平均服务时间(svc_t)
- 如果输出显示当前磁盘正持续保持高读写度且平均服务时间 (svc_t) 超过 30 毫秒，那么我们需要采取以下行动 -
 - 1) 调节应用以更有效使用磁盘输入输出
 - 2) 将文件移至其它 i/o 更快的磁盘/控制器或者直接替换 i/o 性能更佳的磁盘/控制器。

OSWatcher: OSWMEMINFO (Linux 平台)

```
oracle@pd1:~/oswbb/archive/oswmeminfo
[oracle@pd1 oswmeminfo]$ cat pd1.oracle.com_meminfo_14.07.27.0200.dat | less
zzz ***Sun Jul 27 02:20:25 EDT 2014
MemTotal:      4050940 kB
MemFree:       3539980 kB
Buffers:       25896 kB
Cached:        287540 kB
SwapCached:    0 kB
Active:        85988 kB
Inactive:      278000 kB
Active(anon):  53748 kB
Inactive(anon): 400 kB
Active(file):  32240 kB
Inactive(file): 277600 kB
Unevictable:   4444 kB
Mlocked:       4444 kB
SwapTotal:     6094840 kB
SwapFree:      6094840 kB
Dirty:         4 kB
Writeback:     0 kB
AnonPages:     55020 kB
Mapped:        19948 kB
Shmem:         528 kB
Slab:          104184 kB
SReclaimable: 24740 kB
SUnreclaim:   79444 kB
KernelStack:  1088 kB
PageTables:    7736 kB
NFS_Unstable:  0 kB
Bounce:        0 kB
WritebackTmp:  0 kB
CommitLimit:  8120308 kB
Committed_AS: 171532 kB
VmallocTotal: 34359738367 kB
VmallocUsed:   20468 kB
VmallocChunk: 34359709920 kB
HugePages_Total: 0
HugePages_Free: 0
HugePages_Rsvd: 0
HugePages_Surp: 0
Hugepagesize: 2048 kB
DirectMap4k:   8128 kB
DirectMap2M:  4186112 kB
zzz ***Sun Jul 27 02:20:55 EDT 2014
MemTotal:      4050940 kB
```

通过检查/proc/meminfo 可以确实诊断内存不足和碎片化问题。

- 以上截图相关系统使用 4GB RAM 和 6GB Swap 空间。

OSWatcher: OSWMEMINFO (Linux 平台)

- 在/proc/meminfo 文件中的几个主要参数(依版本不同有所不同) -

- **HighTotal:** 高位区在内存中所占总量. 高位内存区(Highmem)是指物理内存中高于(大约)860MB 的所有内存. 数据缓冲可以存放于这片内存区.
- **LowTotal:** 非高内存区内存所占总量.
- **LowFree:** 低位内存区中的空闲内存总量. 其内存可以被系统内核直接定位到. 所有内核数据结构都需要在低位内存区中存放.
- **SwapTotal:** 物理交换内存总量.
- **SwapFree:** 空闲交换内存总量.
- **HugePagesize:** 在内核中大页内存的配置大小.

单个大页内存的可变大小取决于内核版本和 linux 发布的硬件平台。

例如 - Linux Itanium (IA64) -256 MB , Linux x86-64 (AMD64, EM64T) -2 MB

- 在特定系统中大页内存的实际大小可通过 以下命令查看:

```
$ grep Hugepagesize /proc/meminfo
```

OSWatcher: OSWMPSTAT


```
oracle@pd1:~/oswbb/archive/oswmpstat
[oracle@pd1 oswmpstat]$ cat pd1.oracle.com_mpstat_14.07.27.0200.dat | less
Linux OSWbb v7.3.0
zzz ***Sun Jul 27 02:20:25 EDT 2014
Linux 2.6.32-200.13.1.el5uek (pd1.oracle.com) 07/27/2014

02:20:25 AM CPU %user %nice %sys %iowait %irq %soft %steal %idle
intr/s
02:20:26 AM all 1.00 0.00 3.00 0.00 0.50 0.00 0.00 95.50
238.38
02:20:26 AM 0 1.01 0.00 3.03 0.00 0.00 0.00 0.00 95.96
16.16
02:20:26 AM 1 1.01 0.00 3.03 0.00 0.00 0.00 0.00 95.96
2.02

02:20:26 AM CPU %user %nice %sys %iowait %irq %soft %steal %idle
intr/s
02:20:27 AM all 0.00 0.00 0.50 0.00 0.00 0.00 0.00 99.50
184.85
02:20:27 AM 0 0.00 0.00 1.01 0.00 0.00 0.00 0.00 98.99
0.00
02:20:27 AM 1 0.00 0.00 0.98 0.00 0.00 0.00 0.00 99.02
5.05

Average: CPU %user %nice %sys %iowait %irq %soft %steal %idle
intr/s
Average: all 0.50 0.00 1.75 0.00 0.25 0.00 0.00 97.49
211.62
Average: 0 0.51 0.00 2.02 0.00 0.00 0.00 0.00 97.47
8.08
Average: 1 0.50 0.00 1.99 0.00 0.00 0.00 0.00 97.51
3.54

zzz ***Sun Jul 27 02:20:55 EDT 2014
Linux 2.6.32-200.13.1.el5uek (pd1.oracle.com) 07/27/2014

02:20:55 AM CPU %user %nice %sys %iowait %irq %soft %steal %idle
intr/s
02:20:56 AM all 0.50 0.00 3.02 0.00 0.00 0.00 0.00 96.48
238.38
02:20:56 AM 0 1.01 0.00 2.02 0.00 0.00 0.00 0.00 96.97
16.16
02:20:56 AM 1 1.01 0.00 3.03 0.00 0.00 0.00 0.00 95.96
2.02
```

- **mpstat** 命令生成的报告信息可被用于判断当前进程负载是否均匀分布在每一个已存进程中，其多进程服务器是否正被有效利用。
- 系统活动情况报告(SAR: System activity Report)是除了 **mpstat** 命令之外，另一种可用命令。在 HP-UX 平台上 OSWatcher 常使用 SAR 工具命令。

使用 OSWatcher: OSWMPSTAT

Solaris 输出 =

PRM For Oracle 数据库灾难拯救工具下载: <http://www.parnassusdata.com/>

```
CPU minf mjf xcal intr ithr csw icsw migr smtx srw syscl usr sys wt idl
0      0      0      0      0      483 383 118      1      0      0      0      64
0      0 0      100
0      1268      0      0      486 382 414 42      0      0      0 2902      8
24 0      68
0      4      0      0      479 379 144      3      0      0      0      96
0      0 0 100
```

在 mpstat 中的几个关键统计值 -

- **icsw** - 当检测性能问题时, icsw 是一个更相关的统计参数。强制的上下文转换常发生在当进程/线程在其时间片持续执行时或当系统确定有更高运行级别的线程需要去运行时。
强制上下文转换 (**Involuntary context switches**) 产生意味着存在 CPU 的争用情况。
- **Smtx** - CPU 获取互斥锁 (mutex 内存锁) 的失败次数. Number of times a CPU failed to obtain a mutex (memory lock). 当其值超过平均每 CPU 200 次的话会导致系统时间的增加。
- **xcal** - 显示进程的交叉调用情况(当一个 CPU 通过打断的方式调用另一个进程)。如果其值超过 200 次每秒, 那么就说明需要检查这个有问题的应用了。

OSWatcher: OSWNETSTAT

```
oracle@pd1:~/oswbb/archive/oswnetstat
[oracle@pd1 oswnetstat]$ cat pd1.oracle.com_netstat_14.07.27.0200.dat | less
Linux OSWbb v7.3.0
zzz ***Sun Jul 27 02:20:25 EDT 2014
Kernel Interface table
Iface      MTU Met    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR
Flg
eth0       1500  0      6344   0      0      0      738   0      0      0
BMRU
eth1       1500  0         0   0      0      0       20   0      0      0
BMRU
lo         16436 0      1199   0      0      0      1199  0      0      0
LRU
Ip:
 6683 total packets received
 355 with invalid addresses
  0 forwarded
  0 incoming packets discarded
 6309 incoming packets delivered
 1931 requests sent out
Icmp:
  0 ICMP messages received
  0 input ICMP message failed.
ICMP input histogram:
  0 ICMP messages sent
  0 ICMP messages failed
ICMP output histogram:
Tcp:
 121 active connections openings
  7 passive connection openings
 120 failed connection attempts
  0 connection resets received
  2 connections established
 5767 segments received
 1869 segments send out
  0 segments retransmited
  0 bad segments received.
 120 resets sent
Udp:
 33 packets received
  0 packets to unknown port received.
  0 packet receive errors
 45 packets sent
UdpLite:
TcpExt:
  4 TCP sockets finished time wait in fast timer
 219 delayed acks sent
  5 packets directly queued to recvmsg prequeue.
 61156 packets directly received from backlog
 132860 packets directly received from prequeue
 4256 packets header predicted
```

netstat 命令显示当前 TCP/IP 网络连接和其相关协议的数据统计。

netstat 中的几个重要统计值 -

-冲突 Collisions (Collis), 输出包 Output packets (Opkts), 输入错误 Input errors (Ierrs), 输入包 Input packets (Ipkts)

网络冲突率(Network Collision Ratio) = Collis / Opkts (不应该超过 10%)

高冲突率意味着网络处于饱和状态。

误包率 Input Packet Error Rate = Ierrs / Ipkts.

如果主机丢包严重, 误包率很高(超过百分之 0.25). 那么就需要检查其相关集线器/ 交换机电缆等是否存在潜在问题。

-在大部分 UNIX 平台, 判断当前是否存在 UDP 端口缓存溢出和丢包问题. 可执行 `netstat -s` 或者 `netstat -su` 并按相应平台查找“udpInOverflows”, “packet receive errors” 或者 “fragments dropped (段删除)”的数据统计。

-最新文档提供了关于各种网络层问题的大量细节 - [Note 563566.1](#) Title: gc lost blocks diagnostics

OSWatcher: OSWPRVTNET

- 私有网络统计被用于检查内联网络是否可达
- \$cat pd1.oracle.com_prvtnet_14.07.27.0200.dat

zzz ***Tue Jan 6 14:00:02 IST 2009

traceroute to pd1-priv (172.168.1.191), 10 hops max, 40 byte packets

1 pd1-priv.oracle.com (172.168.1.191) 0.027 ms 0.012 ms 0.009 ms

traceroute to pd2-priv (172.168.1.192), 10 hops max, 40 byte packets

1 pd2-priv.oracle.com (172.168.1.192) 0.128 ms 0.108 ms 0.103 ms

- 请注意: 这个统计值并非被 OSWatcher 默认收集。
- 客户需要手动设置 OSWatcher 使其对于私有网络进行统计收集, 具体请参考 OSW 文件夹下的 Exampleprivate.net 文件。

OSWatcher - 启用对私有网络统计信息收集功能

- 对私有网络统计收集的设置 -

拷贝 OSW 文件夹下的 Exampleprivate.net 文件并重命名为 private.net.

PRM For Oracle 数据库灾难拯救工具下载: <http://www.parnassusdata.com/>

删除文件中其它行并保留对你的操作系统所需的行

例如 - 在 Linux 平台, 文件 `private.net` 有以下行信息:

```
-----  
traceroute -r -F -m 10 pd1-priv  
traceroe -r -F -m 10 pd2-priv  
rm locks/lock.file  
-----
```

- 请记住保留最后一行 - `rm locks/lock.file`
- 修改后保存 `private.net` 文件
- 修改模式: `Chmod 777 private.net` 以使得此文件具有执行权限
- 这样就启用了 OSWatcher 收集私有网络统计信息的功能.
- 如果在 `private.net` 建立并修改的同时 OSWatcher 仍在运行, 则需要重启 OSWatcher 以使其功能生效

OSWatcher: OSWPRVTNET

- 哪些收集信息是我们需要关注的
- 例 1: 查看端口是否开启并有应答:-

```
traceroute to pd2-priv (172.168.1.192), 10 hops max, 40 byte packets  
1 pd2-priv.oracle.com (172.168.1.192) 0.128 ms 0.108 ms 0.103 ms
```
- 例 2: 目标端并不在直连网络上, 所以需要检查目标端地址是否正确或交换机是否在相同的虚拟网络上 (或者需要检查其他问题):

```
Traceroute to pd3-priv (10.0.0.1), 10 hops max, 40 byte packets  
traceroute: host pd3-priv is not on a directly-attached network
```

- Example 3: 网络不可达(需检查路由表)

```
traceroute to pd3-priv (10.0.0.1), 10 hops max, 40 byte packets  
Network is unreachable
```

- Example 4: 在 Linux 上启用了 IPTables / 防火墙 - 请在 Linux 上禁用 IPTables

```
traceroute to pd3-priv (10.0.0.1), 30 hops max, 46 byte packets
```

Icmp checksum is wrong

OSWatcher: OSWPS

ps(进程状态 process state)命令列出了所有当前运行在系统上的进程并提供了关于 CPU 消耗, 进程状态, 进程优先级等信息。

OSWatcher: OSWPS

```
oracle@pd1:~/oswbb/archive/oswps
[oracle@pd1 oswps]$ cat pd1.oracle.com_ps_14.07.27.0200.dat | less
Linux OSWbb v7.3.0

zzz ***Sun Jul 27 02:20:25 EDT 2014
USER      PID    PPID  PRI  %CPU  %MEM    VSZ   RSS  WCHAN  S  STARTED      TIME  COMMAND
gdm       2716   2698   19   0.0   0.4   220900 16220 poll_s S 01:53:05 00:00:00 /usr/libexec/gdmgreeter
root     2757     1     0   0.0   0.3   256768 15656 poll_s S 01:53:07 00:00:00 /usr/bin/python -tt /usr/sbin/yum-updatesd
root     2701   2698   19   0.0   0.1   83984   7272 poll_s S 01:53:02 00:00:00 /usr/bin/Xorg :0 -br -audit 0 -auth /var/gdm/:0.Xauth -nolisten tcp vt7
root     2700     1     0   0.0   0.1   187828 4100 poll_s S 01:53:01 00:00:00 /usr/libexec/gdm-rh-security-token-helper
root     2405     1     0   0.1   0.1   154880 6440 poll_s S 01:52:59 00:00:00 /usr/bin/python ./hpssd.py
root     1770     1     0   0.1   0.1   12640  4452 poll_s S 01:52:43 00:00:00 iscsid
68       2305     1     0   0.1   0.1   31428  4364 poll_s S 01:52:55 00:00:01 hald
xfs      2537     1     0   0.0   0.0   20976  1796 poll_s S 01:53:00 00:00:00 xfs -droppriv -daemon
USER      PID    PPID  PRI  %CPU  %MEM    VSZ   RSS  WCHAN  S  STARTED      TIME  COMMAND
smmsp    2489     1     0   0.0   0.0   57712  1764 pause S 01:53:00 00:00:00 sendmail: Queue runner@01:00:00 for /var/spool/clientmqueue
rpcuser  2148     1     0   0.0   0.0   10172   792 poll_s S 01:52:54 00:00:00 rpc.statd
rpc      2113     1     0   0.0   0.0   8064    608 poll_s S 01:52:54 00:00:00 portmap
root      9        2     19   0.0   0.0     0      0 worker S 01:52:03 00:00:00 [events/0]
root     90        2     19   0.0   0.0     0      0 kjourn S 01:52:24 00:00:00 [kjournald]
root     88        2     19   0.0   0.0     0      0 worker S 01:52:24 00:00:00 [kdmflush]
root     84        2     19   0.0   0.0     0      0 worker S 01:52:24 00:00:00 [kdmflush]
root      8        2     139  0.0   0.0     0      0 watchd S 01:52:03 00:00:00 [watchdog/1]
root      7        2     19   0.0   0.0     0      0 ksofti S 01:52:03 00:00:00 [ksoftirqd/1]
root     70        2     19   0.1   0.0     0      0 scsi_e S 01:52:04 00:00:01 [scsi_ah_2]
root     68        2     19   0.0   0.0     0      0 scsi_e S 01:52:04 00:00:00 [scsi_ah_1]
root      6        2     139  0.0   0.0     0      0 migrat S 01:52:03 00:00:00 [migration/1]
root     53        2     19   0.0   0.0     0      0 scsi_e S 01:52:03 00:00:00 [scsi_ah_0]
root      5        2     139  0.0   0.0     0      0 watchd S 01:52:03 00:00:00 [watchdog/0]
root     47        2     19   0.0   0.0     0      0 worker S 01:52:03 00:00:00 [usbhid_resumer]
root     46        2     19   0.0   0.0     0      0 worker S 01:52:03 00:00:00 [ksnapd]
root     45        2     19   0.0   0.0     0      0 worker S 01:52:03 00:00:00 [kstriped]
root     44        2     19   0.0   0.0     0      0 worker S 01:52:03 00:00:00 [kpsmouse]
root      4        2     19   0.0   0.0     0      0 ksofti S 01:52:03 00:00:00 [ksoftirqd/0]
root     38        2     19   0.0   0.0     0      0 worker S 01:52:03 00:00:00 [crypto/1]
root     37        2     19   0.0   0.0     0      0 worker S 01:52:03 00:00:00 [crypto/0]
```

在 ps 命令输出中我们需要关注哪些? -

- ps 命令的输出信息将主要用于对于 RAC 的诊断。
- 进程状态为 S 表示睡眠状态(sleeping), D 表示不可中断的闲置状态(uninterrupted), R 表示运行状态(running), T 表示停止状态/被追踪状态(stopped/traced), 而 Z 则表示僵尸状态 (zombie).
- 实例/系统崩溃前的进程状态可能有助于原因的分析。

例如. - PS 命令输出中有进程显示 “T” 表示其进场当前处于停止状态或者被追踪状态.

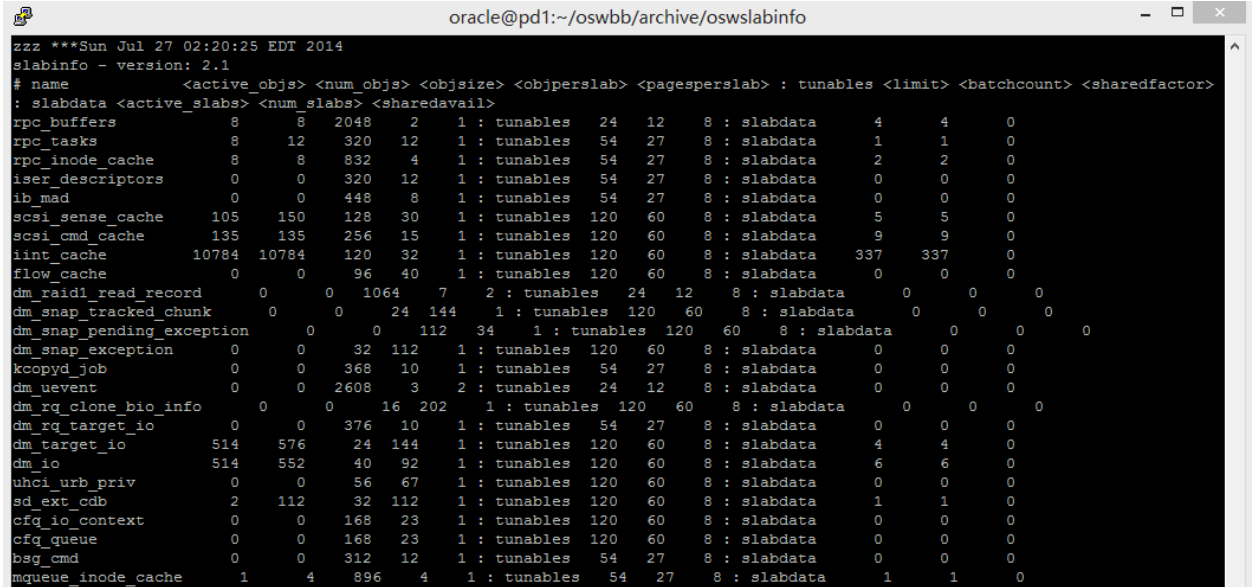
240001 T oracle 4001974 1 0 60 20 2debdf510 106608 08 Aug - 906:58 ora_lms1_srs1

- 进程优先级(PRI, NI)设定值也很重要(如查看 LMS 进程优先级)

如 - ps -efc | grep lms (在 Solaris 系统中) -

oracle 18098 1 RT 100 May 23 ? 39:30 ora_lms1_IEI10TRN2

OSWatcher: OSWSLABINFO (Linux 平台)



```
oracle@pd1:~/oswbb/archive/oswslabinfo
zzz ***Sun Jul 27 02:20:25 EDT 2014
slabinfo - version: 2.1
# name          <active_objs> <num_objs> <objsize> <objperslab> <pagesperslab> : tunables <limit> <batchcount> <sharedfactor>
: slabdata <active_slabs> <num_slabs> <sharedavail>
rpc_buffers      8      8    2048     2     1 : tunables    24    12     8 : slabdata     4     4     0
rpc_tasks        8     12     320    12     1 : tunables    54    27     8 : slabdata     1     1     0
rpc_inode_cache  8      8     832     4     1 : tunables    54    27     8 : slabdata     2     2     0
lser_descriptors 0      0     320    12     1 : tunables    54    27     8 : slabdata     0     0     0
ib_mad           0      0     448     8     1 : tunables    54    27     8 : slabdata     0     0     0
scsi_sense_cache 105    150    128    30     1 : tunables   120    60     8 : slabdata     5     5     0
scsi_cmd_cache   135    135    256    15     1 : tunables   120    60     8 : slabdata     9     9     0
iint_cache       10784  10784   120    32     1 : tunables   120    60     8 : slabdata   337   337     0
flow_cache       0      0     96     40     1 : tunables   120    60     8 : slabdata     0     0     0
dm_raid1_read_record 0      0    1064     7     2 : tunables    24    12     8 : slabdata     0     0     0
dm_snap_tracked_chunk 0      0     24    144     1 : tunables   120    60     8 : slabdata     0     0     0
dm_snap_pending_exception 0      0     112    34     1 : tunables   120    60     8 : slabdata     0     0     0
dm_snap_exception 0      0     32    112     1 : tunables   120    60     8 : slabdata     0     0     0
kcopyd_job       0      0     368    10     1 : tunables    54    27     8 : slabdata     0     0     0
dm_uevent        0      0    2608     3     2 : tunables    24    12     8 : slabdata     0     0     0
dm_rq_clone_bio_info 0      0     16    202     1 : tunables   120    60     8 : slabdata     0     0     0
dm_rq_target_io  0      0     376    10     1 : tunables    54    27     8 : slabdata     0     0     0
dm_target_io     514    576     24    144     1 : tunables   120    60     8 : slabdata     4     4     0
dm_io            514    552     40     92     1 : tunables   120    60     8 : slabdata     6     6     0
uhci_urb_priv    0      0     56     67     1 : tunables   120    60     8 : slabdata     0     0     0
sd_ext_cdb       2     112     32    112     1 : tunables   120    60     8 : slabdata     1     1     0
cfq_io_context   0      0    168    23     1 : tunables   120    60     8 : slabdata     0     0     0
cfq_queue        0      0    168    23     1 : tunables   120    60     8 : slabdata     0     0     0
bsg_cmd          0      0     312    12     1 : tunables    54    27     8 : slabdata     0     0     0
mqueue_inode_cache 1      4     896     4     1 : tunables    54    27     8 : slabdata     1     1     0
```

/proc/slabinfo - 内核 slab 分配器统计信息。

在 Linux 内核中频繁使用的 SLAB 分配对象(缓冲区头 buffer heads, 索引节点 inodes, 目录项 dentries 等)都会进行缓存存储。

OSWatcher: OSWTOP

```
oracle@pd1:~/oswbb/archive/oswtop
[oracle@pd1 oswtop]$ cat pd1.oracle.com_top_14.07.27.0200.dat | less
Linux OSWbb v7.3.0
zzz ***Sun Jul 27 02:20:25 EDT 2014
top - 02:20:26 up 28 min,  2 users,  load average: 0.00, 0.00, 0.00
Tasks: 133 total,  1 running, 132 sleeping,  0 stopped,  0 zombie
Cpu(s):  0.0%us,  0.5%sy,  0.0%ni, 99.5%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   4050940k total,  510764k used,  3540176k free,   25896k buffers
Swap:  6094840k total,    0k used,  6094840k free,   287620k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 2700 root        20   0  183m 4100 3360  S   1.0   0.1   0:00.55  gdm-rh-security
 3165 oracle     20   0 12760 1124  828  R   1.0   0.0   0:00.01  top
    1 root        20   0 10364  688  568  S   0.0   0.0   0:00.52  init
    2 root        20   0   0     0   0  S   0.0   0.0   0:00.00  kthreadd
    3 root        RT   0   0     0   0  S   0.0   0.0   0:00.03  migration/0
    4 root        20   0   0     0   0  S   0.0   0.0   0:00.02  ksoftirqd/0
    5 root        RT   0   0     0   0  S   0.0   0.0   0:00.00  watchdog/0
    6 root        RT   0   0     0   0  S   0.0   0.0   0:00.02  migration/1
    7 root        20   0   0     0   0  S   0.0   0.0   0:00.03  ksoftirqd/1
    8 root        RT   0   0     0   0  S   0.0   0.0   0:00.00  watchdog/1
    9 root        20   0   0     0   0  S   0.0   0.0   0:00.12  events/0
   10 root        20   0   0     0   0  S   0.0   0.0   0:00.10  events/1
   11 root        20   0   0     0   0  S   0.0   0.0   0:00.00  cpuset
   12 root        20   0   0     0   0  S   0.0   0.0   0:00.01  khelper
   13 root        20   0   0     0   0  S   0.0   0.0   0:00.00  netns
   14 root        20   0   0     0   0  S   0.0   0.0   0:00.00  async/mgr
   15 root        20   0   0     0   0  S   0.0   0.0   0:00.00  sync_supers
   16 root        20   0   0     0   0  S   0.0   0.0   0:00.00  bdi-default
   17 root        20   0   0     0   0  S   0.0   0.0   0:00.00  kintegrityd/0
   18 root        20   0   0     0   0  S   0.0   0.0   0:00.00  kintegrityd/1
   19 root        20   0   0     0   0  S   0.0   0.0   0:00.00  kblockd/0
   20 root        20   0   0     0   0  S   0.0   0.0   0:00.00  kblockd/1
   21 root        20   0   0     0   0  S   0.0   0.0   0:00.00  kacpid
```

top 命令能够实时显示运行系统中各个进程的资源占用状况。

OSWatcher: OSWTOP

top 命令输出中有哪些需要关注? -

负载均值(Load average) 是指在运行队列中进程的平均数量。如果出现大量进程等待则可能意味着当前系统 CPU 性能不足。

- 进程往往会消耗大量 CPU 资源。因此如果一个进程正占有着 CPU 资源, 我们就需要留心了。如果这个进程是一个 oracle 前台进程, 那么它很可能正运行一个高成本查询, 这就需要进行优化。通常 Oracle 后台进程也不应该长时间占有 CPU 大量资源。
- 是否负载均值很高。进程不应该在运行队列中被挂起以延长运行时间。
- 是否交换空间很低。这也意味着内存正处于低效运行。

OSWatcher :OSWVMSTAT

```
oracle@pd1:~/oswbb/archive/oswvmstat
[oracle@pd1 oswvmstat]$ cat pd1.oracle.com_vmstat_14.07.27.0200.dat | less
Linux OSWbb v7.3.0 pd1.oracle.com
SNAP_INTERVAL 30
CPU_COUNT 2
OSWBB_ARCHIVE_DEST /home/oracle/oswbb/archive
zzz ***Sun Jul 27 02:20:25 EDT 2014
procs -----memory----- --swap-- -----io----- --system-- -----cpu-----
r b  swpd  free  buff  cache  si  so  bi  bo  in  cs  us  sy  id  wa  st
3 0    0 3542668 25896 287540  0  0  90  7  58  47  1  1 98  1  0
0 0    0 3540176 25896 287608  0  0  0  0 255 240  2  5 94  0  0
0 0    0 3540780 25896 287620  0  0  0  0 161 131  0  1 99  0  0
zzz ***Sun Jul 27 02:20:55 EDT 2014
procs -----memory----- --swap-- -----io----- --system-- -----cpu-----
r b  swpd  free  buff  cache  si  so  bi  bo  in  cs  us  sy  id  wa  st
3 0    0 3542032 25904 287604  0  0  89  7  58  47  1  1 98  1  0
0 0    0 3540052 25904 287640  0  0  0  0 250 237  1  4 95  0  0
0 0    0 3540788 25904 287652  0  0  0  0 207 124  1  2 97  0  0
zzz ***Sun Jul 27 02:21:25 EDT 2014
procs -----memory----- --swap-- -----io----- --system-- -----cpu-----
r b  swpd  free  buff  cache  si  so  bi  bo  in  cs  us  sy  id  wa  st
4 0    0 3541744 25912 287640  0  0  87  7  58  47  1  1 98  1  0
0 0    0 3540052 25916 287680  0  0  0  0 261 238  2  4 95  0  0
0 0    0 3540796 25916 287692  0  0  0  0 163 123  1  1 99  0  0
```

vmstat 命令被用于查询虚拟内存统计信息。

vmstat 输出中包括虚拟内存 virtual memory 统计信息, 进程的内核统计信息以及 CPU 活动情况。

OSWatcher: OSWVMSTAT

- 在 vmstat 哪些需要我们关注 -
- 运行队列. (r)
- 一个大的运行队列中如果没有空闲的 CPU 资源那么意味着系统 CPU 资源不足。
- 在 CPU 使用中, 系统调用的时间不应超过 30%, 特别当空闲 CPU 时间近于 0%时。
- 扫描率(sr: scan rate)的大小能用于判断内存是否存在瓶颈。

扫描率 sr 是指每秒系统在找到足够的空闲页内存之前需扫描的页数。如果扫描率(sr)持续高于 200 页每秒那意味着存在内存不足问题。

- 如果存在堵塞的进程数量超过运行队列的进程数量, 则可能存在磁盘问题。

- 此类问题的解决的方法包括对应用的调优以使其能充分有效利用 cpu 资源, 通过增加 cpu 性能, 在系统中增加 cpu 等多种方法。

在解决节点重启问题中 OSWatcher 的作用 :

在分析 OCSSD 重启问题原因时很有用:

- 由于 CPU 资源不足造成的重启。在一些情况下会发生心跳丢失, 而这类事件的发生原因却仅是由于用户正在运行一个长时间高负载操作造成的。

当服务器处于超高负载状态, 其计划调度进程的可靠性会变得很差。这可能会造成其集群同步服务(CSS) 未能及时运行, 不能完成工作。一旦出现这种情况, 节点会被认为在集群工作中是不活动的并遭到驱逐。

- 由于私有网络问题造成的重启。查看 ocssd 日志-

```
WARNING: clssnmPollingThread: node <node> (1) at 75% heartbeat fatal, eviction in 13.950 seconds
```

以上问题可能由于私有网络问题或操作系统资源竞争/资源匮乏造成的。

在解决 OPROCD/OCLSOMON 重启问题原因时很有用:

-操作系统调度程序 (scheduler) 问题导致的重启.

服务器过量的负载造成了调度程序不能正常运行.

oprocd 日志 -

```
May 10 18:01:40.668 | INF | monitoring started with timeout(1000), margin(500)
```

```
May 10 18:23:02.490 | ERR | AlarmHandler: timeout(1739751316) exceeds I(1000000000)+margin(500000000)
```

以上日志说明 OPROCD 没有按时在 1.5(1000+500)秒内被调用。

它是在 1.73 秒时被运行。因此 OPROCD 计划调度问题将造成节点重启。

OSWatcher - “节点重启”的问题解决

案例研究 1:

此案例是针对 Linux 平台(9iRAC + 10g RAC 集群)和大多 Linux 平台常见问题的研究

- 由于“kswapd0”(内核交换守护进程 Kernel Swapper Deamon)造成的 CPU 使用高峰, 进而导致频繁的节点重启和实例驱逐事件的发生。

- 从 mpstat 命令输出中, 我们可以看到大量的 CPU 资源被与内核内存管理有关的系统 (SYS)进程占据。

```

zzz ***Tue Aug 28 21:53:26 GMT 2007
09:53:26 PM  CPU    %user   %nice  %system %iowait   %irq    %soft   %idle
intr/s
09:53:27 PM  all     13.49   0.00   86.37   0.00     0.00    0.14    0.00
1132.93
09:53:28 PM  all     14.12   0.00   85.88   0.00     0.00    0.00    0.00
1093.98
09:53:29 PM  all     13.59   0.00   86.28   0.00     0.00    0.14    0.00
1120.22
Average:      all     13.73   0.00   86.17   0.00     0.00    0.09    0.00
1115.75
    
```

在重启发生之前, vmstat 最后一次输出中显示 CPU 空闲时间为 0%-

```

zzz ***Tue Aug 28 21:52:46 GMT 2007
procs -----memory----- --swap-- -----io---- --system-- -----cpu----
r b swpd free buff cache si so bi bo in cs us sy id wa
8 0 812100 55252 2004 7410588 0 0 112 81 7 7 20 50 30 1
3 0 812092 50316 2004 7411204 4 0 32 5 1409 5624 10 90 0 1    << 0% idle, 90% sys
    << 数据收集中断 (节点重启) ~~ >>
    
```

```

zzz ***Tue Aug 28 22:07:06 GMT 2007
procs -----memory----- --swap-- -----io---- --system-- -----cpu----
r b swpd free buff cache si so bi bo in cs us sy id wa
8 4 830064 24608724 2456 7323912 0 0 112 81 8 7 4 2 93 1
3 1 829600 27076844 2936 4966952 0 0 9932 653 1991 6930 17 57 20 7
    
```

从 top 命令输出中, 观察到在重启时间前“kswapd0”进程正占用大量 CPU 时间

Top 命令输出显示如下-

```

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
177 root 16 0 0 0 0 0 S 97 0.0 3:38.35
    
```

kswapd0 (Kernel Swap daemon)

这意味着当时 kswapd0 正在 CPU 高峰时间忙碌运行。这种现象可能是由于节点未设置大页内存, 需要维护大量内存页造成的。

/proc/meminfo shows:

HugePages_Total: 0 → 表明此节点不存在大页内存

HugePages_Free: 0

Hugepagesize: 2048 kB

此机拥有 32G 内存，系统全局区最大达 12GB，但未设置大页内存，因为所有内存由 4k 内存页来管理。

由于需要管理大量的内存页，这种内核中的 spinning 造成了 CPU 运行高峰，当心跳 (heartbeat) ping 由于没有 CPU 资源而未能得到回复，这就进一步造成了实例的驱逐/节点重启事件的发生。

解决方案 -

- 当使用较大系统全局区时，请配置大页
- 修改/etc/sysctl.conf 内核参数文件并执行 sysctl 命令。
计算 $vm.nr_hugepages = sga_max_size / Hugepagesize = 12GB / 2048KB = 6144$ (在设置中可以设得比此计算值稍大)

```
# echo "vm.nr_hugepages=6146" >> /etc/sysctl.conf
# sysctl -p
# grep HugePages_Total /proc/meminfo.
```

HugePages_Total: 6146

HugePages_Free: 2

- 据观察 - 大页内存的设置，理论上，正常大小为 4k，而 64 为 Linux 系统中的大页内存可设置 2Mb 到 256Mb 大小（多数情况更大）。

相应的，这降低了对内存结构的持有数量。

由于大页内存能缓存更多信息，这也降低了内核对更大范围数据修改的操作量。

案例研究 2 -

- AIX 平台运行 10G R2 CRS +RDBMS 问题报告
- Oprocd 进程引起的节点重启

<从集群同步服务 CSS 和 oprocd.log 日志文件都没有找到关于重启的有用信息>

看重启时间段 vmstat 中的部分摘录 -

System configuration: lcpu=32 mem=31616MB

```
kthr memory page faults cpu time
```

```
-----
r b avm fre re pi po fr sr cy in sy cs us sy id wa hr mi se
1 24 8943723 44097 0 3 1654 1058 1697 0 627 4342 3342 0 1 54 45 14:46:05
```

PRM For Oracle 数据库灾难拯救工具下载: <http://www.parnassusdata.com/>

2 5 8945962 45813 0 15 152 119 139 0 393 45536 5107 3 3 90 4 14:46:35

2 0 8945371 44793 0 10 0 0 0 0 351 50764 5264 4 4 92 0 14:47:05

- 观察重启前最后时刻 avm (虚拟活动页 active virtual pages)在 14:47:05 记录值.
- 可以看到 $8945371 * 4096 / 1MB = 34942.8MB$ 大于当前系统配置内存=31616MB.
- 这意味着计算内存超出了实际物理内存.
- 随着虚拟内存管理(VMM)活动, 系统严重过载. 这导致了在 oprocd 进程调度的长时间延迟. 当 oprocd 再次运行起来后它就会重启当前节点进行 IO 隔离(IO fencing).

IBM 的建议及评论-

- 1) 在 vmstat 中监控 avm (active virtual pages) , 其不允许超过对实际物理内存的 95%.
- 2) 在 AIX 系统中, 没有任何调整方法可以解决当计算内存需求高于实际物理内存时的虚拟页管理问题. 只能要么增加物理内存, 要么设法降低对内存的需要.
- 3) 以下是对 Oracle RAC 安装的虚拟内存管理 VMM 参数设置:

minperm%=3, maxperm%=90 maxclient%=90, lru_file_repage=0.

OSWatcher - 用于进程间通信超时(IPC Send Timeout)问题的分析使用

- 什么是“进程间通信超时”?
当消息被发送后, 我们希望收到接收者的消息收到确认信息. 如果在特定时间间隔内没有收到 ACK 确认信息, 那么发送方将认为消息没能到达接收方.

这些消息的发送超时说明存在 lmon-lmon 或者 GES/GCS 通信问题-

对于 LMON-LMON 问题, 可以在 lmon 追踪文件中看到:

```
kjxgfipccb: msg 0x065C76CC, mbo 0x065C76C8, type 24, ack 0, ref 0, stat 3
```

```
kjxgfipccb: Send timed out, stat 3 inst 0, type 24, tkt (1224,0)
```

对于 GES/GCS 问题, 可以在 bdump/udump 追踪文件中看到:

```
kjctipccb: send timed out for msg 0x4e18cab00 to (1 4), inc 7 type 34 waited 294044059 usec  
kjctipccb: stat 3 dest_inc 6 sys_inc 7
```

- 在这种情况下, IPC 发送超时会导致无应答实例被驱逐, 并报 ORA-29740 reason 3 (通信失败) 错误.

案例研究 3 =

在 HP- Itanium 服务器通过 2 节点 RAC 集群运行 10.2.0.1 (CRS+RDBMS)中出现过的问题。

在一些报告出现'IPC Send timeout'报错后,RAC 集群中的一个实例被驱逐并报 ORA-29740 错误。

查看 Alert 日志 (在数据库实例 1 中) =

```
Thu Nov 27 11:32:05 2008
IPC Send timeout detected. Receiver ospid 4001974
Thu Nov 27 11:33:08 2008
Trace dumping is performing id=[cdmp_20081127113236]
Thu Nov 27 11:34:37 2008
Errors in file /oracle/app/product/admin/srs/bdump/srs1_lms1_4001974.trc:
Thu Nov 27 11:34:38 2008
Errors in file
/oracle/app/product/admin/srs/bdump/srs1_lmon_3977348.trc:
ORA-29740: evicted by member 1, group incarnation 32
Thu Nov 27 11:34:38 2008
LMON: terminating instance due to error 29740
```

查看 Alert 日志 (在数据库实例 2 中) =

```
Thu Nov 27 11:32:52 2008
IPC Send timeout detected.Sender: ospid 3227684
Receiver: inst 1 binc 1560281564 ospid 4001974
Thu Nov 27 11:32:55 2008
IPC Send timeout detected.Sender: ospid 3236078
Receiver: inst 1 binc 1560281564 ospid 4001974
  • 接收者的操作系统进程 Id 显示是实例 1 上的 LMS1
```

通过 OSWatcher 在数据库实例 1 中得到 ps 输出, 其显示 LMS1 在节点被驱逐前有 6 分钟时长处于'STOPPED/TRACED'状态。

< LMON 在调用 gdb debugger 转储 LMS 时使得 LMS 处于被追踪状态 >

可以看到进程状态被设置为'T':

```
240001 T oracle 4001974 1 0 60 2debd510 106608 08 Aug - 906:58 ora_lms1_srs1
```

- 从 LMON 追踪文件中我们看到 LMON 不停得在尝试转储 LMS 进程信息:

```
*** 2008-11-27 11:26:55.640
```

```
Dumping diagnostic information for ospid 4001974:
```

```
OS pid = 4001974
```

loadavg : 2.21 0.89 0.77

swap info: free_mem = 2150.32M rsv = 94.00M

alloc = 2751.73M avail = 24064.00M swap_free = 21312.27M

从 LMON 追踪文件中多行类似信息中我们了解到 DRM 被启用:

Begin DRM(1219)

sent syncr inc 30 lvl 12041 to 0 (30,0/31/0)

原因:

通用 Bug 5190596 'LMON dumps LMS0 too often during DRM leading to IPC send timeout'

解决方法:

检查平台是否已打上补丁 5190596.

此 bug 在版本 10.2.0.4 中已得到解决。

案例研究 4 =

AIX 服务器通过 2 节点 RAC 集群运行 9.2.0.6 版本数据库时出现的问题

以下是一个由于 LMON-LMON 通信发送超时导致的 9i 数据库 RAC 实例驱逐案例。

进程间发送延时(IPC Send timeout)发生于 Jan 11 02:04:29 =

- Alert 日志 (实例-1) 摘录 =

Fri Jan 11 02:04:29 2008

IPC Send timeout detected.Sender: ospid 2506878

Receiver: inst 2 binc -298848812 ospid 2596948

Sender: ospid 2506878 - LMON

Receiver: ospid 2596948 - LMON

- 以下是 OS Watcher vmstat 在节点 2 上那个时间段的输出

在隔离节点上获得的 vmstat 数据收集 =

zzz ***Fri Jan 11 02:01:51 CST 2008

System Configuration: lcpu=32 mem=128000MB

kthr	memory	page	faults	cpu													
r	b	avm	fre	re	pi	po	fr	sr	cy	in	sy	cs	us	sy	id	wa	
25	1	7532667	19073986	0	0	0	0	0	5	0	9328	88121	20430	32	10	47	11
58	0	7541201	19065392	0	0	0	0	0	0	0	11307	177425	10440	87	13	0	0

<< CPU 0 空闲资源


```
61 1 7552592 19053910 0 0 0 0 0 0 11122 206738 10970 85 15 0 0  
<< CPU 0 空闲资源
```

节点 2 当时非常忙 (CPU 空闲 idle 为 0)

除此之外我们还在 vmstat 输出中看到有很高的运行队列排程:

```
zzz ***Fri Jan 11 02:03:52 CST 2008
```

```
System Configuration: lcpu=32 mem=128000MB
```

```
  kthr      memory          page          faults          cpu  
-----  
  r  b   avm   fre re  pi  po  fr   sr  cy  in   sy  cs us sy id wa  
25  1 7733673 18878037  0  0  0  0   5  0 9328 88123 20429 32 10 47 11  
81  0 7737034 18874601  0  0  0  0   0  0 9081 209529 14509 87 13  0  0 <<<
```

这也糟糕

```
80  0 7736142 18875418  0  0  0  0   0  0 9765 156708 14997 91  9  0  0
```

- 通过观察-
- 可以推断由于 CPU 高峰, 没有 CPU 资源分配给 LMON, 导致 LMON 无法及时回复来自其他实例上的 LMON 心跳 ping, 错过了_cgs_send_timeout 时间. 最后发生了“IPC Send timeout”并造成此实例被其他实例驱逐。

OSWatcher – 对 RAC 性能问题的分析解决

案例研究 5 -- GC CR MULTIBLOCK REQUEST

Linux x86-64 RHAL 4.0 平台 4 节点 RAC 集群运行 10.2.0.2 (CRS+RDBMS) 问题报告

当执行一段简单查询时, 进程 hung 住并等待 “GC CR MULTIBLOCK REQUEST”

Oracle 块大小被设置为 16k 且参数 db_file_multiblock_read_count=16

10046 事件追踪文件显示 -

```
*** 2007-10-23 10:42:22.610
```

```
WAIT #1: nam='gc cr multi block request' ela= 1221289 file#=9 block#=40325 class#=1  
obj#=51434 tim=1118154397080504
```

UDP 套接字缓冲区(socket buffer) 参数值是按推荐设置-

```
net.core.rmem_default = 2621440
```

```
net.core.wmem_default = 2621440
```

```
net.core.rmem_max = 2621440
```

```
net.core.wmem_max = 2621440
```


OSWatcher NETSTAT 输出清楚显示 在查询运行时发生了“段丢失”(fragment dropped)

zzzTue Oct 23 10:42:18 BEIST 2007

Ip:

406541 fragments dropped after timeout

406541 packet reassembles failed

zzzTue Oct 23 10:44:18 BEIST 2007

Ip:

406542 fragments dropped after timeout

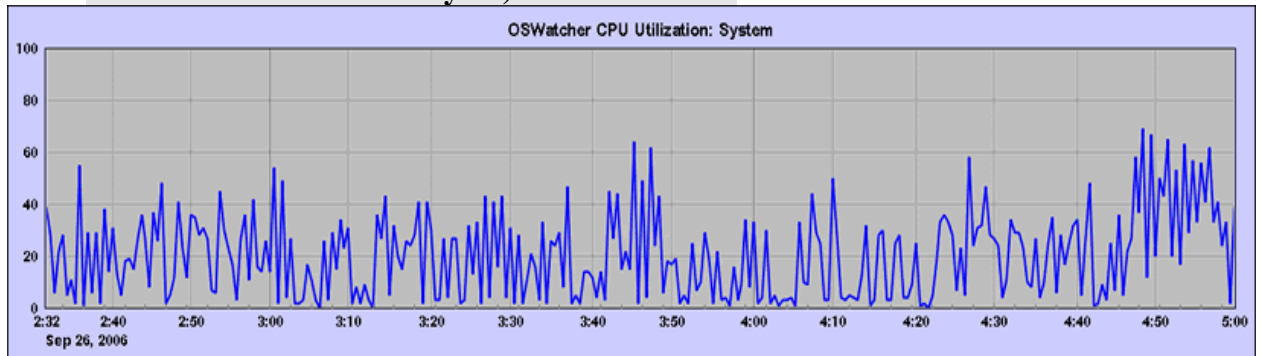
406541 packet reassembles failed

这表明问题出在网络层.

经观察 -

最终问题被确认是当 Oracle 块大小被设置为 16k 并 db_file_multiblock_read_count 为 16 后, CISCO 交换机固件未能处理其引发的网络拥塞造成的。当客户升级交换机固件至最新版本后, 这个问题也就被修正解决, RAC 性能也提升了上去。

OSWatcher 的图形化输出 (OSWg, 后更名为 oswbba: OSWatcher Analyzer)



- OS Watcher Graph (OSWg) 是一个图形化数据解析工具. 它和 OSW v2.0.0 或其更高版本中一起绑定发布。
- OSWg 提供对所有支持的 Unix 平台进行 vmstat 数据文件解析, 并仅对 Solaris, AIX 和 Linux 平台进行 iostat 数据文件解析。
- OSWg 工具由 Java 开发, 因此使用时需要 java 1.4.2 或以上版本虚拟机的支持。OSWg 能运行在 Unix X Windows 或者 Windows XP 平台上。

IPD/OS 工具 (后名为 CHM: Cluster Health Monitor 集群健康监视器)- OSWatcher 工具扩展

- IPD - 系统及时问题检测工具(Instantaneous Problem Detection/ OS Tool)
- 其能够定时自动地收集操作系统性能数据.
- 可以做线下和线上数据分析
- 其有助于 Oracle 集群节点驱逐/hang 问题的原因查找
- 总是以高频率(小于 3 秒)的实时优先级运行并进行数据抽样
- 检查的数据可用于对单实例 RAC 性能调优.
- 此工具计划面向全平台使用, 但现在仅在 Linux 平台可用。具体下载可“otn.oracle.com/rac”下查找。
- IPD(CHM)可以提供以上所有功能并将最终替换 OSWatcher.

OSWatcher - 常见问题:

- 如何判断 OSWatcher 是否正在系统上运行?

```
$ ps -ef | grep OSW
```

我们应该可以看见 2 个 shell 进程- OSWatcher.sh and OSWatcherFM.sh

- OSWatcher 是否能在系统 Reboot 后自动重启?

不能 - OSWatcher 在系统 Reboot 后需要手工重启.

如果需要其自动化, 我们可以使用一个简单的 shell 脚本.

在 Linux 平台上, 有一个“osw-service” RPM 包可供下载使用

请参考文档 Note-580513.1 “How To Start OSWatcher Every System Boot”以获取此 RPM 包。

- 在 OSWatcher 中哪些数据对分析有帮助?

PRM For Oracle 数据库灾难拯救工具下载: <http://www.parnassusdata.com/>

通常, 最后小时的数据对于导致节点重启/实例驱逐的原因分析非常有帮助。

- 使用 OSW 需要多大空间?

OSW 工具 (shell 脚本) 需要约 1.5 Mb 大小.

而 OSW 归档数据的空间需求则取决于一些其他因素, 如:

- 当前硬件设置(处理器数量, 磁盘数量)
- 归档数据需要保留的时间.

使用默认 OSWatcher 数据收集(48 小时数据归档)在一个 Linux 平台的双核 2 个节点 11g RAC 集群上做一个测试, 其归档空间消耗约为每节点 300MB 大小。

- 运行 OSWatcher 是否需要使用操作系统账户?

-OSW 可以在 Unix 平台上的任何操作系统账户下运行 - 但 OSW 将需要相关系统工具权限: top,vmstat,iostat,mpstat,ps, netstat 和 traceroute.

需要给使用 OSW 的用户设置这些工具的执行权限。

-在 Windows 平台, OSWatcher 则必须以 Administrator 账户运行。

可参考阅读以下文档:

Note 301137.1 Title: OS Watcher User Guide

Note 265769.1 Title:Troubleshooting CRS Reboots

Note 434351 Title: Linux Kernel : The Slab Allocator

Note 361323.1 HugePages on Linux

Note 563566.1 Title: gc lost block diagnostics

Note 461053.1 Title:OS Watcher Graph (OSWg) User Guide.

Note 736752.1 Title: Introducing Oracle Instantaneous Problem detection -os tool (IPD/OS)

Note 4295291 Title: Performance counters Windows

Q & A

Find More

技术资源 : <http://www.parnassusdata.com/resources/>

技术支持: service@parnassusdata.com

销售: sales@parnassusdata.com

下载 PRM FOR ORACLE 灾难恢复软件: <http://www.parnassusdata.com/>

联系诗檀软件: <http://www.parnassusdata.com/zh-hans/contact>

Conclusion

ParnassusData

ParnassusData Corporation , Shanghai , GaoPing Road No. 733 . China

Phone: (+86) 400-690-3643

ParnassusData.com

Facebook: <http://www.facebook.com/parnassusData>

Twitter: <http://twitter.com/ParnassusData>

Weibo: <http://weibo.com/parnassusdata>

Copyright © 2013, ParnassusData and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

PRM For Oracle 数据库灾难拯救工具下载: <http://www.parnassusdata.com/>

诗檀软件 专业 Oracle 数据库服务 www.parnassusdata.com
Oracle 紧急服务国内热线电话: 400-690-3643

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd. 0410

Copyright © 2014 ParnassusData Corporation. All Rights Reserved.