

监控优化篇-操作系统监控

nmon 使用与介绍

(FOR AIX&Linux)

作者:DKM

1	描述.....	3
2	Nmon介绍.....	3
3	Nmon数据采集.....	7
3.1	数据采集.....	7
3.2	生成图形化结果.....	8

1 描述

监控，在检查系统问题或优化系统性能工作上是一个不可缺少的部分。

做好监控工作，会使得我们在优化和排查故障上更好地定位问题，定位了问题那剩下来的就是采用的方法了。

通过操作系统监控工具监视操作系统资源的使用情况，间接地反映了各服务器程序的运行情况。根据运行结果分析可以帮助我们快速定位系统问题范围或者性能瓶颈点。因此操作系统的监控是不容忽视的。

2 Nmon介绍

这里我们介绍下如何使用 nmon 来监视操作系统资源使用情况。

nmon 是一种在 aix 与各种 Linux 操作系统上广泛使用的监控与分析工具，相对于系统资源上的一些监控工具来说，nmon 所记录的信息是比较全面的，它能在系统运行过程中实时地捕捉系统资源的使用情况，并且能输出结果到文件中，然后通过 nmon_analyzer 工具产生数据文件，与图形化结果。通过图形化界面分析，得出系统在一段时间内资源占用的变化趋势，有了这个分析结果就可以帮助我们更好定位问题。

nmon 所记录的数据包含以下一些方面（也是我们在寻找问题过程中所关注的资源点）：

- cpu 占用率
- 内存使用情况
- 磁盘 I/O 速度、传输和读写比率
- 文件系统的使用率
- 网络 I/O 速度、传输和读写比率、错误统计率与传输包的大小
- 消耗资源最多的进程
- 计算机详细信息和资源
- 页面空间和页面 I/O 速度
- 用户自定义的磁盘组
- 网络文件系统

另外在 AIX 操作系统上，nmon 还能监控到其他的一些信息，如异步 I/O 等。

如何获取nmon与nmon_analyzer呢？我们可以在IBM的官方网站上免费下载获取，或者到<http://www.webspherechina.net>论坛，『WebSphere资源分享』区下载。

URL: nmon操作手册与nmon工具下载

<http://www.webspherechina.net/club/viewthread.php?tid=7650>

使用 nmon 是比较简单的事情，只要把 nmon 脚本上传到服务中，直接运行即可，执行命令如下：

`#!/nmon` 或者 `#!/tmp/nmon/nmon`

以下是 aix5.3 下使用 nmon 的一些截图：

```
-----  
N   N M   M   0000 N   N   For online help type: h  
NN  N MM  MM  0   0 NN  N   For command line option help:  
N N  N M MM  M  0   0 N N  N   quick-hint nmon -?  
N N  N M   M  0   0 N N  N   full-details nmon -h  
N   NN M   M  0   0 N   NN   To start nmon the same way every time?  
N   N M   M   0000 N   N   set NMON ksh variable, for example:  
                               export NMON=cmt  
-----  
Version v11e for AIX53  
      4 - CPUs currently  
      4 - CPUs configured  
    2097 - MHz CPU clock rate  
PowerPC_POWER5 - Processor  
    64 bit - Hardware  
    64 bit - Kernel  
   Dynamic - Logical Partition  
 5.3.0.61 ML06 - AIX Kernel Version  
gdjtglttest1 - Hostname
```

分别输入 c、t、n、m，可以了解系统 cpu，内存，消耗资源最高的线程的使用情况。

```

nmmon C=many-CPU's Host=gdjtglttest1 Refresh=2 secs 21:57.03
CPU-Utilisation-Small-View
0-----25-----50-----75-----100
CPU User% Sys% Wait% Idle%|
0 0.0 0.0 0.0 100.0|
1 0.0 0.0 0.0 100.0|
2 5.3 0.5 0.0 94.3|UU|
3 0.0 0.0 0.0 100.0|
Physical Averages
All 0.9 1.0 0.0 98.1|>|
Memory
Physical PageSpace | pages/sec In Out | FileSystemCache
% Used 99.7% 2.3% | to Paging Space 0.0 0.0 | (numperm) 12.0%
% Free 0.3% 97.7% | to File System 0.5 0.0 | Process 71.2%
MB Used 3908.7MB 487.4MB | Page Scans 0.0 | System 16.5%
MB Free 11.3MB 20504.6MB | Page Cycles 0.0 | Free 0.3%
Total(MB) 3920.0MB 20992.0MB | Page Steals 0.0 | Total 100.0%
| Page Faults 1.5 | numclient 12.0%
Min/Maxperm 750MB( 19%) 3000MB( 77%) <--% of RAM | maxclient 76.5%
Min/MaxFree 960 1088 Total Virtual 24.3GB | User 79.0%
Min/Maxpagehead 2 8 Accessed Virtual 3.7GB 15.2% Pinned 18.9%
Network
I/F Name Recv=KB/s Trans=KB/s packin packout insize outsize Peak->Recv Trans
en0 0.1 0.5 2.5 0.5 46.0 946.0 0.4 1.0
lo0 2.8 2.8 17.0 17.0 170.8 170.8 2.8 2.8
Total 0.0 0.0 in Mbytes/second
I/F Name MTU ierror oerror collision Mbits/s Description
en0 1500 0 3 0 1024 Standard Ethernet Network Interface
lo0 16896 0 0 0 0 Loopback Network Interface
Top-Processes-(69) Mode=3 [1=Basic 2=CPU 3=Perf 4=Size 5=I/O 6=Cmnds]
PID %CPU Size Res Res Res Char RAM Paging Command
Used KB Set Text Data I/O Use io other repage
241692 1.1 361180 320976 68 320908 0 8% 0 0 0 java
270354 0.5 204780 120148 68 120080 62959 3% 0 0 0 java
336112 0.4 110136 69448 68 69380 62959 2% 0 0 0 java
86132 0.3 362576 313720 68 313652 0 8% 0 0 0 java
217180 0.3 6948 7004 200 6804 445 0% 0 0 0 nmmon
135270 0.2 1900 412 4 408 0 0% 0 0 0 dtgreet
0 0.0 384 384 0 384 0 0% 0 0 0 Swapper
69666 0.0 960 960 0 960 0 0% 0 0 0 gil = TCP/IP
24588 0.0 512 512 0 512 0 0% 0 0 0 psmd
127092 0.0 7612 508 132 376 0 0% 0 0 0 X
311464 0.0 416 336 36 300 0 0% 0 0 0 cron
290974 0.0 1216 1216 0 1216 0 0% 0 0 0 rpc.lockd
245898 0.0 276 92 24 68 0 0% 0 0 0 qdaemon

```

```

nmon-----5=Top-by-I/O-use-----Host=gdjtgltst1-----Refresh=2 secs-----21:57.29
CPU-Utilisation-Small-View
0-----25-----50-----75-----100
CPU User% Sys% Wait% Idle%|
0 2.0 0.0 0.0 98.0|
1 0.0 0.0 0.0 100.0|
2 0.0 0.0 0.0 100.0|
3 0.0 0.0 0.0 100.0|
-----|-----|-----|-----|
Physical Averages |
All 1.1 1.0 0.0 97.9|
-----|-----|-----|-----|
Memory
Physical PageSpace | pages/sec In Out | FileSystemCache
% Used 99.7% 2.3% | to Paging Space 0.5 0.0 | (number) 12.0%
% Free 0.3% 97.7% | to File System 0.0 0.0 | Process 71.2%
MB Used 3908.7MB 487.4MB | Page Scans 0.0 | System 16.5%
MB Free 11.3MB 20504.6MB | Page Cycles 0.0 | Free 0.3%
Total (MB) 3920.0MB 20992.0MB | Page Steals 0.0 | Total 100.0%
-----|-----|-----|-----|
Min/Maxperm 750MB( 19%) 3038( 7%) <--% of RAM | numclient 12.0%
Min/Maxfree 960 1088 Total Virtual 24.3GB | maxclient 76.5%
Min/Maxpagehead 2 8 Accessed Virtual 3.7GB 15.2% Pinned 18.9%
-----|-----|-----|-----|
Network
I/F Name Recv=KB/s Trans=KB/s packin packout insize outside Peak->Recv Trans
en0 0.5 0.3 4.5 0.5 103.3 699.0 0.5 1.0
lo0 1.9 1.9 8.5 8.5 230.6 230.6 3.5 3.5
Total 0.0 0.0 in Mbytes/second
-----|-----|-----|-----|
I/F Name MTU ierror oerror collision Mbits/s Description
en0 1500 0 3 0 1024 Standard Ethernet Network Interface
lo0 16896 0 0 0 0 Loopback Network Interface
-----|-----|-----|-----|
Top-Processes-(69) Mode=3 [1=Basic 2-CPU 3=Perf 4=Size 5=I/O 6=Cmnds]
PID %CPU Size Res Res Res Char RAM Paging Command
Used KB Set Text Data I/O Use io other repage
241692 1.1 361180 320976 68 320908 0 8% 0 0 0 java
86132 1.0 362576 313720 68 313652 0 8% 0 0 0 java
336112 0.9 110136 69448 68 69380 47451 2% 0 0 0 java
270354 0.4 204780 120148 68 120080 41972 3% 0 0 0 java
217180 0.3 6948 7004 200 6804 322 0% 0 0 0 nmon
135270 0.2 1900 412 4 408 0 0% 0 0 0 dtgreet
0 0.0 384 384 0 384 0 0% 0 0 0 Swapper
69666 0.0 960 960 0 960 0 0% 0 0 0 gil = TCP/IP
49176 0.0 448 448 0 448 0 0% 0 0 0 xmgc
127092 0.0 7612 508 132 376 0 0% 0 0 0 X
290974 0.0 1216 1216 0 1216 0 0% 0 0 0 rpc.lockd
143478 0.0 1072 108 20 88 0 0% 0 0 0 aixmibd
307370 0.0 2100 280 84 196 0 0% 0 0 0 rncd

```

输入 h, 打开 nmon 帮助页面

```

HELP-----key=statistics which toggle on/off-----
h = This help information - use h to remove this screen
r = Resources CPU/cache/AIX etc. p = Partitions (LPAR) Stats
t = Top Process Stats 1=basic 2=CPU-Use 3=CPU 4=Size 5=Disk-I/O
u = shows command arguments (hit u again to refresh)
c = CPU by processor #=PURR for p5 MicroLPAR
C = CPU can show up to 128 CPUs l = longer term CPU averages
m = Memory and Paging stats k = Kernel Internal stats
n = Network stats j = JFS Usage Stats
d = Disk I/O Graphs D=Stats o = Disks %Busy Map
a = Adapter I/O (not on AIX 5.1) e = ESS vpath Stats
A = Async I/O Servers g = Disk Groups (see cmdline -g)
v = VerboseChecks-OK/Warn/Danger N = NFS stats
b = black & white mode w = view AIX wait processes
U = WLM Classes (Top Section) W = WLM Section (S=SubClasses)
V = Volume Group disk Stats P = Paging Space Stats
--- controls ---
+ and - = double or half the screen refresh time
q = quit space = refresh screen now
. = Minimum Mode=display only busy disks and processes
0 = reset peak counts to zero (peak = ">")

nmon version v11e - written by Nigel Griffiths, nag@uk.ibm.com

```

3 nmon数据采集

3.1 数据采集

为了实时监控系统在一段时间内的使用情况并将结果记录下来,我们可以通过运行以下命令实现:

```
#!/ nmon -f -t -r test -s 30 -c 180
```

- test: 这次监控记录的标题 () 与生产的文件名称
- -s 30: 每 30 秒进行一次数据采集
- -c 180:一共采集 180 次

以上就是 nmon 记录了系统在 1.5 小时内的使用情况

输入命令后,将自动在当前目录生成一个 hostname_timeSeries.nmon 的文件(hostname 为当前见识的服务器的主机名)如:

- hostname 为 test1
- 生产的文件为: test1_090320_2213.nmon

通过以下命令将 nmon 结果转换为 csv 文件:

```
# sort -A test1_090320_2213.nmon > test1_090320_2213.csv
```

即可在当前目录生产 test1_090320_2213.csv 文件。

我们将 test1_090320_2213.csv 文件下载到本地,通过 nmon_analyser 工具转换为 excel 文件,此时打开 excel 文件,我们即可通过图形化方式查看到系统的运行趋势图了。

nmon 参数介绍:

```
-f          spreadsheet output format [note: default -s300 -c288]
           Output file is <hostname>_YYYYMMDD_HHMM.nmon

           -F <filename>  same as -f but user supplied filename
-c <number>  number of snapshots
-d          requests disk service and wait times (DISKSERV and DISKWAIT)
-i <percent> Ignore processes using less than this amount of CPU when
           generating TOP section - useful for reducing data volumes
-g <filename> file containing disk group definitions
-l <dpl>     number of hdisks per sheet - defaults to 150, maximum 250.  See
notes

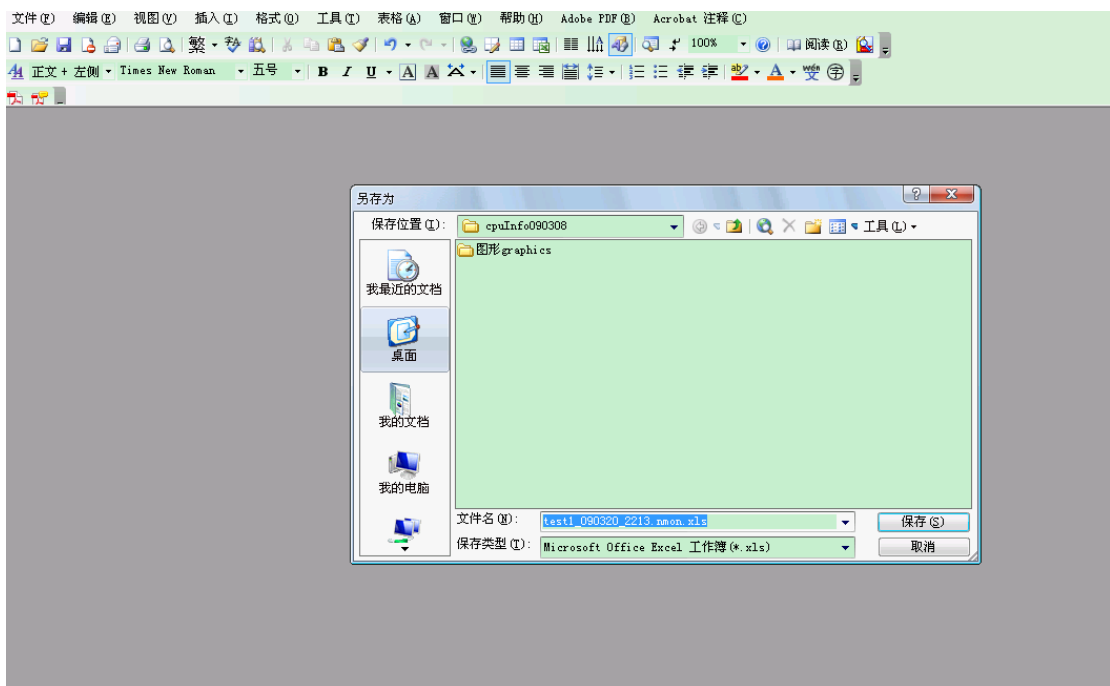
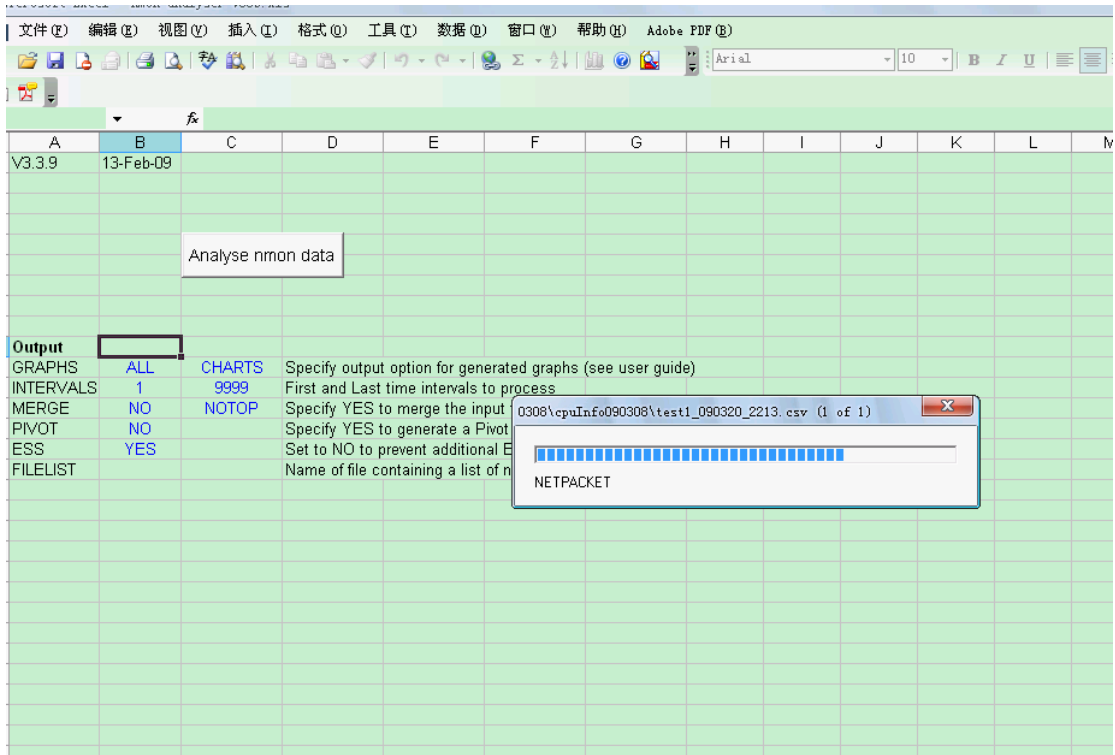
-m <dir>     NMON changes to this directory before saving the file
-r <runname> goes into spreadsheet file [default hostname]
-s <seconds> interval between snap shots
-x          capacity planning (15 mins for 1 day = -fdt -s900 -c96)
-t          include top processes in the output
-T          as -t plus saves command line arguments in UARG section
-A          include data for async I/O (PROCAIO) sections
-D          prevents DISK sections being produced (useful when Disk Groups
           are being used because there are too many hdisks to process)
```

- E stops ESS sections being produced (necessary when Disk Groups are being used because there are too many vpaths to process)
- J prevents JFS sections being produced (prevents Excel errors when you have more than 255 filesystems)
- L includes LARGEPAGE section
- N include NFS sections
- S include WLM sections with subclasses
- W include WLM sections without subclasses
- Y include SUMMARY section (very efficient alternative to `-t` if PID level data is not required)

3.2 生成图形化结果

如何使用 `nmon analyser.xls` 工具生成 excel 文件呢？

- 1、 下载 `nmon analyser.xls` 工具
- 2、 打开 `nmon analyser.xls` 工具
- 3、 调整 excel 宏安全性：工具-宏-安全性
(修改安全级别与可靠发行商)
安全级别:低
(勾上) 信任所有安装的加载项和模板
(勾上) 信任对于“Visual Basic 项目” 的访问
- 4、 修改完后，确定-关闭 `nmon analyser.xls`，重新打开
- 5、 点击 `Analyse nmon data` 按钮，加载之前下载 `test1_090320_2213.csv` 文件
以下是执行过程截图：



图形化结果:

