Troubleshoot StarOs CPU/Memory/Files Usage Monitoring

Contents

Introduction

This document describes fundimentals of CPU/Memory/Files usage on StarOS systems and how to troubleshoot when problem occures.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- StarOs

Components Used

This document is not restricted to specific software and hardware versions.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

Resource management subsystem assigns a set of resource limits for each task in the system. It monitors each task's resource usage to ensure it is staying within the limit. If a task has exceeded its limits it notifies to operators via Syslog or Simple Network Management Protocol (SNMP) traps. This document explains

how it works and what logs you must collect for further troubleshooting.

You can check the basic info in the output of **show task resources** command line interface (CLI).

The allocated resource limits can not be changed by user.

The allocated resource limits are different based on StarOS version.

good/warn/over

```
[local]asr5500-2# show task resources
Sunday January 12 01:03:42 JST 2014
                    task   cputime        memory      files       sessions
 cpu facility       inst used allc   used    alloc used allc  used   allc S status
------------------------ ---------- -------------- ---------- -------------- ------
 2/0 sitmain          20 0.1%  15% 10.54M  16.00M   13 1000    --     -- -   good
 2/0 sitparent        20 0.0%  20%  7.92M  14.00M   10  500    --     -- -   good
 2/0 hatcpu           20 0.1%  10%  8.16M  15.00M   11  500    --     -- -   good
 2/0 afmgr            20 0.1%  10% 11.40M  20.00M   13  500    --     -- -   good
 2/0 rmmgr            20 0.7%  15% 11.12M  23.00M  212  500    --     -- -   good
 2/0 hwmgr            20 0.1%  15%  8.06M  15.00M   12  500    --     -- -   good
 2/0 dhmgr            20 0.1%  15% 11.16M  26.00M   14 6000    --     -- -   good
 2/0 connproxy        20 0.1%  50%  9.09M  26.00M   11 1000    --     -- -   good
 2/0 dcardmgr         20 0.2%  60% 40.00M  600.0M   12  500    --     -- -   good
 2/0 npumgr           20 0.6% 100% 475.0M   2.27G   21 1000    --     -- -   good
 2/0 npusim           21 0.1%  33% 12.45M  60.00M   12  500    --     -- -   good
 2/0 sft             200 0.1%  50% 11.89M  30.00M   10  500    --     -- -   good
 2/0 vpnmgr            2 0.1% 100% 20.60M  37.00M   20 2000    --     -- -   good
 2/0 zebos             2 0.1%  50% 10.07M  25.00M   14 1000    --     -- -   good
 2/0 vpnmgr            3 0.1% 100% 20.73M  37.00M   20 2000    --     -- -   good
 2/0 zebos             3 0.1%  50% 10.07M  25.00M   15 1000    --     -- -   good
 2/0 vpnmgr            4 0.1% 100% 32.31M  73.74M   20 2000    --     -- -   good
 2/0 zebos             4 0.1%  50% 10.07M  30.00M   15 1000    --     -- -   good
 2/0 vpnmgr            5 0.1% 100% 21.27M  37.00M   30 2000    --     -- -   good
 2/0 zebos             5 0.1%  50% 10.20M  25.00M   15 1000    --     -- -   good
 2/0 aaaproxy          1 0.1% 100% 17.99M  160.0M   11 1000    --     -- -   good
 2/0 gtpumgr           1 0.3%  90% 21.52M   2.00G  160 1000    --     -- -   good
```

This is example of SNMP that occures when problem is present on the system:

<#root>

Mon Aug 26 11:32:19 2013 Internal trap notification 1221 (

MemoryOver

) facility sessmgr instance 16 card 1 cpu 0 allocated 204800 used 220392
Mon Aug 26 11:32:29 2013 Internal trap notification 1222 (

MemoryOverClear

) facility sessmgr instance 16 card 1 cpu 0 allocated 1249280 used 219608

Fri Dec 20 13:52:20 2013 Internal trap notification 1217 (

MemoryWarn

) facility npudrv instance 401 card 5 cpu 0 allocated 112640 used 119588
Fri Dec 20 14:07:26 2013 Internal trap notification 1218 (

MemoryWarnClear

) facility cli instance 5011763 card 5 cpu 0 allocated 56320 used 46856

Wed Dec 25 12:24:16 2013 Internal trap notification 1220 (

CPUOverClear

```
) facility cli instance 5010294 card 5 cpu 0 allocated 600 used 272
Wed Dec 25 12:24:16 2013 Internal trap notification 1216 (
```

CPUWarnClear

```
) facility cli instance 5010294 card 5 cpu 0 allocated 600 used 272

Wed Dec 25 17:04:56 2013 Internal trap notification 1215 (
```

CPUWarn

```
) facility cli instance 5010317 card 5 cpu 0 allocated 600 used 595
Wed Dec 25 17:05:36 2013 Internal trap notification 1216 (
```

CPUWarnClear

```
) facility cli instance 5010317 card 5 cpu 0 allocated 600 used 220
```

CPU Usage Monitoring

CPUWarn SNMP trap is generated when proclet's cpu usage reaches 90% of its allocated.

Once **CPUWarn** is generated, **CPUOver** is generated when proclet's cpu usage reaches more 50% of its allocated from the warned value.

If proclet's cpu usage reaches its allocated usage before **CPUWarn** is generated, then **CPUOver** is generated.

CPUWarn/Over is cleared when usage goes back to 50% of allocated.

Example:

If system allocation for facility is 60, when the value reach 54, system generates SNMP trap (**CPUWarn**).

Since system allocation for facility is 60, when proclet's cpu usage reaches more then 50% of its allocated from the warned value, in this scenario when system reach value 84 (54+30) system generates SNMP trap (**CPUOver**).

Memory Usage Monitoring

MemoryWarn is generated when proclet's memory usage reaches its allocation.

MemoryOver is generated when proclet's memory usage reaches more than its allocated + 15MB, or double of its allocation.

MemoryWarn/**MemoryOver** are cleared when usage goes back to 95% of its allocation.

Example:

If system allocation for facility is 60MB, then for any value larger than 60MB, system generates SNMP trap **MemoryWarn**.

Since system allocation for facility is 60MB, when task memory utilisation reaches 75MB, system generates SNMP trap **MemoryOver**.

Files Usage Monitoring

Files indicates the number of open files, or the file descriptor the process is using.

There is no SNMP trap implemented for the files usage, but logging message is generated for over/clear state.

The over log is generated when proclet's file usage reaches more than its allocated + 10% of it's allocated.

The clear log is generated when proclet's file usage goes back to 90% of its allocated.

<#root>

```
2013-May-28+14:16:18.746 [
```

resmgr 14517 warning

```
] [8/0/4440 <rmmgr:80>
_resource_cpu.c:3558] [software internal system syslog]
```

The task cli-8031369 is over its
open files limit. Allocated 2000, Using 2499

Status in show task resource

Status field in the output of **show task resources** CLI has different criteria.

In the below picture WARN is warn and ALARM is over status.

```
/*
 * WHAT      WARN > than                ALARM > than
 * --------  -------------------------  -------------------------
 * cputime   limit*0.99                 MAX(limit*1.2, limit+5% )
 * mem       limit*0.99                 MAX(limit*1.2, limit+5MB)
 * fds       limit*0.99                 MAX(limit*1.2, limit+50 )
 */
```

Troubleshooting

For CPU usage

When system starts to generate SNMP traps related to CPU, collect the following information during the active problem:

show task resources

Check if any proclet goes warn/over state

show task resource max

Check max usage rather than current usage

show snmp trap history

Check if there is any CPUWarn/Over event

Note: This is hidden/test command, Refer to the Documentation how to enable and enter Test mode in StarOs.

This command is not service impacting and can be run in production.

---

show profile card <card number> cpu <cpu number> depth <value>

This is so-called Background profiler.

Background Profiler is always running, even in production, with a fixed sampling period of 1s.

We can know which PC consumes CPU resource, per card/cpu/facility/instance, etc.

Recommend to specify depth rather using default value 1.(e.g. 4)

For Memory usage

When system starts to generate SNMP traps related to Memory, collect the following information during the active problem:

show task resources

Check if any proclet goes warn/over state

show task resource max

Check max usage rather than current usage

show snmp trap history

Check if there is any MemoryWarn/Over event

show logs

Check if there is any warning/error reported by resmgr.

Note: This is hidden/test command, Refer to the Documentation how to enable and enter Test mode in StarOs.

This command is not service impacting and can be run in production.

---

show messenger proclet facility <name> instance <x> heap

Check heap usage of the proclet

Note: This is hidden/test command, Refer to the Documentation how to enable and enter Test mode in StarOs.

This command is not service impacting and can be run in production.

show messenger proclet facility <name> instance <x> system heap

Check system heap information for containing process

Tip: Take multiple outputs of cpu related commands every 10 minutes and 4 outputs before raising Service Request towards TAC.

For Files usage

The actual file limit at OS level is set higher then the files usages limit in StarOs.

Example for task Diameter Proxy (diaproxy), OS level limit is 8192 the process can consume up to 8192 while the files limit is set as 1000 at StarOS.

<#root>

asr5500:card3-cpu0#

ps -ef | grep diam

root 5934 4555 0 Jul02 ? 00:07:52 diamproxy --readypipe 8 --limit_mode 8 --card_number 3 --cpu_number 0

asr5500:card3-cpu0#

cat /proc/5934/limits | grep open

Max open files 8192

8192

files

[local]asr5500-2#

show task resources facility diamproxy all

Friday July 11 10:05:54 JST 2014
task cputime memory files sessions
cpu facility inst used allc used alloc used allc used allc S status
---------- --------- ------------- --------- ------------- ------
3/0 diamproxy 2 0.3% 90% 22.83M 250.0M 216

1000

-- -- - good
8/0 diamproxy 1 0.4% 90% 22.71M 250.0M 69 1000 -- -- - good

There is a CPU level limit as well, please check it also and you would be fine as long as you have enough available.

<#root>

```
[local]ASR5500# show cpu info card 1 cpu 0
Card 1, CPU 0:
  Status                : Active, Kernel Running, Tasks Running
  Load Average          : 0.26, 0.39, 0.44 (1.78 max)
  Total Memory          : 32768M (16384M node-0, 16384M node-1)
  Kernel Uptime         : 3D 22H 11M
  Last Reading:
    CPU Usage All       : 0.1% user, 0.3% sys, 0.0% io, 0.0% irq, 99.6% idle
      Node 0            : 0.1% user, 0.3% sys, 0.0% io, 0.0% irq, 99.5% idle
      Node 1            : 0.1% user, 0.2% sys, 0.0% io, 0.0% irq, 99.7% idle
    Processes / Tasks   : 185 processes / 29 tasks
    Network             : 0.326 kpps rx, 0.912 mbps rx, 0.208 kpps tx, 3.485 mbps tx

    File Usage          : 1792 open files, 3279141 available

    Memory Usage        : 1619M 4.9% used (1209M 7.4% node-0, 409M 2.5% node-1)
```

When available becomes less than 256, this warning message is generated:

```
event 14516
user_resource_cpu_cpu_low_files(uint32 card, uint32 cpu, uint32 used, uint32 remain)
"The CPU %d/%d is running low on available open files. (%u used, %u remain)"
warning
software internal system critical-info
```