

# 排除StarOs CPU/内存/文件使用监控故障

## 目录

---

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[CPU使用率监控](#)

[内存使用率监控](#)

[文件使用情况监控](#)

[显示任务资源中的状态](#)

[故障排除](#)

[对于CPU使用率](#)

[对于内存使用](#)

[对于文件使用](#)

---

## 简介

本文档介绍StarOS系统上CPU/内存/文件使用情况的基本信息，以及发生问题时如何排除故障。

## 先决条件

### 要求

Cisco 建议您了解以下主题：

- StarOs

### 使用的组件

本文档不限于特定的软件和硬件版本。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

## 背景信息

[资源管理](#)子系统为系统中的每个任务分配一组资源限制。它会监控每个任务的资源使用情况，以确保资源使用量保持在限制内。如果任务超出其限制，它会通过系统日志或简单网络管理协议 (SNMP) 陷阱通知操作员。本文档介绍其工作原理以及您必须收集哪些日志才能进一步进行故障排除。

您可以在show task resources命令行界面(CLI)的输出中检查基本信息。

用户无法更改已分配的资源限制。

分配的资源限制因StarOS版本而异。

```
[local]asr5500-2# show task resources
Sunday January 12 01:03:42 JST 2014
```

cpu	facility	task inst	cputime		memory		files		sessions			status
			used	allc	used	alloc	used	allc	used	allc	S	
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	rmngr	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

以下是系统出现问题时发生的SNMP示例：

<#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使用率监控

CPUWarn SNMP陷阱在proclet的cpu使用率达到其分配的90%时生成。

生成CPUWarn后，当proclet的cpu使用率达到其从警告值中分配的50%以上时生成CPUOver。

如果proclet的cpu使用率在生成CPUWarn之前达到其分配的使用率，则生成CPUOver。

当使用率恢复为已分配的50%时，将清除CPUWarn/Over。

示例：

如果设备的系统分配是60，当值达到54时，系统生成SNMP陷阱(CPUWarn)。

由于系统的设备分配是60，当proclet的cpu使用率达到其从警告值分配的50%以上，在这种情况下，当系统达到值84(54+30)时，系统生成SNMP陷阱(CPUOver)。

## 内存使用率监控

MemoryWarn在proclet的内存使用达到其分配时生成。

当proclet的内存使用率超过其分配的+ 15MB时，即生成MemoryOver，即为其分配的两倍。

当使用率回到95%分配时，MemoryWarn/MemoryOver会被清除。

示例：

如果设备的系统分配为60MB，则对于大于60MB的任何值，系统会生成SNMP陷阱MemoryWarn。

由于设备的系统分配为60MB，因此当任务内存利用率达到75MB时，系统生成SNMP陷阱MemoryOver。

## 文件使用情况监控

Files表示打开的文件数，或进程使用的文件描述符。

没有针对文件使用情况实施SNMP陷阱，但会为超出/清除状态生成日志记录消息。

当procllet的文件使用率超过其分配的+ 10%分配时，生成超日志。

当procllet的文件使用率恢复为其已分配的90%时，生成清除日志。

```
<#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
```

## 显示任务资源中的状态

show task resources CLI输出中的状态字段具有不同的条件。

在下面的图片中，WARN为warn，而ALARM为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 )  
*/
```

## 故障排除

### 对于CPU使用率

当系统开始生成与CPU相关的SNMP陷阱时，请在出现活动问题时收集以下信息：

```
show task resources
```

检查是否有进程进入warn/over状态

```
show task resource max
```

检查最大使用量而不是当前使用量

```
show snmp trap history
```

检查是否有任何CPUWarn/Over事件

---

 注意：这是隐藏/测试命令，请参阅[文档](#)如何在StarOs中启用并进入测试模式。

---

 此命令不影响服务，可以在生产环境中运行。

---

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

这就是所谓的后台分析器。

后台分析器始终为 即使在生产中，也可以运行，固定采样周期为1.

我们可以知道哪台PC占用CPU资源、每个卡/cpu/设备/实例等。

建议使用默认值1指定深度。（例如4）

## 对于内存使用

当系统开始生成与内存相关的SNMP陷阱时，请在出现活动问题时收集以下信息：

```
show task resources
```

检查是否有进程进入warn/over状态

```
show task resource max
```

检查最大使用量而不是当前使用量

```
show snmp trap history
```

检查是否有任何MemoryWarn/Over事件

```
show logs
```

检查resmgr是否报告了任何警告/错误。

---

 注意：这是隐藏/测试命令，请参阅[文档](#)如何在StarOs中启用并进入测试模式。

---

 此命令不影响服务，可以在生产环境中运行。

---

```
show messenger proclat facility <name> instance <x> heap
```

## 检查procllet的堆使用情况

---

 注意：这是隐藏/测试命令，请参阅[文档](#)如何在StarOs中启用并进入测试模式。

---

 此命令不影响服务，可以在生产环境中运行。

---

```
show messenger procllet facility <name> instance <x> system heap
```

检查包含进程的系统堆信息

---

 提示：在向TAC提出服务请求之前，每10分钟执行多次cpu相关命令输出和4次输出。

---

## 对于文件使用

操作系统级别的实际文件限制设置高于StarOs中的文件使用量限制。

例如，任务Diameter Proxy(diaproxy),OS级别限制为8192，进程最多可以消耗8192，而文件限制在StarOS中设置为1000。

```
<#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
```

还有CPU级别限制，请也进行检查，只要您有足够的可用资源，就没问题。

<#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)
```

当可用时间小于256时，将生成以下警告消息：

```
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
```

## 关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。