

# 排除处于" ; Warn" ; 或" ; Over" ; 状态的 Sessmgr/Aaamgr故障

## 目录

---

[简介](#)

[概述](#)

[日志/基本检查](#)

[基本检查](#)

[日志](#)

[分析](#)

[行动计划](#)

[场景 1.由于内存利用率高](#)

[场景 2 : 由于高CPU利用率](#)

---

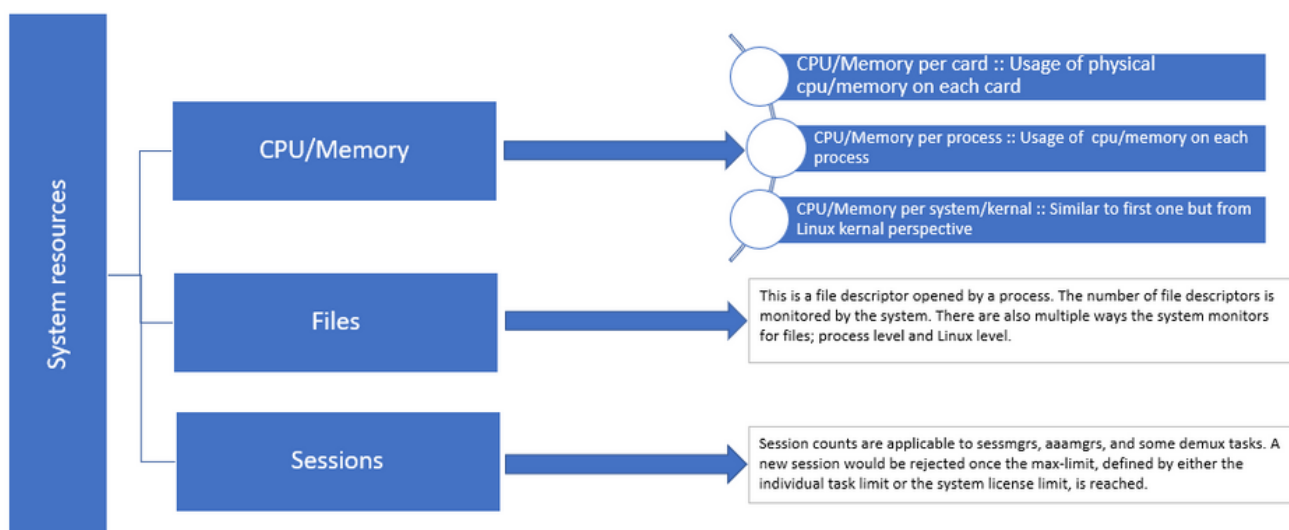
## 简介

本文档介绍如何对处于“warn”或“over”状态的sessmgr或aaamgr进行故障排除。

## 概述

会话管理器(Sessmgr) -是一个用户处理系统，支持多种会话类型，并负责处理用户事务。Sessmgr通常与AAAManagers配对。

授权、身份验证和记帐管理器(Aaamgr) -负责执行系统中用户和管理用户的所有AAA协议操作和功能。



# 日志/基本检查

## 基本检查

要收集有关问题的详细信息，您需要向用户验证以下信息：

1. sessmgr/aaamgr处于“warn”或“over”状态的时间有多长？
2. 此问题影响多少个sessmgrs/aaamgrs？
3. 您需要确认sessmgr/aaamgr是否由于内存或CPU而处于“warn”或“over”状态。
4. 您还需要检查流量是否突然增加，可以通过检查每个sessmgr的会话数来评估这一点。

通过获取此信息，您可以更好地了解并解决当前问题。

## 日志

1. 获取显示支持详细信息(SSD)和捕获有问题的时间戳的系统日志。建议在问题发生前至少2小时收集这些日志，以确定触发点。
2. 捕获有问题和无问题的sessmgr/aaamgr的核心文件。有关此过程的详细信息，请参阅分析部分。

## 分析

步骤1:通过命令检查受影响的sessmgr/aaamgr的状态。

```
show task resources
----- to check detail of sessmgr/aamgr into warn/over state and from the same you also get to know
```

Output ::

```
***** show task resources *****
Monday May 29 08:30:54 IST 2023
      task          cputime      memory      files      sessions
cpu facility inst used  alloc used alloc  used allc  used allc  S  status
-----
2/0 sessmgr  297  6.48% 100% 604.8M 900.0M  210 500    1651 12000  I  good
2/0 sessmgr  300  5.66% 100% 603.0M 900.0M  224 500    1652 12000  I  good
2/1 aaamgr   155  0.90% 95%  96.39M 260.0M   21 500     -- --    -  good
2/1 aaamgr   170  0.89% 95%  96.46M 260.0M   21 500     -- --    -  good
```

---

注意：此命令可以检查每个sessmgr的会话数，如命令输出所示。

---

这两个命令都有助于检查自节点重新加载后的最大内存使用率：

```
show task resources max
show task memory max
```

```
***** show task memory max *****
```

```
Monday May 29 08:30:53 IST 2023
```

cpu	facility	task inst	heap max	physical max	alloc		virtual max	alloc	status
2/0	sessmgr	902	548.6M	66% 602.6M	900.0M	29%	1.19G	4.00G	good
2/0	aaamgr	913	68.06M	38% 99.11M	260.0M	17%	713.0M	4.00G	good



注意：memory max命令提供自节点重新加载后使用的最大内存。此命令可帮助我们识别与问题相关的所有模式，例如问题是在最近重新加载之后开始的，或者如果最近重新加载允许我们检查最大内存值。另一方面，“show task resources”和“show task resources max”提供类似的输出，区别在于max命令显示自重新加载后特定sessmgr/aaamgr使用的内存、CPU和会话的最大值。

---

```
show subscriber summary apn <apn name> smgr-instance <instance ID> | grep Total  
----- to check no of subscribers for that particular APN in sessmg
```

## 行动计划

场景 1.由于内存利用率高

1. 在重新启动/终止sessmgr实例之前收集SSD。

2. 为任何受影响的sessmgr收集核心转储。

```
task core facility sessmgr instance <instance-value>
```

3. 在隐藏模式下使用这些命令为同一受影响的sessmgr和aaamgr收集堆输出。

```
show session subsystem facility sessmgr instance <instance-value> debug-info verbose  
show task resources facility sessmgr instance <instance-value>
```

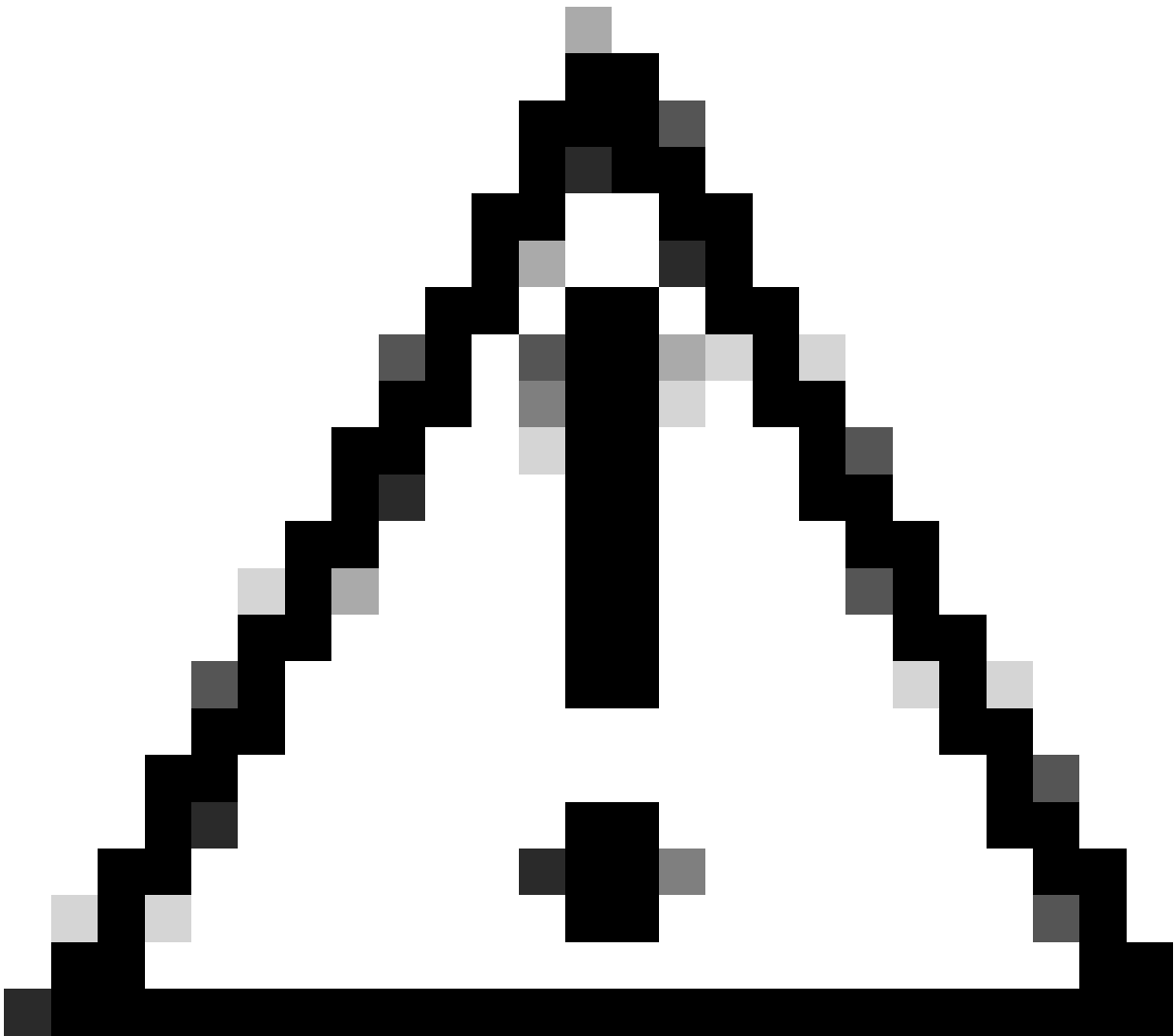
Heap outputs:

```
show messenger procllet facility sessmgr instance <instance-value> heap depth 9  
show messenger procllet facility sessmgr instance <instance-value> system heap depth 9  
show messenger procllet facility sessmgr instance <instance-value> heap  
show messenger procllet facility sessmgr instance <instance-value> system
```

```
show snx sessmgr instance <instance-value> memory ldbuf  
show snx sessmgr instance <instance-value> memory mblk
```

4. 使用以下命令重新启动sessmgr任务：

```
task kill facility sessmgr instance <instance-value>
```



注意：如果有多个处于“warn”或“over”状态的会话，建议以2到5分钟的间隔重新启动 sessmgr。首先最初仅重新启动2到3个 sessmgr，然后等待最多10到15分钟，以观察这些 sessmgr 是否恢复到正常状态。此步骤有助于评估重新启动的影响并监控恢复进度。

---

#### 5. 检查 sessmgr 的状态。

```
show task resources facility sessmgr instance <instance-value> ----- to check if sessmgr is back in
```

#### 6. 收集另一个 SSD。

#### 7. 收集步骤3中提及的所有 CLI 命令的输出。

#### 8. 使用步骤2中提到的命令收集任何运行正常的 sessmgr 实例的核心转储。



注意：要获取有问题和无问题设施的核心文件，您有两个选项。第一，在重新启动后恢复正常后，您可以收集同一sessmgr的核心文件。或者，您可以从另一个运行正常的sessmgr捕获核心文件。这两种方法都为分析和故障排除提供了宝贵的信息。

---

收集堆输出后，请与Cisco TAC联系以查找确切的堆消耗表。

从这些堆输出中，您需要检查占用更多内存的功能。基于此，TAC将调查功能利用率的预期目的，并确定其使用是否与增加的流量/交易量或其他任何有问题的原因一致。

通过使用由链接访问的工具(名为[Memory-CPU-data-sorting-tool](#))，可对堆输出进行排序。



注意：在此工具中，有多种适用于不同设施的选项。但是，您需要选择“堆消耗表”，在其中上传堆输出并运行该工具以获取已排序格式的输出。

---

## 场景 2：由于高CPU利用率

1. 在重新启动或终止sessmgr实例之前收集SSD。
2. 为任何受影响的sessmgr收集核心转储。

```
task core facility sessmgr instance <instance-value>
```

3. 为同一受影响的sessmgr/aamgr在隐藏模式下收集这些命令的堆输出。

```
<#root>
```



```
show session subsystem facility sessmgr instance <instance-value> debug-info verbose
show task resources facility sessmgr instance <instance-value>
show cpu table
show cpu utilization
```

```
show cpu info ----- Display detailed info of CPU.
show cpu info verbose ----- More detailed version of the above
```

#### Profiler output for CPU

This is the background cpu profiler. This command allows checking which functions consume the most CPU time. This command requires CLI test command password.

```
show profile facility <facility instance> instance <instance ID> depth 4
show profile facility <facility instance> active facility <facility instance> depth 8
```

#### 4. 使用以下命令重新启动sessmgr任务：

```
task kill facility sessmgr instance <instance-value>
```

#### 5. 检查sessmgr的状态。

```
show task resources facility sessmgr instance <instance-value> ----- to check if sessmgr is back in
```

#### 6. 收集另一个SSD。

#### 7. 收集步骤3中提及的所有CLI命令的输出。

#### 8. 使用步骤2中提到的命令收集任何运行正常的sessmgr实例的核心转储。

要分析高内存和CPU两种情况，请检查批量统计数据以确定流量趋势是否存在合法增长。

此外，验证卡/CPU级别统计数据的批量统计数据。

## 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）。