

# 排除故障在ASR 1K的高中央处理器/存储器问题与CUBE-ENT

## 目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[故障排除](#)

[步骤1:检查从show version的内存使用。](#)

[第二步：检查内存& CPU使用情况在IOSd内。](#)

[步骤3.检查在IOS XE的内存使用。](#)

[步骤4.语音命令平行运行。](#)

[案例研究](#)

[测验](#)

[应急方案-做的配置更改减少CPU使用情况](#)

[解决方法](#)

## 简介

本文描述能使用排除故障，并且识别高中央处理器/存储器根本原因在聚合服务路由器的一些有用的命令(ASR)与Cisco Unified Border Element企业(耳鼻喉科的多维数据集的1K发出-)。

## [先决条件](#)

## [要求](#)

Cisco 建议您了解以下主题：

- Cisco IOS
- Cisco Unified Border Element (多维数据集)

## 使用的组件

本文档中的信息基于以下软件版本：

- 与CUBE-ENT的ASR 1000

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。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

# 背景信息

ASR的1000多维数据集企业版可以支持超过15,000会话和150呼叫每秒在单个机箱根据：

[http://www.cisco.com/en/US/prod/collateral/voicesw/ps6790/gatecont/ps5640/data\\_sheet\\_c78\\_57025.html](http://www.cisco.com/en/US/prod/collateral/voicesw/ps6790/gatecont/ps5640/data_sheet_c78_57025.html)

## 故障排除

**步骤1:检查从show version的内存使用。**

```
Router#show version

Cisco IOS Software, IOS-XE Software (PPC_LINUX_IOSD-ADVIPSERVICESK9-M), Version 12.2(33)XNB,
RELEASE SOFTWARE (fc1) Technical Support:
http://www.cisco.com/techsupport Copyright (c) 1986-2008 by Cisco Systems, Inc.
Compiled Fri 05-Sep-08 08:56 by mcpre
...

If you require further assistance please contact us by sending email to export@cisco.com.

cisco ASR1006 (RP1) processor with 1779130K/6147K bytes of memory. <--- total memory allocated
to IOSd.

16 Gigabit Ethernet interfaces

21 Gigabit Ethernet interfaces

2 Ten Gigabit Ethernet interfaces

32768K bytes of non-volatile configuration memory.

4194304K bytes of physical memory. <--- IOS-XE total memory size.

955063K bytes of eUSB flash at bootflash:.

39004543K bytes of SATA hard disk at harddisk:.
```

**第二步：检查内存& CPU使用情况在IOSd内。**

```
Router#show processes memory

Processor Pool Total: 1821391588 Used: 218319000 Free: 1603072588 lsmpi_io Pool
Total: 6295088 Used: 6294116 Free: 972

PID  TTY  Allocated    Freed  Holding  Getbufs  Retbufs  Process
0      0  174405308  8586260 13472552      811    137870  *Init*
0      0      65688    393404    152         0         0  *Sched*
```

0	0	21603272	48285960	274932	3	1	*Dead*
0	0	0	0	406304	0	0	*MallocLite*
1	0	431576	0	448716	0	0	Chunk Manager
2	0	236	236	11140	0	0	Load Meter
3	0	2785880	2782996	32092	0	0	Exec
4	0	0	0	17140	0	0	Retransmission
5	0	34360	0	17140	0	0	IPC ISSU Dispatc
6	0	3336	236	20240	0	0	Check heaps
7	0	32780	32780	17140	45	0	Pool Manager
8	0	236	236	17140	0	0	Timers
9	0	206550924	20696084	71980	9326586	9326586	ARP Input
10	0	24356	24356	17140	111	111	ARP Background
11	0	236	236	17140	0	0	ATM Idle Timer
12	0	0	0	17140	0	0	ATM ASYNC PROC
13	0	0	0	17140	0	0	AAA_SERVER_DEADT
14	0	0	0	29140	0	0	Policy Manager
15	0	59092	692	74972	172	172	Entity MIB API

您能也运行排序的show process memory。

Router# show processes cpu

CPU utilization for five seconds: 0%/0%; one minute: 0%; five minutes: 0%

PID	Runtime(uS)	Invoked	uSecs	5Sec	1Min	5Min	TTY	Process
1	4000	67	59	0.00%	0.00%	0.00%	0	Chunk Manager
2	4000	962255	0	0.00%	0.00%	0.00%	0	Load Meter
3	0	1	0	0.00%	0.00%	0.00%	0	cpf_process_tp
4	0	1	0	0.00%	0.00%	0.00%	0	EDDRI_MAIN
5	586520704	732013	6668	0.00%	0.11%	0.08%	0	Check heaps
6	4000	991	4	0.00%	0.00%	0.00%	0	Pool Manager
7	0	1	0	0.00%	0.00%	0.00%	0	DiscardQ Backg
8	0	2	0	0.00%	0.00%	0.00%	0	Timers
9	0	2	0	0.00%	0.00%	0.00%	0	ATM AutoVC Per
10	0	2	0	0.00%	0.00%	0.00%	0	ATM VC Auto Cr

```
11 215495600 4809201 448 0.00% 0.03% 0.03% 0 EnvMon
Router#show memory statistics
```

Head	Total(b)	Used(b)	Free(b)	Lowest(b)	Largest(b)
Processor 7FEB87F20010	1235972656	672082844	563889812	562129816	459757740
lsmpi_io 7FEB876C51A8	6295128	6294212	916	916	916

```
Router#show memory allocating-process totals Head Total(b) Used(b) Free(b) Lowest(b) Largest(b)
Processor 7FEB87F20010 1235972656 672078752 563893904 562129816 459757740 lsmpi_io 7FEB876C51A8
6295128 6294212 916 916 916
```

```
Router#show memory debug leaks {|chunks}
```

```
Adding blocks for GD...
```

#### lsmpi\_io memory

Address	Size	Alloc_pc	PID	Alloc-Proc	Name
---------	------	----------	-----	------------	------

#### Processor memory

Address	Size	Alloc_pc	PID	Alloc-Proc	Name
---------	------	----------	-----	------------	------

7FEB984B4A30	360	4F50BBE	62	IOSD ipc task	IOSD ipc task 7FEB9853FC68
528	1267A20	421 SBC main proces		Name info	
7FEB9F40D9D0	424	1267A20	421	SBC main proces	Name info
7FEB9F40DB78	472	1267A20	421	SBC main proces	Name info
7FEBA63E7338	456	1267A20	421	SBC main proces	Name info
7FEBA6528758	448	1267A20	421	SBC main proces	Name info
7FEBA655B7F0	456	1267A20	421	SBC main proces	Name info

### 步骤3.检查在IOS XE的内存使用。

为了显示在Cisco IOS XE的当前系统内存使用情况，请发出此命令：

```
Router#show platform software status control-processor brief Load Average Slot Status 1-Min 5-
Min 15-Min RP0 Healthy 0.20 0.23 0.19 RP1 Healthy 0.19 0.19 0.12 ESP0 Healthy 0.65 0.54 0.47
SIP1 Healthy 0.17 0.07 0.01 SIP2 Healthy 0.02 0.06 0.01 Memory (kB) Slot Status Total Used (Pct)
Free (Pct) Committed (Pct) RP0 Healthy 3919872 2710788 (65%) 1209084 (29%) 2327484 (56%) RP1
Healthy 3919872 2377136 (57%) 1542736 (37%) 2320964 (56%) ESP0 Healthy 2030444 1112344 (53%)
918100 (43%) 3409068 (162%) SIP1 Healthy 484452 293408 (55%) 191044 (36%) 244180 (46%) SIP2
Healthy 484452 293408 (55%) 191044 (36%) 244020 (46%) CPU Utilization Slot CPU User System Nice
Idle IRQ SIRQ Iowait RP0 0 10.91 1.88 0.00 86.67 0.38 0.13 0.00 RP1 0 8.06 1.22 0.00 90.11 0.00
0.03 0.55 ESP0 0 5.78 3.61 0.00 90.51 0.02 0.05 0.00 SIP1 0 4.32 0.45 0.00 95.20 0.00 0.01 0.00
SIP2 0 3.95 0.44 0.00 95.57 0.00 0.01 0.00
```

为了显示运行在Cisco IOS XE的每进程的内存使用，请发出此命令：

```
Router#monitor platform software process rp active
```

```
top - 05:18:46 up 14 days, 17:33, 0 users, load average: 0.00, 0.01, 0.00 Tasks: 119 total,
1 running, 118 sleeping, 0 stopped, 0 zombie Cpu(s): 0.4% us, 0.4% sy, 0.0% ni, 99.1% id,
0.0% wa, 0.0% hi, 0.0% si Mem: 3714760k total, 1454344k used, 2260416k free, 97952k
buffers Swap: 0k total, 0k used, 0k free, 875376k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
17385	root	20	0	1874m	338m	75m	S	0.2	9.3	65:59.18	ppc_linux_iosd-
18098	root	20	0	71880	59m	6324	S	0.2	1.6	10:48.84	smand
16521	root	20	0	87868	51m	47m	S	0.0	1.4	0:02.80	fman_rp
16903	root	20	0	27788	16m	14m	S	0.0	0.5	15:41.61	imand
15957	root	20	0	24776	9696	6880	S	0.2	0.3	12:49.67	cmand
17697	root	20	0	19504	6160	4544	S	0.0	0.2	0:00.95	psd
16316	root	20	0	18232	5972	3736	S	0.0	0.2	12:43.32	emd
16732	root	20	0	16184	5556	3900	S	0.4	0.1	21:22.61	hman
17237	root	20	0	15892	5456	3088	S	0.0	0.1	0:00.99	plogd
15166	root	20	0	4056	2396	1248	S	0.0	0.1	0:00.72	pvp.sh
16937	root	9	-11	3992	2308	1232	S	0.0	0.1	0:00.13	pman.sh
15559	root	9	-11	3992	2304	1228	S	0.0	0.1	0:00.13	pman.sh
17978	root	9	-11	3992	2304	1228	S	0.0	0.1	0:00.13	pman.sh

**提示：**在屏幕出现后，请选择"shift + M"为了排序与内存使用的显示的进程。

**注意：**RES指示非交换的物理存储器进程用途和SHR指示共享内存量由进程使用了。RES + SHR是进程的总量，并且%MEM指示可用的物理存储器当前使用的共享进程的。

#### 步骤4.语音命令平行运行。

```
Router#show sip-ua calls summary
```

```
Router#show sip-ua statistics
```

```
Router#show call history stats cps <-- Displays the call rate per second for CUBE
```

```
Call switching rate / CPS (last 60 seconds)
```

```
Period Actual Average
```

```
-----
```

```
1-5 61 12
```



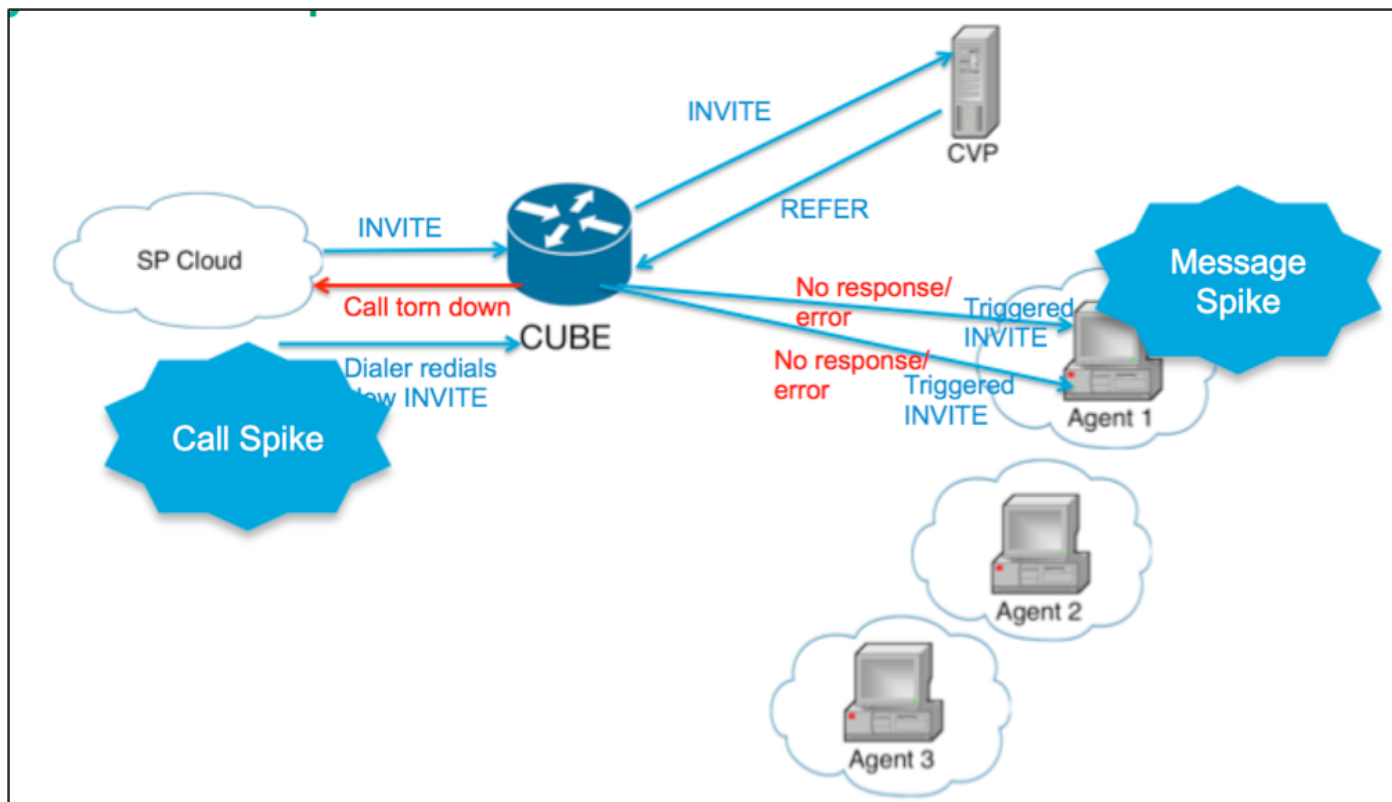


# = average sip messages/s

由于大呼叫量，推荐使用eem脚本。

## 案例研究

CPU峰值解剖学。



## 测验

测试 1 :

- 被处理的总呼叫量(Site1 – 11,000 & Site2 – 95000)。
- 内存统计信息是正如所料。
- 没有挂起会话

测试 2 :

- 托塔尔35,000呼叫。
- 内存统计信息是正如所料
- 没有挂起会话
- 装载6550安排CPU CAC被本格化并且限制了呼叫 静态呼叫报告在最大负荷-多维数据集统计信息没有指示任何反常性

收集数据和命令使用的EEM脚本在测验期间。

在测验前开始 :

1. 清楚SIP UA统计信息



2. clear counters

3. 删除所有现有文件“BB\_Workaround\_Load\_Info.txt”从硬盘(那么没有包含旧有数据)的eem脚本文件。确保使用文件的备份。

4. 等待3分钟检查eem是否收集数据并且开始测验。

```
applet BB_WORKAROUND

cronmem cron"* /3 * * * *"

01.0 cli

01.1 cli"LEN 0"

02.0 cli"show process CPU|
harddisk:/XE310_1/BB_Workaround_Load_Info.txt";

02.1 cli"|harddisk:/XE310_1/BB_Workaround_Load_Info.txt";

cli02.2 "|harddisk:/XE310_1/BB_Workaround_Load_Info.txt";

02.3 cli"show sip-ua|harddisk:/XE310_1/BB_Workaround_Load_Info.txt";

02.5 cli"show sip-ua|harddisk:/XE310_1/BB_Workaround_Load_Info.txt";

02.6 cli"show call history stats cps|
harddisk:/XE310_1/BB_Workaround_Load_Info.txt";

02.7 cli"show sip-uastats|
harddisk:/XE310_1/BB_Workaround_Load_Info.txt";

02.8 cli"show processes cpu|
harddisk:/XE310_1/BB_Workaround_Load_Info.txt";

02.9 cli"qfp|harddisk:/XE310_1/BB_Workaround_Load_Info.txt";
```

在完成测验后：

收集以下使用PuTTY日志或会话登陆终端。

```
APP

APP

show call active voice

show sip-ua

show sip-ua

show voip rtp connections

voip fpi

sbc RP
```

```
sbc fp
qfpsbclvl2-hash-table

show call history stats cps

show sip-uastats

show sip-ua

show memory statistics

show memory processor

show memory allocating-process totals

show processes cpu

show processes cpu
```

## 应急方案-做的配置更改减少CPU使用情况

步骤1.添加媒体range命令地址的/端口。

```
voip
RTP18000 32768 <---
10.252.47.201 10.252.47.201 <---dial-peer
18000 32768 <---RTPIOSDCPU
```

步骤2.对话选项ping删除。

"sip options-ping 60"删除从dial-peer

减少每呼叫16个SIP消息。

步骤3. CPU呼叫接纳控制设置在75%

分支25%处理例如参考的呼叫的， MID呼叫邀请等。

一旦一突然的尖峰，我们不点击100% -

```
CPU avg6875
```

```
mem7585
```

## 解决方法

修正在新的IOS提供 : asr1000rp2-adventerprisek9.03.10.01.S.153-3.S1-ext.bin