

MallocLite存储器泄漏排除故障

Contents

[Introduction](#)

[背景信息](#)

[Troubleshoot](#)

[识别应用程序负责对泄漏](#)

[解码分配程序PC](#)

[调查MallocLite内存统计数据](#)

[禁用MallocLite](#)

Introduction

本文描述如何排除MallocLite在Cisco IOS软件平台的存储器泄漏故障。

它也指定您应该收集的信息，在您开Cisco技术支持中心(TAC) Case或重新载入设备前。收集在本文提及的输出，并且附有他们到TAC案例为了帮助加快问题解决方法。

背景信息

内存管理器用于MallocLite为了分配内存小，固定尺寸的部分，叫作大块，分配的小于或等于128个字节。小的存储器分配没有一个块首的开销每个分配的。此功能为仅处理器内存池支持。

每个存储器块报头取走大约48内存字节数，并且最小的块取走大约24个字节。使用在Cisco IOS软件的一传统方法的每个分配，您会消耗至少72 (48 + 24)内存字节数，即使您只需要分配8字节的实际数据。

使用MallocLite，此开销可以减少使用大块。因为大块必须被管理，仍有若干开销。然而，因为大块是固定尺寸的，他们与块不同方式被管理，并且开销是较少。

是使用MallocLite内存适当地释放它应用程序的责任。MallocLite屏蔽内存的用户。

Troubleshoot

Note: [思科 CLI 分析器](#) ([仅适用于注册客户](#)) 支持某些 **show** 命令。要查看对 **show** 命令输出的分析，请使用思科 CLI 分析器。

识别应用程序负责对泄漏

如果由`malloclite`关键字，仅搜索识别一个现有的Bug通常是难的。

此示例表示，*MallocLite*进程暂挂一异常内存数量：

```
#show processes memory sorted
```

```
Processor Pool Total: 1614282720 Used: 1544726580 Free: 69556140
I/O Pool Total: 313524224 Used: 115564032 Free: 197960192
```

```
PID TTY Allocated Freed Holding Getbufs Retbufs Process
0 0 0 0 1476043512 0 0 *MallocLite*
```

您需要识别对泄漏负责的确切的应用程序。三个可能的证明方法是：

- 解码分配程序PC。
- 调查MallocLite内存统计数据。
- 禁用MallocLite。

解码分配程序PC

与MallocLite打开了，您能通常看到什么作用请求内存。**show memory allocating-process totals**命令威力的输出显示不同的PC值，即使报告的名字是MallocLite：

```
#show processes memory sorted
```

```
Processor Pool Total: 1614282720 Used: 1544726580 Free: 69556140
I/O Pool Total: 313524224 Used: 115564032 Free: 197960192
```

```
PID TTY Allocated Freed Holding Getbufs Retbufs Process
0 0 0 0 1476043512 0 0 *MallocLite*
```

Cisco TAC工程师能从列表的顶端解码PC值(与最高的总数)。这帮助识别有存储器泄漏的应用程序。

调查MallocLite内存统计数据

在Cisco IOS软件版本添加的增进中显示每个PC分配的MallocLite内存汇总的15.1T是新的CLI。**show memory轻大块**命令可帮助您识别使用很多MallocLite块的应用程序。

```
#show processes memory sorted
```

```
Processor Pool Total: 1614282720 Used: 1544726580 Free: 69556140
I/O Pool Total: 313524224 Used: 115564032 Free: 197960192
```

```
PID TTY Allocated Freed Holding Getbufs Retbufs Process
0 0 0 0 1476043512 0 0 *MallocLite*
```

请参见[命令参考](#)关于**show memory轻大块**命令的详细资料。

```
#show processes memory sorted
```

```
Processor Pool Total: 1614282720 Used: 1544726580 Free: 69556140
I/O Pool Total: 313524224 Used: 115564032 Free: 197960192
```

```
PID TTY Allocated Freed Holding Getbufs Retbufs Process
0 0 0 0 1476043512 0 0 *MallocLite*
```

输出示例此include命令的：

```
#show processes memory sorted
```

```
Processor Pool Total: 1614282720 Used: 1544726580 Free: 69556140
I/O Pool Total: 313524224 Used: 115564032 Free: 197960192
```

```
PID TTY Allocated Freed Holding Getbufs Retbufs Process
0 0 0 0 1476043512 0 0 *MallocLite*
```

再次，TAC工程师能解码PC值以最高总和识别漏内存的应用程序。

功能失效MallocLite

默认情况下MallocLite功能被启用。为了调查MallocLite泄漏，您能禁用MallocLite：

```
#show processes memory sorted
```

```
Processor Pool Total: 1614282720 Used: 1544726580 Free: 69556140
I/O Pool Total: 313524224 Used: 115564032 Free: 197960192
```

```
PID TTY Allocated Freed Holding Getbufs Retbufs Process
0 0 0 0 1476043512 0 0 *MallocLite*
```

漏的内存存在直到下一次重新加载的MallocLite下;然而，您能开始监控与被排序的show processes memory的进一步泄漏，并且show memory allocating-process共计命令。泄漏当前将出现在真正的流程下。

如果设备在内存非常低速运行，您必须保存配置和重新载入设备为了发布内存：

```
#show processes memory sorted
```

```
Processor Pool Total: 1614282720 Used: 1544726580 Free: 69556140
I/O Pool Total: 313524224 Used: 115564032 Free: 197960192
```

```
PID TTY Allocated Freed Holding Getbufs Retbufs Process
0 0 0 0 1476043512 0 0 *MallocLite*
```

内存也许随着时间的推移再耗尽，因此请使用被排序的show processes memory，并且show memory allocating-process共计命令为了从那时起监控存储器使用。

Note:如果有效禁用MallocLite用没有内存轻命令并且重新载入设备，show memory轻大块命令的输出将是空的。

请参见[命令参考](#)关于内存轻命令的详细资料。