



目 录

目 录.....	2
第 13 章 异常高 CPU 占用率故障排查.....	3
13.1 异常高 CPU 占用率.....	3
13.2 异常高 CPU 占用率的故障症状.....	3
13.3 异常高 CPU 占用率探查.....	3
13.3.1 探查概述.....	3
13.3.2 初步探查.....	4
13.3.3 进一步跟踪.....	4
13.4 异常高 CPU 占用率故障排除策略.....	7

Beijing Landing Technologies

第 1 章 异常高 CPU 占用率故障排查

1.1 异常高 CPU 占用率

异常高 CPU 占用率：当一个进程或线程占用的 CPU 资源异常高时发生，可能会使其他进程或线程丧失或缺乏执行要求的处理任务所需的 CPU 处理能力，应进行定期检查。

1.2 异常高 CPU 占用率的故障症状

- 用户响应时间长；
- Tuxedo 服务器运行速度异常慢；
- 请求或操作开始出现超时。

1.3 异常高 CPU 占用率探查

1.3.1 探查概述

基本步骤：

- 首先确定哪个进程或是线程导致了高 CPU 占用率
- 通过各种系统工具或命令探究具体的问题原因

以下就以最简单的 Tuxedo 程序 Demo simpapp 为例，为了更好的复现 CPU 异常走高，Tuxedo 服务端添加了一个不带 sleep 的死循环。

这时客户端会发现请求始终 没有返回，从而导致超时：

```
[tuxedo@landingbj test]# simpcl test
Can't send request to service TOUPPER
Tperrno = 13
```

示例 13-1

此时在服务机 输入 top 命令：

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
4843	tuxedo	25	0	6424	2512	1712	R	90.0	1.0	1:29.33	simpserv
1	root	17	0	2072	620	532	S	0.0	0.2	0:01.36	init
2	root	RT	-5	0	0	0	S	0.0	0.0	0:00.00	migration/
3	root	34	19	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/
4	root	RT	-5	0	0	0	S	0.0	0.0	0:00.00	watchdog/0
5	root	10	-5	0	0	0	S	0.0	0.0	0:00.01	events/0
6	root	12	-5	0	0	0	S	0.0	0.0	0:00.00	khelper
7	root	13	-5	0	0	0	S	0.0	0.0	0:00.00	kthread
10	root	10	-5	0	0	0	S	0.0	0.0	0:00.02	kblockd/0

示例 13-2

很明显，进程 4843（即 simpserv 所属进程）导致 CPU 占用率很高

为了探究具体是哪一个系统调用或 API 上引起的问题下面在几个不同操作系统上的探查。

1.3.2 初步探查

1. 在 Linux 上初步探查

可以运行 `strace` 命令探查系统调用：

```
strace -o outfile -p <PID>
```

示例 13-3

例如，针对上述 `top` 结果中在第一位的 `simpsserv`：

```
strace -o strace.out -p 4843
```

示例 13-4

2. 在 HP-UX 上初步探查

可以运行 `tusc` 命令探查进程的系统调用，使用 `tusc` 可以获得进程系统调用的所有信息，HP-UX 也提供了 `truss` 和 `tusc` 一起打包，命令行如下：

```
truss -d -o <outfile> -p <pid>
```

示例 13-5

“-d” 参数是列出所有系统调用的时间戳。

3. 在 Solaris/AIX 上初步探查

Solaris 或 AIX 上也可以使用 `truss` 命令查看系统调用。

1.3.3 进一步跟踪

操作系统提供了很多工具可以辅助我们进行追踪，比如 `truss`、`strace`、`gdb`、`dbx` 等系统调用工具。

所有的 UNIX 程序都是通过调用操作系统提供的这些底层服务来完成它们的任务，使用这些工具，可以清楚地看到这些调用过程及其使用的参数。

通过这种方式，也可以了解程序与操作系统之间的底层交互

```
strace -p 4843
Process 4843 attached - interrupt to quit
msgrcv(65538, 65538, {536870962,
"p\0\0\0\2\0\1\0\0\0\0\0\270\1\0\0\0\0\200\0\0\0\0\0\0X\230\3\221"...},
4572, -1073741824, 0) = 436
_llseek(8, 8704, [8704], SEEK_SET) = 0
read(8,
"\375\365\0\0\310\0\0\0, \1\0\0\0\0\0\244\0\0\0\363.\0\01\1\0\0\330\1\0\0"...
, 4096) = 4096
futex(0x293440, FUTEX_WAKE_PRIVATE, 2147483647) = 0
futex(0x441844, FUTEX_WAKE_PRIVATE, 2147483647) = 0
_llseek(8, 4608, [4608], SEEK_SET) = 0
```

```
read(8,
"\270\26\0\0\310\0\0\0004\1\0\0\0\0\0\0\244\0\0\0}\351\0\01\1\0\0000\3\0\0"...
, 4096) = 4096
futex(0x441ac0, FUTEX_WAKE_PRIVATE, 2147483647) = 0
futex(0x4418a4, FUTEX_WAKE_PRIVATE, 2147483647) = 0
time(NULL) = 1302007088
access("/home/tuxedo/test/ULOG.040511", F_OK) = 0
open("/home/tuxedo/test/ULOG.040511", O_WRONLY|O_CREAT|O_APPEND|O_LARGEFILE,
0666) = 3
write(3, "053808.landingbj!simpserv.4843."..., 131) = 131
close(3) = 0
Process 4843 detached (此处为 Ctrl+C 结束)
```

示例 13-6

当执行到 `simpserv` 时一直阻塞在此处，不往下执行任何操作（表现：不再往下打印各种系统调用信息）

再比如，`gdb` 等程序调试命令（确定问题后，用于调试问题程序）

```
gdb --quiet
(gdb) attach 4843
Reading symbols from /home/tuxedo/test/simperv... (no debugging symbols
found)... done.
Reading symbols from /home/tuxedo/tux11/tuxedo11gR1/lib/libtux.so... done.
Loaded symbols for /home/tuxedo/tux11/tuxedo11gR1/lib/libtux.so
Reading symbols from /home/tuxedo/tux11/tuxedo11gR1/lib/libbuft.so... done.
Loaded symbols for /home/tuxedo/tux11/tuxedo11gR1/lib/libbuft.so
Reading symbols from /home/tuxedo/tux11/tuxedo11gR1/lib/libfml.so... done.
Loaded symbols for /home/tuxedo/tux11/tuxedo11gR1/lib/libfml.so
Reading symbols from /home/tuxedo/tux11/tuxedo11gR1/lib/libfml32.so... done.
Loaded symbols for /home/tuxedo/tux11/tuxedo11gR1/lib/libfml32.so
Reading symbols from /home/tuxedo/tux11/tuxedo11gR1/lib/libengine.so... done.
Loaded symbols for /home/tuxedo/tux11/tuxedo11gR1/lib/libengine.so
Reading symbols from /lib/libdl.so.2... done.
Loaded symbols for /lib/libdl.so.2
Reading symbols from /lib/libpthread.so.0... done.
[Thread debugging using libthread_db enabled]
[New Thread 0xb7f9d8d0 (LWP 4843)]
Loaded symbols for /lib/libpthread.so.0
Reading symbols from /lib/libc.so.6... done.
Loaded symbols for /lib/libc.so.6
Reading symbols from /home/tuxedo/tux11/tuxedo11gR1/lib/libutrace.so... done.
Loaded symbols for /home/tuxedo/tux11/tuxedo11gR1/lib/libutrace.so
Reading symbols from /home/tuxedo/tux11/tuxedo11gR1/lib/libgiconv.so... done.
Loaded symbols for /home/tuxedo/tux11/tuxedo11gR1/lib/libgiconv.so
Reading symbols from /home/tuxedo/tux11/tuxedo11gR1/lib/libusort.so... done.
Loaded symbols for /home/tuxedo/tux11/tuxedo11gR1/lib/libusort.so
Reading symbols from /lib/libnsl.so.1... done.
Loaded symbols for /lib/libnsl.so.1
```

```
Reading symbols from /lib/ld-Linux.so.2...done.
Loaded symbols for /lib/ld-Linux.so.2
Reading symbols from /home/tuxedo/tux11/tuxedollgR1/lib/registry.so...done.
Loaded symbols for /home/tuxedo/tux11/tuxedollgR1/lib/registry.so
0x08048710 in TOUPPER ()
```

示例 13-7

attach 4843 命令启动对当前正在运行的 Tuxedo 服务器的调试工作。

用 info 命令则列出程序信息：

```
info proc
process 4843
cmdline = 'simpserv'
cwd = '/home/tuxedo/test'
exe = '/home/tuxedo/test/simpserv'
```

示例 13-8

info functions 命令得到函数列表：

```
(gdb) info functions
All defined functions:

Non-debugging symbols:
0x080484ac  _init
0x080484d4  __gmon_start__@plt
0x080484e4  toupper@plt
0x080484f4  __libc_start_main@plt
0x08048504  tpsvrdone
0x08048504  tpsvrdone@plt
0x08048514  _tmstartserver@plt
0x08048524  _tmrunserver
0x08048524  _tmrunserver@plt
0x08048534  userlog@plt
0x08048544  tpsvrthrdone
---Type <return> to continue, or q <return> to quit---Type
0x08048544  tpsvrthrdone@plt
0x08048554  tpsvrthrinit
0x08048554  tpsvrthrinit@plt
0x08048570  _start
0x08048594  call_gmon_start
0x080485c0  __do_global_dtors_aux
0x08048620  frame_dummy
0x08048644  _tmgetsvrargs
0x0804866c  main
0x080486a4  tpsvrinit
0x080486bd  TOUPPER
```

示例 13-9

以上是 simpserv 应用程序部分信息。

通过 where 命令获取进程堆栈信息:

```
(gdb) where
#0  0x08048710 in TOUPPER ()
#1  0x00171bf5 in _tmsvcdsp ()
    from /home/tuxedo/tux11/tuxedo11gR1/lib/libtux.so
#2  0x0019c22c in _tmrunserver ()
    from /home/tuxedo/tux11/tuxedo11gR1/lib/libtux.so
#3  0x0016f72d in _tmstartserver ()
    from /home/tuxedo/tux11/tuxedo11gR1/lib/libtux.so
#4  0x0804869a in main ()
(gdb) detach
Detaching from program: /home/tuxedo/test/simpserv, process 4843
(gdb) quit
```

示例 13-10

1.4 异常高 CPU 占用率故障排除策略

1. 通过重复收集信息 确定 CPU 占用率异常;
2. 在服务器端通过系统命令查找 CPU 占用率过高的进程;
3. 通过各种系统工具对进程的分析找出导致高 CPU 占用率的具体原因;
4. 继续监视, 看是否还会发生故障;
5. 控制端超时时间 (UBBCONFIG 中 WSL CLOPT [-t timeout]), 以减少系统故障的危害。