

Анализ загрузки процессора

В некоторых случаях возникает необходимость выяснить, какой именно поток (thread) и на сколько загружает процессор сервера. Для этого:

1. Определите идентификатор процесса (pid) WCS при помощи команды

```
pgrep -fn com.flashphoner.server.Server
```

или

```
top
```

2. Получите информацию о потоках процесса при помощи `jstack`

```
jstack 4240 >> 4240.jstack
```

Здесь `4240` - pid процесса WCS, полученный на шаге 1.

Файл, полученный в результате выполнения команды, будет следующим

```
2018-08-21 09:44:05
Full thread dump Java HotSpot(TM) 64-Bit Server VM (25.161-b12 mixed mode):

"Attach Listener" #213 daemon prio=9 os_prio=0 tid=0x00007f6bb8341800 nid=0x4f0b
waiting on condition [0x0000000000000000]
  java.lang.Thread.State: RUNNABLE

"process reaper" #119 daemon prio=10 os_prio=0 tid=0x000000000257a000 nid=0x7196
waiting on condition [0x00007f6bac072000]
  java.lang.Thread.State: TIMED_WAITING (parking)
    at sun.misc.Unsafe.park(Native Method)
      - parking to wait for <0x00000000f3d49ac8> (a java.util.concurrent.Synchron
hronousQueue$TransferStack)
    at java.util.concurrent.locks.LockSupport.parkNanos(LockSupport.java:215
)
    at java.util.concurrent.SynchronousQueue$TransferStack.awaitFulfill(Sync
hronousQueue.java:460)
    at java.util.concurrent.SynchronousQueue$TransferStack.transfer(Synchron
ousQueue.java:362)
    at java.util.concurrent.SynchronousQueue.poll(SynchronousQueue.java:941)
    at java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.ja
va:1073)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.
java:1134)
```

Здесь можно просмотреть статусы потоков, а также их идентификаторы в шестнадцатиричном виде в поле 'nid'

3. Выполните команду

```
top -H -p 4240
```

Здесь 4240 - pid процесса WCS, полученный на шаге 1.

На консоль будут выведены потоки процесса и потребление ими процессора и памяти

```
top - 11:59:35 up 134 days, 14:13, 1 user, load average: 0.00, 0.01, 0.05
Threads: 50 total, 0 running, 50 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.8 us, 1.5 sy, 0.0 ni, 97.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 1883696 total, 76664 free, 1636952 used, 170080 buff/cache
KiB Swap: 2097148 total, 2016960 free, 80188 used. 75508 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
4301	root	20	0	2965436	1.003g	19112	S	0.3	55.9	0:52.40	java
7018	root	20	0	2965436	1.003g	19112	S	0.3	55.9	0:04.60	java
4240	root	20	0	2965436	1.003g	19112	S	0.0	55.9	0:00.00	java
4241	root	20	0	2965436	1.003g	19112	S	0.0	55.9	0:02.63	java
4243	root	20	0	2965436	1.003g	19112	S	0.0	55.9	0:00.23	java
4244	root	20	0	2965436	1.003g	19112	S	0.0	55.9	0:00.23	java
4245	root	20	0	2965436	1.003g	19112	S	0.0	55.9	0:00.30	java
4246	root	20	0	2965436	1.003g	19112	S	0.0	55.9	0:00.85	java
4247	root	20	0	2965436	1.003g	19112	S	0.0	55.9	0:00.02	java
4248	root	20	0	2965436	1.003g	19112	S	0.0	55.9	0:00.03	java
4249	root	20	0	2965436	1.003g	19112	S	0.0	55.9	0:00.00	java
4250	root	20	0	2965436	1.003g	19112	S	0.0	55.9	0:00.00	java
4251	root	20	0	2965436	1.003g	19112	S	0.0	55.9	0:23.22	java
4252	root	20	0	2965436	1.003g	19112	S	0.0	55.9	0:03.99	java
4253	root	20	0	2965436	1.003g	19112	S	0.0	55.9	0:00.00	java
4254	root	20	0	2965436	1.003g	19112	S	0.0	55.9	0:00.00	java
4255	root	20	0	2965436	1.003g	19112	S	0.0	55.9	0:00.00	java

В колонке 'PID' указаны идентификаторы потоков в десятичном виде. По этому идентификатору можно найти поток в файле, полученном на шаге 2, и определить его имя.