06 | 案例篇：系統的 CPU 使用率很高，但為啥卻找不到高 CPU 的應用？

John Chen

2020/05/07

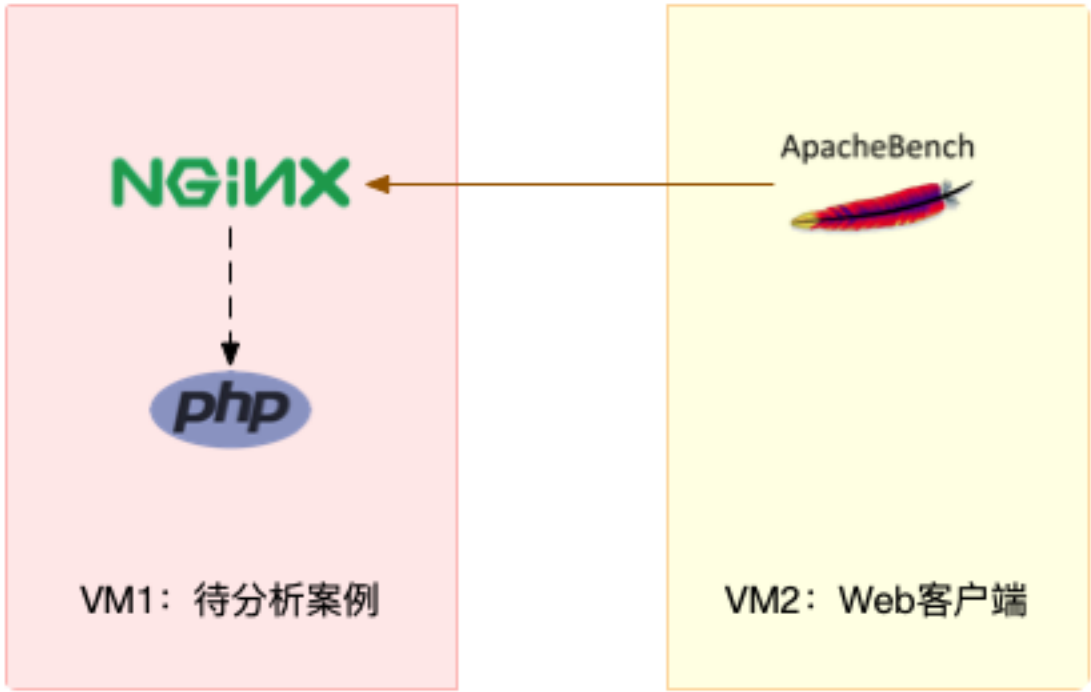
案例分析

- 機器配置：2 CPU，8GB 記憶體

- 預先安裝 docker、sysstat、perf、ab 等工具，如 apt install docker.io sysstat linux-tools-common apache2-utils

- https://github.com/feiskyer/linux-perf-examples/tree/master/nginx-short-process

案例分析



設定過程 1/3

```
root@ip-172-31-12-157:~# docker run --name nginx -p 10000:80 -itd feisky/nginx:sp
Unable to find image 'feisky/nginx:sp' locally
sp: Pulling from feisky/nginx
802b00ed6f79: Pull complete
c16436dbc224: Pull complete
683eac851b28: Pull complete
77b147d01876: Pull complete
b69b3b49a38c: Pull complete
Digest: sha256:0ad15ef0534a615a6ed20b0637255a67cbb4c67492e3e416257ac4515aec6b04
Status: Downloaded newer image for feisky/nginx:sp
a2287703b399d694d8b8c89c27444dc217e04da1d0697be0c78b2833f31fa89a
```

設定過程 2/3

```
root@ip-172-31-12-157:~# docker run --name phpfpm -itd --network container:nginx feisky/php-fpm:sp
Unable to find image 'feisky/php-fpm:sp' locally
sp: Pulling from feisky/php-fpm
a02fbcf48e6b: Pull complete
0fd90e182cc5: Pull complete
554d282927ec: Pull complete
805a76c208d2: Pull complete
7815988b0b73: Pull complete
5937bbd596b6: Pull complete
eb60d700b9c1: Pull complete
a1fc81bebde7: Pull complete
a4ae4e49d6a5: Pull complete
b71e96a90b11: Pull complete
8ce6d254af53: Pull complete
Digest: sha256:b3835a298915058f41cf4c01895dd6556b272f6035a77ce1dad1f0c39f37b19f
Status: Downloaded newer image for feisky/php-fpm:sp
d1fd94ceb276a6557e38088bd30b3560a41b34e00f0894bd10d449f73dc73bbe
```

設定過程 3/3

```
[root@ip-172-31-12-157:~# curl http://172.31.12.157:10000/
[It works!root@ip-172-31-12-157:~#
```

測試結果

```
root@ip-172-31-12-157:~# ab -c 100 -n 1000 http://172.31.12.157:10000/ | grep "Requests"
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests
Requests per second:    218.70 [#/sec] (mean)
```

```
top - 11:48:02 up 13 days,  4:48,  2 users,  load average: 4.21, 2.62, 1.53
Tasks: 148 total,   6 running,  93 sleeping,   0 stopped,   0 zombie
%Cpu0  : 70.8 us, 16.9 sy,  0.0 ni, 11.5 id,  0.0 wa,  0.0 hi,  0.7 si,  0.0 st
%Cpu1  : 73.1 us, 16.5 sy,  0.0 ni, 10.4 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu2  : 72.0 us, 15.2 sy,  0.0 ni, 11.8 id,  0.0 wa,  0.0 hi,  1.0 si,  0.0 st
%Cpu3  : 68.3 us, 19.6 sy,  0.0 ni, 11.4 id,  0.0 wa,  0.0 hi,  0.7 si,  0.0 st
KiB Mem :  7807720 total,  4575940 free,   332752 used,  2899028 buff/cache
KiB Swap:        0 total,        0 free,        0 used.  7170028 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 1255 daemon    20   0  336696  16616   8940 S   5.0  0.2   0:04.46 php-fpm
 1253 daemon    20   0  336696  16552   8876 S   4.7  0.2   0:04.49 php-fpm
 1254 daemon    20   0  336696  16552   8876 S   4.7  0.2   0:04.52 php-fpm
 1256 daemon    20   0  336696  16552   8876 S   4.7  0.2   0:04.53 php-fpm
 1257 daemon    20   0  336696  16552   8876 S   4.7  0.2   0:04.49 php-fpm
 2269 systemd+  20   0   33996   4684   2440 S   3.7  0.1   0:17.17 nginx
29103 root      20   0   33228   6736   4964 S   2.7  0.1   0:00.20 ab
 2205 root      20   0   10772   4544   3944 S   1.3  0.1   0:06.17 containerd-shim
 4501 root      20   0 1352224 105392  49352 S   1.3  1.3   4:36.48 dockerd
 2270 systemd+  20   0   33220   3972   2440 S   1.0  0.1   0:02.42 nginx
   10 root      rt   0       0      0      0 S   0.3  0.0   0:00.22 migration/0
   16 root      20   0       0      0      0 S   0.3  0.0   0:00.29 ksoftirqd/1
 1249 root      20   0   44560   3816   3192 R   0.3  0.0   0:00.39 top
 1787 daemon    20   0    8188   1380    556 R   0.3  0.0   0:00.01 stress
 1788 root      20   0       0      0      0 I   0.3  0.0   0:00.61 kworker/u8:1
 2403 root      20   0 1330816  41864  24340 S   0.3  0.5  35:01.00 containerd
 2458 root      20   0  336312  48068  40672 S   0.3  0.6   0:01.36 php-fpm
    1 root      20   0  225128   9088   6912 S   0.0  0.1   0:07.82 systemd
    2 root      20   0       0      0      0 S   0.0  0.0   0:00.08 kthreadd
    4 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 kworker/0:0H
    6 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 mm_percpu_wq
    7 root      20   0       0      0      0 S   0.0  0.0   0:00.37 ksoftirqd/0
    8 root      20   0       0      0      0 I   0.0  0.0   0:06.88 rcu_sched
    9 root      20   0       0      0      0 I   0.0  0.0   0:00.00 rcu_bh
   11 root      rt   0       0      0      0 S   0.0  0.0   0:01.50 watchdog/0
```

nginx 的測試結果

```
root@ip-172-31-12-157:~# docker run --name nginx -p 10000:80 -itd nginx
a199781e9b016ac044bef689f5b0f197ef08aefed3dec1c42128bb40e7c9733a
root@ip-172-31-12-157:~# ab -c 100 -n 1000 http://172.31.12.157:10000/ | grep "Requests"
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests
Requests per second:    14540.80 [#/sec] (mean)
```

```
top - 14:15:38 up 6 min,  2 users,  load average: 1.54, 0.48, 0.20
Tasks: 122 total,   4 running,  73 sleeping,   0 stopped,   0 zombie
%Cpu0  : 17.6 us, 43.4 sy,  0.0 ni, 23.4 id,  0.0 wa,  0.0 hi, 15.3 si,  0.3 st
%Cpu1  : 17.2 us, 49.7 sy,  0.0 ni, 12.8 id,  0.0 wa,  0.0 hi, 20.3 si,  0.0 st
%Cpu2  : 15.6 us, 40.3 sy,  0.0 ni, 34.7 id,  0.0 wa,  0.0 hi,  9.4 si,  0.0 st
%Cpu3  : 17.2 us, 47.2 sy,  0.0 ni, 21.0 id,  0.0 wa,  0.0 hi, 14.5 si,  0.0 st
KiB Mem :  7807728 total,  6208536 free,   276276 used,  1322916 buff/cache
KiB Swap:        0 total,        0 free,        0 used.  7272356 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
5095	**systemd+**	**20**	**0**	**11088**	**2604**	**1468**	**R**	**99.0**	**0.0**	**0:32.17**	**nginx**
5214	**root**	**20**	**0**	**64164**	**19680**	**4976**	**R**	**98.3**	**0.3**	**0:25.52**	**ab**
4532	root	20	0	1278876	102524	48452	S	55.7	1.3	0:20.88	dockerd
5029	root	20	0	11828	6592	4328	S	54.3	0.1	0:17.61	containerd-shim
217	root	20	0	0	0	0	I	7.3	0.0	0:01.05	kworker/u8:3
221	**root**	**20**	**0**	**0**	**0**	**0**	**R**	**7.0**	**0.0**	**0:01.83**	**kworker/u8:4**
7	root	20	0	0	0	0	S	0.7	0.0	0:00.22	ksoftirqd/0
16	root	20	0	0	0	0	S	0.7	0.0	0:00.24	ksoftirqd/1
28	root	20	0	0	0	0	S	0.7	0.0	0:00.20	ksoftirqd/3
8	root	20	0	0	0	0	I	0.3	0.0	0:00.06	rcu_sched
22	root	20	0	0	0	0	S	0.3	0.0	0:00.17	ksoftirqd/2
2420	root	20	0	1034828	42552	24384	S	0.3	0.5	0:00.71	containerd
1	root	20	0	225404	9116	6636	S	0.0	0.1	0:02.71	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H
5	root	20	0	0	0	0	I	0.0	0.0	0:01.84	kworker/u8:0
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
9	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_bh
10	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
11	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/0
12	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
13	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/1
14	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/1
15	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/1
18	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/1:0H

問題點

- 明明用戶 CPU 使用率都 70% 了，可我們挨個分析了一遍進程列表，還是找不到高 CPU 使用率的進程。看來 top 是不管用了，那還有其它的工具可以查看進程 CPU 使用情況嗎？不知道你記不記得我們的老朋友 pidstat，它可以用來分析進程的 CPU 使用情況。

Average:	UID	PID	%usr	%system	%guest	%wait	%CPU	CPU	Command
Average:	0	8	0.00	0.12	0.00	0.00	0.12	-	rcu_sched
Average:	0	10	0.00	0.12	0.00	0.00	0.12	-	migration/0
Average:	0	27	0.00	0.12	0.00	0.00	0.12	-	migration/3
Average:	0	632	0.00	0.12	0.00	0.00	0.12	-	kworker/2:2
Average:	0	1739	0.00	0.12	0.00	0.00	0.12	-	kworker/u8:2
Average:	0	1788	0.00	0.12	0.00	0.12	0.12	-	kworker/u8:1
Average:	0	2205	0.37	0.87	0.00	0.00	1.25	-	containerd-shim
Average:	101	2269	1.12	2.75	0.00	0.87	3.87	-	nginx
Average:	101	2270	0.12	0.50	0.00	0.25	0.62	-	nginx
Average:	0	2403	0.12	0.12	0.00	0.00	0.25	-	containerd
Average:	0	2458	0.25	0.00	0.00	4.12	0.25	-	php-fpm
Average:	0	3578	0.37	2.25	0.00	0.25	2.62	-	ab
Average:	1	3580	0.87	3.75	0.00	0.87	4.62	-	php-fpm
Average:	1	3581	0.62	3.87	0.00	1.25	4.49	-	php-fpm
Average:	1	3582	0.50	4.24	0.00	1.25	4.74	-	php-fpm
Average:	1	3583	0.50	4.12	0.00	1.12	4.62	-	php-fpm
Average:	1	3586	0.62	4.00	0.00	1.00	4.62	-	php-fpm
Average:	0	4501	0.87	0.50	0.00	0.00	1.37	-	dockerd
Average:	1000	5840	0.12	0.25	0.00	0.12	0.37	-	sshd
Average:	0	16097	0.25	0.50	0.00	0.12	0.75	-	pidstat
Average:	1	21351	0.12	0.00	0.00	0.00	0.12	-	stress
Average:	1	21356	0.12	0.00	0.00	0.00	0.12	-	stress
Average:	0	29334	0.00	0.12	0.00	0.00	0.12	-	kworker/3:0

疑點 1/11

- 所有進程的 CPU 使用率也都不高啊，最高的 Docker 和 Nginx 也只有 4% 和 3%，即使所有進程的 CPU 使用率都加起來，也不過 21%，離 80% 還差的遠呢！

- 很可能是因為前面的分析漏了一些關鍵訊息。你可以先暫停一下，自己往上翻，重新操作檢查一遍。或者，我們一起返回去分析 top 的輸出，看看能不能有新發現。

```
top - 12:08:52 up 13 days,  5:09,  2 users,  load average: 4.32, 4.49, 3.68
Tasks: 150 total,   6 running,  96 sleeping,   0 stopped,   0 zombie
%Cpu0  : 72.0 us, 15.9 sy,  0.0 ni, 11.1 id,  0.0 wa,  0.0 hi,  1.0 si,  0.0 st
%Cpu1  : 72.1 us, 16.5 sy,  0.0 ni, 10.8 id,  0.0 wa,  0.0 hi,  0.7 si,  0.0 st
%Cpu2  : 73.3 us, 16.6 sy,  0.0 ni,  9.8 id,  0.0 wa,  0.0 hi,  0.3 si,  0.0 st
%Cpu3  : 72.2 us, 16.4 sy,  0.0 ni, 11.0 id,  0.0 wa,  0.0 hi,  0.3 si,  0.0 st
KiB Mem :  7807720 total,  4375204 free,   346052 used,  3086464 buff/cache
KiB Swap:        0 total,        0 free,        0 used.  7156760 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
9086	daemon	20	0	336696	16616	8940	S	4.7	0.2	0:24.79	php-fpm
22353	daemon	20	0	336696	16616	8940	S	4.7	0.2	0:46.41	php-fpm
22354	daemon	20	0	336696	16616	8940	S	4.7	0.2	0:46.28	php-fpm
22355	daemon	20	0	336696	16616	8940	S	4.3	0.2	0:46.45	php-fpm
22356	daemon	20	0	336696	16552	8876	S	4.3	0.2	0:45.98	php-fpm
2269	systemd+	20	0	33960	4688	2440	S	3.7	0.1	1:00.22	nginx
3876	root	20	0	36040	6800	5040	S	2.7	0.1	0:00.27	ab
4501	root	20	0	1352224	105392	49352	S	1.3	1.3	4:52.90	dockerd
2205	root	20	0	10772	4416	3944	S	1.0	0.1	0:21.94	containerd-shim
2270	systemd+	20	0	33328	3960	2440	S	1.0	0.1	0:10.87	nginx
2458	root	20	0	336312	48068	40672	S	0.7	0.6	0:04.39	php-fpm
16	root	20	0	0	0	0	S	0.3	0.0	0:00.75	ksoftirqd/1
1739	root	20	0	0	0	0	I	0.3	0.0	0:01.57	kworker/u8:2
7755	root	20	0	44560	3832	3192	R	0.3	0.0	0:00.03	top
10862	daemon	20	0	8192	1380	556	R	0.3	0.0	0:00.01	stress
29334	root	20	0	0	0	0	I	0.3	0.0	0:01.52	kworker/3:0
1	root	20	0	225128	9088	6912	S	0.0	0.1	0:07.83	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.08	kthreadd
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
7	root	20	0	0	0	0	S	0.0	0.0	0:00.84	ksoftirqd/0
8	root	20	0	0	0	0	I	0.0	0.0	0:07.92	rcu_sched
9	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_bh
10	root	rt	0	0	0	0	S	0.0	0.0	0:00.78	migration/0
11	root	rt	0	0	0	0	S	0.0	0.0	0:01.50	watchdog/0

```
top - 07:27:24 up  2:20,  3 users,  load average: 3.90, 1.39, 0.98
Tasks: 149 total,   5 running,  95 sleeping,   0 stopped,   1 zombie
%Cpu0  : 69.7 us, 17.7 sy,  0.0 ni, 11.7 id,  0.0 wa,  0.0 hi,  1.0 si,  0.0 st
%Cpu1  : 70.6 us, 16.9 sy,  0.0 ni, 12.2 id,  0.0 wa,  0.0 hi,  0.3 si,  0.0 st
%Cpu2  : 72.1 us, 15.9 sy,  0.0 ni, 12.1 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu3  : 74.6 us, 13.9 sy,  0.0 ni, 11.2 id,  0.0 wa,  0.0 hi,  0.3 si,  0.0 st
KiB Mem :  7807724 total,  5098028 free,   332884 used,  2376812 buff/cache
KiB Swap:        0 total,        0 free,        0 used.  7167860 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
30584	daemon	20	0	336696	16592	8912	S	4.7	0.2	0:04.15	php-fpm
30585	daemon	20	0	336696	16528	8848	S	4.7	0.2	0:04.15	php-fpm
30583	daemon	20	0	336696	16528	8848	S	4.3	0.2	0:04.16	php-fpm
30588	daemon	20	0	336696	16592	8912	S	4.3	0.2	0:04.10	php-fpm
30590	daemon	20	0	336696	16528	8848	S	4.3	0.2	0:04.09	php-fpm
810	daemon	20	0	28332	2348	2140	S	0.0	0.0	0:00.00	atd
30247	daemon	20	0	4292	720	648	S	0.0	0.0	0:00.00	sh
30249	**daemon**	**20**	**0**	**4292**	**752**	**680**	**R**	**0.0**	**0.0**	**0:00.00**	**sh**
30251	daemon	20	0	7284	896	816	S	0.0	0.0	0:00.00	stress
30252	daemon	20	0	4292	752	676	S	0.0	0.0	0:00.00	sh
30253	**daemon**	**20**	**0**	**8184**	**1316**	**492**	**R**	**0.0**	**0.0**	**0:00.00**	**stress**
30254	daemon	20	0	0	0	0	Z	0.0	0.0	0:00.00	stress
30256	daemon	20	0	7284	916	832	S	0.0	0.0	0:00.00	stress
30257	**daemon**	**20**	**0**	**8188**	**104**	**0**	**R**	**0.0**	**0.0**	**0:00.00**	**stress**
30258	daemon	20	0	4292	792	716	S	0.0	0.0	0:00.00	sh
30259	daemon	20	0	7284	884	800	S	0.0	0.0	0:00.00	stress
30260	**daemon**	**20**	**0**	**8188**	**1376**	**556**	**R**	**0.0**	**0.0**	**0:00.00**	**stress**

疑點 2/11

- 再仔細看進程列表，這次主要看 Running（R） 狀態的進程。你有沒有發現，Nginx 和所有的 php-fpm 都處于 Sleep（S）狀態，而真正處於而 Running（R）狀態的，確是幾個 stress 進程。這幾個 stress 進程就比較奇怪了，需要我們做進一步的分析。

```
[root@ip-172-31-3-86:~# pidstat -p 30257
Linux 4.15.0-1065-aws (ip-172-31-3-86)  05/02/20        _x86_64_        (4 CPU)

07:28:38      UID       PID    %usr %system  %guest   %wait    %CPU   CPU  Command
[root@ip-172-31-3-86:~# ps aux | grep 30257
root     22736  0.0  0.0  14856  1044 pts/3    S+   07:28   0:00 grep --color=auto 30257
```

疑點 3/11

- 居然沒有任何輸出。難到是 pidstat 命令出了問題嗎？之前我說過，在懷疑性能工具出問題之前，最好是先用其它工具交叉確認一下。那用什麼工具呢？ ps 應該是最簡單易用的。我們在終端裡運行下面的指令，看看 30257 進程的狀態，還是沒有輸出。

- 現在終於發現問題，原來這個進程已經不存在了。所以 pidstat 就沒有任何輸出。既然進程都沒了，那性能問題應該也跟著沒了吧。我們再用 top 指令確認一下。

```
top - 07:31:13 up  2:24,  4 users,  load average: 3.99, 2.87, 1.68
Tasks: 159 total,   6 running, 105 sleeping,   0 stopped,   0 zombie
%Cpu0  : 70.7 us, 17.7 sy,  0.0 ni, 11.6 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0
%Cpu1  : 73.2 us, 14.9 sy,  0.0 ni, 11.2 id,  0.0 wa,  0.0 hi,  0.7 si,  0.0
%Cpu2  : 71.7 us, 16.8 sy,  0.0 ni, 11.4 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0
%Cpu3  : 73.8 us, 14.6 sy,  0.0 ni, 11.3 id,  0.0 wa,  0.0 hi,  0.3 si,  0.0
KiB Mem :  7807724 total,  5053052 free,   340192 used,  2414480 buff/cache
KiB Swap:        0 total,        0 free,        0 used.  7160544 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
30583	daemon	20	0	336696	16592	8912	S	4.3	0.2	0:14.06	php-fpm
30584	daemon	20	0	336696	16592	8912	S	4.3	0.2	0:14.03	php-fpm
30588	daemon	20	0	336696	16592	8912	S	4.3	0.2	0:13.94	php-fpm
30590	daemon	20	0	336696	16528	8848	S	4.3	0.2	0:13.92	php-fpm
30585	daemon	20	0	336696	16528	8848	S	4.0	0.2	0:14.00	php-fpm
21125	**daemon**	**20**	**0**	**8188**	**1380**	**556**	**R**	**0.3**	**0.0**	**0:00.01**	**stress**
810	daemon	20	0	28332	2348	2140	S	0.0	0.0	0:00.00	atd
21119	daemon	20	0	4292	756	684	S	0.0	0.0	0:00.00	sh
21121	daemon	20	0	7284	940	856	S	0.0	0.0	0:00.00	stress
21126	daemon	20	0	4292	800	728	S	0.0	0.0	0:00.00	sh
21127	daemon	20	0	7284	884	800	S	0.0	0.0	0:00.00	stress
21128	daemon	20	0	4292	756	684	S	0.0	0.0	0:00.00	sh
21129	daemon	20	0	4292	720	644	S	0.0	0.0	0:00.00	sh
21130	daemon	20	0	7284	884	800	S	0.0	0.0	0:00.00	stress
21131	**daemon**	**20**	**0**	**8192**	**1096**	**540**	**R**	**0.0**	**0.0**	**0:00.00**	**stress**
21132	**daemon**	**20**	**0**	**8188**	**1112**	**556**	**R**	**0.0**	**0.0**	**0:00.00**	**stress**
21133	daemon	20	0	7284	880	800	S	0.0	0.0	0:00.00	stress
21134	**daemon**	**20**	**0**	**7284**	**100**	**0**	**R**	**0.0**	**0.0**	**0:00.00**	**stress**
21135	daemon	20	0	4292	704	632	S	0.0	0.0	0:00.00	sh
21136	daemon	20	0	7284	884	800	S	0.0	0.0	0:00.00	stress
21137	**daemon**	**20**	**0**	**8188**	**100**	**0**	**R**	**0.0**	**0.0**	**0:00.00**	**stress**

疑點 4/11

- 好像又錯了。結果還是跟原來一樣，用戶 CPU 使用率還是高達 73%，系統 CPU 接近 17%，而空閒 CPU 只有 11%，Running 狀態的進程有 nginx、stress 等。

- 可是，剛剛我們看到 stress 進程不存在了，怎麼現在還在運行呢？再細看一下 top 的輸出，原來，這次 stress 進程的 PID 跟前面不一樣了，原來的 PID 30257 不見了。

- 進程的 PID 在改變，這說明什麼呢？在我看來，要麼是這個進程在不斷地重啟，要麼就是全新的進程。

疑點 5/11

- 第一個原因，進程在不斷地崩潰重啟，比如因為設定錯誤等等，這時進程在退出後可能又被監控系統自動重啟了。

- 第二個原因，這些進程都是短時進程，也就是在其它應用內部通過 exec 調用的外部指令。這些指令一般都只運行很短的時間就會結束，很難用 top 這種間隔時間比較長的工具發現。

疑點 6/11

- 至于 stress，我們前面提到過，它是一個常用的壓力測試工具。它的 PID 在不斷的變化中，看起來像是被其它進程調用的短時進程。想要繼續分析下去，還得找到它的父進程。

```
root@ip-172-31-12-157:~# pstree | grep stress
        |            |-containerd-shim-+-php-fpm---5*[php-fpm---sh---stress---stress]
```

疑點 7/11

- 從這裡可以看到， stress 是被 php-fpm 調用的子進程，並且進程數量不止一個。找到父進程後，我們能進入 app 的內部分析了。首先，當然應該去看看它的程式碼。

- 運行下面的指令，把案例應用的程式碼 copy 到 app 目錄，然後再執行 grep 查找是不是有程式碼在調用 stress 指令。

```
root@ip-172-31-12-157:~# docker cp phpfpm:/app .
root@ip-172-31-12-157:~# ls
app  snap
root@ip-172-31-12-157:~# grep stress -r app
app/index.php:// fake I/O with stress (via write()/unlink()).
app/index.php:$result = exec("/usr/local/bin/stress -t 1 -d 1 2>&1", $output, $status);
```

```
[root@ip-172-31-12-157:~# cat app/index.php
<?php
// fake I/O with stress (via write()/unlink()).
$result = exec("/usr/local/bin/stress -t 1 -d 1 2>&1", $output, $status);
if (isset($_GET["verbose"]) && $_GET["verbose"]==1 && $status != 0) {
  echo "Server internal error: ";
  print_r($output);
} else {
  echo "It works!";
}

[?>root@ip-172-31-12-157:~#
```

疑點 8/11

- 程式碼裡對每個請求都會去調用一個 stress 指令，模擬 I/O 壓力。從注釋上看，stress 會通過 write() 和 unlink() 對 I/O 進程進行壓測，看來，這些應該就是系統 CPU 使用率升高的根源了。

- 不過，stress 模擬的是 I/O 壓力，而之前在 top 的輸出中看到的，卻一´直是用戶 CPU 和系統 CPU 升高，並沒見到 iowait 升高。這又是怎麼回事呢？ stress 到底是不是 CPU 使用率升高的原因呢？

- 我們還得繼續往下走。從程式碼中可以看到，給請求加入 verbose=1 參數後，就可以查看 stress 的輸出。

```
Server internal error: Array
(
    [0] => stress: info: [2585] dispatching hogs: 0 cpu, 0 io, 0 vm, 1 hdd
    [1] => stress: FAIL: [2586] (563) mkstemp failed: Permission denied
    [2] => stress: FAIL: [2585] (394) <-- worker 2586 returned error 1
    [3] => stress: WARN: [2585] (396) now reaping child worker processes
    [4] => stress: FAIL: [2585] (400) kill error: No such process
    [5] => stress: FAIL: [2585] (451) failed run completed in 0s
)
```

疑點 9/11

- 看錯誤訊息 mkstemp failed: Permission denied，以及 failed run completed in 0s。原來 stress 指令並沒有成功，它因為權限問題失敗而退出了。看來，我們發現了一個 PHP 調用外部 stress 指令的 bug：沒有權限創建臨時檔案。

- 從這里我們可以猜測，正是由于權限錯誤，大量的 stress 進程在啟動時初始化失敗，進而導致用戶 CPU 使用率的升高。

疑點 10/11

- 分析出問題來源，下一步是不是要開始優化了呢？當然不是！既然只是猜測，那就需要再確認一下，這個猜測到底對不對，是不是真的有大量的 stress 進程。該用什麼工具或指標呢？

疑點 11/11

- 我們前面已經用了 top、pidstat、pstree 等工具，沒有發現大量的 stress 進程。那麼，還有什麼其它的工具可以使用嗎？

- 還記得上一期提到的 perf 嗎？它可以用來分析 CPU 性能事件，用在這裡就很合適。依舊在第一個終端中運行 perf record -g 指令，並等待一會兒（比如15秒）後按 Ctrl+C 退出。然後再運行 perf report 查看報告。


Processing events... [12M/23M]
Failed to open /lib/x86_64-linux-gnu/libm-2.24.so, continuing without symbols

Samples: 200K of event 'cpu-clock', Event count (approx.): 50242000000

Children	Self	Command	Shared Object	Symbol
+ 71.28%	0.00%	stress	stress	[.] 0x000000000000168d
+ 10.56%	0.00%	swapper	[kernel.kallsyms]	[k] secondary_startup_64
+ 10.56%	0.00%	swapper	[kernel.kallsyms]	[k] cpu_startup_entry
+ 10.56%	0.00%	swapper	[kernel.kallsyms]	[k] do_idle
+ 10.45%	0.00%	swapper	[kernel.kallsyms]	[k] arch_cpu_idle
+ 10.45%	0.00%	swapper	[kernel.kallsyms]	[k] default_idle_call
+ 10.45%	10.41%	swapper	[kernel.kallsyms]	[k] native_safe_halt
+ 10.45%	0.00%	swapper	[kernel.kallsyms]	[k] __cpuidle_text_start
+ 7.89%	0.00%	swapper	[kernel.kallsyms]	[k] start_secondary
+ 4.75%	0.00%	php-fpm	[kernel.kallsyms]	[k] entry_SYSCALL_64_after_hwframe
+ 4.75%	0.05%	php-fpm	[kernel.kallsyms]	[k] do_syscall_64
+ 4.08%	4.06%	stress	stress	[.] 0x0000000000002f1b
+ 4.05%	4.02%	stress	libc-2.24.so	[.] 0x0000000000036387
+ 3.51%	0.00%	php-fpm	[unknown]	[.] 0x6cb6258d4c544155
+ 3.51%	0.00%	php-fpm	libc-2.24.so	[.] 0x00000000000202e1
+ 3.35%	0.00%	stress	[kernel.kallsyms]	[k] async_page_fault
+ 3.35%	0.00%	stress	[kernel.kallsyms]	[k] do_async_page_fault
+ 3.35%	0.00%	stress	[kernel.kallsyms]	[k] do_page_fault
+ 3.34%	0.97%	stress	[kernel.kallsyms]	[k] __do_page_fault
+ 3.26%	3.25%	stress	stress	[.] 0x0000000000002eff
+ 3.18%	3.16%	stress	stress	[.] 0x0000000000002f29
+ 3.04%	0.83%	stress	stress	[.] 0x0000000000002f25
+ 3.01%	3.00%	stress	stress	[.] 0x0000000000002f0d
+ 2.99%	2.97%	stress	libc-2.24.so	[.] 0x000000000003651b
+ 2.96%	2.94%	stress	stress	[.] 0x0000000000002f09
+ 2.68%	0.00%	php-fpm	[kernel.kallsyms]	[k] async_page_fault
+ 2.68%	0.00%	php-fpm	[kernel.kallsyms]	[k] do_async_page_fault
+ 2.68%	0.00%	php-fpm	[kernel.kallsyms]	[k] do_page_fault
+ 2.68%	0.15%	php-fpm	[kernel.kallsyms]	[k] __do_page_fault
+ 2.67%	0.00%	swapper	[kernel.kallsyms]	[k] x86_64_start_kernel
+ 2.67%	0.00%	swapper	[kernel.kallsyms]	[k] x86_64_start_reservations
+ 2.67%	0.00%	swapper	[kernel.kallsyms]	[k] start_kernel

```
Samples: 200K of event 'cpu-clock', Event count (approx.): 50242000000
  Children      Self  Command          Shared Object              Symbol
-   71.28%     0.00%  stress           stress                     [.] 0x000000000000168d
   - 0x168d
        4.08% 0x2f1b
        4.05% 0x36387
        3.26% 0x2eff
        3.18% 0x2f29
      + 3.04% 0x2f25
        3.01% 0x2f0d
        2.99% 0x3651b
        2.96% 0x2f09
        2.66% 0x2ee5
        2.64% 0x2ef6
        2.28% 0x36512
        2.09% 0x36514
        2.04% 0x36538
        1.99% 0x3652e
        1.85% 0x36829
        1.80% 0x36360
        1.80% 0x364fd
        1.79% 0x36820
        1.78% 0x364f0
        1.71% 0x3636b
        1.70% 0xe80
        1.70% 0x363c8
        1.70% 0x363ce
        1.65% 0x2ef3
        1.65% 0x36522
        1.23% 0x3650c
        1.15% 0x363ea
        1.02% 0x2f18
        1.01% 0x2ef9
        0.72% 0x36508
```

```
Samples: 281K of event 'cpu-clock', Event count (approx.): 70334750000
  Children      Self  Command          Shared Object                      Symbol
-   71.41%     0.00%  stress           stress                             [.] 0x000000000000168d
   - 0x168d
        21.60% random_r
        13.89% random
        4.07% 0x2f1b
        4.02% rand
        3.31% 0x2eff
        3.23% 0x2f29
      + 3.04% 0x2f25
        3.01% 0x2f0d
        2.93% 0x2f09
        2.65% 0x2ef6
        2.65% 0x2ee5
        1.70% 0xe80
        1.62% 0x2ef3
        1.04% 0x2f18
        1.02% 0x2ef9
+   21.60%    21.50%  stress           libc-2.24.so                       [.] random_r
+   13.89%    13.81%  stress           libc-2.24.so                       [.] random
+   10.57%     0.00%  swapper          [kernel.kallsyms]                  [k] secondary_startup_64
+   10.57%     0.00%  swapper          [kernel.kallsyms]                  [k] cpu_startup_entry
+   10.57%     0.00%  swapper          [kernel.kallsyms]                  [k] do_idle
+   10.46%     0.00%  swapper          [kernel.kallsyms]                  [k] default_idle_call
+   10.46%     0.00%  swapper          [kernel.kallsyms]                  [k] arch_cpu_idle
+   10.46%    10.43%  swapper          [kernel.kallsyms]                  [k] native_safe_halt
+   10.46%     0.00%  swapper          [kernel.kallsyms]                  [k] __cpuidle_text_start
+    7.97%     0.00%  swapper          [kernel.kallsyms]                  [k] start_secondary
+    4.75%     0.00%  php-fpm          [kernel.kallsyms]                  [k] entry_SYSCALL_64_after_hwframe
+    4.74%     0.05%  php-fpm          [kernel.kallsyms]                  [k] do_syscall_64
+    4.07%     4.05%  stress           stress                             [.] 0x0000000000002f1b
+    4.02%     3.98%  stress           libc-2.24.so                       [.] rand
+    3.52%     0.00%  php-fpm          [unknown]                          [.] 0x6cb6258d4c544155
```

思路 1/2

- stress 佔了所有 CPU 時鐘事件的 77%，而 stress 調用調用棧中比例最高的，是隨機數生成函數 random()，看來它的確就是 CPU 使用率升高的元凶了。隨後的優化就很簡單了，只要修復權限問題，並減少或刪除 stress 的調用，就可以減輕系統的 CPU 壓力。

- 實際生產環境中的問題一般都要比這個案例複雜，在你找到觸發瓶頸的指令後，卻可能發現，這個外部指令的調用過程是應用核心邏輯的一部份，並不能輕易減少或者刪除。

思路 2/2

- 這時，你就得繼續排查，為什麼被調用的指令，會導致 CPU 使用率升高或 I/O 升高等問題。這些複雜場景的案例，會在後面的綜合實戰裡詳細分析。

原因 1/2

- 在這個案例中，我們使用了 top、pidstat、pstree 等工具分析了系統 CPU 使用率高的問題，並發現 CPU 升高是短時進程 stress 導致的，但是整個分析過程還是比較複雜的。對於這類問題，有沒有更好的方法監控呢？

- execsnoop 就是一個專為短時進程設計的工具。它通過 ftrace 實時監控進程的 exec() 行為，並輸出短時程的基本訊息，包括進程 PID、父進程 PID、指令參數以及執行的結果。

原因 2/2

- 用 execsnoop 監控上述案例，就可以直接得到 stress 進程的父進程 PID 以及它的指令參數，並可以發現大量的 stress 進程不斷啟動。

```
6669   6666 /usr/local/bin/stress -t 1 -d 1
6668   8004          php-fpm-6668  [001] .... 1144204.088141: execsnoop_sys_execve: (SyS_execve+0x0/0x40)
6671   6668 /usr/local/bin/stress -t 1 -d 1
6673   8003          php-fpm-6673  [003] .... 1144204.091739: execsnoop_sys_execve: (SyS_execve+0x0/0x40)
6674   6673 /usr/local/bin/stress -t 1 -d 1
6675   8002          php-fpm-6675  [002] .... 1144204.094366: execsnoop_sys_execve: (SyS_execve+0x0/0x40)
6677   6675 /usr/local/bin/stress -t 1 -d 1
6679   8005          php-fpm-6679  [001] .n.. 1144204.104461: execsnoop_sys_execve: (SyS_execve+0x0/0x40)
6681   6679 /usr/local/bin/stress -t 1 -d 1
6680   8006          php-fpm-6680  [001] .... 1144204.106489: execsnoop_sys_execve: (SyS_execve+0x0/0x40)
6682   6680 /usr/local/bin/stress -t 1 -d 1
6684   8004          php-fpm-6684  [003] .... 1144204.108154: execsnoop_sys_execve: (SyS_execve+0x0/0x40)
6686   6684 /usr/local/bin/stress -t 1 -d 1
6688   8003          php-fpm-6688  [002] .... 1144204.111234: execsnoop_sys_execve: (SyS_execve+0x0/0x40)
6689   6688 /usr/local/bin/stress -t 1 -d 1
6691   8002          php-fpm-6691  [000] .... 1144204.121955: execsnoop_sys_execve: (SyS_execve+0x0/0x40)
6692   6691 /usr/local/bin/stress -t 1 -d 1
6694   8005          php-fpm-6694  [003] .... 1144204.127990: execsnoop_sys_execve: (SyS_execve+0x0/0x40)
6696   6694 /usr/local/bin/stress -t 1 -d 1
6693   8006          php-fpm-6693  [002] .... 1144204.131334: execsnoop_sys_execve: (SyS_execve+0x0/0x40)
6698   6693 /usr/local/bin/stress -t 1 -d 1
6699   8004          php-fpm-6699  [002] .... 1144204.132732: execsnoop_sys_execve: (SyS_execve+0x0/0x40)
6701   6699 /usr/local/bin/stress -t 1 -d 1
6703   8003          php-fpm-6703  [002] .... 1144204.134474: execsnoop_sys_execve: (SyS_execve+0x0/0x40)
6704   6703 /usr/local/bin/stress -t 1 -d 1
6706   8002          php-fpm-6706  [001] .... 1144204.145688: execsnoop_sys_execve: (SyS_execve+0x0/0x40)
6707   6706 /usr/local/bin/stress -t 1 -d 1
6709   8006          php-fpm-6709  [002] .... 1144204.153680: execsnoop_sys_execve: (SyS_execve+0x0/0x40)
6711   6709 /usr/local/bin/stress -t 1 -d 1
6710   8004          php-fpm-6710  [002] .... 1144204.155109: execsnoop_sys_execve: (SyS_execve+0x0/0x40)
6713   6710 /usr/local/bin/stress -t 1 -d 1
6714   8005          php-fpm-6714  [002] .... 1144204.156543: execsnoop_sys_execve: (SyS_execve+0x0/0x40)
```

```
Tasks: 148 total,   3 running,  91 sleeping,   0 stopped,   0 zombie
%Cpu0  : 70.9 us, 17.6 sy,  0.0 ni, 11.1 id,  0.0 wa,  0.0 hi,  0.3 si,  0.0 st
%Cpu1  : 69.8 us, 17.9 sy,  0.0 ni, 11.6 id,  0.0 wa,  0.0 hi,  0.7 si,  0.0 st
%Cpu2  : 73.0 us, 16.3 sy,  0.0 ni, 10.3 id,  0.0 wa,  0.0 hi,  0.3 si,  0.0 st
%Cpu3  : 70.5 us, 16.3 sy,  0.0 ni, 12.5 id,  0.0 wa,  0.0 hi,  0.7 si,  0.0 st
KiB Mem :  7807720 total,  1012648 free,   342404 used,  6452668 buff/cache
KiB Swap:        0 total,        0 free,        0 used.  7160692 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
19638 daemon    20   0  336696  16056   8376 S   4.7  0.2   0:00.74 php-fpm: pool www
19640 daemon    20   0  336696  16456   8776 S   4.7  0.2   0:00.73 php-fpm: pool www
19641 daemon    20   0  336696  16456   8776 S   4.7  0.2   0:00.72 php-fpm: pool www
19637 daemon    20   0  336696  17564   9880 S   4.3  0.2   0:00.73 php-fpm: pool www
19639 daemon    20   0  336696  16124   8444 S   4.3  0.2   0:00.72 php-fpm: pool www
  799 daemon    20   0   28332   2448   2240 S   0.0  0.0   0:00.00 /usr/sbin/atd -f
30297 daemon    20   0    4292    724    648 S   0.0  0.0   0:00.00 sh -c /usr/local/bin/stress -t 1 -d 1 2>&1
30299 daemon    20   0    7284    844    760 S   0.0  0.0   0:00.00 /usr/local/bin/stress -t 1 -d 1
30300 daemon    20   0    8188    788    492 R   0.0  0.0   0:00.00 /usr/local/bin/stress -t 1 -d 1
30301 daemon    20   0    4292    764    688 S   0.0  0.0   0:00.00 sh -c /usr/local/bin/stress -t 1 -d 1 2>&1
30302 daemon    20   0    7284    928    844 S   0.0  0.0   0:00.00 /usr/local/bin/stress -t 1 -d 1
30303 daemon    20   0    8188   1372    552 R   0.0  0.0   0:00.00 /usr/local/bin/stress -t 1 -d 1
30304 daemon    20   0    4292    708    632 S   0.0  0.0   0:00.00 sh -c /usr/local/bin/stress -t 1 -d 1 2>&1
```

小結

碰到常規問題無法解釋的 CPU 使率用率情況時，首先要想到有可能是短時間應用導致的問題，比如是下面這兩種情況：

1. 應用裡直接調用了其它二進制程式，這些程式通常運行時間比較短，通過 top 等工具也不容易發現。

2. 應用本身在不停的崩潰重啟，而啟動過程的資源初始化，很可能會佔用相當多的 CPU 。

思考題

- 之前所碰到的 CPU 性能問題，有沒有那些印象深刻的經歷可以分享呢？
- 在今天的案例操作中，你遇到了什麼問題，又解決了那些呢？你可以結合我的講述，總結自己的思路。
- 看過最高的 load average 是多少呢？

Start page | root@dellr720:/barajoun

```
top - 22:57:17 up 13 days,  5:00,  1 user,  load average: 172.56, 177.59, 136.02
Tasks: 556 total,   9 running, 526 sleeping,   0 stopped,  21 zombie
Cpu0  : 14.4%us, 45.9%sy,  0.0%ni,  0.0%id, 39.7%wa,  0.0%hi,  0.0%si,  0.0%st
Cpu1  : 17.7%us, 13.0%sy,  0.0%ni,  0.0%id,  4.7%wa,  0.0%hi, 64.7%si,  0.0%st
Cpu2  :  3.0%us, 51.5%sy,  0.0%ni,  0.0%id, 44.8%wa,  0.0%hi,  0.7%si,  0.0%st
Cpu3  : 17.7%us,  7.0%sy,  0.0%ni,  0.0%id,  2.0%wa,  0.0%hi, 73.3%si,  0.0%st
Mem:  65926268k total, 65576052k used,   350216k free,    45192k buffers
Swap:  8142840k total,   128216k used,  8014624k free, 57995116k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 3190 root      20   0  103m  56m 2580 R 50.5  0.1   5668:26 bandwidthd
 3192 root      20   0 57956 9140 2580 R 48.8  0.0   5839:10 bandwidthd
 3191 root      20   0 77264  25m 2580 R 47.5  0.0   5776:37 bandwidthd
 2524 root      20   0     0    0    0 D  4.0  0.0 191:12.00 nfsd
 2540 root      20   0     0    0    0 S  4.0  0.0 192:27.65 nfsd
 2517 root      20   0     0    0    0 D  3.6  0.0 194:12.37 nfsd
 2546 root      20   0     0    0    0 D  3.6  0.0 190:11.20 nfsd
 2581 root      20   0     0    0    0 R  3.6  0.0 193:05.78 nfsd
 2593 root      20   0     0    0    0 S  3.6  0.0 193:04.25 nfsd
 2506 root      20   0     0    0    0 S  3.3  0.0 192:12.87 nfsd
 2520 root      20   0     0    0    0 S  3.3  0.0 199:57.02 nfsd
 2569 root      20   0     0    0    0 R  3.3  0.0 194:18.51 nfsd
 2572 root      20   0     0    0    0 D  3.3  0.0 191:56.73 nfsd
 2580 root      20   0     0    0    0 D  3.3  0.0 193:09.95 nfsd
 2586 root      20   0     0    0    0 S  3.3  0.0 188:41.60 nfsd
 2590 root      20   0     0    0    0 S  3.3  0.0 190:28.17 nfsd
 2543 root      20   0     0    0    0 S  3.0  0.0 190:46.01 nfsd
```

```
top - 15:42:22 up 240 days, 12:00,  1 user,  load average: 5035.60, 2194.72, 925.32
Tasks: 163 total,   2 running, 161 sleeping,   0 stopped,   0 zombie
%Cpu(s): 92.8 us,  4.3 sy,  0.0 ni,  0.0 id,  0.8 wa,  0.0 hi,  2.0 si,  0.0 st
KiB Mem :  3805120 total,   224720 free,  3244060 used,   336340 buff/cache
KiB Swap:        0 total,        0 free,        0 used.   318332 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 9865 root      20   0  814540 579756   7104 S 182.4 15.2  14:58.19 kube-apiserver
10741 root      20   0 44.630g 177400   2656 D   6.0  4.7   0:19.78 dockerd
24168 root      20   0  461924 237724   7104 S   2.7  6.2  13:55.10 kube-controller
 2879 root      20   0 1040284  99120  10572 S   2.3  2.6  10190:26 kubelet
25144 root      20   0 10.358g 353572   2336 S   1.3  9.3  19:51.49 etcd
16886 root      20   0  117688   9932   6060 D   1.0  0.3   0:00.03 docker-containe
   27 root      39  19       0      0      0 R   0.3  0.0 119:25.84 khugepaged
  155 root      20   0       0      0      0 S   0.3  0.0   7:27.20 jbd2/nvme0n1p2-
10919 root      20   0   43784   7172      0 S   0.3  0.2   4:13.31 kube-proxy
16867 admin     20   0   42828   3616   2992 R   0.3  0.1   0:00.03 top
28217 root      20   0 1525852 245152      0 S   0.3  6.4  38:41.04 agent
    1 root      20   0   57420   5252   3424 S   0.0  0.1  43:05.56 systemd
    2 root      20   0       0      0      0 S   0.0  0.0   0:03.66 kthreadd
    3 root      20   0       0      0      0 S   0.0  0.0  25:27.38 ksoftirqd/0
    5 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/0:0H
    7 root      20   0       0      0      0 S   0.0  0.0  93:13.48 rcu_sched
    8 root      20   0       0      0      0 S   0.0  0.0   0:00.00 rcu_bh
    9 root      rt   0       0      0      0 S   0.0  0.0   4:34.10 migration/0
   10 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 lru-add-drain
   11 root      rt   0       0      0      0 S   0.0  0.0   0:21.15 watchdog/0
   12 root      20   0       0      0      0 S   0.0  0.0   0:00.00 cpuhp/0
   13 root      20   0       0      0      0 S   0.0  0.0   0:00.00 cpuhp/1
   14 root      rt   0       0      0      0 S   0.0  0.0   0:18.45 watchdog/1
   15 root      rt   0       0      0      0 S   0.0  0.0   4:34.90 migration/1
   16 root      20   0       0      0      0 S   0.0  0.0  25:13.63 ksoftirqd/1
   18 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/1:0H
   19 root      20   0       0      0      0 S   0.0  0.0   0:00.00 kdevtmpfs
   20 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 netns
```

END