
Subject: Re: Hang with fair cgroup scheduler (reproducer is attached.)
Posted by [KAMEZAWA Hiroyuki](#) on Fri, 14 Dec 2007 10:58:37 GMT
[View Forum Message](#) <> [Reply to Message](#)

Here is much easier test.
(I'm sorry I'll be absent tomorrow.)

the number of cpus is 8. ia64/NUMA.

The hang occurs when the number of tasks is not smaller than available cpus.
Can be a hint ?

==

```
[root@rhel51GA testpro]# cat yield.c
#include <sched.h>
```

```
int main()
{
    while (1)
        sched_yield();
}
```

```
[root@rhel51GA testpro]# cat batch-test.sh
#!/bin/bash -x
```

```
mount -t cgroup none /opt/cgroup -o cpu
mkdir /opt/cgroup/group_1
mkdir /opt/cgroup/group_2
```

```
./yield &
PIDA=$!
./yield &
PIDB=$!
```

```
while true; do
    echo $PIDA > /opt/cgroup/group_1/tasks
    echo $PIDB > /opt/cgroup/group_1/tasks
    echo $PIDA > /opt/cgroup/group_2/tasks;
    echo $PIDB > /opt/cgroup/group_2/tasks
done
```

```
[root@rhel51GA testpro]# ./batech-test.sh
no hang.
```

```
[root@rhel51GA testpro]# taskset 0f ./batech-test.sh
no hang
```

```
[root@rhel51GA testpro]#taskset 03 ./batech-test.sh
hang.
Pid: 8132, CPU 0, comm:          yield
psr : 00001210085a2010 ifs : 8000000000000206 ip : [<a000000100067c01>]   Not tainted
ip is at pick_next_task_fair+0x81/0xe0
unat: 0000000000000000 pfs : 0000000000000b1d rsc : 0000000000000003
rnat: 0000000000000000 bsp: 0000000000000000 pr : 000000000566959
ldrs: 0000000000000000 ccv : 0000000000000000 fpsr: 0009804c0270033f
csd : 0000000000000000 ssd : 0000000000000000
b0 : a0000001006f6ac0 b6 : a000000100076a60 b7 : a000000100067b80
f6 : 000000000000000000000000 f7 : 000000000000000000000000
f8 : 1003e00000000a0000007 f9 : 1003e0000004633b23e65
f10 : 1003ee04f68ea89dfb4c3 f11 : 1003e0000000000000002b
r1 : a000000100d87a60 r2 : e000000011082f0 r3 : 0000000000000060
r8 : 0000000000000000 r9 : e00000001108310 r10 : e000004080032018
r11 : 00000000f86ccc70 r12 : e00000408394fe10 r13 : e000004083940000
r14 : 0000000000000001 r15 : 0000000000000064 r16 : e000000011089f0
r17 : ffffffffrrrrrr r18 : e00000001108360 r19 : 0000000000000000
r20 : e00000408003ef10 r21 : 0000000001e9555b r22 : 000000af762794d4
r23 : 00000015e1abc70b r24 : ffffffffrrrrrr463 r25 : e00000408003ef10
r26 : 0000000000000002 r27 : e000000011082f0 r28 : e00000001108288
r29 : a0000001008a5468 r30 : a000000100076a60 r31 : a000000100b726e0
```

Call Trace:

```
[<a000000100013bc0>] show_stack+0x40/0xa0
      sp=e00000408394f860 bsp=e000004083940f18
[<a000000100014840>] show_regs+0x840/0x880
      sp=e00000408394fa30 bsp=e000004083940ec0
[<a000000100036fa0>] die+0x1a0/0x2a0
      sp=e00000408394fa30 bsp=e000004083940e78
[<a0000001000370f0>] die_if_kernel+0x50/0x80
      sp=e00000408394fa30 bsp=e000004083940e48
[<a000000100038260>] ia64_fault+0x1140/0x1260
      sp=e00000408394fa30 bsp=e000004083940de8
[<a00000010000ae20>] ia64_leave_kernel+0x0/0x270
      sp=e00000408394fc40 bsp=e000004083940de8
[<a000000100067c00>] pick_next_task_fair+0x80/0xe0
      sp=e00000408394fe10 bsp=e000004083940db8
[<a0000001006f6ac0>] schedule+0x940/0x1280
      sp=e00000408394fe10 bsp=e000004083940d08
[<a000000100074e20>] sys_sched_yield+0xe0/0x100
      sp=e00000408394fe30 bsp=e000004083940ca8
[<a00000010000aca0>] ia64_ret_from_syscall+0x0/0x20
      sp=e00000408394fe30 bsp=e000004083940ca8
[<a000000000010720>] __kernel_syscall_via_break+0x0/0x20
      sp=e000004083950000 bsp=e000004083940ca8
```

Thanks,
-Kame

Containers mailing list
Containers@lists.linux-foundation.org
<https://lists.linux-foundation.org/mailman/listinfo/containers>
