Subject: [PATCH] block layer: ioprio_best function fix
Posted by Anonymous Coward on Thu, 12 Oct 2006 12:13:30 GMT
View Forum Message <> Reply to Message

Currently ioprio_best function first checks wethere aioprio or bioprio equals
IOPRIO_CLASS_NONE (ioprio_valid() macros does that) and if it is so it returns
bioprio/aioprio appropriately. Thus the next four lines, that set aclass/bclass
to IOPRIO_CLASS_BE, if aclass/bclass == IOPRIO_CLASS_NONE, are never executed.

The second problem: if aioprio from class IOPRIO_CLASS_NONE and bioprio from
class IOPRIO_CLASS_IDLE are passed to ioprio_best function, it will return
IOPRIO_CLASS_IDLE. It means that during __make_request we can merge two
requests and set the priority of merged request to IDLE, while one of
the initial requests originates from a process with NONE (default) priority.
So we can get a situation when a process with default ioprio will experience
IO starvation, while there is no process from real-time class in the system.

Just removing ioprio_valid check should correct situation.

Signed-off-by: Vasily Tarasov <vtaras@openvz.org>

--

```
--- linux-2.6.18/fs/ioprio.c.orig 2006-09-20 07:42:06.000000000 +0400
+++ linux-2.6.18/fs/ioprio.c 2006-10-12 14:52:36.000000000 +0400
@@ -145,11 +145,6 @@ int ioprio_best(unsigned short aprio, un
  unsigned short aclass = IOPRIO_PRIO_CLASS(aprio);
  unsigned short bclass = IOPRIO_PRIO_CLASS(bprio);

- if (!ioprio_valid(aprio))
-  return bprio;
- if (!ioprio_valid(bprio))
-  return aprio;
-
  if (aclass == IOPRIO_CLASS_NONE)
   aclass = IOPRIO_CLASS_BE;
  if (bclass == IOPRIO_CLASS_NONE)
```

Subject: Re: [PATCH] block layer: ioprio_best function fix
Posted by Jens Axboe on Thu, 12 Oct 2006 13:09:07 GMT
View Forum Message <> Reply to Message

On Thu, Oct 12 2006, Vasily Tarasov wrote:
> Currently ioprio_best function first checks wethere aioprio or bioprio equals
> IOPRIO_CLASS_NONE (ioprio_valid() macros does that) and if it is so it returns
> bioprio/aioprio appropriately. Thus the next four lines, that set aclass/bclass
> to IOPRIO_CLASS_BE, if aclass/bclass == IOPRIO_CLASS_NONE, are never executed.

>
> The second problem: if aioprio from class IOPRIO_CLASS_NONE and bioprio from
> class IOPRIO_CLASS_IDLE are passed to ioprio_best function, it will return
> IOPRIO_CLASS_IDLE. It means that during __make_request we can merge two
> requests and set the priority of merged request to IDLE, while one of
> the initial requests originates from a process with NONE (default) priority.
> So we can get a situation when a process with default ioprio will experience
> IO starvation, while there is no process from real-time class in the system.
>
> Just removing ioprio_valid check should correct situation.

Analysis looks correct, thanks.

--
Jens Axboe
