Subject: [PATCH] cfq: wrong sync writes detection
Posted by Vasily Tarasov on Mon, 11 Dec 2006 15:45:19 GMT
View Forum Message <> Reply to Message

CFQ I/O scheduler does the following actions to
find out whether the request is sync:

rw = rq_data_dir(rq); => possible values for rw are 0 or 1

```
static inline pid_t cfq_queue_pid(struct task_struct *task, int rw)
{
        if (rw == READ || rw == WRITE_SYNC) => second condition is always false
                return task->pid;

        return CFQ_KEY_ASYNC;
}
```

The following patch fixes the bug by adding sync parameter,
wich is obtained through bio_sync macros.

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

--

```
--- ./block/cfq-iosched.c.syncwrite 2006-09-20 07:42:06.000000000 +0400
+++ ./block/cfq-iosched.c 2006-12-11 07:23:03.000000000 +0300
@@ -324,9 +324,9 @@ static int cfq_queue_empty(request_queue
  return !cfqd->busy_queues;
 }

-static inline pid_t cfq_queue_pid(struct task_struct *task, int rw)
+static inline pid_t cfq_queue_pid(struct task_struct *task, int rw, int sync)
 {
- if (rw == READ || rw == WRITE_SYNC)
+ if (rw == READ || sync)
    return task->pid;

  return CFQ_KEY_ASYNC;
@@ -621,7 +621,7 @@ static struct request *
 cfq_find_rq_fmerge(struct cfq_data *cfqd, struct bio *bio)
 {
  struct task_struct *tsk = current;
- pid_t key = cfq_queue_pid(tsk, bio_data_dir(bio));
+ pid_t key = cfq_queue_pid(tsk, bio_data_dir(bio), bio_sync(bio));
  struct cfq_queue *cfqq;
  struct rb_node *n;
  sector_t sector;
@@ -1958,7 +1958,8 @@ static int cfq_may_queue(request_queue_t
```

```
   * so just lookup a possibly existing queue, or return 'may queue'
   * if that fails
   */
- cfqq = cfq_find_cfq_hash(cfqd, cfq_queue_pid(tsk, rw), tsk->ioprio);
+ cfqq = cfq_find_cfq_hash(cfqd, cfq_queue_pid(tsk, rw,
+     bio_sync(bio)), tsk->ioprio);
  if (cfqq) {
   cfq_init_prio_data(cfqq);
   cfq_prio_boost(cfqq);
@@ -2020,7 +2021,7 @@ cfq_set_request(request_queue_t *q, stru
  struct task_struct *tsk = current;
  struct cfq_io_context *cic;
  const int rw = rq_data_dir(rq);
- pid_t key = cfq_queue_pid(tsk, rw);
+ pid_t key = cfq_queue_pid(tsk, rw, bio_sync(bio));
  struct cfq_queue *cfqq;
  struct cfq_rq *crq;
  unsigned long flags;
```

Subject: Re: [PATCH] cfq: wrong sync writes detection
Posted by Jens Axboe on Mon, 11 Dec 2006 16:07:11 GMT

View Forum Message <> Reply to Message

On Mon, Dec 11 2006, Vasily Tarasov wrote:
> CFQ I/O scheduler does the following actions to
> find out whether the request is sync:
>
> rw = rq_data_dir(rq); => possible values for rw are 0 or 1
>
> static inline pid_t cfq_queue_pid(struct task_struct *task, int rw)
> {
>   if (rw == READ || rw == WRITE_SYNC) => second condition is always false
>   return task->pid;
>
>   return CFQ_KEY_ASYNC;
> }
>
> The following patch fixes the bug by adding sync parameter,
> wich is obtained through bio_sync macros.

Good catch! Applied.

--
Jens Axboe
