

---

Subject: Re: [PATCH] cfq: get rid of cfqq hash  
Posted by [Jens Axboe](#) on Wed, 25 Apr 2007 10:27:49 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Wed, Apr 25 2007, Jens Axboe wrote:

> On Wed, Apr 25 2007, Vasily Tarasov wrote:

> > > @@ -1806,7 +1765,11 @@ 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, key, tsk->ioprio);

> > > + cic = cfq\_get\_io\_context\_noalloc(cfqd, tsk);

> > > + if (!cic)

> > > + return ELV\_MQUEUE\_MAY;

> > > +

> > > + cfqq = cic->cfqq[rw & REQ\_RW\_SYNC];

> > > if (cfqq) {

> > > cfq\_init\_prio\_data(cfqq);

> > > cfq\_prio\_boost(cfqq);

> > >

> > Ahem, how well did you test this patch?

> >

> > Ugh, again: bio\_sync() returns not only 0/1

> > Sorry for giving so much trouble...

>

> Right, and REQ\_RW\_SYNC isn't 1 either, so it returns a large number if

> set.

This is also needed, you can't just rely on bio\_sync(), a check for a  
READ is needed as well.

```
diff --git a/block/cfq-iosched.c b/block/cfq-iosched.c
```

```
index 17e4660..b793817 100644
```

```
--- a/block/cfq-iosched.c
```

```
+++ b/block/cfq-iosched.c
```

```
@@ -215,6 +215,18 @@ static inline void cic_set_cfqq(struct cfq_io_context *cic,  
{
```

```
/*
```

```
+ * We regard a request as SYNC, if it's either a read or has the SYNC bit
```

```
+ * set (in which case it could also be direct WRITE).
```

```
+ */
```

```
+static inline int cfq_bio_sync(struct bio *bio)
```

```
+{
```

```
+ if (bio_data_dir(bio) == READ || bio_sync(bio))
```

```
+ return 1;
```

```
+
```

```
+ return 0;
```

```

+}
+
+/*
 * scheduler run of queue, if there are requests pending and no one in the
 * driver that will restart queueing
 */
@@ -594,7 +606,7 @@ cfq_find_rq_fmerge(struct cfq_data *cfqd, struct bio *bio)
 if (!cic)
 return NULL;

- cfqq = cic_to_cfqq(cic, bio_sync(bio));
+ cfqq = cic_to_cfqq(cic, cfq_bio_sync(bio));
 if (cfqq) {
 sector_t sector = bio->bi_sector + bio_sectors(bio);

@@ -694,7 +706,7 @@ static int cfq_allow_merge(request_queue_t *q, struct request *rq,
 /*
 * Disallow merge of a sync bio into an async request.
 */
- if ((bio_data_dir(bio) == READ || bio_sync(bio)) && !rq_is_sync(rq))
+ if (cfq_bio_sync(bio) && !rq_is_sync(rq))
 return 0;

/*
@@ -705,7 +717,7 @@ static int cfq_allow_merge(request_queue_t *q, struct request *rq,
 if (!cic)
 return 0;

- cfqq = cic_to_cfqq(cic, bio_sync(bio));
+ cfqq = cic_to_cfqq(cic, cfq_bio_sync(bio));
 if (cfqq == RQ_CFQQ(rq))
 return 1;

```

--  
Jens Axboe

---