

---

Subject: [PATCH] cfq: async queue allocation per priority  
Posted by [Vasily Tarasov](#) on Wed, 18 Jul 2007 14:35:49 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Jens, I think the last patch, that makes queues allocation per priority, has a problem.

If we have two processes with different `ioprio_class`, but the same `ioprio_data`, their async requests will fall into the same queue. I guess such behavior is not expected, because it's not right to put real-time requests and best-effort requests in the same queue.

The attached patch fixes the problem by introducing additional `*cfqq` fields on `cfqd`, pointing to per-(class,priority) async queues.

Thanks,  
Vasily

---

#### File Attachments

1) [diff-cfq-asyn-queues-per-prio](#), downloaded 279 times

---

---

Subject: Re: [PATCH] cfq: async queue allocation per priority  
Posted by [Jens Axboe](#) on Wed, 18 Jul 2007 18:51:45 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Wed, Jul 18 2007, Vasily Tarasov wrote:

> Jens, I think the last patch, that makes queues allocation per priority,  
> has a problem.  
>  
> If we have two processes with different `ioprio_class`, but the same  
> `ioprio_data`, their async requests will fall into the same queue. I guess  
> such behavior is not expected, because it's not right to put real-time  
> requests and best-effort requests in the same queue.  
>  
> The attached patch fixes the problem by introducing additional `*cfqq`  
> fields on `cfqd`, pointing to per-(class,priority) async queues.

Ugh yes. I'm pretty tempted just to reinstate the `cfqq` hash again, it used to be a clean up but now the it's not stacking up so well.

--  
Jens Axboe

---

---

Subject: Re: [PATCH] cfq: async queue allocation per priority

Posted by [Vasily Tarasov](#) on Thu, 19 Jul 2007 07:52:36 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Wed, 2007-07-18 at 20:51 +0200, Jens Axboe wrote:

> On Wed, Jul 18 2007, Vasily Tarasov wrote:  
> > Jens, I think the last patch, that makes queues allocation per priority,  
> > has a problem.  
> >  
> > If we have two processes with different ioprio\_class, but the same  
> > ioprio\_data, their async requests will fall into the same queue. I guess  
> > such behavior is not expected, because it's not right to put real-time  
> > requests and best-effort requests in the same queue.  
> >  
> > The attached patch fixes the problem by introducing additional \*cfqq  
> > fields on cfqd, pointing to per-(class,priority) async queues.  
>  
> Ugh yes. I'm pretty tempted just to reinstate the cfqq hash again, it  
> used to be a clean up but now the it's not stacking up so well.  
>

Hello, Jens,

>From my humble point of view cfqq hash has two problems:

1. It is excess data structure. All needed information can be obtained from other structures easily, so the presence of hash is a bit strange... I mean that it's aim is not obvious :)
2. Hash hides from a developer a pretty important concept of CFQ: there are shared between processes per-priority async queues. I think the code is the best documentation, so the explicit async cfqq pointers at cfqd structure reveal this concept greatly.

Summary:

IMHO the hash revival is not very good way. However, this is of course fully in your competence to choose the right decision! ;)

Thank you,  
Vasily

---

Subject: Re: [PATCH] cfq: async queue allocation per priority  
Posted by [Jens Axboe](#) on Thu, 19 Jul 2007 17:30:53 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On Thu, Jul 19 2007, Vasily Tarasov wrote:

> On Wed, 2007-07-18 at 20:51 +0200, Jens Axboe wrote:

> > On Wed, Jul 18 2007, Vasily Tarasov wrote:  
> > > Jens, I think the last patch, that makes queues allocation per priority,  
> > > has a problem.  
> > >  
> > > If we have two processes with different ioprio\_class, but the same  
> > > ioprio\_data, their async requests will fall into the same queue. I guess  
> > > such behavior is not expected, because it's not right to put real-time  
> > > requests and best-effort requests in the same queue.  
> > >  
> > > The attached patch fixes the problem by introducing additional \*cfqq  
> > > fields on cfqd, pointing to per-(class,priority) async queues.  
> >  
> > Ugh yes. I'm pretty tempted just to reinstate the cfqq hash again, it  
> > used to be a clean up but now the it's not stacking up so well.  
> >  
>  
> Hello, Jens,  
>  
> From my humble point of view cfqq hash has two problems:  
>  
> 1. It is excess data structure. All needed information can be obtained  
> from other structures easily, so the presence of hash is a bit  
> strange... I mean that it's aim is not obvious :)  
>  
> 2. Hash hides from a developer a pretty important concept of CFQ: there  
> are shared between processes per-priority async queues. I think the code  
> is the best documentation, so the explicit async cfqq pointers at cfqd  
> structure reveal this concept greatly.  
>  
> Summary:  
>  
> IMHO the hash revival is not very good way. However, this is of course  
> fully in your competence to choose the right decision! ;)

Yeah, it's probably still better off without the hash. I'll play with it  
a bit and see what comes of it.

--  
Jens Axboe

---