

Re[2]: How does disk caching work?

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/performance/2004-04/0043.html>

From: Q?=Igor ShmuklerQ=20?= (shmukler_at_mail.ru)

Date: 04/20/04

To: Q?=Uwe DoeringQ=20?= <gemini@geminix.org>

Date: Tue, 20 Apr 2004 22:04:11 +0400

> >> *The reason for this algorithm is that it is better to keep pages in the
> >> inactive queue for as long as possible, rather than moving them over to
> >> the cache queue prematurely. Pages in the inactive queue can be still
> >> mapped into the memory space of processes, while pages in the cache
> >> queue have lost this association. So, quite naturally, when the VM
> >> system has to reactivate a page (put it back into the active queue) this
> >> operation tends to be less expensive when the page is still in the
> >> inactive queue.*

> >

> > *While you are correct that when cache is empty kernel will dip into the inactive queue. You are mistaken
about other things. Pages on the cache queue still have the association. I wrote that one of the previous posts.*

> >

> > *To sum it up: cache queue is same as inactive queue except it has only clean pages.*

> >

> > *If things were the you suggest, cache queue would be totally useless.*

>

> *I think you're mixing up two different things here. The way I
> understand the kernel sources, the pages in the cache queue of course
> still have their association with the underlying VM object. Otherwise
> caching these pages would be useless. But they are no longer mapped
> into any process address space. If I may quote the relevant comment
> from vm_page_cache():*

>

> /*

> * Remove all pmaps and indicate that the page is not

> * writeable or mapped.

> */

>

> *vm_page_cache() is the function that moves the pages from the inactive
> to the cache queue once they are clean. Restoring the process address
> space mapping is what makes reactivating pages from the cache queue more
> expensive than just relinking them from the inactive queue, because a
> fault gets generated when the process tries to access the page. This
> fault then maps the page from the VM object into the process address
> space. This causes additional overhead.*

>

freebsd-performance: Re[2]: How does disk caching work?

> > *I actually pretty much explain the whole rotation process. If you read my email again, you should understand what happens whenever page is moved from inactive to cache and then to free.*

>

> *You may want to study the kernel sources some more, I'm afraid.*

First you explicitly write that "pages in the cache queue have lost this association" then you tell me that I don't understand how VM works.

Are you trying to suggest that mapping page and change permission is comparable with reading page from backing store?

Studying sources never hurts, but filtering lingo is just as helpful.

> >> *So, for reasons like these, I keep recommending to either study the*

> >> *kernel sources before you try to tune the VM system, or leave these*

> >> *variables alone.*

> >

> > *I am not sure whether studying kernel sources is really necessary. Virtually every UNIX (R) admin had to tune the machine, despite sources not being available.*

>

> *Sorry, but you just proved my point ...*

What was that?

In first email you write that size of cache queue does not affect disk traffic. In next you say, no I did not mean that. I just wanted to say that cache queue holds pages that lost association. Now you say, no of course there is an association, someone just has to study kernel sources better.

Last argument is valid only because it's a moot point. Studying kernel sources never hurts anyone ...

I did not originally intend to flame you. I simply thought that some of your answers were not correct. If you had answered off the list, I would not bother, but this a public mailing list and it is a source of knowledge for many people. VM in particular is a grey area for many developers. Everyone knows what it's for, but few programmers really understand VM or VFS. Now you start giving me funny advices. That's not wise.

Sincerely,

IS.

freebsd-performance@freebsd.org mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-performance>

To unsubscribe, send any mail to "freebsd-performance-unsubscribe@freebsd.org"