Subject: How Inactive may be much greather than cached?
Posted by vaverin on Thu, 18 Oct 2007 06:24:04 GMT
View Forum Message <> Reply to Message

Hi all,

could anybody explain how "inactive" may be much greater than "cached"?
stress test (http://weather.ou.edu/~apw/projects/stress/) that writes into
removed files in cycle puts the node to the following state:

MemTotal: 16401648 kB
MemFree: 636644 kB
Buffers: 1122556 kB
Cached: 362880 kB
SwapCached: 700 kB
Active: 1604180 kB
Inactive: 13609828 kB

At the first glance memory should be freed on file closing, nobody refers to
file and ext3_delete_inode() truncates inode. We can see that memory is go away
from "cached", however could somebody explain why it become "invalid" instead be
freed? Who holds the references to these pages?

thank you,
Vasily Averin

Subject: Re: How Inactive may be much greather than cached?
Posted by Nick Piggin on Thu, 18 Oct 2007 06:27:27 GMT
View Forum Message <> Reply to Message

Hi,

On Thursday 18 October 2007 16:24, Vasily Averin wrote:
> Hi all,
>
> could anybody explain how "inactive" may be much greater than "cached"?
> stress test (http://weather.ou.edu/~apw/projects/stress/) that writes into
> removed files in cycle puts the node to the following state:
>
> MemTotal: 16401648 kB
> MemFree: 636644 kB
> Buffers: 1122556 kB
> Cached: 362880 kB
> SwapCached: 700 kB
> Active: 1604180 kB
> Inactive: 13609828 kB
>

> At the first glance memory should be freed on file closing, nobody refers
> to file and ext3_delete_inode() truncates inode. We can see that memory is
> go away from "cached", however could somebody explain why it become
> "invalid" instead be freed? Who holds the references to these pages?

Buffers, swap cache, and anonymous.

---

Subject: Re: How Inactive may be much greather than cached?
Posted by vaverin on Thu, 18 Oct 2007 07:14:05 GMT
View Forum Message <> Reply to Message

Nick Piggin wrote:
> Hi,
>
> On Thursday 18 October 2007 16:24, Vasily Averin wrote:
>> Hi all,
>>
>> could anybody explain how "inactive" may be much greater than "cached"?
>> stress test (http://weather.ou.edu/~apw/projects/stress/) that writes into
>> removed files in cycle puts the node to the following state:
>>
>> MemTotal: 16401648 kB
>> MemFree: 636644 kB
>> Buffers: 1122556 kB
>> Cached: 362880 kB
>> SwapCached: 700 kB
>> Active: 1604180 kB
>> Inactive: 13609828 kB
>>
>> At the first glance memory should be freed on file closing, nobody refers
>> to file and ext3_delete_inode() truncates inode. We can see that memory is
>> go away from "cached", however could somebody explain why it become
>> "invalid" instead be freed? Who holds the references to these pages?
>
> Buffers, swap cache, and anonymous.

But buffers and swap cache are low (1.1 Gb and 700kB in this example) and
anonymous should go away when process finished.

---

Subject: Re: How Inactive may be much greather than cached?
Posted by Nick Piggin on Thu, 18 Oct 2007 07:27:00 GMT
View Forum Message <> Reply to Message

On Thursday 18 October 2007 17:14, Vasily Averin wrote:
> Nick Piggin wrote:

> > Hi,
> >
> > On Thursday 18 October 2007 16:24, Vasily Averin wrote:
> >> Hi all,
> >>
> >> could anybody explain how "inactive" may be much greater than "cached"?
> >> stress test (http://weather.ou.edu/~apw/projects/stress/) that writes
> >> into removed files in cycle puts the node to the following state:
> >>
> >> MemTotal: 16401648 kB
> >> MemFree:    636644 kB
> >> Buffers:   1122556 kB
> >> Cached:     362880 kB
> >> SwapCached:    700 kB
> >> Active:    1604180 kB
> >> Inactive: 13609828 kB
> >>
> >> At the first glance memory should be freed on file closing, nobody
> >> refers to file and ext3_delete_inode() truncates inode. We can see that
> >> memory is go away from "cached", however could somebody explain why it
> >> become "invalid" instead be freed? Who holds the references to these
> >> pages?
> >
> > Buffers, swap cache, and anonymous.
>
> But buffers and swap cache are low (1.1 Gb and 700kB in this example) and
> anonymous should go away when process finished.

Ah, I didn't see it was an order of magnitude out.

Some filesystems, including I believe, ext3 with data=ordered,
can leave orphaned pages around after they have been truncated
out of the pagecache. These pages get left on the LRU and vmscan
reclaims them pretty easily.

Try ext3 data=writeback, or even ext2.

---

Subject: Re: How Inactive may be much greather than cached?
Posted by vaverin on Thu, 18 Oct 2007 10:33:32 GMT
View Forum Message <> Reply to Message

Nick Piggin wrote:
> Some filesystems, including I believe, ext3 with data=ordered,
> can leave orphaned pages around after they have been truncated
> out of the pagecache. These pages get left on the LRU and vmscan
> reclaims them pretty easily.
>

> Try ext3 data=writeback, or even ext2.

thanks, data=writeback helps.

Resume: ext3 with data=ordered gets bh with data and moves it to journal
transaction. If transaction handled immediately, ext3 frees bh on this page, and
then frees this page.
However if journal delays processing of this transaction, ext3 cannot free bh
that is still busy. Later jbd layer decrements bh counter but it makes nothing
with data page that is not freed and stays inactive.

---

Subject: Re: How Inactive may be much greather than cached?
Posted by Rik van Riel on Thu, 18 Oct 2007 14:17:15 GMT
View Forum Message <> Reply to Message

On Thu, 18 Oct 2007 17:27:00 +1000
Nick Piggin <nickpiggin@yahoo.com.au> wrote:

> Some filesystems, including I believe, ext3 with data=ordered,
> can leave orphaned pages around after they have been truncated
> out of the pagecache. These pages get left on the LRU and vmscan
> reclaims them pretty easily.

How can the VM recognize those pages? Are they part of
the buffer cache, part of the page cache, or different?

I think it would make sense to at least try to rotate
those pages to the end of the LRU so kswapd can get rid
of them quickly.

--
"Debugging is twice as hard as writing the code in the first place.
Therefore, if you write the code as cleverly as possible, you are,
by definition, not smart enough to debug it." - Brian W. Kernighan

---