Subject: Re: [Q] missing unused dentry in prune_dcache()?
Posted by David Howells on Wed, 25 Oct 2006 14:23:29 GMT
View Forum Message <> Reply to Message

Vasily Averin <vvs@sw.ru> wrote:

> >> The patch adds this dentry into tail of the dentry_unused list.
> >
> > I think that's reasonable. I wonder if we can avoid removing it from the
> > list in the first place, but I suspect it's less optimal.
>
> Could you please explain this place in details, I do not understand why tail
> of the list is better than head. Also I do not understand why we should go
> to out in this case. Why we cannot use next dentry in the list instead?

I meant adding it back into the list is reasonable; I didn't actually consider where you were adding it back.

However, you've just raised a good point: moving a dentry to the head of the list is what happens when we see it's been referenced recently. The most recently used entry is at the head and the least at the tail.

If you look at the list scan algorithm, you can see it works from the tail towards the head.

We could re-insert the dentry at the position from which we removed it, but that would mean retaining a pointer to one of its neighbouring dentries. We can do that - it's just stack space after all...

I think the main point though, is generally that the superblock that owns the dentry is very busy in some critical way; either it's being unmounted (in which case we may as well put the dentry at the MRU end), or it's being remounted (in which case, it's probably best putting it back where we found it).

You also have to consider how often superblocks are going to be unmounted or remounted vs the dcache being pruned...

There is also grab_super(), but that's only used by sget() when a superblock is being mounted again - also unlikely.

So, given that the three ops that affect this are very unlikely to be happening at any one time, it's probably worth just slapping it at the head and ignoring it. The main thing is that it's reinserted somewhere.

Hope that helps...

David