Subject: Re: Re: [PATCH 2.6.19-rc3] VFS: per-sb dentry lru list
Posted by vaverin on Wed, 01 Nov 2006 13:32:03 GMT
View Forum Message <> Reply to Message

Hello Neil,

Neil Brown wrote:
> On Tuesday October 31, dhowells@redhat.com wrote:
>> Neil Brown <neilb@suse.de> wrote:
>>
>>> When we unmount a filesystem we need to release all dentries.
>>> We currently
>>>   - move a collection of dentries to the end of the dentry_unused list
>>>   - call prune_dcache to prune that number of dentries.
>> This is not true anymore.
>
> True. That should read:
>
> When we remount a filesystem or invalidate a block device which has a
> mounted filesystem we call shrink dcache_sb which currently:
>   - moves a collection of dentries to the end of the dentry_unused list
>   - calls prune_dcache to prune that number of dentries.
>
> but the patch is still valid.
>
> Any objections to it going in to -mm and maybe .20 ??

Currently we have 3 type of functions that works with dentry_unused list:

1) prune_dcache(NULL) -- called from shrink_dcache_memory, frees the memory and requires global LRU. works well in current implementation.
2) prune_dcache(sb) -- called from shrink_dcache_parent(), frees subtree, LRU is not need here. Current implementation uses global LRU for these purposes, it is ineffective, and patch from Neil Brown fixes this issue.
3) shrink_dcache_sb() -- called when we need to free the unused dentries for given super block. Current implementation is not effective too, and per-sb LRU would be the best solution here. On the other hand patch from Neil Brown is much better than current implementation.

In general I think that we should approve Neil Brown's patch. We (I and Kirill Korotaev) are ready to acknowledge it when the following remarks fill be fixed:

- it seems for me list_splice() is not required inside prune_dcache(),
- DCACHE_REFERENCED dentries should not be removed from private list to dentry_unused list, this flag should be ignored if the private list is used,
- count argument should be ignored in this case too, we want to free all the dentries in private list,
- when we shrink the whole super block we should free per-sb anonymous dentries

too (please see Kirill Korotaev's letter)

Then I'm going to prepare new patch that will enhance the shrink_dcache_sb() performance:
- we can add new list head into struct superblock and use it in shrink_dcache_sb() instead of temporal private list. We will check is it empty in dput() and add the new unused dentries to per-sb list instead of dentry_unused list.

thank you,
Vasily Averin

SWsoft Virtuozzo/OpenVZ Linux kernel team