# fs - Bug #15508

## client: simultaneous readdirs are very racy

04/14/2016 11:01 PM - Greg Farnum

| | | | |
|---|---|---|---|
| **Status:** | Resolved | **Start date:** | 04/14/2016 |
| **Priority:** | High | **Due date:** | |
| **Assignee:** | Zheng Yan | **% Done:** | 0% |
| **Category:** | | **Estimated time:** | 0.00 hour |
| **Target version:** | | | |
| **Source:** | Development | **Affected Versions:** | |
| **Tags:** | | **ceph-qa-suite:** | |
| **Backport:** | jewel | **Component(FS):** | Client |
| **Regression:** | No | **Labels (FS):** | |
| **Severity:** | 3 - minor | **Pull request ID:** | |
| **Reviewed:** | | | |

**Description**

Imagine we have a ceph-fuse user doing readdirs a and b on a very large directory (which requires multiple MDS round-trips, and multiple local readdir syscalls for every MDS round trip).

a finishes first. Because the directory wasn't changed, it marks the directory COMPLETE|ORDERED
b has last received an MDS readdir for offsets x to y and is serving those results

readdir c starts from offset 0.
b finishes up to y, and sends off an MDS request to readdir starting at y+1
readdir c reaches location y+1 from cache
b's response comes in. It pushes the range y+1 to z to the back of the directory's dentry xlist!
readdir c continues up to z before readdir b manages to get z+1 read back from the MDS.
readdir c ends prematurely because xlist::iterator::end() returns true.

**Related issues:**

| | | |
|---|---|---|
| Related to fs - Bug #13271: Missing dentry in cache when doing readdirs under... | **Resolved** | **09/29/2015** |
| Copied to fs - Backport #16251: jewel: client: simultaneous readdirs are very... | **Resolved** | |

### History

**#1 - 04/14/2016 11:06 PM - Greg Farnum**

*- Priority changed from Normal to High*

Some obvious solutions are disqualified, both because we can't really track what directory listing's are in progress (via dirp's), and in particular because the client might just drop a readdir set or crash before finishing. So the solution needs to depend only on internal state tracking.

I'm working on it. So far the winning approach is

- keep track of the shared_gen when starting an MDS listing from offset 0 (well, 2, I guess)
- when we get a response, if the shared_gen hasn't changed, set an "ordered_thru" to the latest offset
- when satisfying a readdir, reference that ordered_thru instead of the simple COMPLETE and ORDERED flags :/

There are plenty of missing parts to that, but I think the basic scheme should be sound. (It sounds just a little bit like PG backfilling...)

**#2 - 04/14/2016 11:06 PM - Greg Farnum**

*- Related to Bug #13271: Missing dentry in cache when doing readdirs under cache pressure (?????s in ls-l) added*

#### #3 - 04/15/2016 03:12 AM - Zheng Yan

Another option is assign dentry a cache index and use array to track the dentry list. If the shared_gen hasn't changed, a given dentry is always at the same position of the array. This is how kernel client currently does.

#### #4 - 04/15/2016 03:19 AM - Greg Farnum

Hmm, I think the end result would be pretty much the same, although just having an array might be simpler. A pointer per dentry in an open frag isn't that expensive even if we are evicting stuff...*ponders*

#### #5 - 05/09/2016 12:19 PM - Zheng Yan

*- Assignee changed from Greg Farnum to Zheng Yan*

I found that seekdir can also trigger this issue. I'm working on fixing it.

#### #6 - 05/09/2016 02:30 PM - Zheng Yan

*- Status changed from New to Need Review*

last commit of https://github.com/ceph/ceph/pull/8739

#### #7 - 06/12/2016 09:35 PM - Greg Farnum

*- Status changed from Need Review to Pending Backport*

Backport PR: https://github.com/ceph/ceph/pull/9655

#### #8 - 06/13/2016 04:52 AM - Nathan Cutler

*- Backport set to jewel*

#### #9 - 06/13/2016 04:54 AM - Nathan Cutler

*- Copied to Backport #16251: jewel: client: simultaneous readdirs are very racy added*

#### #10 - 06/13/2016 08:23 AM - Greg Farnum

*- Status changed from Pending Backport to Resolved*

#### #11 - 07/13/2016 12:27 AM - Greg Farnum

*- Component(FS) Client added*