

NFS cache consistency appears to be broken...

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2005-11/9609.html>

From: Steve Dickson (SteveD_at_redhat.com)

Date: 11/30/05

Date: Tue, 29 Nov 2005 21:29:20 -0500
To: Trond Myklebust <trond.myklebust@fys.uio.no>

Hey Trond,

The attached patch seems to break cache consistence in a big way....

Doing the following:

1. On server:

```
$ mkdir ~/t
```

```
$ echo Hello > ~/t/tmp
```

2. On client, wait for a string to appear in this file:

```
$ until grep -q foo t/tmp ; do echo -n . ; sleep 1 ; done
```

3. On server, create a *new* file with the same name containing that string:

```
$ mv ~/t/tmp ~/t/tmp.old; echo foo > ~/t/tmp
```

will shows how the client will never (and I mean never ;-)) see the updated file. I reverted this patch and everything started work as expected... so it appears using a jiffy-based cache verifiers may not be such a good idea....

Note: I am using 2.6.15-rc2 kernel.

steved.

Subject:

NFS: Convert cache_change_attribute into a jiffy-based value

From:

Linux Kernel Mailing List <linux-kernel@vger.kernel.org>

Date:

Fri, 28 Oct 2005 09:07:29 -0700

To:

git-commits-head@vger.kernel.org

tree 1f3d5db26462d02ecca383794b3061a5eae8d9cc
parent 0e574af1be5f569a5d7f2800333b0bfb358a5e34
author Trond Myklebust <Trond.Myklebust@netapp.com> Fri, 28 Oct 2005 06:12:38 -0400

Linux-Kernel: NFS cache consistency appears to be broken...

committer Trond Myklebust <Trond.Myklebust@netapp.com> Fri, 28 Oct 2005 06:12:38 -0400

NFS: Convert cache_change_attribute into a jiffy-based value

Signed-off-by: Trond Myklebust <Trond.Myklebust@netapp.com>

fs/nfs/inode.c | 8 ++++-----

include/linux/nfs_fs.h | 2 +-
2 files changed, 5 insertions(+), 5 deletions(-)

diff --git a/fs/nfs/inode.c b/fs/nfs/inode.c

--- a/fs/nfs/inode.c

+++ b/fs/nfs/inode.c

@@ -1135,7 +1135,7 @@ __nfs_revalidate_inode(struct nfs_server

 * We may need to keep the attributes marked as invalid if

 * we raced with nfs_end_attr_update().

 */

- if (verifier == nfsi->cache_change_attribute)

+ if (time_after_eq(verifier, nfsi->cache_change_attribute))

 nfsi->cache_validity &= ~(NFS_INO_INVALID_ATTR|NFS_INO_INVALID_ETIME);

 spin_unlock(&inode->i_lock);

@@ -1202,7 +1202,7 @@ void nfs_revalidate_mapping(struct inode

 if (S_ISDIR(inode->i_mode)) {

 memset(nfsi->cookieverf, 0, sizeof(nfsi->cookieverf));

 /* This ensures we revalidate child dentries */

- nfsi->cache_change_attribute++;

+ nfsi->cache_change_attribute = jiffies;

 }

 spin_unlock(&inode->i_lock);

@@ -1242,7 +1242,7 @@ void nfs_end_data_update(struct inode *i

 nfsi->cache_validity |= NFS_INO_INVALID_DATA;

 spin_unlock(&inode->i_lock);

 }

- nfsi->cache_change_attribute ++;

+ nfsi->cache_change_attribute = jiffies;

 atomic_dec(&nfsi->data_updates);

 }

@@ -1391,7 +1391,7 @@ static int nfs_update_inode(struct inode

 /* Do we perhaps have any outstanding writes? */

 if (nfsi->npages == 0) {

 /* No, but did we race with nfs_end_data_update()? */

- if (verifier == nfsi->cache_change_attribute) {

+ if (time_after_eq(verifier, nfsi->cache_change_attribute)) {

 inode->i_size = new_isize;

 invalid |= NFS_INO_INVALID_DATA;

 }

diff --git a/include/linux/nfs_fs.h b/include/linux/nfs_fs.h

--- a/include/linux/nfs_fs.h

NFS cache consistency appears to be broken...

Linux-Kernel: NFS cache consistency appears to be broken...

```
+++ b/include/linux/nfs_fs.h
@@ -280,7 +280,7 @@ static inline long nfs_save_change_attr
static inline int nfs_verify_change_attribute(struct inode *inode, unsigned long chattr)
{
    return !nfs_caches_unstable(inode)
- && chattr == NFS_I(inode)->cache_change_attribute;
+ && time_after_eq(chattr, NFS_I(inode)->cache_change_attribute);
}
```

/*

-

To unsubscribe from this list: send the line "unsubscribe git-commits-head" in the body of a message to majordomo@vger.kernel.org

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

-

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@vger.kernel.org

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>