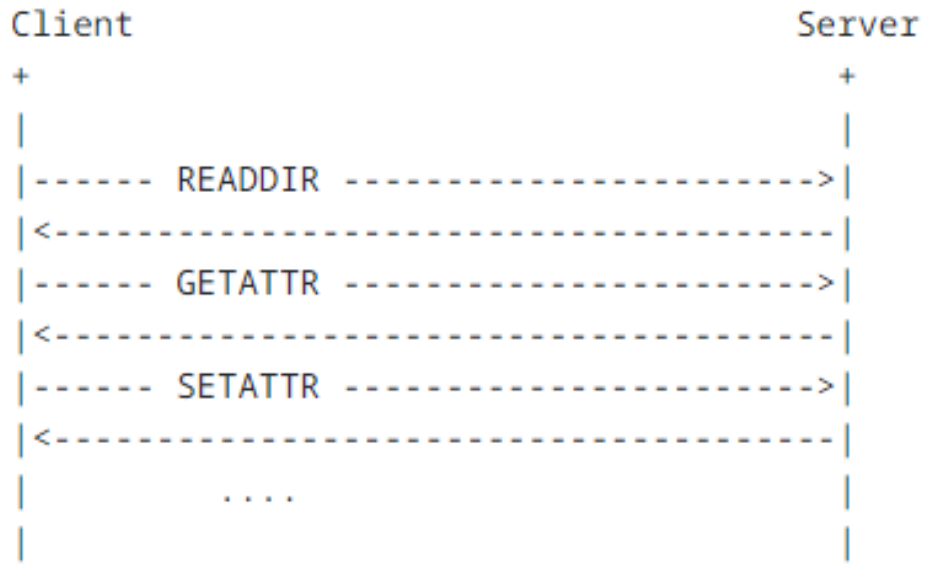
Recursively Setting Attributes of Subdirectories and files

Authors:
Minqian Zhang Huawei Technologies
Sunil Kumar Bhargo Broadcom Inc.
Rijesh Kunhi Parambattu Huawei Technologies
Dongyu Geng Huawei Technologies

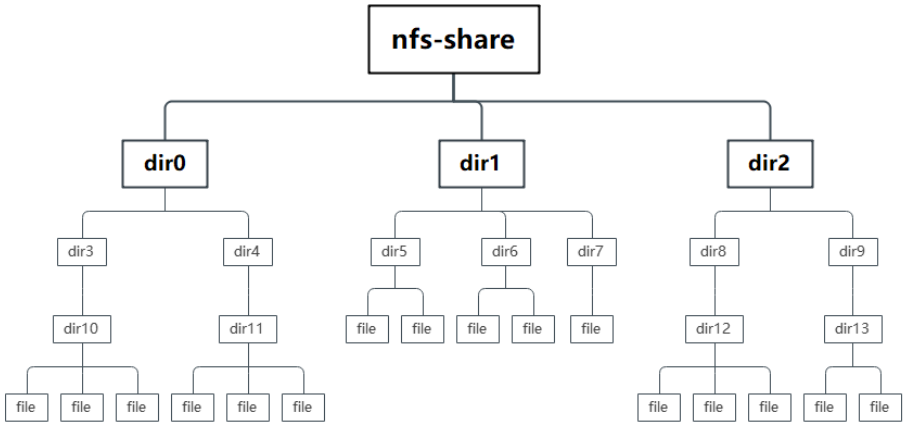
Backgroun
d

Existing sequence of recursively setting the attributes of all files under directory.



Step 1: The client sends the READDIR command to obtain the list of all files in dir1.
Step 2: The storage server responds to the READDIR operation. If the directory contains many subdirectories and files the client needs to run the READDIR operation for multiple times.
Step 3: The client sends a SETATTR request for each subdirectory and file.
Step 4: The storage server responds to the SETATTR request.

Example: 15 directories and 16 files



Problems of current solution:
1) Lots of loops between client and server: Readdir 15 times + Getattr 31 times + Setattr 31 times
2) Client need maintain the readdir results.
3) Consumes lots of bandwidth resources.

Motivation

- **Make setattr for huge files more faster than current solution.**
- **Reduce the bandwidth resource consumes.**
- **Reduce the client pressure during setattr for huge files.**

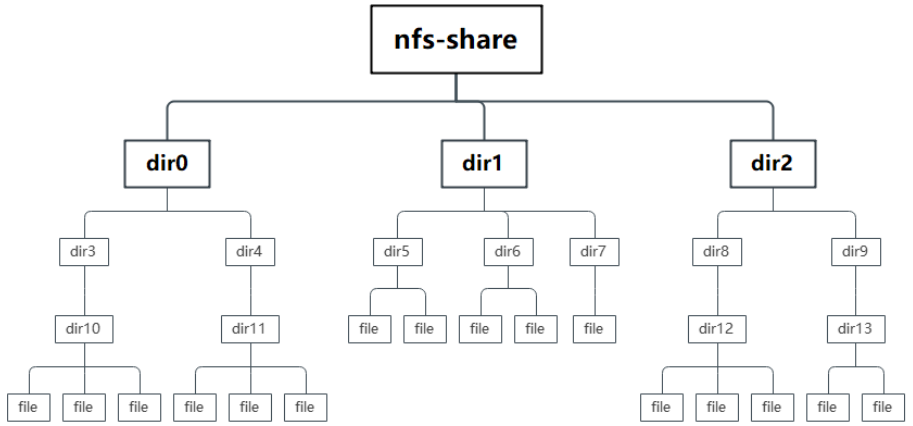
Our proposal

RECURSIVE_SET –
Recursively sets the attributes of a directory and its subdirectories and files.

RECURSIVE_SET_STATUS –
Query the result of the recursively setting the attributes of subdirectories and files.

RECURSIVE_SET_CANCEL –
Canceling a Running Task on the Client

CB_RECURSIVE_SET_NOTIFY –
Notify the recursively setting result to client

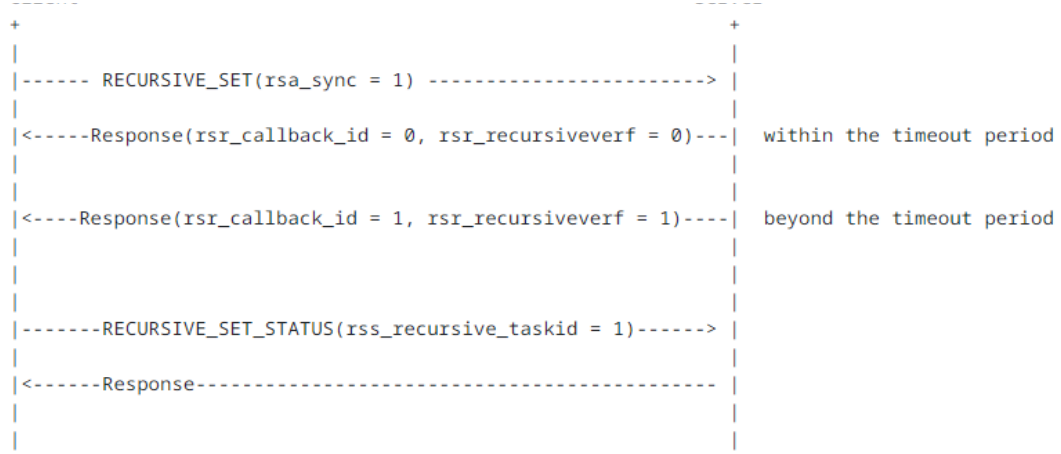


Usage:
Step 1 Client send compound request to server.
Step 2 Server recursive search all the subdirectories and file and do setattr.
Step 3 Server response the client.

Compared with current solution, our proposal reduce the loop counts between client from 46 to 2 and need less resources and bandwidth.

Detail of proposal

A synchronous Recursive Set

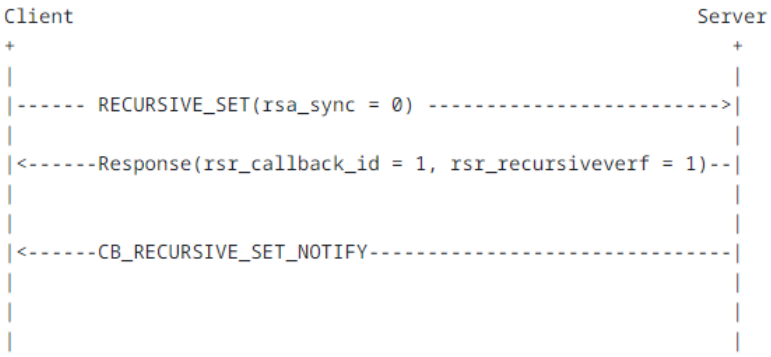


Client send request with sync is true.

Small count of directories and files scenarios, server can finish the recursive setattr operations within the timeout period and directly response.

Large count of directories and files scenarios, server can not finish the recursive setattr operations within the timeout period and server return the task id, client query the status periodly.

An asynchronous Recursive Set



Client send request with sync is false.

Client send the recursive set request.
Server response with task id to client.
Server send the result to client once finished during backchannel.

Definition

RECURSIVE_SET –
Recursively sets the attributes of a directory and its subdirectories and files.

```
ARGUMENT

Struct RECURSIVE_SET4args {
        bool rsa_sync;
};

RESULT

struct recursive_set_response4 {
        recursive_taskid4 rsr_callback_id;
        verifier4 rsr_recursiveverf;
};

union RECURSIVE_SET4res (nfsstat4 rsr_status) {
        case NFS4_OK:
                recursive_set_response4 rsr_resok4;
        default:
                void;
};
```

Definition

RECURSIVE_SET_STATUS –
Query the result of the recursively setting the attributes of subdirectories and files.

ARGUMENT

```
struct RECURSIVE_SET_STATUS4args {
        stateid4 rssa_recursive_taskid;
};
```

RESULT

```
#define NFS4ERR_PENDING 10090

struct RECURSIVE_SET_STATUS4res {
        nfsstat4 rssr_status;
};
```

Definition

RECURSIVE_SET_CANCEL –
Canceling a Running Task on the Client

ARGUMENT

```
struct RECURSIVE_SET_CANCEL4args {
        stateid4 rsca_recursive_taskid;
};
```

RESULT

```
struct RECURSIVE_SET_CANCEL4res {
        nfsstat4 rscr_status;
};
```

Definition

CB_RECURSIVE_SET_NOTIFY –
Notify the recursively setting result to client

ARGUMENT

```
struct CB_RECURSIVE_SET_NOTIFY4args {
        nfs_fh4 crsna_fh;
        stateid4 crsna_recursive_taskid;
        verifier4 crsna_recursiveverf;
        nfsstat4 crsna_status;
};
```

RESULT

```
struct CB_RECURSIVE_SET_NOTIFY4res {
        nfsstat4 crsnr_status;
};
```

Thank you!