NetApp™

Go further, faster™

Interpreting LAYOUTCOMMIT in RFC5661

Mike Eisler

Interim NFSv4 WG Meeting

2011-02-19

Background

- LAYOUTCOMMIT is needed in the blocks and objects layouts in order to synchronize file system meta data state between MDS and Data Servers (Storage Devices)
- LAYOUTCOMMIT has secondary purpose of optionally synchronizing the change, modification, and size attributes between DSes and MDS
- Many LAYOUT4_NFSV4_1_FILES pNFS servers do not want LAYOUTCOMMITs (or at least not all the time)
  - They are non-operations
  - Or worse, they are performance killers

Background (Continued)

- Requirement:
  - define more explicitly when LAYOUTCOMMIT is needed when using LAYOUT4_NFSV4_1_FILES layouts
  - indicate to client unambiguously when LAYOUTCOMMIT of LAYOUT4_NFSV4_1_FILES layout is needed
  - define consequences of not sending LAYOUTCOMMIT on a LAYOUT4_NFSV4_1_FILES layout
- Desire:
  - Do this without another minor version of NFSv4
    - Do this as errata or update to NFSv4.1 (RFC5661)

Disclaimer

- The remainder of this presentation is in the context of the LAYOUT4_NFSV4_1_FILES layout type

WRITE to a DS can sometimes obviate LAYOUTCOMMIT

- WRITE returns a stable_how4 value, one of:

```
UNSTABLE4 = 0,
DATA_SYNC4 = 1,
FILE_SYNC4 = 2
```

  - Difference between `DATA_SYNC4` and `FILE_SYNC4`:
    - Latter means data and file system metadata are on stable storage
    - Former means just data is on stable storage
- Assertion: any time a WRITE to a DS returns FILE_SYNC4, LAYOUTCOMMIT is **not** needed

COMMIT can sometimes obviate LAYOUTCOMMIT

- COMMIT returns a write verifier that applies to specified range
  - (which can be the whole file: offset and count equal to zero)
  - if verifier returned by WRITE and a subsequent COMMIT equal, then data is on stable storage
- The LAYOUT4_NFSV4_1_FILES layout type has a NFL4_UFLG_COMMIT_THRU_MDS flag
  - If set, client MUST NOT send COMMIT to DS, and MUST send COMMIT to MDS whenever WRITE returns UNSTABLE4
- Assertion: any time NFL4_UFLG_COMMIT_THRU_MDS is set, LAYOUTCOMMIT (of the byte range specified by the layout) is **not** needed.

When is a LAYOUTCOMMIT Needed?

Value of NFL4 UFLG COMMIT THRU MDS bit in layout	WRITE to Data Server returns	Meaning of COMMIT to Data Server	Meaning of COMMIT to MDS	LAYOUTCOMMIT needed?
Not set	UNSTABLE4	DATA_SYNC4	Nothing	Yes
Not set	DATA_SYNC4	Nothing	Nothing	Yes
Not Set	FILE_SYNC4	Nothing	Nothing	No
Set	UNSTABLE4	Nothing	FILE_SYNC4	No
Set	DATA_SYNC4	Nothing	FILE_SYNC4	No
Set	FILE_SYNC4	Nothing	Nothing	No

- Note that client can always demand FILE_SYNC4 or DATA_SYNC4 in WRITE's arguments.

If LAYOUTCOMMIT is needed, then when?

- Before CLOSE
  - Maintains Close-to-Open semantics
- Before LOCKU, OPEN_DOWNGRADE
- close(), fsync(), before unlocking a byte range
- Periodically to keep file size/modification time synchronized (tail –f use case)
  - Every N’th COMMIT to a DS
  - Every N’th unit of time
  - After writing a stripe to set of DSes

What if a needed LAYOUTCOMMIT is not sent?

- DSes MUST/SHOULD/MAY commit data and synchronize modification and size attributes with MDS when:
  - MUST: layout is revoked
    - This is already part of RFC5661 errata; rough consensus achieved
  - SHOULD: When a server crashes
    - In theory client should commit the data, but if client and server crashes overlap this could be missed
  - MAY: The MDS periodically polls DSes to synchronize attributes