Sptlrpc Interoperability HLD

Eric Mei

April 2008

1 Introduction

This document only regards to sptlrpc part. The whole picture of requirement of interoperability is documented on wiki "CMD3 Interoperability Architecture", from which some terms about release number are used here:

- *OLD*: last major release of 1.8 which based on b1_6 line of development.
- *OLD.x*: a 1.8.x release in b1_6 line containing client that is able to interact with a NEW.0 md server.
- *NEW.0*: first release based on HEAD. This features kernel server, and uses ldiskfs as a back-end.
- *NEW.DMU*: first release based on HEAD with user-level servers and DMU as a back-end.

It based on the assumption that we can't upgrade/downgrade directly between OLD and NEW.0, instead must use OLD.x as intermediate step.

2 Requirements

Keep HEAD-based 2.x interoperable with b1_6 based 1.8.x at sptlrpc part.

3 Functional specification

The RPC wire format is compatible between all versions, as long as *null* flavor or equivalent are used.

In OLD.x code, OBD (MDT, OST, client) config log interpreter should be modified to accept the new SPTLRPC_CONF records, but simply ignore them without report error.

After MGS upgrade to NEW.0, it converts existing config logs into new format which can be recognized by OBD of NEW.0 and OLD.x , but not OLD; After MGS downgrade to OLD.x, it converts existing config logs into old format which can be recognized by OBD of NEW.0, OLD.x, and OLD.

After everything upgrade to NEW.0, we can start adding sptlrpc rules, and corresponding ptlrpc connections will switch to designated flavor dynamically; Before downgrade MGS from NEW.0 to OLD.x, all existing sptlrpc config rules must be removed on NEW.0 MGS, thus all ptlrpc connections will switch back to the compatible null flavor dynamically.

Summary the limitations of upgrade/downgrad brought in by sptlrpc:

- Before OLD.x MGS upgrade to NEW.0, all other part of Lustre must have upgraded to OLD.x or NEW.0.
- Before add any sptlrpc rule on NEW.0 MGS, the whole Lustre must have upgraded to NEW.0.
- Before NEW.0 MGS downgrade to OLD.x, all sptlrpc rules must have been removed.
- Before any other part of Lustre of OLD.x downgrade to OLD, MGS must be OLD.x or OLD.

4 Use cases

- Rolling upgrade everything from OLD to OLD.x.
- Rolling upgrade everything from OLD.x to NEW.0.
- Adding sptlrpc rules on NEW.0 MGS, connection flavor change accordingly.
- Remove all sptlrpc rules on NEW.0 MGS, connection flavor change back to null accordingly.
- Rolling downgrade everything from NEW.0 to OLD.x.
- Rolling downgrade everything from OLD.x to OLD.

5 Logic specification

5.1 RPC wire data format

In OLD and OLD.x, there's no concept of sptlrpc flavor, all RPCs are sent in let's called "1.6 format". In NEW.0 and NEW.DMU, sptlrpc introduces several flavors which wrap

RPC on-wire messages in different, thus incompatible, ways. But the flavor null is the special one, which actually keep RPC wire format exactly the same as of old version (OLD.x and OLD). So as long as null flavor is the only flavor used through out the system, there's be no interoperability issues between all OLD, OLD.x, NEW.0 and NEW.DMU.

Specifically to each RPC exchange:

- OLD/OLD.x client send request in 1.6 format, NEW.0/NEW.DMU server will recognize it as null flavor; server send reply also in null flavor, which is recognized as 1.6 format by OLD/OLD.x client.
- NEW.0/NEW.DMU client send request in null flavor, which is recognized as 1.6 format by OLD/OLD.x server; server send reply also in 1.6 format, which is recognized as null flavor by NEW.0/NEW.DMU client.

The 1.6 format RPC means *lustre_msg_v1* and *lustre_msg_v2*, both could be accepted by NEW.0/NEW.DMU as null flavored RPC.

The incompatible formats of RPC will be sent out only if some sptlrpc rules was added on NEW.0 or NEW.DMU MGS. In order to prevent that:

- Don't add sptlrpc rules on NEW.0 MGS until every part of the system have been upgraded to NEW.0.
- Delete all sptlrpc rules (if any) on NEW.0 MGS before downgrade any part of the system from NEW.0 to OLD.x.

5.2 RPC flavor configuration

On a fresh installed NEW.0 system, following behavior related to sptlrpc configuration:

- MGS has a new database which hold all sptlrpc configuration rules. The database is a separate on-disk file, read/write using standard llog interface, and is created when first rule be inserted.
- There's a new record *LCFG_SPTLRPC_CONF* lives in each device's config log (MDT, OST, MDC, OSC), which specifying what security flavor to use/accept. Because security flavor must be decided before the device's first connect attemp, this record must be in the middle of log (right after LCFG_ATTACH). If the rules changed during runtime, another SPTLRPC_CONF record will also be append at end of the log.

5.2.1 OBD

OLD don’t recognize SPTLRPC_CONF record at all. If an OLD.x or NEW.0 obd device receives a config log without SPTLRPC_CONF record, by default null flavor will be used for future RPCs. If OLD.x obd device receives a config log which do contain SPTLRPC_CONF record, it check the ’flavor’ part: If it’s ’null’ then do nothing but return success, otherwise print out a warning and force to use ’null’ flavor.

5.2.2 MGS

If NEW.0 MGS is upgraded from OLD/OLD.x, there will be no SPTLRPC_CONF in the already existed logs. In this case MGS should still work by sending out logs without SPTLRPC_CONF records. But it won’t be able to change security flavors. To make sptlrpc configuration fully functional, all config logs must be recreated. This perhaps can be done somehow automatically when NEW.0 MGS is mounted at the first time.

If NEW.0 MGS downgrade to OLD.x, similarly we remove SPTLRPC_CONF records from all config logs at OLD.x MGS mount time. Thus config log sent out by OLD.x MGS will be able to recognized by OLD.x and OLD OBDs.

5.2.3 Sptlrpc rule database

The database is created when a NEW.0 MGS being requested to add the first rule. If NEW.0 MGS downgrade to OLD.x, all rules must be removed, and this database file will still exist but ignored, thus no harm.

6 State management

In any kind of recovery situations, even if which involves version changes, e.g. OLD.x MDS failover to NEW.0 MDS, only the null flavor or equivalent “1.6 format” RPC wire format will be used, thus all ptlrpc connections will always be compatible.

No persistent data changes. No wire-data, RPC order or protocol changes.

7 Alternatives

8 Focus for inspections

- Does this match with the overall Lustre 2.0 compatibility plan?