State of large NID support

Expand to new LNet protocols, May 2023

James Simmons

Storage Systems Engineer

Oak Ridge National Laboratory

Large NID support

- Long wanted feature
  - Once work started people started requesting status of this work
- Main goal is to allow Lnet setup with IPv6
  - Other protocols are possible like IB hardware addressing
- This implementation is a collaboration between SUSE and ORNL
  - Additional testing is done by Yehuda Yitshak (amazon)
- The goal to complete the foundational LNet support for the 2.16 release
- Lustre 2.17 will complete the support of large NID for everything (nodemap, GSS)

Progress up to 2.15 LTS

- Ticket LU-10391 opened Feb 2016
- Discussion with Linux community about native client lead to IPv6 requirement.
  - SUSE involvement
- Late 2019 discussion of LNet IPv6 design.
  - Lustre 2.13.52 we see first landings.
  - Changes are far reaching
- Lustre 2.15 LTS changed most of LNet core supports large NID
- No actual transmission of large NIDs with wire protocol
- No user land tool changes to allow large NIDs
  - Will back port patches to ignore large NIDs

Small change, big impact

- lnet_nid_t → struct lnet_nid
  - lnet_nid_t (64 bit – net_type | address)
  - struct lnet_nid {

```
        __u8    nid_size;       /* total bytes - 8 */
        __u8    nid_type;
        __be16  nid_num;
        __be32  nid_addr[4];
    } __attribute__((packed));
```

- nid_size == 0 then struct lnet_nid == lnet_nid_t (big endian)
- nid_type == 0xff means wild card (LNET_ANY_NID)
- Can be expanded up to 256 bits address.

Macro changes for user land

- Converstion functions:
  - void lnet_nid4_to_nid(lnet_nid_t nid4, struct lnet_nid *nid)
  - lnet_nid_t lnet_nid_to_nid4(const struct lnet_nid *nid)
- nid_is_same() is needed for comparison
- struct lnet_process_id → struct lnet_processid
- LNET_NID_ANY → LNET_ANY_NID
- LNET_NID_LO_0 → nid_is_lo0(nid)
- Wire shark changes (lnet-idl.h)
  - struct lnet_hdr_nid4 → struct lnet_hdr
  - Struct lnet_acceptor_connreq → struct lnet_acceptor_connreq_v2
  - Struct lnet_ni_status → struct lnet_ni_large_status

LNet tool changes

- Only visible change to lnetctl / lctl is taking large NID strings
    - lctl list_nids fe80:f68:45bd:7b60:e933@tcp
    - lnetctl ping fe80:f68:45bd:7b60:e933@tcp
- Internal code changes
    - Migrate to Netlink / YAML API
        - Allows changing userland interface without API breakage
    - Merged pre multi-rail APIs with multi-rail APIs (LU-10003)
    - At this time several patches are done but not merged.
    - Pieces still in progress
        - lnetctl net [show | set], lnetctl discover
        - lnetctl import / export, lnetctl udsp
        - lctl net fault

Other tool changes

- LNet selftest
  - Internal move to Netlink API (LU-8915)
    - WIP
  - Implement YAML configuration file support (LU-10975)
- Lustre changes
  - NRS debugfs / proc files take large NID string
  - NID export hash supports large NID string
    - lctl get_parm mdt.*.exports.$NID.hash
  - mount -t lustre [fe80:f68:45bd:7b60:e933]@tcp
    - https://review.whamcloud.com/#/c/fs/lustre-release/+/50362/

Lustre work left

- Enhance MGS NID table to handle large NIDS (LU-13306)
- Update Lustre UUIDs to handle large NIDs (LU-13340)
  - Found MGC UUID (“MGC”NID”_0) string can overflow
- Handle large NIDs for change logs (LU-13308)
- Remaining LU-10391 work for Lustre
  - Update l_getidentity to handle large NIDs
  - Update sptlrpc to handle large NIDs (LU-10937)
    - Could be completed during 2.17 cycle
- Add Large NID support to nodemap (LU-14288 / LU-13307)
  - Also could be completed during 2.17 cycle
- Interop testing between LTS and latest master
  - We need testers

New future Lustre functionality coming

- Do LNet discovery in background (LU-14668) (Mostly done)
- Allow specification with an IP. Currently interface only supported (LU-13642)
- Use imperative recovery logs for client to server connections (LU-10360)
  - Use LNet discovery and IR logs to bring up LNet instead of YAML config files
  - Can add new network to file system without write conf (LU-14608)
- Use hostnames in config llogs (LU-10359)

Conclusion

- Core functionality should be completed for 2.16 release
- Completion by 2.17 release
- New functionality that is the result of this work.
- Once complete and ported to native client the native client will be pushed to Linus

Acknowledgments

This work was performed under the auspices of the U.S. DOE by Oak Ridge Leadership Computing Facility at ORNL under contract DE-AC05-00OR22725.