# Support of large NIDs (IPv6)

Expand to new LNet protocols, May 2024

*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
  - Special thanks to Chris Horn (HPE) and members of the whamcloud team
- The goal is complete foundational LNet support for the 2.16 release
- Lustre 2.17 will complete the support of large NID for everything (full netmask support for Nodemap + NRS, GSS, LNet selftest)

## Progress up to 2.15 LTS

- Ticket LU-10391 opened Feb 2016
- Discussion with Linux community about native client acceptance included IPv6 support.
  - 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
- User land tool don't support large NIDs

## Where is LNet at today ?

- All required LNet functionality finished
  - lnetctl import, lnetctl export, lnetctl net, etc
    - Only visible change to lnetctl / lctl is taking large NID strings
  - Merged most pre multi-rail APIs with multi-rail APIs (LU-10003)
    - lctl fail_nid, lctl net fault, lctl conn_list, lnd peer handling (LU-5960 ) are still missing but some patches exist. Both fail_nid and net_fault exist for testing.
  - socklnd support complete. o2iblnd is in the works (LU-17743)
  - Internal code changes
    - Migrate to Netlink / YAML API
      - Allows changing userland interface without API breakage
      - Much more robust YAML handling (LU-17719 for example)
      - Support of very large setups (LU-14391 for example)
- Sanity-lnet testing is starting

# What is missing from LNet?

- LNet UDSP support
- Support large NIDs with module parameters : ip2nets + routes (LU-17457)
- LNet selftest
  - Internal move to Netlink API (LU-8915)
  - Implement YAML configuration file support (LU-10975)
- Bug fixes that are needed to land before 2.16 release
  - Support hostname with some of the lnetctl commands [ping] (LU-17629)
  - Support netdevice events for health state for IPv6 (LU-17460)
  - Others to be discovered when testing.

# Where is Lustre large NID support at ?

- Able to mount lustre with large NIDs (IPv6)
  - mount -t lustre fe80:f68:45bd:7b60:e933@tcp /mnt/lustre
  - Large NID support for failover nodes
  - Handle strings better (LU-17367)
- Basic Large NID support for nodemap
  - No NID range support using netmask (Only netmask /64 is supported)
- NID export hash supports large NID strings
  - lctl get_param mdt.*.exports.$NID.hash
- UUID with large NIDs supported (LU-13340)
  - MGC UUID ("MGC"NID"_0) string can overflow has been reported *

# What Large NID features does Lustre still need?

- Completed work
  - Enable netmask for IPv6 NID ranges (LU-14288) (2.16 ??)
    - Needed kernel side and user land level
    - Impacts mount strings, nodemap, noroot_squash, NRS TBF
  - Changelog support (LU-13308)
    - Patch is nearly complete (2.16)
- Work slated for 2.17
  - GSS support (LU-17273)
  - Update sptlrpc to handle large NIDs (LU-10937)
  - Update l_getidentity to handle large NIDs

## New future Lustre functionality added and coming

- Completed work
  - Allow NI setup with an IP. Currently interface only supported (LU-13642)
    - lnetctl net add –net tcp –nid 10.0.0.1@tcp
  - LNet discovery in background (LU-14668)
- Future work
  - Mapping hostname@nettype to many addresses (LU-16738)
    - mount –t  myhost@tcp:/lustre /mnt/lustre
  - 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)
  - Remove NIDs from config llogs (LU-10359)
  - Allow more than 32 NIDs for ptlrpc connections.

## Conclusion

- Core functionality 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.