LNet Roadmap & Development

Amir Shehata
Lustre* Network Engineer
Intel High Performance Data Division

Outline

- LNet Roadmap
- Non-contiguous buffer support
- Map-on-Demand re-work

LNet Roadmap (2.12)

LNet Health

- Increase LNet’s resiliency
- Re-transmit messages on different available local and remote interfaces
- Monitor various LND failures
- Monitor PUT/GET responses, ACK/REPLY respectively, and timeout if not received
- Documentation
  - Scope and Requirements Document: https://wiki.hpdd.intel.com/display/LNet/LNet+Health
  - High-level Design: https://wiki.hpdd.intel.com/display/LNet/LNet+Health+HLD
- Implementation will take place on the Multi-Rail branch

LNet Sysfs Interface (2.12)

- Currently `lnetctl` uses IOCTL to collect statistics and configure the systems
- Move to using sysfs interface for keeping and querying statistics
- Expose more LNet, o2iblnd and socklnd statistics
- Present them in YAML format
- Documentation
  - Scope & Reqs: https://wiki.hpdd.intel.com/pages/viewpage.action?pageId=65700164
  - HLD: https://wiki.hpdd.intel.com/display/LNet/Sysfs+Interface+HLD
  - Test Plan: https://wiki.hpdd.intel.com/display/LNet/Sysfs+Interface+Test+Plan

Multi-Rail User Defined Policies (2.13)

- Fine tune Multi-Rail’s selection algorithm
- Allow specifying preferences of Network and Network Interfaces
- Documentation
  - Scope & Requirements
    - https://wiki.hpdd.intel.com/display/LNet/Multi-Rail+User+Defined+Policies
  - High-level Design
    - https://wiki.hpdd.intel.com/display/LNet/User+Defined+Selection+Policies

Multi-Rail User Defined Policies - Rules

- LNet Network priority rule
  - Assigns a priority to a network
  - During selection the network with the highest priority is preferred
- Local NID rule
  - Assigns a priority to a local NID within an LNet network
  - NID is preferred during selection
- Remote NID rule
  - Assigns a priority to a remote NID within an LNet network
  - NID is preferred during selection
- Peer-to-peer rules
  - Associates local NIs with peer NIs
  - When selecting a peer NI to send to, the one associated with the selected local NI is preferred

LNet Unit Test Framework (2.13)

- Complex LNet features in development need to be unit tested
- These unit tests need to be repeatable for regression
- Use python for writing test scripts
- Interface with `lnetconfig` library to configure and query LNet
- Interface with `lnet_selftest` to perform complex functional tests
- Will be integrated with the current Autotest system
- Documentation
  - Scope & Requirements
    - https://wiki.hpdd.intel.com/display/LNet/LNet+Unit+Test+Infrastructure+%28LUTF%29+Requirements
  - High-level Design
    - https://wiki.hpdd.intel.com/display/LNet/LUTF+High+Level+Design

LNet Documentation

- Create “Scope & Requirements” and “HLD” documents for all new projects
- Need detailed design documentation for LNet
- Makes it easier for new developers to understand the code
- Detailed-level design type documentation is incrementally being added:
  - Connection Management
  - Map-on-demand, etc.
- https://wiki.hpdd.intel.com/display/LNet/LNet+Documentation

Adapt o2iblnd to latest RDMA changes

New Fast Memory Registration API

- https://www.openfabrics.org/images/eventpresos/2016presentations/204KernelVerbs.pdf

CQ Polling API

- https://review.whamcloud.com/#/c/27028/
- Simplify completion queue polling and interrupt handling
- Resolve the error completion unreliability

Draining QP

- Don’t have to wait for WR to complete to destroy a QP
- Current method in o2iblnd risks waiting indefinitely

LNet Router Testing

- Multiple requests received to outline how to test LNet routers
- A test plan has been created

  https://wiki.hpdd.intel.com/display/LNet/LNet+Router+Testing
- Need to translate the test plan into LUTF test scripts

`lnet_selftest` Improvements

- Improve the `lnet_selftest` user interface
  - Provide parameters and results using YAML format
- Allow users to specify different traffic flows
- Better integration with the LUTF for more comprehensive functional testing

Multi-hop route failure detection

- LU-9238 – entered by Cray*
- Current proposal on the ticket
  - Extend LNet ping to include route up/down status
  - Peers get route status from their next hop
  - Percolate to peers that use that route
- Gossip protocol
  - Gossip protocol should be used as a general solution for Network Discovery
    - This should also handle the route health case
  - Look into the potential of integrating it into Lustre

IPv6 Support

- Expand NIDs to support IPv6 addresses
- Will break compatibility with older LNet versions
- Potentially use LNet routers to route between IPv4 and IPv6 networks

Recent Developments

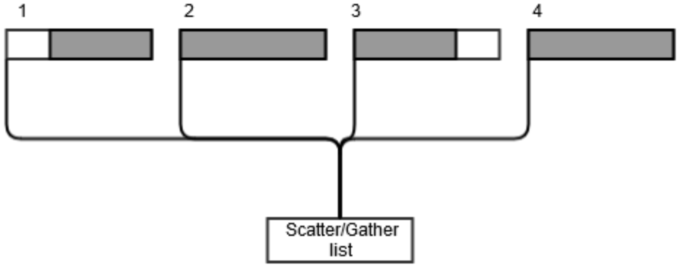
Non-contiguous Buffer Support

Problem Statement

- FMR and FastReg (FRMR) do not support non-contiguous RDMA buffers
- O2iblnd didn’t check if the RDMA buffers were contiguous or not.
- Changes in OSP resulted in non-contiguous buffers being passed to the o2iblnd
- Buffers were not RDMAed properly resulting in corrupted data and operation failure
- Global memory regions usage did not exhibit this problem
- RHEL 7.4+ removed Global memory regions and use FMR/FRMR exclusively

Overview of Memory Mapping

- Scatter/gather list is formed to point to the pages to be RDMAed
- `ib_dma_map_sg()` maps the scatter/gather list in to the DMA memory space
- An RDMA descriptor structure describes each fragment to be RDMAed
- RHEL 7.3 and earlier, global memory regions were supported and used by o2iblnd
- Since the RDMA descriptor described all the fragments correctly there was no problem



The Problem

- Since RHEL 7.4, global memory regions support ceased
  - FMR/FRMR is now the default for `o2iblnd`
- FMR/FRMR pools are used
  - Fragments are mapped into the FMR/FRMR memory region
  - RDMA descriptor describes it as one large fragment
- However, this exposed a problem when the fragments were not contiguous
  - Page 3 had a gap, which resulted in some data from page 4 not being RDMAed

o2iblnd Behavior Changes

- This led to a series of patches which brought behavioral changes to o2iblnd:
  - LU-9983 osp: align the OSP request size by 4k
    - Avoids gaps in the IOV buffer to RDMA
  - LU-9983 ko2iblnd: allow for discontiguous fragments
    - Describe each buffer in the RDMA descriptor
    - Problem with different map_on_demand settings
  - LU-10089 o2iblnd: use IB_MR_TYPE_SG_GAPS
    - MLX5 support
    - Drop in performance
  - LU-10129 lnd: rework map_on_demand behavior

Full Solution

- Do not make map-on-demand configurable
- Set the maximum number of fragments supported on a QP to 256
- Continue negotiation with the peer to handle older versions
  - Could have map-on-demand < 256 and therefore QP's WRQ size could be less
- Detect if fragments passed to o2iblnd are non-contiguous
- FMR requires specifying each non-contiguous fragment in the RDMA descriptor
  - Could fail if the negotiated fragments on the QP is less than the fragments buffer number
  - Early failure with clear message to easily detect the situation
- If FRMR with GAPS then handle non-contiguous fragments, or fail RDMA write as above

Fallout

- Since we use the maximum number of fragments, 256, QP creation could fail
- Reduce the total number of fragments and attempt to recreate the QP
- OPA TID-RDMA uses too much memory.
  - OPA TID-RDMA statically allocates memory based on provided values
  - With conns-per-peer set to 4 memory consumption is multiplied.
    - Servers with many QPs run into OOM errors. We had several bugs related to this issue
- LU-10875 – open to track
  - Devise a method to use fewer WRs
- The map-on-demand rework is available in 2.11

Conclusion

Major LNet projects:

- LNet Health – Lustre 2.12
- LNet Sysfs – Lustre 2.12
- Multi-Rail User Defined Policies – Lustre 2.13
- LNet Unit Test Framework – Lustre 2.13

O2iblnd non-contiguous buffer support

Legal Notices and Disclaimers

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY RELATING TO SALE AND/OR USE OF INTEL PRODUCTS, INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life-saving, life-sustaining, critical control or safety systems, or in nuclear facility applications.

Intel products may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

This document may contain information on products in the design phase of development. The information herein is subject to change without notice. Do not finalize a design with this information. Intel may make changes to dates, specifications, product descriptions, and plans referenced in this document at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.

No computer system can be absolutely secure.

Intel Corporation or its subsidiaries in the United States and other countries may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights that relate to the presented subject matter. The furnishing of documents and other materials and information does not provide any license, express or implied, by estoppel or otherwise, to any such patents, trademarks, copyrights, or other intellectual property rights.

Performance estimates or simulated results based on internal Intel analysis or architecture simulation or modeling are provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice Revision #20110804.

Intel® Advanced Vector Extensions (Intel® AVX)* provides higher throughput to certain processor operations. Due to varying processor power characteristics, utilizing AVX instructions may cause a) some parts to operate at less than the rated frequency and b) some parts with Intel® Turbo Boost Technology 2.0 to not achieve any or maximum turbo frequencies. Performance varies depending on hardware, software, and system configuration and you can learn more at http://www.intel.com/go/turbo.

Intel processors of the same SKU may vary in frequency or power as a result of natural variability in the production process.

Intel, the Intel logo, 3D-Xpoint, Optane, Xeon Phi, and Xeon are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

* Other names and brands may be claimed as the property of others.

intel®

LU-9983 osp: align the OSP request size by 4k

- The first solution was to simply ensure that you always give 4K aligned buffers to the o2iblnd
- This would hide the problem
- LNet would still not support non-contiguous fragments.
- Any future feature which would make use of non-contiguous fragments would still expose the problem

LU-9983 ko2iblnd: allow for discontiguous fragments

- Instead of collapsing the RDMA fragments into 1 when using FMR/FRMR, continue to describe them fully
- Theoretically, this should avoid the problem described, but it resulted in a different problem
- Map-on-demand value was used to negotiate the maximum number of fragments on the connection. This could be set to a value between 2 and 256
  - Many deployments set it to 32
  - With LU-9983 1M RDMA buffers would get fragmented into 256 which would exceed the negotiated maximum number of fragment on a connection, leading to RDMA failure
- This solution was not enough to fully solve the problem

LU-10089/LU-10394

- For FRMR Mellanox provides a flag, IB_MR_TYPE_SG_GAPS, when creating the memory regions, which would support RDMA fragments with GAPS
- However, according to Cray testing using IB_MR_TYPE_SG_GAPS had a rather significant performance impact; up to 2 GB/s reduction in performance
- Added a flag to turn on FRMR GAPS support:
  - use_fastreg_gaps
  - It’s 0 by default
  - If set to 1 and the HCA supports FRMR then we create the FRMR memory regions using that flag
- Again this does not address FMR and it’s not a sufficient solution for FRMR

LU-10129 lnd: rework map_on_demand behavior

What's the use of map-on-demand?

- Turn on FMR/FRMR usage
- Determine the max size of the send work request queue (WRQ) per Queue Pair (QP)

How did map-on-demand work? (assuming Global Memory Region support)

- If map-on-demand == 0 use Global Memory Region exclusively
- If the RDMA's number of fragments < configured map-on-demand then use Global Memory regions, otherwise use FMR or FRMR (whichever the HW supports)

The Map-on-demand primary benefit is to reduce the max send work request queue size

RDMA mapping in ko2iblnd

- Looking forward, Global Memory Regions are no longer supported in the kernel
- Map-on-demand usage complicates the code
- No major advantage to having the max_send_wrq for the QPs be configurable
- When using FMR/FRMR only 1 WR is used for the RDMA transfer
- Ideally we'd be using the least number of WRs possible