Portals4 LND Overview

Grégoire Pichon - Atos

05-10-2017

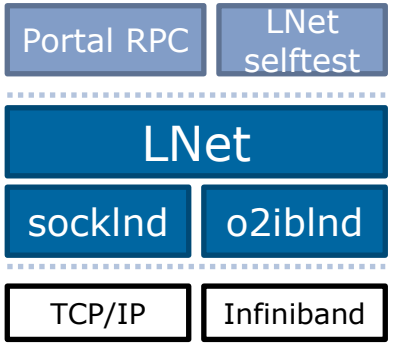
Bull
atos technologies

New Lustre Network Driver

LNet

Lustre Network layer

- communication infrastructure
  - between Lustre clients and servers
- supports many commonly-used network types
  - Infiniband, TCP/IP networks
  - allows routing between networks
- key features
  - high availability
  - recovery
- Lustre Network Driver
  - provides support for a particular network type
  - implements LNet-LND api



Portals 4 LND

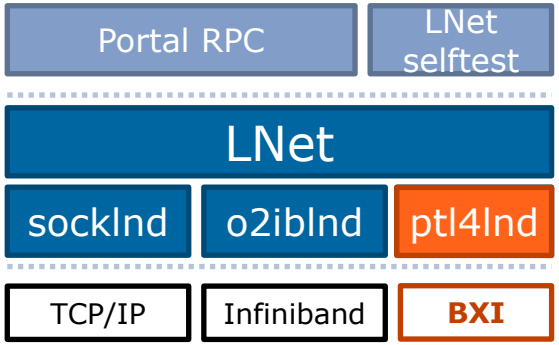
New Lustre Network Driver

- ptl4lnd
  - based on Sandia National Laboratories Portals 4 Network Programming Interface

  LU-8932 lnet: define new network driver ptl4lnd
  - kernel module: kptl4lnd.ko
  - network name: ptlf
  - network adapter identified by device number

```
networks=ptlf0(0),ptlf1(1)
```

- hardware
  - **B**ull e**X**ascale **I**nterconnect



Portals 4

API for high-performance networking

- **Common semantic** for MPI and PGAS
  - Target memory descriptors: "persistent" (one-sided ops.) or "use once" (two-sided ops.)
  - Rich operations library: Put, Get, Swap, FetchAtomic
- **Full hardware offloading** from the host CPU
  - Performance is not impacted by heavy load on the host CPU
  - Triggered operations for protocol offloading (collectives, etc.)
- **Non-connected reliable protocol**
  - No connection establishment time
  - Constant memory footprint whatever the number of communicators
- **Unexpected messages aggregation**
  - N to 1 communications optimization – single buffer for multiple messages

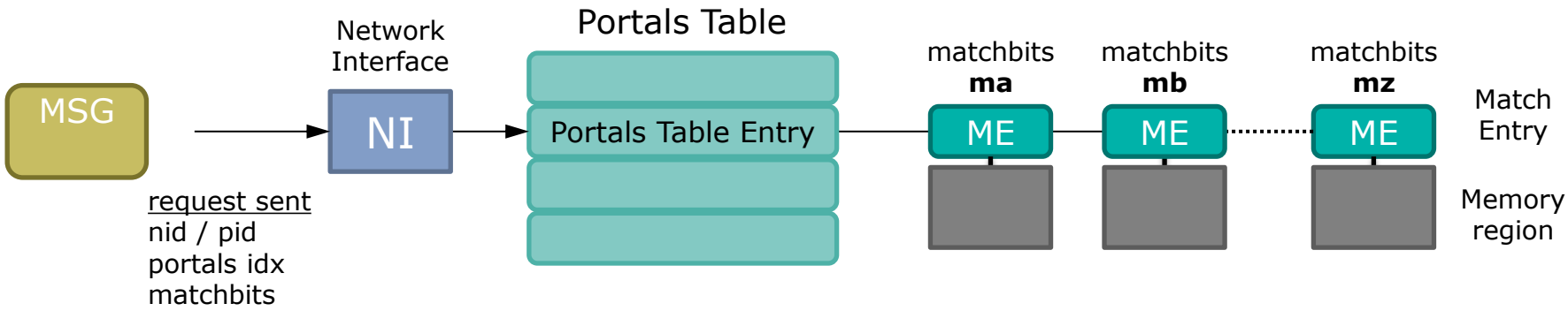
Sandia National Laboratories

portals

Bull atos technologies

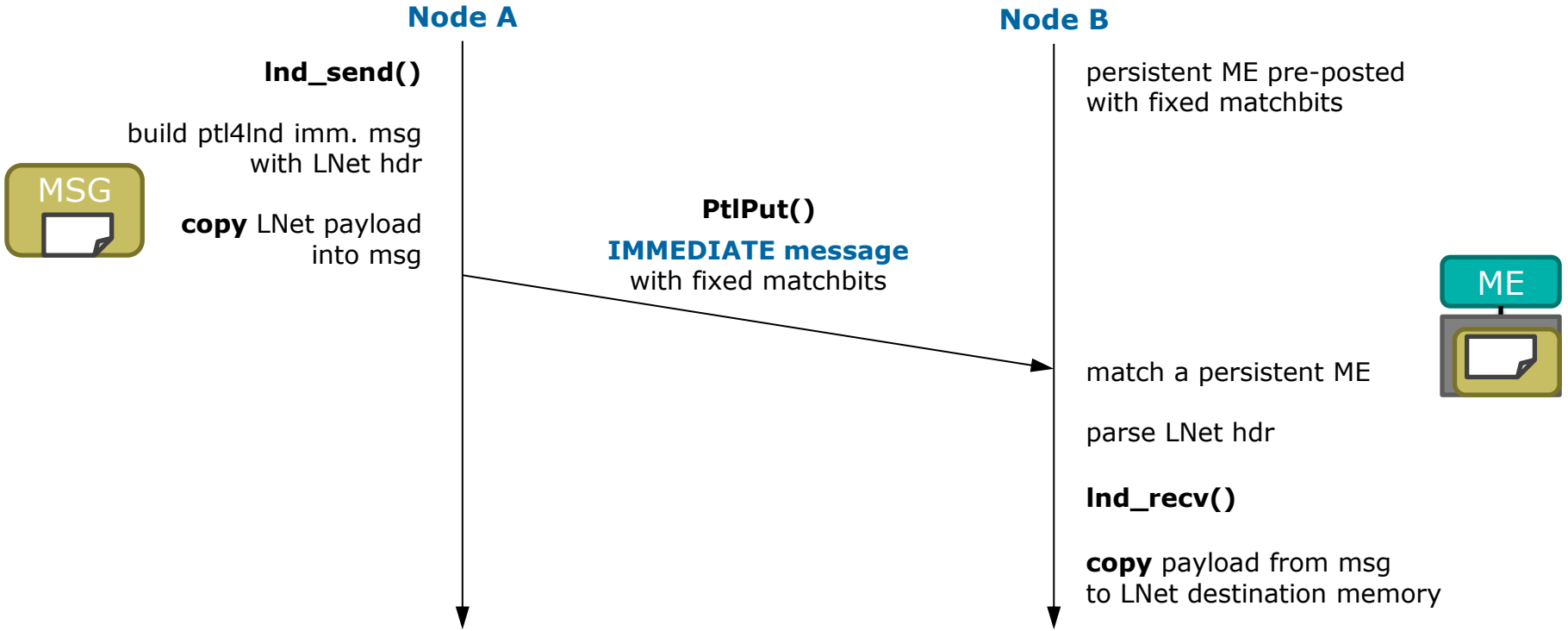
Portals 4

Communication Model



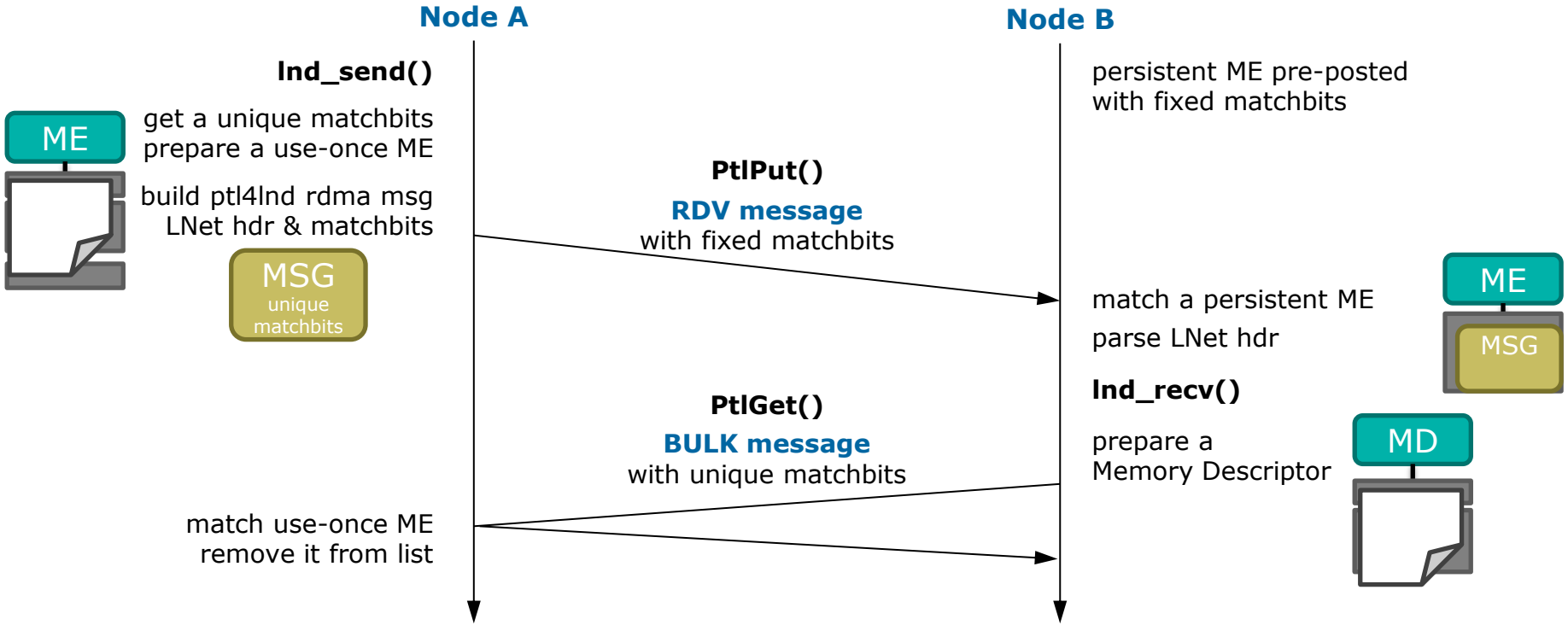
Ptl4lnd internals

Small data transfer (from node A to node B)



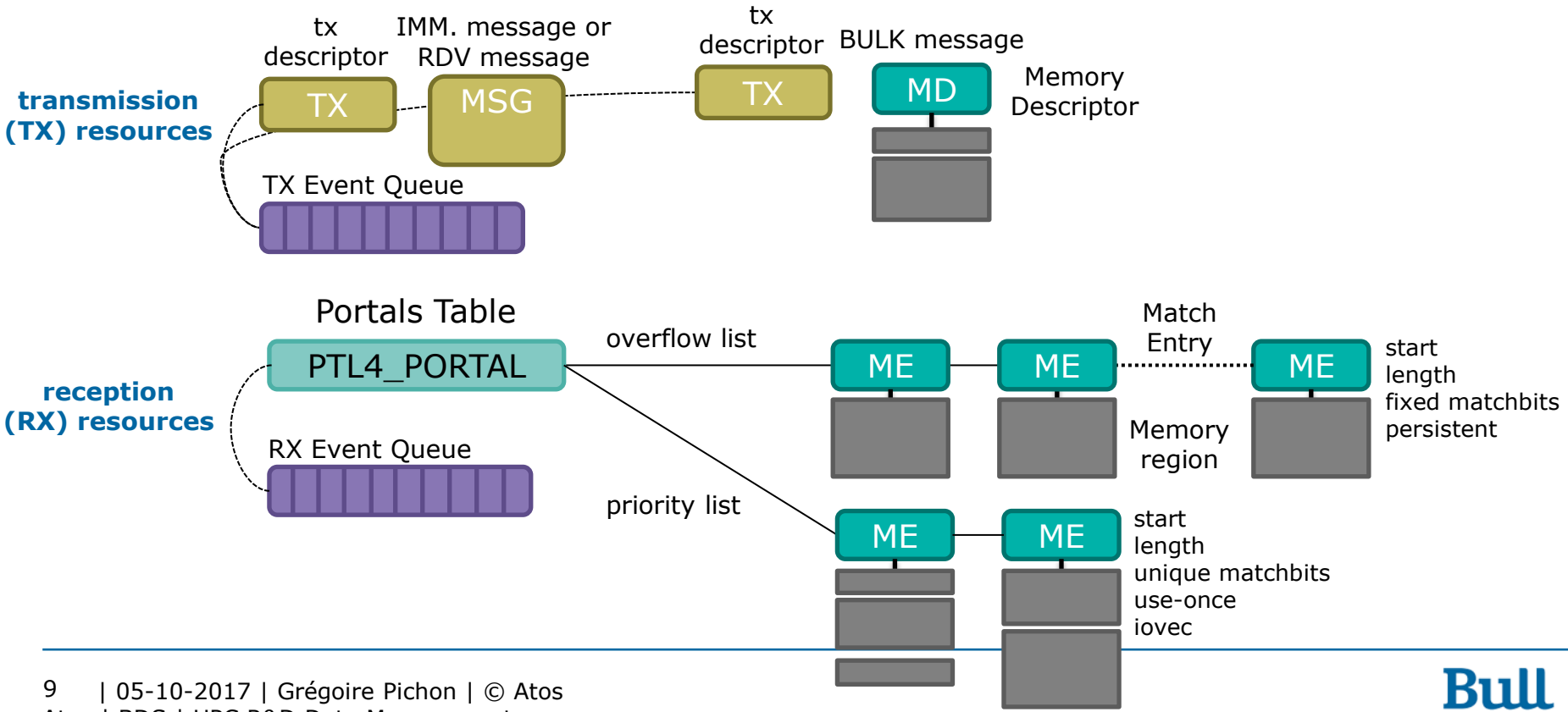
Ptl4lnd internals

RDMA data transfer (from node A to node B)



Ptl4Ind internals

resource management



Ptl4lnd parameters and tuning

- credits (32) and peercredits (8)
  - number of concurrent sends to all and to one peer
  - need uniform setting between cluster nodes
  - higher value allows higher bandwidth but consumes more resources
- checksum (off)
  - check integrity of non-bulk messages
- nscheds (2)
  - number of scheduler threads that handle RX buffer, TX finalization, …

Ptl4lnd debug and statistics

- slots in event queues
  - number of reserved slots
  - total number of slots

```
# cat /sys/kernel/kptl4lnd/0/events
tx events: 14 reserved / 65536
```

- peer status
  - nid-pid, state, credits, alive time

```
# cat /sys/kernel/kptl4lnd/0/peers
<NID>:<PID> |  <state> |  <credits>   |  <alive>
   24:9     |  ACTIVE  |  8 / 1 + 7   |  253s
   25:9     |  ACTIVE  |  8 / 3 + 5   |  212s
```

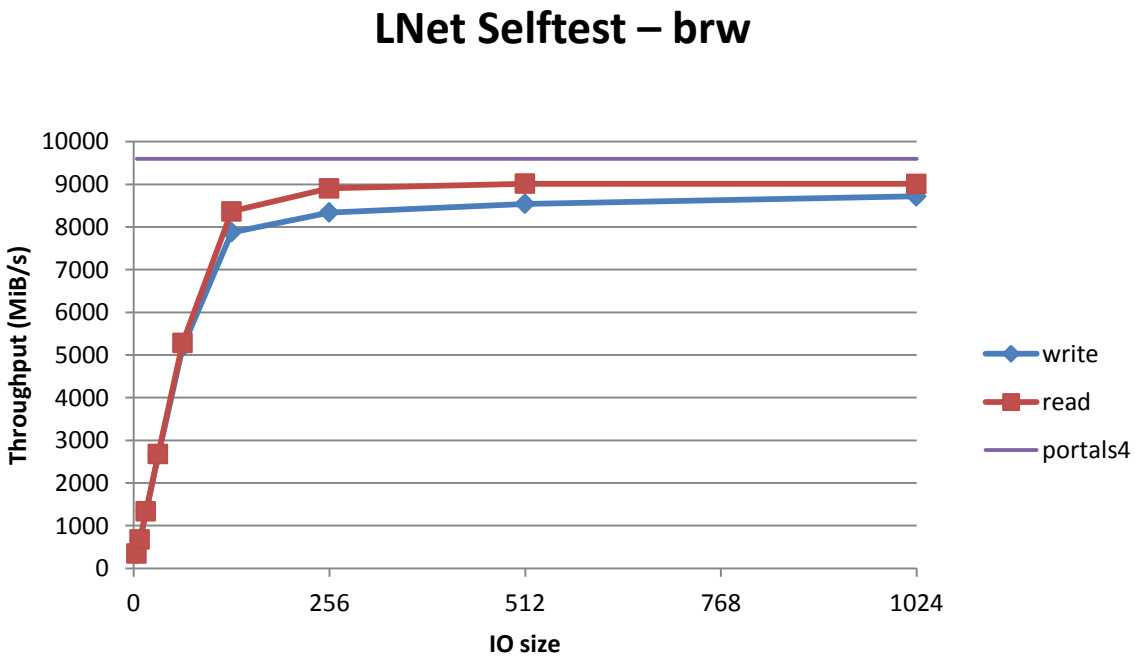
- statistics
  - number of TX of different types

```
# cat /sys/kernel/kptl4lnd/0/stats
    PUT  GET  IMM NOOP HELL  NAK BULK
tx    3    0   23    1    0    0    0
```

- dump TX and RX states

Ptl4lnd performance achievements

LNet performance

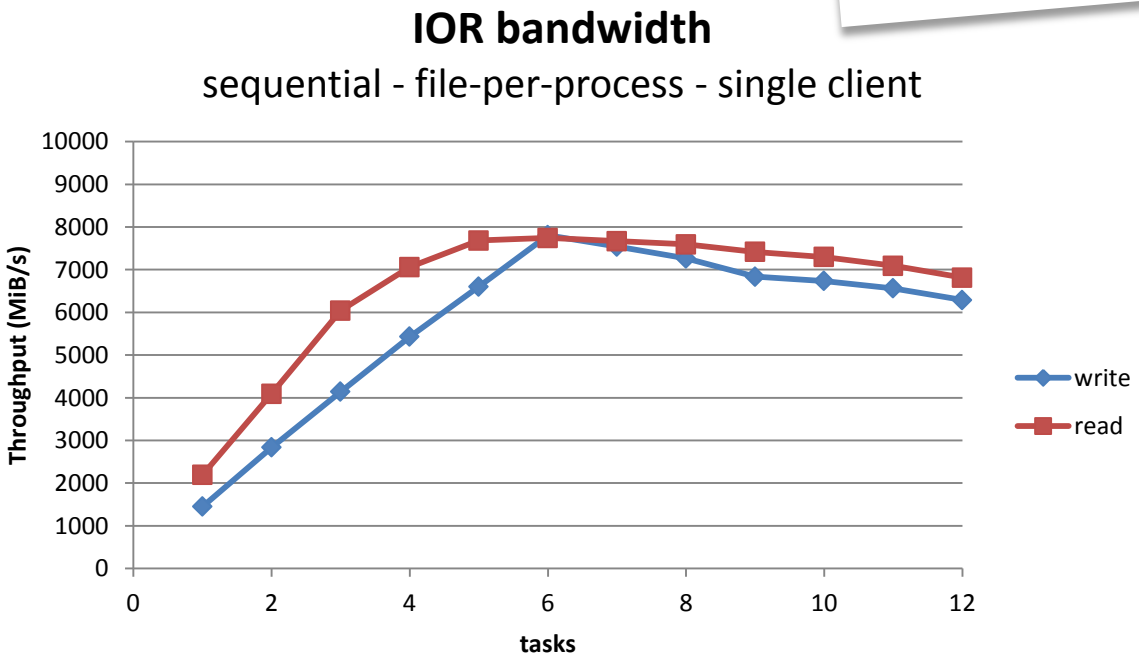


Lustre IEEL 2.7.21
BXI 1.2 NIC – SWITCH - NIC

Ptl4Ind performance achievements

Lustre performance

WORK IN PROGRESS



Lustre IEEL 2.7.21
BXI 1.2 NIC – SWITCH - NIC

Bull
atos technologies

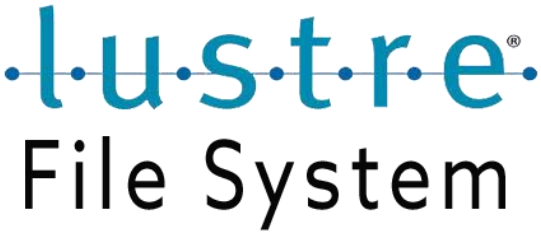
Ptl4lnd development

Next steps

- Finalize LND code
  - perform some optimizations
  - handle remaining corner cases

- Ensure compatibility with recent LNet changes
  - currently runs on Lustre IEEL 2.7.21
  - build and test on Lustre master

- **Push code to the Lustre community**




2
Development Challenges

LNet – LND interface

looks simple …

```
struct lnet_lnd ptlflnd = {
        .lnd_type       = PTL4LND,
        .lnd_startup    = ptlf_startup,
        .lnd_shutdown   = ptlf_shutdown,
        .lnd_ctl        = ptlf_ctl,
        .lnd_send       = ptlf_send,
        .lnd_recv       = ptlf_recv,
        .lnd_query      = ptlf_query,
};
```

- register to LNet
  - lnet_register_lnd()
  - lnet_unregister_lnd()
- provide a lnet_lnd structure
  - lnd id (SOCKLND, O2IBLND, LOLND, GNILND, PTL4LND)
  - startup/shutdown network communication on the network interface
  - send/receive LNet message on the network interface
  - notification /query on peer health / aliveness
  - control commands
- use LNet callbacks
  - lnet_parse(), lnet_finalize(), lnet_set-reply_msg_len(), …

LNet – LND interface

… but still some interrogations

- lack of documentation
  - routine semantic
    - when should a LND call lnet_notify() ?
    - what LNet callbacks should/shall a LND use ?
  - parameters description
  - call context

```
 /* Start receiving 'mlen' bytes of payload data, skipping the following
  * 'rlen' - 'mlen' bytes. 'private' is the 'private' passed to
  * lnet_parse().  Return non-zero for immedaite failure, otherwise
  * complete later with lnet_finalize().  This also gives back a receive
  * credit if the LND does flow control. */
int (*lnd_recv)(struct lnet_ni *ni, void *private, struct lnet_msg *msg,
                int delayed, unsigned int niov,
                struct kvec *iov, lnet_kiov_t *kiov,
                unsigned int offset, unsigned int mlen, unsigned int rlen);
```

Tests and bringup

- LNet selftest is not convenient for LNet debugging or LND bringup
  - test infrastructure is setup with … LNet messages
  - not designed to perform 1 LNetPut() or LNetGet() operation
  - cannot test
    - specific (exotic) memory region transfers
    - error cases (non matching ME, short MD)

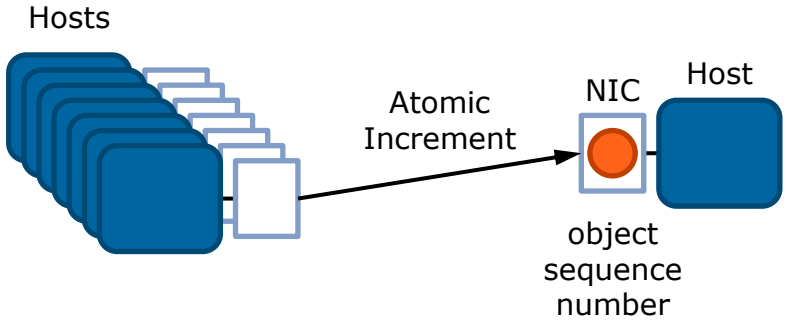
- development of a LNet test kernel module
  - define a pseudo device for ioctl command
    - post LNet ME-MD for reception
    - launch LNet data transfer
  - also post permanent LNet ME-MD


3
Opportunities offered by modern Interconnects

Network advanced features (1)

… that Lustre could benefit

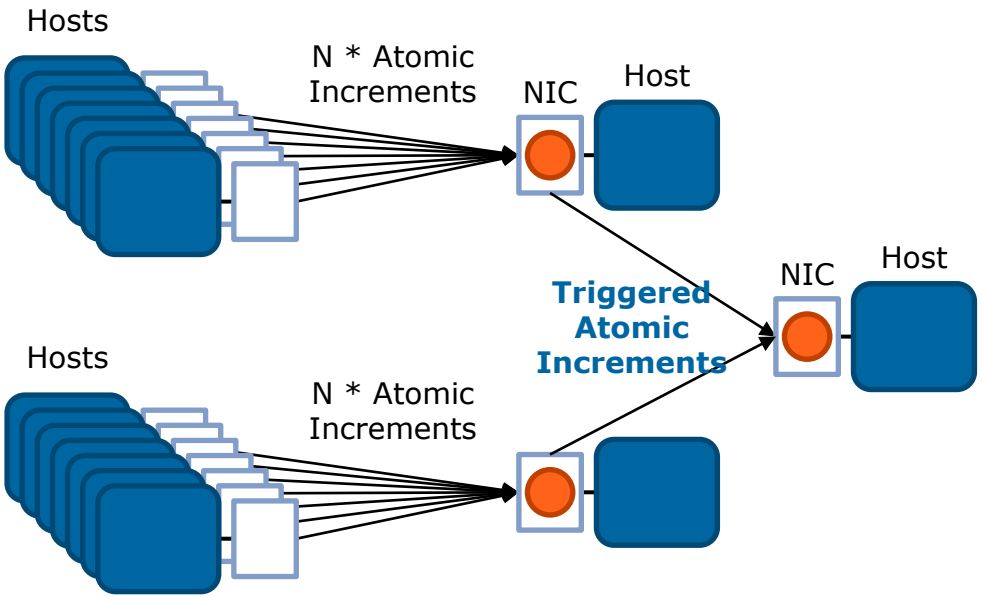
- **Atomic operations**
  - atomically read and update data located in remote memory regions
  - host bypass, low latency
  - operations
    - min, max, sum, prod, or, and, swap, conditional swap, …
  - usage
    - distributed lock
    - object sequence number
  - LNetAtomic(), LNetFetchAtomic(), LNetSwap()



Network advanced features (2)

… that Lustre could benefit

- **Triggered operations**
  - setup operations triggered by incoming messages
  - host bypass, low latency
  - usage:
    - tree based reduction
    - recovery synchronization
  - LNetTriggeredPut(), LNetTriggeredAtomic()



Thanks

gregoire.pichon@atos.net

Bull
atos technologies