LNET Multi-rail Improvement

Tatsushi Takamura and Shinji Sumimoto, Ph.D.
Next Generation Technical Computing Unit
FUJITSU LIMITED   Sept. 24th, 2019
Outline of This Talk

- Backgrounds

- Introduction of FEFS IB Multi-rail and Lustre LNet Multi-rail

- Evaluation of LNet Multi-rail
  - The result of Evaluation
  - Problems and How to fix them

- Summary and Conclusion
Fujitsu developed FEFS IB Multi-rail and operated on K computer and other HPC systems for over 7 years.

IB Multi-rail Features:

- High availability even if a single point of IB failure occurs
- High throughput by using multiple IB interfaces
- Various configurations
  - Not only Symmetric connections but also Asymmetric connections
Lustre community is now developing similar Multi-rail features on LNet level

- LNet Multi-rail, LNet Network Health, Etc…

However the development is still going on

Therefore, we have evaluated the Lustre LNet Multi-rail assuming the same features of FEFS IB Multi-rail

- In order to give feedback to current LNet Multi-rail implementation
FEFS IB Multi-rail (presented at LAD14)

- **FEFS Approach:** Add IB Multi-rail function into Lustre network driver (o2iblnd).
  - All IB I/F on the client can be used to communicate with a server.
  - All IB connections are used by round-robin order.
- **Continue communication when single point of IB failure occurs.**
  - All IB connections are used by round-robin order by each requests.

---

![Diagram](attachment:image.png)

**Left:**
- Clients A
- Clients B
- IB SW
- MDS/OSS
- Multi-rail by o2iblnd

**Right:**
- Clients A
- Clients B
- IB SW
- MDS/OSS
- Failure
- No failover required

---

Copyright 2019 FUJITSU LIMITED
Lustre LNet Multi-rail and LNet Health

- **LNet Multi-rail**: Introduced in Lustre 2.10 (LU-7734)
  - Using multiple interfaces including Ethernet and InfiniBand

- **LNet Network Health**: Introduced in Lustre 2.12 (LU-9120)
  - Detecting network failures of local interface, remote interface, network timeouts and etc.
  - Switching and resending among different interfaces

Flow of IB Multi-rail

Flow of LNet Multi-rail

The basic idea is the same as FEFS IB Multi-rail
(The difference is LND level or LNet level)
Overview of LNet Health

- **Detecting device status**
  - A local Network Interface (NI) is marked fatal, if the device has gone into a fatal
    - ex. IB_EVENT_DEVICE_FATAL, IB_EVENT_PORT_ERR

- **Maintaining health value**
  - Each NI (both local and remote) has a health value (HV)
  - HV is decremented when communications fail and incremented when succeeds

- **Controlling path selection**
  - Selecting the healthiest local NI by health value
    - Fatal NI is removed from the candidates
  - Selecting the healthiest remote NI which belong to the same network which the local NI connected
  - Communicating using the selected NIs
Evaluation of LNet Multi-rail

Evaluation Items

### I/O Continuity

<table>
<thead>
<tr>
<th>Check Items</th>
<th>How to check</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/O failover works correctly (No I/O hang)</td>
<td>Single NI failure on server side</td>
</tr>
<tr>
<td>against single NI failure</td>
<td>Single NI failure on client side</td>
</tr>
<tr>
<td>No I/O error after recovery</td>
<td>All NI failures on server side</td>
</tr>
<tr>
<td></td>
<td>All NI failures on client side</td>
</tr>
</tbody>
</table>

*NI failure: rejecting IB cable from HCA*

### I/O Throughput

<table>
<thead>
<tr>
<th>Check Items</th>
<th>How to check</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same I/O performance of FEFS IB Multi-rail</td>
<td>Comparing LST and IOR: Server only configuration</td>
</tr>
<tr>
<td></td>
<td>Comparing LST and IOR: Server and Client configuration</td>
</tr>
</tbody>
</table>
## Evaluation Result Summary

<table>
<thead>
<tr>
<th>Check Items</th>
<th>IB(FEFS)</th>
<th>LNet(Lustre)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/O failover works correctly (No I/O hang) against single NI failure</td>
<td>✔</td>
<td>X</td>
<td>We found 4 problems</td>
</tr>
<tr>
<td>No I/O error after recovery</td>
<td>✔</td>
<td>✔</td>
<td>OK</td>
</tr>
<tr>
<td>Same I/O performance of FEFS IB Multi-rail</td>
<td>✔</td>
<td>✔</td>
<td>OK</td>
</tr>
</tbody>
</table>

### Issue Description

<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No.1</td>
<td>Unable to detect IB hardware failure (NI is not marked fatal). This may cause selecting a failed NI for sending.</td>
</tr>
<tr>
<td>No.2</td>
<td>Decrementing a health value of normal NI. This may cause sending messages to a failed NI.</td>
</tr>
<tr>
<td>No.3</td>
<td>Unable to use Multi-rail on asymmetric NIs.</td>
</tr>
<tr>
<td>No.4</td>
<td>After recovery of NI failure, the NI is not used for a while (1000 sec).</td>
</tr>
</tbody>
</table>
Problem #1: Unable to detect IB hardware failure

**Issue**

- LNet health unable to detect IB hardware failure (NI is not marked fatal)
- This may cause extra retry messages because of sending messages from a failed NI

**Why came from?**

- IB driver notifies “IB event” if IB device failure occurs
- But in current implementation, o2iblnd only detects these event from “QP event”
Problem #1: Proposed Solution

- Adopting FEFS IB Multi-rail scheme
  - Using an event handler (ib_register_event_handler) which kernel provides
  - The NI is set fatal correctly and will not be used

- Effect
  - LNet health can select correct local NI without message retry

LNet health detect IB failure and mark fatal
-> The NI will not be used
Problem #2: Decrementing health value of normal NI

**Issue**

- LNet health could decrement a health value (HV) of normal NI
  - Though remote NI failure occurs, HV of local NI would be decremented
- This may cause extra retry messages because sending messages to a failed NI

**Why came from?**

- LNet health unable to detect local NI failure (resolved by problem #1)
- LNet health decrement local health value if connection failed
Problem #2: Proposed Solution

- LNet health can detect local NI failure using solution of problem #1
  - Failure local NI will not be used, so we can judge that remote NI is the cause of connection fail
- We modify to decrement of remote health value

Effect

- LNet health can set health value correctly
  - LNet health can select correct remote NI and reduce message resending
Problem #3: Unable to use Multi-rail on asymmetric NIs

- Issue
  - If the sending node is Multi-rail and the receiving node is non-Multi-rail, the sending node uses always the same NI.
  - Even if the sending NI is blocken, the blocken NI is used.

- Why came from?
  - This seems to be a specification on asymmetric Multi-rail environment.

Normal State
- remote (non-Multi-rail)
- only use specific NI
- local (Multi-rail)

Failure State
- remote (non-Multi-rail)
- don’t use another device
- local (Multi-rail)
Problem #3: Proposed Solution

- Switching another normal NI if LNet health detects NI failure (by using solution of problem#1)
  - These configurations are common for our users

Effect

- LNet can continue communicating unless all NI failure
Problem #4: After recovery of NI failure, the NI is not used

- **Issue**
  - It takes for a while (1000s) to recovery health value and to multiple NIs

- **Why came from?**
  - The health value will be decremented by periodically by recovery process if the NI failed
    - Always fail and health value is decremented (the health value hit the floor soon)
Problem #4: Proposed Solution

- Stopping health value decrementing after a device failure is detected
  - Not decrementing by periodically by recovery process
  - Better to use quickly because recent IB is stable and high quality

- Effect
  - Able to use the NI in a few seconds after device recovery

- Comment form community
  - Could a better approach be a more weighted recovery in consideration of flapping hardware (LU-12292)
  - This idea sounds good
    - Recover in 15 sec is reasonable
<table>
<thead>
<tr>
<th>Issue</th>
<th>Description</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>No.1</td>
<td>Unable to detect IB hardware failure (NI is not marked fatal).</td>
<td>We handled IB hardware failure and a path is selected without waiting for health value decremented</td>
</tr>
<tr>
<td>No.2</td>
<td>Decrementing a health value of normal NI.</td>
<td>We set health value appropriately and reduced extra resending</td>
</tr>
<tr>
<td>No.3</td>
<td>Unable to use Multi-rail on asymmetric NIs</td>
<td>We switched switch another normal NI to avoid for the system to become unusable</td>
</tr>
<tr>
<td>No.4</td>
<td>After recovery of NI failure, the NI is not used for a while (1000sec)</td>
<td>We stopped health value decrement at recovery processing to use the NI in a few seconds after device recovery</td>
</tr>
</tbody>
</table>
Conclusion

- We evaluated LNet Multi-rail and improved it

- Fujitsu continues to improve Lustre features and give feedbacks

- Any questions?
shaping tomorrow with you