Getting the best from Lustre in a NUMIOA and multirail InfiniBand environment

sebastien.buisson@bull.net
Getting the best from Lustre in a NUMIOA and multirail InfiniBand environment

sebastien.buisson@bull.net
Lustre with NUMIOA and multirail IB

- What is NUMIOA?
- What is multirail IB?
- How does Lustre behave on a NUMIOA server with multiple Infiniband interfaces?
- Evolutions for the multirail part
- Evolutions for the NUMIOA part
- Application to the OSS server
- Coming next...
What is NUMIOA?

Non-Uniform Memory Access + Non-Uniform IO Access = Non-Uniform Memory and IO Access
What is NUMIOA?

--- delimits a physical node
What is NUMIOA?

Non-Uniform Memory Access

--- delimits a NUMA node
What is NUMIOA?

Non-Uniform IO Access

--- delimits a NUMIOA node
What is NUMIOA?

- **Non-Uniform Memory and IO Access**

---

delimits a NUMIOA node
What is multirail IB?

- Taking advantage of several Infiniband interfaces
  - on clients and servers
  - for bandwidth aggregation

- Our goal:
  - Lustre network bandwidth = \( \sum \) individual link bandwidths
How does Lustre behave on a NUMIOA server?

Our testbed:

- Nehalem EP
- IOH (Tylersburg)
- IOH (Tylersburg) Legacy
- QPI
- 12.8GB/s each direction
- 8x, 16x, 8x
How does Lustre behave on a NUMIOA server?

Elementary performance in R&D labs without Lustre

Methodology
- Disk IO bandwidth measured with xdd
- IB bandwidth measured with qperf
- run independently
Simulated read, threads correctly localized

Nehalem EP

QPI

QPI

IOH (Tylersburg)

IOH (Tylersburg) Legacy

IB

FC

3190 MB/s

2970 MB/s
Simulated write, threads correctly localized

Nehalem EP

QPI

QPI

IOH (Tylersburg)

IOH (Tylersburg) Legacy

IB

FC

2670 MB/s  3249 MB/s
Simulated read, threads NOT correctly localized

Nehalem EP
QPI
QPI
IOH (Tylersburg)

IOH (Tylersburg) Legacy
IB
FC
3160 MB/s
1939 MB/s
Simulated write, threads NOT correctly localized

Nehalem EP  QPI  Nehalem EP  QPI

IOH (Tylersburg)  IOH (Tylersburg) Legacy

IB  FC

1680 MB/s  3249 MB/s
How does Lustre behave on a NUMIOA server?

“real life” case: customer cluster

- **IB switch Voltaire 9036**
- **IB QDR 4x**: 3.2 GB/s per link
- **MDS**
  - R423E2
  - FC8, 1 link
- **OSS-1**
  - R423E2
  - FC8, 4 links
- **OSS-2**
  - R423E2
  - FC8, 4 links
- **Singlet 1**
  - 24 SATA
  - FC8, 4 links
  - 0.78 GB/s per link
  - 3.2 GB/s total
- **Singlet 2**
- **DDN 9900**: 5.3 GB/s (22 MB/s/disk)

©Bull, 2010

Lustre with NUMIOA and multirail IB
How does Lustre behave on a NUMIOA server?

With Lustre 1.6:

- One Compute Node gets 3 GB/s
  - OK, network bandwidth on the client is the limiting factor
- Two Compute Nodes get 4.0 GB/s
  - expected 5 GB/s or more

- What’s happening in the OSS servers?
  - Non Uniform IO Access
  - Non Uniform Memory Access
NUMIOA phenomenon

Lustre threads

Nehalem EP

12.8 GB/s
Each direction

QPI

IOH (Tylersburg)

Legacy

Dual FC8

Dual FC8

Legacy

8x 16x 8x

8x 16x 8x

QPI

QPI

QPI

QPI
NUMIOA phenomenon

Lustre threads

Nehalem EP

QPI

12.8 GB/s

Each direction

QPI

IOH
(Tylersburg)

Legacy
(Tylersburg)

QPI

8x 16x 8x

8x 16x 8x

2.0 GB/s
NUMIOA workaround

Lustre threads

IOH (Tylersburg) 12.8GB/s Each direction IOH (Tylersburg) Legacy

8x 16x 8x 8x 16x 8x

QPI QPI QPI

Dual FC8 Dual FC8 QDR
NUMIOA workaround

Lustre with NUMIOA and multirail IB

IOH (Tylersburg)  QPI  12.8 GB/s Each direction  QPI

Nehalem EP

CPU  QPI  QPI

Legacy

Dual FC8  B QDR

2.6 GB/s

Lustre threads

Nehalem EP

IOH (Tylersburg)

Legacy

Dual FC8  B QDR

8x  16x  8x

8x  16x  8x

©Bull, 2010

Lustre with NUMIOA and multirail IB

Architect of an Open World™
How does Lustre behave on a NUMIOA server with multiple IB interfaces?

- R&D experiments with Lustre 2.0:

![Diagram showing IB interfaces and QPI connections between Nehalem EP and IOH](image)
How does Lustre behave on a NUMIOA server with multiple IB interfaces?
How does Lustre behave on a NUMIOA server with multiple IB interfaces?

Cost: 2

Lustre threads
How does Lustre behave on a NUMIOA server with multiple IB interfaces?

Lustre threads

Cost: 4

Lustre with NUMIOA and multirail IB
How does Lustre behave on a NUMIOA server with multiple IB interfaces?

Lustre with NUMIOA and multirail IB
How does Lustre behave on a NUMIOA server with multiple IB interfaces?
How does Lustre behave on a NUMIOA server with multiple IB interfaces?

Lustre with NUMIOA and multirail IB

Cost: 3

- Nehalem EP to Nehalem EP via QPI
- 12.8GB/s each direction
- Oh (Thiersburg) to IOU (Tylersburg) Legacy
- IB QDR to Dual FC8
- IB QDR to Dual FC8
- IB QDR to Dual FC8
- IB QDR to Dual FC8

Lustre threads
Evolutions for the multirail part

“get rid of the 'cost 3' cases”

- An OST must be bound to a unique NID.
- With the current Lustre code
  - an OST can be reached via all the network interfaces available on the OSS
- It is a problem on NUMIOA platforms
  - Avoiding NUMIOA factor => choosing the "good" interface
    - “good”: network adapter connected to the same IOH as the FC adapter that gives access to the LUN.
Evolutions for the multirail part

Patch Lustre code: **bug 22078**

- Giving the ability to restrict the NIDs that a target (MDT or OST) registers on the MGS.
  - force client requests to go through the desired network interface

- **--network** option of mkfs.lustre and tunefs.lustre controls target binding
  - takes as values one or more LNET networks.

```bash
mkfs.lustre --ost --fsname=numafs --mgsnode=inti5@tcp0
--network=o2ib0,tcp0 /dev/sdc
```

- **At mount.lustre time:**
  - target registers on the MGS with only the NIDs that pertain to the specified networks.
  - If no specific network is given, all NIDs are registered.
Evolutions for the NUMIOA part

“get rid of the 'cost 4' case”

- Current Lustre code is already NUMA aware.
- But this is not enough
  - Lustre code must be NUMIOA aware.

- b_hd_numa branch for SMP scaling improvements (Liang Zhen's work)
  - **Bugs 19411 and 19412**
    - options libcfs cpu_numa=1
    - options lnet ni_affinity=1
Application to the OSS server
Application to the OSS server

With current Lustre 2.0 code

options lnet networks="o2ib0(ib1),o2ib1(ib0)"
--network=o2ib1 for OSTs on IOH 0
--network=o2ib0 for OSTs on IOH 1

2\leq \text{cost} \leq 4
Application to the OSS server

With current Lustre 2.0 code

options Inet networks="o2ib0(ib1),o2ib1(ib0)"
--network=o2ib0 for OSTs on IOH 0
--network=o2ib1 for OSTs on IOH 1

Cost: 3

Threads usage - Lustre 2.0 - OST on IOH 0

Threads usage - Lustre 2.0 - OST on IOH 1
With b_hd_numa branch

options lnet networks="o2ib0(ib1),o2ib1(ib0)"

--network=o2ib1 for OSTs on IOH 0
--network=o2ib0 for OSTs on IOH 1

Cost: 2

Threads usage - b_hd_numa - OST on IOH 0

Threads usage - b_hd_numa - OST on IOH 1
Application to the OSS server

With `b_hd_numa` branch

```plaintext
options lnet networks="o2ib0(ib0),o2ib1(ib1)"
--network=o2ib0 for OSTs on IOH 0
--network=o2ib1 for OSTs on IOH 1
```

Cost: 4

Threads usage - `b_hd_numa` - OST on IOH 0

Threads usage - `b_hd_numa` - OST on IOH 1
Lustre with NUMIOA and multirail IB

Application to the OSS server

Outcome:

Lustre throughput

<table>
<thead>
<tr>
<th>Cost</th>
<th>Write (GB/s)</th>
<th>Read (GB/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 b_hd_numa</td>
<td>2.27</td>
<td>2.62</td>
</tr>
<tr>
<td>2 ≤ cost ≤ 4 b.0 alpha 7</td>
<td>2.15</td>
<td>2.52</td>
</tr>
<tr>
<td>cost = 3 b.0 alpha 7</td>
<td>2.13</td>
<td>2.35</td>
</tr>
<tr>
<td>cost = 4 b_hd_numa</td>
<td>1.27</td>
<td>1.43</td>
</tr>
</tbody>
</table>

- write
- read
Application to the OSS server

BUT:

- With Lustre 2.0
  - We use only ib0
  - Threads are spread among socket 0 and socket 1
  - Depending on target and thread, cost can be 2, 3 or 4
  - 2.2 GB/s in write, 2.5 GB/s in read

- With obdfilter-survey
  - 3.5 GB/s in write, 4 GB/s in read

Explanation for performance saturation:
- Problem in QPI management with Nehalem EP and Tylersburg on this platform
Coming next

- Bull MESCA 4S server

Lustre with NUMIOA and multirail IB
Bull MESCA 4S server

- Elementary performance tests
  - xdd for disk IOs
  - qperf for network IOs
  - Show that QPI management is better
    - we can reach QPI maximum bandwidth

Will see with Lustre!
What about a manual thread binding?

kibInd_sd and ll_ost_io threads usage - b_hd_numa