Experiences with 40G End-hosts

Wenji Wu, Liang Zhang, Phil DeMar
FNAL Network Research Group
wenji@fnal.gov, liangz@fnal.gov, demar@fnal.gov

2014 Technology Exchange
October 26 - 31, 2014 Indianapolis, IN
Outline

• Test environment and methodology
  – FNAL 40G System Test Configurations
  – Methodology

• Case 1: Packet drop

• Case 2: I/O locality
FNAL 40G Test Configurations - Hardware

**System A**
- 4 NUMA nodes
- 24 Intel E5-4607 cores
- 64GB memory
- PCIE-Gen3
- ConnectX®-3 EN 40G NIC

**System B**
- 2 NUMA nodes
- 16 Intel E5-2680 cores
- 32GB memory
- PCIE-Gen3
- ConnectX®-3 EN 40G NIC

Two systems are connected back to back.
FNAL 40G Test Configurations - Software

• System A:
  – Linux kernel 3.12.23
  – Network stack parameters are tuned
  – Iperf 2.0.5
  – Mellanox driver mlnx-en-2.1-1.0.0

• System B:
  – Linux kernel 3.12.12
  – Network stack parameters are tuned
  – Iperf 2.0.5
  – Mellanox driver mlnx-en-2.1-1.0.0
Methodology

- Run data transfers between System A and B using **iperf**
- Use **taskset** to pin **iperf** to specific core(s)
- Use Mellanox adapter IRQ affinity tuning tools
  - [http://www.mellanox.com/related-docs/prod_software/mlnx_irq_affinity.tgz](http://www.mellanox.com/related-docs/prod_software/mlnx_irq_affinity.tgz)
- Use **tcpdump** and **tcptrace** to capture/analyze packet traces
Case 1 – Packet drop

• Experiment A:
  – Turn off the *IRQ balancer* on both System A and B
  – *No IRQ affinity* tuning on System A and B (Default)
  – Run data transfers with 20 parallel streams from System A to B
  – Run *tcpdump* at System A to capture packet traces

• Experiment B:
  – Turn off the *IRQ balancer* on both System A and B
  – Use *Mellanox IRQ affinity tuning tools* to spread NIC irqs to different cores
  – Run data transfer with 20 parallel streams from System A to B
  – Run *tcpdump* at System A to capture packet traces.
Case 1 – Packet drop (cont.)

Packet trace of a single stream (Experiment A)

R in read represent packet drops

Significant packet drops!!!
Case 1 – Packet drop (cont.)

No packet drops are detected!

Packet trace of a single stream (Experiment B)
Case 1 – Packet drop

Why?

Without Affinity Tuning

With Affinity Tuning

• Networks are getting faster and CPU cores are not.
• A single core cannot keep up with the high-speed link rates
• We must spread traffic to multiple cores
Case 2 – I/O locality

• Experiment C:
  – Turn off the *IRQ balancer* on both System A and B
  – System A
    • run *Mellanox IRQ affinity tuning tools* to spread NIC irqs to cores on NUMA node 0
    • run “*numactl –N n iperf –s –w 2M*” to pin iperf to NUMA node *n*
      o *n* is varied, ranging from 0-3
  – Run data transfers with single streams from System B to A multiple times
Case 2 – I/O locality (cont.)

System A has four NUMA nodes
Each NUMA nodes has 6 cores

<table>
<thead>
<tr>
<th></th>
<th>node 0</th>
<th>node 1</th>
<th>node 2</th>
<th>node 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpus</td>
<td>0 1 2 3 4 5</td>
<td>6 7 8 9 10 11</td>
<td>12 13 14 15 16 17</td>
<td>18 19 20 21 22 23</td>
</tr>
<tr>
<td>size</td>
<td>16051 MB</td>
<td>16159 MB</td>
<td>16159 MB</td>
<td>16158 MB</td>
</tr>
<tr>
<td>free</td>
<td>14592 MB</td>
<td>15697 MB</td>
<td>15577 MB</td>
<td>15745 MB</td>
</tr>
</tbody>
</table>

System A NUMA parameters
Case 2 – I/O locality (cont.)

The results of running Mellanox IRQ Affinity tuning tools on System A

The 40GE NIC is configured with 16 queues
Each queue is tied to a specific core on NUMA node 0
Case 2 – I/O locality (cont.)

![Throughput vs Node Number Graph]

- **Node0**: High throughput
- **Node1**: Moderate throughput
- **Node2**: Lower throughput
- **Node3**: Moderate throughput

Throughput (Gbps) vs Node Number
Case 2 – I/O locality

Why?

Remote I/O access is more costly than local I/O access
I/O locality can significantly improve the overall performance