Know your Cluster Bottlenecks and Maximize Performance

Hands-on training

March 2013

Mellanox TECHNOLOGIES

Connect. Accelerate. Outperform.™

Agenda

- Overview
- Performance Factors
  - General System Configuration
    - PCI Express (PCIe) Capabilities
    - Memory Configuration
    - BIOS Settings
    - Tuning Power Management
    - Tuning for NUMA Architecture
  - Validating the Network
    - Node Validation
    - Subnet Manager Validation
    - Validating an Error Free Network with ibdiagnet
    - Expected Link Speed/Width
    - Cable Validation

Overview

- Depending on the application of the user's system, it may be necessary to modify the default configuration of the network adapters and the system/chipset configuration.
- This slide deck describes common tuning parameters, settings & procedures that can improve performance of the network adapter.
- Different Server & NIC vendors may have different recommendations for the values to be set – But the general tuning approach should be similar.
- For the hands-on demo we will utilize Mellanox ConnectX® adapters – thus we will implement the recommended settings issued by Mellanox.

Performance Factors

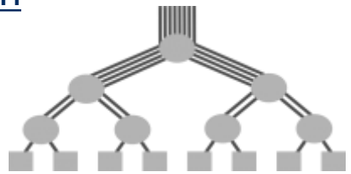
Dividing the performance affecting factors into 2 areas:

- Single System Configuration
- Network Configuration

- System General Configuration

  - PCI Express (PCIe) Capabilities
  - Memory Configuration
  - BIOS Settings
  - Tuning power management
  - Tuning for NUMA Architecture

- Network Configuration



  - Error free
  - Expected link speed/width
  - Correct topology
  - QoS

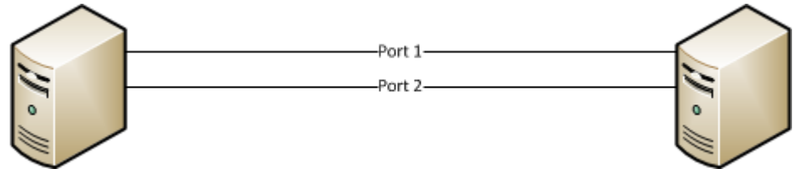
Single System Configuration

General System Configurations – PCIe BW Limitations

PCIe Generation	2.0	3.0
Speed	5GT/s	8GT/s
Width	x8	x8 or x16
PCIe BW Constraint	x8 * 5Gbps * (8:10) = 32Gbps Taking into account: PCI FC and PCI headers we get max of: ~27-29Gbps	x8 * 8Gbps * (128:130) = 63Gbps Taking into account: PCI FC and PCI headers we get max of: ~52-54Gbps
QDR rate x4*10 Gbps * (8:10) **Max BW = 32Gbps**	PCI limits BW @ 27Gbps	IB limits Max BW = 32Gbps
FDR14 rate x4*14.4375 Gbps * (64:66) **Max BW = 56Gbps**	PCI limits BW @ 27Gbps	PCI limits Max BW = 52-54Gbps

Network Configuration – PCIe Demo

- **Running a simple RDMA test on port#1 between 2 gen3 servers:**
  - **Server: ib_write_bw -d mlx5_0 -i 1 --run_infinitely**
  - **Client: ib_write_bw -d mlx5_0 -i 1 --run_infinitely mtlae01**



- **Querying the device capabilities using:**

  lspci –s [slot_num] –xxxvvv

  - Find the **PCIe capability register** (using # lspci –v)
  - Locate the **Link Capabilities Register** (offset 0Ch)
    - Locate the **Max Link Speed** (bits 3:0) & **Width** (bits 9:4)
  - Locate the **Link Status Register** (offset 12h)
    - Locate the **Current Link Speed** (bits 3:0) & **Width** (bits 9:4)
  - Verify the device had gone up with the max supported speed/width.
  - **In real life scenario** – when encountering an issue with the PCIe, the user will have to check the card and the board itself.

```
83:00.0 Network controller: Mellanox Technologies MT27500 Family [ConnectX-3]
00: b3 15 03 10 06 04 10 00 00 00 80 02 10 00 00 00
10: 04 00 e0 fb 00 00 00 00 0c 00 00 fb 00 00 00 00
20: 00 00 00 00 00 00 00 00 00 00 00 00 b3 15 50 00
30: 00 00 d0 fb 40 00 00 00 00 00 00 00 0b 01 00 00
40: 01 48 03 00 00 00 00 00 03 9c fc 8f 00 00 00 78
50: 00 00 00 00 00 00 00 00 14 00 0f 00 f5 01 01 00
60: 10 00 02 00 01 8e d0 11 20 20 00 00 83 f4 43 08
70: 40 00 83 10 00 00 00 00 00 00 00 00 00 00 00 00
80: 00 00 00 00 1f 00 00 00 00 00 00 00 0e 00 00 00
90: 03 00 00 00 00 00 00 00 00 00 00 00 11 60 7f 80
a0: 00 c0 07 00 00 d0 07 00 05 00 8a 00 00 00 00 00
b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

General System Configurations – Tuning for NUMA Architecture

- **From Wikipedia – Non-Uniform Memory Access (NUMA)**
  is a computer memory design used in multiprocessing, where the memory access time depends on the memory location relative to a processor. Under NUMA, **a processor can access its own local memory faster than non-local memory** (memory local to another processor or memory shared between processors).

- **Tuning for Intel® Microarchitecture Code name Sandy Bridge**
  The Intel Sandy Bridge processor has an integrated PCI express controller. Thus every PCIe adapter is connected directly to a NUMA node. On a system with more than one NUMA node, performance will be better when using the local NUMA node to which the PCIe adapter is connected.
  In order to identify which NUMA node is the adapter's node the system BIOS should support the proper ACPI feature.

  - **To see whether your system supports PCIe adapter's NUMA node detection run:**
    # cat /sys/devices/[PCI root]/[PCIe function]/numa_node
    Or
    # cat /sys/class/net/[interface]/device/numa_node

    Example for a supported system:

    # cat /sys/devices/pci0000\:00/0000\:00\:05.0/numa_node
    0

    Example for a supported system:

    # cat /sys/devices/pci0000\:00/0000\:00\:05.0/numa_node
    -1

General System Configurations – Tuning for NUMA Architecture

Tuning for AMD® Architecture

On AMD architecture there is a difference between a 2 socket system and a 4 socket system.
In order to identify which NUMA node is the adapter's node the system BIOS should support the proper ACPI feature.

- With a 2 socket system the PCIe adapter will be connected to socket 0 (nodes 0,1).
- With a 4 socket system the PCIe adapter will be connected either to socket 0 (nodes 0,1) or to socket 3 (nodes 6,7).

General System Configurations – Tuning for NUMA Architecture

- **Recognizing NUMA Node Cores**
  To recognize NUMA node cores, run the following command:
  # cat /sys/devices/system/node/node[X]/cpulist || cpumap
  Example:
  # cat /sys/devices/system/node/node1/cpulist
  1,3,5,7,9,11,13,15
  # cat /sys/devices/system/node/node1/cpumap
  0000aaaa

  OR
  # lscpu

- **Running an Application on a Certain NUMA Node**
  In order to run an application on a certain NUMA node, the process affinity should be set in either in the command line or an external tool. For example, if the adapter's NUMA node is 1 and NUMA 1 cores are 8-15 then a multi thread application should run with process affinity that uses 8-15 cores only.

  To run an application, run the following commands:

  # taskset -c 8-15 ib_write_bw –a

  or:

  # taskset 0xff00 ib_write_bw –a

General System Configurations – Memory Configuration

- Common Knowledge –
  For high performance it is recommended to use the highest memory speed with fewest DIMMs.

  Please look for your vendor's memory configuration instructions or memory configuration tool available Online for tuning the system’s memory to the max performance

General System Configurations - Recommended BIOS Settings

General

- Set BIOS power management to Maximum Performance
  NOTE: These performance optimizations may result in higher power consumption.

Intel Processors

- The following table displays the recommended BIOS settings in machines with Intel Nehalem-based processors.

BIOS Option		Values
General	**Operating mode**	**Performance**
Processor	**C-States**	**Disabled**
	Turbo Mode	Disabled
	CPU Frequency	**Max performance**
Memory	Memory speed	Max performance
	Memory channel mode	Independent
	Socket Interleaving	**NUMA**
	Memory Node Interleaving	**OFF**
	Patrol Scrubbing	Disabled
	Demand Scrubbing	Enabled
	Thermal Mode	Performance

General System Configurations - Power Management Tuning

- C-States
  - In order to save energy when the CPU is idle, the CPU can be commanded to enter a low-power mode. Each CPU has several power modes which are collectively called “C-States”. These states are numbered starting at C0 which is the normal CPU operating mode, i.e. the CPU is 100% turned on. The higher the C number is, the deeper is the CPU sleep mode, i.e. more circuits and signals are turned off - **prolonging the time it’ll take the CPU to go back to C0 mode.**

- Some operating systems can override BIOS power management configuration and enable c-states by default, which results in a higher latency.
  - To resolve the high latency issue, please follow the instructions below:
    1. Edit the **/boot/grub/grub.conf** file or any other bootloader configuration file.
    2. Add the following kernel parameters to the bootloader command:

       intel_idle.max_cstate=0 processor.max_cstate=1
    3. Reboot the system

    Example:

    title RH6.2x64
    root (hd0,0)
    kernel /vmlinuz-RH6.2x64-2.6.32-220.el6.x86_64
    root=UUID=817c207b-c0e8-4ed9-9c33-c589c0bb566f console=tty0
    console=ttyS0,115200n8 rhgb ***intel_idle.max_cstate=0 processor.max_cstate=1***
    initrd /initramfs-RH6.2x64-2.6.32-220.el6.x86_64.img

General System Configurations – Checking Core Frequency

- Check that the output CPU frequency for each core is equal to the maximum supported and that all core frequencies are consistent.

  - **Check the maximum supported CPU frequency using:**
    # cat /sys/devices/system/cpu/cpu/cpufreq/cpuinfo_max_freq*

  - **Check that core frequencies are consistent using:**
    # cat /proc/cpuinfo | grep “cpu MHz”

  - **Check that output frequencies are the same as the maximum supported**
    If the CPU frequency is not at the maximum, check the BIOS settings according to table in Recommended BIOS Settings (on slide 7) to verify that power state is disabled.

Network Configuration

Network Configuration – Node Validation

- **Validating nodes condition:**

  - **Responding (and checking kernel version)**
    - **e.g.** ‘pdsh –w node[01-10] /bin/uname –r | dshbak -c’
  - **Have the same OFED version**
    - **e.g.** ‘pdsh –w node[01-10] /usr/bin/ofed_info | head -n 1 | dshbak -c’
  - **HCAs are attached to driver and have expected FW version**
    - **e.g.** ‘pdsh –w node[01-10] ibstat | grep Firmware | dshbak -c’
  - **Validate port state (after SM)**
    - **e.g.** pdsh -w node0[01-10] ‘ibstat | grep State’ | dshbak –c
      - State: Active/Down

Network Configuration – Validating an Error Free Network with ibdiagnet

- **Fabric cleanup is used to filter out bad HW including bad cables, bad ports or bad connectivity.**

- **Fabric cleanup algorithm**
  1. Zero all counters (ibdiagnet –pc)
  2. Wait X time
  3. Check for errors exceeding allowed threshold during this X time (ibdiagnet –lw 4x –ls 10 –P all=1)
  4. Fix problematic links (re-sit or swap cables, replace switch ports or HCAs etc.)
  5. Go to 1

Network Configuration – Validating an Error Free Network with ibdiagnet

- -p|--port <port-num>
- -pc - Reset all the fabric links pmCounters
- -P|--counter <<PM>=<value>>
- -r|--routing - provides a report of the fabric qualities
- -u|--fat_tree - Indicates that UpDown credit loop checking should be done against automatically determined roots.
- -o|--output_path <out-dir>
- -h|--help
- -V|--version

Network Configuration – Validating an Error Free Network with ibdiagnet

- Symbol and Receive Errors Thresholds
  - Thresholds are link speed and target BER dependent.
  - For a QDR link with expected BER of 10e-12 (IB spec) the threshold will be: 10e12/40x10e9 = **25 sec/err**
  - For 10-15 BER: 10e15/40x10e9 = **25000 sec/err** = ~**7 Hour/err**
- Link down and link retrain
  - Both this counters should be **zero**
- Rcv Remote Phys Errors
  - Indicates an issue (Symbol or Receive errors) on some remote port.
- VL15Dropped – In most cases should be ignored as it counts MADs that are received before the link is up.

Network Configuration – Subnet Manager Optimization

- **One of the common issues is not having the SM running. There are various scenarios where the SM may have dropped and no other SM has been configured on the subnet.**
  - **Check the presence of the SM using sminfo cmd.**
  - **Check there are no errors with the SM by looking for error messages in the SM log (**(/var/log/opensm.log)
  - **Select the right routing algorithm, use opensm -R <couting engine> or routing_engine parameter in opensm config file**
  - **For fat tree or up down routing GUID file may be needed. GUID file is a txt file with a list of GUID of the highest hierarchical switch ASIC's**
  - **Validate that routing succeeded in opensm log (/var/log/opensm.log)**
  - **Validate that there are no errors in the log file**
  - **Use ibdmchk to validate routing**
  - **ibdmchk -s ./opensm-subnet.lst -f ./opensm.fdbs -m ./opensm.mcfdbs**

- **QOS setting - SL2VL mapping.**

Network Configuration – Link Speed/Width & Cable Validation

- **Validate the link speed & width using ibportstate or any OpenIB diagnostic tools.**

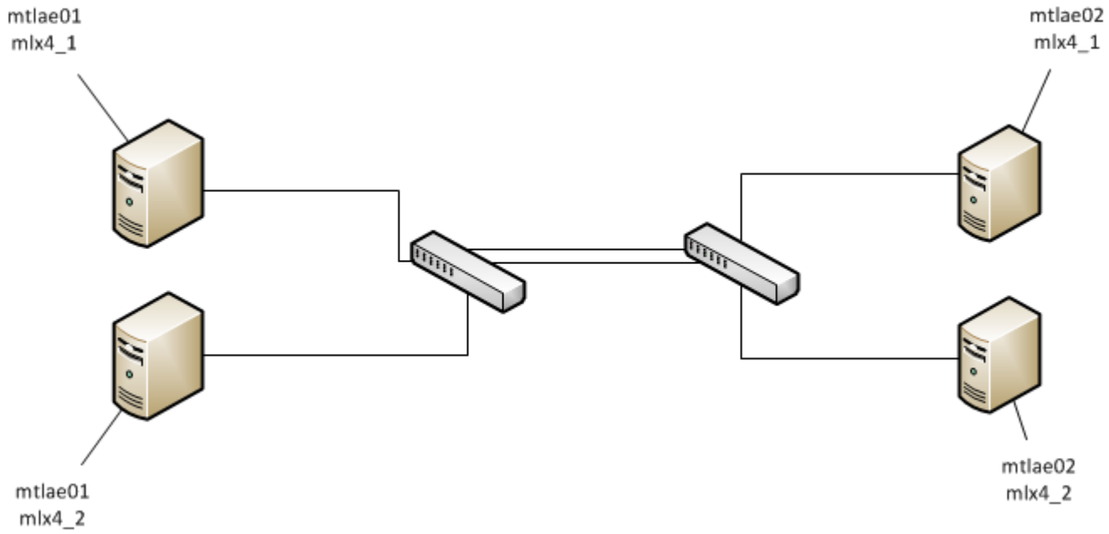
  Validate the fabric's cables using the Cable Diagnostic plugin in ibdiagnet.

  - *--get_cable_info : Indicates to query all QSFP cables for cable information. Cable information will be stored in "ibdiagnet2.cables".*
  - *--cable_info_disconnected : Get cable info on disconnected ports.*

- **Validate the correct cables are being utilized on the respected links.**

Network Configuration – Congestion Example

- **Disabling one of the links between the switches while running RDMA tests between 2 interfaces on both servers.**
  - **SM configures the routing to be divided between the ports.**
- **Reducing the link width of one of the switches connections**
  - **The SM won't re-route and we'll see the BW drops only on the relevant interfaces which are routed through the link with the reduced width while the other link works in good performance.**



Thank You