Intel® True Scale Fabric Quick Start Guide

Contents

Revision	Revision History	Date
1.0	Initial draft. Author: Andrew Russell, Intel Corp.	Feb 2011
2.0	Intel® template. Added Download and Contexts sections	Sep 2013
3.0	Publish to Intel.com	July 2014

1 Document Scope

This guide shows a tried and tested method of installing and testing an Intel® True Scale Fabric. It has been used to install HPC clusters of all sizes and should be appropriate for most deployments. It omits discussion and alternatives in order to keep it brief and easy to follow.

Intel® True Scale Fabric can be installed on Red Hat Enterprise Linux (RHEL) 5.x, 6.x (including CentOS and Scientific Linux), and SLES10 and SLES11. For simplicity, this guide uses RHEL6.

There are many ways to perform the installation of a True Scale Fabric, consult the product documentation to see the full capabilities of the commands and methods available.

The guide does not describe the following topics:

- How to configure the switches and verify the cabling in larger fabrics with multiple switches. These procedures are additional to this guide.
- How to install on clusters using management software that requires special packaging of software components, e.g. IBM Platform Cluster Manager, Rocks Cluster Distribution, etc. The section “Install the Host Stack” in this guide is not applicable for such systems.
- Multivendor InfiniBand* fabrics: mixing Intel® True Scale Architecture components with InfiniBand* components from other vendors.

Please contact an Intel® True Scale Fabric subject matter expert for help with fabrics using any of these items.

1.1 Intended Audience

This document is intended for use by Linux administrators installing Intel® True Scale Architecture fabrics for evaluation or production. The reader must have a basic understanding of Linux administration, installation, and cluster management systems, and be able to build a Linux cluster.

2 Outline of installation steps

- Build a typical Linux HPC cluster, with head node and compute nodes, ideally with a cluster management system to enable the automated re-build of any node. Include a non-root user, and set up passwordless ssh for root and a non-root user from the head node to all the compute nodes, and a shared /home directory (possibly using NFS).
- Cable the nodes to the IB switch.
- Download the IntelIB host stack from www.intel.com
- Install the host stack on all nodes. Integrate this step into the node re-build method.
- Run simple MPI programs to test fabric functionality and performance.
- Check fabric quality.

3 Download the IntelIB host stack

The Intel® True Scale Fabric host stack is available in two versions:

- Basic
- Intel® Fabric Suite (IFS)

The Basic bundle is freely downloadable from the Intel® Download Center.

The IFS bundle is an extra-cost item and is a superset of the Basic bundle, additionally containing the following packages:

- Fastfabric: A collection of utilities for installation, testing and monitoring the fabric.
- Intel® Fabric Manager (ifs_fm): Intel®'s host based Subnet Manager.

Both of these packages offer valuable benefits, so if you have purchased IFS, then you should download and install the IFS bundle.

Downloading IntelIB-Basic

Visit http://downloadcenter.intel.com. Use Find by Category, select Network Connectivity, Intel® True Scale Fabric Host Channel Adapters, Intel® True Scale Fabric Host Channel Adapter 7340. Choose the file: IntelIB-Basic.RHEL6-x86_64.x.x.x.x.tgz.

Downloading IntelIB-IFS

If you have purchased IFS then you should receive an Intel Software Serial Number, which has the form XXXX-XXXXXXXX.

If you purchased IFS from QLogic, the serial number will have the form LKNXXXXXXXXXX. Contact ibsupport@intel.com and ask for the LKN number to be converted to an Intel serial number. If you purchased your switches from IBM*, IFS is bundled, and you should receive an Intel serial number.

When you have your Intel® Software Serial Number, visit the Intel Software Products Registration Center (http://registrationcenter.intel.com) and register your serial number. You will then be able to download the IFS bundle. Choose the file IntelIB-IFS.RHEL6-x86_64.x.x.x.x.tgz.

4 Install the Host Stack

Install the same host stack on all the nodes. There is little point in choosing different installation profiles for different types of node. I suggest you install one node first, to ensure that the installation is successful (i.e., that you have all the dependencies), then install the rest of the nodes.

```
cd
tar xf /tmp/IntelIB-IFS.RHEL6x86_64.*.tgz
cd IntelIB-IFS.RHEL6-x86_64.*
./INSTALL -i ib_stack -i truescale -i ib_stack_dev -i fastfabric \
       -i ofed_ipoib -i ofed_debug \
       -i mvapich_gcc_qlc  -i mvapich_intel_qlc  \
       -i mvapich2_gcc_qlc -i mvapich2_intel_qlc \
       -i openmpi_gcc_qlc  -i openmpi_intel_qlc  \
       -i ifs_fm -i opensm -D ifs_fm -D opensm

On the node(s) which will be SM servers  {
       chkconfig ifs_fm on
}

reboot
```

Note: If using the IntelIB-Basic bundle, then omit `-i fastfabric` and `-i ifs_fm`, and replace `chkconfig ifs_fm on` with `chkconfig opensmd on`

If you want to use IPoIB, then you will need to create /etc/sysconfig/network-scripts/ifcfg-ib0. You probably have a method to do this, as it is a standard network interface config file.

Scaling up the installation procedure

You can perform the installation on multiple nodes using tools like pdsh and xdsh, or integrate it into your node-build method, for example as a kickstart post-install script.

When the ./INSTALL script is run on a system with a non-default kernel version, it will attempt to rebuild its modules using the kernel-devel support on the system. The rebuild takes a noticeable amount of time. To avoid each node doing its own rebuild, you should perform the ./INSTALL on the headnode, and share the directory to the other nodes. When you run ./INSTALL on the other nodes, it will find the modules pre-built for this kernel version.

5 Test the fabric

Check that your local HCA is Up/Active:

```
ibstat
```

Example output:

```
CA 'qib0'
        CA type: InfiniPath_QLE7340
        Number of ports: 1
        Firmware version:
        Hardware version: 1
        Node GUID: 0x00117500005a6a90
        System image GUID: 0x00117500005a6a90
        Port 1:
                State: Active
                Physical state: LinkUp
                Rate: 40
                Base lid: 27
                LMC: 0
                SM lid: 1
                Capability mask: 0x0761086a
                Port GUID: 0x00117500005a6a90
                Link layer: InfiniBand
```

If you have just two nodes connected back to back with a cable, you may find the Physical state is 'Sleep'. To fix this, run "`ibportstate -D 0 1 enable`" on one of the nodes.

If the State is 'Init', then your SM may not be running. Start an SM on the switch, or on one of the nodes.

Check that all the fabric components are visible: hosts, switches and cables.

```
fabric_info
```

Example output:

```
Fabric 0:0 Information:
SM: head-node HCA-1 Guid: 0x001175000078f214 State: Master
Number of CAs: 32
Number of CA Ports: 32
Number of Switch Chips: 1
Number of Links: 32
Number of 1x Ports: 0
```

Check that these numbers are correct for your fabric. For simple fabrics with one 36-port switch and single-port HCAs, check that the number of CA Ports equals the number of hosts. For more complex fabrics with director switches or multiple switches, see later sections.

5.1 Performance tests

The full performance of Intel® True Scale HCAs is only attainable from MPI, so performance tests should be performed using small MPI test programs. This is best done as a non-root user with a shared home directory.

Do not use commands such as ib_read, ib_write, ib_rdma, qperf to test performance.

First, copy the example programs to your home directory, and build them.

```
cd /home/mpiuser
cp -a /opt/iba/src/mpi_apps .
cd mpi_apps
export MPICH_PREFIX=/usr/mpi/gcc/openmpi-*-qlc
make OSU3 DEVIATION
```

Next, create an mpi_hosts file, and run some tests. Use the hostname associated with the primary Ethernet interface of the node, one per line.

```
vi mpi_hosts
    add two hostnames
cat mpi_hosts
node01
node02
./run_lat3
./run_bw3
```

Latency for 0-byte messages should be less than 2us, and bandwidth for larger messages should be more than 3000MB/s.

You can test all of the nodes in the cluster using the 'deviation' program. The deviation program will identify any nodes with results outside of the thresholds as FAILED.

```
vi mpi_hosts
              add all nodes
./run_deviation 16
   Example output for a cluster with E5-2697 v2 @ 2.70GHz CPUs:
Sequential MPI Performance Test Results
  Latency Summary:
    Min: 1.15 usec, Max: 1.16 usec, Avg: 1.15 usec
    Range: +1.0% of Min, Worst: +0.4% of Avg
    Cfg: Tolerance: +50% of Avg, Delta: 0.80 usec, Threshold: 1.95 usec
         Message Size: 0, Loops: 4000

  Bandwidth Summary:
    Min: 3285.5 MB/s, Max: 3307.6 MB/s, Avg: 3305.6 MB/s
    Range: -0.7% of Max, Worst: -0.6% of Avg
    Cfg: Tolerance: -20% of Avg, Delta: 150.0 MB/s, Threshold: 2644.5 MB/s
         Message Size: 2097152, Loops: 30 BiDir: no

Latency: PASSED
Bandwidth: PASSED
```

About PCIe slots

Starting with the Sandybridge CPU architecture, the PCIe controller has been included in the processor package. A dual-socket server will have PCIe slots driven by socket 0, and PCIe slots driven by socket 1. If a processor needs to access the HCA which is driven by the other socket, the latency will be increased by approximately 0.4us as the message is transferred over the Intel® QuickPath Interconnect (Intel® QPI). You can observe this latency difference using taskset in the mpirun command to select different cores.

```
source /usr/mpi/gcc/openmpi-1.4.3-qlc/bin/mpivars.sh
cd OMB-3.1.1
mpirun -host node01,node02 -np 2 taskset –c  0 ./osu_latency
mpirun -host node01,node02 -np 2 taskset –c 12 ./osu_latency
```

About QDR80

QDR80 (sometimes known as Dual-Rail) systems have two HCAs in each server connected to a single fabric. You should ensure that one HCA is in a slot driven by socket 0, and the other HCA is in a slot driven by socket 1. The ib_qib driver will try to set processor affinity such that messages do not need to cross the QPI.

(Note – Testing Dual-Plane systems is not described here)

When testing a QDR80 system, it is important to test both HCAs. Assuming the affinity control is working correctly, you should be able to select the HCA to use by selecting the CPU core. However, you can also select the use of a particular HCA using the IPATH_UNIT environment variable. So, a belt and braces test of a QDR80 system would be:

```
mpirun -host node01,node02 –x PATH_UNIT=0 -np 2 taskset –c 0 ./osu_latency
mpirun -host node01,node02 –x PATH_UNIT=1 -np 2 taskset –c 12 ./osu_latency
```

About CPU clock speed

These tests are micro-benchmarks, so take a very small time to execute. If you have some cpu throttling or other power saving features enabled, the CPUs may not come up to full speed within the duration of the test. Usually, disabling C-States in the BIOS and stopping the Linux cpuspeed service will achieve good results. For more information, see the Intel® True Scale Tuning Guide.

If you do not have a shared home directory

OpenMPI includes some pre-built tests that will be present on every node (However, it does not include 'deviation'). Run these tests as follows:

```
source /usr/mpi/gcc/openmpi-1.4.3-qlc/bin/mpivars.sh
cd /usr/mpi/gcc/openmpi-1.4.3-qlc/tests/osu_benchmarks-3.1.1
mpirun -host node01,node02 -np 2 ./osu_latency
```

You can also build and copy a test program:

```
source /usr/mpi/gcc/openmpi-1.4.3-qlc/bin/mpivars.sh
cd /tmp
cp /opt/iba/src/mpi_apps/deviation/deviation.c .
mpicc -o deviation deviation.c
seq -f 'node%02.0f' 1 16 > mpi_hosts
for n in `cat mpi_hosts`; do scp deviation $n:/tmp; done
mpirun -hostfile mpi_hosts -np 16 ./deviation
```

5.2 Fabric Quality tests

5.2.1 Error rates

InfiniBand* ports have counters that record any errors occurring at the port, as well as recording traffic throughput. For a full description of each counter type, refer to the documentation. For fabric quality, we are primarily interested in Symbol Errors and Link Error Recoveries. Link Error Recoveries should be zero, and Symbol Errors should be roughly less than one per hour per port.

We will use the `iba_report` command, provided in the fastfabric package of the IntelIB IFS bundle. If you do not have IFS, then commands such as `ibcheckerrors` can be used. All this can be performed on a single node, typically the head node.

First, set the error thresholds used by iba_report such that it will report single errors.

Edit /etc/sysconfig/iba/iba_mon.conf

Set “Threshold” to “Equal”, and set all error threshold values to 1.

Next, clear the counters on every port in the fabric

iba_report -o none --clearall

Now, wait 30 seconds, and query the error counters

iba_report -o errors -o slowlinks

This reports links that have come up at the wrong speed, and links that have data errors. Very bad links will show up immediately. If you see no errors, wait a longer time (say five minutes) and repeat the test. Also, run some workload on the fabric, and check again.

```
iba_report -o errors –o slowlinks
```

The most common causes of errors are poor or damaged cables, though can also be bad ports on the equipment. If errors are present, try reseating the cable, or replacing it. Note that some errors will be generated when a node reboots. A clue that a reboot has occurred is that there will be some Link Downed errors reported on the associated links. If this has happened, any errors on that link should be ignored, and it is best to clear all the errors to get a clean error report.

5.2.2 Fabric Stability

Check that the SM is not reporting any error messages. This example is from the ifs_fm (Intel® Fabric Suite Fabric Manager) provided in the ifs_fm package of the IntelIB-IFS bundle. By default, this SM writes messages to the normal log service.

```
tail -f /var/log/messages
```

or

```
grep fm0 /var/log/messages | tail
```

You should see no significant messages from the SM. Ideally, you should see something like this, once the system has settled down:

```
Dec 15 00:22:01 head-node fm0_sm[31698]: PROGR[topology]: SM: TT: DISCOVERY CYCLE START.
Dec 15 00:22:02 head-node fm0_sm[31698]: PROGR[topology]: SM: TT: DISCOVERY CYCLE END. 21 SWs,
202 HCAs, 202 end ports, 857 total ports, 2 SM(s), 3324 packets, 0 retries, 0.363 sec sweep
Dec 15 00:27:02 head-node fm0_sm[31698]: PROGR[topology]: SM: TT: DISCOVERY CYCLE START.
Dec 15 00:27:02 head-node fm0_sm[31698]: PROGR[topology]: SM: TT: DISCOVERY CYCLE END. 21 SWs,
202 HCAs, 202 end ports, 857 total ports, 2 SM(s), 3324 packets, 0 retries, 0.342 sec sweep
```

See the following characteristics in this example:

- Each sweep is five minutes from the last.
- There are no errors during the sweep.
- The sweep is quick (less than 1 second for this size of fabric) with no retries.

Any fabric changes, such as the reboot of a node, or the removal of a cable, will be noted and will pro-actively cause a new sweep.

6 Running MPI Applications

When running MPI programs on a cluster with Intel® True Scale HCAs, it is important to direct MPI to use the PSM interface, otherwise you will get sub-optimal interconnect performance. Typically, IBV (InfiniBand Verbs), OpenIB, DAPL, etc should not be selected, even if the documentation for your benchmark or application suggests that it should.

How do you know if you have Intel®True Scale HCAs?

```
ssh computenode ibstat | grep type
       CA type: InfiniPath_QLE7340     << Intel® example
ssh computenode ibstat | grep type
       CA type: MT25208                << Mellanox example
```

Your result may differ slightly according to the particular HCA that you have, but it should resemble this.

How do you know if you are using PSM?

While the MPI job is running, ssh to a compute node and see how many PSM contexts are open. There should be one context per MPI process running on that node.

```
ssh computenode ipathstats | grep CtxtsOpen
```

Sanity check:

If you compile and run osu_bandwidth and osu_latency, they should show bandwidth greater than 3000MB/s (or 2500MB/s on AMD cpus), and less than 2us latency.

A latency of 5us suggests that you are using the Verbs interface instead of PSM.

Do not use qperf; this will measure the verbs performance and not indicate the MPI/PSM performance.

How do you direct MPI to use the PSM interface?

MVAPICH
MVAPICH2
OpenMPI
	Using one of MPIs supplied with the IntelIB host stack.
	`source /usr/mpi/gcc/mvapich-*-qlc/bin/mpivars.sh`
	`source /usr/mpi/gcc/mvapich2-*-qlc/bin/mpivars.sh`
	`source /usr/mpi/gcc/openmpi-*-qlc/bin/mpivars.sh`
	Substitute gcc with intel or pgi if using different compilers.

Intel® MPI	`source /opt/intel/impi/*/bin64/mpivars.sh`
	`mpirun -PSM`
	`or`
	`export I_MPI_FABRICS=tmi; mpirun`

IBM* / Platform* MPI	`mpirun -PSM` Ensure that hpmpi.conf refers to ib_qib, and not ib_ipath.

OpenMPI that you have built yourself	`mpirun --mca mtl psm`
	`[Not: --mca btl openib]`
	`[Not: --mca mtl ^psm]`

7 Understanding HCA Contexts

Understanding Contexts is particularly important if you have more processor cores than HCA Contexts. Intel® True Scale 7400 series HCAs have 18 hardware contexts. That is 1 kernel context for each port, and 16 user contexts.

Each MPI process requires a context. If there are more MPI processes than hardware contexts, the hardware contexts will be shared. They can be shared 2, 3 or 4 ways, supporting a maximum of 4x16=64 processes. Sharing contexts incurs some performance penalty, though it is not noticeable in most applications.

7.1 Out of Contexts errors

From time to time, you may find that MPI jobs fail to start, and report that they cannot assign a context. The errors have the form:

```
jf-1-14.19191 ipath_userinit: assign_context command failed: Invalid argument
jf-1-14.19191 Driver initialization failure on /dev/ipath (err=23)
```

The most common causes are listed below.

Badly terminated jobs

A previous job may have terminated badly, leaving stranded MPI processes on the cluster. These processes will be holding on to the HCA contexts. You need to kill the processes to free up the contexts. See how many contexts are in use using one of the methods below.

Running more process per node than cores

This requires sharing of contexts, and in most cases should work fine. However, the allocation of contexts can be different from what is required; as in the 'multiple jobs' example below.

Running multiple jobs per node

Example: Each of my cluster nodes has 64 cores. I can run jobs with ppn=64 (64 processes per node). However, if I start a job with ppn=32, I cannot start another job that uses the remaining cores in each node.

Explanation: When the ppn=64 job starts, PSM shares the 16 hardware contexts 4 ways to provide 64 contexts. When I start the ppn=32 job, PSM shares the 16 hardware contexts 2 ways to provide 64 contexts, thus consuming all the contexts. When I try to start another ppn=32 job on the same nodes, it will fail because there are no free contexts.

Remedy: Use `export PSM_RANKS_PER_CONTEXT=4` to force PSM to always share the contexts 4 ways.

7.2 How many contexts are in use?

Run `ipathstats`, and look for CtxtsOpen. Eg:

```
pdsh –a ipathstats | grep Open
```

If the `ipathstats` command is not available:

```
cat /sys/class/infiniband/qib0/nfreectxts
cat /sys/class/infiniband/qib0/nctxts
```

Or, an optimal command might be:

```
pdsh -a '(cd /sys/class/infiniband; grep -H . qib*/*ctxts)'
```

8 Legal Notices

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request. Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or by visiting Intel's Web site at www.intel.com.

*Other names and brands may be claimed as the property of others.