Running IB on the Cray XT3

Presented by
Makia Minich
Oak Ridge National Laboratory
US Department of Energy
Questions to answer...

• Why would we want to even do that?
• What exactly are we trying to do here?
• What does it take to make it work?
• What kind of performance can we expect to see?
• What does the future hold?
Why would we want to do this?

• Lower cost, high bandwidth, low latency solution
• Growing open-source community involvement
• Pretty darn cool thing to do
• Did I mention lower cost?
Center-Wide File System (Spider)

Phoenix Cray X1E

NFS Servers

HPSS

ESnet, USN, TeraGrid, Internet2, NLR

Data Analysis & Visualization

Jaguar Cray XT3

HPSS

Late 2006
• 200 TB
• 10 GB/s (aggregate)

2008
• 10 PB
• 240s GB/s (aggregate)
What are we trying to do here?

- Fill the PCI-* slot with an infiniband card
- Bring up the OFED software stack
- Use our new found, high bandwidth, low latency connection.
How do we make it work?

- **Prior to Unicos 1.5:**
  - Need to modify the kernel
  - Export `bad_dma_address` and `dev_change_flags`

```c
# Patch to arch/x86_64/kernel/pci-nommu.c
@@ -10,6 +10,9 @@
 * Dummy IO MMU functions
 */

+dma_addr_t bad_dma_address;
+EXPORT_SYMBOL(bad_dma_address);
+
void *pci_alloc_consistent(struct pci_dev *hwdev, size_t size,
        dma_addr_t *dma_handle)
{

# Patch to net/core/dev.c
@@ -3482,10 +3482,7 @@
+#if defined(CONFIG_BRIDGE) || defined(CONFIG_BRIDGE_MODULE)
 EXPORT_SYMBOL(br_handle_frame_hook);
 #endif
-/* for 801q VLAN support */
-#if defined(CONFIG_VLAN_8021Q) ||
 defined(CONFIG_VLAN_8021Q_MODULE)
 EXPORT_SYMBOL(dev_change_flags);
-#endif
 #ifdef CONFIG_KMOD
 EXPORT_SYMBOL(dev_load);
 #endif
```
How ... continued

- Need to match gcc versions with the kernel in question
- OFED utilities don’t like gcc-3.2, so build just the modules
- Use separate conf files (they’ll save you time)
  - Invert the example and you will build everything else

```bash
STACK_PREFIX=/usr/ofed
BUILD_ROOT=/var/tmp/OFE
D
kernel_ib=y
ib_verbs=y
ib_mthca=y
ib_ipoib=y
ib_ipath=n
ib_sdp=y
ib_rds=n
ib_srp=n
kernel_ib_devel=y
libibverbs=n
libibverbs_devel=n
libibverbs_utils=n
libibcm=n
libibcm_devel=n
libmthca=n
libmthca_devel=n
libibcommon=n
libibcommon_devel=n
libibmad=n
libibmad_devel=n
libibumad=n
libibumad_devel=n
opensm=n
opensm_devel=n
openib_diags=n
librdmacm=n
librdmacm_devel=n
librdmacm_utils=n
dapl=n
dapl_devel=n
mpi_osu=n
openmpi=n
mpitests=n
ibutils=n
```
How ... continued ... again

• Build the stack
  • Needs to point to kernel headers
  • Unicos <1.5 use your own kernel source code
  • Otherwise, point to /opt/xt-os/default/linux/ss-lustre26
  • Check your kernel version
  • Seem to have good luck building on the SMW

```
# K_SRC=/opt/xt-os/default/linux/ss-lustre26 K_VER=2.6.5-7.283-ss ./build.sh -c ofed.conf.modules
```
Yup ... still continuing

• Boot your kernel (if you had to build one)
  • xtcli boot_cfg update -i /tmp/boot/kernel.cpio-1.5.31
  • xtbootsys --reboot c0-0c1s2n3

• Load your modules on the infiniband node
  • Fairly large solution set
  • Personally, use /tmp and “mount –bind” tricks

• Configure your network
So, what can it do?

Bandwidth Comparison
IB on XT3 Low-Level Bandwidth Results

Throughput (MB/s)

Message Size (bytes)

1872MB/s
1395MB/s
937MB/s
904MB/s
Latency Comparison
IB on XT3 Low-Level Latency Results

- RC-pinto
- UC-pinto
- UD-pinto
- RC-rizzo
- UC-rizzo
- UD-rizzo

Latency (us)

Message Size (bytes)

5us
6us

1 10 100 1000 10000 100000 1e+06
What does the future hold?

- Lustre
- NFS over RDMA
- HPSS
- Long range wide-area storage
- Ruling the world
Links/Email

- OpenFabrics – http://www.openfabrics.org
- http://jobs.ornl.gov
- Email: minich@ornl.gov
Any Questions?