

The Cray XT6(m) Programming Environment

Cray Workshop

UDE June 9-10, 2010

Stefan Andersson

stefan@cray.com

Agenda

- Programming Environment Overview
 - Modules
- Compilers
 - PGI
 - GNU
 - Cray
- CNL programming considerations
- MPI Communications
- Cray XT I/O
- Running an application

Programming Environment Overview



Cray XT6(m) programming environment overview

- Several compilers available
 - PGI, Cray, GNU
 - Intel, Pathscale supported, but not installed at UDE
- Optimized libraries
 - 64 bit AMD Core Math library (ACML)
 - Scilib: Level 1,2,3 of BLAS, LAPACK, FFT, ScaLAPACK, BLACS
 - FFTW, netCDF, HDF5, PETSc
 - MPI-2 message passing library for communication between nodes
 - SHMEM one-sided communication library
 - PGAS : CAF and UPC
- Aprun command to launch jobs; similar to mpirun command
- PBSPro batch system (Hopefully installed 😊)
- Performance tools: CrayPat, Apprentice2, PAPI
- Totalview debugger (?)

Cray XT6(m) programming environment is SIMPLE

- Edit and compile MPI program (no need to specify include files or libraries)

```
$ vi mysrc.f90
$ ftn -o myexe mysrc.f90
```

- Edit PBSPro job file (myjob.job)

```
#PBS -N myjob
#PBS -l mppwidth=256
#PBS -j oe
cd $PBS_O_WORKDIR
aprun -n 256 ./myexe
```

- Run the job (output will be myjob.oxxxxx)

```
$ qsub myjob.job
```

The module tool on the Cray XT6(m)

- How can we get appropriate Compiler and Libraries to work with?
- module tool used on XT6(m) to handle different versions of packages (compiler, tools,...):
e.g.: **module load compiler1**
e.g.: **module switch compiler1 compiler2**
e.g.: **module load totalview**
.....
- taking care of changing of PATH, MANPATH, LM_LICENSE_FILE,... environment.
- user should not set those environment variable in his shell startup files, makefiles,....
- keep things flexible to other package versions

Cray XT6(m) programming environment: module list

```
fih000@ci203:~> module list
```

```
Currently Loaded Modulefiles:
```

- 1) modules
- 2) portals/2.2.0-1.0300.20621.14.2.ss
- 3) nodestat/2.2-1.0300.19875.7.12.ss
- 4) sdb/1.0-1.0300.20338.7.9.ss
- 5) MySQL/5.0.64-1.0000.2899.20.1.ss
- 6) lustre-cray_ss_s/1.8.1_2.6.27.42_0.1.1_1.0300.5160.7.1-1.0300.20335.8.30
- 7) Base-opts/1.0.2-1.0300.20274.11.1.ss
- 8) xtpe-network-seastar
- 9) cce/7.2.3
- 10) acml/4.4.0
- 11) xt-libsci/10.4.4
- 12) xt-mpt/4.1.1
- 13) pmi/1.0-1.0000.7628.10.2.ss
- 14) xt-asyncpe/3.9
- 15) PrgEnv-cray/3.0.20

- Current versions
 - PrgEnv- cray 3.0.20
 - Libsci 10.4.4

Cray XT6(m) programming environment: module show

```

fih000@ci203:~> module show pgi
-----
/opt/modulefiles/pgi/10.4.0:

setenv          PGI_VERSION 10.4
setenv          PGI_VERS_STR 10.4.0
setenv          PGI_PATH /opt/pgi/10.4.0
setenv          PGI /opt/pgi/10.4.0
prepend-path    PGROUPD_LICENSE_FILE /opt/pgi/license.dat
prepend-path    PATH /opt/pgi/10.4.0/linux86-64/10.4/bin
prepend-path    MANPATH /opt/pgi/10.4.0/linux86-64/10.4/man
prepend-path    LD_LIBRARY_PATH /opt/pgi/10.4.0/linux86-64/10.4/lib
prepend-path    LD_LIBRARY_PATH /opt/pgi/10.4.0/linux86-64/10.4/libso
module-whatis   PGI compiler for use on XTs
-----

fih000@ci203:~>

```

- Provides details about the module settings and files
 - Environment variables defined
 - Location of files

Cray programming environment: module avail

```
rosa> module avail
```

```
----- /apps/modulefiles -----
affinity/1.0.0          gsl/1.12                parpack/96
amber/10                hpct/2209_gnu432        scalasca/1.1
arpack/96               hpct/4.9.5              siesta/2.0.2
boost/1.38              ipm/0.923               szip/2.1
cpmd/3.13.2            lammmps/17Apr09         tau/2.18.1p1
ddt/2.4.1               lammmps/19May09         tau/2.18.2p2
espresso/4.0.4 (default) marmot/2.2.2_gnu        taujg/2.18.2p2
espresso/4.0.5          namd/2.6                udunits/2.1.5
gadget2/2.0.3           namd/2.7b1 (default)    vim/7.2
gnuplot/4.3svn          netcdf/4.0.0.3.1jg64    zlib/1.2.3
gotoblas/1.26           netcdf-hdf5parallel/4.0.0.3.1jg64
gromacs/4.0.3           nwchem/5.1.1
```

- Output not from the UDE machine !!!
- Provides info about the available modules
 - /opt/modulefiles: Cray provided modules
 - /apps/modulefile: modules created by CSCS

Cray XT6(m) programming environment: module avail

```
----- /opt/modulefiles -----
Base-opts/2.2.21A(default)
Base-opts/2.2.21A.lusrelsave
PrgEnv-cray/1.0.0(default)
PrgEnv-cray/1.0.1
PrgEnv-gnu/2.2.21A(default)
PrgEnv-pathscale/2.2.21A(default)
PrgEnv-pgi/2.2.21A(default)
acml/4.2.0(default)
apprentice2/4.4.0(default)
blcr/0.7.3
cce/7.0.3
cce/7.0.4(default)
cce/7.1.0
chapel/0.9(default)
cray/MySQL/5.0.64-1.0000.2260.12.4
cray/MySQL/5.0.64-1.0000.2342.16.1
cray/account/1.0.0-2.0202.16872.22.2
cray/audit/1.0.0-1.0202.15885.0
cray/csa/3.0.0-1_2.0202.16880.43.2
cray/job/1.5.5-0.1_2.0202.16889.26.2
cray/projdb/1.0.0-1.0202.16894.25.1
dwarf/8.8.0(default)
elf/0.8.10(default)
fftw/2.1.5.1
fftw/3.2.0(default)
fftw/3.2.1
gcc/4.2.4
gcc/4.3.2(default)
gcc/4.3.3
gnet/2.0.5
hdf5/1.8.2.1
hdf5/1.8.2.2
hdf5/1.8.2.3.1(default)
hdf5-parallel/1.8.2.1
hdf5-parallel/1.8.2.2
hdf5-parallel/1.8.2.3.1(default)
java/jdk1.6.0_11(default)
libfast/1.0.2(default)
libfast/1.0.4
mazama/4.0.0(default)
modules/3.1.6(default)
netcdf/4.0.0.1(default)
netcdf/4.0.0.2
netcdf/4.0.0.3.1
netcdf-hdf5parallel/4.0.0.1
netcdf-hdf5parallel/4.0.0.2
netcdf-hdf5parallel/4.0.0.3.1(default)
pathscale/3.2(default)
petsc/3.0.0.1(default)
petsc/3.0.0.3
petsc-complex/3.0.0.1(default)
petsc-complex/3.0.0.3
pgi/8.0.4
pgi/8.0.5(default)
pgi/8.0.6
pkgconfig/0.15.0(default)
totalview-support/1.0.6
xt-asyncpe/2.3
xt-asyncpe/2.4(default)
xt-asyncpe/2.5
xt-asyncpe/3.0
xt-boot/2.2.21
xt-boot/2.2.21A
xt-craypat/4.4.1(default)
xt-libsci/10.3.3
xt-libsci/10.3.4(default)
xt-libsci/10.3.6
xt-lustre-ss/2.2.21A_1.6.5
xt-mpt/3.1.2
xt-mpt/3.2.0(default)
xt-os/2.2.21A
xt-papi/3.6.2(default)
xt-pbs/10.0.0.82981
xt-pe/2.2.21A
xt-service/2.2.21A
xt-totalview/8.6.0
xtpe-target-cn1
```

Useful module commands

- Use profiling
module load xt-craypat
- Load the GNU environment
module swap PrgEnv-pgi PrgEnv-gnu
- Load Cray environment and use a specific version of the compiler
module swap PrgEnv-pgi PrgEnv-cray
module swap cce cce/7.2.2

Compilers



Compiler drivers to create CLE executables

- When the PrgEnv is loaded the compiler drivers are also loaded
 - By default CRAY compiler under compiler drivers (site setting)
 - the compiler drivers also take care of loading appropriate libraries (-Impich, -lsci, -lacml, -lpapi)
- Available drivers (also for linking of MPI applications):
 - Fortran 90/95 programs: ftn
 - Fortran 77 programs: f77
 - C programs: cc
 - C++ programs: CC
- Cross compiling environment
 - Compiling on a Linux service node
 - Generating an executable for a CLE compute node
 - Do not use standard compiler names (pgf90, mpicc,...) unless you want a Linux executable for the service node
 - Information message:

ftn: INFO: linux target is being used

Cray XT6(m) Compilers

- PGI
 - man pages: pgf90, pgcc, pgCC
 - PGI User's Guide (Chapter 2) <http://www.pgroup.com/doc/pgiug.pdf>
- GNU
 - module swap PrgEnv-pgi PrgEnv-gnu
 - man pages: gfortran, gcc, g++
- Cray
 - Loaded by default
 - man pages: crayftn, intro_directives, intro_pragmas
 - craydoc
http://docs.cray.com/cgi-bin/craydoc.cgi?mode=View;id=S-3901-71;idx=books_search;this_sort=;q=3901;type=books;title=Cray%20Fortran%20Reference%20Manual
 - Free bonus: assign, explain

Cray programming environment: assign

- Assigns options for library file open processing
assign [assign options] assign_object
- **Interesting assign options**
 - -R removes all assign options for assign_object
 - -N <numcon> specifies foreign numeric conversion
 - swap_endian the endian of data is swapped during unformatted input and output.
- **assign object** used to specify the object of assign options
 - f:<filename> applies to filename
 - p:<pattern> applies to files with name matching pattern
 - u:<unit> applies to Fortran unit number
 - g:su applies to all Fortran sequential unformatted files
 - g:du applies to all Fortran direct unformatted files

How to handle byte-swapped files with Cray compiler

- Explicit usage of assign
 - Can control which files are byte-swapped


```
export FILENV=.assign
assign -R
assign -N swap_endian f:aof
aprun a.out
```
- Link the application with -hbyteswapio
 - All unformatted Fortran I/O are byte-swapped
 - This is equivalent to set


```
assign -N swap_endian g:su
assign -N swap_endian g:du
```

Cray programming environment: explain

- Displays the explanation for an error message: compile, run time
- Compiler error

```
cft90: llvm/lib/VMCore/Instructions.cpp:328: void llvm::CallInst::init(llvm::Value*,
    llvm::Value* const*, unsigned int):
Assertion `(i >= FTy->getNumParams() || FTy->getParamType(i) == Params[i]->getType()) &&
    "Calling a function with a bad signature!" failed.
```

```
ftn-2116 crayftn: INTERNAL
```

```
rosa> explain ftn-2116
```

```
Internal : "Program" was terminated due to receipt of signal signal.
The process spawned by the command to run the specified program was
terminated by a system signal. See the signal(2) man page for more
information about this signal.
```

This message does not indicate a problem with your code, and you may be able to change your program so that this error does not occur.

Please notify your product support representative with the text of this message, the source code being compiled, the version of the compiler, and the Command line options in effect.

Cray programming environment: explain

- I/O runtime error

Open IOSTAT=5016

```
rosa> explain lib-5016
```

An EOF or EOD has been encountered unexpectedly.

The file terminated unexpectedly, perhaps without the proper end-of-file record or end-of-data control information. The file specification may be incorrect, or the file may be corrupted.

Ensure that the file specification is correct. If the file is corrupted, create the data again, if possible.

See the man pages for assign(1) and asgcmd(1).

The error class is UNRECOVERABLE (issued by the run time library).

Cray XT Compilers : Comparison

Feature	PGI	Cray
Listing	-Mlist	-ra
Diagnostic	-Minfo -Mneginfo	(produced by -ra)
Free format	-Mfree	-f free
Preprocessing	-Mpreprocess	-eZ -F
Suggested Optimization	-fast	(default)
Aggressive Optimization	-Mipa=fast,inline	-O3
Variables size	-r8 -i8	-s real64 -s integer64
Byte swap	-byteswapio	-hbyteswapio
OpenMP recognition	-mp=nonuma	(default)
Automatic parallelization	-Mconcur	n.a. (yet)

CNL Programming Considerations



Memory allocation

- Linux provides some environment variables to control how malloc behaves
 - Equivalent to using the mallopt system call
- MALLOC_MMAP_MAX_
 - Number of 'internal' non heap, mmap regions (default 64)
 - Setting this to 0 means that the program will not use any
 - Using MMAP regions turns out to be very costly compared to just using the heap
 - They exist to allow a program to return unused memory back to the system more easily so it may be used by other processes on the node: no need for it on the XT6(m)
 - Suggested value: `export MALLOC_MMAP_MAX_=0`

Memory allocation

- MALLOC_TRIM_THRESHOLD_
 - Amount of free space at the top of the heap after a free() that needs to exist before malloc will return the memory to the OS
 - Returning memory to the OS is costly. The default setting of 128 KBytes is much too low for a node with >4 GBytes of memory and one application.
 - Suggested value: `export MALLOC_TRIM_THRESHOLD_=536870912`
- GNU Description of mallopt options:

http://www.gnu.org/software/libc/manual/html_node/Malloc-Tunable-Parameters.html
- P. Ezolt - USENIX paper on malloc tuning:

http://www.usenix.org/publications/library/proceedings/als01/full_papers/ezolt/ezolt.pdf

Huge pages - description

- The AMD Opteron supports multiple page sizes
 - The base page size is 4 Kbytes.
 - The huge page size (also called "large pages") is 2 Mbytes.
- Huge pages can provide better performance by reducing the number of TLB misses and by enforcing larger sequential physical memory inside each page
- Memory fragmentation issue
 - the number of huge pages available on a node declines over time
 - it's affected by all applications that have run on the node since it was last booted.
- Applications that use huge pages but are unable to allocate enough:
 - if huge pages are required: **will fail to start**
 - if huge pages are advisory: **may run slower**

Huge pages - howto

- Link with the correct library: `-lhugetlbfs`
- Activate the library at run time: `export HUGETLB_MORECORE=yes`
- Launch the program specifying use of huge pages (**required mode**)
 - request <size> MBytes per PE `-m<size>h`
 - By default, it is fatal to the application if the request can't be satisfied
- Launch the program specifying use of huge pages (**advisory mode**)
 - request <size> MBytes per PE `-m<size>ha`
 - The application will proceed even if the request can't be satisfied – may run slower
- Example: `aprun -m700h -N2 -n8 ./my_app`
 - Requires 1400 MBytes of huge page memory on each node

Using the GNU „configure“ script on CLE

- Compiling on Linux, generating code for compute nodes => cross-compiling
- Setting variables:

```
setenv CC cc
setenv CXX CC
setenv F77 ftn
setenv F90 ftn
```

- Configure <options>
 - cross-compile to prevent execution of CLE executable on Linux node
`--host=x86_64-unknown-linux-gnu`
 - build only statically linked executables
`--enable-static`

MPICH name conflict

- There is a name conflict between `stdio.h` and MPI C++ binding in relation to the names `SEEK_SET`, `SEEK_CUR`, `SEEK_END`
- If your application does not use those names:
 - work with `-DMPICH_IGNORE_CXX_SEEK` to come around this
- If your application does use those names:
 - Set


```
#undef SEEK_SET
#include mpi.h
```
 - or change order of includes: `mpi.h` before `stdio.h` or `iostream`

MPI Communication on the Cray XT6(m)

(more in the next talk)

Cray XT MPI implementation – MPT (Message Passing Toolkit)

- Based on MPICH2
- Support for 256000 MPI ranks
 - Actually tested up to 150000 MPI ranks
- Environment variables available to tune communication

MPT 4.1 - features

- Automatically-tuned default values for MPICH environment variables
 - Several of the MPICH environment variable default values are now dependent on the total number of processes in the job
 - The new default values are displayed setting `MPICH_ENV_DISPLAY`
- Optimized collectives
 - Cray version of MPI collective routines are available and enabled by default (`MPI_Allreduce`, `MPI_Bcast`, `MPI_Alltoallv`)
 - The original MPICH-2 algorithm can be selected with the `MPICH_COLL_OPT_OFF` environment variable
- SMP aware global communication
 - `MPI_Bcast`, `MPI_Reduce`, ...
- MPI-IO performance improvements for collective buffering on MPI collective writes

Some MPI environment variables

Environment Variable	Description
MPICH_MAX_SHORT_MSG_SIZE	Maximum message size (bytes) to use EAGER protocol where sender sends message header+data to receiver. Default: $2048 * (100000 / \text{num_ranks})$ Min: 1024; Max: 128000
MPICH_UNEX_BUFFER_SIZE	Global size of the three unexpected message buffers maintained by each process. Default: $(\text{num_ranks} * 3000)$ Min: 60M; Max: 260M
MPICH_PTL_UNEX_EVENTS	Size of the event queue associated with short message buffers. Basically maximum number of short messages that can be outstanding at a given process. Default: $\max(\text{num_ranks} * 2.2, 20480)$

Some MPI environment variables

Environment Variable	Description
MPICH_PTL_MATCH_OFF	If set, disables registration of receive requests with portals. Setting this allows MPI to perform the message matching for the portals device. It may be beneficial to set this variable when an application exhausts portals internal resources and for latency-sensitive applications. Default: Not enabled
MPICH_MSGS_PER_PROC	Specifies the initial number of internal message headers allocated by MPI. If additional message headers are required during program execution, MPI dynamically allocates more message headers in quantities of MPICH_MSGS_PER_PROC. Default: 16384

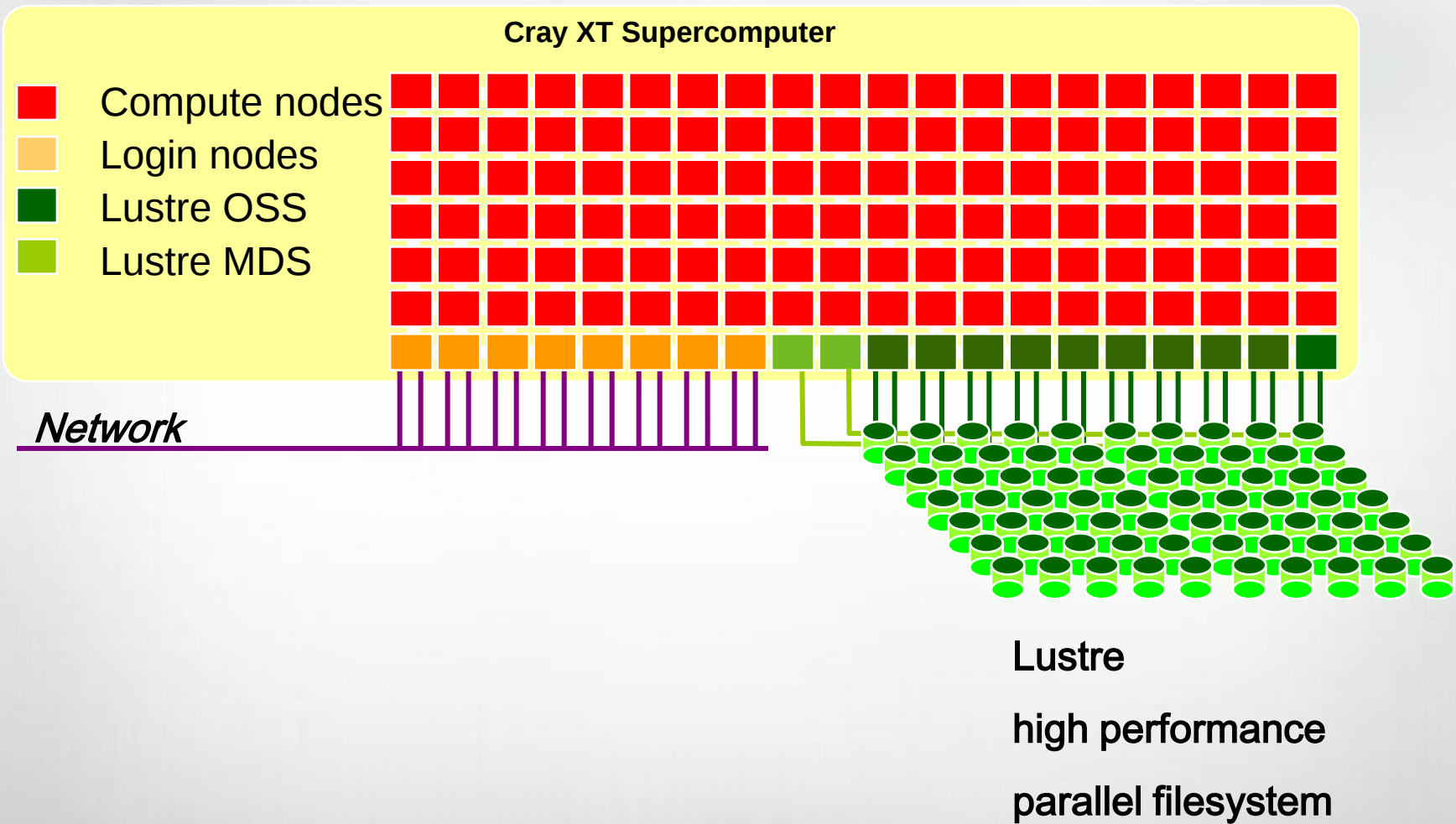
MPI-IO Collective Buffering

- MPI-IO performance improvements for collective buffering on MPI collective writes
- This optimization is enabled by
 - Setting the MPIIO hint romio_cb_write to "enable"
 - Setting the environment variable MPICH_MPIIO_CB_ALIGN to 2.
- For more info: mpi man page sections:
 - MPICH_MPIIO_CB_ALIGN
 - MPICH_MPIIO_HINT (collective buffering)
- New MPI-IO White Paper
ftp://ftp.cray.com/pub/pe/download/MPI-IO_White_Paper.pdf

I/O on the Cray XT6(m)



The Storage Environment



Cray XT I/O architecture

- All I/O is offloaded to service nodes
- Lustre
 - High performance parallel I/O file system
 - Direct data transfer between compute nodes and files
- DVS
 - Virtualization service
 - Allows compute nodes to access NFS mounted on service node
- No local disks
- /tmp is a MEMORY file system, on each login node

Lustre



- A scalable cluster file system for Linux
 - Developed by Cluster File Systems, Inc.
 - Name derives from “Linux Cluster”
 - The Lustre file system consists of software subsystems, storage, and an associated network
- **MDS** – metadata server
 - Handles information about files and directories
- **OSS** – Object Storage Server
 - **The hardware entity**
 - The server node
 - Support multiple OSTs
- **OST** – Object Storage Target
 - **The software entity**
 - This is the software interface to the backend volume

Lustre File Striping

- Stripes defines the number of OSTs to write the file across
 - Can be set on a per file or directory basis
- CRAY recommends that the default be set to
 - not striping across all OSTs, but
 - set default stripe count of one to four
- But not always the best for application performance.
 As a general rule of thumbs :
 - If you have one large file: stripe over all OSTs
 - If you have a large number of files (~2 times #OSTs): turn off striping

Lustre lfs command

- lfs is a lustre utility that can be used to create a file with a specific striping pattern, displays file striping patterns, and find file locations
- The most used options are :
 - setstripe
 - getstripe
 - df
- For help execute lfs without any arguments

```
$ lfs
```

```
lfs > help
```

```
Available commands are:
```

```
    setstripe
```

```
    find
```

```
    getstripe
```

```
    check
```

```
.....
```

lfs setstripe

- Sets the stripe for a file or a directory
- `lfs setstripe <file | dir> <-s size> <-i start> <-c count>`
 - stripe size: Number of bytes on each OST (0 filesystem default)
 - stripe start: OST index of first stripe (-1 filesystem default)
 - stripe count: Number of OSTs to stripe over (0 default, -1 all)
- Comments
 - The stripes of a file is given when the file is created. It is not possible to change it afterwards.
 - If needed, use `lfs` to create an empty file with the stripes you want (like the `touch` command)

Lustre striping hints

- For maximum aggregate performance: **Keep all OSTs occupied**
- Many clients, many files: **Don't stripe**
 - If number of clients and/or number of files $\gg$ number of OSTs:
Better to put each object (file) on only a single OST.
- Many clients, one file: **Do stripe**
 - When multiple processes are all accessing one large file:
Better to stripe that single file over all of the available OSTs.
- Some clients, few large files: **Do stripe**
 - When a few processes access large files in large chunks:
Stripe over enough OSTs to keep the OSTs busy on both write and read paths.

Running an application on the Cray XT6(m)



Running an application on the Cray XT

- ALPS
 - Application Level Placement Scheduler
- aprun is the ALPS application launcher
 - It must be used to run application on the XT compute nodes
 - If aprun is not used, the application is launched on the login node (and likely fails)

Job Launch



XT6(m) User

Login Node

SDB Node

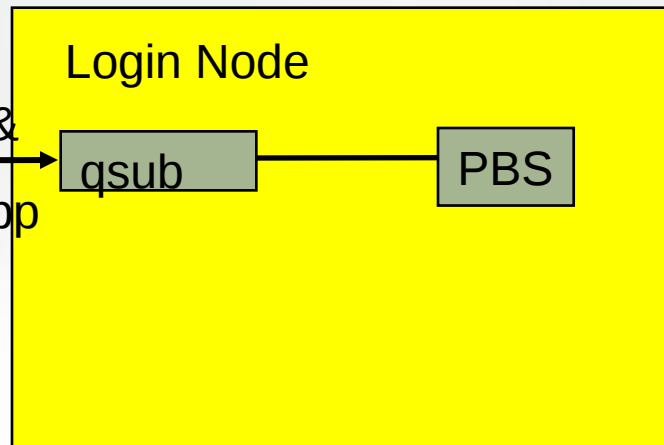
Compute Nodes

Job Launch



XT6(m) User

Login &
Start App



SDB Node

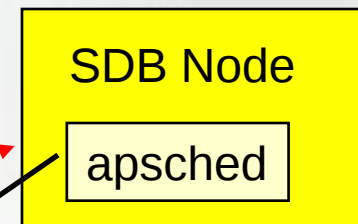
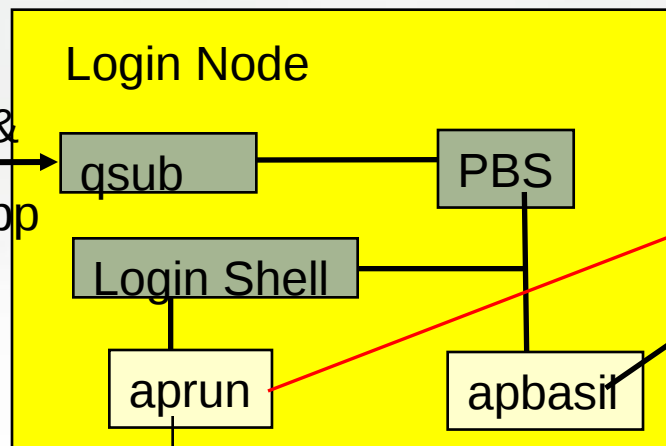
Compute Nodes

Job Launch

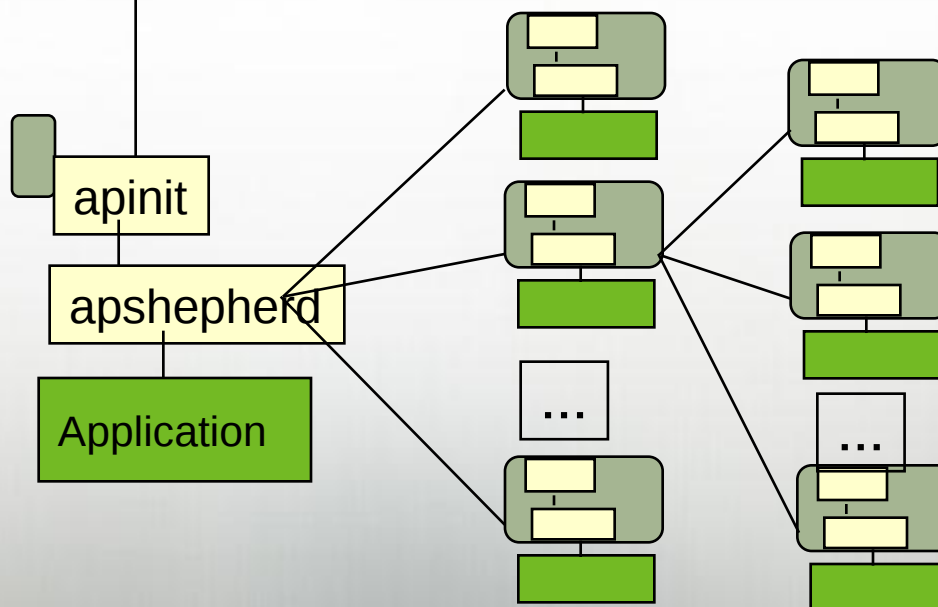


XT6(m) User

Login &
Start App



Compute Nodes

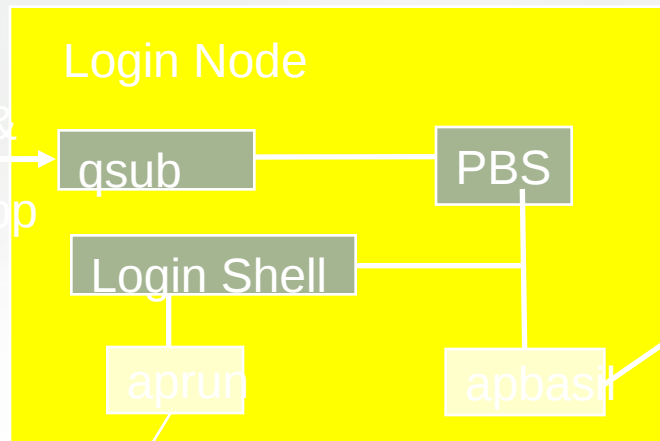


Job Launch

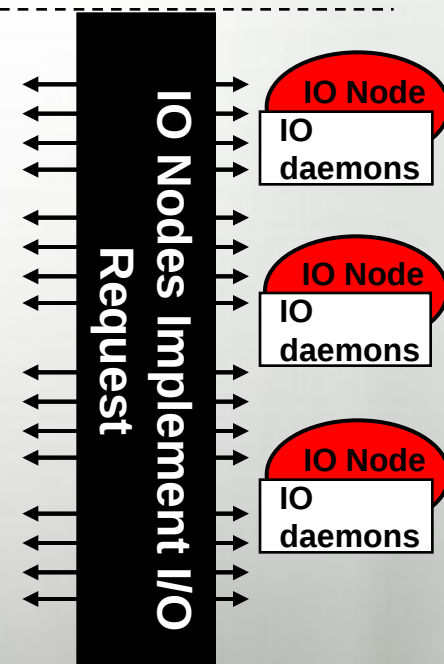
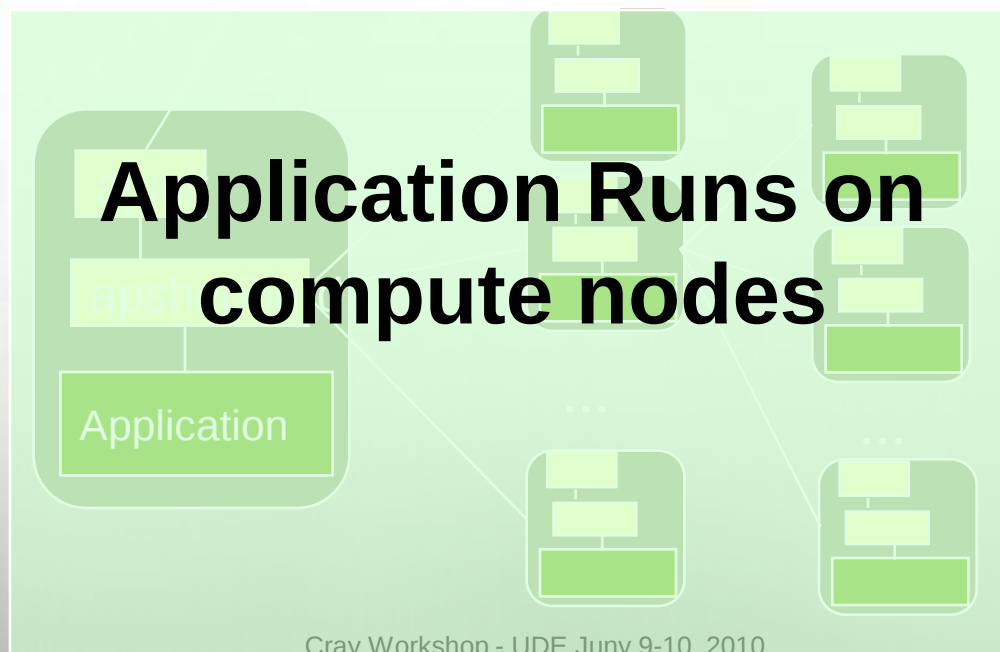


XT6(m) User

Login &
Start App



Compute Nodes

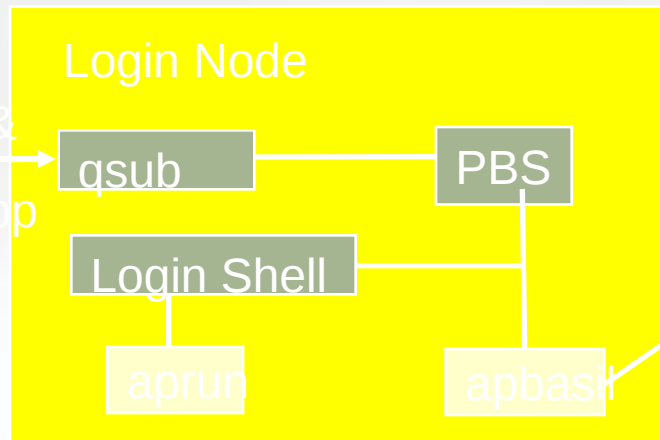


Job Launch



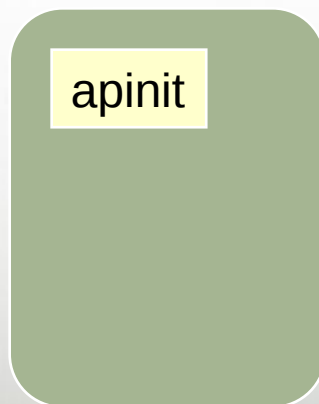
XT6(m) User

Login &
Start App



Compute Nodes

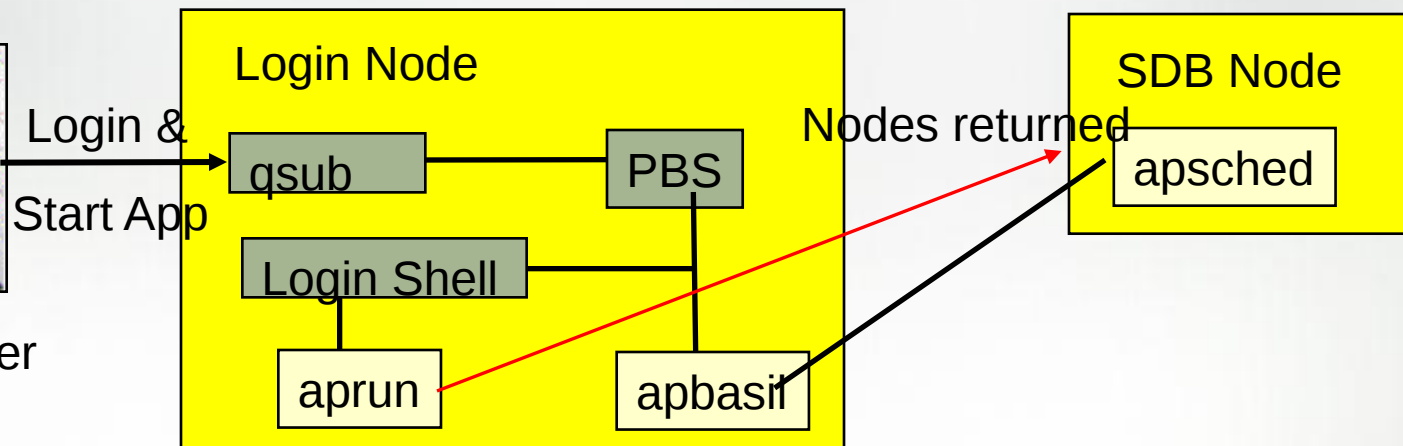
Job is
cleaned up



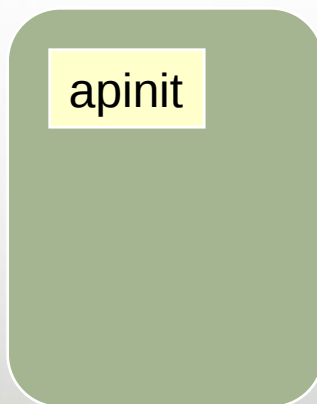
Job Launch



XT6(m) User



Compute Nodes



Running an application on the Cray XT - aprun

- aprun has (at least) 3 important parameters to control:
 - how many nodes are required
 - how many MPI tasks (PEs) are used
 - how many cores are used on a multicore chip
 - how many OpenMP threads are required
- These aprun parameters correspond to PBS parameters
 - If an aprun parameter is specified, also the corresponding PBS one should be specified

Description	aprun	PBS
Total number of MPI tasks	-n	-l mppwidth
Number of MPI tasks per node	-N	-l mppnppn
Number of MPI tasks per socket	-S	>None<
Number of OpenMP threads per task	-d	-l mppdepth

Running an application on the Cray XT - examples

Remember to set the corresponding PBS option in the job header

- Assuming a XT6(m) system (24 cores per node)
- Pure MPI application, using all the available cores in a node
 - npes MPI tasks, npes cores, npes/24 nodes

```
$ aprun -n <npes>
```
- Pure MPI application, using only 1 core per node
 - npes MPI tasks, 24*npes cores allocated, npes nodes allocated
 - Can be done to give all the memory on the node to the MPI tasks

```
$ aprun -N 1 -n <npes>
```
- Hybrid MPI/OpenMP application
 - npes MPI tasks, 6 OpenMP threads each, 6*npes cores, npes/4 nodes
 - need to set OMP_NUM_THREADS

```
$ export OMP_NUM_THREADS=6
$ aprun -N 4 -d 6 -n <npes>
```

aprun CPU Affinity control

- CPU affinity options enable to bind a PE or thread to a particular CPU or a subset of CPUs on a node
- CNL can dynamically distribute work by allowing PEs and threads to migrate from one CPU to another within a node
- In some cases, moving PEs or threads from CPU to CPU increases cache and translation lookaside buffer (TLB) misses and therefore reduces performance
- aprun CPU affinity option:
 - -cc cpu_list | keyword
 - suggested settings: -cc cpu
 - The cpu keyword (the default) binds each PE to a CPU within the assigned NUMA node

aprun Memory Affinity control

- Cray XT6(m) systems use dual-socket 12-core compute nodes
 - The XT6(m) has 4 NUMA nodes with 6 cores each.
- Remote-NUMA-node memory references, such as a process running on NUMA node 0 accessing NUMA node 1 memory, can adversely affect performance.
- aprun memory affinity options:
 - -S pes_per_numa_node #PEs to allocate per NUMA node
 - -sl list_of_numa_nodes list of NUMA nodes to use [0-3]
 - -sn numa_nodes_per_node #NUMA nodes to consider [1,2,3,4]
 - -ss strict memory containment per NUMA node
a PE can allocate only the memory local to its assigned NUMA node
 - Suggested settings: -ss

Running an application on the Cray XT - MPMD

- aprun supports MPMD – Multiple Program Multiple Data
- Launching several executables on the same MPI_COMM_WORLD

```
$ aprun -n 128 exe1 : -n 64 exe2 : -n 64 exe3
```

Running a batch application with PBSPro

- The number of required nodes and cores is determined by the parameters specified in the job header

```
#PBS -l mppwidth=256
```

```
#PBS -l mppnppn=4
```

- The job is submitted by the qsub command
- At the end of the execution output and error files are returned to submission directory
- PBS environment variable: \$PBS_O_WORKDIR
Set to the directory from which the job has been submitted

Other PBSPro parameters

- #PBS -N job_name
the job name is used to determine the name of job output and error files
- #PBS -l walltime=hh:mm:ss
Maximum job elapsed time
should be indicated whenever possible: this allows PBS to determine best scheduling strategy
- #PBS -j oe
job error and output files are merged in a single file
- #PBS -q queue
request execution on a specific queue: usually not needed

MPI Multi-core vs MPI Single-core vs Hybrid MPI+OpenMP

All the examples allocate 32 XT6(m) nodes (=768 cores)

ALL CORES (most common)

```
#PBS -N ALLjob
#PBS -l mppwidth=768
#PBS -j oe

aprun -n768 myfile.exe
```

SINGLE CORE

```
#PBS -N SCjob
#PBS -l mppwidth=32
#PBS -l mppnppn=1
#PBS -j oe

aprun -N1 -n32 myfile.exe
```

Hybrid MPI + OpenMP

```
#PBS -N OMPjob
#PBS -l mppwidth=128
#PBS -l mppnppn=4
#PBS -l mppdepth=6
#PBS -j oe

export OMP_NUM_THREADS=6
aprun -N4 -d6 -n128 myfile.exe
```

Watching a launched job on the Cray XT

- xtnodestat
 - Shows XT nodes allocation and aprun processes
 - Both interactive and PBS
- apstat
 - Shows aprun processes status
 - apstat overview
 - apstat -a [apid] info about all the applications or a specific one
 - apstat -n info about the status of the nodes
- PBS qstat command
 - shows batch PBS jobs
 - qstat -r check running jobs
 - qstat -n check running and queued jobs
 - qstat -s <job_id> reports comments about the job

The ideal job

```
#PBS -N goodjob
#PBS -l mppwidth=1024
#PBS -l mppnppn=4
#PBS -l mppdepth=6
#PBS -l walltime=01:00:00
#PBS -j oe

cd $PBS_O_WORKDIR

export MPICH_ENV_DISPLAY=1
#export MPICH_PTL_MATCH_OFF=1
#export MPICH_MSGS_PER_PROC=64000

export MALLOC_MMAP_MAX_=0
export MALLOC_TRIM_THRESHOLD_=536870912

export OMP_NUM_THREADS=6
#export PSC_OMP_AFFINITY=FALSE

aprun -N4 -d6 -n 1024 -cc cpu -ss ./goodexe
```

Documentation

- Cray docs site
 - <http://docs.cray.com>
- Starting point for Cray XT info
 - http://docs.cray.com/cgi-bin/craydoc.cgi?mode=SiteMap;f=xt3_sitemap
- Twitter ?!?



- <http://twitter.com/craydocs>



Extra: basic compiler options



PGI programming environment

- **Overall Options**

- -Mlist creates a listing file
- -Wl,-M generates a loader map (to stdout)
- -Minfo info about optimizations performed
- -Mneginfo info on why certain optimizations are not performed

- **Preprocessor Options**

- -Mpreprocess runs the preprocessor on Fortran files (default on .F, .F90, or .fpp files)

- **Optimisation Options**

- -fast chooses generally optimal flags for the target platform
- -Mipa=fast,inline Inter Procedural Analysis
- -Minline=levels:n number of levels of inlining

PGI programming environment

- **Language Options**

- -Mfree process Fortran source using free form specifications
- -Mnomain useful for using the ftn driver to link programs with the main program written in C or C++ and one or more subroutines written in Fortran
- -i8, -r8 treat INTEGER and REAL variables as 64-bit
- -Mbyteswapio big-endian files in Fortran; XT6(m) is little endian

- **Parallelization Options**

- -mp=nonuma recognize OpenMP directives
- -Mconcur automatic parallelization

man pages: pgf90, pgcc, pgCC

PGI User's Guide (Chapter 2) <http://www.pgroup.com/doc/pgiug.pdf>

GNU programming environment

- Module available
module swap PrgEnv-pgi PrgEnv-gnu
- Default compiler version is 4.3.2

man pages: gfortran, gcc, g++

Cray Compiler Environment (CCE): Brief History of Time

- Cray has a long tradition of high performance compilers
 - Vectorization
 - Parallelization
 - Code transformation
 - More...
- Began internal investigation leveraging an open source compiler called LLVM
- Initial results and progress better than expected
- Decided to move forward with Cray X86 compiler
- 7.2 is the current version

Cray Opteron Compiler: Current Capabilities

- Fortran, C and C++ support
- Excellent Vectorization
 - Vectorize more loops than other compilers
- OpenMP
 - 3.0 standard
 - Nesting
- PGAS: Functional UPC and CAF available today.
 - Performance with Gemini
- Excellent Cache optimizations
 - Automatic Blocking
 - Automatic Management of what stays in cache
- Prefetching, Interchange, Fusion, and much more...

Cray compiler flags

- Module available
module swap PrgEnv-pgi PrgEnv-cray

- **Overall Options**
 - -ra creates a listing file with optimization info

- **Preprocessor Options**
 - -eZ runs the preprocessor on Fortran files
 - -F enables macro expansion throughout the source file

- **Optimisation Options**
 - -O2 optimal flags [enabled by default]
 - -O3 aggressive optimization
 - -O ipa<n> inlining, n=0-5

Cray compiler flags

- **Language Options**

- -f free process Fortran source using free form specifications
- -s real64 treat REAL variables as 64-bit
- -s integer64 treat INTEGER variables as 64-bit
- -hbyteswapio big-endian files in Fortran; XT6(m) is little endian
this is a link time option

- **Parallelization Options**

- -O omp Recognize OpenMP directives [enabled by default]
- -O thread<n> n=0-3, aggressive parallelization, default n=2

man pages: [crayftn](#), [intro_directives](#), [intro_pragmas](#), [assign](#)

http://docs.cray.com/cgi-bin/craydoc.cgi?mode=View;id=S-3901-71;idx=books_search;this_sort=;q=3901;type=books;title=Cray%20Fortran%20Reference%20Manual