CRAY® THE SUPERCOMPUTER COMPANY

Cliff's Developing for the Cray XT3

Jeff Larkin
Cray Supercomputing Center of Excellence
larkin@cray.com
http://www.ccs.ornl.gov/~larkin/

2/16/2006

Overview

- Programming Environment
- Program Execution
- Debugging Tools
- Performance Tools

Programming Environment

- Remember, you are cross-compiling for catamount.
- Compiler wrappers handle the cross-compiling
  - C – cc
  - C++ – CC
  - F77/F90 – ftn
- Wrappers handle includes and linking for MPI and set include and link paths for Shmem, CrayPat, ACML, and LibSci (when modules are loaded)
- PGI, GCC compiler suites available through compiler wrappers
  - PGI (default) – module load PrgEnv-pgi
  - GCC – module load PrgEnv-gcc
    - No F90 support
- Note: to suppress target warnings use:
  - -target=catamount

PGI Compiler Options

- Option help: -help <option>
  - Must call PGI compiler directly for –help.
- Listing file: -Mlist
- Additional compile-time information: -Minfo
  - =[inline|ipa|loop|opt|stat|time|all]
- -byteswapio: Swap byte-order for unformatted input/output
  - Useful when moving data between jaguar and phoenix
- -r[4,8] –i[2,4,8]: Controls interpretation of real and integer sizes.
  - This may affect library compatibility.

PGI Optimization Flags

- -O0-4: Set optimization level, -O0 to -O4, default -O2
- -fast: “Good” optimizations
  - -O2 -Munroll=c:1 -Mnoframe -Mlre
- -fastsse: “Good” SSE optimizations
- -fast -Mvect=sse -Mscalarsse -Mcache_align –Mflushz
- More Advanced Options
  - -Mvect: Control automatic vector pipelining
  - -Mipa: Enable Interprocedural Analysis
  - -Mscalarsse: Generate scalar sse code with xmm registers; implies –Mflushz
  - -Munroll: Enable loop unrolling
  - -Minline: Inline all functions that were extracted

Extra Libraries

- I have provided several additional libraries in my home directory (~larkin/xt3)
  - fftw2, fftw3
  - HDF5
  - NetCDF, parallel-NetCDF
  - zlib, and szip (for HDF5)
- To load as modules:
  - module use ~larkin/xt3/modules
- If you have problems with these libraries, please e-mail larkin, not the helpdesk.
- Some of these libraries have now been made available by NCCS as well.

Program Execution: yod

- The XT3 program launcher is yod
- Important options:
  - -sz: number of nodes
  - -small_pages: use 4KB memory pages rather than the default 2MB
  - -shmem size: sets symmetric heap size
  - -stack size: sets the stack size
  - -heap size: sets heap size
- Examples:
  - yod –sz 16 a.out
  - yod –sz 128 –shmem 2M a.out

Program Execution: PBS

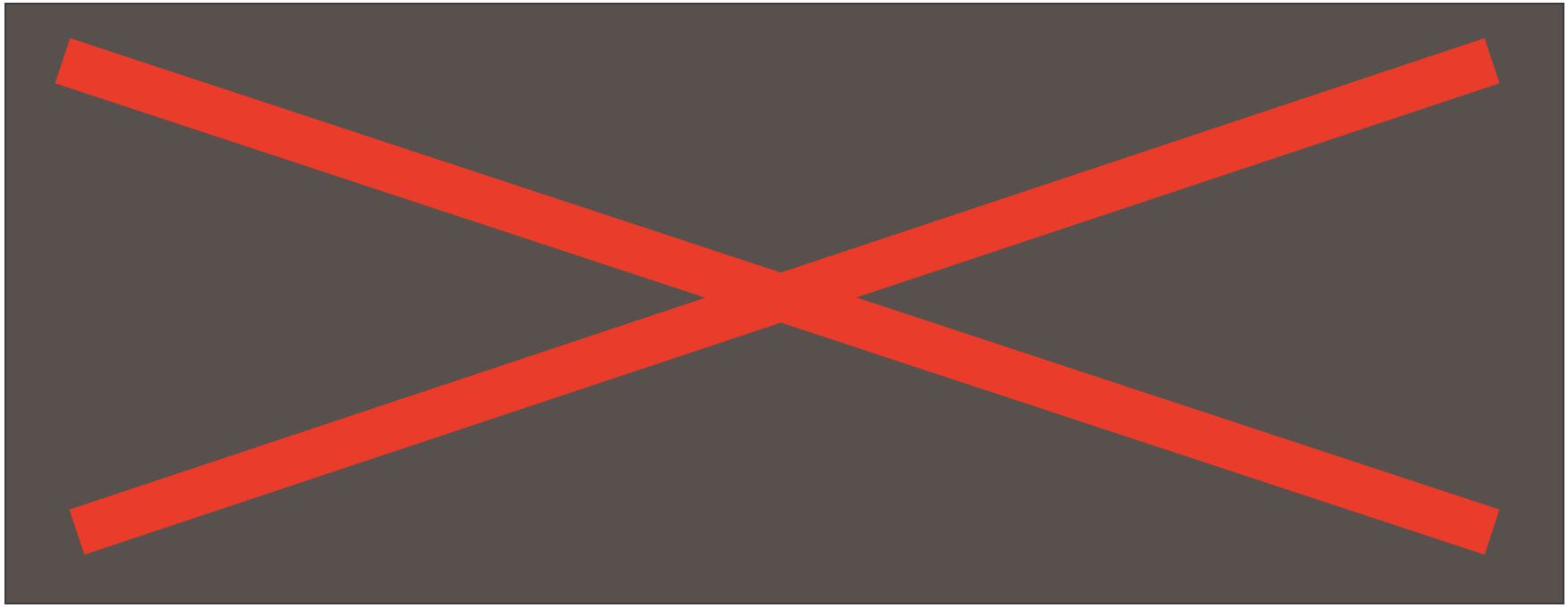
```
#!/bin/bash
 Request 64 nodes for up to 90 min.
#PBS -l size=64,walltime=1:30:00
 Name my job “example” and give my account
#PBS -N example -A acctnum
 Join my output and mail me when done
#PBS -j oe -m ae

cd $PBS_O_WORKDIR
yod -sz $PBS_NNODES a.out
```

Program Execution: Useful Commands

- xtps: Lists running processes
  - -Y: Display the status of all compute (yod) jobs.
- xtkill: Kills processes running on an Cray XT3 system
  - Use only in emergency
  - Use yod pid from xtps
- xtshowmesh: Shows information about compute and service partition processors and the jobs running in each partition
  - -y <pid>: Displays just mesh allocated to <pid>
- qstat: Show status of pbs batch jobs

Program Execution: Filesystems



- /tmp/work/USER is a parallel filesystem. It provides excellent performance and lots of space. If possible, **run your codes here**.
  - Data should be stored on this filesystem!

Program Execution: Lustre

- A lustre filesystem is available on Jaguar at /tmp/work/USER
- OST: Object Storage Target, the basic storage unit.
- Stripe width: The level of parallelism for your file
- Stripe size: The size of the buffer
- The default stripe width is 1 and all files will inherit this unless set otherwise
- The default stripe size is 1048576.
- The total number of OSTs is 48.

Program Execution: Lustre

- What should my stripe width be?
  - Lots of processes writing to individual files: sw = 1
  - 1 I/O process writing for all: sw = small, but > 1
  - Lots of processes all writing to 1 file: sw = large
- Set your stripe width:
  - `lfs setstripe <filename> <stripe-size> <start-ost> <stripe-cnt>`
  - Using 5 OSTs
    - `$ lfs setstripe file 0 -1 5`
  - Using all OSTs
    - `$ lfs setstripe file 0 0 -1`
- See ~jlbeck/lustre.101, or man lfs for more information.

Debugging Tools: totalview

- Totalview 7.1.0 is available on Jaguar
- To use: module load totalview
- Core file analysis
  - `$ tv7 a.out core`
- Launch from within interactive session
  - `$ tv7 <tvopts> yod -a <yodopts> a.out <opts>`
- Attach to running processes
  - Launch totalview as above
  - From within totalview, attach to your yod process.
- Totalview Users' Guide: http://www.etnus.com/Documentation/index.php

Cray PAT: Building & Executing

- Load the craypat module: module load craypat
- Build your executable: make clean; make
  - Fortran90 modules require `-Mprof=func` during compile and link
- Add profiling: `pat_build <options> app app+pat`
  - Object files must be present and built after craypat module was loaded
- Execute your new executable: `yod ... app+pat`
  - A file or directory will be created at the end of execution
  - The name of the file or directory is printed at the end of your program output

Cray PAT: Build Options

- `-u`
  - Profile user functions
- `-g <group>`
  - Profiles pre-defined groups (MPI, I/O, etc.)
- `-T <entry point>`
  - Trace individual functions
- `-w`
  - Do a tracing experiment by default.

Cray PAT: HWPC Data

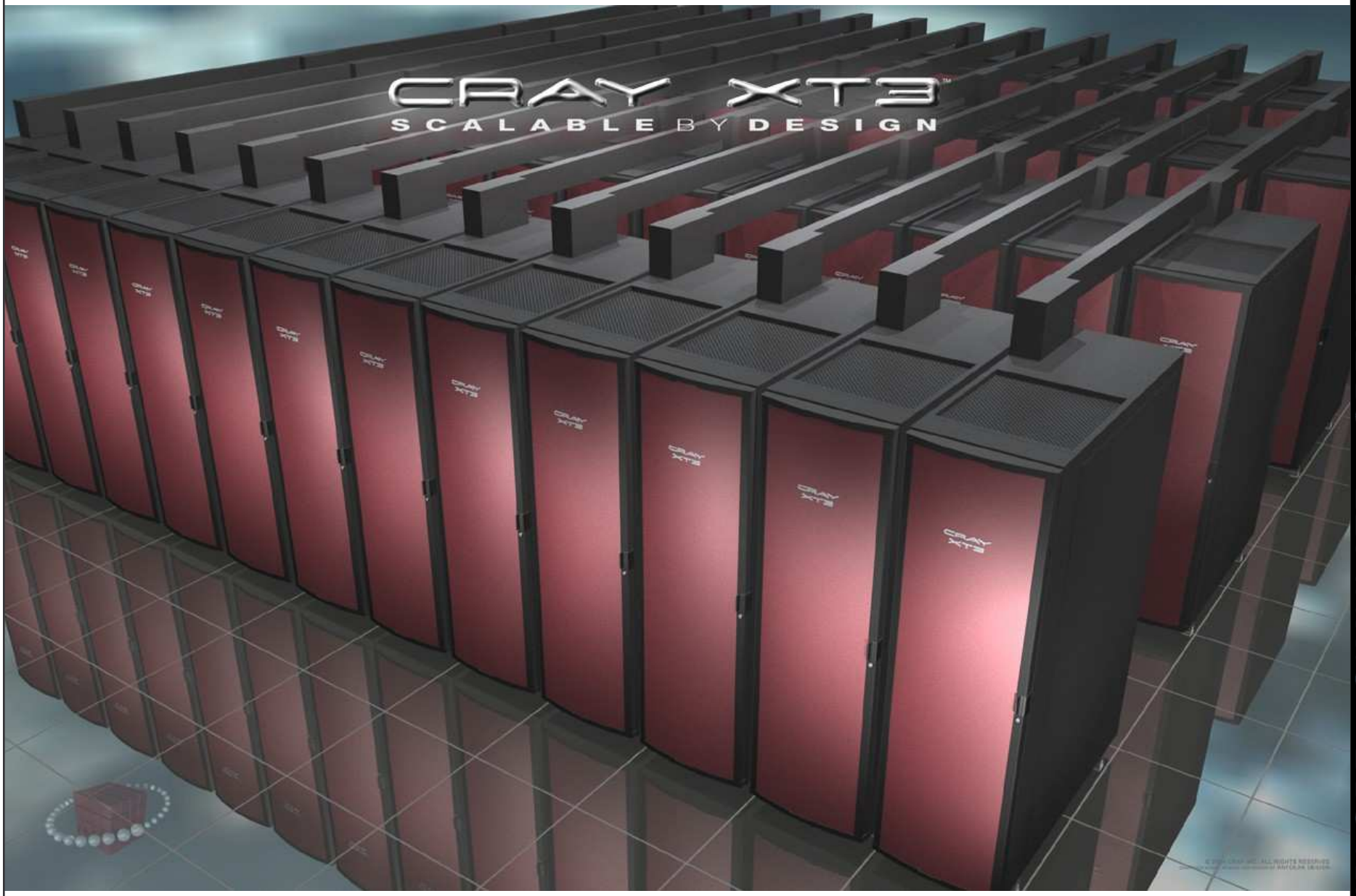
- HWPC data is organized into 9 groups
  1. FP, LS, L1 Misses, & TLB Misses
  2. L1 & L2 Data Accesses & Misses
  3. L1 Accesses, Misses, & bandwidth
  4. Floating Point Mix
  5. Floating Point Mix (2)
  6. Total cycles stalled
  7. Total cycles stalled (2)
  8. Instructions & Branches
  9. Instruction cache
- See man hwpc for details
- Set PAT_RT_HWPC=grpnum in your PBS script to define which group to use.

Cray PAT: Generating a Report

- pat_report generates a report from the performance file.
  - Text Report
  - XML Report
  - Apprentice^2 report
- `pat_report [-c stats|records] [-f txt|xml|ap2] [-o output_file] [-d d-opts] [-b [b-opts] performance_file`

Apprentice^2: Introduction

- When graphics say it best…
- Apprentice 2 can provide a more interactive look at your PAT data.
- Using Apprentice 2:
  - Generate an apprentice file from pat_report
    - *pat_report –f ap2 …*
  - Launch Apprentice 2
    - *module load apprentice2*
    - *app2 report.ap2*


CRAY XT3™
SCALABLE BY DESIGN