Introduction to Fish

David Newman

(from Thomas Logan

Slides created by Oralee Nudson)

ARSC

UNIVERSITY OF ALASKA FAIRBANKS

Overview

- Connecting to Fish
- Hardware
- Programming Environment
- Compilers
- Queueing System
- Interactive Work
- Software Stack

Image 1: Fish Cabinets

Connecting to Fish

- ARSC supports **ssh**, **sftp** and **scp** clients
- Linux and OS come with native command line versions of **ssh**, **sftp** and **scp**.
- Windows requires you download a terminal and file transfer program (and optionally an X11 server). We have the most experience supporting PuTTY, FileZilla, and Xming.

ARSC

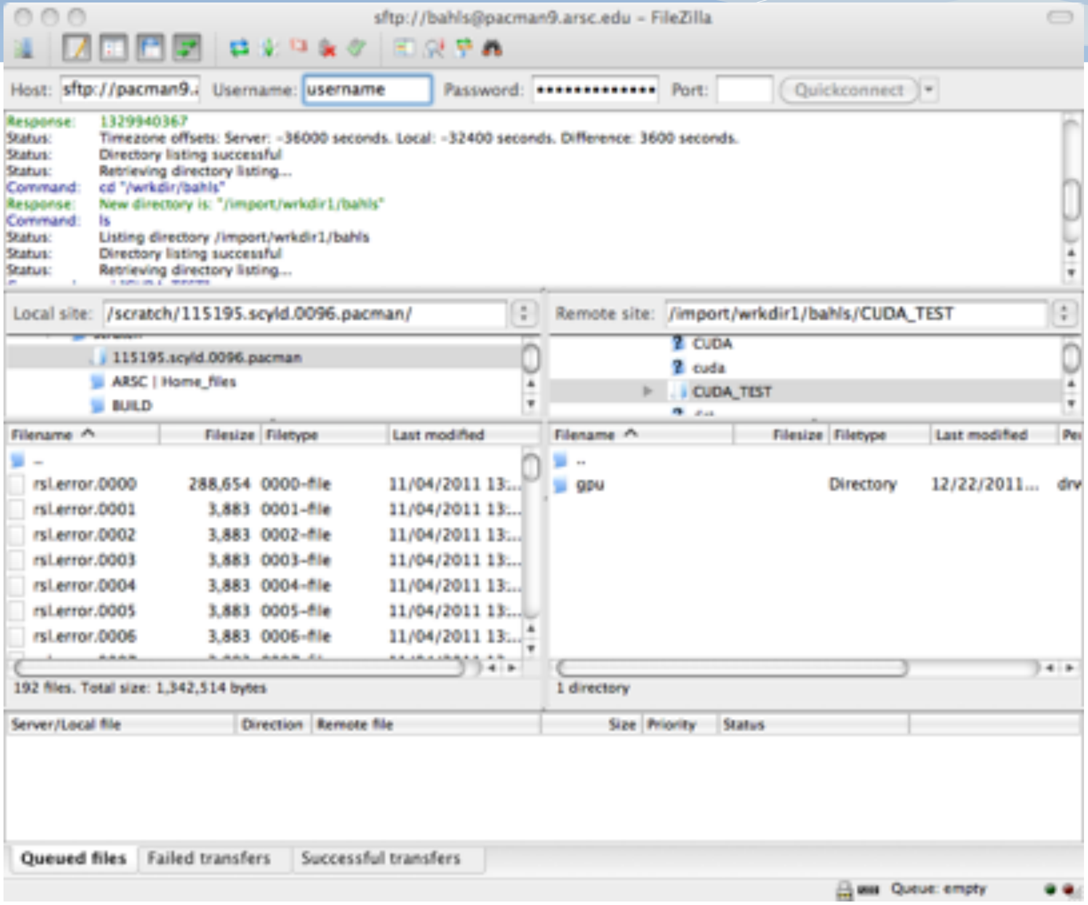
UAF UNIVERSITY OF ALASKA FAIRBANKS

Other File Transfer Alternatives

- File Transfer – Mac and Linux have GUI based file transfer programs (e.g. Fetch and Filezilla) which can be convenient. Others programs should work so long as they support either sftp or scp. Plain ftp ***will not*** work for file transfer files to ARSC systems.
- Rsync works too when started via ssh.

ARSC

UAF UNIVERSITY OF ALASKA FAIRBANKS



sftp://bahls@pacman9.arsc.edu - FileZilla

Host: sftp://pacman9. Username: username Password: •••••••••••••• Port:

```
Response:   1329940367
Status:     Timezone offsets: Server: -36000 seconds. Local: -32400 seconds. Difference: 3600 seconds.
Status:     Directory listing successful
Status:     Retrieving directory listing...
Command:    cd "/wrkdir/bahls"
Response:   New directory is: "/import/wrkdir1/bahls"
Command:    ls
Status:     Listing directory /import/wrkdir1/bahls
Status:     Directory listing successful
Status:     Retrieving directory listing...
```

Local site: /scratch/115195.scyld.0096.pacman/

Remote site: /import/wrkdir1/bahls/CUDA_TEST

Filename	Filesize	Filetype	Last modified
..			
rsl.error.0000	288,654	0000-file	11/04/2011 13:...
rsl.error.0001	3,883	0001-file	11/04/2011 13:...
rsl.error.0002	3,883	0002-file	11/04/2011 13:...
rsl.error.0003	3,883	0003-file	11/04/2011 13:...
rsl.error.0004	3,883	0004-file	11/04/2011 13:...
rsl.error.0005	3,883	0005-file	11/04/2011 13:...
rsl.error.0006	3,883	0006-file	11/04/2011 13:...

192 files. Total size: 1,342,514 bytes

Filename	Filesize	Filetype	Last modified	Pe
..				
gpu		Directory	12/22/2011...	drv

1 directory

Server/Local file	Direction	Remote file	Size	Priority	Status

Image 2: Filezilla on a Mac

Other Windows Connection Options

- If you prefer the command line environment, Cygwin offers command line ssh, sftp and scp commands as well as an X11 Server.
- For more information on Cygwin see:
  - http://www.arsc.edu/arsc/support/howtos/usingcygwin/index.xml

FISH – Cray XK6m

- **2 Login Nodes: {fish1,fish2}.arsc.edu**
  - One six core, 2.6 GHz AMD Istanbul Processor
  - 16 GB of memory per node
- **48- GPU Enabled Sixteen Core Compute Nodes**
  - One sixteen core, 2.1 GHz AMD Interlagos Processor
  - 64 GB o memory per node (4 GB per core)
  - One nVIDIA Tesla X2090 GPU accelerator with 6GB RDDR5 memory
- **32- Twelve Core Compute Nodes**
  - 2- Six core 2.6 GHz AMD Istanbul Processors
  - 32 GB of memory per node (2.5 GB per core)
- **Cray Proprietary Gemini Interconnect**
- **20 TB Lustre $HOME file system**
- **275 TB Lustre $CENTER file system**

ARSC

UAF UNIVERSITY OF ALASKA FAIRBANKS

Fish Storage

- Fish supports the ARSC standard storage locations
  - **$HOME**: small, backed up, not purged
  - **$CENTER**: large, not backed up, purged
  - **$ARCHIVE**: no quota, backed up, not purged.
  - NOTE:
    - $ARCHIVE not available from compute nodes! $ARCHIVE is only accessible to the login nodes
    - ARSC recommends avoiding accessing $HOME in parallel jobs.
    - Purging is not currently enabled on $CENTER but may be enabled at a future date (with ample warning)

ARSC

UNIVERSITY OF ALASKA FAIRBANKS

Monitoring Storage Usage

- Quotas are enabled on $HOME and $CENTER. Use the “show_storage” command to list your current usage:

```
fish1 % show_storage
Current Filesystem Quotas as of Thu Mar  8 14:07:47 AKST 2012:

Filesystem  Used_GB   Soft_GB  Hard_GB      Files  Soft Files Hard Files
========== ========  ======== ======== ==========  ========== ==========
$HOME          3.50      4.19    16.50      66712           0          0
$CENTER      240.58    786.43  1048.58     143791     5000000   52000000
```

Parallel Programming Models

- Threads (OpenMP and pthreads)
- MPI- using the Cray's MPI with Gemini support.
- Serial Applications
- Autoparallelization (via the PGI and Cray compilers)
- PGAS languages (via the Cray compilers)
- GPU Acceleration

ARSC

UAF UNIVERSITY OF ALASKA FAIRBANKS

Parallel Programming Models

Level	**Model**	**Description**
Node	Auto	Automatic parallelization of basic loops. Only available with PGI compilers (use –Mconcur= option)
Node	OpenMP	Explicit shared memory model using directives to achieve loop level parallelism (use –mp option with PGI compilers)
System	MPI	Most common and portable method for scalable distributed memory parallelism
Node	OpenACC	Compiler directives to specify loops and regions of code to be offloaded from CPU to GPU accelerators.
Node	GPU	PGI and Cray compilers support OpenACC directives for interacting with GPUs. PGI compiler supports CUDA Fortran and CUDA C. NVidia nvcc compiler is also available.
System	PGAS	Cray Compiler Environment supports the Partitioned Global Address Space (PGAS) languages- UPC and CoArray Fortran.

ARSC

UNIVERSITY OF ALASKA FAIRBANKS

Modules

- Fish uses a package called "modules" which allows you to quickly and easily switch from one software version to another.
- This makes it possible for us to provide multiple versions of a software package
- One person can access the newest version of a package while someone else has an unchanging environment.

Modules

- Modules do the following:
  - Set your $PATH, $MANPATH, $LD_LIBRARY_PATH
  - Set environment variables needed by packages (e.g. NCARG_ROOT for NCL, etc).
  - Keep track of what they set so the environment variables and aliases can be unset when a module is unloaded.

Modules

- By default all accounts have a compiler stack loaded which is set using the modules package.
- You can put alternate and additional modules commands in shell login files.

Module Name	Description
`PrgEnv-pgi`	Programming environment using the PGI compilers & Cray's MPI stack (default module).
`PrgEnv-cray`	Programming environment using the Cray compiler and Cray's MPI stack .
`PrgEnv-gnu`	Programming environment using GNU compilers & Cray's MPI stack .

Sample Module Commands

Command	**Example Use**	**Purpose**
module avail	module avail	lists all available modules for the system.
module load <*pkg*>	module load PrgEnv	loads a module file from the environment
module unload <*pkg*>	module unload PrgEnv	unloads a module file from the environment
module list	module list	displays the modules which are currently loaded.
module switch old new	module switch PrgEnv-pgi PrgEnv-gnu	replaces the module old with module new in the environment
module purge	module purge	unload all module settings, restoring the environment to the state before any modules were loaded. (Avoid this command on fish!)

ARSC

UNIVERSITY OF ALASKA FAIRBANKS

Module Use

- Modules will include (some already installed):
  - abaqus, cmake, ferret, git, grads, idl, java, matlab, nco, ncl, ncview, octave, petsc, python, totalview.
- If the default version of a package is not what you want, check to see if a newer version is available.
  - The modules command lets you see all versions of a package (e.g. “module avail ncl” lists all ncl modules)

Module Command Examples

```
fish1 % module avail PrgEnv-pgi

fish1 % module list

fish1 % module load PrgEnv-pgi

fish1 % module list

fish1 % module swap PrgEnv-pgi/4.0.46 PrgEnv-cray/4.0.46

fish1 % module list
```

Modules in jobs

- If you compile with XTPE_LINK_TYPE=dynamic and used non-default modules, be sure to load which ever modules you had loaded at compile time loaded in your job script.

```
#!/bin/bash
#PBS -q standard
...

source /opt/modules/default/init/modules.sh
module swap pgi pgi/12.6.0

aprun -n 64 ./wrf.exe
```

ARSC

UNIVERSITY OF ALASKA FAIRBANKS

Last Words on Modules

- You can create your own modules if you want to maintain multiple versions of programs you build yourself.
- If you have question how to do it let us know.

Compilers on Fish

Item	Cray	PGI	GNU
Fortran 77	`ftn`	`ftn`	`ftn`
Fortran 90/95	`ftn`	`ftn`	`ftn`
C	`cc`	`cc`	`cc`
C++	`CC`	`CC`	`CC`
Serial Debugger	`lgdb`	`lgdb`	`lgdb`
Parallel Debugger	`totalview`	`totalview`	`totalview`
Performance Analysis	`Cray perftools`	`Cray perftools`	`Cray perftools`
Default MPI Module	`PrgEnv-cray`	`PrgEnv-pgi`	`PrgEnv-gnu`

A few PGI compiler options

-c	Generate object file but don't link
-g	Add debugging information
-O3	Higher level of optimization (default is –O2)
-fast	Higher level of optimization than –O3
-Mipa	Perform interprocedural analysis
-Mconcur	Enable autoparallelization
-mp	Enables parallelization via OpenMP directives
-Minfo	Provides additional information on optimizations done by the compiler. This flag has numerous options (e.g. –Minfo=mp,par)
-Mneginfo	Provides additional information on why optimizations are ***not done*** by the compiler.

ARSC

UAF UNIVERSITY OF ALASKA FAIRBANKS

Batch System

- **Batch queuing**
  - Allows job scheduling on shared compute nodes
  - Requires users to specify resource requests
  - Ensures that nodes are shared fairly
  - Manages resources to maximize throughput
- **Fish uses the Torque/Moab/ALPS batch queuing system, which is based on PBS, the Portable Batch System**

ARSC

UAF UNIVERSITY OF ALASKA FAIRBANKS

Batch Script Contents

1. **Queueing Commands**
   - Shell dialect (e.g. bash, ksh)
   - Execution queue to use
   - Job walltime
   - Number of nodes required
2. **Commands to Execute**
   - File/Directory Manipulations
   - Code to execute

Torque Script - MPI Example

```
#!/bin/bash
#PBS -q standard
#PBS -l walltime=8:00:00
#PBS -l nodes=4:ppn=12
#PBS -j oe

cd $PBS_O_WORKDIR

NP =$(( $PBS_NUM_NODES * $PBS_NUM_PPN ))
aprun -n $NP ./myprog
```

Torque Script - OpenMP

```
#!/bin/bash
#PBS -q standard
#PBS -l walltime=8:00:00
#PBS -l nodes=1:ppn=12
#PBS -j oe

cd $PBS_O_WORKDIR
export OMP_NUM_THREADS=12
aprun -n 1 -d 12 ./myprog
```

Torque Script – MPI w/ OpenMP

```
#!/bin/bash
#PBS -q gpu
#PBS -l walltime=8:00:00
#PBS -l nodes=4:ppn=16
#PBS -j oe

cd $PBS_O_WORKDIR

export OMP_NUM_THREADS=$PBS_NUM_PPN
NP=$PBS_NUM_NODES
aprun -n $NP -d ${OMP_NUM_THREADS} ./myprog
```

Common PBS commands

- `qsub job.pbs` - queue the script "job.pbs" to be run by PBS.
- `qstat` - list jobs which are queued, running or recently completed.
- `qdel` *`jobid`* - delete a job with ID=jobid from the qstat command and is also provided when qsub is run.

ARSC

UAF UNIVERSITY OF ALASKA FAIRBANKS

Common Queues

- **standard** - regular CPU only work. Uses 12 core nodes. 24 hour max walltime.
- **gpu** - CPU + GPU work. Uses 16 core nodes. 24 hour max walltime.
- **debug** – Runs on GPU nodes. 1 hour max walltime
- Use "**news queues**" to find out more details on number of nodes and walltime allowed per queue

ARSC

UAF UNIVERSITY OF ALASKA FAIRBANKS

Example Batch Queue Use

```
# Submit job
fish1 % qsub run_hello.cmd
2067.sdb

# Check job status
fish1 % qstat 2067.sdb
Job id                 Name            User            Time Use S Queue
---------------------- --------------- --------------- -------- - -----
2067.sdb               run_hello.cmd   fred                   0 Q gpu

# Check job status a bit later
fish1 % qstat 2067.sdb
Job id                    Name            User            Time Use S Queue
------------------------- --------------- --------------- -------- - -----
2067.sdb                  run_hello.cmd   fred            00:00:01 C  gpu

# Output from job is returned to the working directory by default.
fish1 % ls run_hello.cmd.o2067
run_hello.cmd.o206782
```

Select Torque Environment

- Torque sets environment variables for use within job scripts.
  - `$PBS_JOBID` – Job id assigned by torque.
  - `$PBS_NUM_NODES` – Number of nodes assigned to job.
  - `$PBS_NUM_PPN` – Number of cores assigned per node.
  - `$PBS_O_WORKDIR` – Original working directory on job submission.
  - `$PBS_ARRAYID` – Assigned to the array index when –t is used.

Example Torque Environment Variable Use

```
#!/bin/bash
#PBS -l nodes=4:ppn=12
#PBS -q standard
#PBS -j oe
#PBS -l walltime=1:00:00


cd $PBS_O_WORKDIR





NP=$(( $PBS_NUM_NODES * $PBS_NUM_PPN ))
aprun -n ${NP} ./a.out > job.${PBS_JOBID}.out 2>&1
```

More Torque Features

- There a number of other features that we aren’t going to cover today, but may be useful:
  - **Job dependencies** – make one job run after another.
  - **Job arrays** – Let you submit an array of jobs.
  - **Command line options** – You can override settings in a torque script when a job is submitted.

ARSC

UAF UNIVERSITY OF ALASKA FAIRBANKS

Interactive Work

- There are two fish login nodes: fish1.arsc.edu and fish2.arsc.edu
- You may run short serial work on fish1 and fish2, however CPU intensive, memory intensive or parallel work should be done on the compute nodes using an interactive job.
- Interactive compute nodes are subject to availability.

Example Interactive Work

```
# Load a Programming Environment
fish2 % module load PrgEnv-gnu

# Compile the code
fish2 % make
cc hello.c -O3  -o hello.exe

# Start an interactive job
fish2 % qsub -lnodes=2:ppn=12 -q standard -I

# run the program on the compute nodes.
nid00186 % cd mpi/
# the "aprun" is important!  Even if you run a serial job!
nid00186 % aprun -n 4 ./hello.exe
nid00064: hello world- 0
nid00064: hello world- 1
nid00064: hello world- 3
nid00064: hello world- 2
```

ARSC

UNIVERSITY OF ALASKA FAIRBANKS

Software Stack

- Fish has a software stack based on requests from people on campus. If we have a package installed there's likely already someone on campus using that package.
- When possible we put example job scripts and test cases in **$SAMPLES_HOME**
- If you want a test case for a package, let us know and we will make one!

ARSC

UAF UNIVERSITY OF ALASKA FAIRBANKS

Additional Information

- Watch www.arsc.edu for upcoming training and additional information about ARSC
- More info at
  - ARSC How to: http://www.arsc.edu/arsc/support/howtos/usingpacman
  - PGI Compilers: http://www.pgroup.com/resources/docs.htm
  - GNU Compilers: http://gcc.gnu.org/
  - OpenACC: http://www.openacc-standard.org/
  - Cray Compilers: http://docs.cray.com

Questions and Feedback

- If you have additional questions, feel free to contact us:

  ARSC Help Desk

  Phone: (907) 450-8602

  Email: consult@arsc.edu

  Web: http://www.arsc.edu/support

- We welcome your feedback on this class!