



Ohio Supercomputer Center

An **OH·TECH** Consortium Member

High-Performance Programming

Using Vectorization, OpenMP, MPI, and OpenACC to get better performance on a single core, a single socket, a single node, and on multi-nodes.

David Hudak

Ohio Supercomputer Center



Ohio Supercomputer Center

An **OH·TECH** Consortium Member

Part 1 - Eclipse “Starter Kit” for OSC

Starter Kit Contents

- This presentation
 - Goal is to start programming ASAP
- Parallel Tool Platform (PTP) tutorial
 - Ptp-xsede2013.pdf – detailed instructions and examples
- Test Project
 - test.c – C program using MPI
 - Makefile - compile script for project
 - test-pbs.sh – job script for project



Agenda

- Installing Eclipse
- Making A New Synchronized Project
- Configuring The Project
- Building The Project
- Running The Project
- Monitoring The Project



See slides 3-19 in PTP tutorial!
Also, slides 20-37 are a good overview.

Installing Eclipse (1 of 2)

- Download Eclipse for parallel application developers
 - <http://www.eclipse.org/downloads/>
 - Use the Indiana University (IU) mirror for faster download
- Open Eclipse
 - Create Workspace
 - Click the check box to use that workspace as default every time Eclipse starts up
- Eclipse is organized as a set of *perspectives*
 - C/C++ Perspective, Remote System Explorer Perspective, etc.
- Code in Eclipse is organized into *projects*



Installing Eclipse (2 of 2)

- Update PTP

- Go to “Help->Install New Software”

- Site: <http://download.eclipse.org/tools/ptp/updates/kepler>

- Click “Select All”, “Finish” and accept license agreement

- Restart Eclipse after updating

- Install Shell Ed (shell script editor)

- Go to “Help->Install New Software”

- Site:

- <http://sourceforge.net/projects/shelled/files/shelled/update/>

- Click “Select All”, “Finish” and accept license agreement

- Restart Eclipse after updating



See slides 39-66 in PTP tutorial!

Making A New Synchronized Project (1 of 6)

- Synchronized Projects contain locally stored files on your laptop that are automatically synced to a remote system to build and execute
- Three step process for a new synchronize project:
 - Step 1: Create a project
 - Step 2: New Synchronized Project Dialog Box
 - Step 3: Add files



Making A New Synchronized Project (2 of 6)

- Step 1: Create a project
 - Right click in Project Explorer window, select “New->Synchronized C/C++ Project”
- Step 2: New Synchronized Project Dialog Box overview:
 - “Project Name” – Name of project
 - “Remote Directory” – Location of project on remote system
 - “Project Type” – C, C++, Fortran, etc.
 - “Remote Toolchain” – compiler on remote system
 - “Local Toolchain” – compiler on local system (not used)



Making A New Synchronized Project (3 of 6)

- Step 2: New Synchronized Project Dialog Box
 - “Project Name”
 - Type in a name like “test”
 - “Remote Directory”
 - “Connection Name”
 - Select an existing connection, or
 - To make a new connection, click the “New...” button
 - » A new window should pop up
 - » “Target name”: type “oakley” *- Glenn*
 - » “Host”: type “oakley01.osc.edu”
 - » “User”: your OSC username (for example osc1234)
 - » “Password”: your OSC password
 - » Click “Finish”



Making A New Synchronized Project (4 of 6)

- Step 2: New Synchronized Project Dialog Box

- ~~Project Name~~

- Remote Directory

- Connection Name - oakley

- Remote Directory

- Click “Browse” button

- You will see a file browser for your home directory at OSC

- Create a directory, or just type “OK” to put project in home directory

— where to

put the code.



Making A New Synchronized Project (5 of 6)

- Step 2: New Synchronized Project Dialog Box
 - ~~Project Name~~
 - ~~Remote Directory~~
 - Project Type
 - Open “Makefile Project” folder ✓
 - Select “Empty Project”
 - Remote Toolchain
 - Select “Linux GCC” —
 - Local Toolchain
 - Don’t select anything —
 - Click “Finish”



Making A New Synchronized Project (6 of 6)

- ~~Step 1: Create a project~~
- ~~Step 2: New Synchronized Project Box~~
- You should see a “test” project folder in Project Explorer
- Step 3: Add files
 - Find “test” project folder in Project Explorer
 - Drag-and-drop project files onto “test” folder
 - test.c, Makefile, test-pbs.sh



See slides 82-96 in PTP tutorial!

Configuring The Project (1 of 4)

- Configure project to use correct compilers, libraries and include files from remote system
- Two step process
 - Step 1: Add remote include paths
 - Step 2: Customize remote environment

*where
is the material*

*Customize
the environment.*



See slides 87-89 in PTP tutorial!

Configuring The Project (2 of 4)

- Step 1: Add remote include paths
 - Right-click on “test” project folder in Project Explorer
 - Select “Properties”
 - Open “C/C++ General”
 - Select “Preprocessor Include Paths”
 - Select “GNU C”, then “CDT User Setting Entries”, then click “Add...” button

*Where
headers
are etc.*



Configuring The Project (3 of 4)

- Step 1: Add remote include paths (continued)
 - In upper right, select “File System Path” in pulldown
 - For path, type “//oakley/usr/local/mvapich2/1.9-intel/include”
 - General syntax for remote path: //<connection><path>
 - Check “Contains system headers”
 - Click “OK” for “Add Include Directory” dialog box
 - Click “OK” for “Properties” dialog box
 - Double-click on test.c to open it in an editor
 - MPI symbols should be resolved



See slides 124-140 in PTP tutorial!


Configuring The Project (4 of 4)

- Step 2: Customize remote build environment
 - Right-click on “test” project folder in Project Explorer
 - Select “Properties”
 - Select “Synchronize”
 - Click the box for “Use an environment management system...”
 - Click the box for “Manually specify environment...”
 - Type (or cut and paste) commands from this box
 - Click “OK”

```
module purge >/dev/null 2>&1  
module load modules/1.0  
module unload mvapich2  
module unload intel  
module load intel/13.1.3.192  
module load mvapich2/1.9
```



Building The Project

- Click the little hammer 
- Console window will have compiler output
- Under the “Project” menu you will find
 - “Build Project”
 - “Clean...” to remove builds



See slides 143-156 in PTP tutorial!

Running The Project

- Run Projects on Oakley via PBS job manager
- Step 1 (and only): Create Run Configuration
 - Under “Run” menu, select “Run Configurations...”
 - Select “Parallel Application”, click “New” button
 - Name: type “test”
 - “Resources” Tab
 - Target system Configuration: PBS-Generic-Batch
 - Connection name: select “oakley”
 - “Resources” Tab (con’t)
 - Click “Import PBS Script”
 - Click “Browse Workspace”
 - Select “test” folder
 - Select “test-pbs.sh” file
 - Click “OK”
 - “Application” Tab
 - “Application Program”, click “Browse”
 - Select “test”
 - Click “Apply”
 - Click “Run”



See slides 157-163 in PTP tutorial!

Monitoring The Project

- After clicking the Run button, Eclipse will change to the “System Monitoring” perspective
- “Active Jobs” has running jobs
- “Inactive Jobs” has queued or completed jobs
- Once job status has changed from Submitted to Completed
 - Go back to C/C++ Perspective
 - Right click on test folder in Project Explorer
 - Select “Synchronize” - “Sync Active Now” to get files from Oakley
 - File with name like test-pbs.o1234567 has job output



See slides 143-156 in PTP tutorial!

Running The Project – Version 2

- Run Projects on Oakley via PBS job manager
- Step 1 (and only): Create Run Configuration
 - Under “Run” menu, select “Run Configurations...”
 - Select “Parallel Application”, click “New” button
 - Name: type “test”
 - “Resources” Tab
 - Target system Configuration: PBS-Generic-Batch
 - Connection name: select “oakley”
 - Set number of nodes to “2:ppn=12”
 - Set wallclock time to “00:05:00”
 - Set MPI command to “mpiexec”
 - Set MPI number of tasks to “24”



See slides 143-156 in PTP tutorial!

Running The Project – Version 2

- Step 1 (and only): Create Run Configuration
 - “Resources” Tab
 - For modules to load, click “Configure...”
 - Click “Use an environment management system”
 - Click “Manually specify environment configuration”
 - Add text from box below
 - Click “OK”
 - “Application” Tab
 - “Application Program”, click “Browse”
 - Select “test”
 - Click “Apply”
 - Click “Run”

```
module purge >/dev/null 2>&1
module load modules/1.0
module unload mvapich2
module unload intel
module load intel/13.1.3.192
module load mvapich2/1.9
```



See slides 157-163 in PTP tutorial!

Monitoring The Project – Version 2

- After clicking the Run button, Eclipse will change to the “System Monitoring” perspective
- “Active Jobs” has running jobs
- “Inactive Jobs” has queued or completed jobs
- Once completed, right click on job and select “Get job output”
 - Output will appear in Console
- Similarly, “Get job error” will return error (if one occurs)

