Eclipse PTP: Advancing the development of scientific applications
Outline

1. introduction of Eclipse PTP
2. Installation
3. Develop with Eclipse PTP
4. Murphy's law time
1

Introduction of Eclipse PTP
Parallel Tools Platform (PTP)

PTP provide a highly integrated environment specifically designed for parallel development:

- An integrated development environment (IDE) that supports a wide range of parallel architectures and runtime systems
- A scalable parallel debugger
- Parallel programming tools for MPI and OpenMP
- Support for the integration of parallel tools
- An environment that simplifies the end-user interaction with parallel systems
- Remote development tools
PTP features
Why PTP?

- Simplify development over several computers / clusters...

  Classic case:
  - 1 source code repository
  - 1 copy on a local computer (the most active)
  - X copy on several clusters and these versions:
    - are not-up-to-date or with local modifications that cannot be committed
    - are under versioning or not
    - receive changes from the repository / rsync / scp / sshfs

- With Eclipse PTP, there are synchronized / remote projects
Synchronized projects
Remote projects
MPI-Specific features

- PTP has several features specifically for developing MPI code
  - Show MPI Artefacts
  - Code completion
  - Context sensitive help for MPI
  - Hover help
  - MPI Templates in the editor
  - MPI Barrier analysis
Debugging features

- Debugging requires interactive access to the application
Performance tuning and analysis tools

- There are tools integrated with PTP to help enhance performance of parallel applications:
  - PTP External Tools Framework (ETFw): provide integration for instrumentation, measurement and analysis for a variety of performance tools
  - TAU: parallel profiling and tracing tool
  - GEM: graphical explorer of MPI Programs
2

Installation
Eclipse packages

- Eclipse is available in several packages for different kinds of development:

- There is a package directly relevant for HPC:
  - This is recommended for all new installs.

- You can also add PTP to an existing Eclipse installation.
Update and install PTP modules

- Enable PTP-specific update site:
  - Help > Install New Software
  - Click **Available Software Sites** link
  - Ensure this checkbox is selected for the PTP site:
  - Choose **OK**

- Now install all features of PTP
  - Help > Install New Software
  - Select PTP update site in the Work with box
  - Check all the box, then click next and accept all...
Add SVN support - Subclipse

- Add Subclipse-specific update site:
  - Help > Install New Software
  - Click Add...
  - Fill name and location of Subclipse update site:
    - http://subclipse.tigris.org/update_1.6.x
  - Choose OK

- Now install SVN feature
  - Help > Install New Software
  - Select Subclipse update site in the Work with box
  - Check the box “Subclipse”, then click next and accept all...
Add GIT support - EGit

- Enable EGit-specific update site:
  - Help > Install New Software
  - Click Available Software Sites link
  - Ensure this checkbox is selected for the Egit site: http://download.eclipse.org/egit/updates
  - Choose OK

- Now install EGit feature
  - Help > Install New Software
  - Select EGit update site in the Work with box
  - Check the box “Eclipse EGit”, then click next and accept all...
Add CUDA support - YDL

- Add YDL-specific update site:
  - Help > Install New Software
  - Click Add...
  - Fill name and location of YDL update site:
    - http://ydl.net/eclipse_cuda_plugin/galileo/
  - Choose OK

- Now install all features of YDL
  - Help > Install New Software
  - Select YDL update site in the Work with box
  - Check the box “CUDA”, then click next and accept all...
Add CMake support - CMakeEd

- Add CMakeEd-specific update site:
  - Help > Install New Software
  - Click Add...
  - Fill name and location of CMakeEd update site:
    - http://downloads.sourceforge.net/project/cmakeed/eclipse/
    - Choose OK

- Now install CMakeEd feature
  - Help > Install New Software
  - Select CmakeEd update site in the Work with box
  - Check the box “CMakeEditor, then click next and accept all...
3

Develop with Eclipse PTP
Importing a project form SVN

Switch to **SVN Repository Exploring** perspective:
- Windows > Open Perspective > Other...
- Select **SVN Repository Exploring** and select **OK**

Import your project:
- Right Click in **SVN Repositories** view
- Select **New > Repository Location**...
- Fill the different fields...
Checking out and create new project

- Checking out the project:
  - Expand the repository location
  - Select what you want to check out (trunk, branches...)
  - Right click and select **Check Out As**...
  - On **Check Out As** dialog, select **Finish**

- Create a new project:
  - As project is checked out, the **New project** Wizard helps you to configure Eclipse information
  - Select **project type** and a **toolchain** that matches your system.
  - Enter a **project name** and click **Finish**
Converting to synchronized project

- Allow to build and execute project on remote machine
  - Select File > New > Other...
  - Open the Remote folder
  - Select Convert C/C++ or Fortran Project to a synchronized Project and select Finish

- Convert the project:
  - Select checkbox next to your project name
  - Select Remote Tools as remote provider
  - Create a new connection to the remote machine you want
  - Select the remote location to store your project
  - Select Finish
Resource managers

- PTP uses the term « resource manager » to refer to any subsystem that controls the resources required for launching a parallel job.

- Examples:
  - Batch scheduler (e.g. LoadLeveler, PBS, SLURM)
  - Interactive execution (e.g. OpenMPI, MPICH2, etc.)

- Each resource manager controls one target system
- Resource managers can be local or remote
Configuring job scheduler

- Setting up a resource manager is done in the System Monitoring perspective:
  - Select **Windows > Open Perspective > Other...**
  - Choose **System Monitoring** and click **OK**
  - Right click in Resource Managers View and select **Add Resource Manager**
  - Choose a resource manager type
  - Configure the connection by choosing created during the conversion of your project to a synchronized project
  - Select **Finish**

- You can now start your now resource manager
Running a job

- In the System Monitoring perspective, you have to create a run configuration (e.g. PBS script...):
  - Select Run > Run Configurations...
  - Select Parallel Applications
  - Select New

- You have to fill all the classic informations about the execution of your program:
  - In Resources tab: resource manager, parameter of scheduler...
  - In Application and Arguments tab: project, name, arguments...

- During the run of your application, you can monitor your job and get job error and output at the end.
Murphy's law time
Some words on the example

- The characteristics of the example are as follows:
  - the project is under cmake,
  - the project require a build out-of-source
  - the project use cuda and others libraries (ex: BLAS).

- Before launching eclipse, you have to configure your environment with the common environment variables (PATH, LD_LIBRARY_PATH, INCLUDE…) to allow eclipse to use it.
Comments on video n°2

- When you build your project with eclipse, it uses the makefile that is in the top directory.
- But, we want to use the cmake mechanism to build our project.
- We have to use the following tricks:
  - Create a build folder in the workspace
  - Create a makefile in the build folder not called makefile
  - Modify the build rule in eclipse to called these makefile in the build folder instead of makefile in the top folder
  - These makefile have to call cmake and makefile generated by cmake
  - That's all...
- Warning: The same have to be done for remote.
Comments on video n°6

- When you want to execute a program on a remote platform and you need to have a specific environment, modify your .bashrc on the remote machine accordingly.
- [http://wiki.eclipse.org/Parallel_Tools_Platform_FAQ#Q:_My_remote_or_synchronized_project_doesn’t_find_the_remote_environment_variables_correctly_.28Interactive_vs._non-interactive_shell.29](http://wiki.eclipse.org/Parallel_Tools_Platform_FAQ#Q:_My_remote_or_synchronized_project_doesn’t_find_the_remote_environment_variables_correctly_.28Interactive_vs._non-interactive_shell.29)
Thank you

Inria Bordeaux – Sud-Ouest
http://sed.bordeaux.inria.fr