

Module 4: Working with MPI

★ Objective

- ★ Learn how to develop, build and launch a parallel (MPI) program on a remote parallel machine

★ Contents

- ★ Remote project setup
- ★ Building with Makefiles
- ★ MPI assistance features
- ★ Working with resource managers
- ★ Launching a parallel application

Local vs. Remote

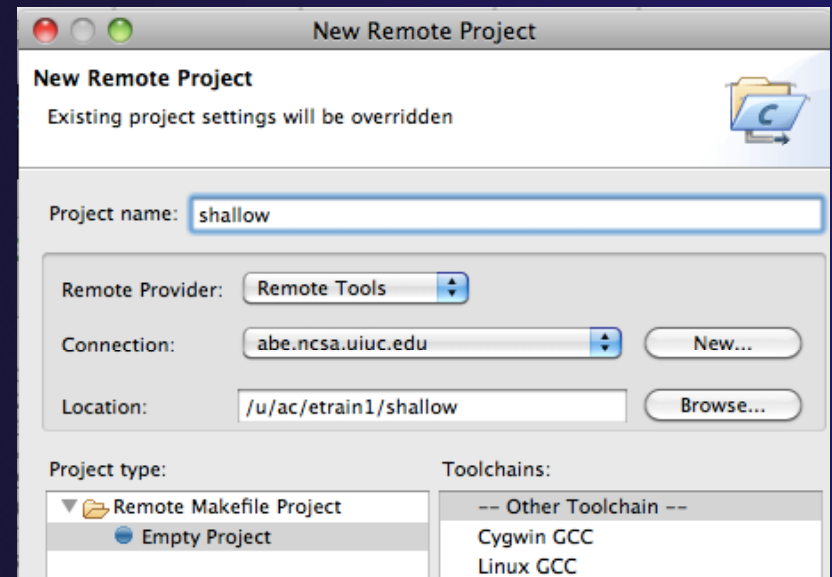
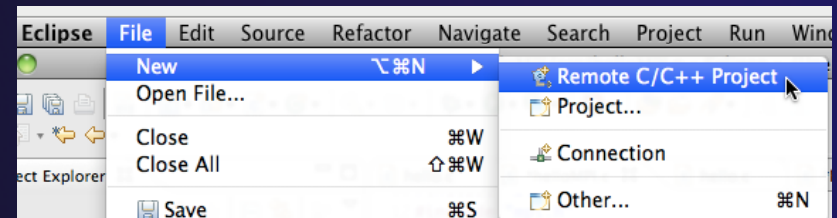
- ★ PTP allows the program to be run locally if you have MPI installed
 - ★ However we want to run the program on a remote machine
- ★ We will now show you how to run a parallel program on a remote machine
 - ★ Interactively
 - ★ Through a batch system
- ★ We have provided the source code to an MPI program on the remote machine
- ★ The project will be created using this source code

Creating a Remote MPI Project



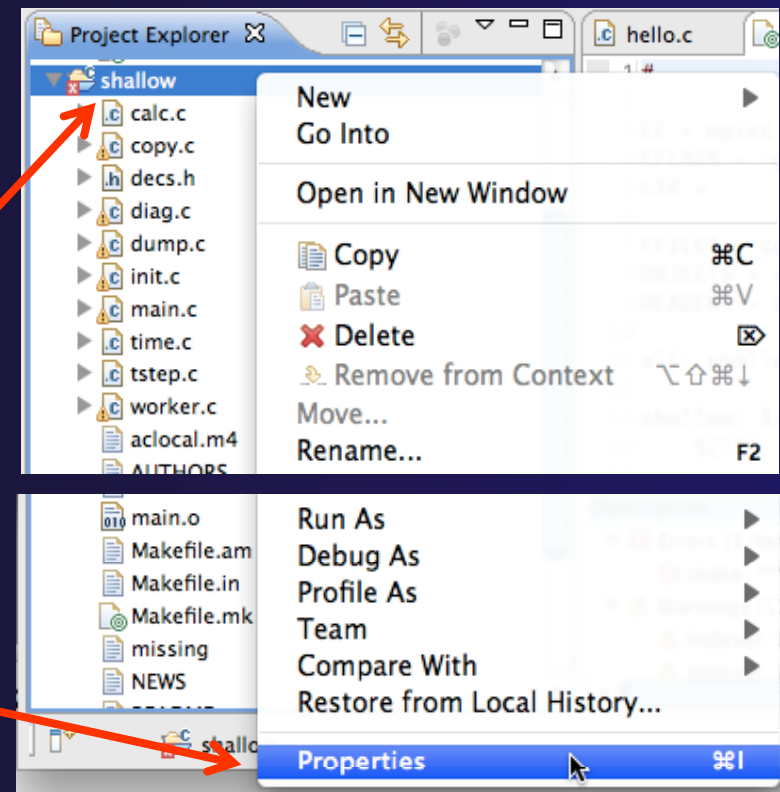
- ★ Like the previous module, create a new Remote C/C++ project
- ★ Enter “shallow” for the **Project Name**
- ★ Use the same **Connection** as before
- ★ Click the **Browse...** button and choose the directory “shallow” in in your home directory
- ★ Select a **Remote Makefile Project** as before
- ★ Click **Finish**

You may be prompted to open the Remote C/C++ Perspective



Changing the Project Build Properties

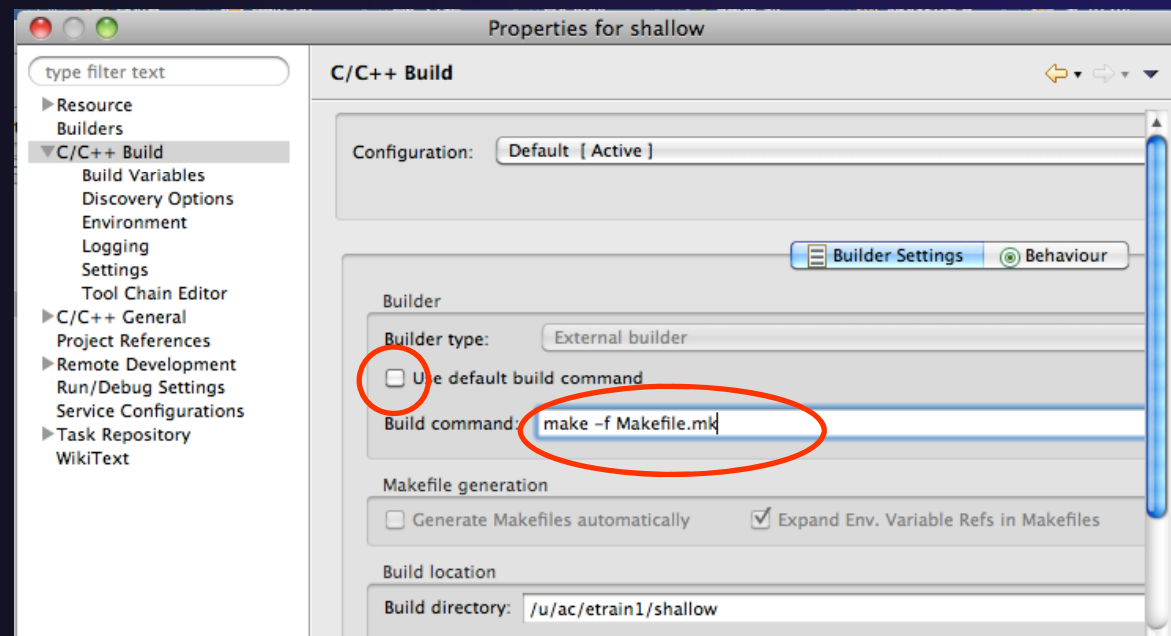
- ✦ The project makefile has a non-standard name Makefile.mk
- ✦ We need to change the build properties so that the project will build
 - ✦ By default, the project is built by running "make"
- ✦ Right-click on project "shallow" in the **Project Explorer**
- ✦ Select **Properties**






Changing the Build Command

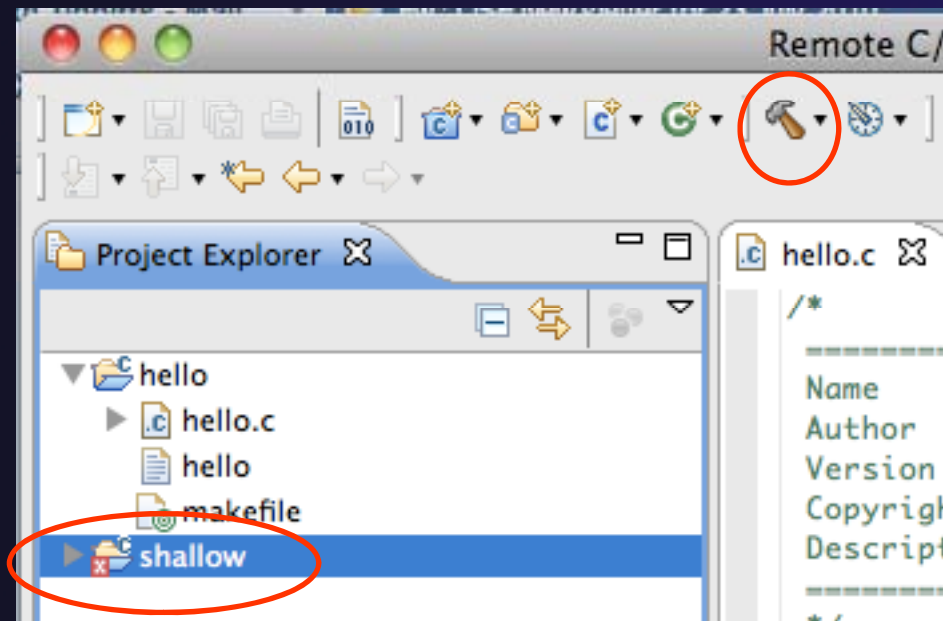
- ✦ Select **C/C++ Build**
- ✦ Uncheck **Use default build command**
- ✦ Change the **Build command** to:
 - ✦ `make -f Makefile.mk`





Building the Project

- ★ Click **OK** to save project properties after changing build command
- ★ Select project and hit the build button 
- ★ The project can be built at any time by hitting this button



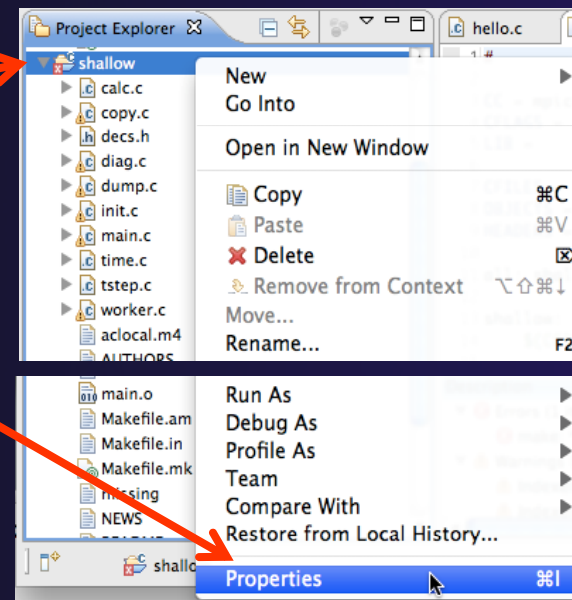


Include File Locations

- ★ Like the previous example, Eclipse content assist and navigation require knowledge of include file locations on the remote system
 - ★ Since the build will be running remotely, the compiler knows how to find include files
 - ★ But Eclipse does not

- ★ In **Project Explorer**, right-click on project

- ★ Select **Properties**



Remote Paths and Symbols



In **Project Properties**,

- ★ Expand **Remote Development**

- ★ Select **Remote Paths and Symbols**

- ★ Select **Languages > GNU C**

 - ★ This is compiler on abe

- ★ Click **Add...**

 - ★ Enter `/usr/local/openmpi-1.4.2-intel-11.1/include`

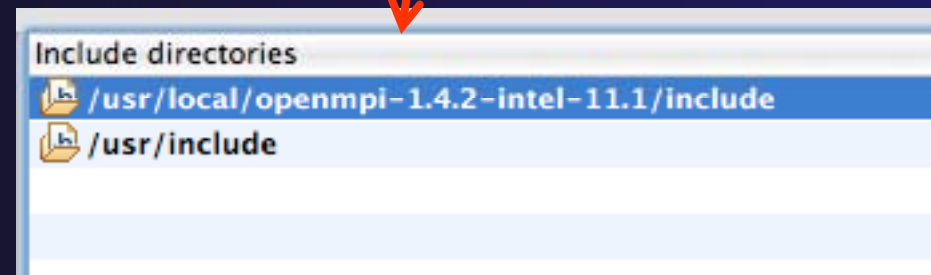
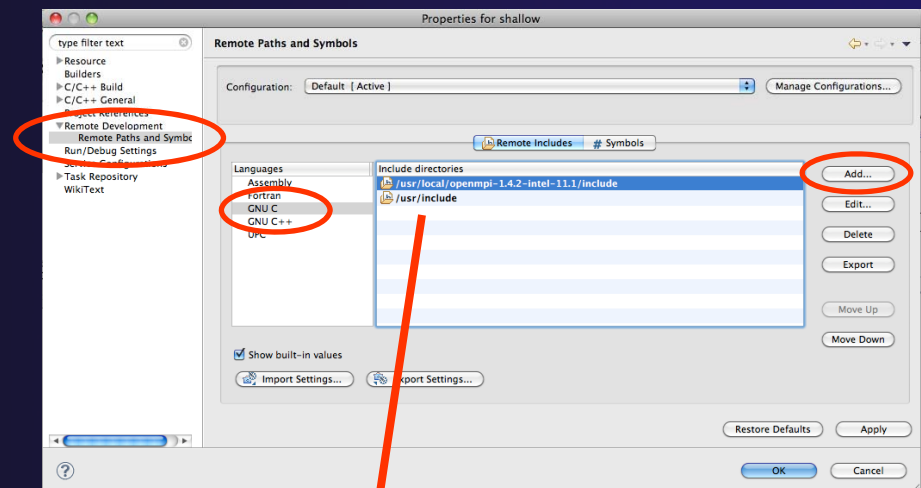
- ★ Click **OK**, then **Add...** again

 - ★ Enter `/usr/include`

- ★ Click **OK**

- ★ Click **OK** to close preferences

- ★ When prompted to rebuild index, click **OK**



MPI-Specific Features

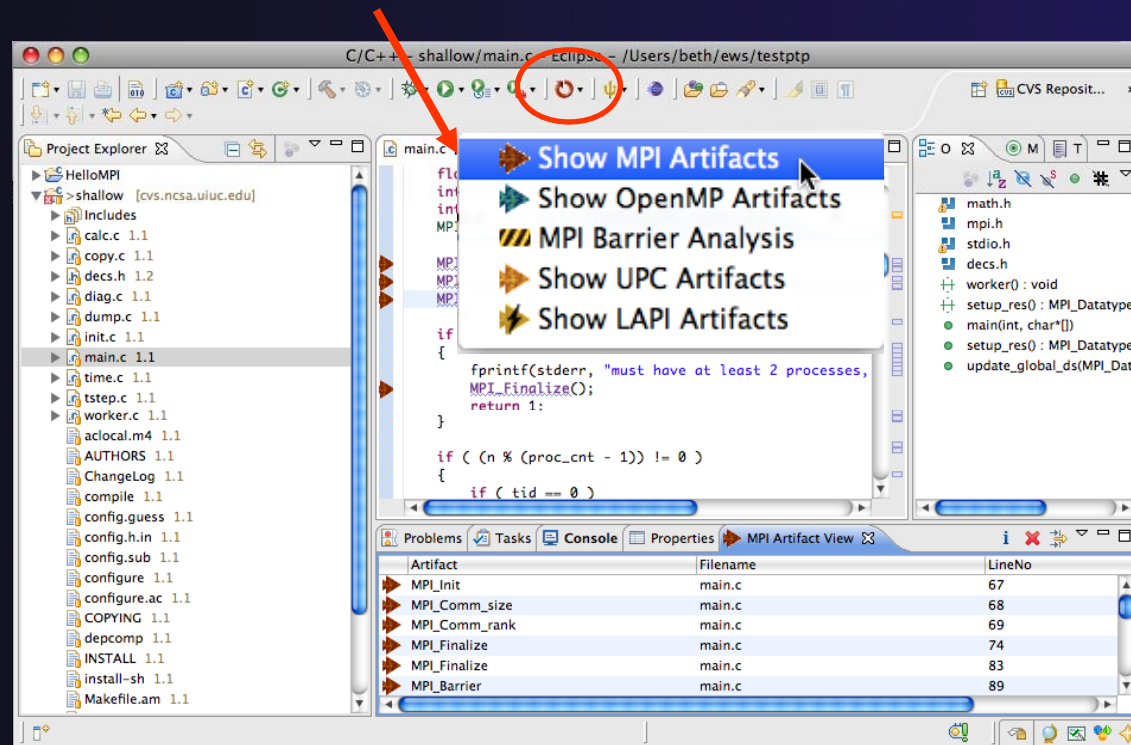
- ★ PTP's Parallel Language Development Tools (PLDT) has several features specifically for developing MPI code
 - ★ Show MPI Artifacts
 - ★ Code completion
 - ★ Context Sensitive Help for MPI
 - ★ Hover Help
 - ★ MPI Templates in the editor

More MPI features covered in
Module 7: Advanced Features

Show MPI Artifacts




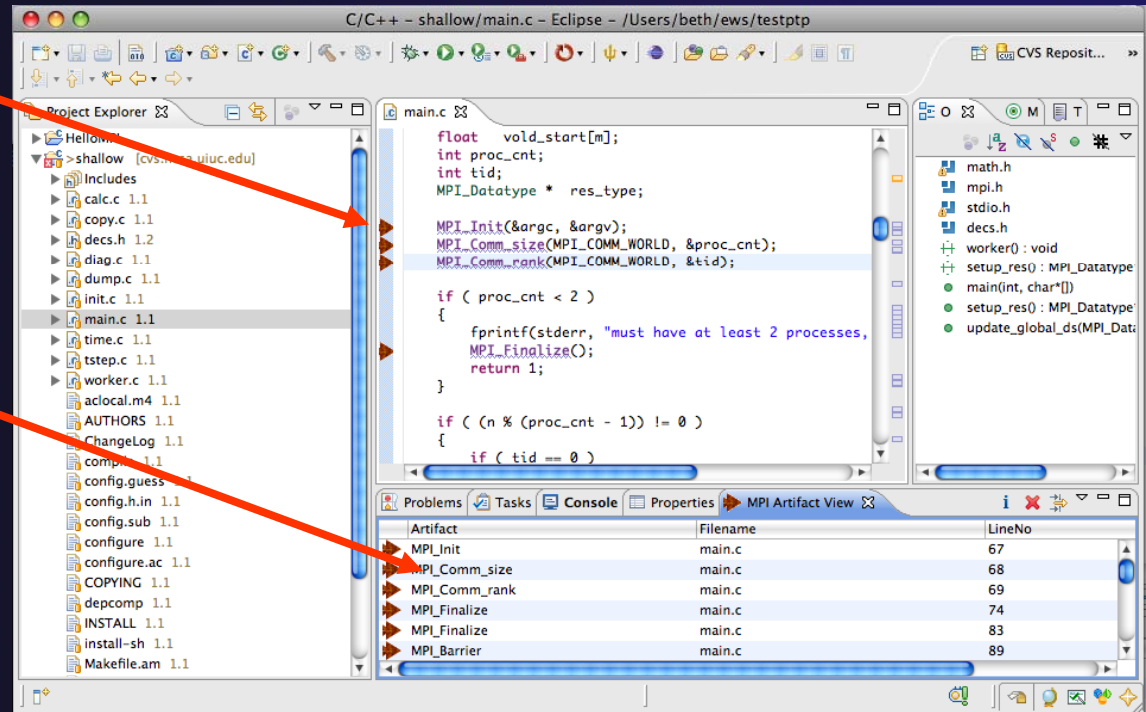
- ✦ In Project Explorer, select a project, folder, or a single source file
 - ✦ The analysis will be run on the selected resources
- ✦ Run the analysis by clicking on drop-down menu next to the analysis button
- ✦ Selecting **Show MPI Artifacts**



MPI Artifact View

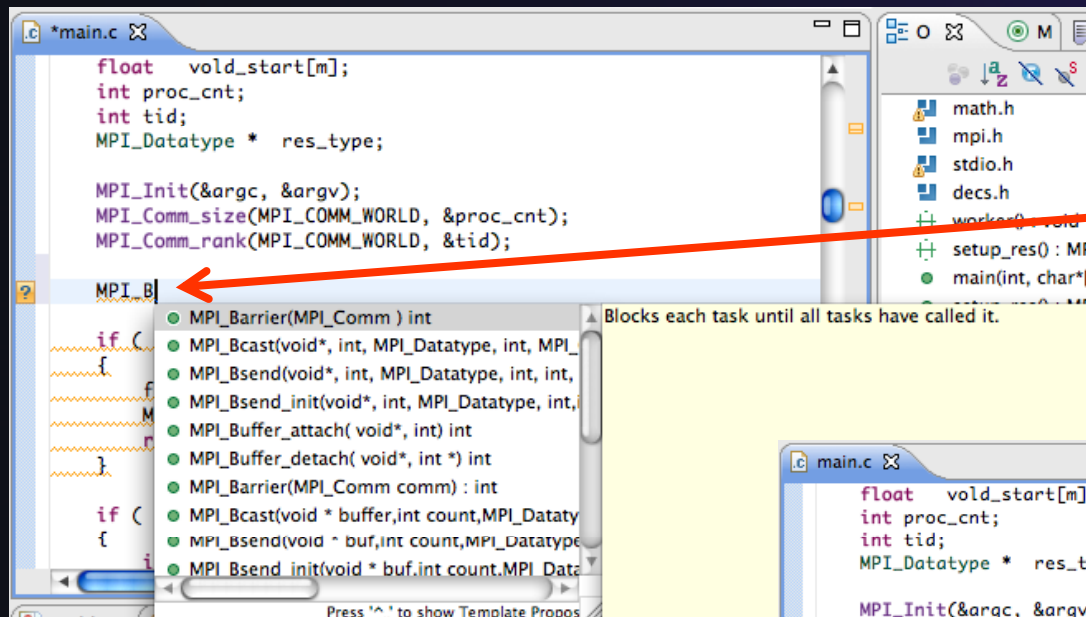


- ✦ Markers indicate the location of artifacts in editor
- ✦ The **MPI Artifact View** list the type and location of each artifact
- ✦ Navigate to source code line by double-clicking on the artifact
- ✦ Run the analysis on another file (or entire project!) and its markers will be added to the view
- ✦ Remove markers via 
- ✦ Click on column headings to sort



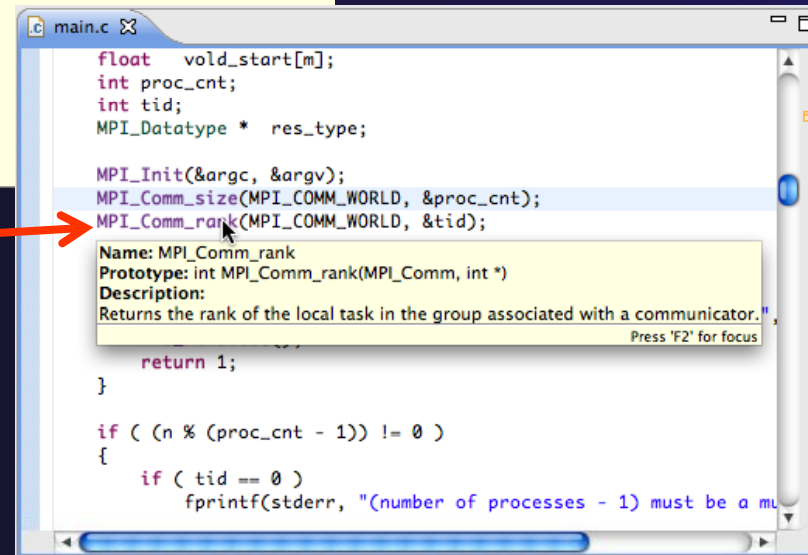


MPI Editor Features



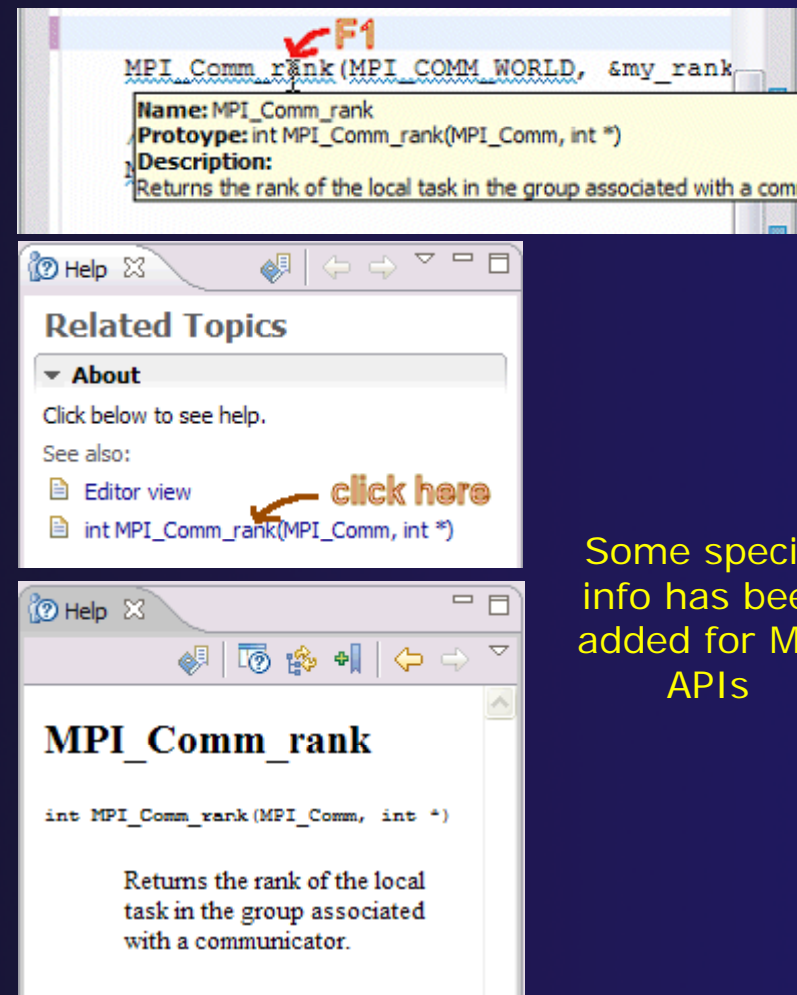
- ★ Code completion will show all the possible MPI keyword completions
- ★ Enter the start of a keyword then press <ctrl-space>

- ★ Hover over MPI API
- ★ Displays the function prototype and a description



Context Sensitive Help

- ★ Click mouse, then press help key when the cursor is within a function name
 - ★ Windows: **F1** key
 - ★ Linux: **ctrl-F1** key
 - ★ MacOS X: **Help** key or **Help►Dynamic Help**
- ★ A help view appears (**Related Topics**) which shows additional information (You may need to click on MPI API in editor again, to populate)
- ★ Click on the function name to see more information
- ★ Move the help view within your Eclipse workbench, if you like, by dragging its title tab



Some special
info has been
added for MPI
APIs

MPI Templates

✦ Allows quick entry of common patterns in MPI programming

✦ Example:

MPI send-receive

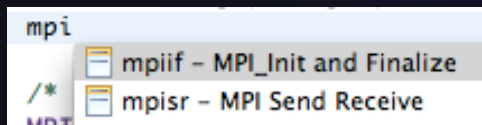
✦ Enter:

mpisr <ctrl-space>

✦ Expands to a send-receive pattern

✦ Highlighted variable names can all be changed at once

✦ Type mpi <ctrl-space> <ctrl-space> to see all templates



```
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
MPI_Comm_size(MPI_COMM_WORLD, &p);
if (rank == 0){ //master task
    printf("Hello From process 0: Num processes: %d\n",p);
    for (source = 1; source < p; source++) {
        MPI_Recv(message, 100, MPI_CHAR, source, tag,
                MPI_COMM_WORLD, &status);
        printf("%s\n",message);
    }
}
else{ // worker tasks
    /* create message */
    sprintf(message, "Hello from process %d!", my_rank);
    dest = 0;
    /* use strlen+1 so that '\0' get transmitted */
    MPI_Send(message, strlen(message)+1, MPI_CHAR,
            dest, tag, MPI_COMM_WORLD);
}
```

Add more templates using Eclipse preferences!

C/C++ > Editor > Templates

Extend to other common patterns

Running the Program

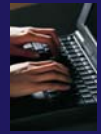
- ✦ Creating a resource manager
- ✦ Starting the resource manager
- ✦ Creating a launch configuration
- ✦ Launching the application
- ✦ Viewing the application run

Terminology

- ★ The **Parallel Runtime** perspective is provided for monitoring and controlling applications
- ★ Some terminology
 - ★ **Resource manager** - Corresponds to an instance of a resource management system (e.g. a job scheduler). You can have multiple resource managers connected to different machines.
 - ★ **Queue** - A queue of pending jobs
 - ★ **Job** – A single run of a parallel application
 - ★ **Machine** - A parallel computer system
 - ★ **Node** - Some form of computational resource
 - ★ **Process** - An execution unit (may be multiple threads of execution)

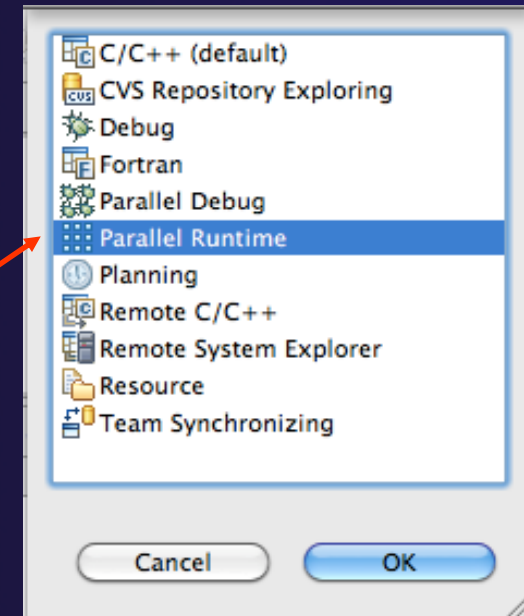
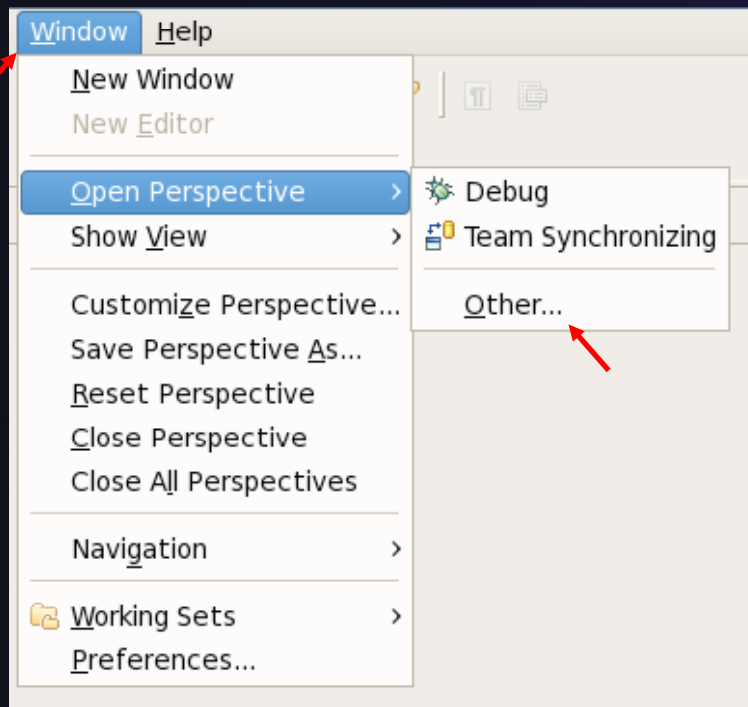
Resource Managers

- ★ PTP uses the term “resource manager” to refer to any subsystem that controls the resources required for launching a parallel job.
- ★ Examples:
 - ★ Job scheduler (e.g. LoadLeveler, PBS, SLURM)
 - ★ Interactive execution (e.g. Open MPI, MPICH2, etc.)
- ★ Each resource manager controls one target system
- ★ Resource Managers can be local or remote



Preparing to Launch

- ✦ Setting up a resource manager is done in the Parallel Runtime perspective
- ✦ Select **Window>Open Perspective>Other**
- ✦ Choose **Parallel Runtime** and click **OK**



Parallel Runtime Perspective

Resource managers view

Machines view

Node details view

Jobs List view

The screenshot shows the Eclipse IDE with the Parallel Runtime Perspective. The top menu bar includes File, Edit, Navigate, Search, Project, Run, Window, and Help. The title bar reads "Parallel Runtime - Eclipse - /Users/beth/ews/tutorial_1021".

On the left side, there are four views:

- Resource Managers**: Shows "Open_MPI@abe.ncsa.uiuc.edu (Open MPI)".
- Machines**: Shows "Open_MPI@abe.ncsa.uiuc.edu: abe.ncsa.uiuc.edu - Root [32]" and a grid of green squares representing nodes.
- Node Attributes**: A table with the following data:

Attribute	Value
Name	honest1.ncsa.uiuc.edu
Node Number	0
Open MPI num	1
- Process Info**: An empty table.
- Jobs List**: A table with the following data:

State	Name

On the right side, there are two views:

- Console**: Shows the output of a "C-Build [shallow]" command: "make: *** No rule to make target 'all'. Stop.".
- Properties**: A table with the following data:

Property	Value
Name	honest1.ncsa.
Node Number	0
Open MPI number of nodes	1

At the bottom, there is a status bar showing "C/C++ Indexer: (0%)".

Console view

Properties view

About PTP Icons

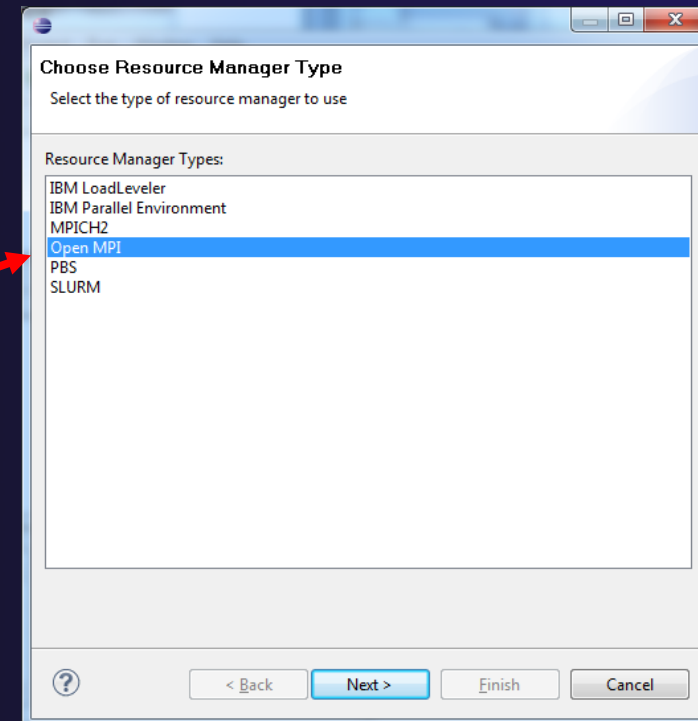
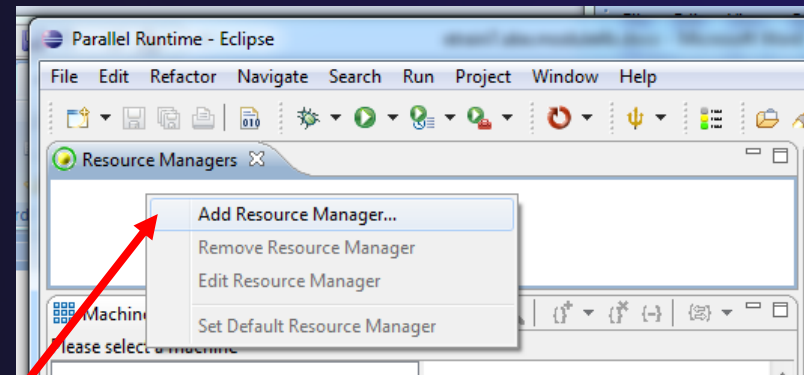


✦ Open using legend icon in toolbar



Running Jobs Interactively

- ★ Interactive resource managers will run the parallel application immediately
- ★ They are also used for debugging the application
- ★ Right-click in Resource Managers view and select **Add Resource Manager**
- ★ Choose the **Open MPI Resource Manager Type**
- ★ Select **Next>**





Configure the Remote Location

Open MPI connection configuration

Enter Open MPI connection information

Remote service provider: Remote Tools

Connection name: abe.ncsa.uiuc.edu New...

Tunneling Options

☐ None

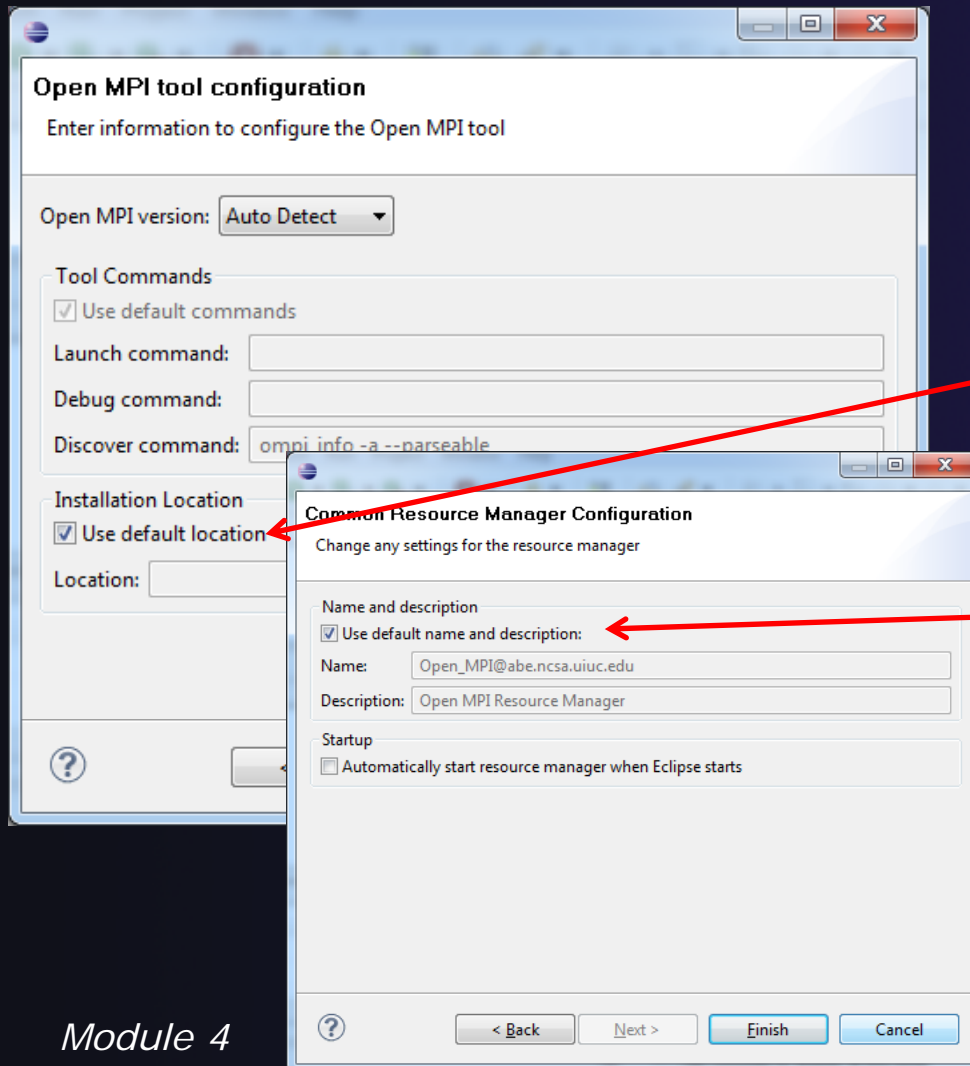
Local address: localhost

☒ SSH port forwarding

? < Back Next > Finish Cancel

- ★ Choose **Remote Tools** for **Remote service provider**
- ★ Choose the remote connection you made previously
- ★ Configure **Tunneling Options** to use **SSH Port Forwarding**
- ★ Click **Next >**

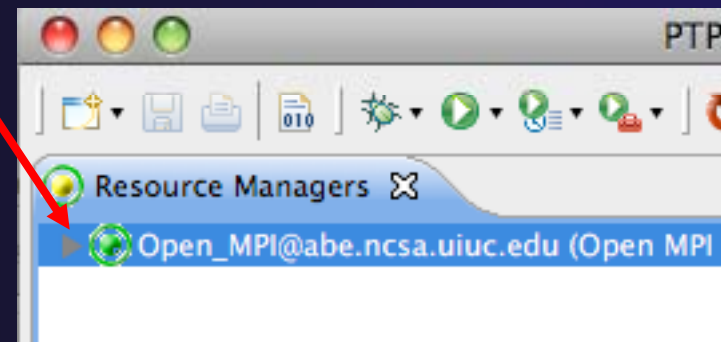
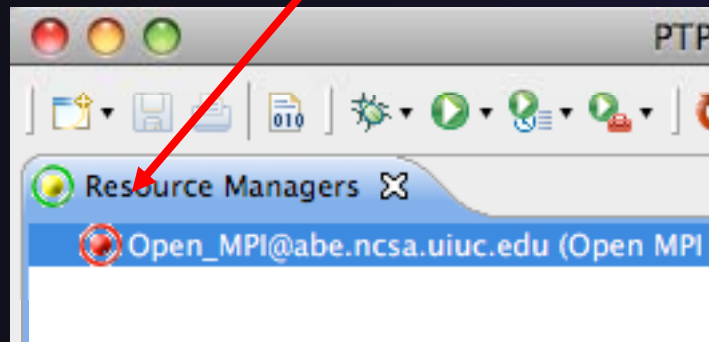
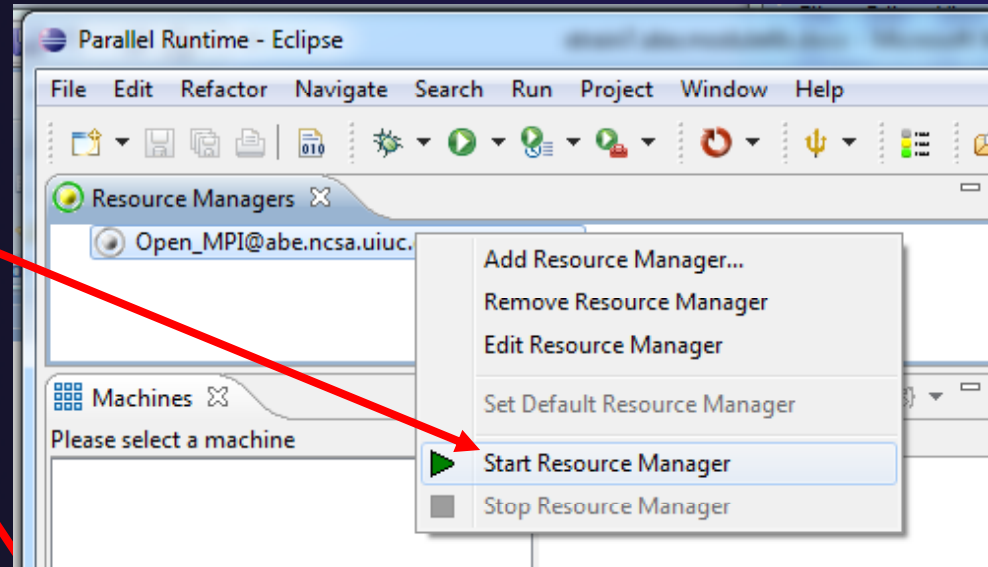
Configure the Resource Manager



- ★ The Open MPI resource manager will auto detect the version and use the appropriate commands
 - ★ Change only if you're an expert
- ★ Set the location of the "mpirun" command if it is not in your path
- ★ Click **Next >**
- ★ Change the **Name** or **Description** of the resource manager if you wish
- ★ You can also set the resource manager to automatically start
- ★ Click **Finish**

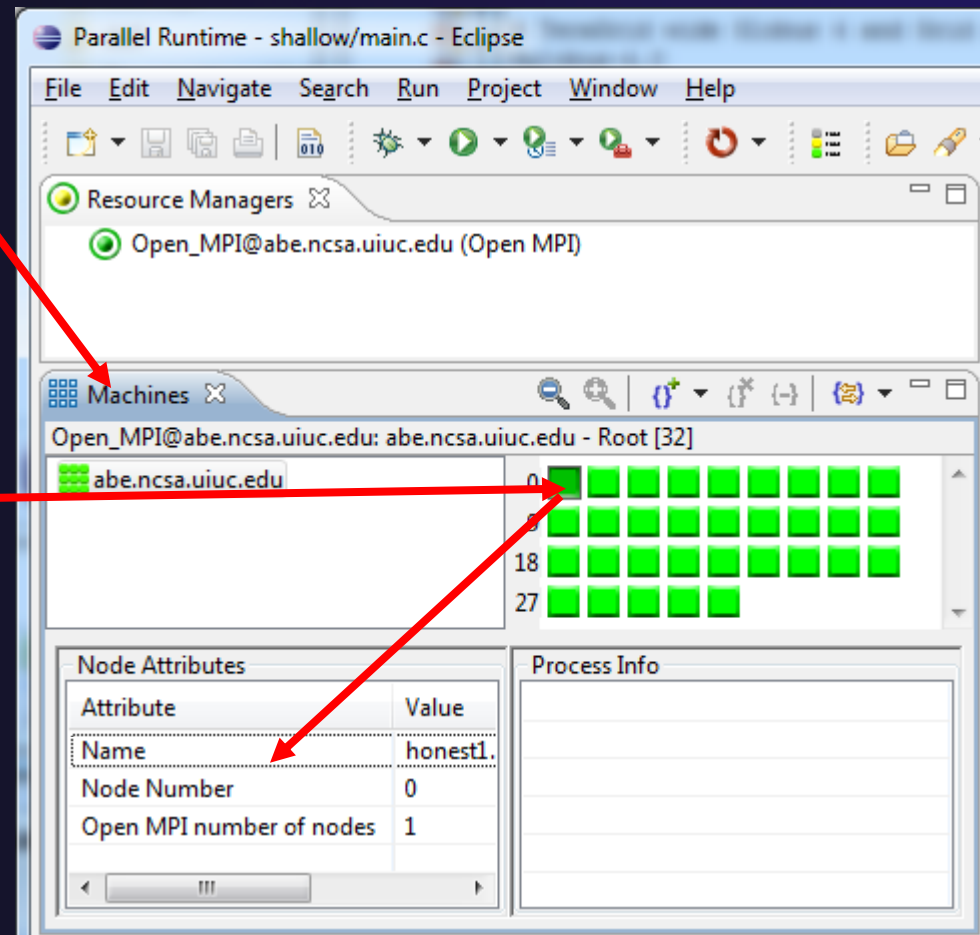
Starting the Resource Manager

- ★ Right click on new resource manager and select **Start resource manager**
- ★ If everything is ok, you should see the resource manager change to **green**
- ★ If something goes wrong, it will change to **red**

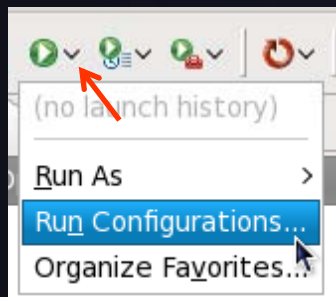


System Monitoring

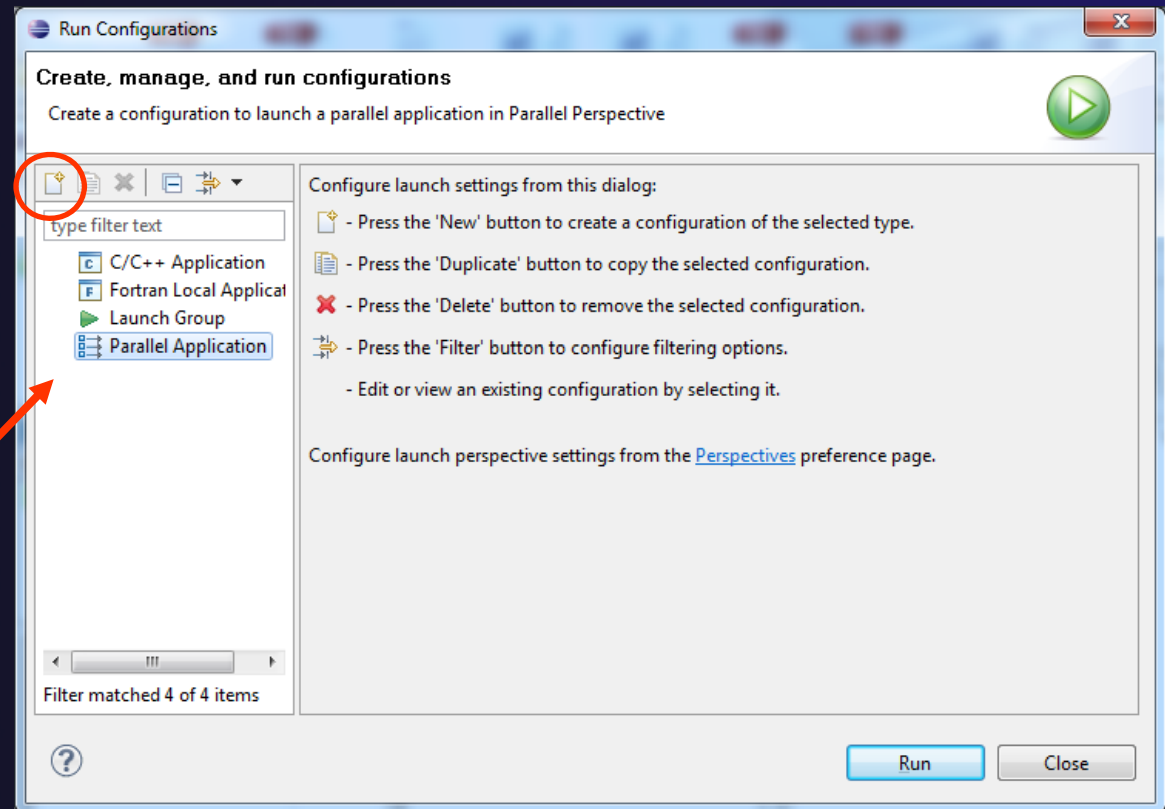
- ✦ Machine status shown in **Machines** view
- ✦ Node status also shown **Machines** view
- ✦ Hover over node to see node name
- ✦ Double-click on node to show attributes



Create a Launch Configuration



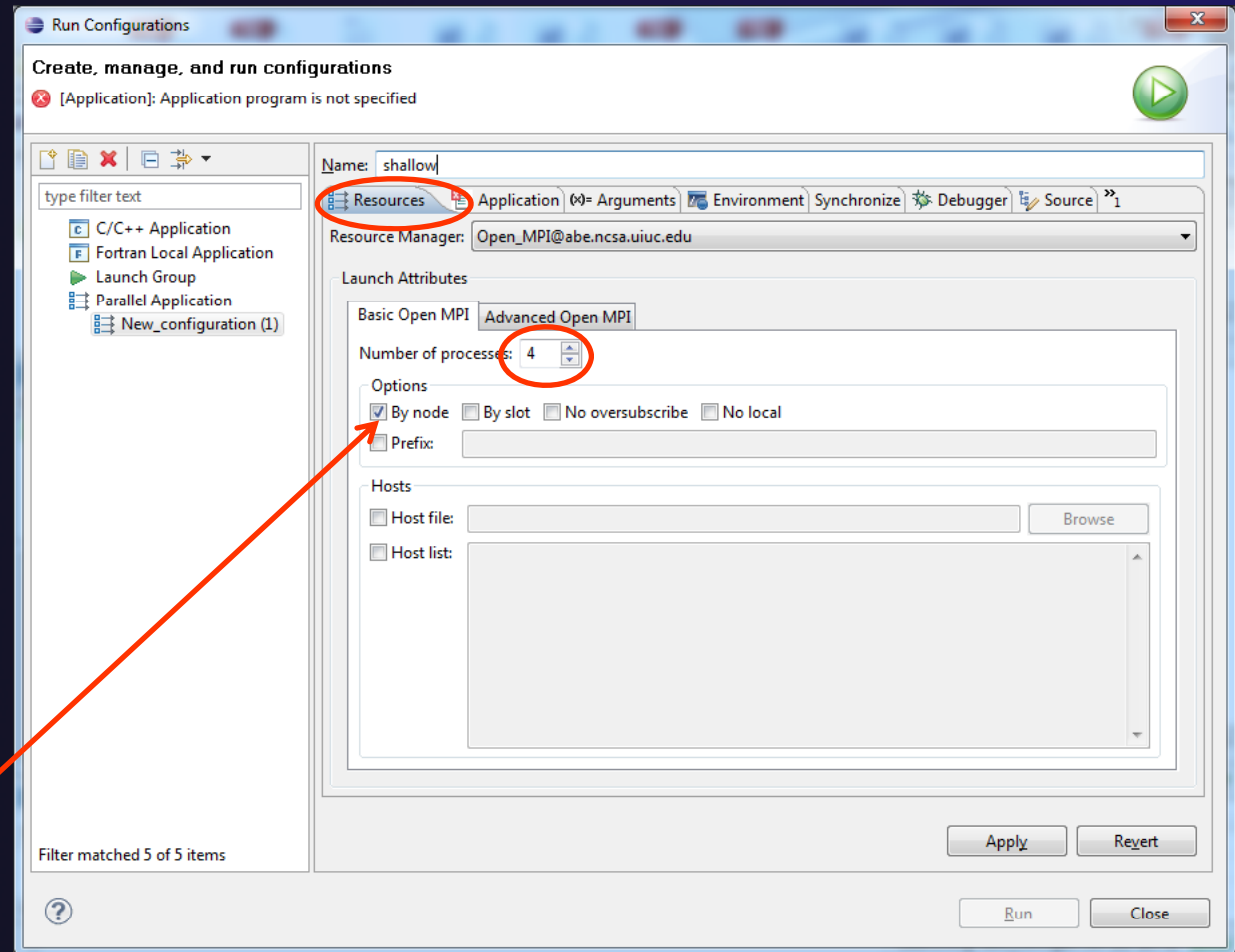
- ✦ Open the run configuration dialog **Run>Run Configurations...**
- ✦ Select **Parallel Application**
- ✦ Select the **New** button



Depending on which flavor of Eclipse you installed, you might have more choices in Application types

Complete the Resources Tab

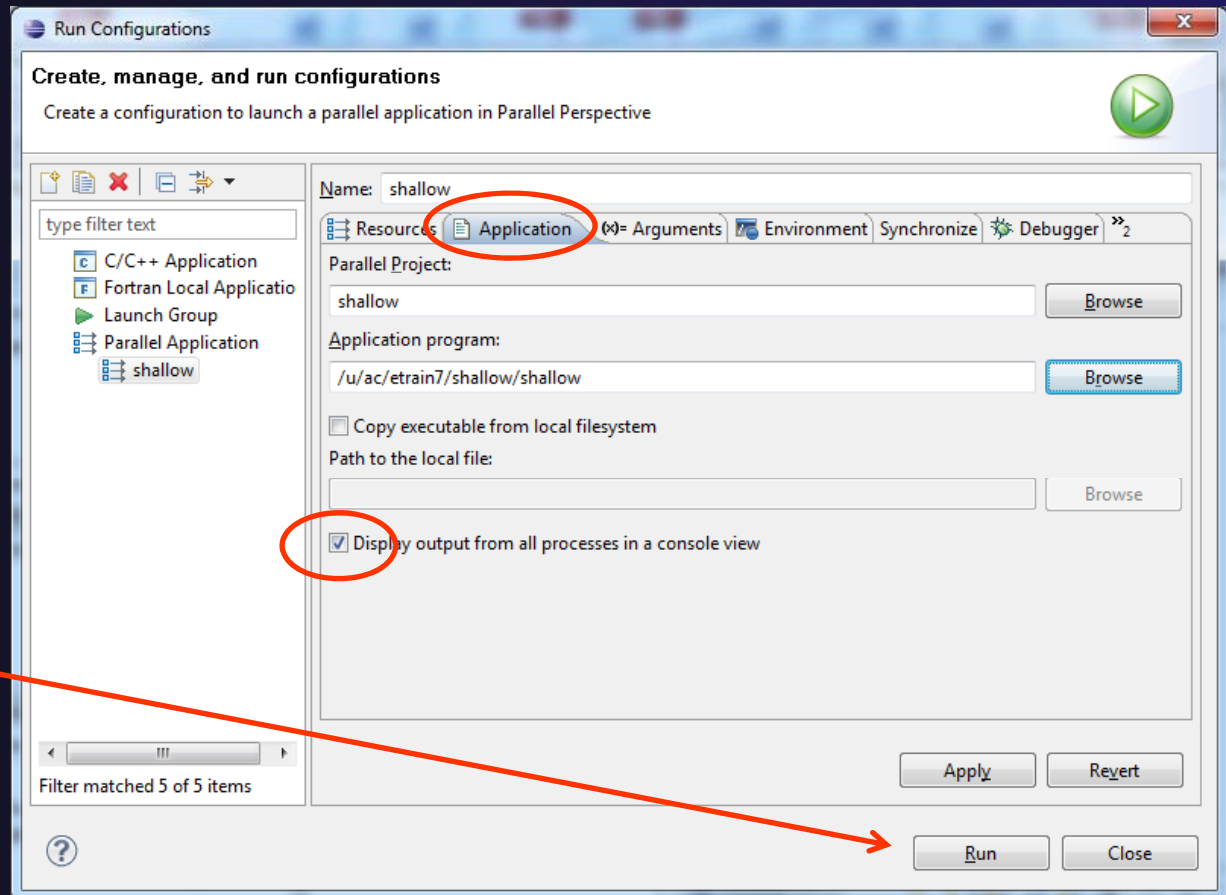
- ★ Enter a name for the launch configuration, e.g. "shallow"
- ★ In **Resources** tab, select the resource manager you want to use to launch this job
- ★ Enter a value in the **Number of processes** field
- ★ Other fields can be used to specify resource manager-specific information
 - ★ E.g. specify **By node** to allocate each process to a different node





Complete the Application Tab

- ★ Select the **Application** tab
- ★ Choose the **Application program** by clicking the **Browse** button and locating the executable on the remote machine
 - ★ There should be a "shallow" executable in the "shallow" directory
- ★ Select **Display output from all processes in a console view**
- ★ Click **Run** to run the application





Viewing The Run

- ★ Double-click a node in machines view to see which processes ran on the node
- ★ Hover over a process for tooltip popup
- ★ Job status and information

The screenshot shows the Eclipse IDE with the Parallel Runtime plugin. The main window is titled "Parallel Runtime - shallow/main.c - Eclipse". The interface is divided into several panes:

- Resource Managers:** Shows "Open_MPI@abe.ncsa.uiuc.edu (Open MPI)".
- Machines:** Shows a list of machines. The first machine is "Open_MPI@abe.ncsa.uiuc.edu: abe.ncsa.uiuc.edu - Root [32]". Below it, a grid of green squares represents processes. A red arrow points to one of these squares.
- Node Attributes:** A table showing attributes for the selected node:

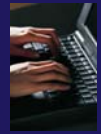
Attribute	Value
Name	honest1.
Node Number	28
Open MPI number of nodes	1
- Process Info:** A table showing process information. A red arrow points to the first entry:

Process
job0:job0.0
- Jobs List:** A table showing the status of jobs. A red arrow points to the first job:

S...	Name	Status	Executable Name	Executable Path	User	An
⊙	job0	NORMAL	shallow	/u/ac/etrain7/shal...		[]
- Console:** Shows the output of the job. It displays cycle numbers (850, 900, 950, 1000) and various energy values (Potential energy, Kinetic Energy, Total Energy, Pot. Enstrophy).
- Properties:** A table showing properties for the selected job:

Property	Value
Name	honest1.ncsa.uiuc.edu
Node Number	28
Open MPI number of nc	1

Viewing Program Output



- ★ Console displays combined output from all processes

- ★ Properties view shows job details

The screenshot shows the Eclipse IDE with the Parallel Runtime plugin. The main window is titled "Parallel Runtime - shallow/main.c - Eclipse". The interface is divided into several panes:

- Resource Managers:** Shows "Open_MPI@abe.ncsa.uiuc.edu (Open MPI)".
- Machines:** Shows a list of machines with a grid of green squares representing processes. The "Node Attributes" table is visible:

Attribute	Value
Name	honest1.
Node Number	28
Open MPI number of nodes	1
- Jobs List:** Shows a table with job details:

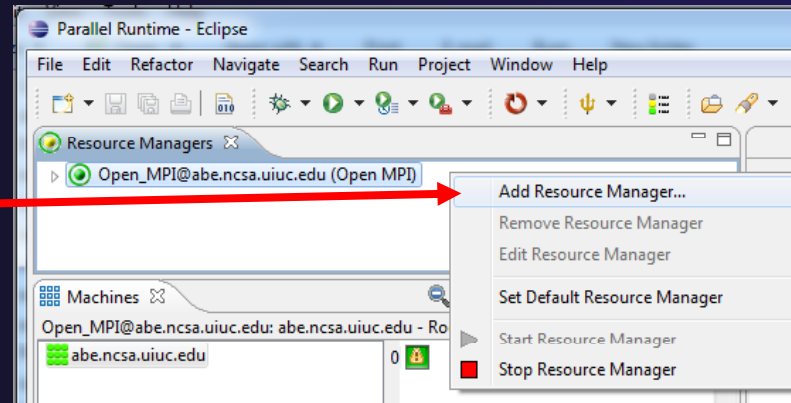
S...	Name	Status	Executable Name	Executable Path	User	An
●	job0	NORMAL	shallow	/u/ac/etrain7/shal...		
- Console:** Displays the combined output from all processes. It shows cycle numbers (850, 900, 950, 1000) and various energy values (Potential, Kinetic, Total). A red arrow points from the "Console displays combined output from all processes" annotation to this pane.
- Properties:** Shows details for the selected job (job0). The "Property" table is visible:

Property	Value
Name	honest1.ncsa.uiuc.edu
Node Number	28
Open MPI number of nc	1

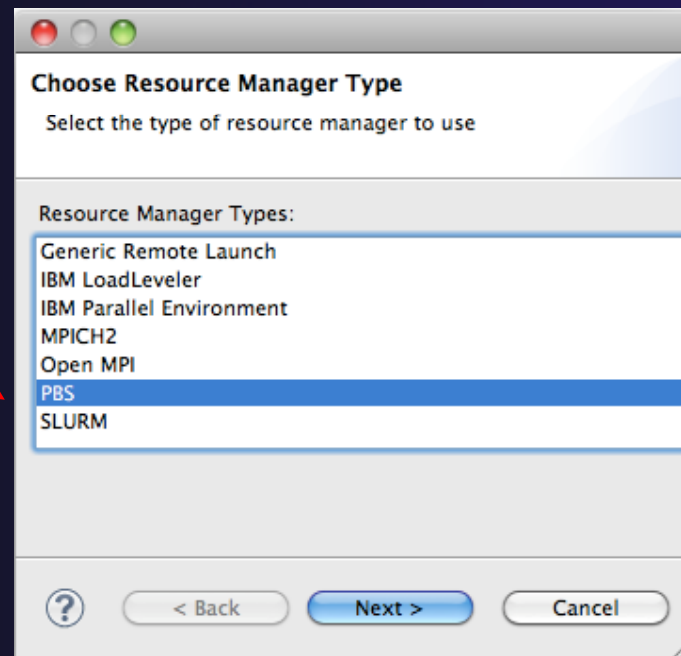
 A red arrow points from the "Properties view shows job details" annotation to this pane.

Using a Job Scheduler

- ★ Right-click in Resource Managers view and select **Add Resource Manager**



- ★ Choose the **PBS Resource Manager Type**



- ★ Select **Next>**



Configure the Remote Location

PBS Proxy Configuration
Enter information to connect to an PBS proxy server

Remote service provider: Remote Tools

Connection name: abe.ncsa.uiuc.edu New...

Proxy Options...

Tunneling Options

☐ None

Local address for proxy connection: localhost

☒ SSH port forwarding

☐ Launch server manually

? < Back Next > Finish Cancel

- ★ Choose **Remote Tools** for **Remote service provider**
- ★ Choose the remote connection you made previously
- ★ Configure **Tunneling Options** to use **SSH Port Forwarding**
- ★ Click **Next >**



Configure the Resource Manager

PBS Batch Script Configuration
Enter information to configure PBS Batch Script Templates

Default Template: default_template

Edit Template Delete Template

Attribute Placeholders

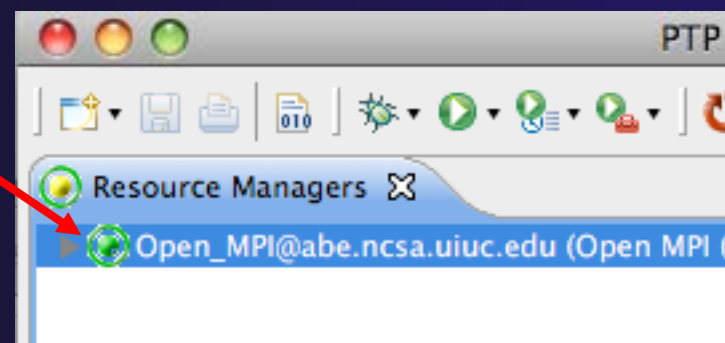
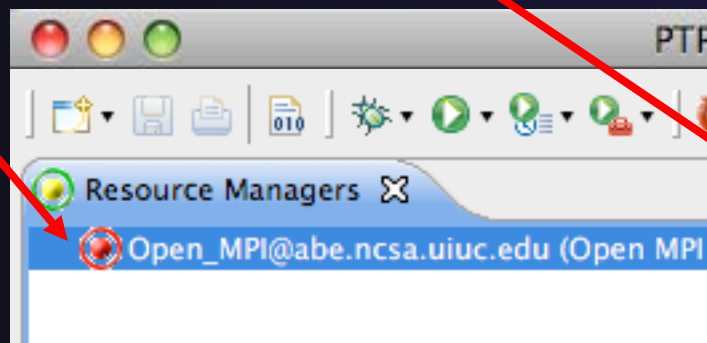
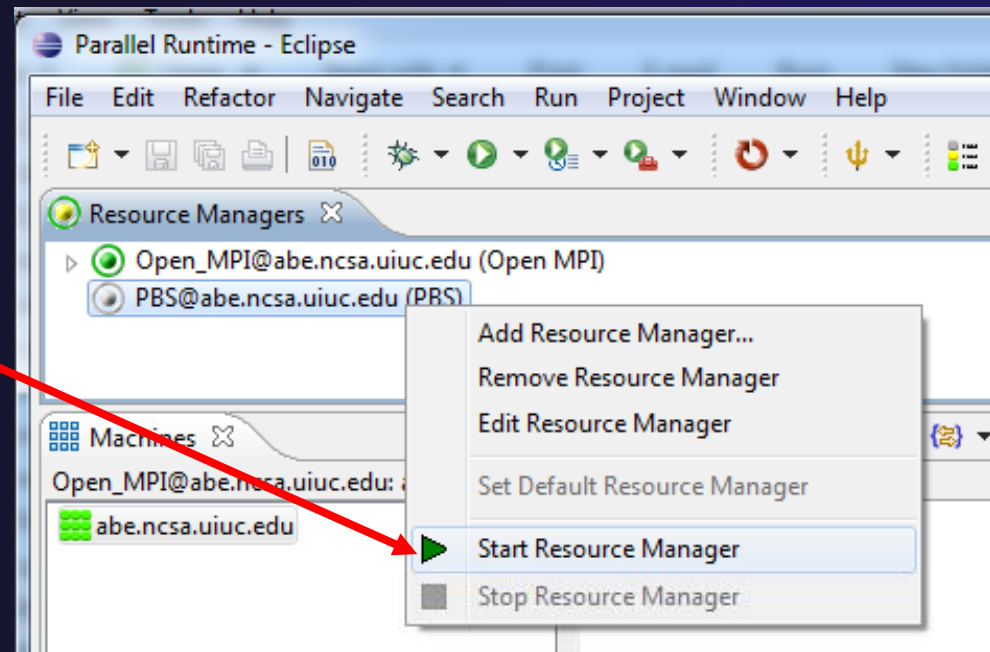
Name	Default Value	Tool Tip
Account_Name		Format: string
Error_Path		Format: "[hostname]:p
Job_Name		Format: string up to 15
Output_Path		Format: "[hostname]:p
Resource_List.nodes		The value is one or mc
Resource_List.walltime	00:30:00	Format: [[hours:]minu
destination		Format: queue[@serve

? < Back Next > Finish Cancel

- ★ The PBS resource manager allows customization to match the local site options for the PBS installation
- ★ By default, all known PBS options will be displayed
- ★ Templates can be used to customize the options for each installation
- ★ We will not change this, just click **Finish** to complete the configuration

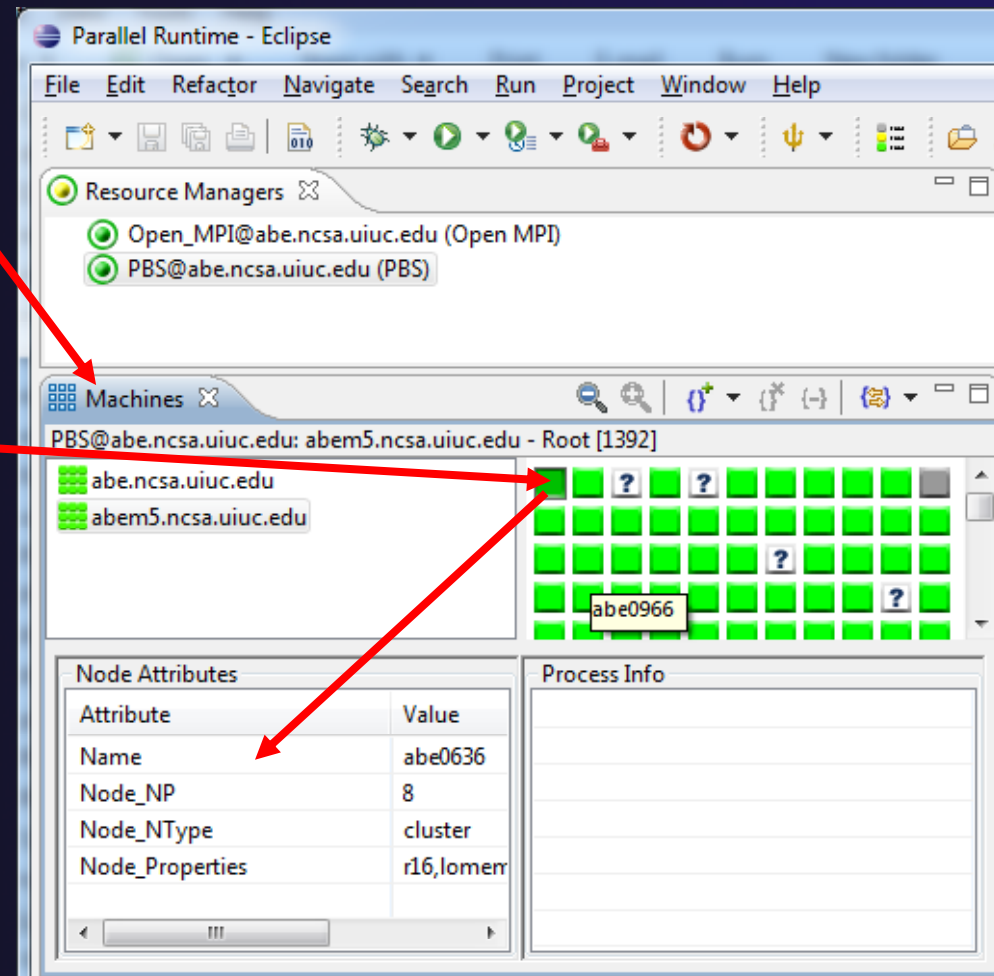
Starting the Resource Manager

- ★ Right click on new resource manager and select **Start resource manager**
- ★ If everything is ok, you should see the resource manager change to **green**
- ★ If something goes wrong, it will change to **red**



System Monitoring

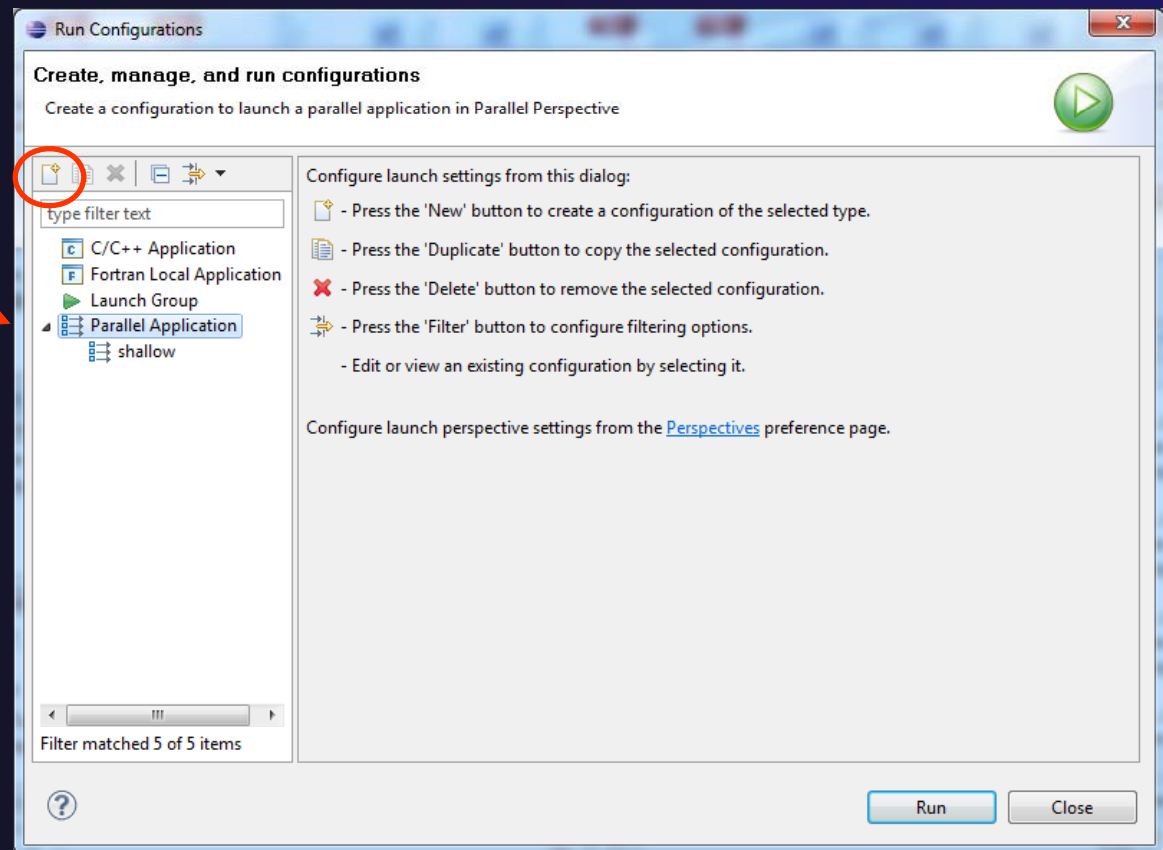
- ✦ Machine status shown in **Machines** view
- ✦ Node status also shown **Machines** view
- ✦ Hover over node to see node name
- ✦ Double-click on node to show attributes





Create a Launch Configuration

- ✦ Open the run configuration dialog **Run>Run Configurations...**
- ✦ Select **Parallel Application**
- ✦ Select the **New** button



Complete the Resources Tab

- ✦ Enter a name for this launch configuration, e.g. "shallow (PBS)"
- ✦ In **Resources** tab, select the PBS resource manager you just created
- ✦ The **MPI Command** field allows this job to be run as an MPI job
 - ✦ Choose **mpirun**
- ✦ Enter account name "dvd"
- ✦ Enter the number of nodes to reserve in the **Resource_List.nodes** field
 - ✦ Use 4 nodes
- ✦ Select the destination queue -- **nomss**

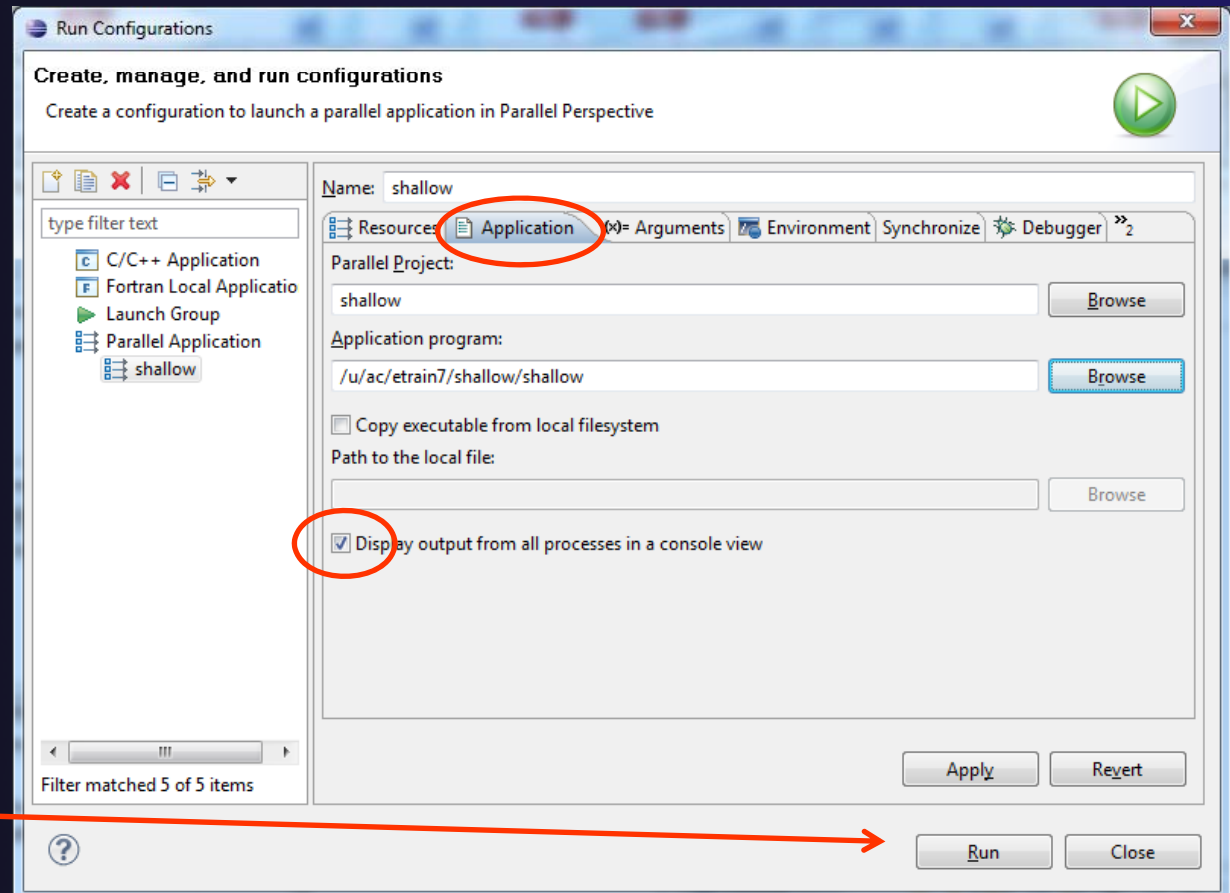
The screenshot shows the 'Run Configurations' dialog box with the 'Resources' tab selected. The configuration is for a launch configuration named 'shallow (PBS)' using the 'PBS@abe.ncsa.uiuc.edu' resource manager. The 'MPI Command' is set to 'mpirun'. The 'Resource_List.nodes' is set to 4, and the 'destination' is set to 'nomss'. The 'Account_Name' is set to 'dvd'.

Attribute	Value	Description
Account_Name	dvd	Account to which to charge this job.
Error_Path		The final path name for the file containing the job's standard error stream.
Job_Name		The name assigned to the job by the qsub or qalter command.
Output_Path		The final path name for the file containing the job's standard output.
Resource_List.nodes	4	Number and/or type of nodes to be reserved for exclusive use by the job.
Resource_List.walltime	00:30:00	Maximum amount of real time during which the job can be in the run queue.
destination	nomss	Designation of the queue to which to submit the job.



Complete the Application Tab

- ★ Select the **Application** tab
- ★ Choose the **Application program** by clicking the **Browse** button and locating the executable on the remote machine
 - ★ Use the same "shallow" executable
- ★ Select **Display output from all processes in a console view**
- ★ If Debugger tab has error, select Debugger: **SDM**
- ★ Click **Run** to submit the application to the job scheduler



Job Monitoring

- ★ Job status is tracked here, successful jobs disappear from list
- ★ To cancel, select job and select Red button in Jobs List

