



parallel tools platform

<http://eclipse.org/ptp>

Using the Eclipse Parallel Tools Platform in Support of Earth Sciences High Performance Computing

Jay Alameda

National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign

Software Engineering Assembly Conference 2014 - 8 April 2014

Acknowledgements

- ◆ Portions of this material are supported by or based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under its Agreement No. HR0011-07-9-0002, the United States Department of Energy under Contract No. DE-FG02-06ER25752, the Blue Waters sustained petascale computing project, which is supported by the National Science Foundation under award number OCI 07-25070, and the SI2-SSI Productive and Accessible Development Workbench for HPC Applications, which is supported by the National Science Foundation under award number OCI 1047956
- ◆ The SI2-SSI team is lead by Jay Alameda (NCSA), Greg Watson (IBM), Steven Brandt (LSU), and Allen Malony (U Oregon). Team members and senior personnel include Beth Tibbitts (NCSA), Ralph Johnson (U Illinois), Chris Navarro (NCSA), Sameer Shende (U Oregon), Wyatt Spear (U Oregon), Brian Jewett (U Illinois), Galen Arnold (NCSA), and Rui Liu (NCSA)

Outline

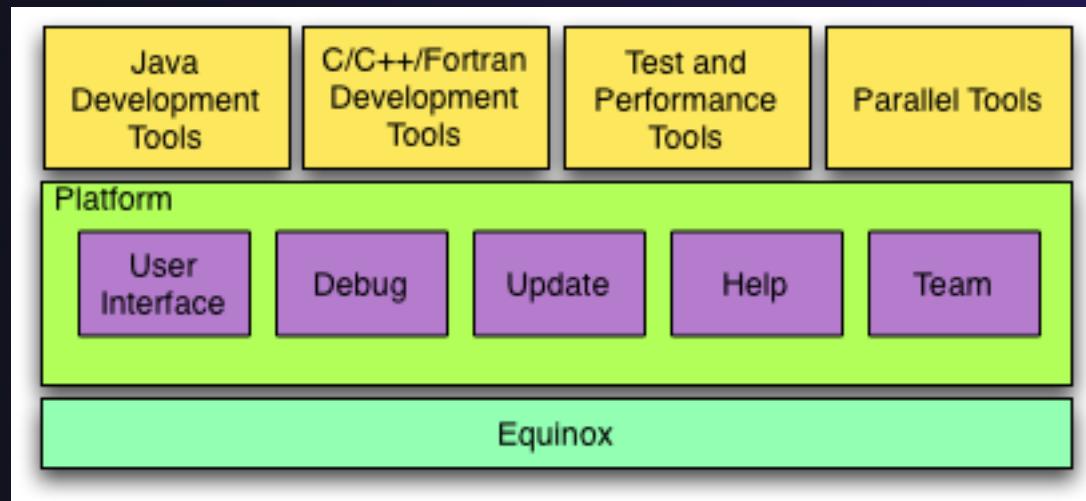
- ◆ Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- ◆ Motivation for Workbench for High Performance Computing (WHPC)
 - ◆ Improvements to Eclipse PTP
- ◆ Software Engineering Practices Enabled by Eclipse PTP
 - ◆ Code visibility
 - ◆ Multi-system build management
 - ◆ Performance tuning
 - ◆ Source code control
 - ◆ Issue Tracking
 - ◆ Documentation
 - ◆ Earth Science/Weather code example
- ◆ Eclipse PTP Resources

Outline

- ★ Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- ★ Motivation for Workbench for High Performance Computing (WHPC)
 - ★ Improvements to Eclipse PTP
- ★ Software Engineering Practices Enabled by Eclipse PTP
 - ★ Code visibility
 - ★ Multi-system build management
 - ★ Performance tuning
 - ★ Source code control
 - ★ Issue Tracking
 - ★ Documentation
 - ★ Earth Science/Weather code example
- ★ Eclipse PTP Resources

What is Eclipse?

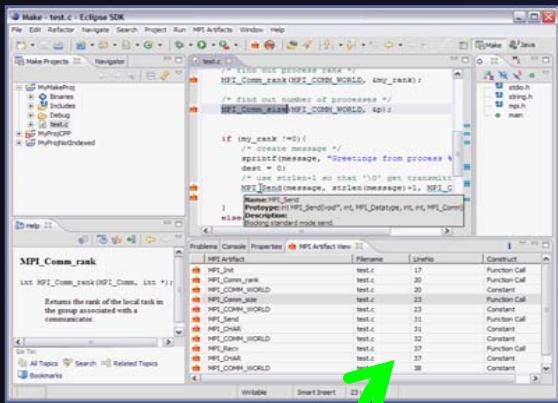
- ★ A vendor-neutral open-source workbench for multi-language development
- ★ A extensible platform for tool integration
- ★ Plug-in based framework to create, integrate and utilize software tools



parallel tools platform

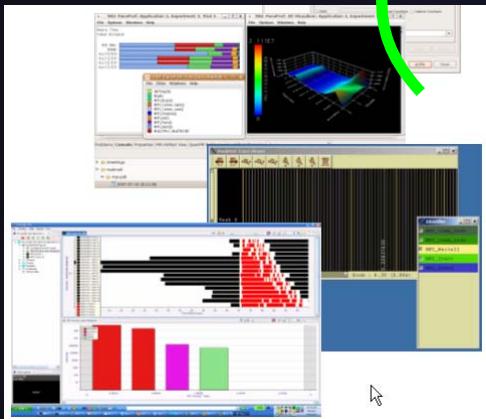
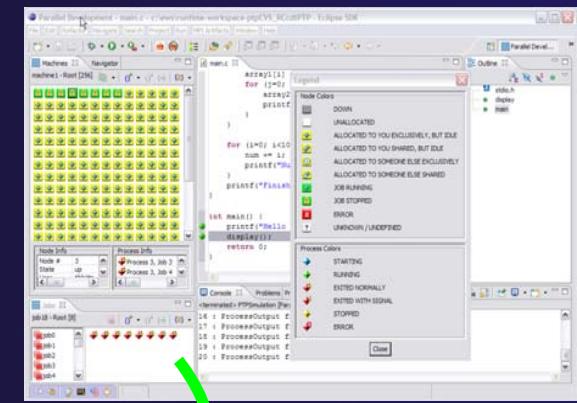
Eclipse Parallel Tools Platform (PTP)

Coding & Analysis



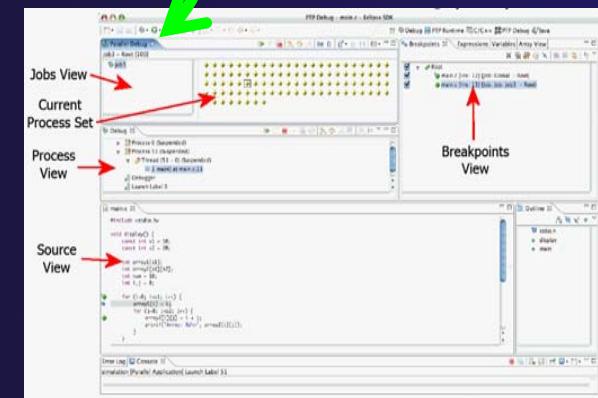
Code development lifecycle

Launching & Monitoring



Performance Tuning

eclipse



Debugging

Outline

- ★ Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- ★ Motivation for Workbench for High Performance Computing (WHPC)
 - ★ Improvements to Eclipse PTP
- ★ Software Engineering Practices Enabled by Eclipse PTP
 - ★ Code visibility
 - ★ Multi-system build management
 - ★ Performance tuning
 - ★ Source code control
 - ★ Issue Tracking
 - ★ Documentation
 - ★ Earth Science/Weather code example
- ★ Eclipse PTP Resources

Motivation for Workbench for High Performance Computing (WHPC)

- ◆ Stable, portable platform for tool development
 - ◆ Focus on tool functionality, manage rapid evolution of HPC platforms
 - ◆ Encourage consistent tool look and feel
 - ◆ Support for HPC application development practices
- ◆ Why Parallel Tools Platform?
 - ◆ High potential to meet needs of a WHPC.
 - ◆ Target next generation of HPC developers growing up with IDEs (Eclipse, Visual Studio, ...)
 - ◆ Need to cultivate community of users!

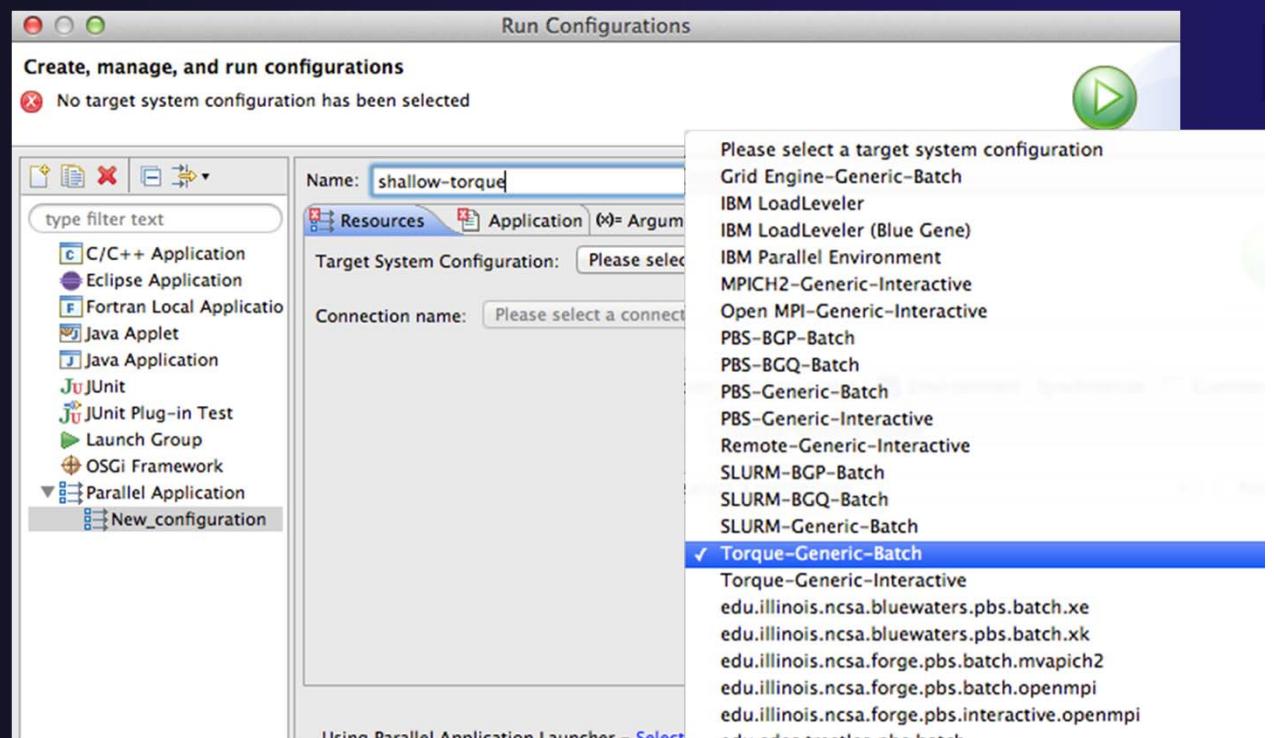
Improvements

- ★ Work within Eclipse release cycle
 - ★ Major (API-breaking) improvements with coordinated June release
 - ★ Last major release Eclipse 4.3 “Kepler” released June 26, 2013
 - ★ Minor enhancements and bug-fixes with two coordinated service releases in September and February
 - ★ Eclipse 4.3 SR2 Released February 28, 2014
- ★ Foci of improvements
 - ★ Improve usability
 - ★ Improve productivity

Significant Recent Improvements

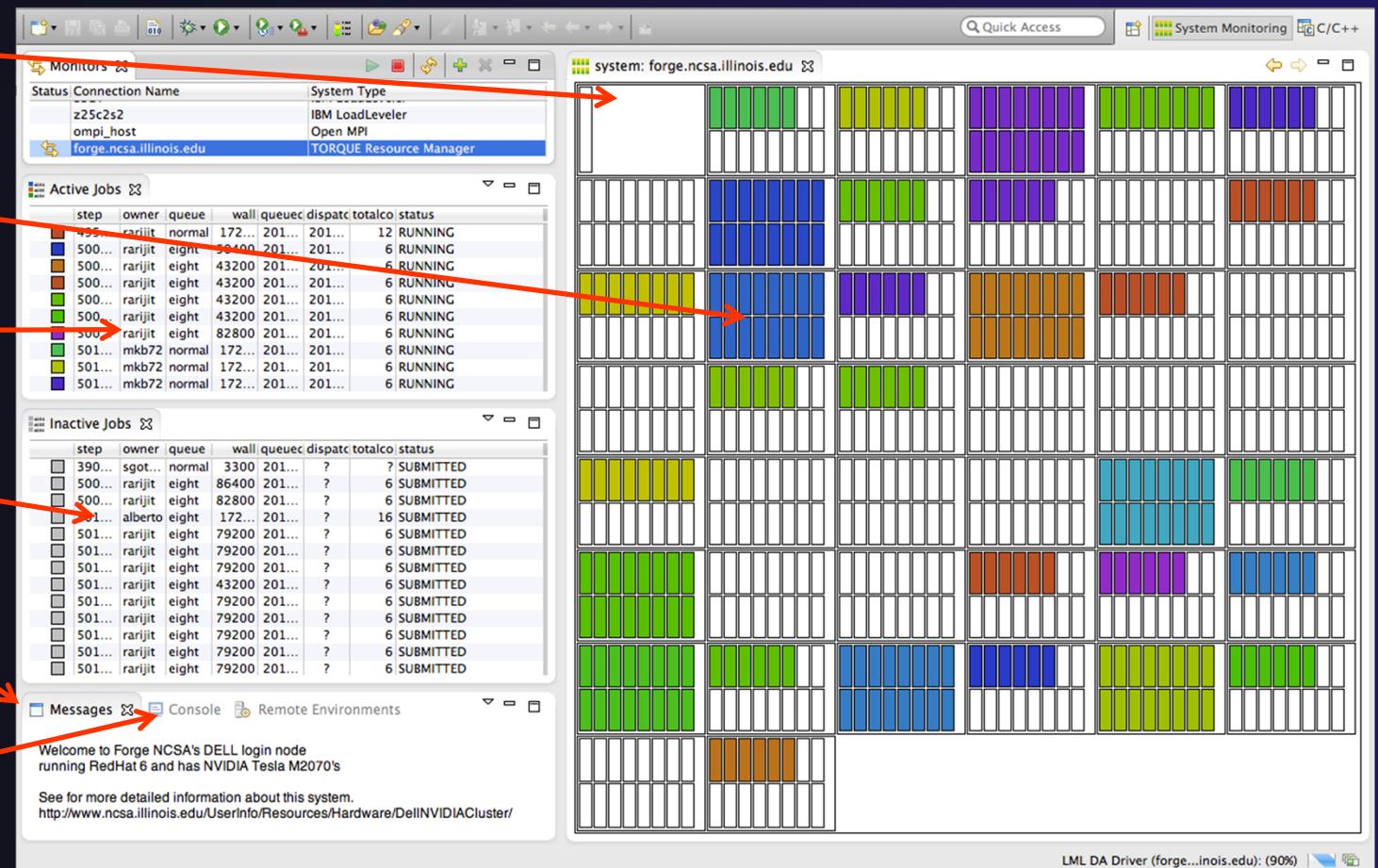
- ★ User-configurable machine configuration
 - ★ Wide variety of configurations now available:
 - ★ Documentation, tutorial at
 - ★ http://wiki.eclipse.org/PTP/designs/Resource_Manager_Configuration

Both generic
and specific



Scalable System Monitoring

- ★ System view
- ★ Jobs running on system
- ★ Active jobs
- ★ Inactive jobs
- ★ Messages
- ★ Console

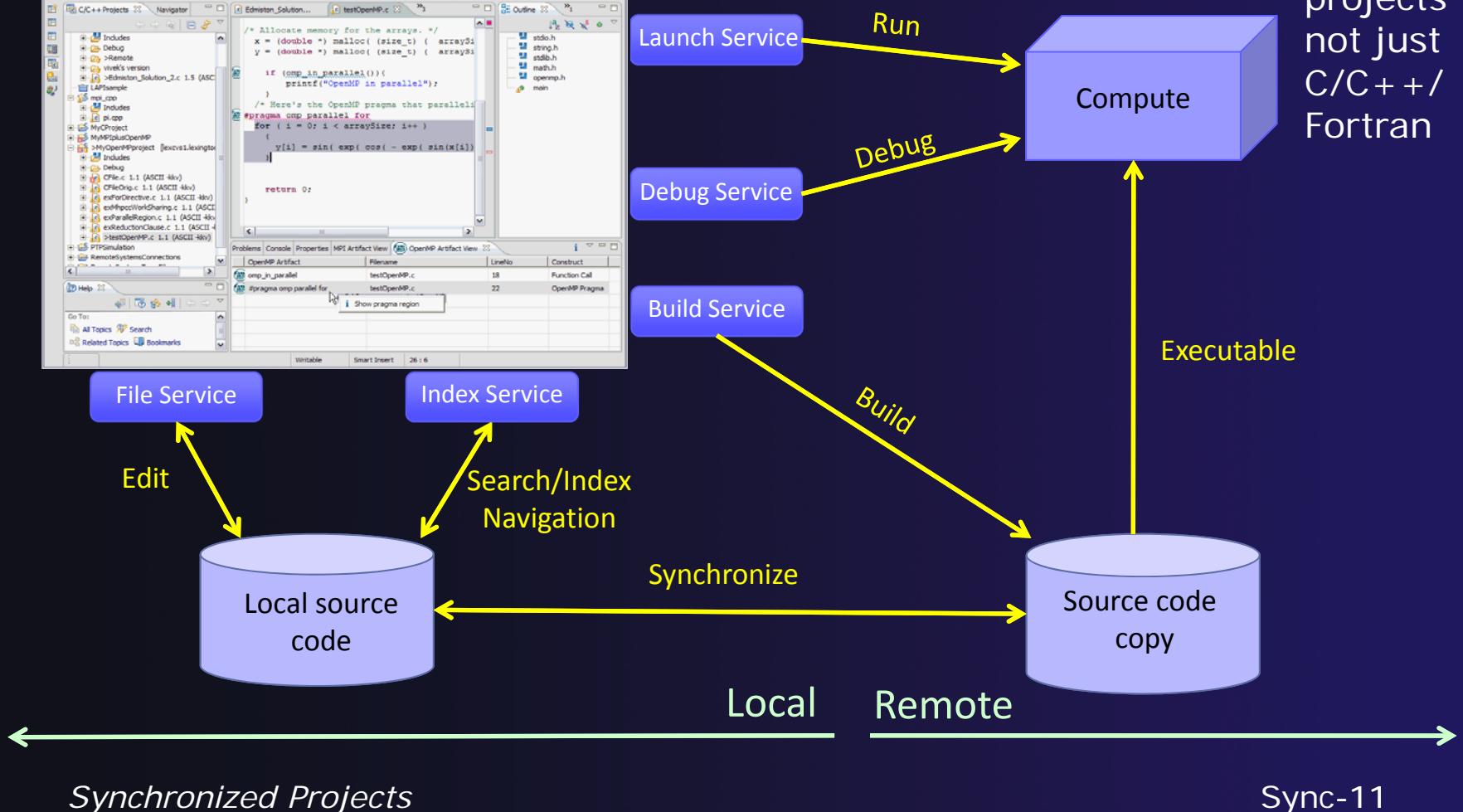
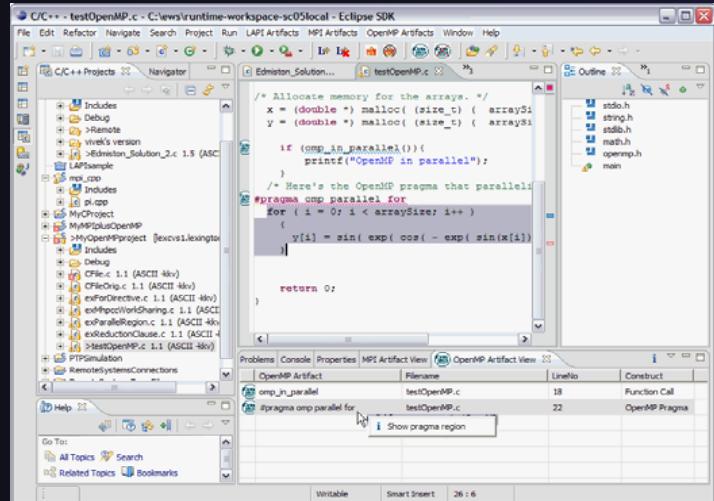


Stand-alone *Sysmon* monitoring application available

Run-10

Synchronized Projects

Now available for all projects not just C/C++/Fortran



Integrated
OpenACC
documentation
and PLDT
support

(added for BW)

Documentation
also available for
MPI, OpenMP

The screenshot shows an IDE interface with a code editor window titled "test1.f90". The code contains OpenACC directives, specifically a parallel loop. A code completion dropdown menu is open over the directive, listing various OpenACC directives. The menu includes:

- !\$acc cache - OpenACC cache directive
- !\$acc data - OpenACC data directive
- !\$acc end data - OpenACC end data directive
- !\$acc declare - OpenACC declare directive
- !\$acc host_data - OpenACC host_data directive
- !\$acc end host_data - OpenACC end host_data directive
- !\$acc kernels - OpenACC kernels directive
- !\$acc end kernels - OpenACC end kernels directive
- !\$acc kernels loop - OpenACC kernels loop directive
- !\$acc end kernels loop - OpenACC end kernels loop directive
- !\$acc loop - OpenACC loop directive

Below the code editor is a tab bar with "Problems" and "Fortran Declaration". The main content area displays the title "OpenACC™ parallel directive" and a description: "Delineates a block of code that will be executed on an accelerator device." It also shows two code snippets for the parallel directive:

```
!$acc parallel [clause [, clause ...]]    #pragma acc parallel [clause [, clause ...]]
  block                                     block
!$acc end parallel
```

Supported clauses are listed as: if, async, num_gangs, num_workers, vector_length, reduction, copy, copyin, copyout, create, present, present_or_copy, present_or_copyin, present_or_copyout, present_or_create, deviceptr, private, firstprivate.

```

5      integer :: i, j
6      double precision :: a(SIZE,SIZE), b(SIZE,SIZE), c(SIZE,SIZE)
7      a(:)=0...! This will raise a warning! ←
8
9      do i=1,SIZE
10     do j=1,SIZE
11       a(i,j)=i*10.d0+j
12       b(i,j)=j*10.d0+i
13     end do
14   end do
15
16 !$acc parallel loop
17   do i=1,1000
18     c(:, :)=(a(:, :)+b(:, :))/2.d0
19     a(:, :)=(a(:, :)+c(:, :))/2.d0
20     b(:, :)=(b(:, :)+c(:, :))/2.d0
21   end do
22 !$acc end parallel loop
23
24 !print *, "Averages:"
25 !print *, sum(a(:, :)/(SIZE*SIZE))
26 !print *, sum(b(:, :)/(SIZE*SIZE))
27 !print *, sum(c(:, :)/(SIZE*SIZE))
28 print *, "Minimuns:", minval(a(:, :)), minval(b(:, :)), minval(c(:, :))

```

Problems X Console Fortran Declaration Fortran Analysis/Refactoring Problems

1 error, 1 warning, 17 others

Description	Resource
⚠ Warnings (1 item)	
⚠ The number of subscripts is smaller than the number of declared dimensions.	test1.f90
ℹ Infos (17 items)	
ℹ A divide was turned into a multiply by a reciprocal	test1.f90
ℹ A divide was turned into a multiply by a reciprocal	test1.f90
ℹ A divide was turned into a multiply by a reciprocal	test1.f90
ℹ A floating point expression involving an induction variable was strength reduced b...	test1.f90
ℹ A loop nest at line 18 collapsed to a single loop.	test1.f90
ℹ A loop starting at line 10 was not vectorized because a better candidate was found...	test1.f90
ℹ A loop starting at line 17 was blocked with block size 512	test1.f90

After the build, compiler errors, warnings, and loopmark information are shown in the Problems view and source code editor

(Cray, PGI support added for BW)

Resources Application Arguments Environment Synchronize Common

Resource Manager: ESS – Batch (XE)

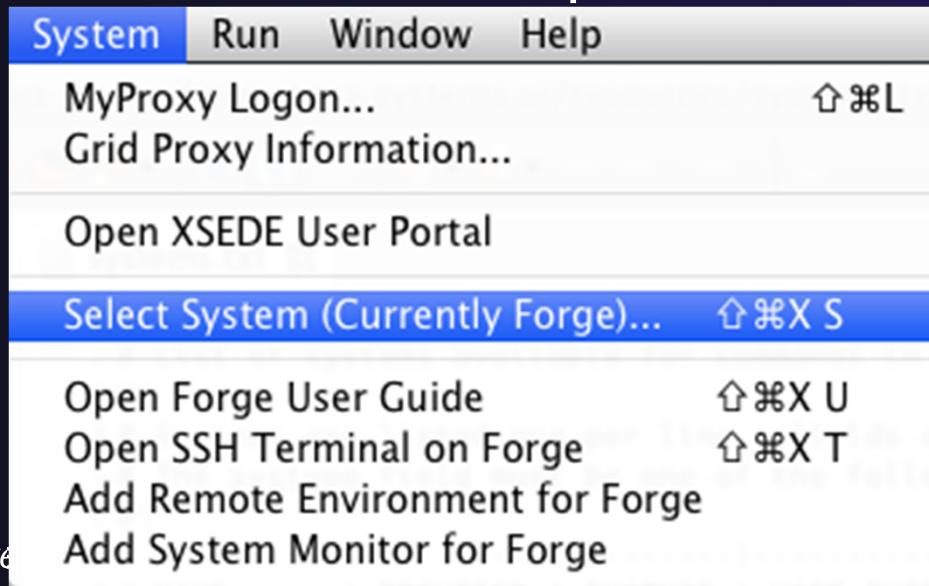
Basic PBS Settings Import PBS Script

Name	Value	Description
Total MPI Tasks:	32	Each XE6 node has two AMD Interlagos CPUs for a total of 32 integer cores and 16 floating point units per node. Therefore, the product of the number of MPI tasks per node and the number of OpenMP threads per task must be less than or equal to 32 (or 16 if running in single-stream mode). The number of MPI tasks per node must not exceed the total number of MPI tasks.
MPI Tasks per Node:	32	
OpenMP Threads per Process:		
Run in Dual-Stream Mode:	<input checked="" type="checkbox"/>	XE6 nodes are normally run in "dual-stream mode," where every integer core is allocated one task (i.e., one MPI task or one OpenMP thread). However, this means that every two tasks share a floating point unit. Some floating-point-intensive computations may need to run in "single-stream mode," where every other integer core is idle but every task has exclusive access to a floating point unit.
Job Name:	ptp_job	The name assigned to the job by the qsub or qalter command.
Account:		Account to which to charge this job.
Queue:		Designation of the queue to which to submit the job.
Total Memory Needed:		Maximum amount of memory used by all concurrent processes in the job.

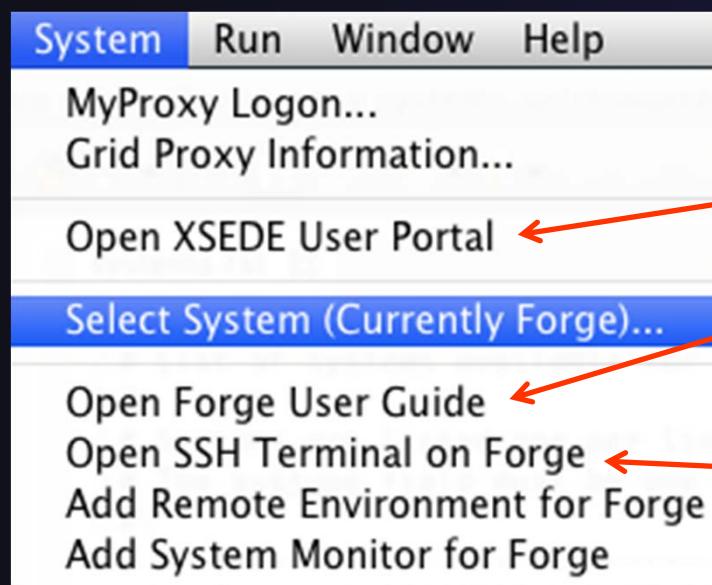
Graphical interface for launching a job (customized for BW)

Additional Plug-ins from NCSA

- ★ NCSA publishes additional plug-ins can be added onto an existing PTP installation
 - ★ <http://forecaster.ncsa.uiuc.edu/help/index.jsp>
 - ★ *(This site to be updated soon with new components)*
- ★ Contribute a **System** menu to the menu bar with XSEDE- and NCSA-specific commands



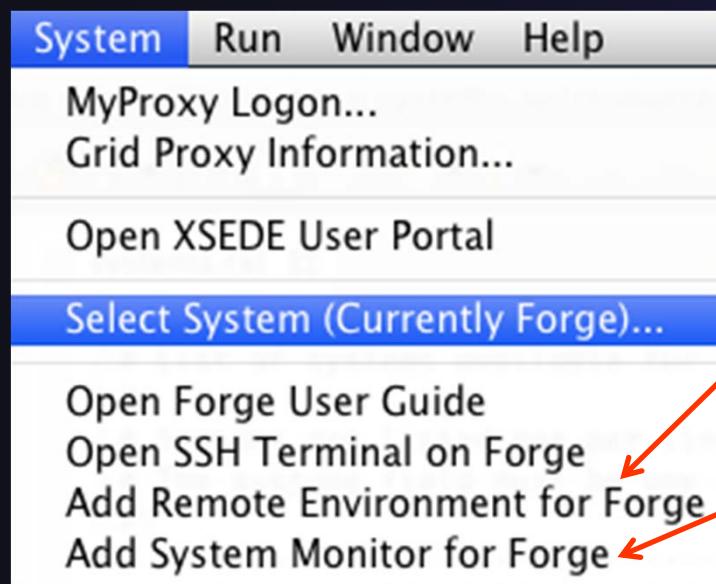
System Menu



- ★ Open Web content in Eclipse:
 - ★ Open XSEDE User Portal
 - ★ Open User Guide for a machine
- ★ Open an SSH terminal
(as an Eclipse view)

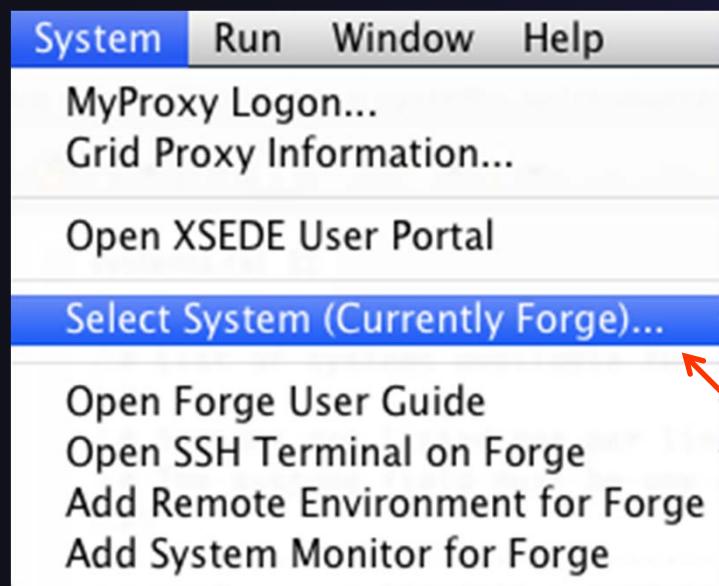
Eclipse-integrated SSH terminals are provided by the Remote System Explorer (RSE), one of the features that is included in the Eclipse for Parallel Application Developers package.

System Menu



- ★ Shortcuts for common PTP tasks:
 - ★ **Add Remote Environment** adds a Remote Tools connection for a particular machine
 - ★ **Add System Monitor** opens the System Monitoring perspective and begins monitoring a particular machine

System Menu



- ★ The plug-in is preconfigured with information about XSEDE and NCSA resources
- ★ The bottom four commands generally prompt for a system
- ★ **Select System** can be used to eliminate this prompt, so these commands always act on a particular system

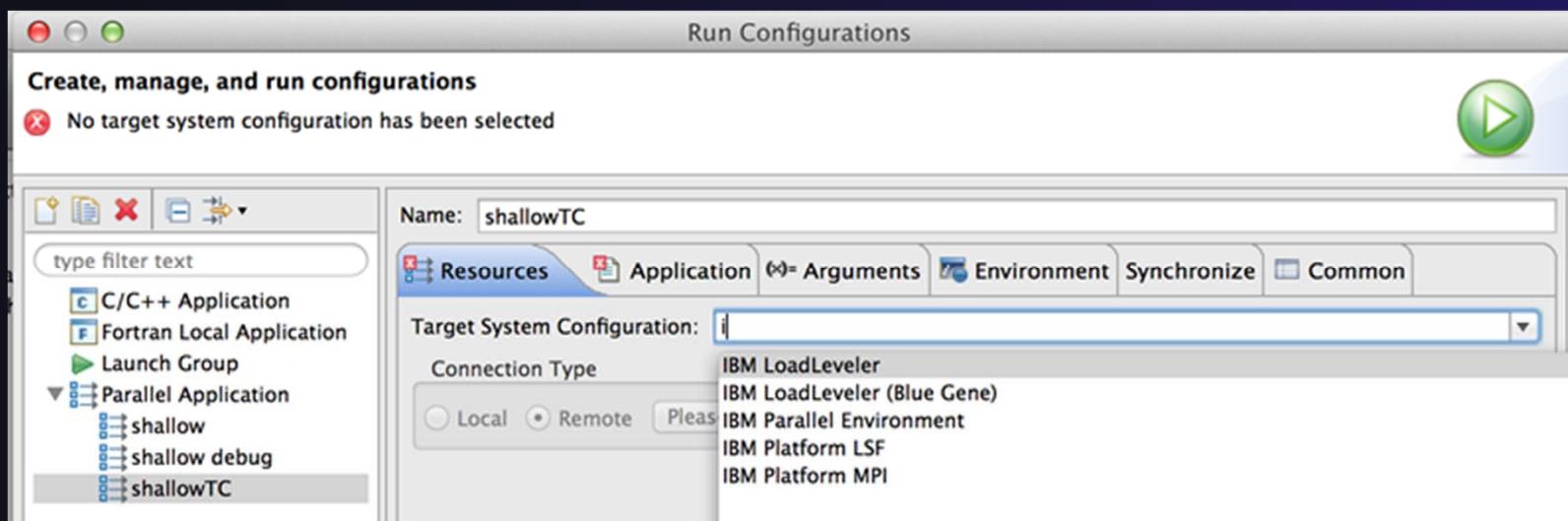
MyProxy Logon



- ★ **MyProxy Logon** allows you to authenticate with a MyProxy server
 - ★ Often **myproxy.teragrid.org**
- ★ It stores a “credential,” which is usually valid for 12 hours
- ★ During these 12 hours, SSH connections to XSEDE resources will not require a password; they can use the stored credential
 - ★ However, you **must** enter the correct username for that machine!

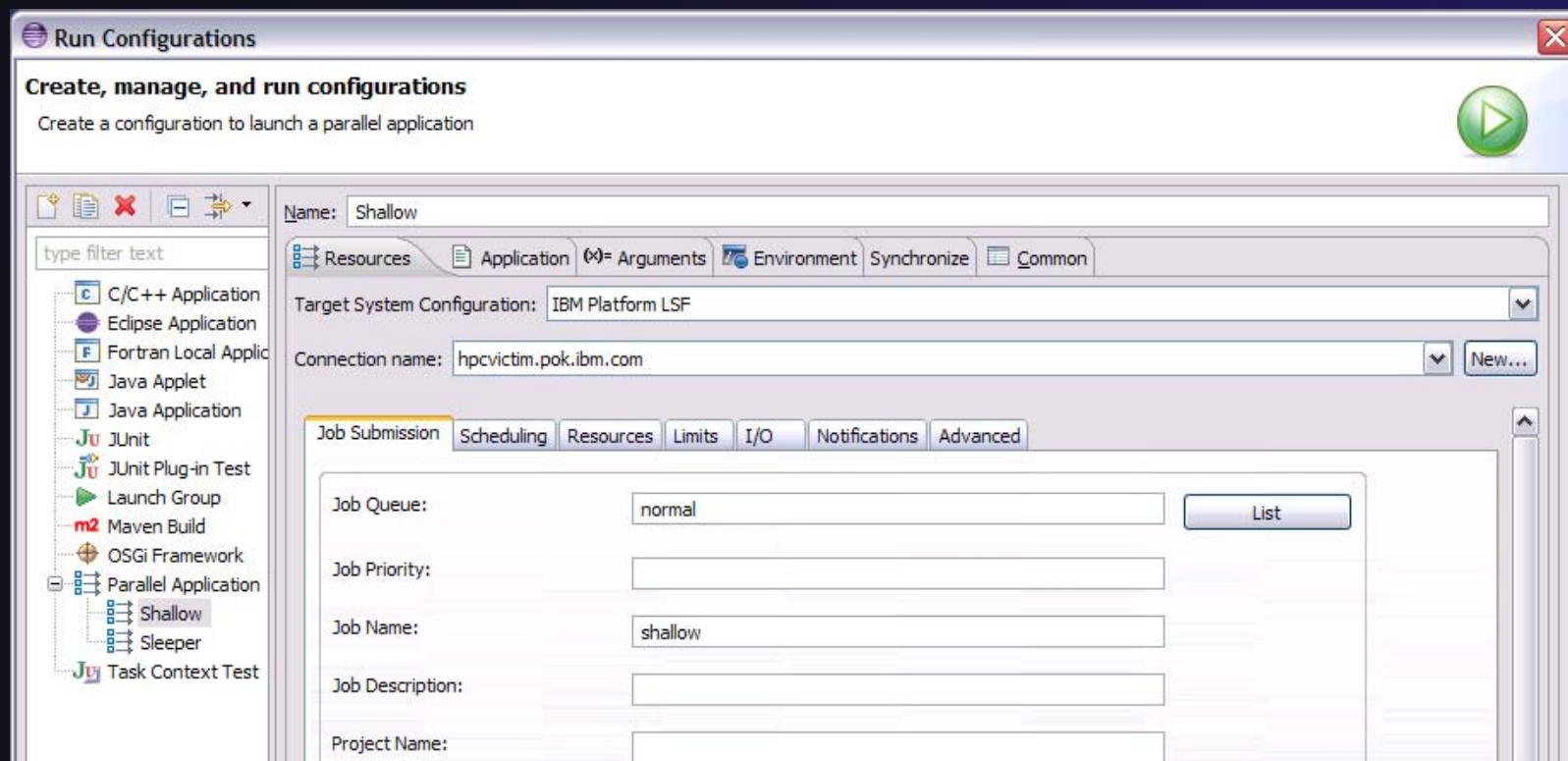
New Features (Kepler)

- ★ Target System Configurations
 - ★ Selection filtering – type starting letters of target system configuration to pull up list of possibilities



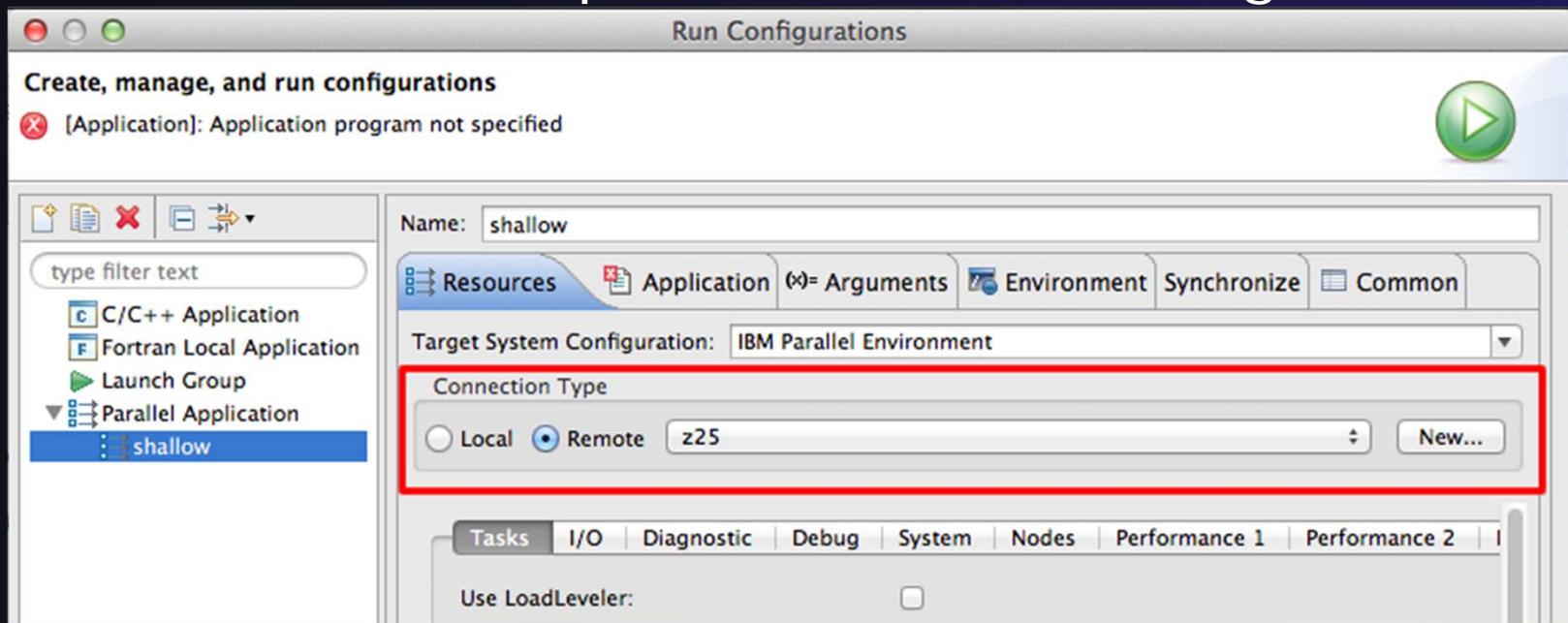
New Features (continued)

- ★ New Target System Configurations
 - ★ IBM Platform LSF, IBM Platform MPI



New Features (continued)

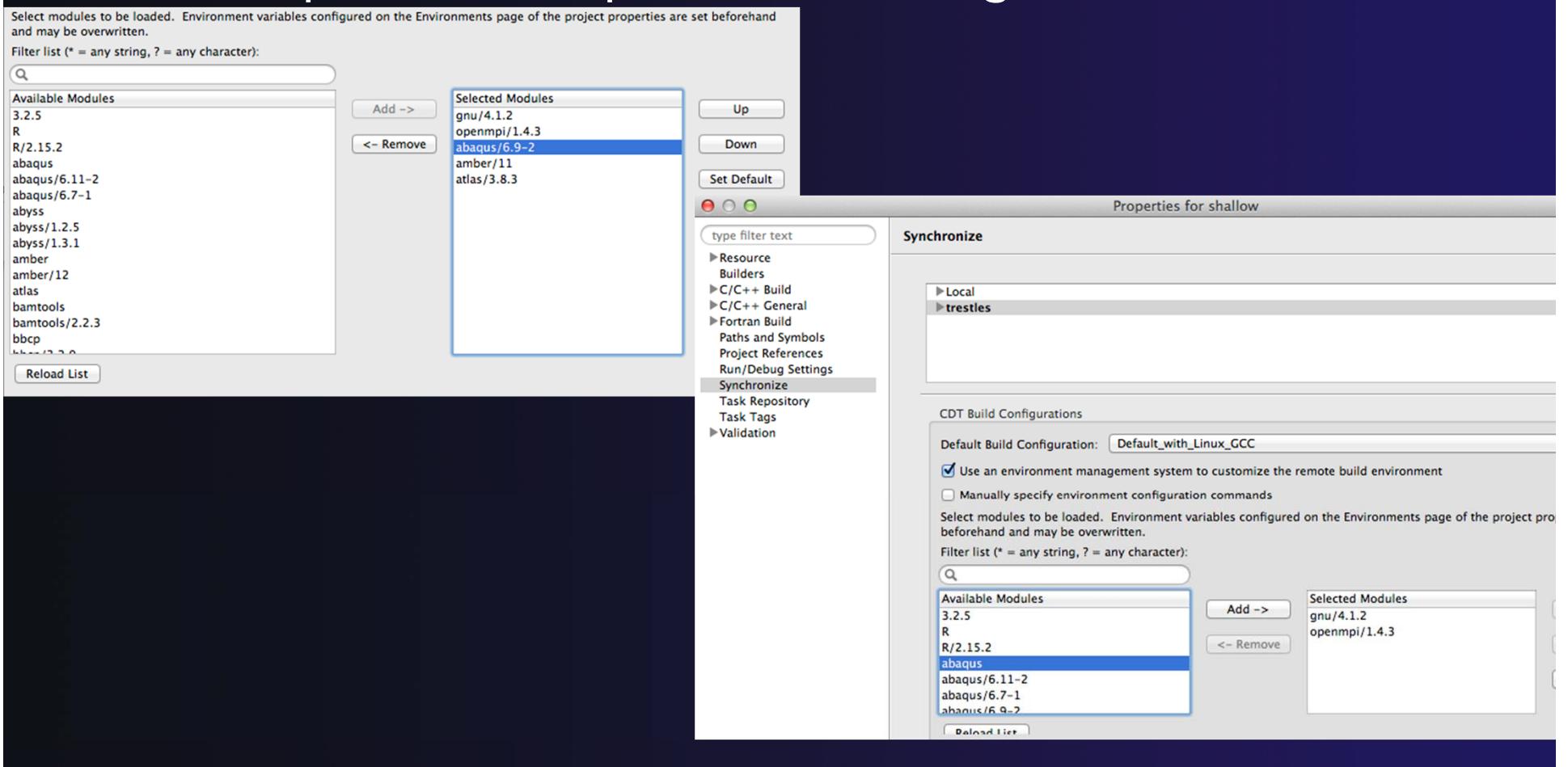
- ★ Direct **Local** option for launch configuration



parallel tools platform

New Features: Environment Management

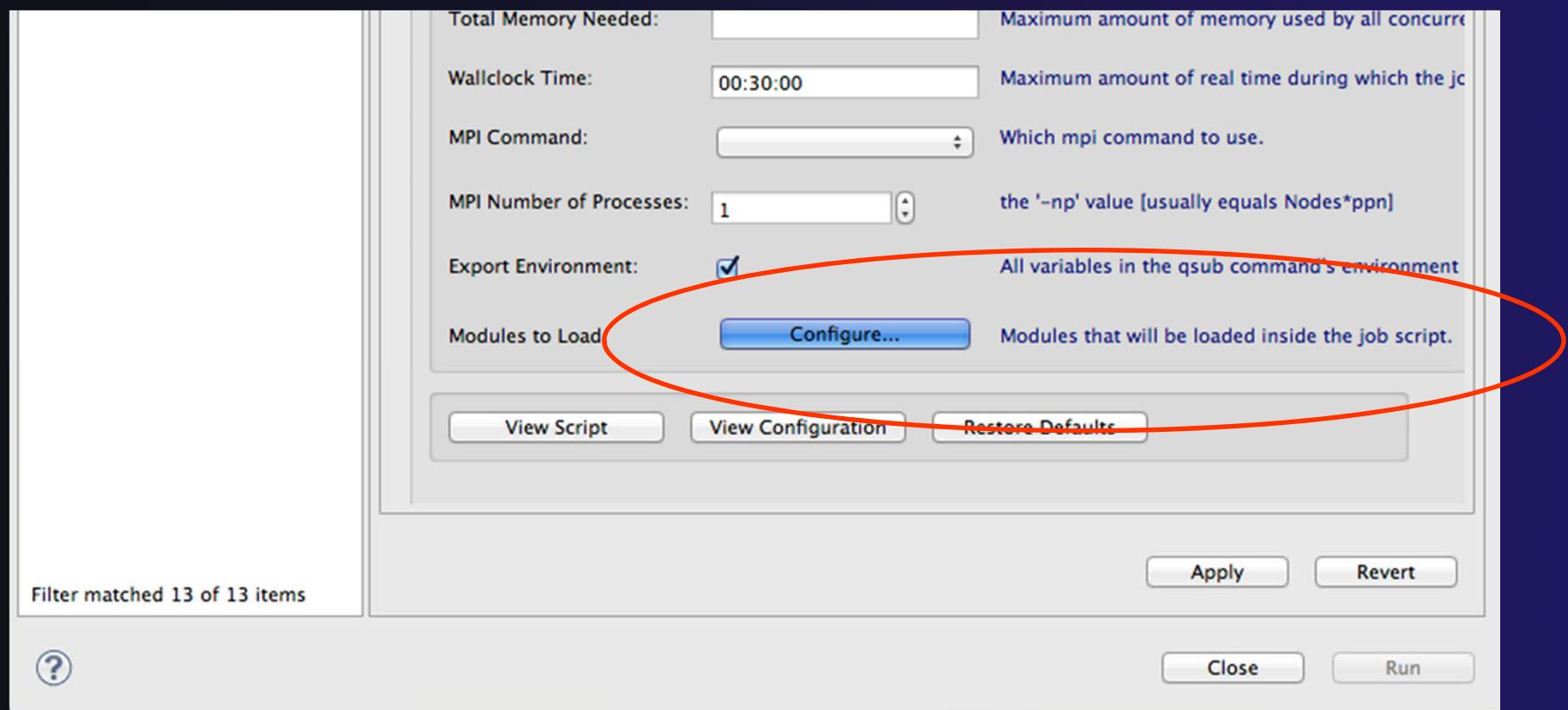
- ★ Module ordering can be specified, modules can be specified on per-build configuration



parallel tools platform

New Features: Environment Management

- ★ Target configurations support module specification for launch/debug:



More recent updates

- ★ External Tools Framework (ETFW) transitioned to use the target configuration “JAXB” XML descriptions.
 - ◆ Enhanced flexibility for tool integration
- ★ Remote scanner-discovery support for synchronized projects/gcc compiler
 - ◆ Eclipse builds index based on remote environment
- ★ Separate Build system from CDT build
 - ◆ Synchronized configuration now separate from build configuration
 - ◆ Cleaner multi-system build management
 - ◆ Builds in any language
- ★ Automatic deployment of scalable debug manager for debugger

Improvements in the works...

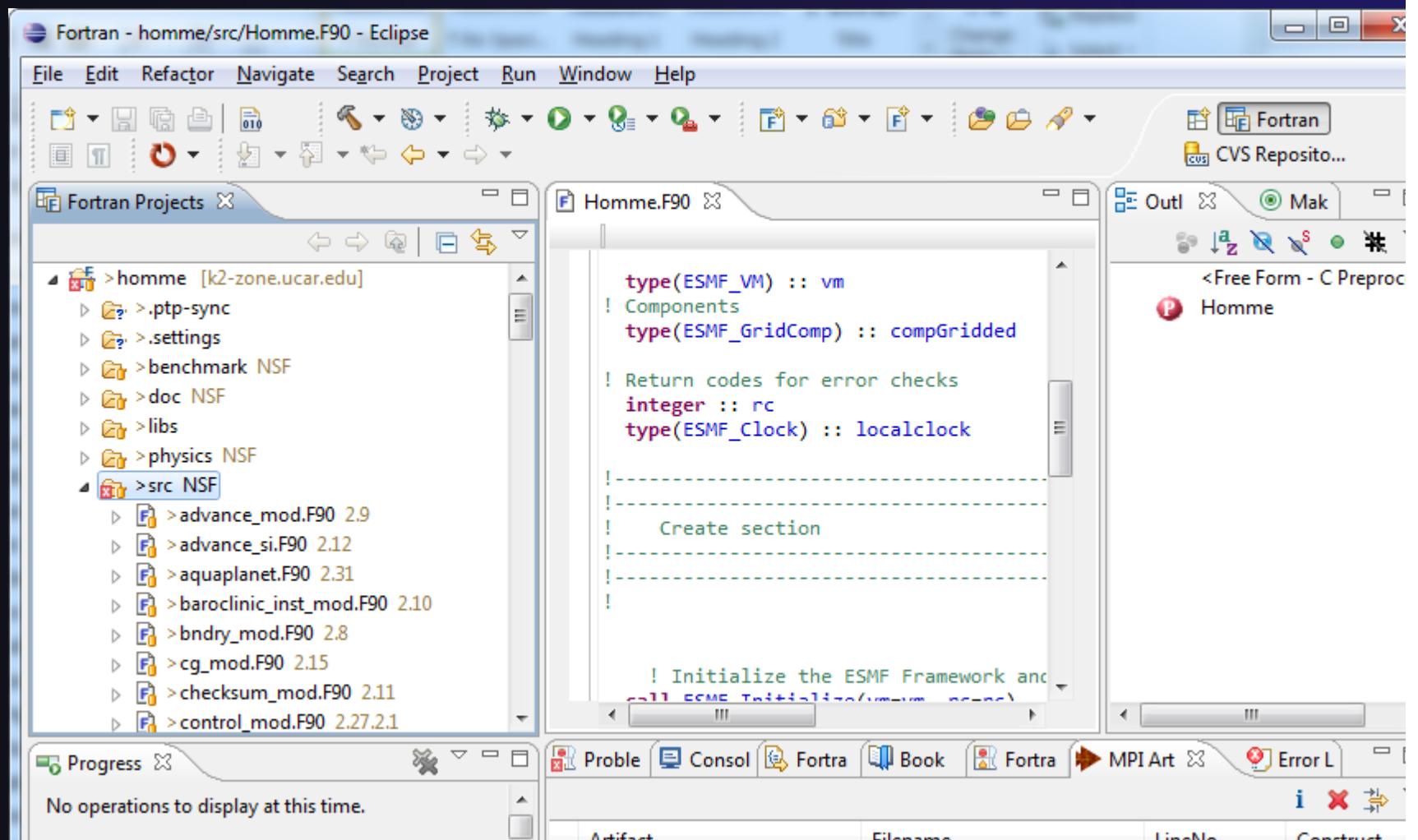
- ★ Environment Management enhancements
 - ★ Correct interaction with hierarchical module systems
(eg, lmod)
- ★ Remote support pushed upstream of PTP
 - ★ Can be used by other projects in PTP, becoming closer to a “platform” feature
- ★ Community input desired:
 - ★ OpenMP 4.0 development support?
 - ★ MPI 3.0 development support?
 - ★ Distribution of GSI components with Parallel Tools Platform download?
 - ★ Anything else?

Outline

- ★ Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- ★ Motivation for Workbench for High Performance Computing (WHPC)
 - ★ Improvements to Eclipse PTP
- ★ Software Engineering Practices Enabled by Eclipse PTP
 - ★ Code visibility
 - ★ Multi-system build management
 - ★ Performance tuning
 - ★ Source code control
 - ★ Issue Tracking
 - ★ Documentation
 - ★ Earth Science/Weather code example
- ★ Eclipse PTP Resources

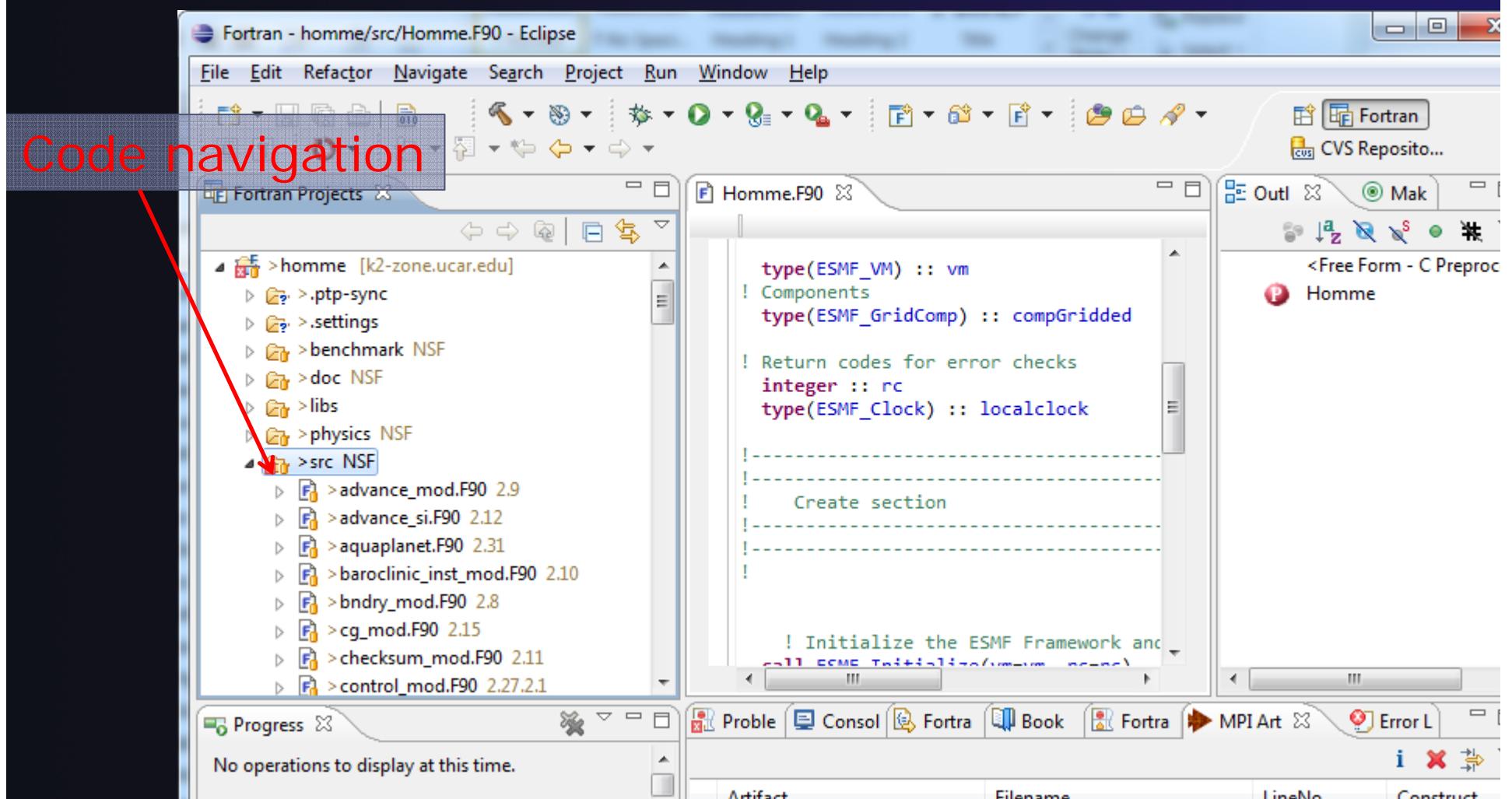
Software Engineering

★ Code Visibility



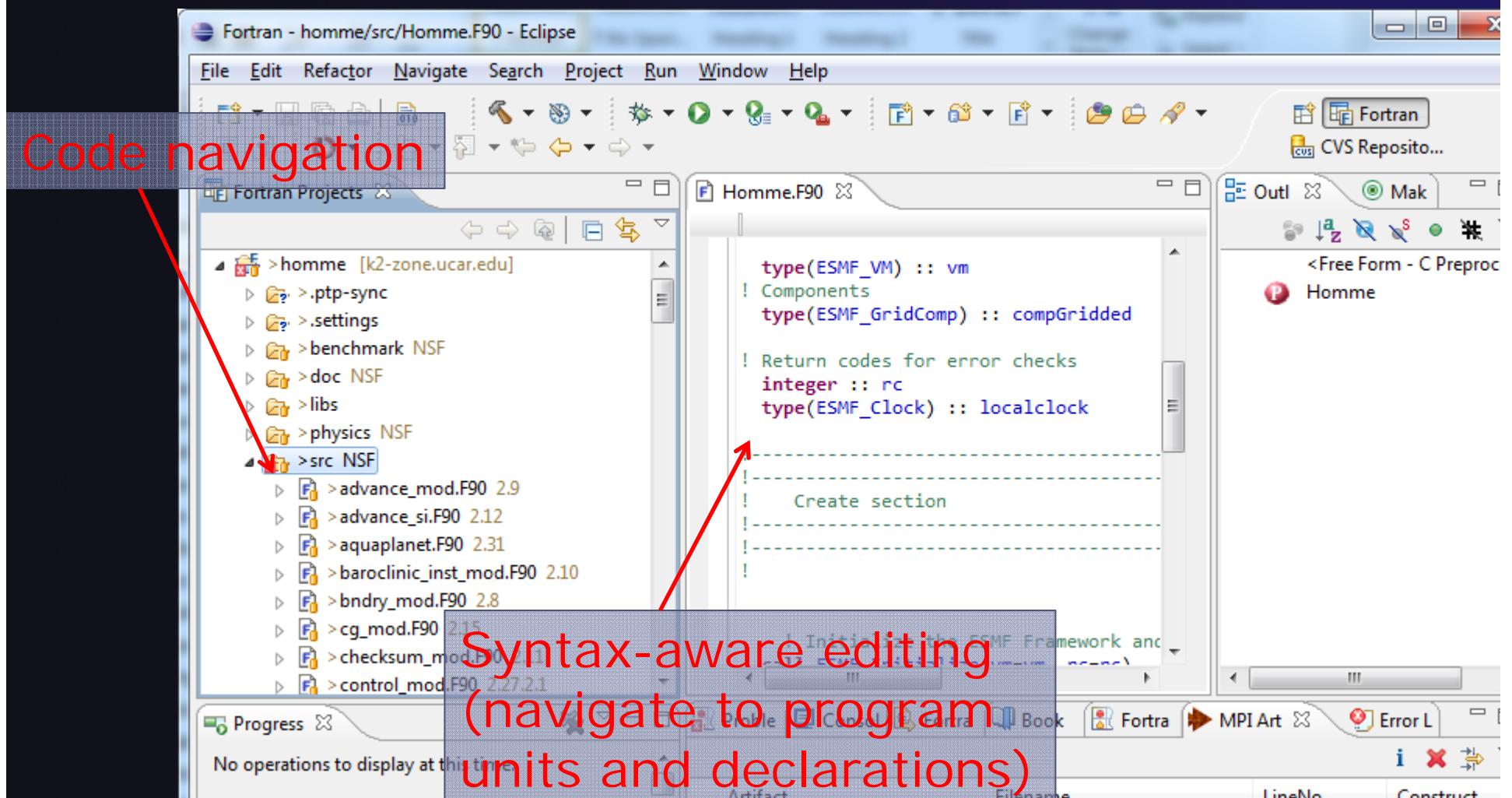
Software Engineering

★ Code Visibility



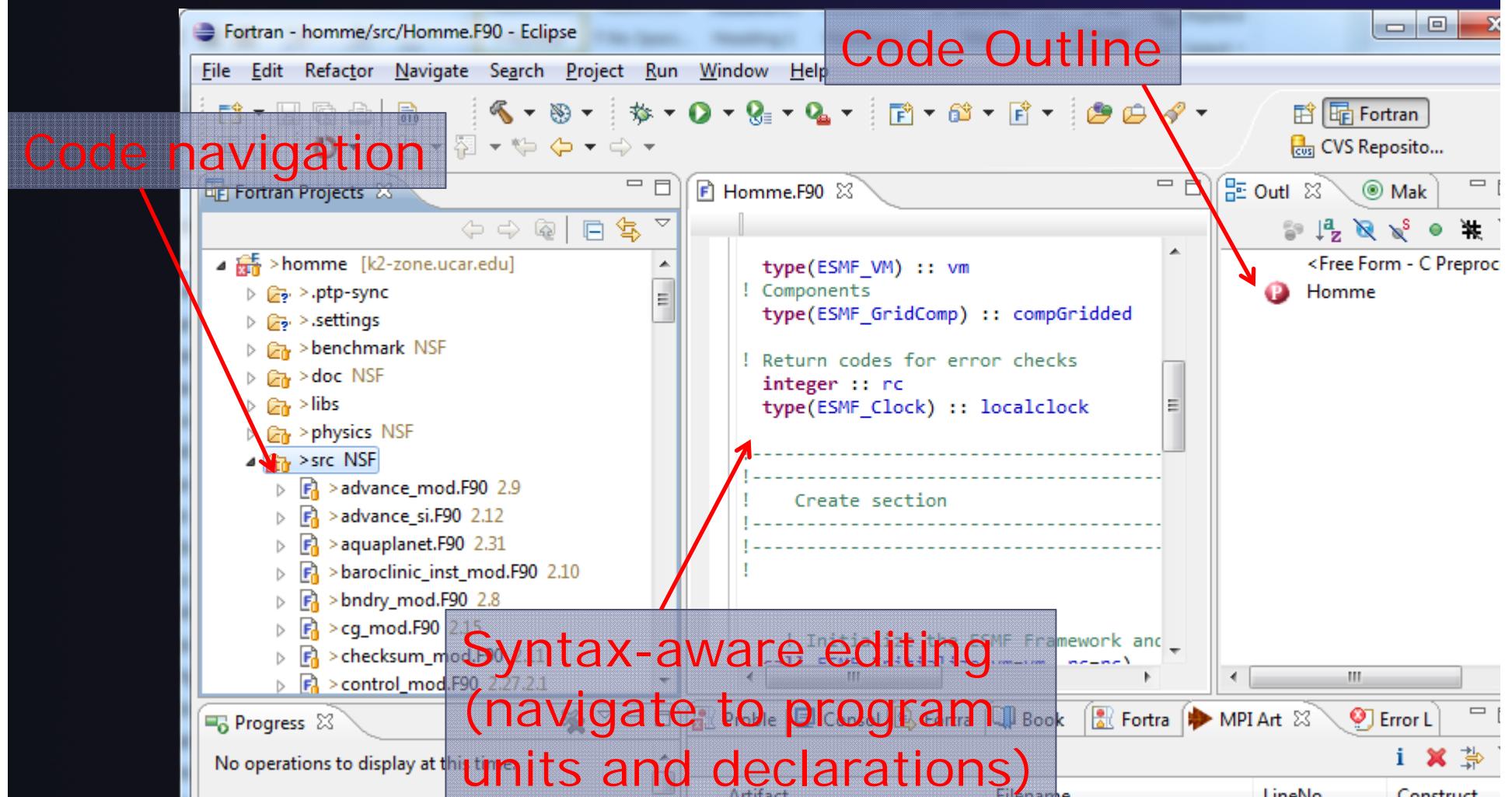
Software Engineering

★ Code Visibility



Software Engineering

★ Code Visibility



Software Engineering

- ★ Code visibility: deducing call hierarchy

The screenshot shows the Eclipse C/C++ IDE interface. The central window displays the code for `startup.c`, specifically the `main` function:

```
int main(int argc, char *argv[])
{
    int ch;
    port = PTP_HOST;
    host = NULL;
    debugger_st = proxy_str = path = NULL;
    p;
    d;
#endif
    if (ch == 0) {
        switch (opt_type) {
            case OPT_TYPE_DEBUGGER:
                debugger_str = "break";
                break;
```

The `Project Explorer` view on the left shows a tree structure of source files under the `proxy` project. A red arrow points from the text "Would like to understand call hierarchy of this code in relation to "main()"" to the `main` function in the code editor.

A red box highlights the `main` function in the code editor. A red arrow points from this box to the `main` entry in the `Project Explorer` tree.

The `Startup.c` file is open in the editor, showing the beginning of the `main` function. The code includes declarations for variables like `ch`, `port`, `host`, etc., and handles the first command-line argument (`argc`) using `getopt_long`.

The `Output` view on the right lists various header files and defines:

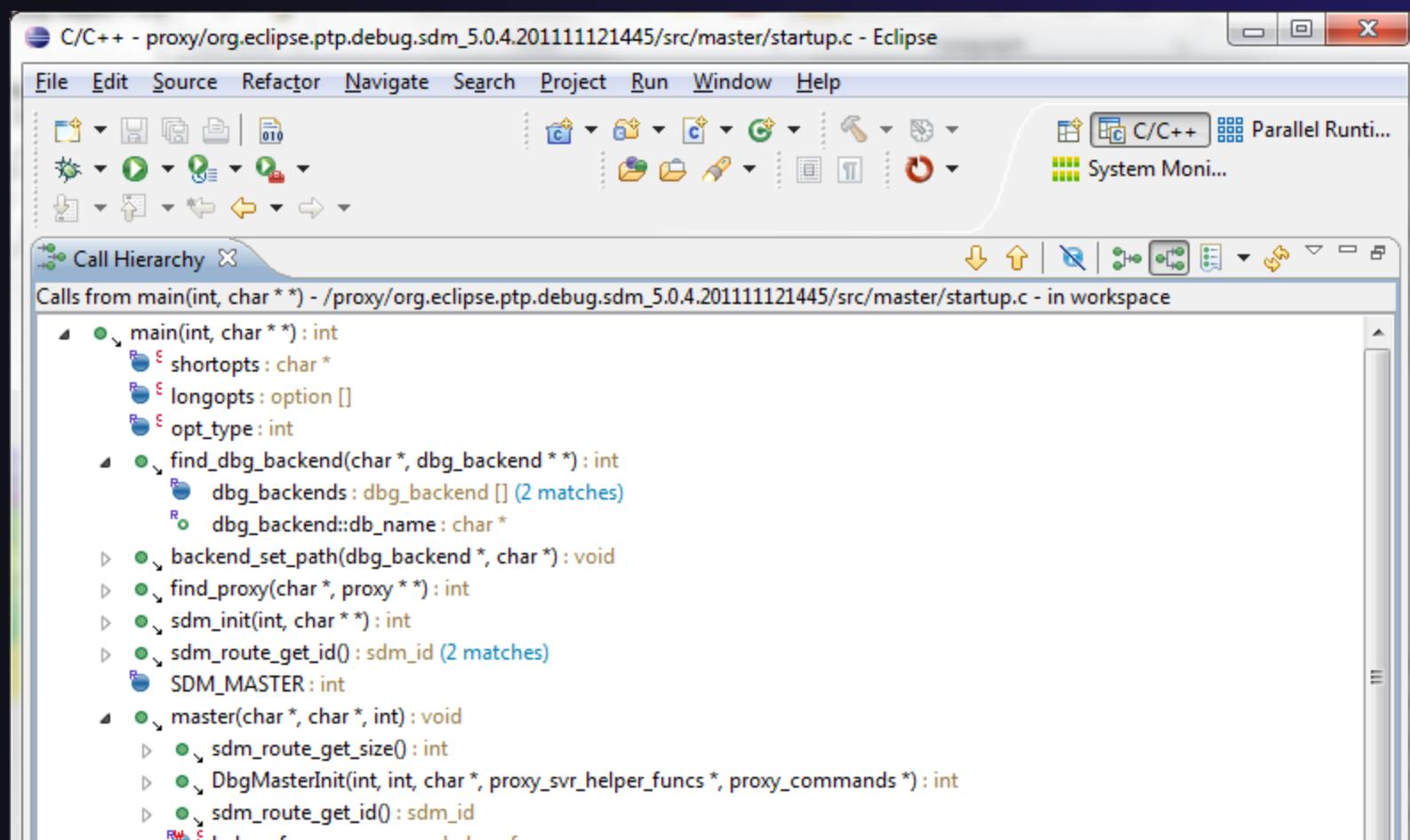
- config.h
- getopt.h
- stdlib.h
- stdarg.h
- backend.h
- proxy.h
- proxy_tcp.h
- sdm.h
- # DEFAULT_BACKEND
- # DEFAULT_PROXY
- master(char*, char*, int) : void
- server(dbg_backend*) : void
- # OPT_TYPE_DEBUGGER
- # OPT_TYPE_DEBUGGER_PAT
- # OPT_TYPE_PROXY
- # OPT_TYPE_PORT
- # OPT_TYPE_HOST
- # OPT_TYPE_MASTER

Would like to understand call hierarchy of this code in relation to "main()" in startup.c

parallel tools platform

Software Engineering: Call Hierarchy (C/C++)

- ★ After selecting main, right click and select <Open Call Hierarchy>



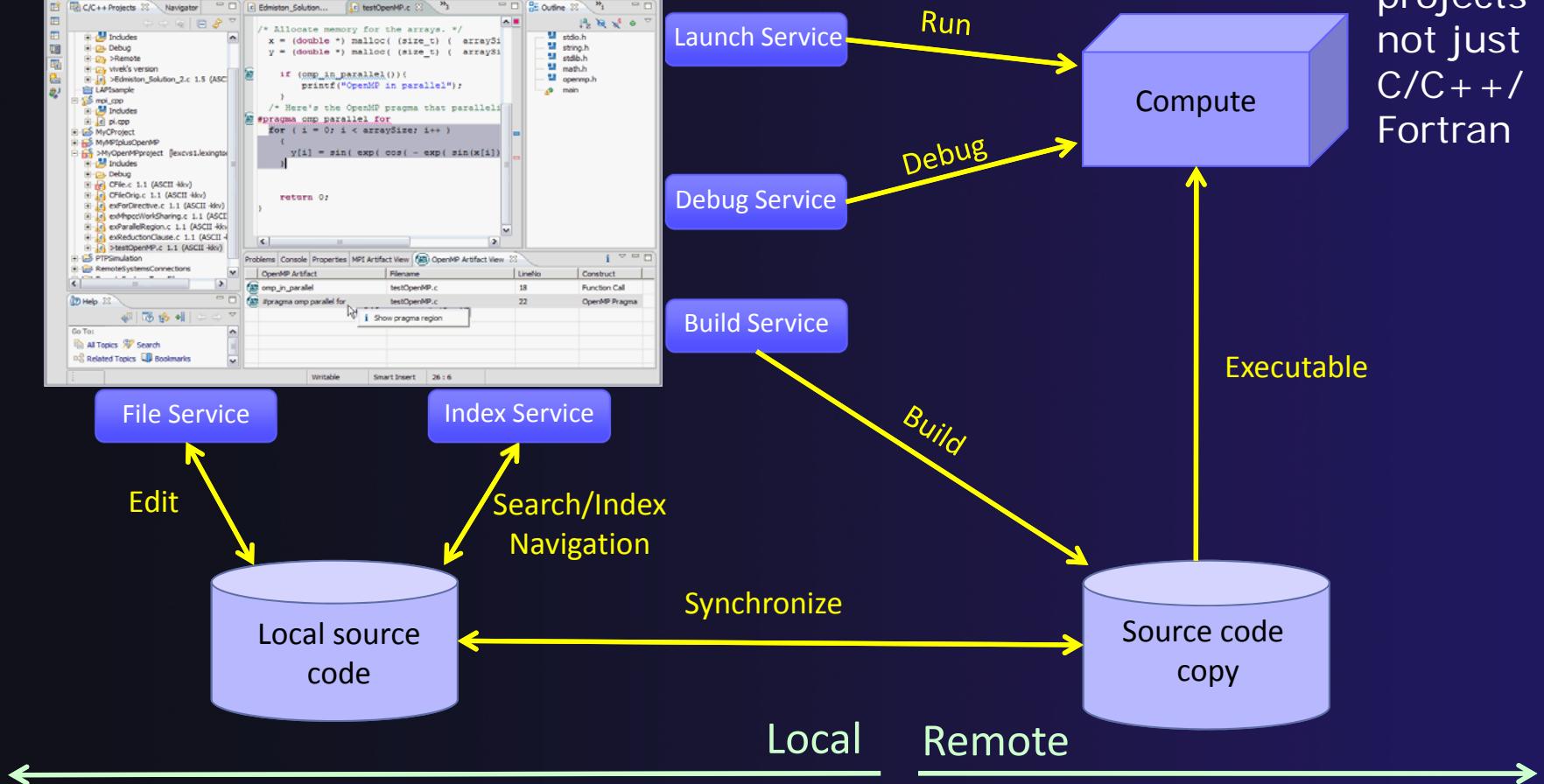
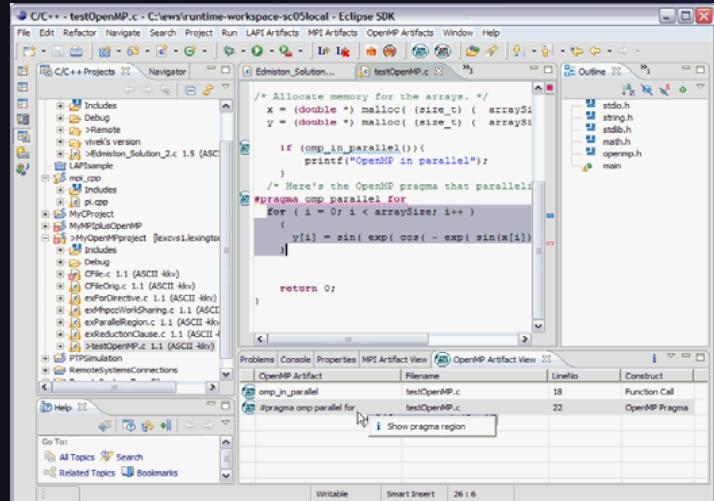
Multi-machine build management

Three types of Eclipse projects

- ★ Local
 - ★ Source is located on local machine, builds happen locally
- ★ Synchronized
 - ★ Source is local, then synchronized with remote machine(s)
 - ★ Building and launching happens remotely (can also happen locally)
- ★ Remote
 - ★ Source is located on remote machine(s), build and launch takes place on remote machine(s)

Synchronized Projects

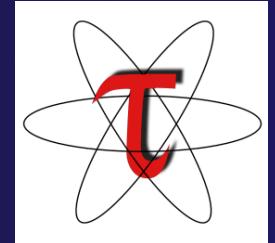
Now available for all projects not just C/C++/Fortran



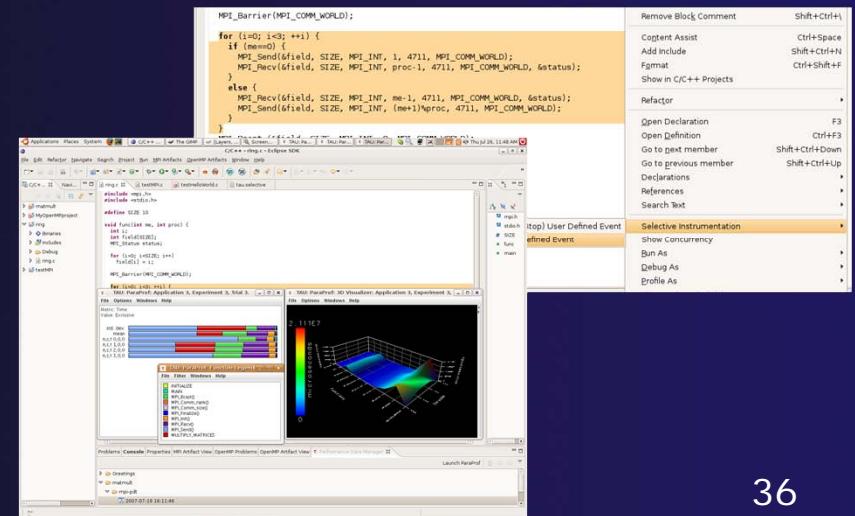
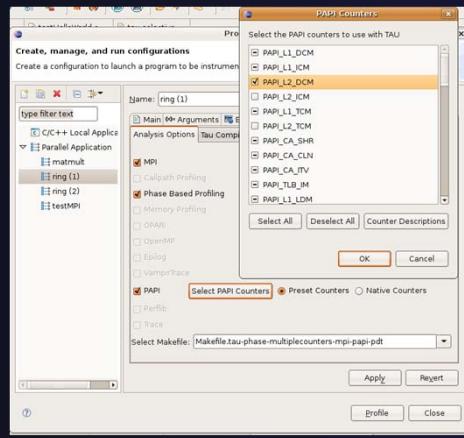
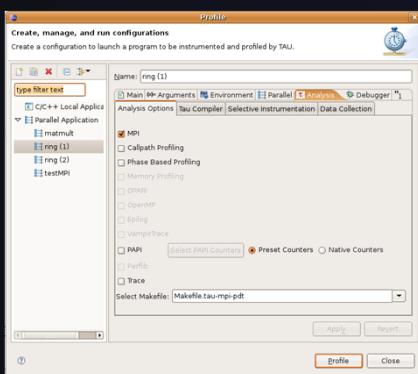
parallel tools platform

Performance Tuning: PTP TAU plug-ins

<http://www.cs.uoregon.edu/research/tau>



- ◆ TAU (Tuning and Analysis Utilities)
- ◆ First implementation of External Tools Framework (ETFw)
 - ◆ Support for additional command-line tools can be added with XML tool definitions
- ◆ Eclipse plug-ins wrap TAU functions, make them available from Eclipse
- ◆ Full GUI support for the TAU command line interface
- ◆ Performance analysis integrated with development environment

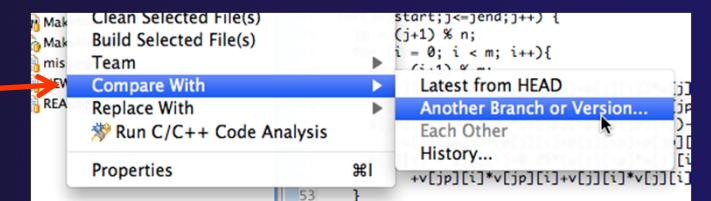
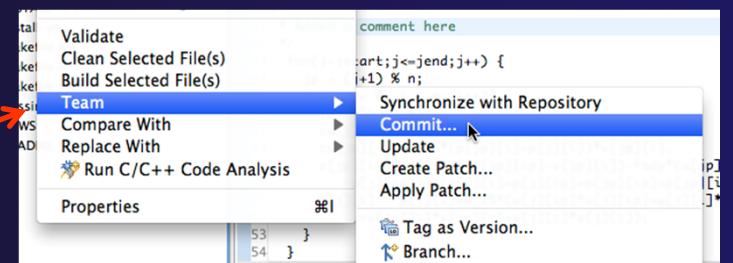
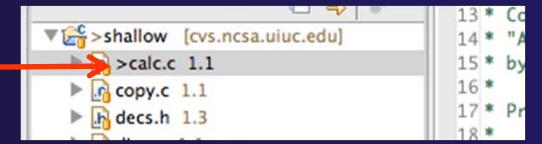
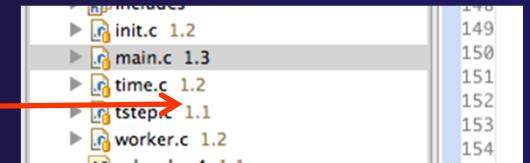


Source Code Control: “Team” Features

- ★ Eclipse supports integration with multiple version control systems (VCS)
 - ★ CVS, SVN, Git, and others
 - ★ Collectively known as “Team” services
- ★ Many features are common across VCS
 - ★ Compare/merge
 - ★ History
 - ★ Check-in/check-out
- ★ Some differences
 - ★ Version numbers
 - ★ Branching

CVS Features

- ★ Shows version numbers next to each resource
- ★ Marks resources that have changed
 - ★ Can also change color (preference option)
- ★ Context menu for Team operations
- ★ Compare to latest, another branch, or history
- ★ Synchronize whole project (or any selected resources)
- ★ Similar support for SVN, Git, ...

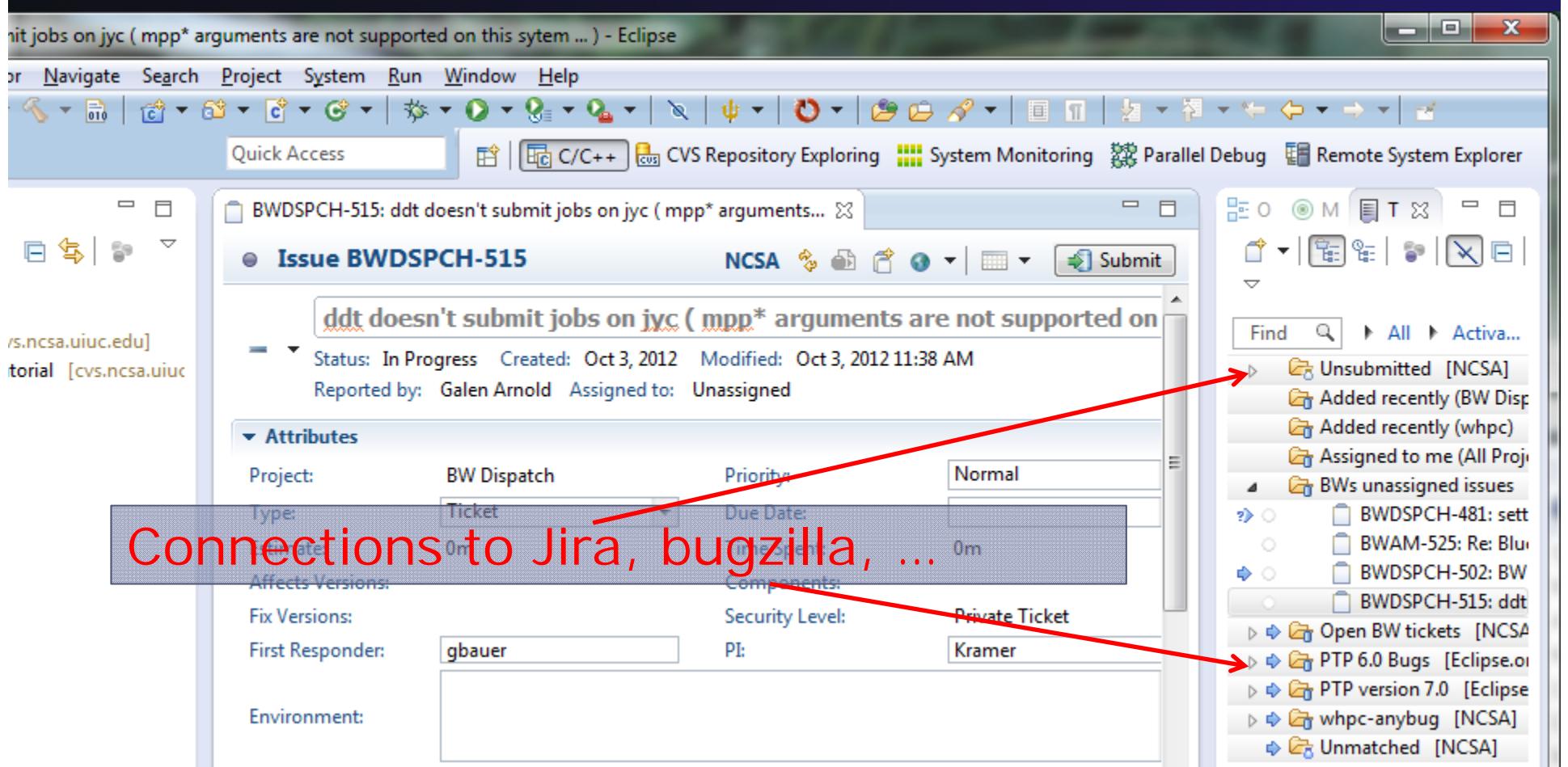


parallel tools platform

Issue Tracking

- ★ Mylyn Bridge

- ★ Tracks tasks, links to source and bug repositories



Eclipse Documentation

- ★ Eclipse Help System – built in and standalone
(<http://help.eclipse.org>)

The screenshot shows the Eclipse Help System interface. The title bar reads "Help - Eclipse". The left sidebar contains a search bar, a "Scope: All topics" button, and a "Contents" button. Below these are links to various user guides, including "Workbench User Guide", "Autotools Plug-in User Guide", "C/C++ Development User Guide", "C/C++ Library Documentation", "ChangeLog Editor User Guide", "Eclipse Marketplace User Guide", "EGit Documentation", "Fortran Development User Guide", "GCov User Guide", "GNU Tools On-line Documentation", "GProf User Guide", "Libhover Developer's Guide", "LTTng Plug-in User Guide", "Mylyn Documentation", and "Parallel Development User Guide". The "Parallel Development User Guide" link is selected, highlighted with a blue border. The main content area displays the "Parallel Development User Guide >" page. A large icon of a purple starburst is on the left. The title "Parallel Tools Platform Release 7.0" is centered above a bulleted list of features:

- tools for developing applications based on the Message Passing Interface (MPI) standard (and other environments and APIs including OpenMP, UPC, OpenSHMEM, OpenACC, etc.)
- the ability to launch, control and monitor the execution of parallel programs
- the ability to monitor parallel system status information
- an integrated parallel debugger
- a framework for integrating external dynamic tools

In addition, PTP provides a platform for parallel tool developers to integrate their tools within the Eclipse framework. New tools can take advantage of the user interface components and parallel services that are provided by PTP without the need to develop and support this infrastructure across multiple platforms.

More information and downloads are available at <http://eclipse.org/ptp>.

PTP 7.0 is released as part of the Eclipse Kepler Simultaneous release (June 2013).

Overview, Background, and Setup information

1. [Overview and Features](#)
2. [Prerequisites](#)

parallel tools platform

Adapting Eclipse Documentation to Other Projects: QMCPack

- ★ See <http://code.google.com/p/qmcpack-doc/>

The screenshot shows a web browser displaying the Google Code project page for 'qmcpack-doc'. The URL in the address bar is <http://code.google.com/p/qmcpack-doc/>. The page title is 'qmcpack-doc - QMCPACK document'. The main navigation menu includes 'Project Home' (which is selected), 'Downloads', 'Wiki', 'Issues', and 'Source'. Below the menu, there are links for 'Summary', 'Updates', and 'People'. A red arrow points from the text 'Instructions for viewing help page in eclipse' down towards the 'Instructions for viewing help page in eclipse' link in the 'Project Information' sidebar.

qmcpack-doc
QMCPACK document

Project Home Downloads Wiki Issues Source

Summary Updates People

Project Information

Activity Low Project feeds

Code license New BSD License

Members

[jeongnim.kim](#),
[david.ce...@gmail.com](#),
[kpes...@gmail.com](#),
[jmcmminis_bkcl...@gmail.com](#),
[jaronkro...@gmail.com](#),
[miguel.m...@gmail.com](#),
[lshulenb...@gmail.com](#)
5 committers

Developers' and users' guides

org.cmscc.qmcpack.doc is developed as an eclipse plug-in for QMCPACK help page. If all goes well, a help document with

- build instructions
- doxygen code documentation
- other materials on wiki

can be downloaded as an eclipse plug-in.

Licensed under [UIUC/NCSA open-source license](#)

See more on [UIUC/NCSA license](#)

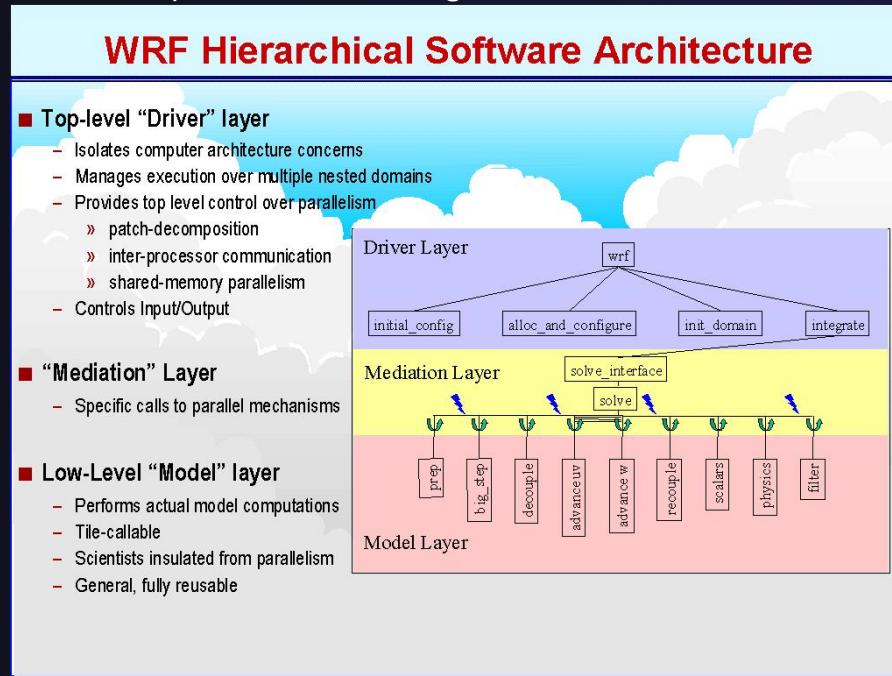
Instructions for viewing help page in eclipse

Consider 2 possible types of users of Eclipse Parallel Tools ...

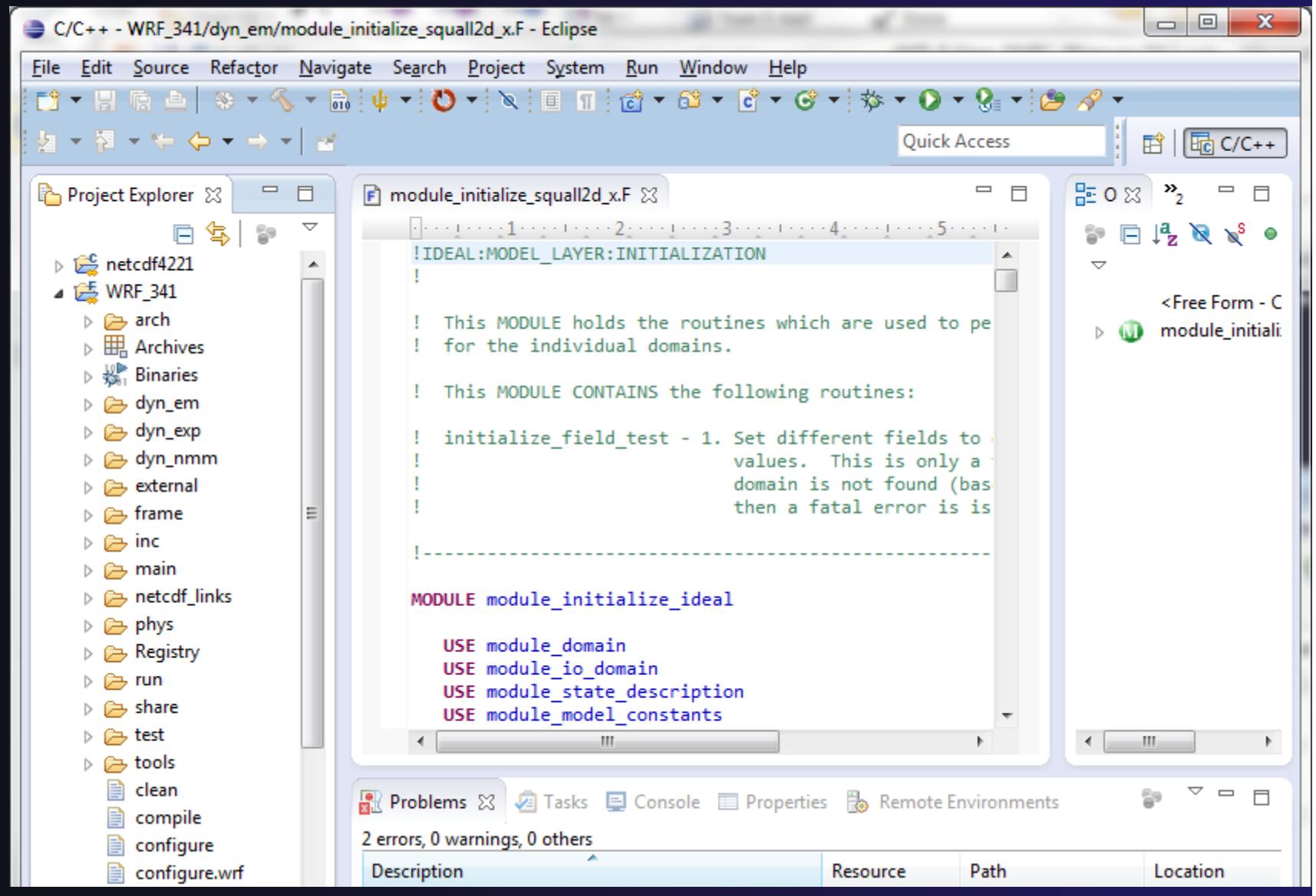
- ★ Weather code users/modelers
 - ★ Need to build weather code
 - ★ May need to modify weather code (and rebuild)
- ★ Software specialists enabling modeling projects
 - ★ Lots of software engineering concerns
- ★ Next set of slides address some of those concerns.

Weather code users/modelers

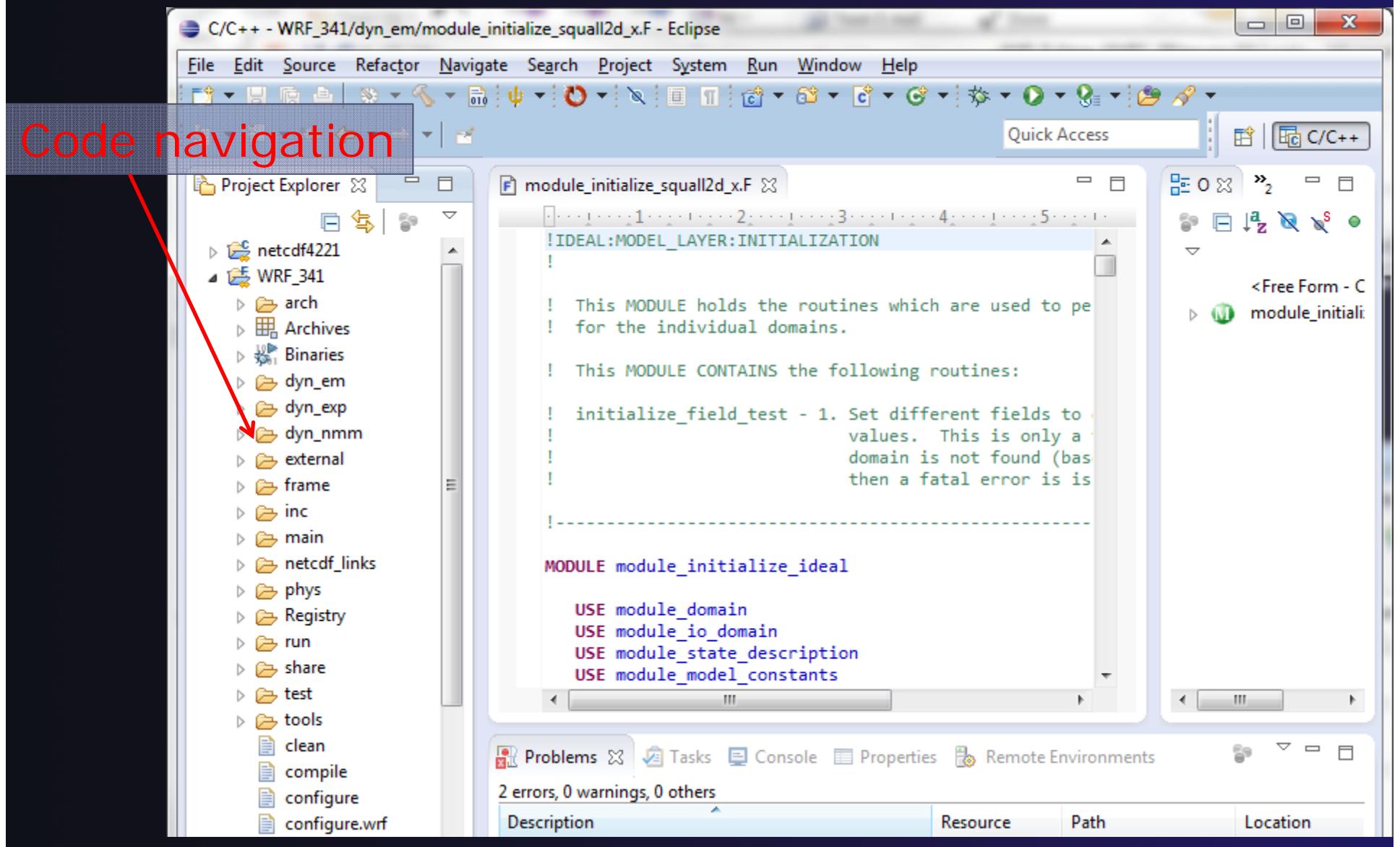
- ★ Some of the challenges
 - ★ Complex codes (eg WRF)
 - ★ Codes + HPC architectures can be daunting
 - ★ Adding user code not always easy
- ★ WRF: from http://wrf-model.org/PRESENTATIONS/2000_04_18_Klemp/sld007.htm



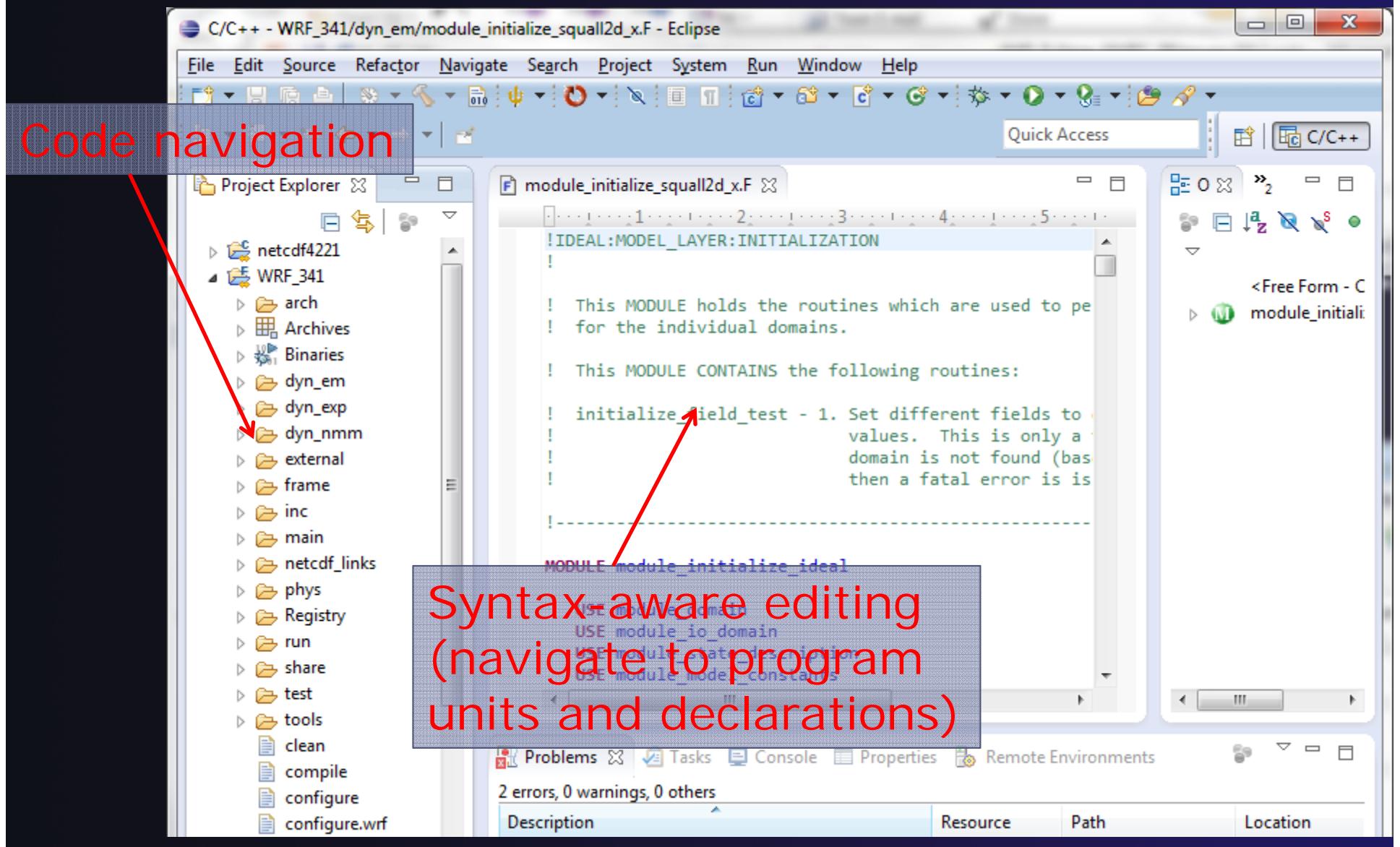
Navigating Weather Codes



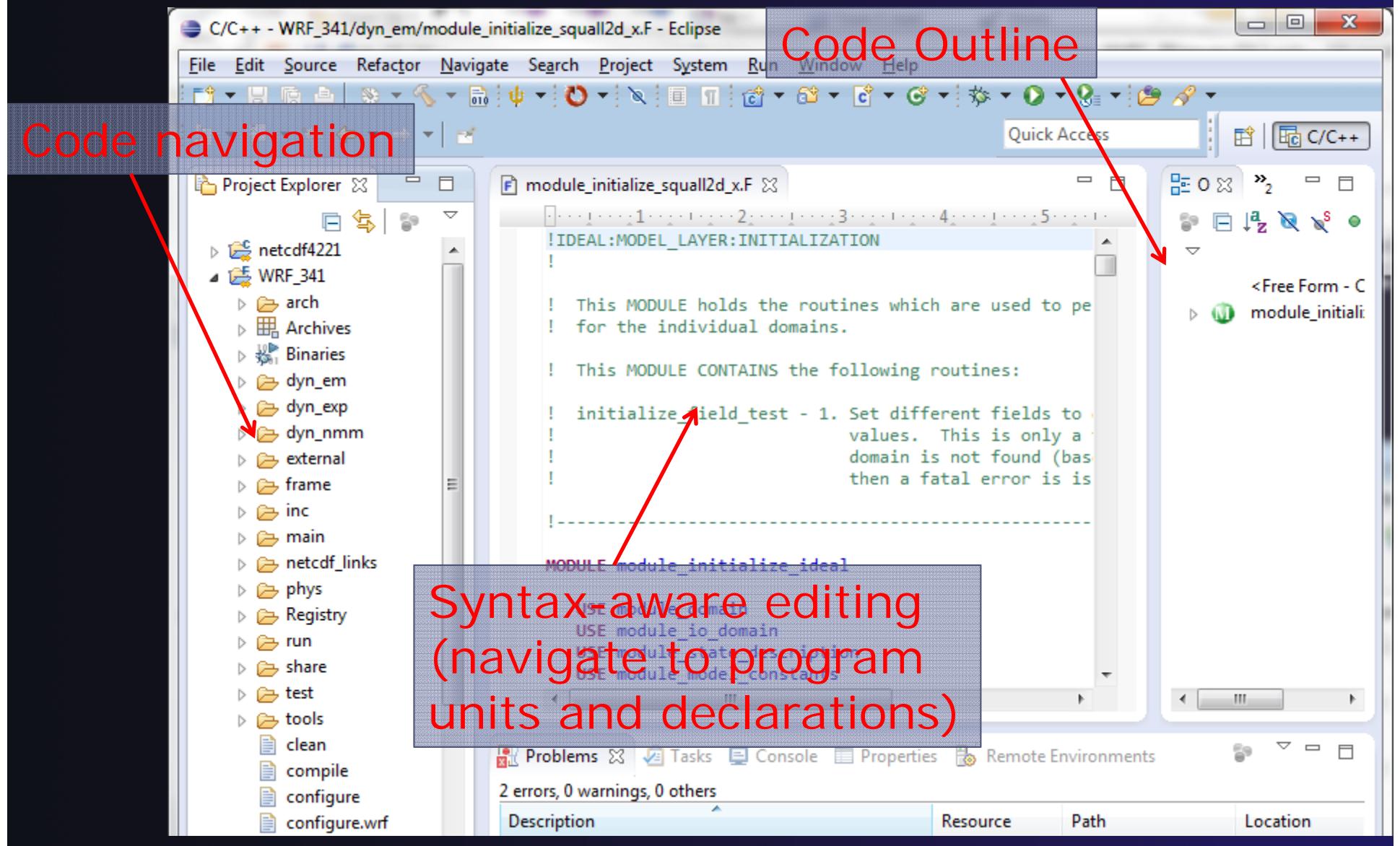
Navigating Weather Codes



Navigating Weather Codes



Navigating Weather Codes



Eclipse aiding in the WRF workflow...

- ★ May want to add a model output variable
- ★ Eclipse PTP makes it easy to navigate source, make changes
- ★ WRF Build
 - ★ Interactive “compile” script – use terminal within Eclipse to complete
 - ★ Configure Eclipse to drive “make” to iteratively build after modifications
- ★ WRF Run – can generate a run configuration for particular system, batch environment

Software Specialists enabling modeling projects

- ★ Need a wider array of software engineering tools
 - ★ Source repository
 - ★ Issue tracking
 - ★ Documentation
 - ★ Performance tuning...
- ★ Eclipse Parallel Tools can help with many of these concerns

Outline

- ★ Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- ★ Motivation for Workbench for High Performance Computing (WHPC)
 - ★ Improvements to Eclipse PTP
- ★ Software Engineering Practices Enabled by Eclipse PTP
 - ★ Code visibility
 - ★ Multi-system build management
 - ★ Performance tuning
 - ★ Source code control
 - ★ Issue Tracking
 - ★ Documentation
 - ★ Earth Science/Weather code example
- ★ Eclipse PTP Resources

Online Information

- ★ Information about PTP
 - ★ Main web site for downloads, documentation, etc.
 - ★ <http://eclipse.org/ptp>
 - ★ **Wiki for designs, planning, meetings, etc.**
 - ★ <http://wiki.eclipse.org/PTP>
 - ★ Articles and other documents
 - ★ <http://wiki.eclipse.org/PTP/articles>
- ★ Information about Photran
 - ★ Main web site for downloads, documentation, etc.
 - ★ <http://eclipse.org/photran>
 - ★ User's manuals
 - ★ <http://wiki.eclipse.org/PTP/photran/documentation>

Mailing Lists

♦ PTP Mailing lists

- ♦ Major announcements (new releases, etc.) - low volume
 - ♦ <http://dev.eclipse.org/mailman/listinfo/ptp-announce>
- ♦ User discussion and queries - medium volume
 - ♦ <http://dev.eclipse.org/mailman/listinfo/ptp-user>
- ♦ Developer discussions - high volume
 - ♦ <http://dev.eclipse.org/mailman/listinfo/ptp-dev>

♦ Photran Mailing lists

- ♦ User discussion and queries
 - ♦ <http://dev.eclipse.org/mailman/listinfo/photran>
- ♦ Developer discussions –
 - ♦ <http://dev.eclipse.org/mailman/listinfo/photran-dev>

Getting Involved

- ★ See <http://eclipse.org/ptp>
- ★ Read the developer documentation on the wiki
 - ★ <https://wiki.eclipse.org/PTP>
- ★ Join the mailing lists
- ★ Attend the monthly developer meetings
 - ★ Conf Call Monthly: Second Tuesday, 1:00 pm ET
 - ★ Details on the PTP wiki
- ★ Attend the monthly user meetings
 - ★ Teleconference Monthly
 - ★ Each 4th Wednesday, 2:00 pm ET
 - ★ Details on the PTP wiki – *webcast videos of previous menu demos available*

PTP will only succeed with your participation! 53



- ✧ Tutorial Thursday and Friday
- ✧ Eclipse and the Parallel Tools Platform
 - ✧ Thursday AM+PM, Friday AM
- ✧ Code Development in C/C++/Fortran
- ✧ Launch on a variety of platforms and schedulers
- ✧ Debugging with PTP Parallel Debugger
- ✧ Performance Optimization tools incl.
TAU