parallel tools platform

http://eclipse.org/ptp

Improving the Eclipse Parallel Tools Platform to Create an Effective Workbench for High Performance Computing

Jay Alameda

National Center for Supercomputing Applications

1st CHANGES Workshop, Jülich

5 September 2012

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), Marc Snir (U Illinois), and Allen Malony (U Oregon). Team members and senior personnel include Beth Tibbitts (IBM), Ralph Johnson (U Illinois), Albert Rossi (NCSA), Rick Kufrin (NCSA), Sameer Shende (U Oregon), Wyatt Spear (U Oregon), Bety Rodriguez-Milla (LSU), Brian Jewett (U Illinois), Galen Arnold (NCSA), and Rui Liu (NCSA)

Outline

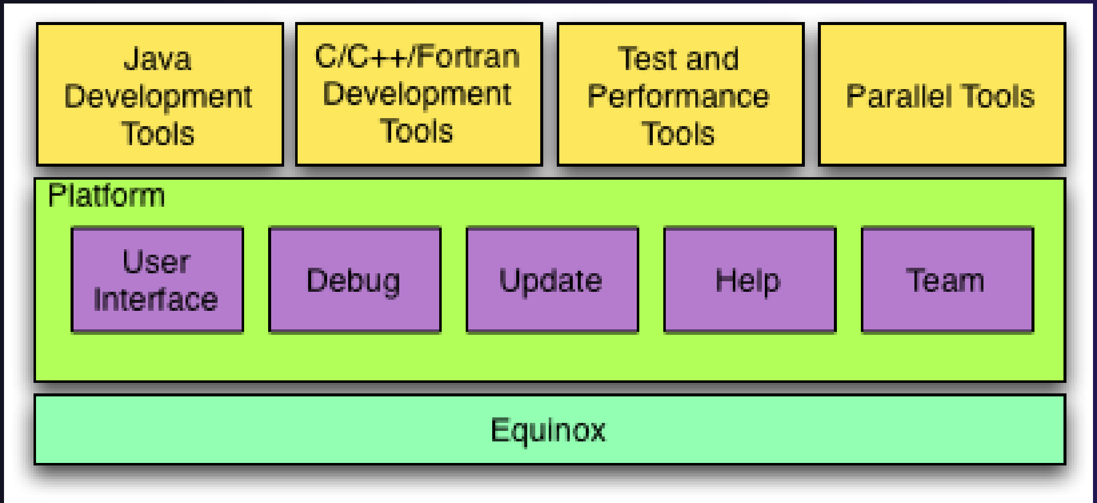
- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Overview of WHPC: NSF-funded SI2-SSI project to produce a productive and accessible development workbench using Eclipse PTP
  - Determining Requirements, Ensuring Impact
  - Improvements to Eclipse PTP
- Software Engineering Practices Enabled by Eclipse PTP
  - Code visibility
  - Multi-system build management
  - Performance tuning
  - Source code control
  - Issue Tracking
  - Documentation
- Eclipse PTP Resources

Outline

- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Overview of WHPC: NSF-funded SI2-SSI project to produce a productive and accessible development workbench using Eclipse PTP
  - Determining Requirements, Ensuring Impact
  - 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
- Eclipse PTP Resources

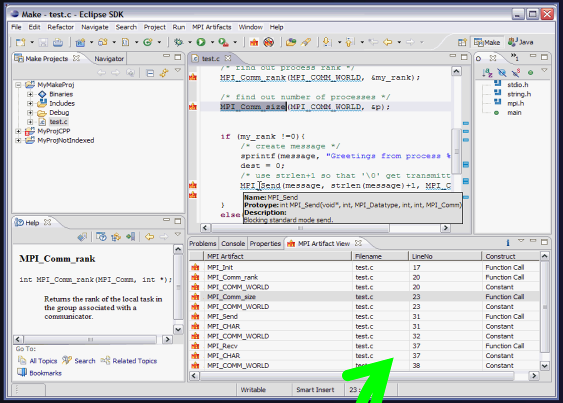
What is Eclipse?

- 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



Eclipse Parallel Tools Platform (PTP)

Coding & Analysis



```
    /* find out process rank */
    MPI_Comm_rank(MPI_COMM_WORLD, &my_rank);

    /* find out number of processes */
    MPI_Comm_size(MPI_COMM_WORLD, &p);


    if (my_rank !=0){
        /* create message */
        sprintf(message, "Greetings from process %
        dest = 0;
        /* use strlen+1 so that '\0' get transmitt
        MPI_Send(message, strlen(message)+1, MPI_C

    }
    else
```

Name: MPI_Send
Protoype: int MPI_Send(void*, int, MPI_Datatype, int, int, MPI_Comm)
Description:
Blocking standard mode send.

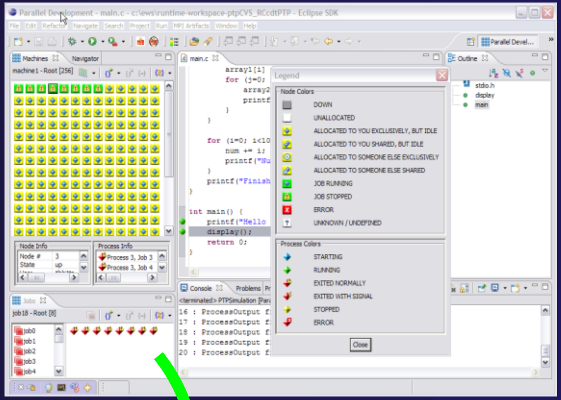
MPI_Comm_rank

int MPI_Comm_rank(MPI_Comm, int *);

Returns the rank of the local task in the group associated with a communicator.

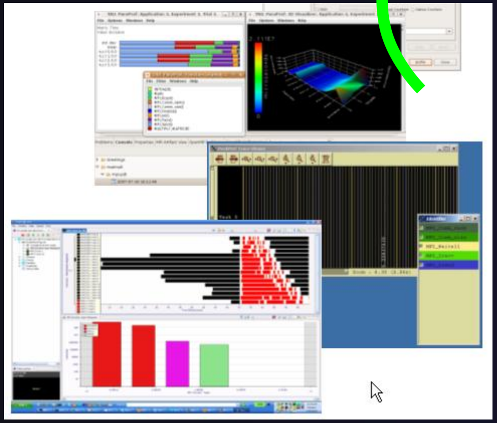
MPI Artifact	Filename	LineNo	Construct
MPI_Init	test.c	17	Function Call
MPI_Comm_rank	test.c	20	Function Call
MPI_COMM_WORLD	test.c	20	Constant
MPI_Comm_size	test.c	23	Function Call
MPI_COMM_WORLD	test.c	23	Constant
MPI_Send	test.c	31	Function Call
MPI_CHAR	test.c	31	Constant
MPI_COMM_WORLD	test.c	32	Constant
MPI_Recv	test.c	37	Function Call
MPI_CHAR	test.c	37	Constant
MPI_COMM_WORLD	test.c	38	Constant

Launching & Monitoring

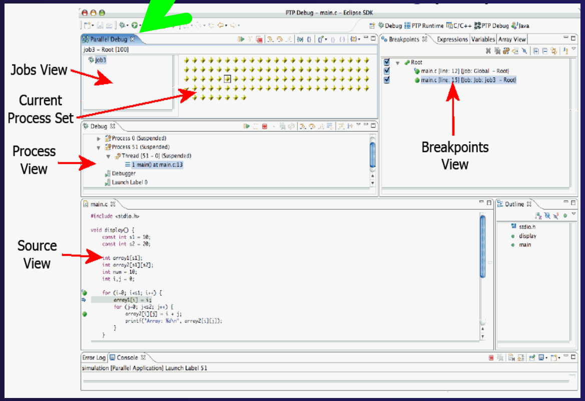


```
16 : ProcessOutput
17 : ProcessOutput
18 : ProcessOutput
19 : ProcessOutput
20 : ProcessOutput
```

Performance Tuning



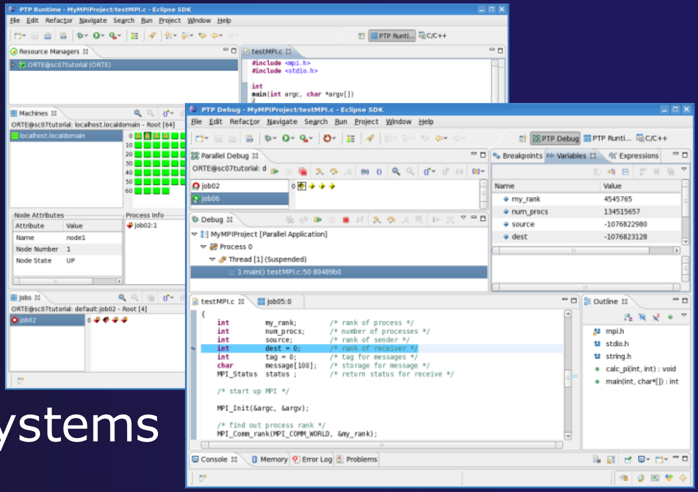
Debugging



5

Parallel Tools Platform (PTP)

- The Parallel Tools Platform aims to provide a highly integrated environment specifically designed for parallel application development
- Features include:
  - An integrated development environment (IDE) that supports a wide range of parallel architectures and runtime systems
  - A scalable parallel debugger
  - Parallel programming tools (MPI, OpenMP, UPC, etc.)
  - Support for the integration of parallel tools
  - An environment that simplifies the end-user interaction with parallel systems
- http://www.eclipse.org/ptp


PTP Runtime - MyMPIProject/testMPI.c - Eclipse SDK
File Edit Refactor Navigate Search Run Project Window Help
PTP Runti... C/C++
Resource Managers
ORTE@sc07tutorial (ORTE)
Machines
ORTE@sc07tutorial: localhost.localdomain - Root [64]
localhost.localdomain
0
10
20
30
40
50
60
Process Info
job02:1
Jobs
ORTE@sc07tutorial: default:job02 - Root [4]
job02
0
PTP Debug - MyMPIProject/testMPI.c - Eclipse SDK
File Edit Refactor Navigate Search Run Project Window Help
PTP Debug PTP Runti... C/C++
Parallel Debug
ORTE@sc07tutorial: d
job02
job06
0
Debug
MyMPIProject [Parallel Application]
Process 0
Thread [1] (Suspended)
1 main() testMPI.c:50 80489b0
Breakpoints Variables Expressions
testMPI.c job05:0
Outline
mpi.h
stdio.h
string.h
calc_pi(int, int) : void
main(int, char*[]) : int
Console Memory Error Log Problems

Node Attributes

Attribute	Value
Name	node1
Node Number	1
Node State	UP

testMPI.c

```
#include <mpi.h>
#include <stdio.h>

int
main(int argc, char *argv[])
```

Name	Value
my_rank	4545765
num_procs	134515657
source	-1076822980
dest	-1076823128

```
{
    int         my_rank;        /* rank of process */
    int         num_procs;      /* number of processes */
    int         source;         /* rank of sender */
    int         dest = 0;       /* rank of receiver */
    int         tag = 0;        /* tag for messages */
    char        message[100];   /* storage for message */
    MPI_Status  status ;        /* return status for receive */

    /* start up MPI */

    MPI_Init(&argc, &argv);

    /* find out process rank */
    MPI_Comm_rank(MPI_COMM_WORLD, &my_rank);
```

Outline

- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Overview of WHPC: NSF-funded SI2-SSI project to produce a productive and accessible development workbench using Eclipse PTP
  - Determining Requirements, Ensuring Impact
  - 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
- Eclipse PTP Resources

Why 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
    - Edit, build, test, debug, maintain, for maximum developer productivity
    - Remote development, batch execution mandatory
  - Track, store, search, browse code artifact provenance
  - Share tool functionality through an integration framework
  - Maintain tool identity
    - Provides for independent tool development pathways and funding

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, …)
- For PTP to become a WHPC need to:
  - Cultivate community of users
  - Make substantial improvements to PTP around two themes:
    - Improving usability
    - Improving productivity

Outline

- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Overview of WHPC: NSF-funded SI2-SSI project to produce a productive and accessible development workbench using Eclipse PTP
  - Determining Requirements, Ensuring Impact
  - 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
- Eclipse PTP Resources

Requirements and Impact

- Application-centric approach
  - Use real application codes, with PTP, on production computational resources
    - Identify specific goals to accomplish with each application
    - Use Eclipse PTP to accomplish the goals
    - Identify shortcomings in Eclipse PTP that need to be rectified for Eclipse PTP to be effective with that application workplan
  - This is part of our project team’s responsibility
  - Work with application community and learn from their experience with Eclipse PTP

Requirements and Impact (2)

- Application-centric approach
  - Work with application community and learn from their experience with Eclipse PTP
    - Bridge to TeraGrid and (now) XSEDE Advanced User Support
    - Work with targeted organizations to assist with adoption of PTP
    - Monthly user calls
    - Annual user group meeting
    - Hands on tutorials
    - Conference Birds of a Feather

Outline

- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Overview of WHPC: NSF-funded SI2-SSI project to produce a productive and accessible development workbench using Eclipse PTP
  - Determining Requirements, Ensuring Impact
  - 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
- Eclipse PTP Resources

Improvements

- Work within Eclipse release cycle
  - Major (API-breaking) improvements with coordinated June release
    - Last major release Eclipse 4.2 “Juno” released June 27, 2012
  - Minor enhancements and bug-fixes with two coordinated service releases in September and February
    - Eclipse 4.2 SR1 due out Sept 26, 2012
- Foci of improvements
  - Improve usability
  - Improve productivity

Improve Usability

- Remote support and scalability enhancements
  - Broaden support of remote capabilities to full PTP
  - Provide for easy platform configuration management
  - Provide additional remote features
    - Automatic remote service deployment
    - Multiple authentication mechanism
    - Support wide range of resource managers
    - Full remote debug support

Improve Usability

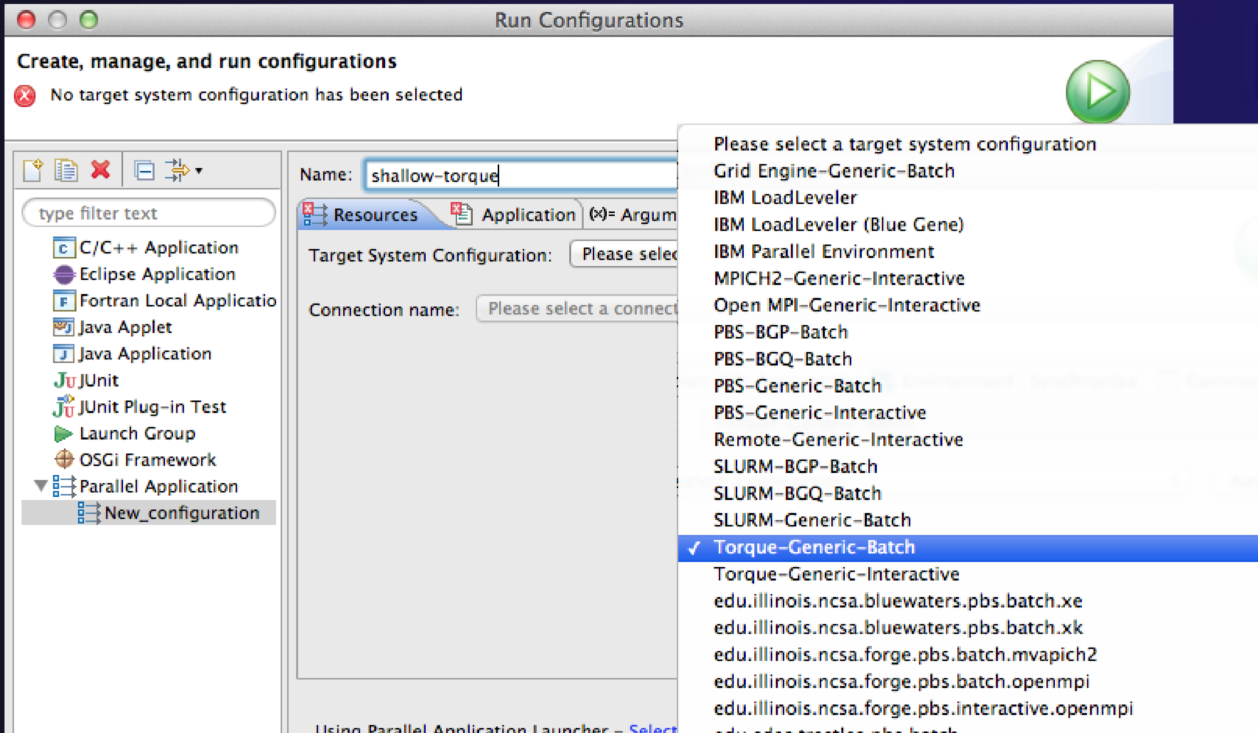
- Integration with other tools
  - Improve External Tools Framework (ETFw)
    - Full remote support
    - Integration of tool output with Eclipse views
- Improve and broaden parallel paradigm support
  - Driven by user needs and feedback

Improve Productivity

- Provide support for performance driven refactoring
- Track source and executable code provenance

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



Run Configurations

Create, manage, and run configurations

No target system configuration has been selected

Name: shallow-torque

Target System Configuration: Please selec

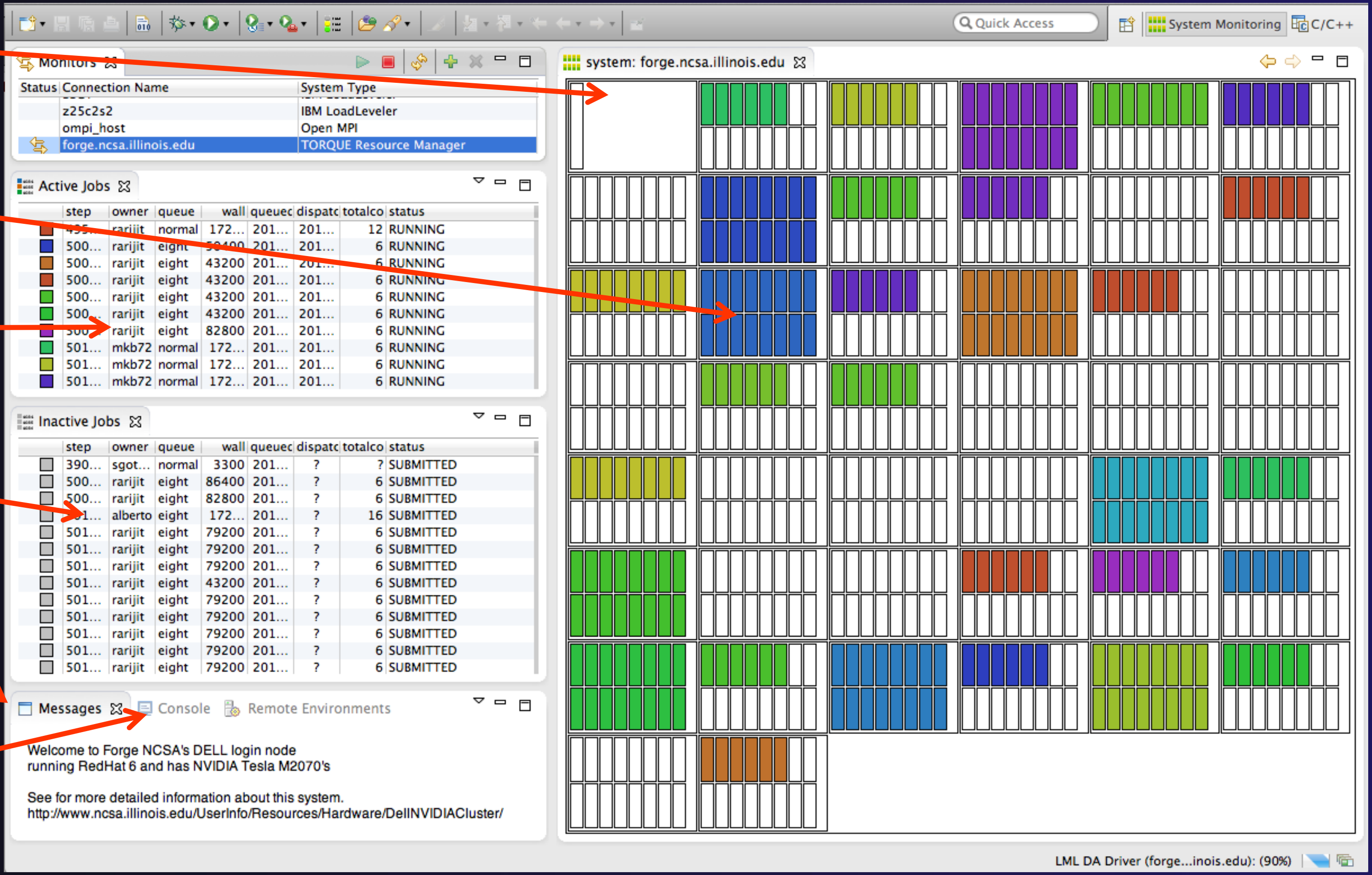
Connection name: Please select a connect

Using Parallel Application Launcher – Select

Please select a target system configuration
Grid Engine-Generic-Batch
IBM LoadLeveler
IBM LoadLeveler (Blue Gene)
IBM Parallel Environment
MPICH2-Generic-Interactive
Open MPI-Generic-Interactive
PBS-BGP-Batch
PBS-BGQ-Batch
PBS-Generic-Batch
PBS-Generic-Interactive
Remote-Generic-Interactive
SLURM-BGP-Batch
SLURM-BGQ-Batch
SLURM-Generic-Batch
✓ Torque-Generic-Batch
Torque-Generic-Interactive
edu.illinois.ncsa.bluewaters.pbs.batch.xe
edu.illinois.ncsa.bluewaters.pbs.batch.xk
edu.illinois.ncsa.forge.pbs.batch.mvapich2
edu.illinois.ncsa.forge.pbs.batch.openmpi
edu.illinois.ncsa.forge.pbs.interactive.openmpi
edu.sdsc.trestles.pbs.batch

Scalable System Monitoring

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


Quick Access
System Monitoring
C/C++
system: forge.ncsa.illinois.edu
LML DA Driver (forge...inois.edu): (90%)

Monitors

Status	Connection Name	System Type
	z25c2s2	IBM LoadLeveler
	ompi_host	Open MPI
	forge.ncsa.illinois.edu	TORQUE Resource Manager

Active Jobs

	step	owner	queue	wall	queuec	dispatc	totalco	status
🟥	495...	rarijit	normal	172...	201...	201...	12	RUNNING
🟦	500...	rarijit	eight	50400	201...	201...	6	RUNNING
🟧	500...	rarijit	eight	43200	201...	201...	6	RUNNING
🟥	500...	rarijit	eight	43200	201...	201...	6	RUNNING
🟩	500...	rarijit	eight	43200	201...	201...	6	RUNNING
🟩	500...	rarijit	eight	43200	201...	201...	6	RUNNING
🟪	500...	rarijit	eight	82800	201...	201...	6	RUNNING
🟩	501...	mkb72	normal	172...	201...	201...	6	RUNNING
🟨	501...	mkb72	normal	172...	201...	201...	6	RUNNING
🟪	501...	mkb72	normal	172...	201...	201...	6	RUNNING

Inactive Jobs

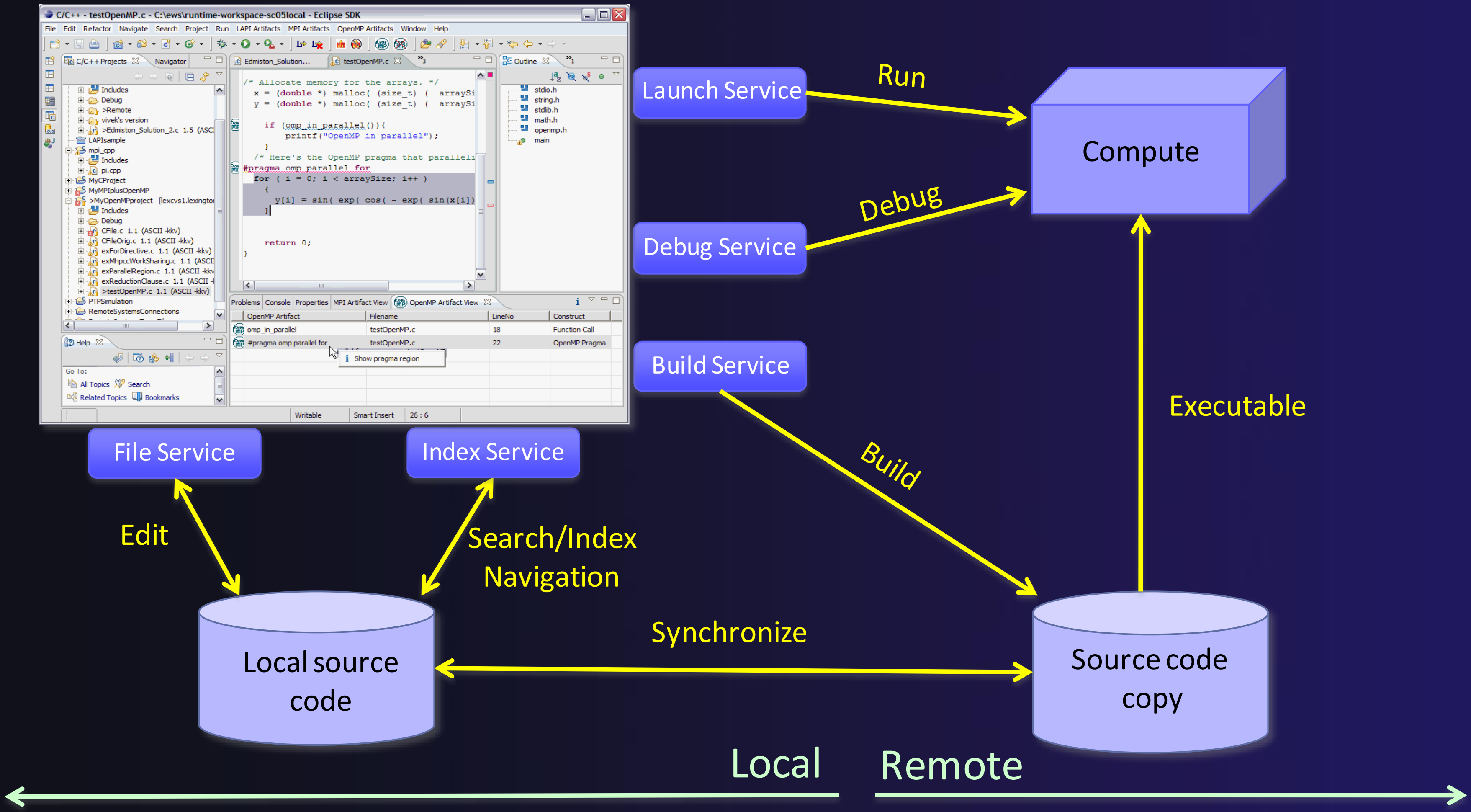
	step	owner	queue	wall	queuec	dispatc	totalco	status
⬜	390...	sgot...	normal	3300	201...	?	?	SUBMITTED
⬜	500...	rarijit	eight	86400	201...	?	6	SUBMITTED
⬜	500...	rarijit	eight	82800	201...	?	6	SUBMITTED
⬜	501...	alberto	eight	172...	201...	?	16	SUBMITTED
⬜	501...	rarijit	eight	79200	201...	?	6	SUBMITTED
⬜	501...	rarijit	eight	79200	201...	?	6	SUBMITTED
⬜	501...	rarijit	eight	79200	201...	?	6	SUBMITTED
⬜	501...	rarijit	eight	43200	201...	?	6	SUBMITTED
⬜	501...	rarijit	eight	79200	201...	?	6	SUBMITTED
⬜	501...	rarijit	eight	79200	201...	?	6	SUBMITTED
⬜	501...	rarijit	eight	79200	201...	?	6	SUBMITTED
⬜	501...	rarijit	eight	79200	201...	?	6	SUBMITTED
⬜	501...	rarijit	eight	79200	201...	?	6	SUBMITTED

Messages Console Remote Environments

Welcome to Forge NCSA's DELL login node
running RedHat 6 and has NVIDIA Tesla M2070's

See for more detailed information about this system.
http://www.ncsa.illinois.edu/UserInfo/Resources/Hardware/DellNVIDIACluster/

Synchronized Projects



C/C++ - testOpenMP.c - C:\ews\runtime-workspace-sc05local - Eclipse SDK

```
Includes
Debug
>Remote
vivek's version
>Edmiston_Solution_2.c  1.5  (ASC
LAPIsample
mpi_cpp
  Includes
  pi.cpp
MyCProject
MyMPIplusOpenMP
>MyOpenMPproject  [lexcvs1.lexingto
  Includes
  Debug
  CFile.c  1.1  (ASCII -kkv)
  CFileOrig.c  1.1  (ASCII -kkv)
  exForDirective.c  1.1  (ASCII -kkv)
  exMhpccWorkSharing.c  1.1  (ASCI
  exParallelRegion.c  1.1  (ASCII -kkv
  exReductionClause.c  1.1  (ASCII -
  >testOpenMP.c  1.1  (ASCII -kkv)
PTPSimulation
RemoteSystemsConnections
```

```
/* Allocate memory for the arrays. */
  x = (double *) malloc( (size_t) (  arraySi
  y = (double *) malloc( (size_t) (  arraySi

    if (omp_in_parallel()){
        printf("OpenMP in parallel");
    }
  /* Here's the OpenMP pragma that paralleli
#pragma omp parallel for
  for ( i = 0; i < arraySize; i++ )
    {
      y[i] = sin( exp( cos( - exp( sin(x[i])
    }


    return 0;
}
```

OpenMP Artifact	Filename	LineNo	Construct
omp_in_parallel	testOpenMP.c	18	Function Call
#pragma omp parallel for	testOpenMP.c	22	OpenMP Pragma

Show pragma region

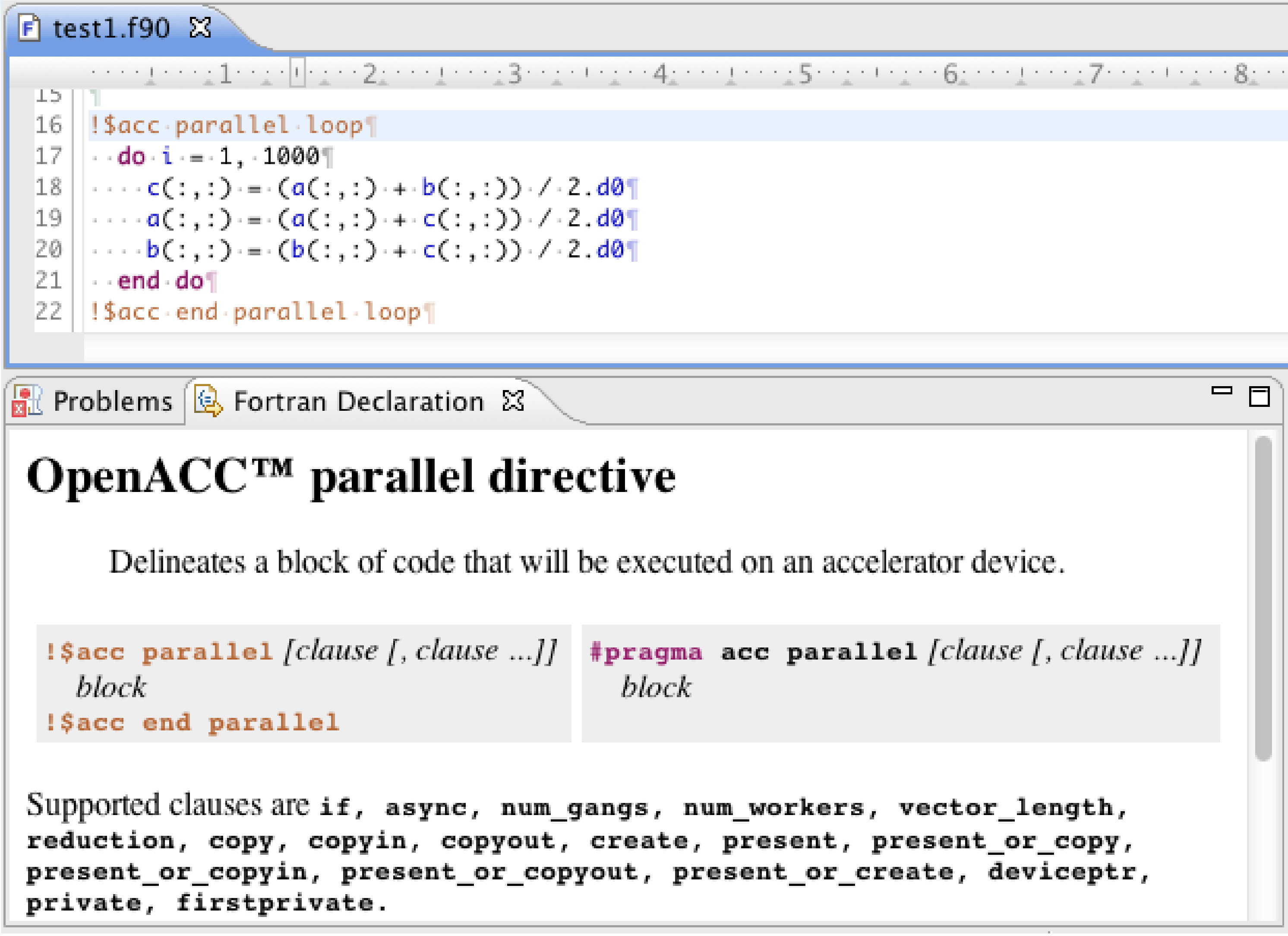
Writable | Smart Insert | 26 : 6

Blue Waters Enhancements

- **Blue Waters:** Cray XE6/XK7 at NCSA
- PTP did not work with Crays “out of the box”
  - Could not submit jobs with appropriate aprun options
  - Could not monitor status of compute nodes
  - Could not set environment modules for build
  - Did not recognize Cray, PGI compilers’ errors messages
  - Did not support OpenACC
  - …
- Less than 6 months to fix these for PTP 6.0 (!)

Integrated OpenACC documentation and PLDT support

(added for BW)



```
!$acc parallel loop
  do i = 1, 1000
    c(:,:) = (a(:,:) + b(:,:)) / 2.d0
    a(:,:) = (a(:,:) + c(:,:)) / 2.d0
    b(:,:) = (b(:,:) + c(:,:)) / 2.d0
  end do
!$acc end parallel loop
```

OpenACC™ parallel directive

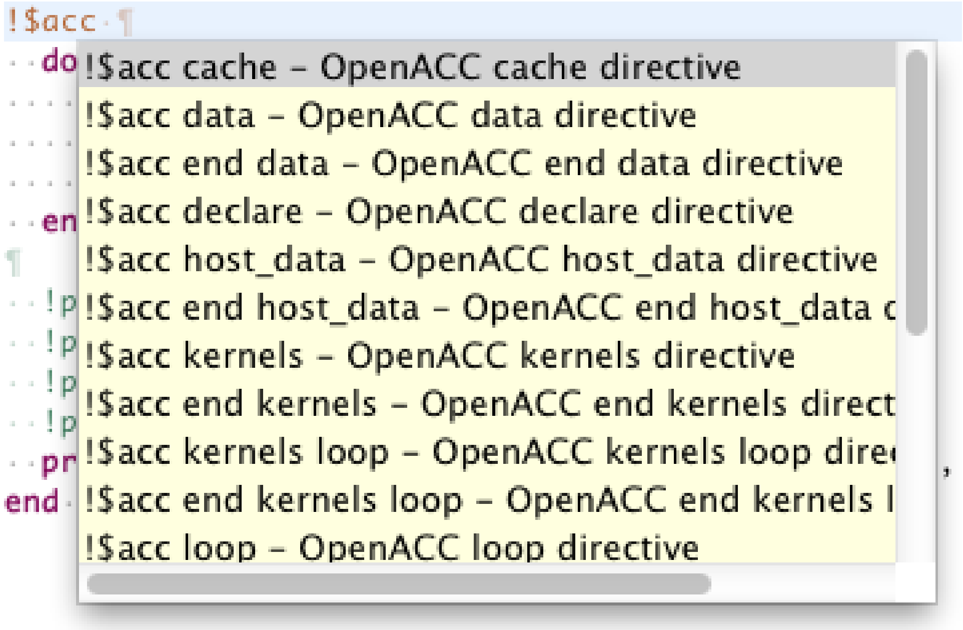
Delineates a block of code that will be executed on an accelerator device.

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

```
#pragma acc parallel [clause [, clause ...]]
  block
```

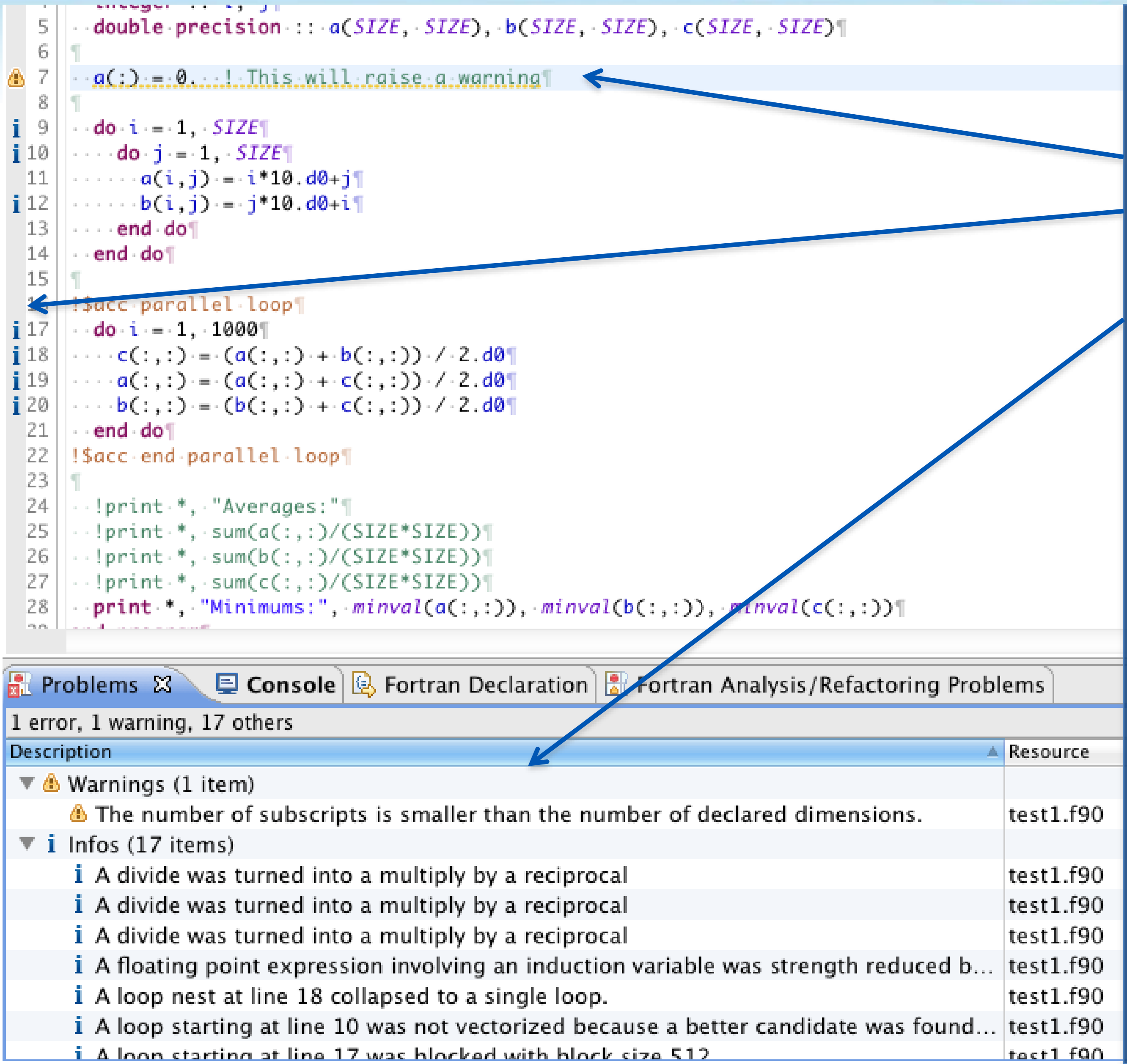
Supported clauses are **if**, **async**, **num_gangs**, **num_workers**, **vector_length**, **reduction**, **copy**, **copyin**, **copyout**, **create**, **present**, **present_or_copy**, **present_or_copyin**, **present_or_copyout**, **present_or_create**, **deviceptr**, **private**, **firstprivate**.

Documentation also available for MPI, OpenMP



```
!$acc
!$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
!$acc end host_data – OpenACC end host_data c
!$acc kernels – OpenACC kernels directive
!$acc end kernels – OpenACC end kernels direct
!$acc kernels loop – OpenACC kernels loop dire
!$acc end kernels loop – OpenACC end kernels l
!$acc loop – OpenACC loop directive
```

Code completion for OpenACC directives (added for BW)



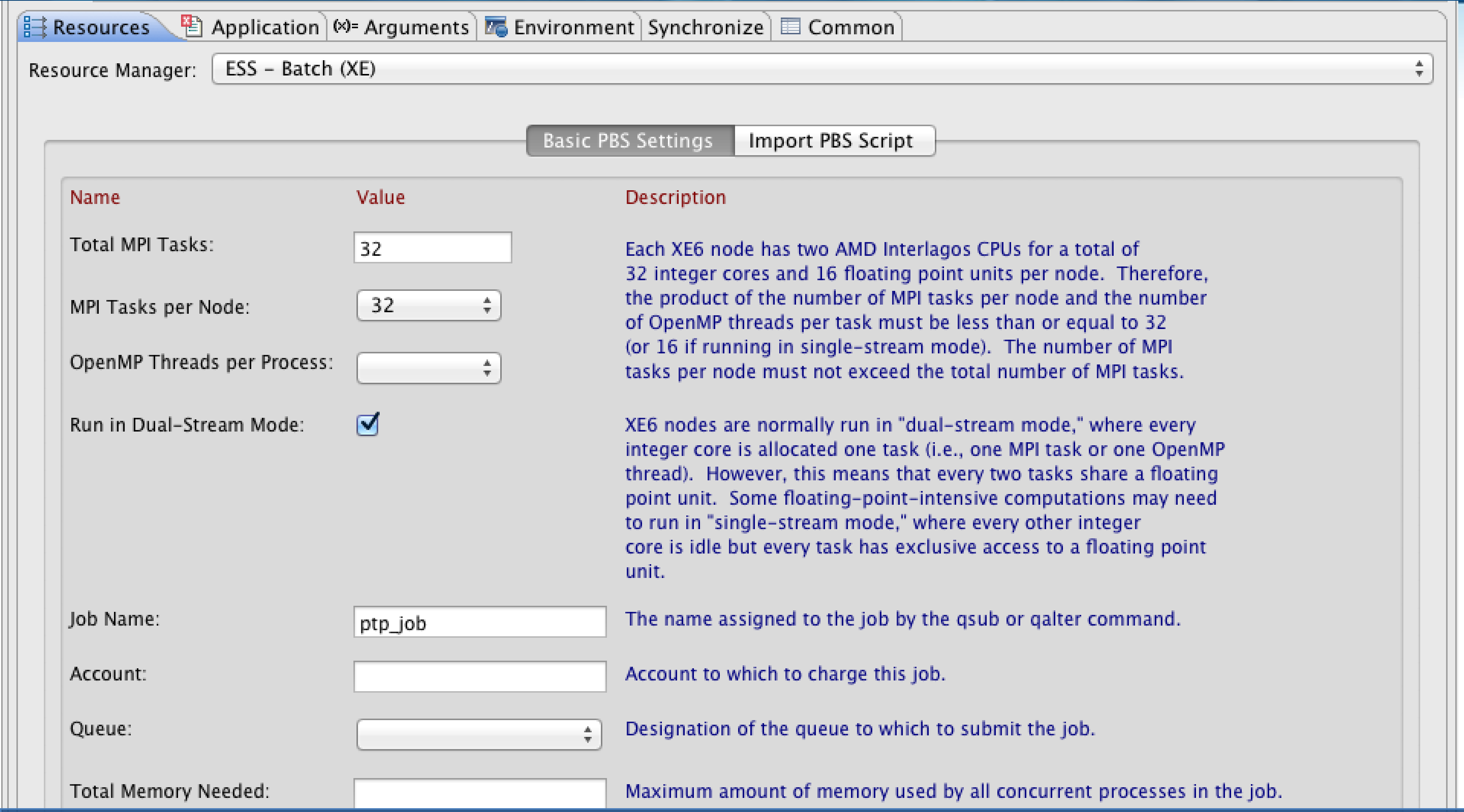
```
 4    integer :: i, j
 5    double precision :: a(SIZE, SIZE), b(SIZE, SIZE), c(SIZE, SIZE)
 6
 7    a(:) = 0.  ! This will raise a warning
 8
 9    do i = 1, SIZE
10      do j = 1, SIZE
11        a(i,j) = i*10.d0+j
12        b(i,j) = j*10.d0+i
13      end do
14    end do
15
16  !$acc parallel loop
17    do i = 1, 1000
18      c(:,:) = (a(:,:) + b(:,:)) / 2.d0
19      a(:,:) = (a(:,:) + c(:,:)) / 2.d0
20      b(:,:) = (b(:,:) + c(:,:)) / 2.d0
21    end do
22  !$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 *, "Minimums:", minval(a(:,:)), minval(b(:,:)), minval(c(:,:))
29  end program
```

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
Infos (17 items)	
A divide was turned into a multiply by a reciprocal	test1.f90
A divide was turned into a multiply by a reciprocal	test1.f90
A divide was turned into a multiply by a reciprocal	test1.f90
A floating point expression involving an induction variable was strength reduced b...	test1.f90
A loop nest at line 18 collapsed to a single loop.	test1.f90
A loop starting at line 10 was not vectorized because a better candidate was found...	test1.f90
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)



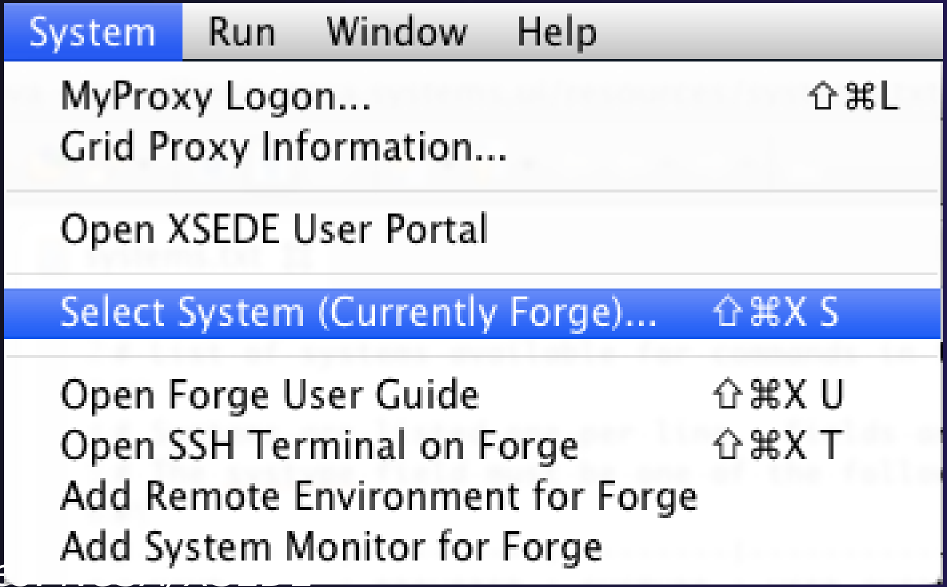
Resource Manager: ESS - Batch (XE)

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:	☑	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.
Wallclock Time:	00:30:00	Maximum amount of real time during which the job can be in the running state.

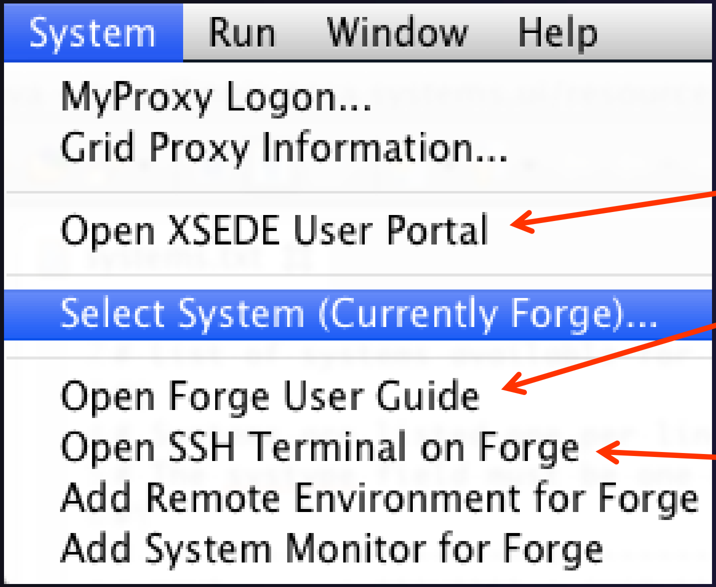
Graphical interface for launching a job (customized for BW)

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
- Contribute a **System** menu to the menu bar with XSEDE- and NCSA-specific commands


System
Run
Window
Help
MyProxy Logon... ⇧⌘L
Grid Proxy Information...
Open XSEDE User Portal
Select System (Currently Forge)... ⇧⌘X S
Open Forge User Guide ⇧⌘X U
Open SSH Terminal on Forge ⇧⌘X T
Add Remote Environment for Forge
Add System Monitor for Forge

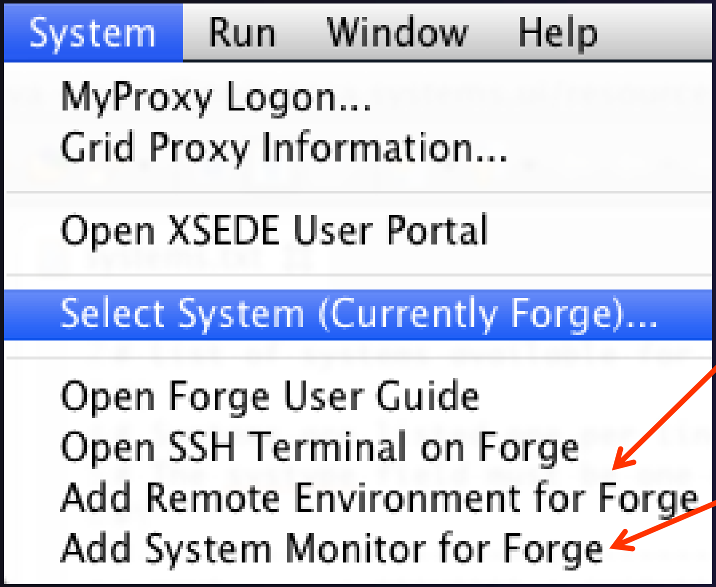
System Menu


System Run Window Help
MyProxy Logon...
Grid Proxy Information...
Open XSEDE User Portal
Select System (Currently Forge)...
Open Forge User Guide
Open SSH Terminal on Forge
Add Remote Environment for Forge
Add System Monitor for Forge

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

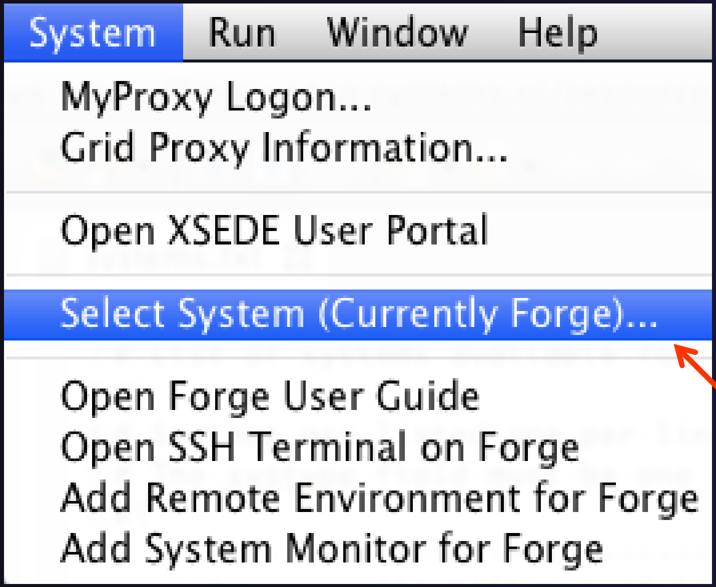
Eclipse-integrated SSH terminals are provided by the Remote System Explorer (RSE), one of the features that is included in the Eclipse for Parallel Application Developers package.

System Menu


System
Run
Window
Help
MyProxy Logon...
Grid Proxy Information...
Open XSEDE User Portal
Select System (Currently Forge)...
Open Forge User Guide
Open SSH Terminal on Forge
Add Remote Environment for Forge
Add System Monitor for Forge

- 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

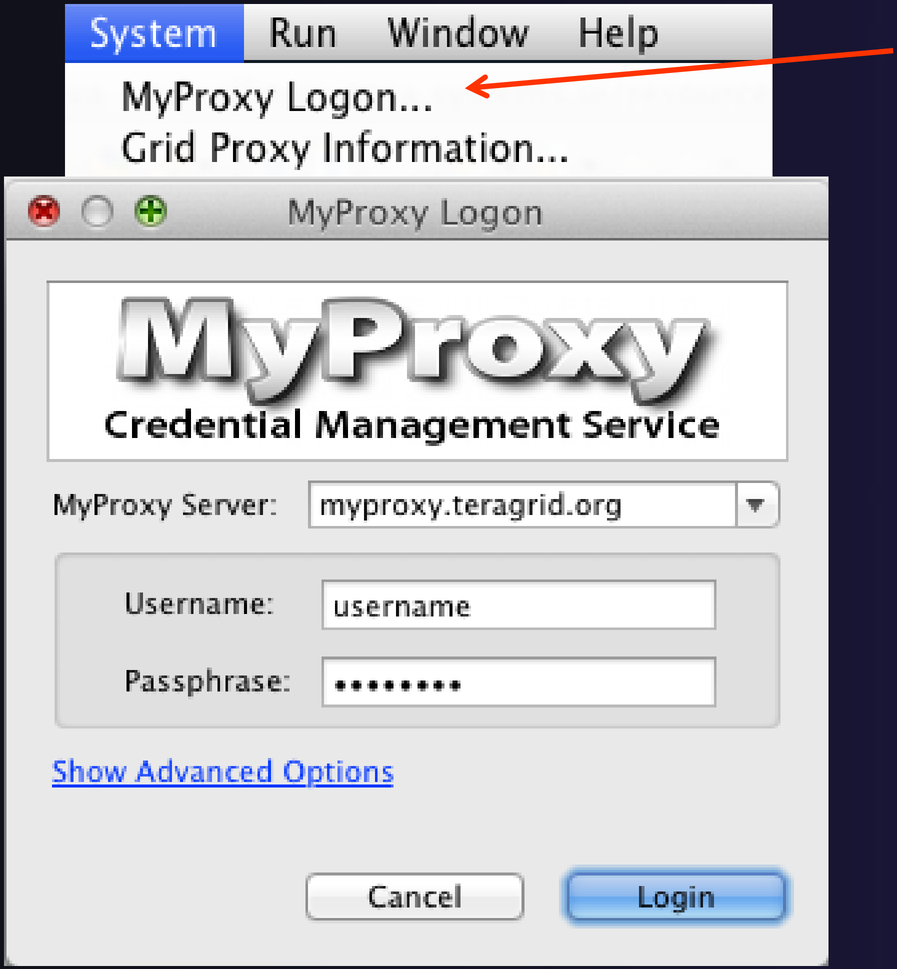
System Menu



System Run Window Help
MyProxy Logon...
Grid Proxy Information...
Open XSEDE User Portal
Select System (Currently Forge)...
Open Forge User Guide
Open SSH Terminal on Forge
Add Remote Environment for Forge
Add System Monitor for Forge

- 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 Server: myproxy.teragrid.org

Username: username

Passphrase: ••••••••

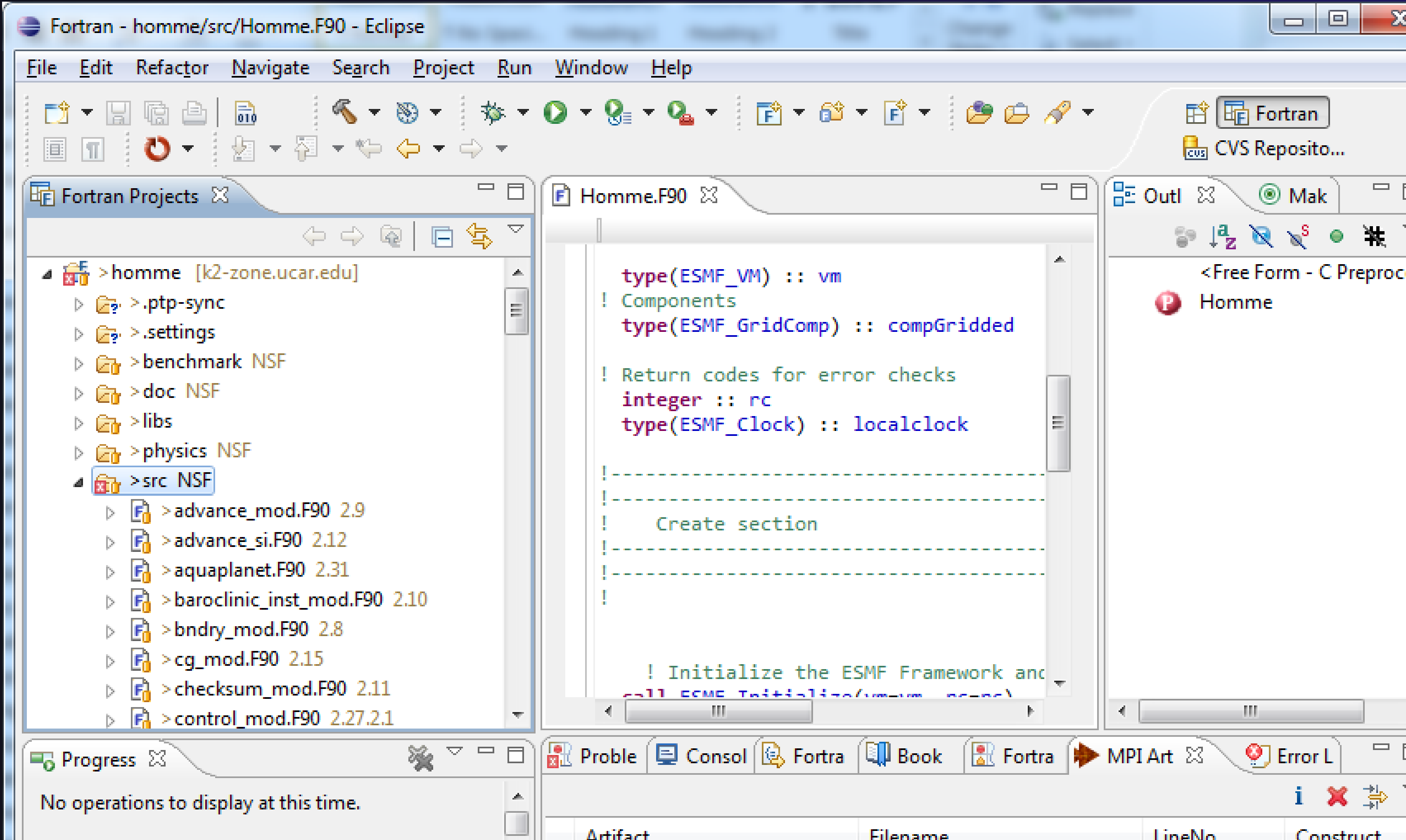
- **MyProxy Logon** allows you to authenticate with a MyProxy server
  - Often **myproxy.teragrid.org**
- It stores a "credential," which is usually valid for 12 hours
- During these 12 hours, SSH connections to XSEDE resources will not require a password; they can use the stored credential
  - However,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

Outline

- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Overview of WHPC: NSF-funded SI2-SSI project to produce a productive and accessible development workbench using Eclipse PTP
  - Determining Requirements, Ensuring Impact
  - 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
- Eclipse PTP Resources

Software Engineering

- Code Visibility



>homme [k2-zone.ucar.edu]
- >.ptp-sync
- >.settings
- >benchmark NSF
- >doc NSF
- >libs
- >physics NSF
- >src NSF
  - >advance_mod.F90 2.9
  - >advance_si.F90 2.12
  - >aquaplanet.F90 2.31
  - >baroclinic_inst_mod.F90 2.10
  - >bndry_mod.F90 2.8
  - >cg_mod.F90 2.15
  - >checksum_mod.F90 2.11
  - >control_mod.F90 2.27.2.1

```
  type(ESMF_VM) :: vm
! Components
  type(ESMF_GridComp) :: compGridded

! Return codes for error checks
  integer :: rc
  type(ESMF_Clock) :: localclock

!-----------------------------------------
!-----------------------------------------
!    Create section
!-----------------------------------------
!-----------------------------------------
!


    ! Initialize the ESMF Framework anc
  call ESMF_Initialize(vm=vm, rc=rc)
```

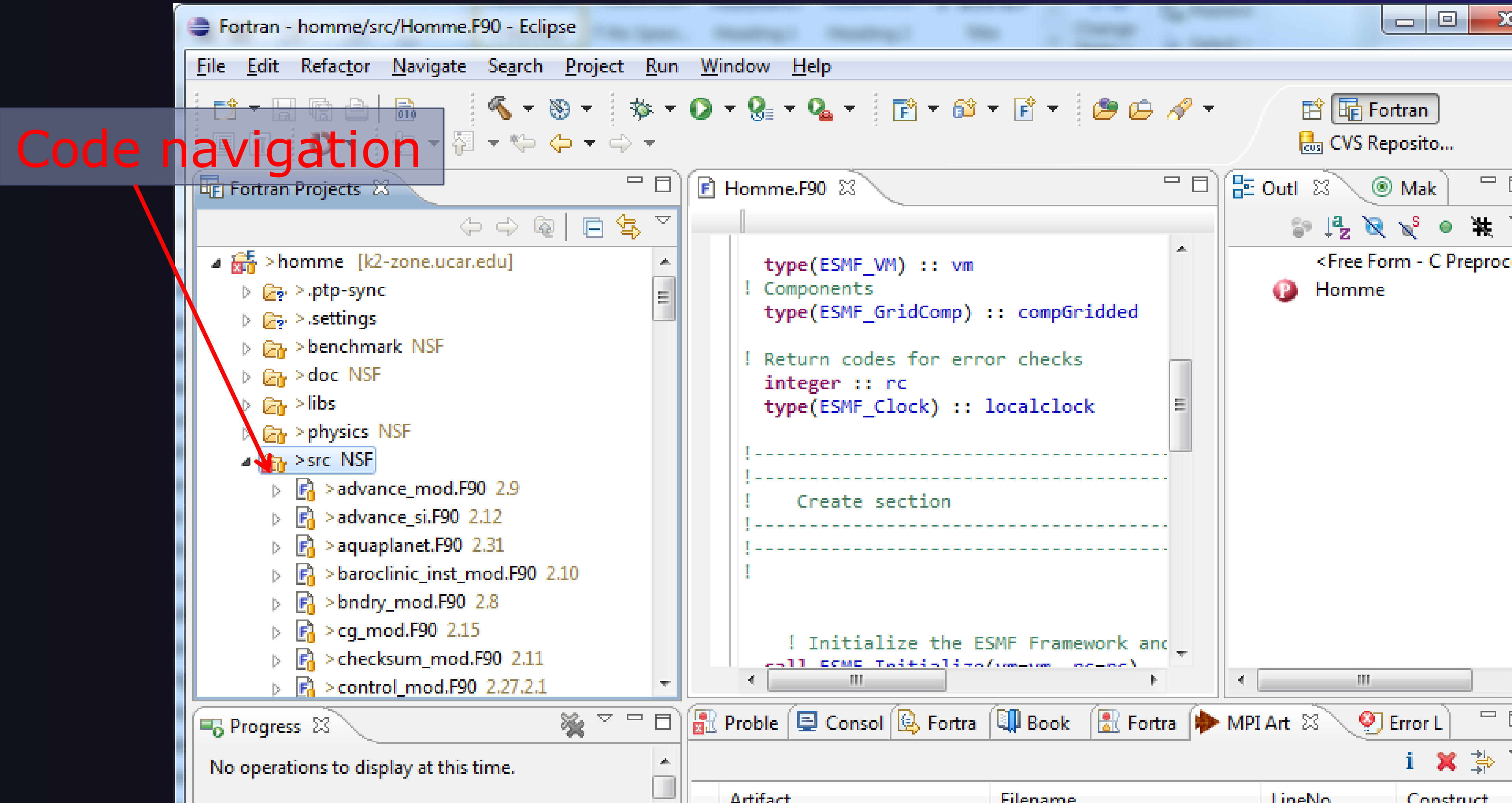
<Free Form - C Preproc
Homme

No operations to display at this time.

Artifact	Filename	LineNo	Construct

Software Engineering

- Code Visibility


Code navigation

Fortran - homme/src/Homme.F90 - Eclipse

- >homme [k2-zone.ucar.edu]
  - >.ptp-sync
  - >.settings
  - >benchmark NSF
  - >doc NSF
  - >libs
  - >physics NSF
  - >src NSF
    - >advance_mod.F90 2.9
    - >advance_si.F90 2.12
    - >aquaplanet.F90 2.31
    - >baroclinic_inst_mod.F90 2.10
    - >bndry_mod.F90 2.8
    - >cg_mod.F90 2.15
    - >checksum_mod.F90 2.11
    - >control_mod.F90 2.27.2.1

```
  type(ESMF_VM) :: vm
! Components
  type(ESMF_GridComp) :: compGridded

! Return codes for error checks
  integer :: rc
  type(ESMF_Clock) :: localclock

!-----------------------------------
!-----------------------------------
!    Create section
!-----------------------------------
!-----------------------------------
!


    ! Initialize the ESMF Framework anc
  call ESMF_Initialize(vm=vm, rc=rc)
```

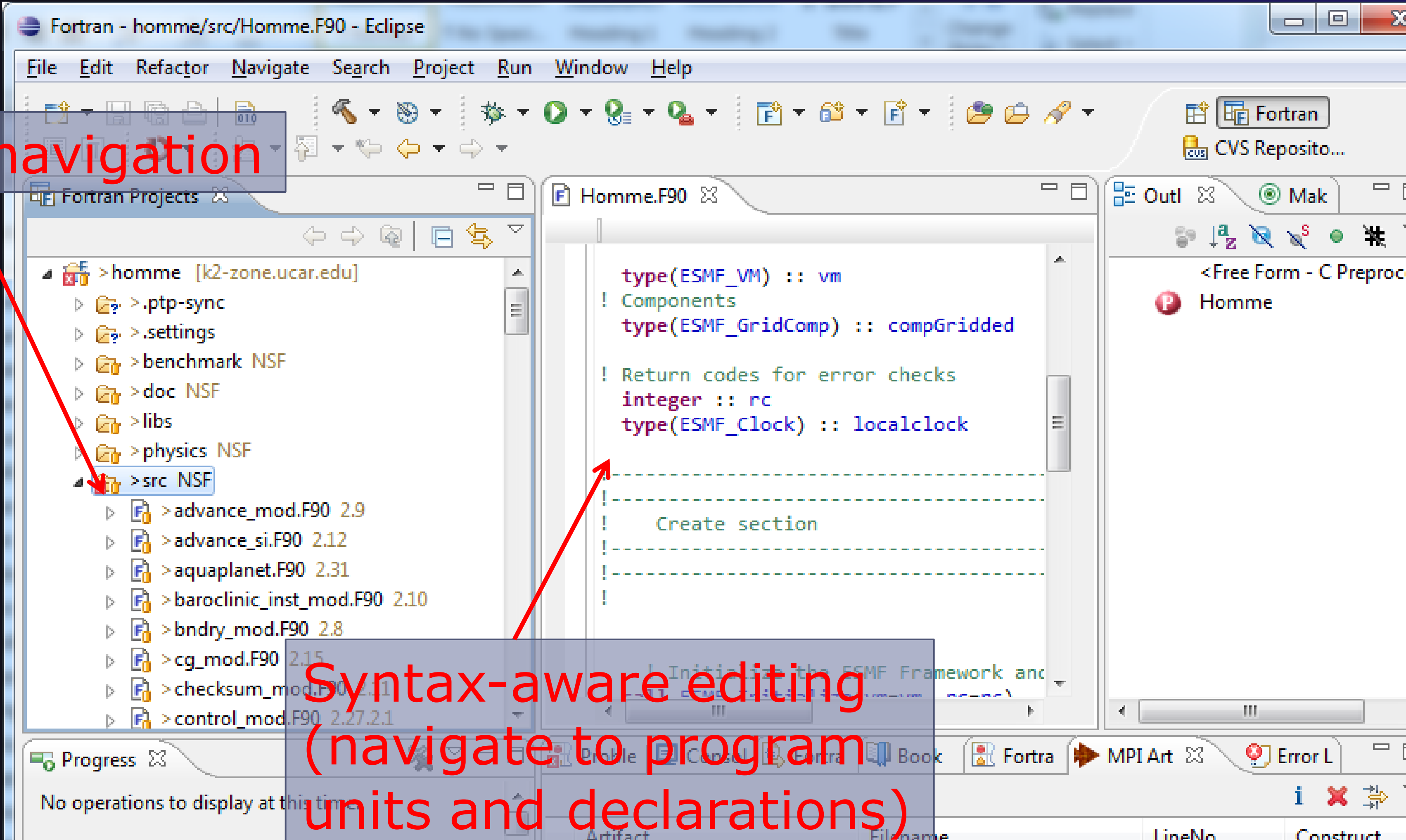
<Free Form - C Preproc

Homme

No operations to display at this time.

Software Engineering

- Code Visibility



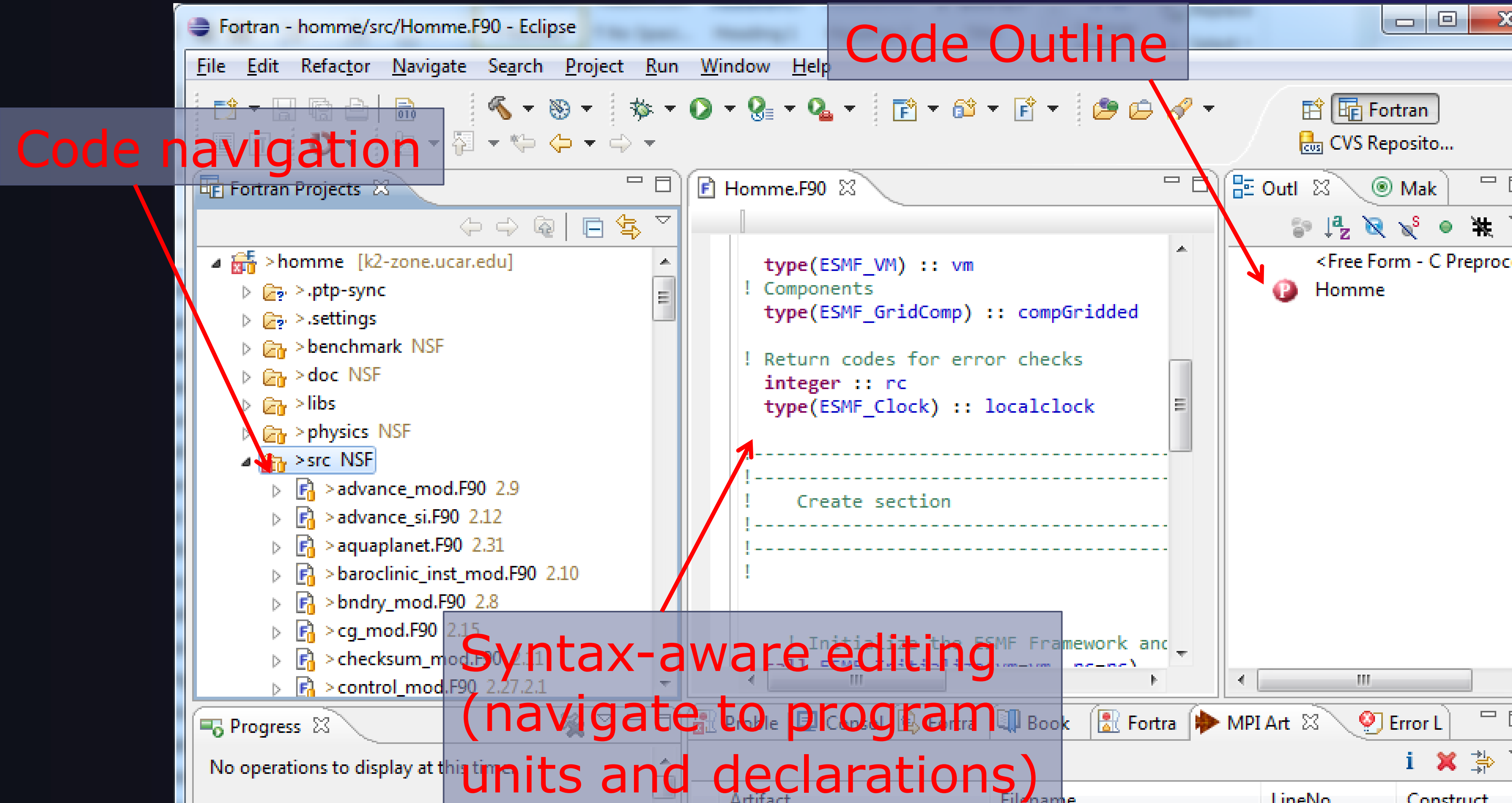
```
   type(ESMF_VM) :: vm
!  Components
   type(ESMF_GridComp) :: compGridded

!  Return codes for error checks
   integer :: rc
   type(ESMF_Clock) :: localclock

!-------------------------------------------------
!-------------------------------------------------
!    Create section
!-------------------------------------------------
!-------------------------------------------------
!

   ! Initialize the ESMF Framework and
```

Code navigation

Syntax-aware editing (navigate to program units and declarations)

Software Engineering

- Code Visibility



Code Outline

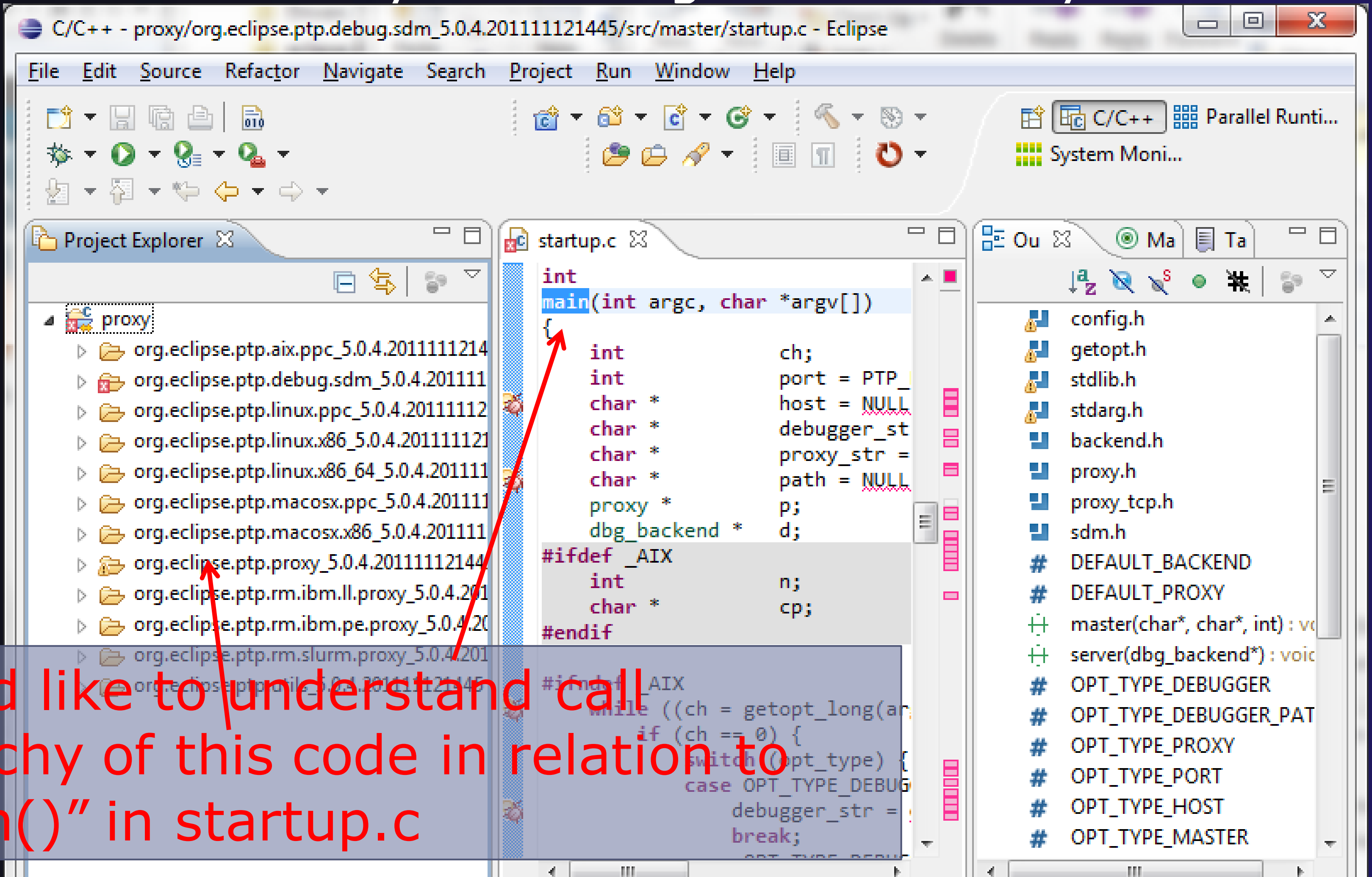
Code navigation

Syntax-aware editing (navigate to program units and declarations)

```
  type(ESMF_VM) :: vm
! Components
  type(ESMF_GridComp) :: compGridded

! Return codes for error checks
  integer :: rc
  type(ESMF_Clock) :: localclock

!-----------------------------------------------------------------
!-----------------------------------------------------------------
!    Create section
!-----------------------------------------------------------------
!-----------------------------------------------------------------
!

  ! Initialize the ESMF Framework and
```

Software Engineering

- Code visibility: deducing call hierarchy



C/C++ - proxy/org.eclipse.ptp.debug.sdm_5.0.4.201111121445/src/master/startup.c - Eclipse

- proxy
  - org.eclipse.ptp.aix.ppc_5.0.4.2011111214
  - org.eclipse.ptp.debug.sdm_5.0.4.201111
  - org.eclipse.ptp.linux.ppc_5.0.4.20111112
  - org.eclipse.ptp.linux.x86_5.0.4.20111112
  - org.eclipse.ptp.linux.x86_64_5.0.4.201111
  - org.eclipse.ptp.macosx.ppc_5.0.4.20111
  - org.eclipse.ptp.macosx.x86_5.0.4.201111
  - org.eclipse.ptp.proxy_5.0.4.20111112144
  - org.eclipse.ptp.rm.ibm.ll.proxy_5.0.4.201
  - org.eclipse.ptp.rm.ibm.pe.proxy_5.0.4.20
  - org.eclipse.ptp.rm.slurm.proxy_5.0.4.201
  - org.eclipse.ptp.utils_5.0.4.201111121445

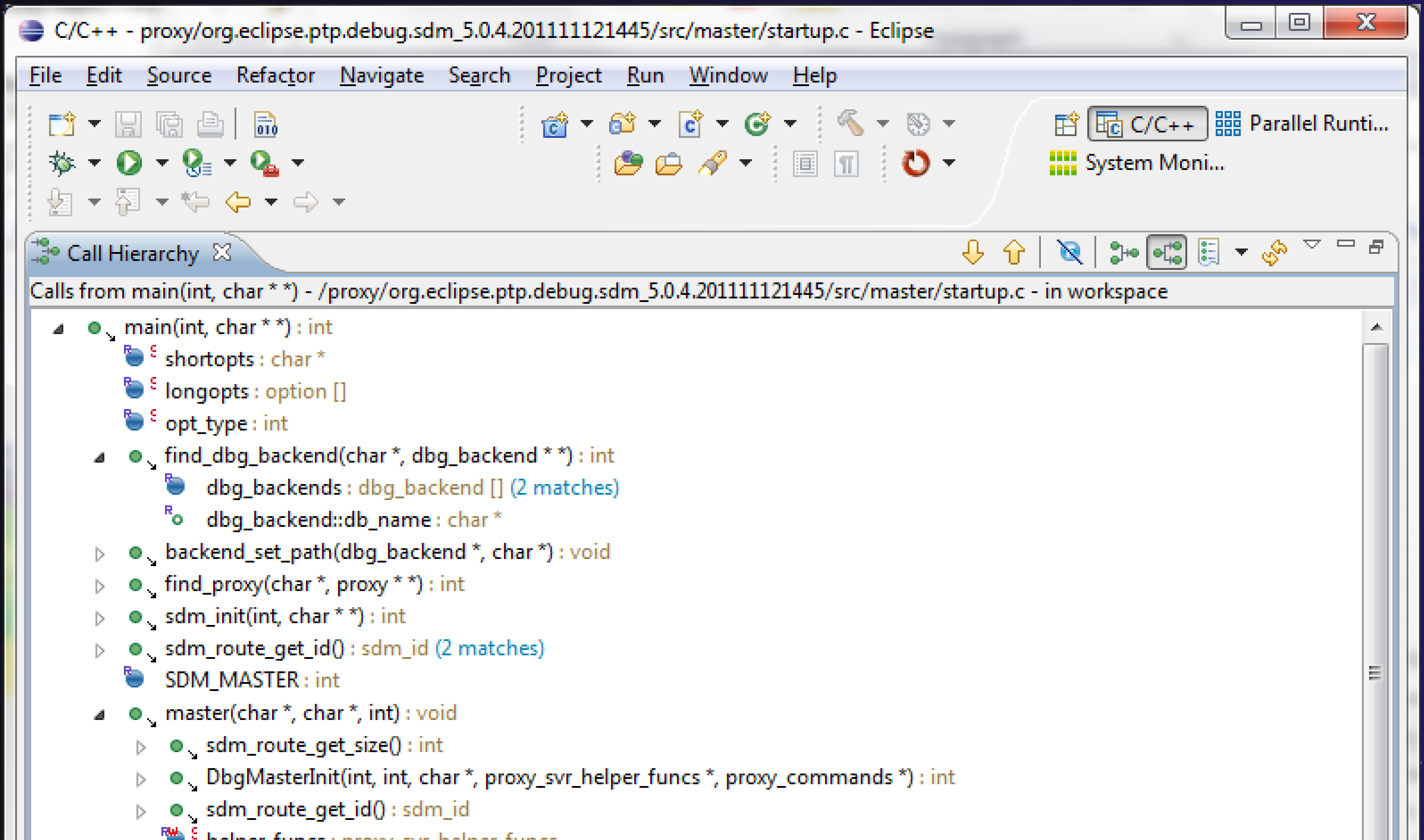
```
int
main(int argc, char *argv[])
{
    int             ch;
    int             port = PTP_
    char *          host = NULL
    char *          debugger_st
    char *          proxy_str =
    char *          path = NULL
    proxy *         p;
    dbg_backend *   d;
#ifdef _AIX
    int             n;
    char *          cp;
#endif

#ifndef _AIX
    while ((ch = getopt_long(ar
        if (ch == 0) {
            switch (opt_type) {
            case OPT_TYPE_DEBUG
                debugger_str =
                break;
```

- 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) : vo
- server(dbg_backend*) : voic
- # OPT_TYPE_DEBUGGER
- # OPT_TYPE_DEBUGGER_PAT
- # OPT_TYPE_PROXY
- # OPT_TYPE_PORT
- # OPT_TYPE_HOST
- # OPT_TYPE_MASTER

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

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

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



C/C++ - proxy/org.eclipse.ptp.debug.sdm_5.0.4.201111121445/src/master/startup.c - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
C/C++
Parallel Runti...
System Moni...
Call Hierarchy

Calls from main(int, char * *) - /proxy/org.eclipse.ptp.debug.sdm_5.0.4.201111121445/src/master/startup.c - in workspace

```
main(int, char * *) : int
    shortopts : char *
    longopts : option []
    opt_type : int
    find_dbg_backend(char *, dbg_backend * *) : int
        dbg_backends : dbg_backend [] (2 matches)
        dbg_backend::db_name : char *
    backend_set_path(dbg_backend *, char *) : void
    find_proxy(char *, proxy * *) : int
    sdm_init(int, char * *) : int
    sdm_route_get_id() : sdm_id (2 matches)
    SDM_MASTER : int
    master(char *, char *, int) : void
        sdm_route_get_size() : int
        DbgMasterInit(int, int, char *, proxy_svr_helper_funcs *, proxy_commands *) : int
        sdm_route_get_id() : sdm_id
        helper_funcs : proxy_svr_helper_funcs
```

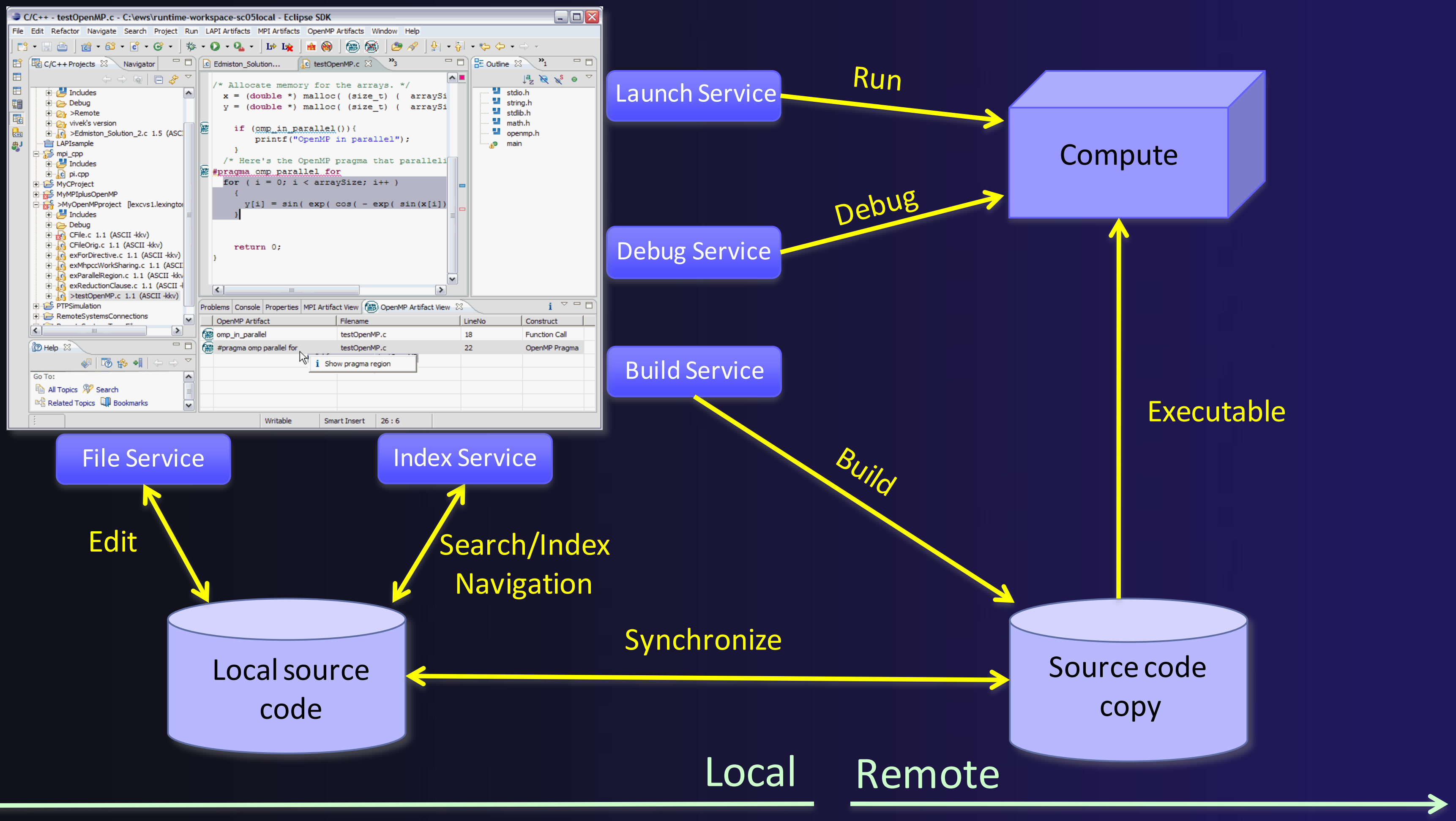
Outline

- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Overview of WHPC: NSF-funded SI2-SSI project to produce a productive and accessible development workbench using Eclipse PTP
  - Determining Requirements, Ensuring Impact
  - 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
- Eclipse PTP Resources

Multi-machine build management

- 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



C/C++ - testOpenMP.c - C:\ews\runtime-workspace-sc05local - Eclipse SDK

```
/* Allocate memory for the arrays. */
  x = (double *) malloc( (size_t) (  arraySi
  y = (double *) malloc( (size_t) (  arraySi

    if (omp_in_parallel()){
        printf("OpenMP in parallel");
    }
  /* Here's the OpenMP pragma that paralleli
#pragma omp parallel for
  for ( i = 0; i < arraySize; i++ )
    {
      y[i] = sin( exp( cos( - exp( sin(x[i])
    }


    return 0;
}
```

OpenMP Artifact	Filename	LineNo	Construct
omp_in_parallel	testOpenMP.c	18	Function Call
#pragma omp parallel for	testOpenMP.c	22	OpenMP Pragma

Show pragma region

Writable | Smart Insert | 26 : 6

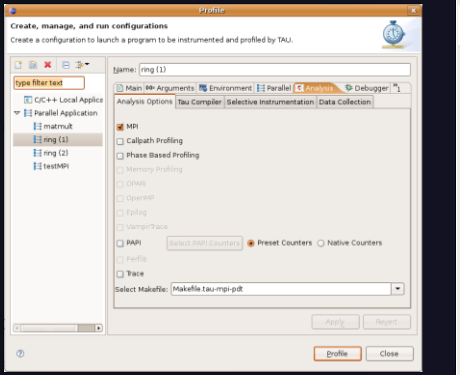
Outline

- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Overview of WHPC: NSF-funded SI2-SSI project to produce a productive and accessible development workbench using Eclipse PTP
  - Determining Requirements, Ensuring Impact
  - 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
- Eclipse PTP Resources

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)
- 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


Profile

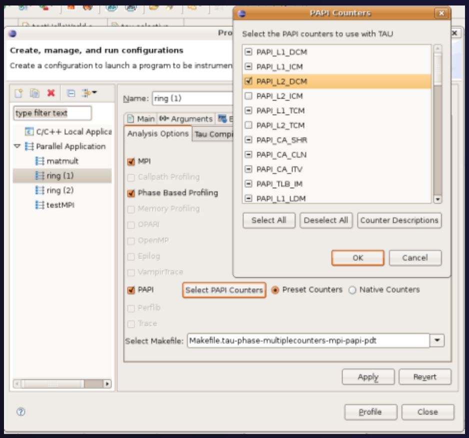
Create, manage, and run configurations

Create a configuration to launch a program to be instrumented and profiled by TAU.

Name: ring (1)

☑ MPI
☐ Callpath Profiling
☐ Phase Based Profiling
☐ Memory Profiling
☐ OPARI
☐ OpenMP
☐ Epilog
☐ VampirTrace
☐ PAPI ● Preset Counters ○ Native Counters
☐ Perflib
☐ Trace

Select Makefile: Makefile.tau-mpi-pdt


PAPI Counters

Select the PAPI counters to use with TAU

☐ PAPI_L1_DCM
☐ PAPI_L1_ICM
☑ PAPI_L2_DCM
☐ PAPI_L2_ICM
☐ PAPI_L1_TCM
☐ PAPI_L2_TCM
☐ PAPI_CA_SHR
☐ PAPI_CA_CLN
☐ PAPI_CA_ITV
☐ PAPI_TLB_IM
☐ PAPI_L1_LDM

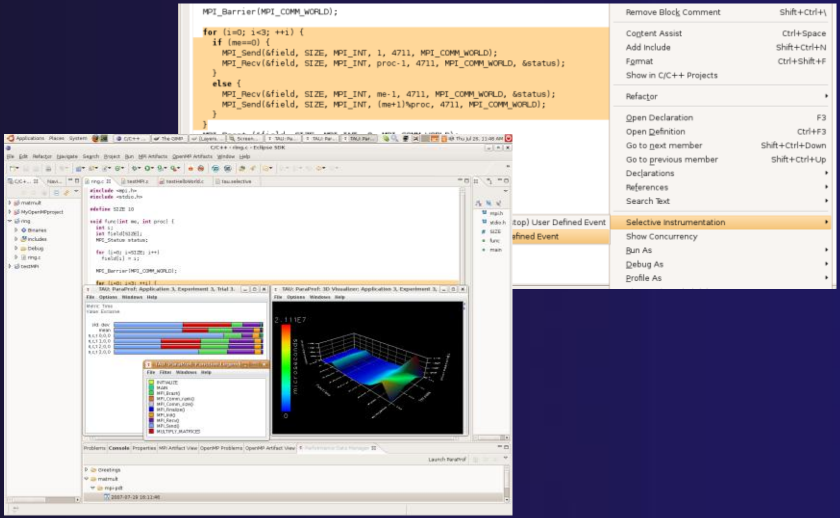
Create, manage, and run configurations

Create a configuration to launch a program to be instrumen

Name: ring (1)

☑ MPI
☐ Callpath Profiling
☑ Phase Based Profiling
☐ Memory Profiling
☐ OPARI
☐ OpenMP
☐ Epilog
☐ VampirTrace
☑ PAPI ● Preset Counters ○ Native Counters
☐ Perflib
☐ Trace

Select Makefile: Makefile.tau-phase-multiplecounters-mpi-papi-pdt


```
MPI_Barrier(MPI_COMM_WORLD);

for (i=0; i<3; ++i) {
  if (me==0) {
    MPI_Send(&field, SIZE, MPI_INT, 1, 4711, MPI_COMM_WORLD);
    MPI_Recv(&field, SIZE, MPI_INT, proc-1, 4711, MPI_COMM_WORLD, &status);
  }
  else {
    MPI_Recv(&field, SIZE, MPI_INT, me-1, 4711, MPI_COMM_WORLD, &status);
    MPI_Send(&field, SIZE, MPI_INT, (me+1)%proc, 4711, MPI_COMM_WORLD);
  }
}
```

Outline

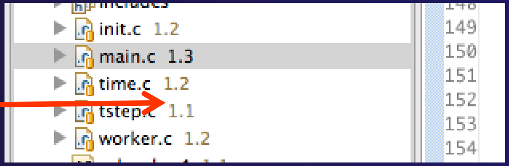
- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Overview of WHPC: NSF-funded SI2-SSI project to produce a productive and accessible development workbench using Eclipse PTP
  - Determining Requirements, Ensuring Impact
  - Improvements to Eclipse PTP
- Software Engineering Practices Enabled by Eclipse PTP
  - Code visibility
  - Multi-system build management
  - Performance tuning
  - Source code control
  - Issue Tracking
  - Documentation
- Eclipse PTP Resources

Source Code Control: "Team" Features

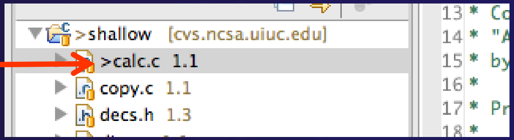
- Eclipse supports integration with multiple version control systems (VCS)
  - CVS, SVN, Git, and others
  - Collectively known as "Team" services
- Many features are common across VCS
  - Compare/merge
  - History
  - Check-in/check-out
- Some differences
  - Version numbers
  - Branching

CVS Features

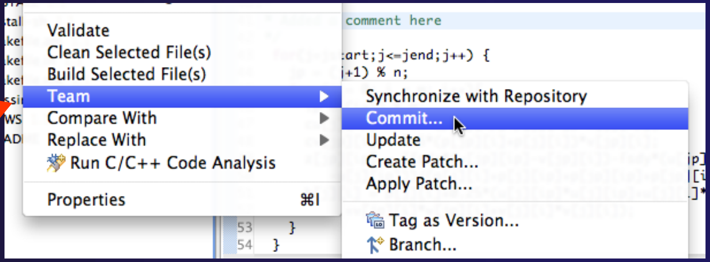
- Shows version numbers next to each resource



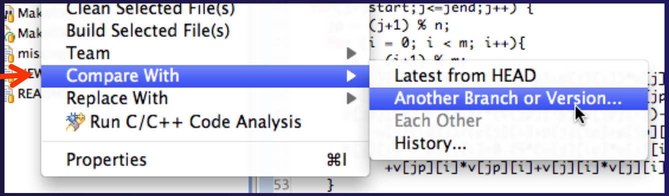
- Marks resources that have changed
  - Can also change color (preference option)



- Context menu for Team operations



- Compare to latest, another branch, or history



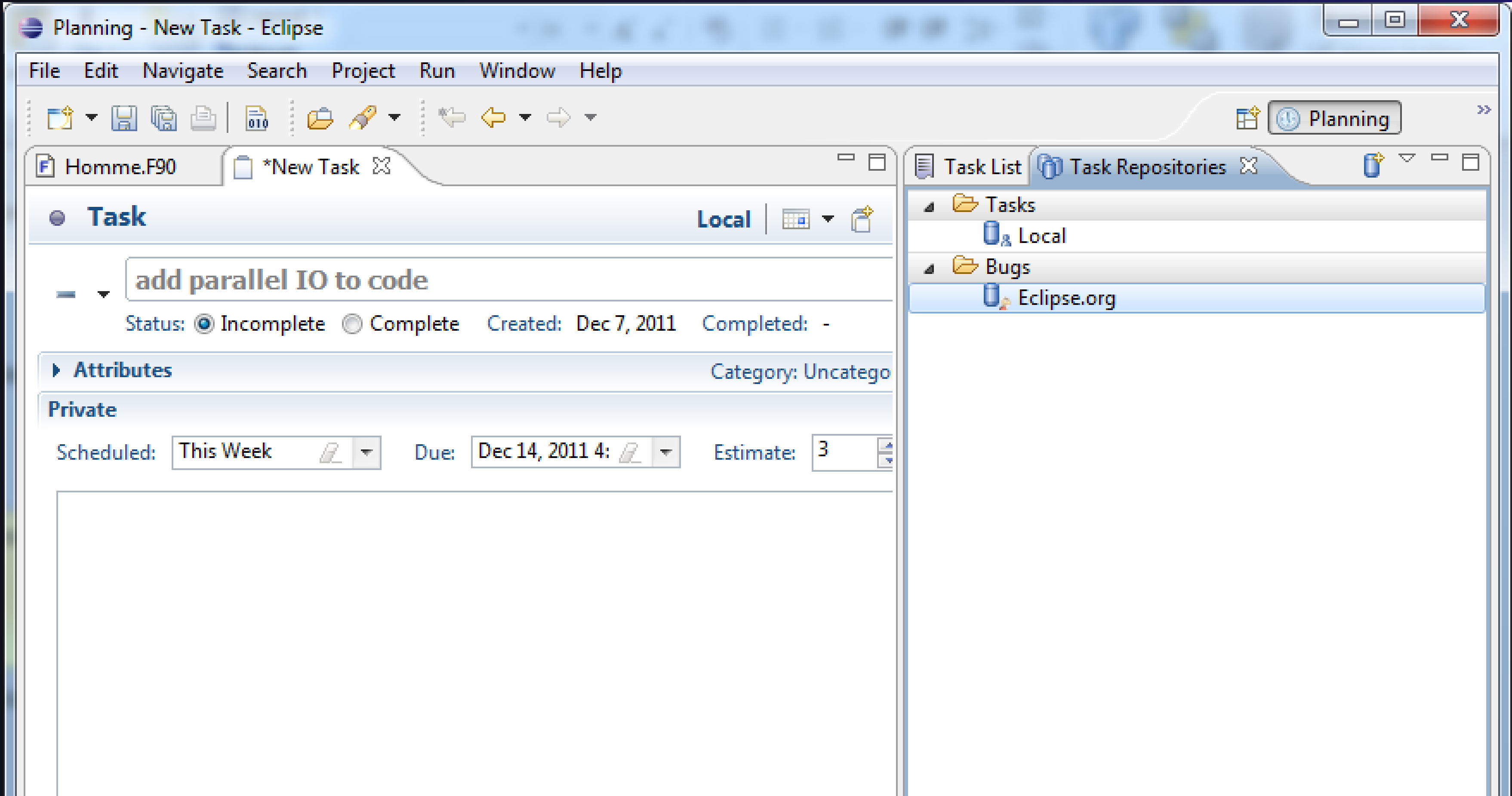
- Synchronize whole project (or any selected resources)

Outline

- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Overview of WHPC: NSF-funded SI2-SSI project to produce a productive and accessible development workbench using Eclipse PTP
  - Determining Requirements, Ensuring Impact
  - 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
- Eclipse PTP Resources

Issue Tracking

- Mylyn Bridge
  - Tracks tasks, links to source and bug repositories



Task — Local

add parallel IO to code

Status: ● Incomplete ○ Complete Created: Dec 7, 2011 Completed: -

Category: Uncatego

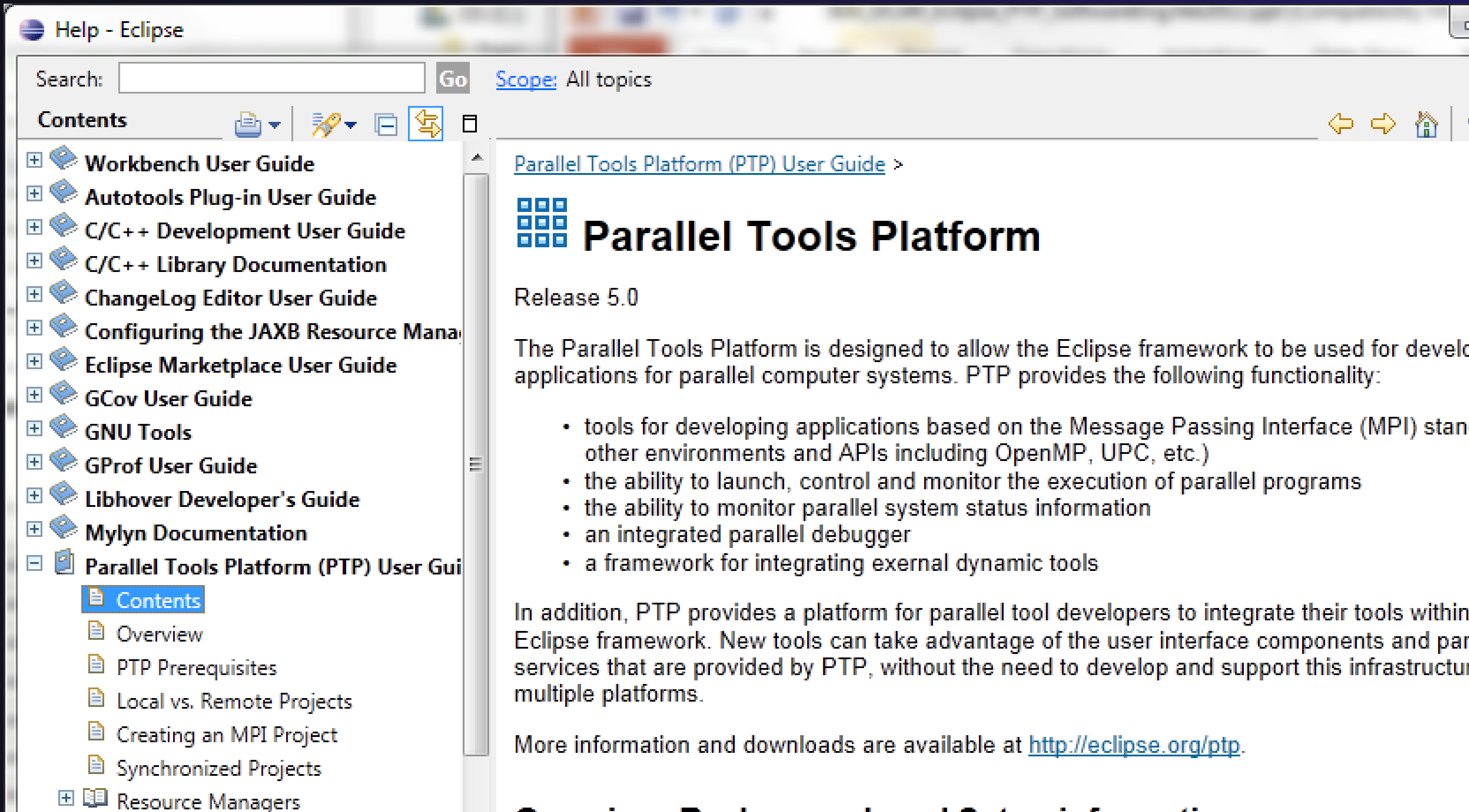
Scheduled: This Week Due: Dec 14, 2011 4: Estimate: 3

Outline

- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Overview of WHPC: NSF-funded SI2-SSI project to produce a productive and accessible development workbench using Eclipse PTP
  - Determining Requirements, Ensuring Impact
  - 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
- Eclipse PTP Resources

Eclipse Documentation

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


Help - Eclipse
Search:
Go
Scope: All topics
Contents
Workbench User Guide
Autotools Plug-in User Guide
C/C++ Development User Guide
C/C++ Library Documentation
ChangeLog Editor User Guide
Configuring the JAXB Resource Mana
Eclipse Marketplace User Guide
GCov User Guide
GNU Tools
GProf User Guide
Libhover Developer's Guide
Mylyn Documentation
Parallel Tools Platform (PTP) User Gui
Contents
Overview
PTP Prerequisites
Local vs. Remote Projects
Creating an MPI Project
Synchronized Projects
Resource Managers

Parallel Tools Platform (PTP) User Guide >

Parallel Tools Platform

Release 5.0

The Parallel Tools Platform is designed to allow the Eclipse framework to be used for develo applications for parallel computer systems. PTP provides the following functionality:

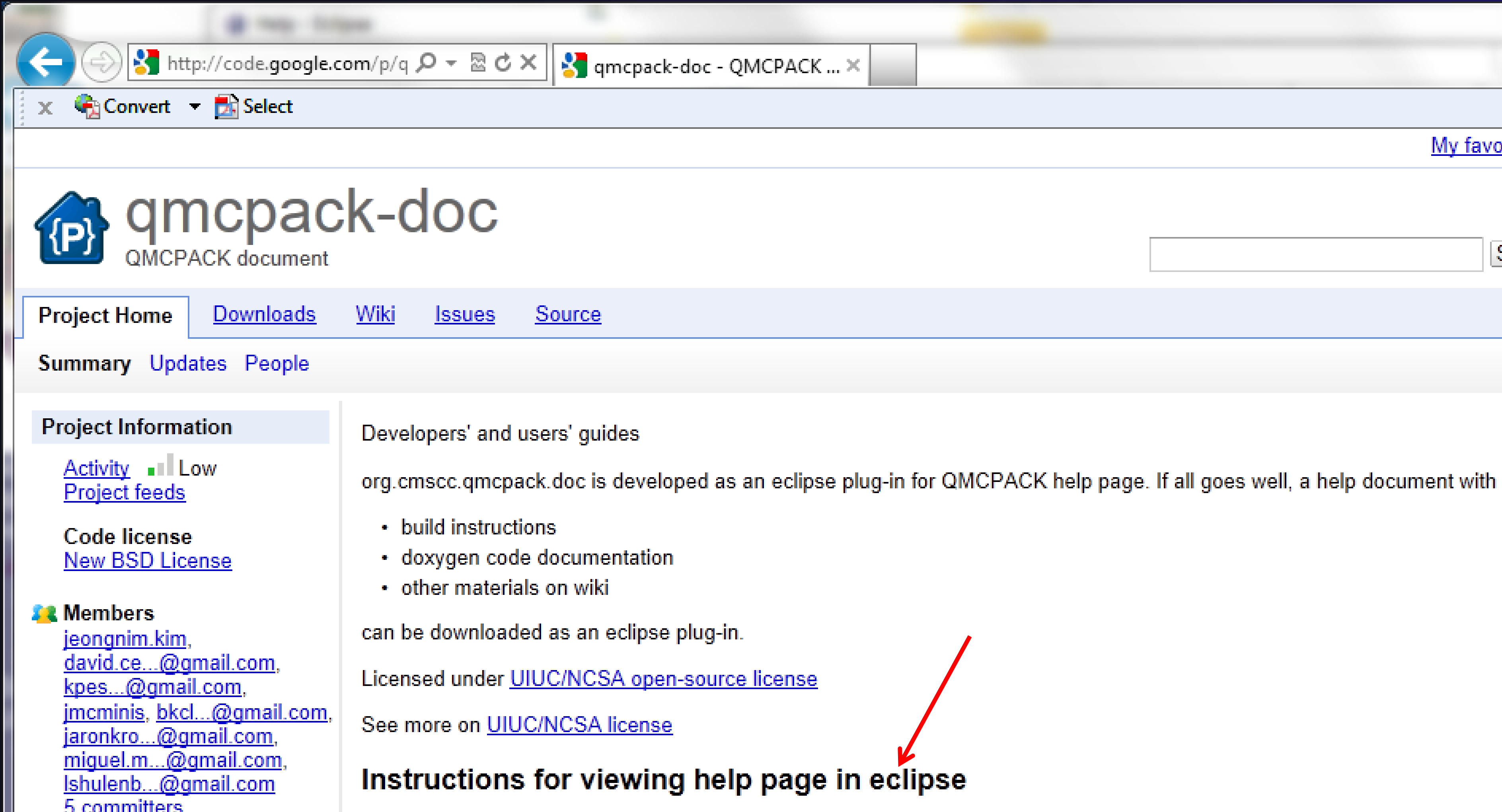
- tools for developing applications based on the Message Passing Interface (MPI) stan other environments and APIs including OpenMP, UPC, etc.)
- the ability to launch, control and monitor the execution of parallel programs
- the ability to monitor parallel system status information
- an integrated parallel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

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

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

Adapting Eclipse Documentation to Other Projects: QMCPack

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



Project Information

Activity Low
Project feeds

Code license
New BSD License

Members
jeongnim.kim, david.ce...@gmail.com, kpes...@gmail.com, jmcminis, 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

Outline

- Overview of Eclipse and Eclipse Parallel Tools Platform (PTP)
- Overview of WHPC: NSF-funded SI2-SSI project to produce a productive and accessible development workbench using Eclipse PTP
  - Determining Requirements, Ensuring Impact
  - 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
- 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
- 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

PTP will only succeed with your participation!