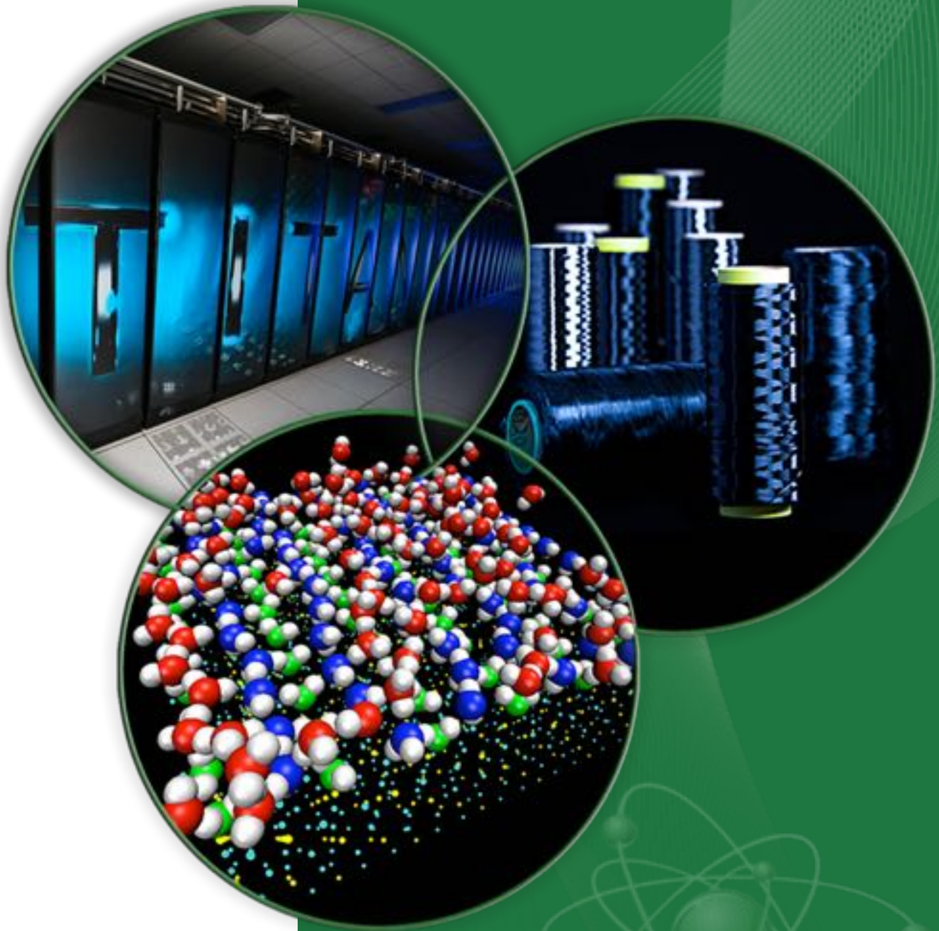
Scientific Software Development with Eclipse



A Best Practices for HPC Developers Webinar

Gregory R. Watson

ORNL is managed by UT-Battelle
for the US Department of Energy

OAK RIDGE
National Laboratory

Contents

- Downloading and Installing Eclipse
- C/C++ Development Features
- Fortran Development Features
- Real-life Development Scenarios
  - Local development
  - Using Git for remote development
  - Using synchronized projects for remote development
- Features for Utilizing HPC Facilities

What is Eclipse?

- An integrated development environment (IDE)
- A platform for developing tools and applications
- An ecosystem for collaborative software development

Getting Started

Downloading and Installing Eclipse

- Eclipse comes in a variety of packages
  - Any package can be used as a starting point
  - May require additional components installed
- Packages that are best for scientific computing:
  - Eclipse for Parallel Application Developers
  - Eclipse IDE for C/C++ Developers
- Main download site
  - https://www.eclipse.org/downloads

Eclipse IDE for C/C++ Developers

- C/C++ development tools
- Git Integration
- Linux tools
  - Libhover
  - Gcov
  - RPM
  - Valgrind
- Tracecompass

Eclipse for Parallel Application Developers

- Eclipse IDE for C/C++ Developers, *plus:*
  - Synchronized projects
  - Fortran development tools
  - Job scheduler support
  - Remote monitoring
  - Remote console

Eclipse for RCP and RAP Developers

280 MB 4,516 DOWNLOADS

A complete set of tools for developers who want to create Eclipse plug-ins, Rich Client Applications or Remote Application Platform (RCP+RAP).

Mac OS X 64 bit

Eclipse for Parallel Application Developers

277 MB 2,267 DOWNLOADS

Tools for C, C++, Fortran, and UPC, including MPI, OpenMP, OpenACC, a parallel debugger, and remotely building, running and monitoring applications...

Mac OS X 64 bit

Eclipse for Testers

149 MB 2,057 DOWNLOADS

Mac OS X 64 bit

Installation

- First, install Java 1.8
  - Check if it is installed using `java -version` from command line
  - Follow procedure for your operating system
- Download Eclipse package
  - Zip for windows
  - Tar.gz for Linux
  - Dmg for Mac OS X
- Uncompress and move to installed location
- Launch Eclipse application

Adding Features

- Eclipse Marketplace
  - Over 1600 packages available
  - Ability to search and browse
  - **Help > Eclipse Marketplace…**
- Eclipse update sites
  - Good for updating installed software to latest version
  - Or if you know the URL
  - **Help > Install New Software…**

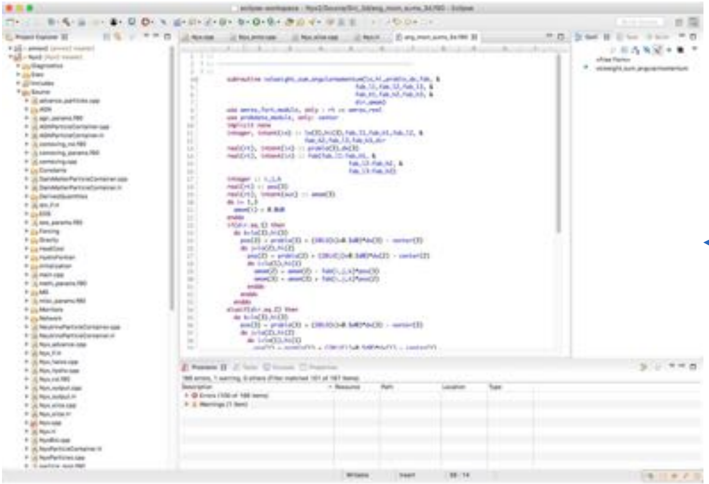
Developing with Eclipse

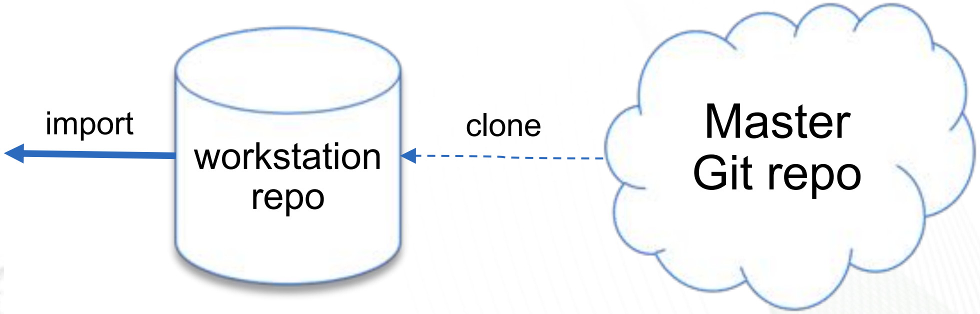
```
elif _operation == "MIRROR_Y":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
elif _operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

    #selection at the end -add back the deselected mirror modifier object
mirror_ob.select= 1
modifier_ob.select=1
bpy.context.scene.objects.active = modifier_ob
print("Selected" + str(modifier_ob)) # modifier ob is the active ob
    #mirror_ob.select = 0
```

C/C++ Development

- Works best on local projects with hierarchical directory structure
- Supports Makefile/CMake based projects
- Can import directly from a Git repository
- Can manage multiple Git repositories



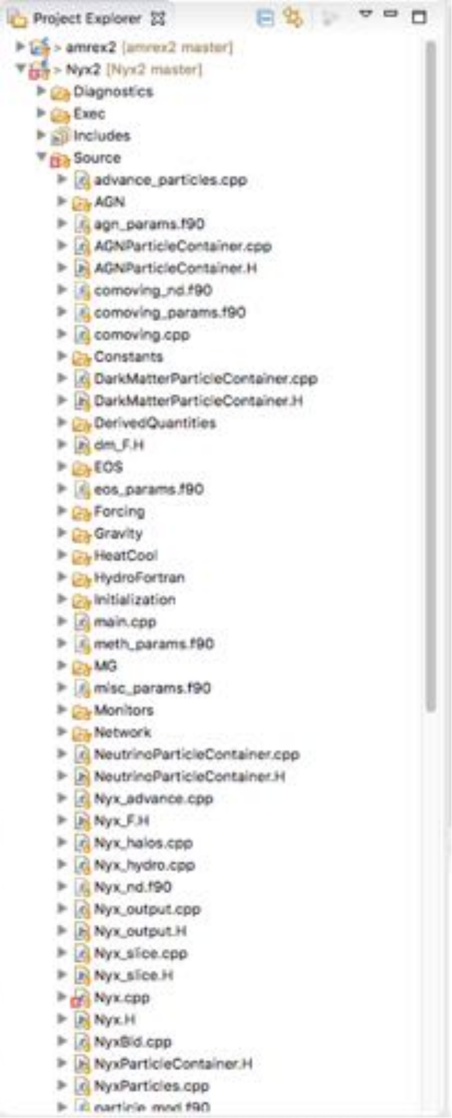


Importing from Git

- Select **File > Import…**
- Select the **Git > Projects from Git** import wizard
- Clone URI
  - https://github.com/AMReX-Astro/Nyx.git
- Once cloned, choose **Import as general project** Wizard then **Finish**
- Then select the project, right click, and choose **New > Convert to a C/C++ Project (Adds C/C++ Nature)**
- Pick **Makefile project** from *Project type*

Project Explorer

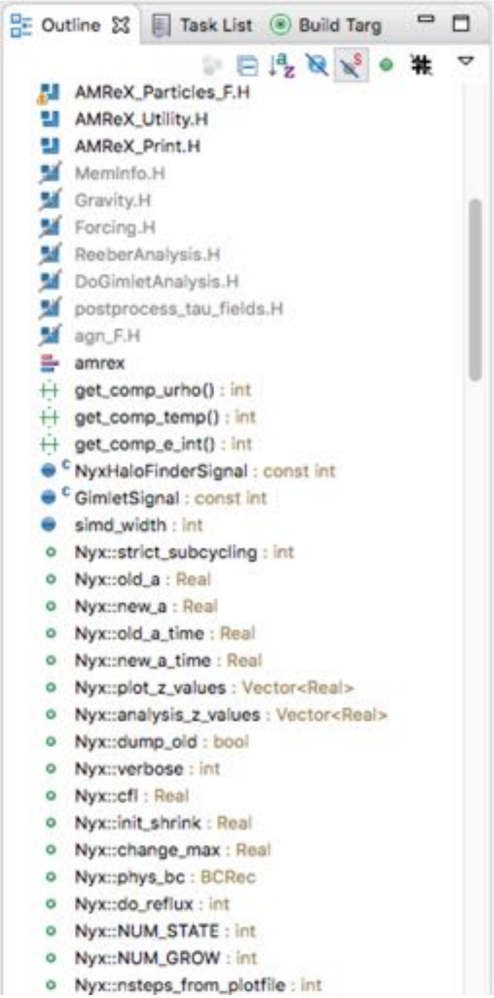
- Shows project tree structure
- Virtual nodes showing
  - Include paths
  - Libraries
  - Binaries and executables
- File nodes can be expanded to show
  - Preprocessor symbols and includes
  - Type and variable declarations
- Compound types can be expanded to show
  - Fields
  - Methods



```
▶ > amrex2 [amrex2 master]
▼ > Nyx2 [Nyx2 master]
    ▶ Diagnostics
    ▶ Exec
    ▶ Includes
    ▼ Source
        ▶ advance_particles.cpp
        ▶ AGN
        ▶ agn_params.f90
        ▶ AGNParticleContainer.cpp
        ▶ AGNParticleContainer.H
        ▶ comoving_nd.f90
        ▶ comoving_params.f90
        ▶ comoving.cpp
        ▶ Constants
        ▶ DarkMatterParticleContainer.cpp
        ▶ DarkMatterParticleContainer.H
        ▶ DerivedQuantities
        ▶ dm_F.H
        ▶ EOS
        ▶ eos_params.f90
        ▶ Forcing
        ▶ Gravity
        ▶ HeatCool
        ▶ HydroFortran
        ▶ Initialization
        ▶ main.cpp
        ▶ meth_params.f90
        ▶ MG
        ▶ misc_params.f90
        ▶ Monitors
        ▶ Network
        ▶ NeutrinoParticleContainer.cpp
        ▶ NeutrinoParticleContainer.H
        ▶ Nyx_advance.cpp
        ▶ Nyx_F.H
        ▶ Nyx_halos.cpp
        ▶ Nyx_hydro.cpp
        ▶ Nyx_nd.f90
        ▶ Nyx_output.cpp
        ▶ Nyx_output.H
        ▶ Nyx_slice.cpp
        ▶ Nyx_slice.H
        ▶ Nyx.cpp
        ▶ Nyx.H
        ▶ NyxBld.cpp
        ▶ NyxParticleContainer.H
        ▶ NyxParticles.cpp
        ▶ particle_mod.f90
```

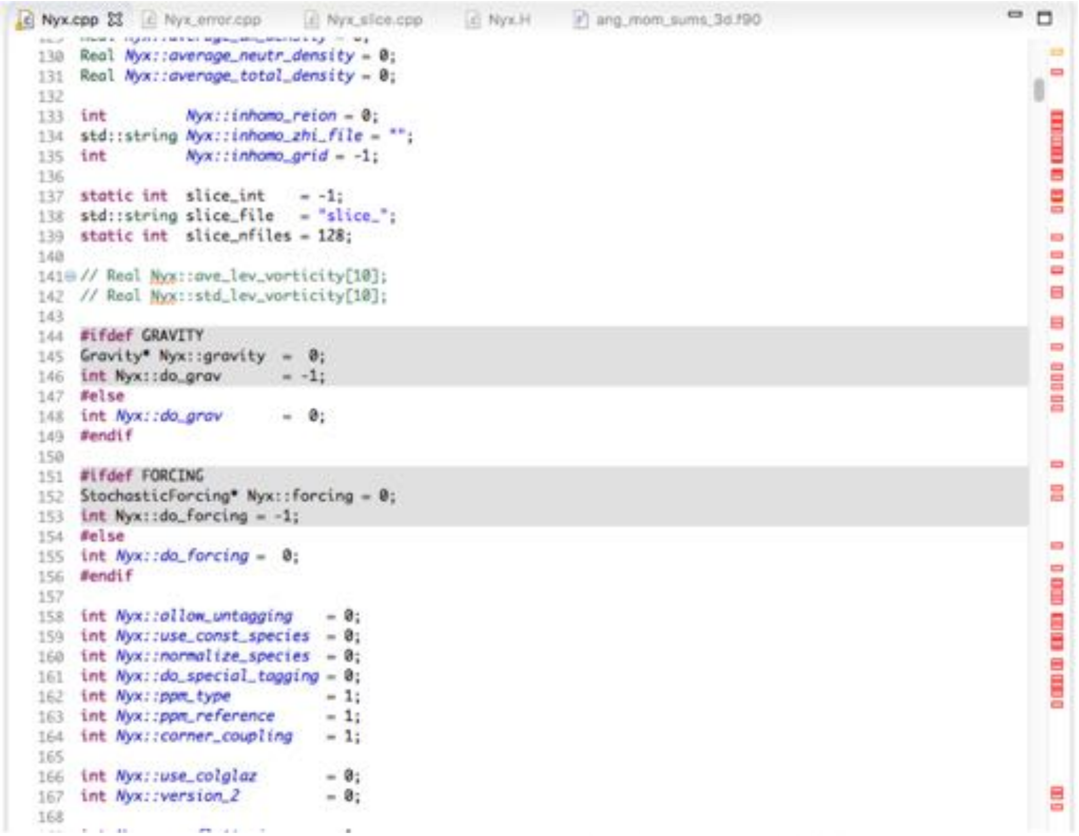
Outline View

- Shows structure of current file in editor
  - Preprocessor symbols and includes
  - Type and variable declarations
- Compound types can be expanded to show
  - Fields
  - Methods
- Can filter what is being shown using buttons or dropdown menu



```
AMReX_Particles_F.H
AMReX_Utility.H
AMReX_Print.H
MemInfo.H
Gravity.H
Forcing.H
ReeberAnalysis.H
DoGimletAnalysis.H
postprocess_tau_fields.H
agn_F.H
amrex
get_comp_urho() : int
get_comp_temp() : int
get_comp_e_int() : int
NyxHaloFinderSignal : const int
GimletSignal : const int
simd_width : int
Nyx::strict_subcycling : int
Nyx::old_a : Real
Nyx::new_a : Real
Nyx::old_a_time : Real
Nyx::new_a_time : Real
Nyx::plot_z_values : Vector<Real>
Nyx::analysis_z_values : Vector<Real>
Nyx::dump_old : bool
Nyx::verbose : int
Nyx::cfl : Real
Nyx::init_shrink : Real
Nyx::change_max : Real
Nyx::phys_bc : BCRec
Nyx::do_reflux : int
Nyx::NUM_STATE : int
Nyx::NUM_GROW : int
Nyx::nsteps_from_plotfile : int
```

Editor Features



```
Real Nyx::average_neutr_density = 0;
Real Nyx::average_total_density = 0;

int         Nyx::inhomo_reion = 0;
std::string Nyx::inhomo_zhi_file = "";
int         Nyx::inhomo_grid = -1;

static int  slice_int    = -1;
std::string slice_file   = "slice_";
static int  slice_nfiles = 128;

// Real Nyx::ave_lev_vorticity[10];
// Real Nyx::std_lev_vorticity[10];

#ifdef GRAVITY
Gravity* Nyx::gravity  =  0;
int Nyx::do_grav       = -1;
#else
int Nyx::do_grav       =  0;
#endif

#ifdef FORCING
StochasticForcing* Nyx::forcing = 0;
int Nyx::do_forcing = -1;
#else
int Nyx::do_forcing =  0;
#endif

int Nyx::allow_untagging    = 0;
int Nyx::use_const_species  = 0;
int Nyx::normalize_species  = 0;
int Nyx::do_special_tagging = 0;
int Nyx::ppm_type           = 1;
int Nyx::ppm_reference      = 1;
int Nyx::corner_coupling    = 1;

int Nyx::use_colglaz        = 0;
int Nyx::version_2          = 0;
```

- Syntax coloring
- Line numbers
- Folding
- Content assist
- Hover help
- Block selection
- Code activation based on preprocessor directives
- Formatting
  - Can be run from the command line
- Display revision information

Oak Ridge National Laboratory

Formatting and Refactoring

- Formatting
  - Generate Getters and Setters
  - Add/Organize Includes
  - Implement Method
  - Toggle Comment

- Refactoring
  - Rename
  - Extract Constant
  - Extract Local Variable
  - Extract Function
  - Toggle Function Definition
  - Hide Method

And many other features…

Fortran Development[1]

- Fortran editor
  - Similar to C/C++ editor
- Fortran perspective
  - Gathers together various Fortran specific views
  - Adds Fortran declaration view
- Fortran feature search
  - Search for language features

[1] Requires Parallel Application Developers Package

Fortran Editor

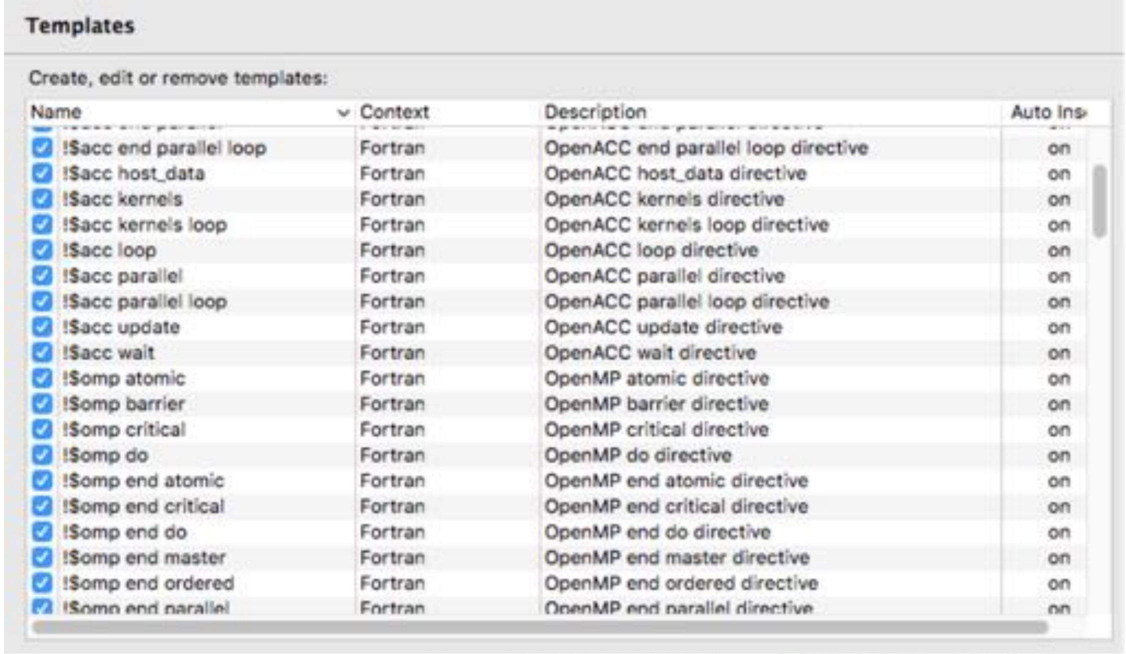
- Supports free and fixed formats
- Opens for any file ending in Fortran suffix
  - .f, .F, etc.: fixed source form
  - .f08, .f90, etc.: free source form with INCLUDE
  - .F08, .F90, etc.: free source form with C preprocessor
- Syntax coloring
- By default, only basic editing features are enabled

Advanced Fortran Development

- Fortran analysis/refactoring is disabled by default
- If not already a Fortran project
  - Right click on project > **Convert to Fortran Project**
- Open project properties
- Select **Fortran General > Analysis/Refactoring**
- Check **Enable Fortran analysis/refactoring**
- Choose analysis properties

Advanced Editor Features

- Folding
- Content assist
- Hover help
- Code templates



Templates

Create, edit or remove templates:

	Name	Context	Description	Auto Ins
☑	!$acc end parallel loop	Fortran	OpenACC end parallel loop directive	on
☑	!$acc host_data	Fortran	OpenACC host_data directive	on
☑	!$acc kernels	Fortran	OpenACC kernels directive	on
☑	!$acc kernels loop	Fortran	OpenACC kernels loop directive	on
☑	!$acc loop	Fortran	OpenACC loop directive	on
☑	!$acc parallel	Fortran	OpenACC parallel directive	on
☑	!$acc parallel loop	Fortran	OpenACC parallel loop directive	on
☑	!$acc update	Fortran	OpenACC update directive	on
☑	!$acc wait	Fortran	OpenACC wait directive	on
☑	!$omp atomic	Fortran	OpenMP atomic directive	on
☑	!$omp barrier	Fortran	OpenMP barrier directive	on
☑	!$omp critical	Fortran	OpenMP critical directive	on
☑	!$omp do	Fortran	OpenMP do directive	on
☑	!$omp end atomic	Fortran	OpenMP end atomic directive	on
☑	!$omp end critical	Fortran	OpenMP end critical directive	on
☑	!$omp end do	Fortran	OpenMP end do directive	on
☑	!$omp end master	Fortran	OpenMP end master directive	on
☑	!$omp end ordered	Fortran	OpenMP end ordered directive	on
☑	!$omp end parallel	Fortran	OpenMP end parallel directive	on

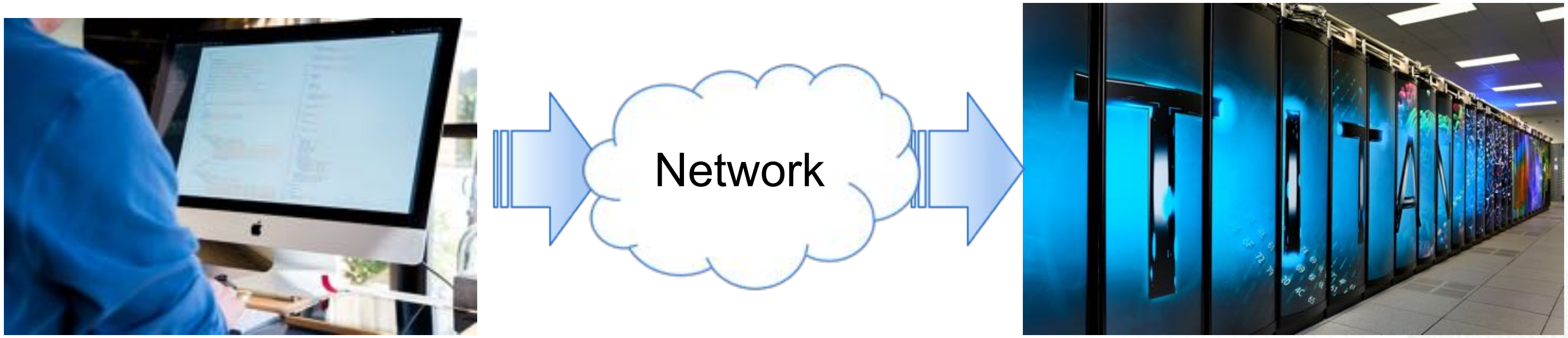
Real-life Development Scenarios

Example Scenarios

- Local development – already covered
- Using Git for remote development
- Using synchronized projects for remote development

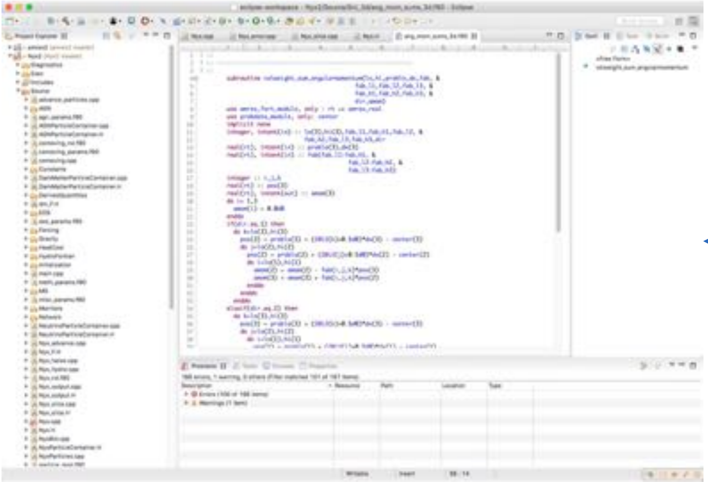
Remote Development

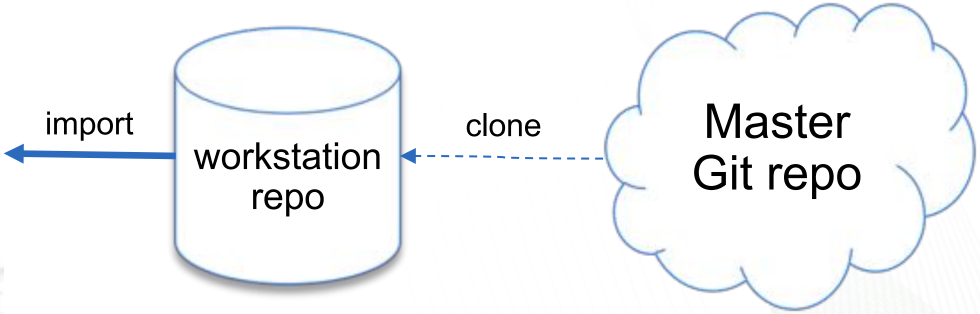
- In scientific computing, application code is normally compiled and run on remote system
- Local machine rarely has same environment, libraries, etc. as target system
- May have different architecture, utilize GPUs, etc.
- Also usually need to submit job via batch scheduler



Remote Development Using Git

- Clone repository to workstation either through UI or command line
- Import into Eclipse as before
- Clone repository on target system if it is not already there

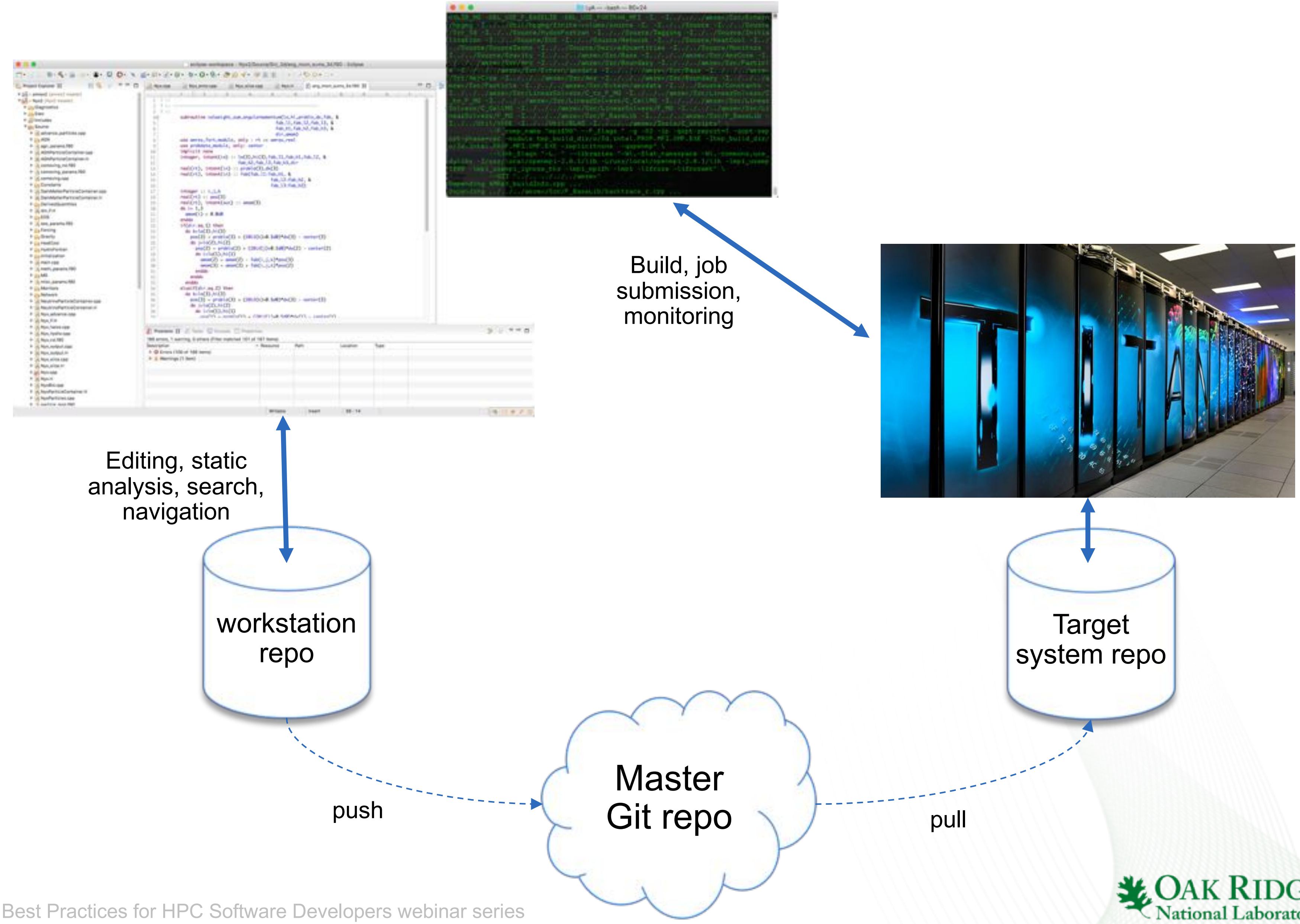




Remote Development Using Git Cont...

- Changes committed to workstation repository
  - Push to central repo (e.g. GitHub) or directly to target system (if allowed)
  - Can utilize code reviews (e.g. Gerrit) and continuous integration if required
- Pull changes into repository on target machine
- Manually run build
- Manually submit to job scheduler

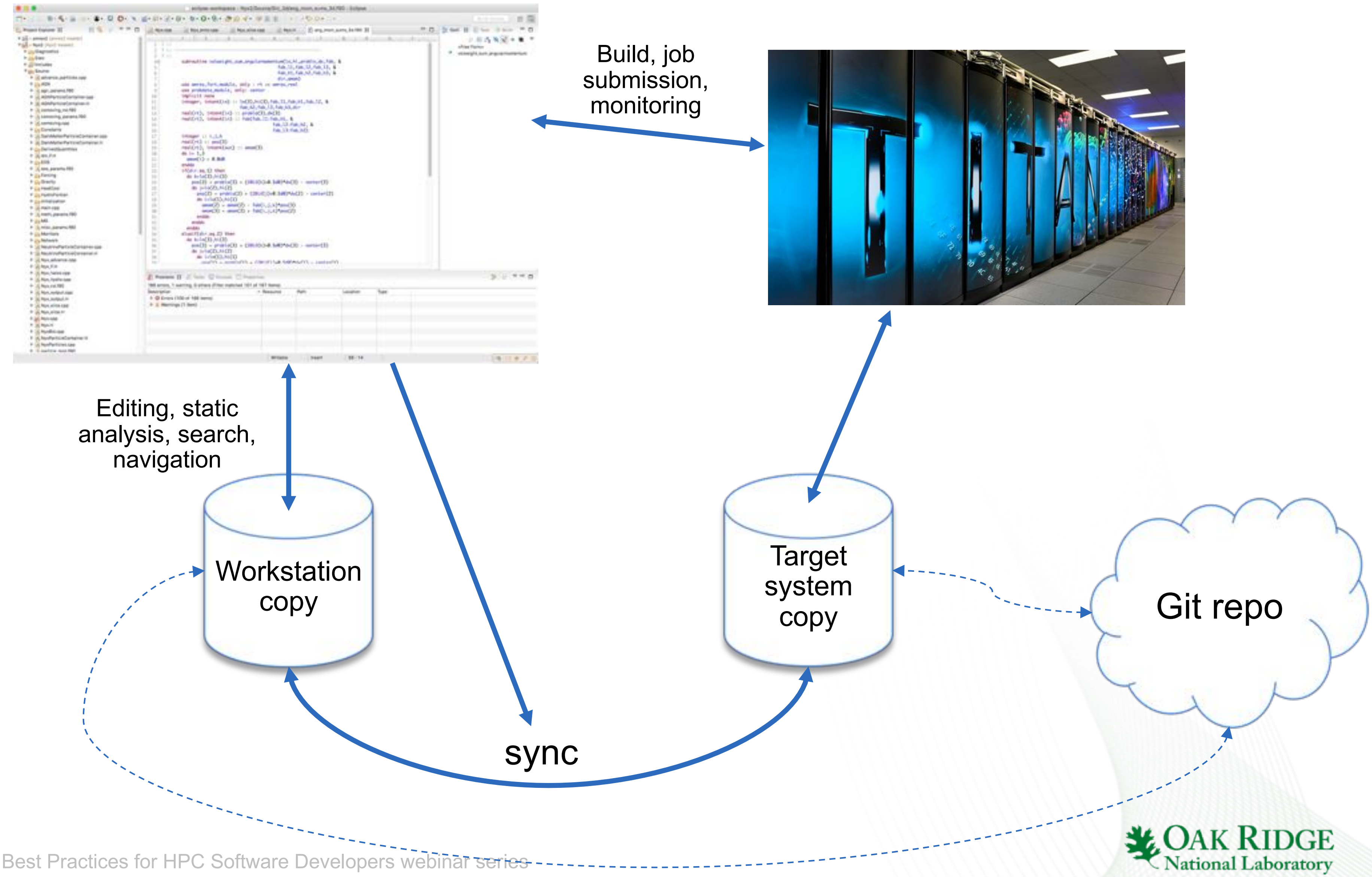
Remote Development Using Git Cont...


Build, job submission, monitoring
Editing, static analysis, search, navigation
workstation repo
Target system repo
Master Git repo
push
pull

Remote Development using Synchronized Projects

- Rather than using Git, Eclipse can manage the synchronization for you
  - Any changes made locally will be automatically synchronized
  - Changes made remotely can be manually synchronized or will be picked up at next sync point
  - Can configure filters to avoid copying large files
- Orthogonal to Git, so both can be used
- Can start with either local or remote source

Synchronized Projects



Build, job submission, monitoring

Editing, static analysis, search, navigation

Workstation copy

Target system copy

Git repo

sync

Starting with Local Source

- Create project as before (e.g. from Git)
- **New > Other**
- **Other > Convert to Synchronized Project**
- Choose project
- Choose connection and remote directory
- After synchronize
  - Go to project properties
  - **C/C++ Build > Tool Chain Editor**
  - Set the current toolchain for the target system (change current build back to “Sync Builder” if necessary)

Remote Building

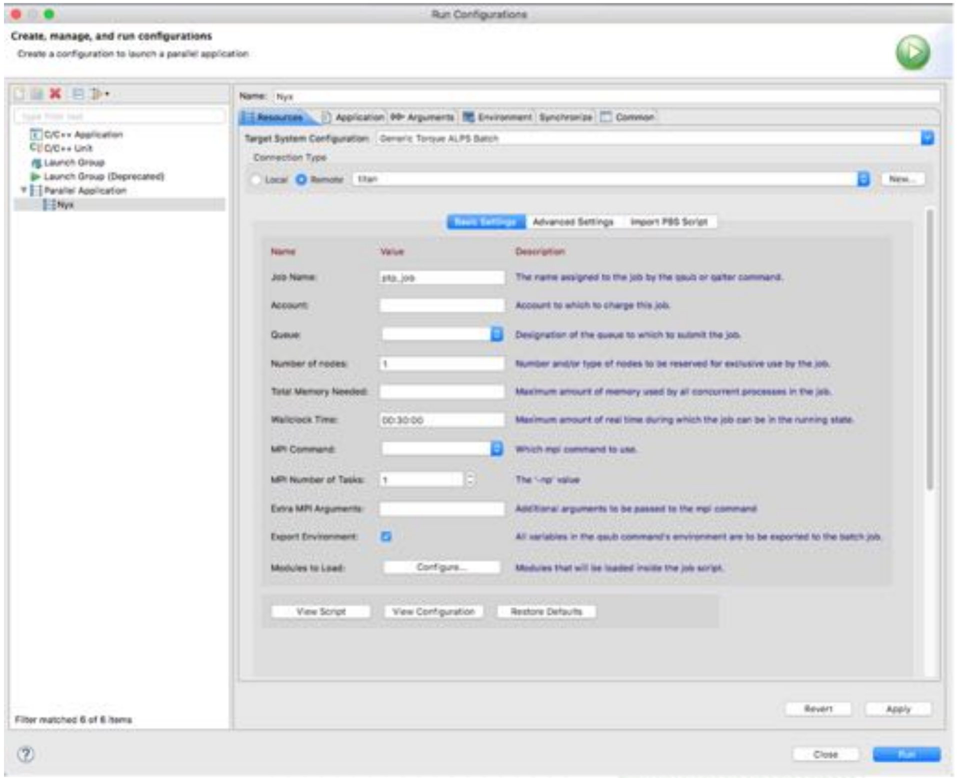
- Synchronized projects automatically set up remote build
- Clicking on the build button will run the build command remotely (normally “make”)
- Add build targets to run “make whatever”
- Can run more complex build commands also

Other Parallel Application Developer Features

- Job submission
- Monitor system/queues
- Remote console

Job Submission

- Use the “Parallel Application” run configuration type
- Comes pre-configured with many generic- and system-specific configurations
- Supports most common job schedulers and runtimes
- Can use to launch remote commands also



Name: Nyx

Target System Configuration: Generic Torque ALPS Batch

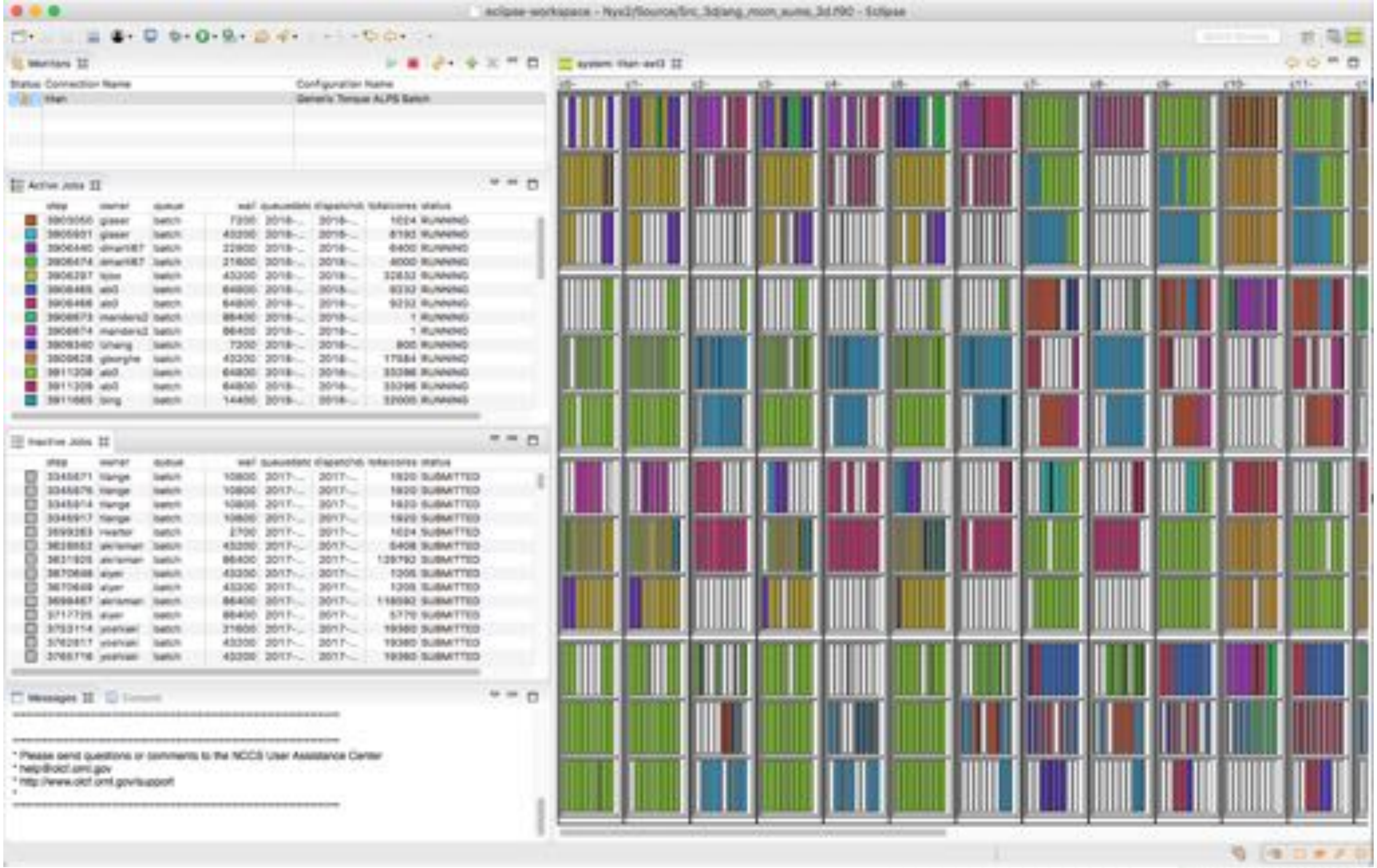
Connection Type: ○ Local ● Remote titan

Name	Value	Description
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.
Number of nodes:	1	Number and/or type of nodes to be reserved for exclusive use by 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.
MPI Command:		Which mpi command to use.
MPI Number of Tasks:	1	The '-np' value
Extra MPI Arguments:		Additional arguments to be passed to the mpi command
Export Environment:	☑	All variables in the qsub command's environment are to be exported to the batch job.
Modules to Load:	Configure...	Modules that will be loaded inside the job script.

System and Job Monitoring

- Comes pre-configured with many generic- and system-specific configurations
- Switch to “System Monitoring” perspective
- Can monitor multiple systems simultaneously
- Jobs launched through PTP can be controlled
- Once job is completed, stdout/stderr is accessible from the UI

System Monitoring Perspective



eclipse-workspace - Nyx2/Source/Src_3d/ang_mom_sums_3d.f90 - Eclipse
Monitors
system: titan-ext3
c0- c1- c2- c3- c4- c5- c6- c7- c8- c9- c10- c11-

Status	Connection Name	Configuration Name
	titan	Generic Torque ALPS Batch

Active Jobs

	step	owner	queue	wall	queuedate	dispatchdate	totalcores	status
🟫	3905050	glaser	batch	7200	2018-...	2018-...	1024	RUNNING
🟦	3905931	glaser	batch	43200	2018-...	2018-...	8192	RUNNING
🟪	3906440	dmart87	batch	22800	2018-...	2018-...	6400	RUNNING
🟩	3906474	dmart87	batch	21600	2018-...	2018-...	4000	RUNNING
🟨	3906297	tplee	batch	43200	2018-...	2018-...	32832	RUNNING
🟦	3906465	ab0	batch	64800	2018-...	2018-...	9232	RUNNING
🟥	3906466	ab0	batch	64800	2018-...	2018-...	9232	RUNNING
🟩	3906673	manders2	batch	86400	2018-...	2018-...	1	RUNNING
🟪	3906674	manders2	batch	86400	2018-...	2018-...	1	RUNNING
🟦	3908340	lzhang	batch	7200	2018-...	2018-...	800	RUNNING
🟫	3909628	gborghe	batch	43200	2018-...	2018-...	17584	RUNNING
🟩	3911208	ab0	batch	64800	2018-...	2018-...	33296	RUNNING
🟥	3911209	ab0	batch	64800	2018-...	2018-...	33296	RUNNING
🟦	3911665	bing	batch	14400	2018-...	2018-...	32000	RUNNING

Inactive Jobs

	step	owner	queue	wall	queuedate	dispatchdate	totalcores	status
□	3345671	tlange	batch	10800	2017-...	2017-...	1920	SUBMITTED
□	3345676	tlange	batch	10800	2017-...	2017-...	1920	SUBMITTED
□	3345914	tlange	batch	10800	2017-...	2017-...	1920	SUBMITTED
□	3345917	tlange	batch	10800	2017-...	2017-...	1920	SUBMITTED
□	3599263	rwalter	batch	2700	2017-...	2017-...	1024	SUBMITTED
□	3628552	akrisman	batch	43200	2017-...	2017-...	5408	SUBMITTED
□	3631505	akrisman	batch	86400	2017-...	2017-...	128792	SUBMITTED
□	3670646	aiyer	batch	43200	2017-...	2017-...	1206	SUBMITTED
□	3670648	aiyer	batch	43200	2017-...	2017-...	1206	SUBMITTED
□	3699467	akrisman	batch	86400	2017-...	2017-...	118592	SUBMITTED
□	3717725	aiyer	batch	86400	2017-...	2017-...	5770	SUBMITTED
□	3753114	yoelsal	batch	21600	2017-...	2017-...	19360	SUBMITTED
□	3762817	yoelsal	batch	43200	2017-...	2017-...	19360	SUBMITTED
□	3765716	yoelsal	batch	43200	2017-...	2017-...	19360	SUBMITTED

Messages Console

```
**************************************************************

**************************************************************
* Please send questions or comments to the NCCS User Assistance Center
* help@olcf.ornl.gov
* http://www.olcf.ornl.gov/support
*
**************************************************************
```

Remote Console

- Select Console view using tab
- Click on open console button and choose “Command Shell Console”
- Select the Connection Type and Connection name you want to use
- Click OK
- You will now have a shell on the target machine
- Open as many consoles as you like

Environment Modules

- Many HPC systems use environment modules
  - Allow different compilers/libraries to be selected
- Environment modules are integrated with the Parallel Application Developer package
  - Modules can be selected before the project is built
  - Modules can be selected before the code is submitted to the job scheduler

Summary

- Eclipse provides a variety of features to support scientific software development
  - C/C++/Fortran development
  - Local/remote project management
  - Integration with Git
  - Support for job submission and monitoring
  - Environment module support
- Allows developers who prefer IDEs to pick and choose how they wish to develop
- Supports complex workflows and provides both automatic and manual configuration options

Additional Material

C/C++

Importing Dependencies (Optional)

- Nyx depends on AMReX
- Repeat the same process for the AMReX repo
  - https://github.com/AMReX-Codes/amrex.git
- Only needed if you
  - Want to build locally
  - Want to resolve include files and types

Project Configuration

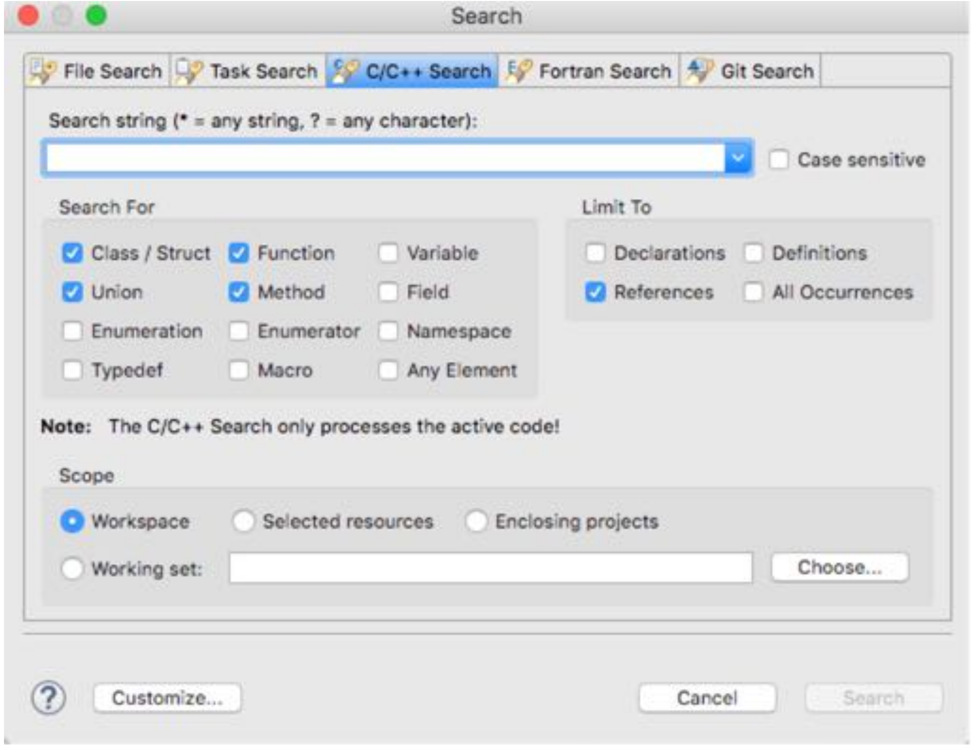
- Some settings are worked out automatically
  - Include paths
  - Compiler defined macros
- Usually need to add includes from dependent libraries manually
  - Open project properties
  - Go to **C/C++ General > Preprocessor Include Paths**
  - Add appropriate entries

Managing Code Analysis

- Code analysis (codan) requires headers to be configured correctly
- If the automatic configuration misses some header files you can add these manually
- You can also disable codan
  - Open project properties
  - Go to **C/C++ General > Code Analysis**
  - Select “Use project settings”
  - Uncheck problems you don’t wish to see

Search

- Search for
  - Class/struct/union
  - Function/method
  - Variable/field
  - Namespace
  - Typedef
  - Macro
- Limit to
  - Declarations
  - References
  - Definitions



Search string (* = any string, ? = any character):

□ Case sensitive

Search For

☑ Class / Struct ☑ Function □ Variable
☑ Union ☑ Method □ Field
□ Enumeration □ Enumerator □ Namespace
□ Typedef □ Macro □ Any Element

Limit To

□ Declarations □ Definitions
☑ References □ All Occurrences

Note: The C/C++ Search only processes the active code!

Scope

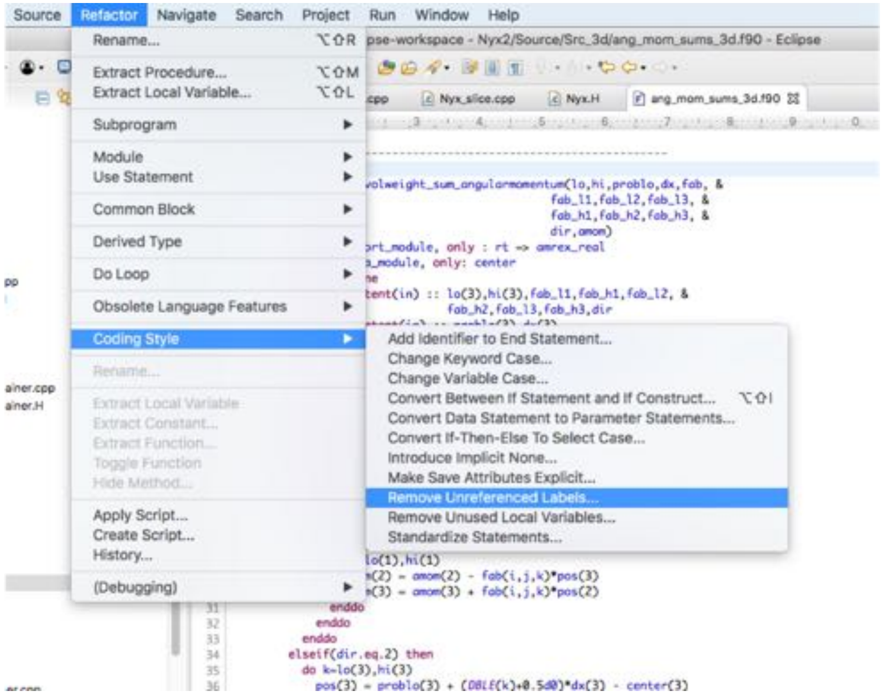
● Workspace ○ Selected resources ○ Enclosing projects
○ Working set:

Other Features

- C/C++ Unit Testing
- Visual debugging
- Multicore debugging
- LLVM support
- And more…

Fortran

Refactoring


Source
Refactor
Navigate
Search
Project
Run
Window
Help
pse-workspace - Nyx2/Source/Src_3d/ang_mom_sums_3d.f90 - Eclipse
Rename... ⌥⇧R
Extract Procedure... ⌥⇧M
Extract Local Variable... ⌥⇧L
Subprogram ▶
Module ▶
Use Statement ▶
Common Block ▶
Derived Type ▶
Do Loop ▶
Obsolete Language Features ▶
Coding Style ▶
Rename...
Extract Local Variable
Extract Constant...
Extract Function...
Toggle Function
Hide Method...
Apply Script...
Create Script...
History...
(Debugging) ▶
Add Identifier to End Statement...
Change Keyword Case...
Change Variable Case...
Convert Between If Statement and If Construct... ⌥⇧I
Convert Data Statement to Parameter Statements...
Convert If-Then-Else To Select Case...
Introduce Implicit None...
Make Save Attributes Explicit...
Remove Unreferenced Labels...
Remove Unused Local Variables...
Standardize Statements...
cpp
Nyx_slice.cpp
Nyx.H
ang_mom_sums_3d.f90
pp
ainer.cpp
ainer.H

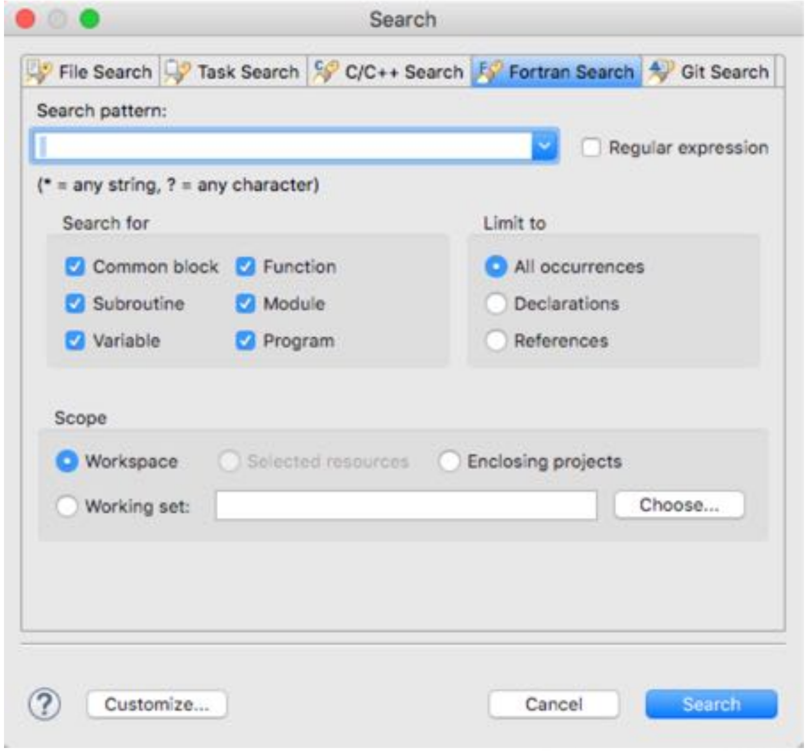
```
volweight_sum_angularmomentum(lo,hi,problo,dx,fab, &
                              fab_l1,fab_l2,fab_l3, &
                              fab_h1,fab_h2,fab_h3, &
                              dir,amom)
ort_module, only : rt => amrex_real
a_module, only: center
ne
tent(in) :: lo(3),hi(3),fab_l1,fab_h1,fab_l2, &
            fab_h2,fab_l3,fab_h3,dir
```

```
lo(1),hi(1)
n(2) = amom(2) - fab(i,j,k)*pos(3)
n(3) = amom(3) + fab(i,j,k)*pos(2)
        enddo
      enddo
    enddo
  elseif(dir.eq.2) then
    do k=lo(3),hi(3)
      pos(3) = problo(3) + (DBLE(k)+0.5d0)*dx(3) - center(3)
```

- Rename
- Extract procedure
- Extract local variable
- Make private entity public
- Add subprogram parameter
- Make common block names consistent
- Unroll loop
- Introduce implicit none
- And more…

Search

- Search for
  - Common block
  - Subroutine
  - Variable
  - Function
  - Module
  - Program
- Limit to
  - Declarations
  - References



Search pattern:

□ Regular expression

(* = any string, ? = any character)

Search for

☑ Common block ☑ Function
☑ Subroutine ☑ Module
☑ Variable ☑ Program

Limit to

● All occurrences
○ Declarations
○ References

Scope

● Workspace ○ Selected resources ○ Enclosing projects
○ Working set:

Synchronized projects

Starting with Remote Source

- **New > Synchronized C/C++ Project**
- Pick project name (can be different from remote)
- Pick remote connection or create a new one
- Browse for remote directory
- Pick project type (normally Makefile > Empty Project)
- Select toolchains for local and remote copies
- Remote source will be automatically copied to a local project

Configuring Synchronized Projects

- Advanced editing features can be used because there is a local copy of the source
- It would be useful if the editor reflected the remote environment
  - System/library include files
  - Architecture specific macro definitions
- This information can be gathered from
  - Automatically from compilers on the remote system
  - Manually from compilers on the remote system (macros file)
  - Entered manually

Automatic Configuration (GCC only)

- From project properties
  - **C/C++ General > Preprocessor Include Paths, Macros, etc.**
  - Click on “Providers”
  - Select
    - Sync GCC Build Output Parser
    - Sync GCC Builtin Compiler Settings
  - Check “Allocated console in Console View” if you want to see the commands that are run
- Should trigger a re-index of the project

Manual Configuration (compiler generated)

- Generate macro definitions by running the appropriate compiler command
  - E.g. gcc -E -P -v –dD file.c > macros
  - Synchronize the project so that “macros” is copied to local
- From project properties
  - **C/C++ General > Preprocessor Include Paths, Macros, etc.**
  - Click on “Entries” and select “CDT User Setting Entries”
  - Click “Add”
  - Choose “Preprocessor Macros File”
  - Navigate to and select the file from the project

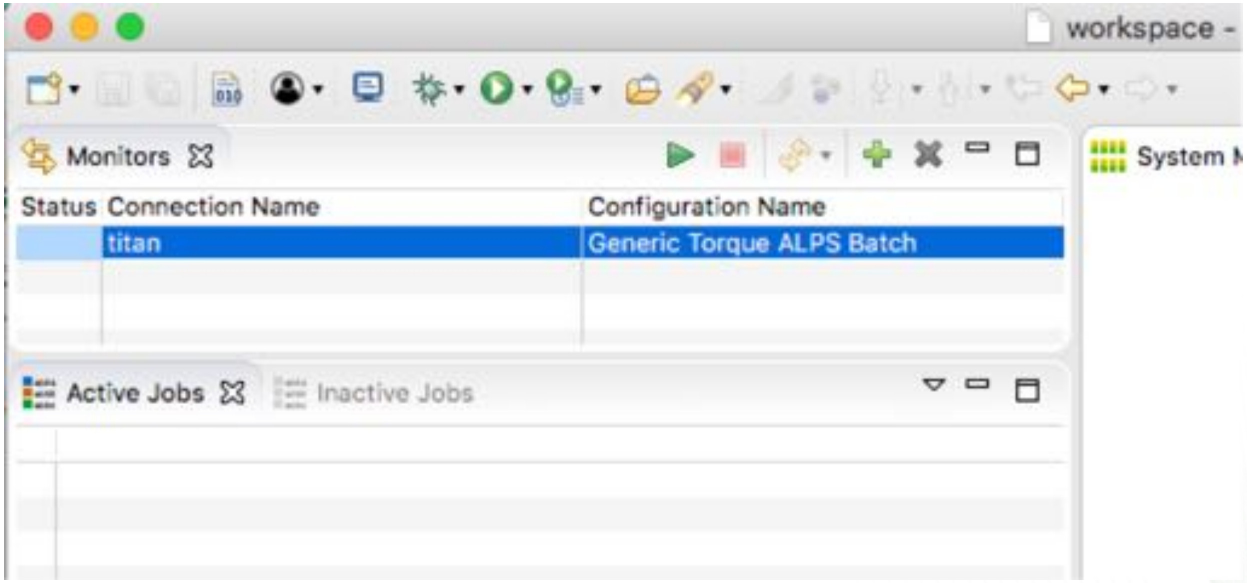
Manual Configuration

- From project properties
  - **C/C++ General > Preprocessor Include Paths, Macros, etc.**
  - Click on “Entries” and select “CDT User Setting Entries”
  - Click the “Add” button
  - Add an include directory or preprocessor macro using the dialog
- Unfortunately only one include or macro can be entered at a time

System Monitoring

Managing Systems

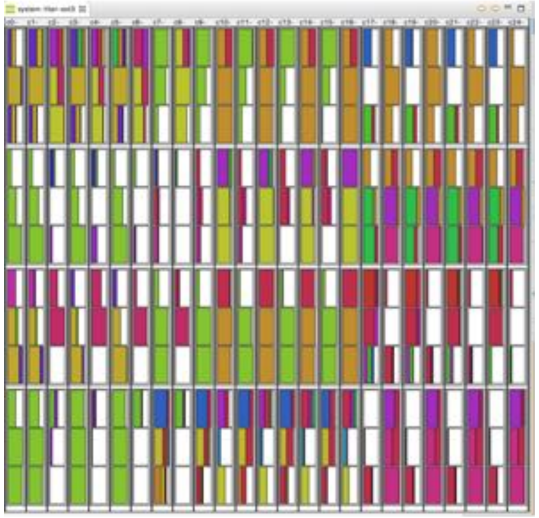
- Switch to the System Monitoring perspective
- Add/delete systems in the “Monitors” view

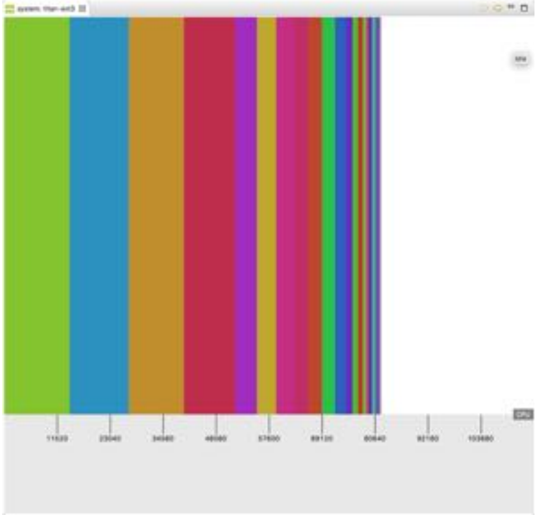


Status	Connection Name	Configuration Name
	titan	Generic Torque ALPS Batch

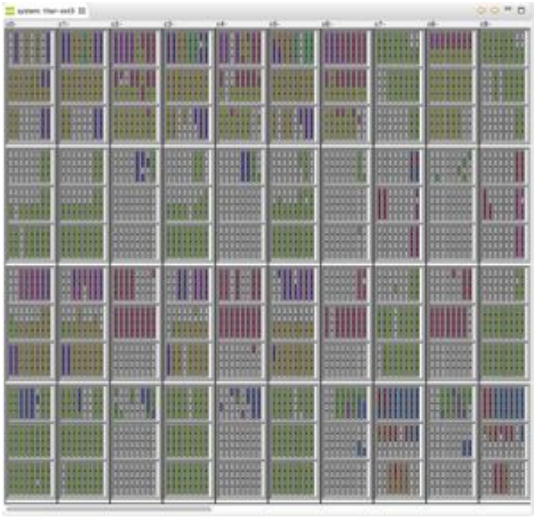
System View

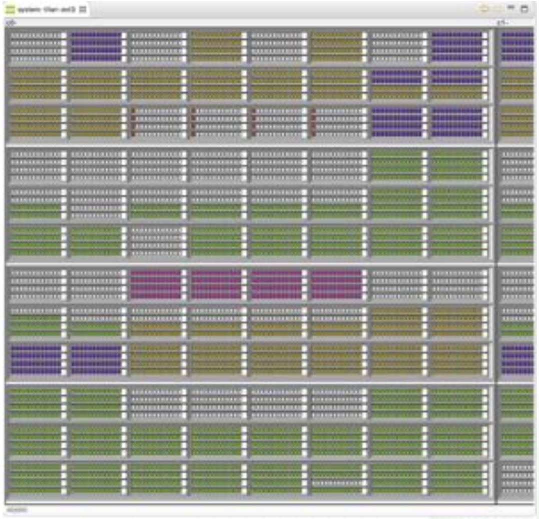
Zoom Out





Zoom In





Environment Modules

Using Environment Modules

- For the build:
  - Open project properties
  - Click on “Synchronize”
  - Select the remote configuration
  - Check the “Use an environment management system to customize the remote build environment”
- When submitting job:
  - Open run configuration for target machine
  - If supported, find the ”Modules to Load” entry and click “Configure”
  - Check the “Use an environment management system to customize the remote build environment”