

Standard Model of Scientific Computing

All users must do these things...

Define the Problem



Write an input file in a format reminiscent of a dead language

Run the Simulator



Manually launch jobs on impressively terrifying machines

Analyze Output

```
01100010
01101001
01101110
01100001
01110010
01111001
```

Analyze simulation output in its most raw and unlimited form

Archive Output



Store data... somewhere!

Super-users think these are easy tasks, but most users are overwhelmed!

A cooler model of Scientific Computing

Define the Problem



Write an input file in a format reminiscent of a dead language

Run the Simulator



Manually launch jobs on impressively terrifying machines

Analyze Output

```
01100010  
01101001  
01101110  
01100001  
01110010  
01111001
```

Analyze simulation output in its most raw and unlimited form

Archive Output



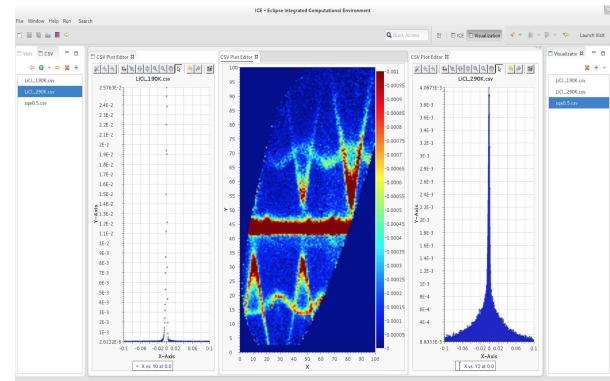
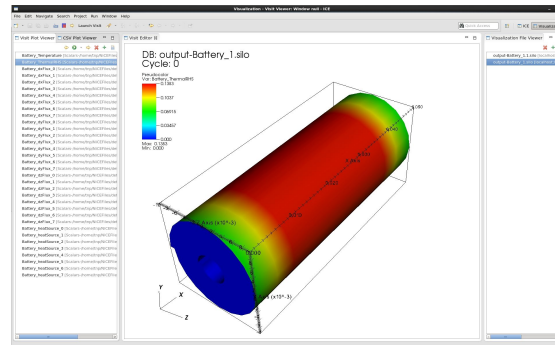
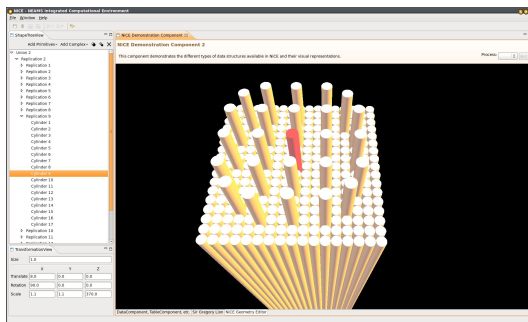
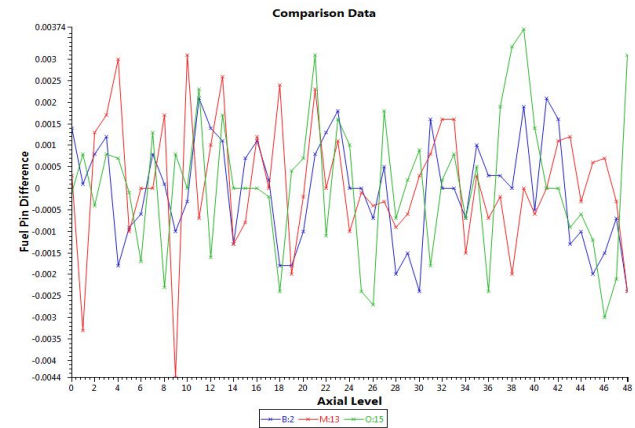
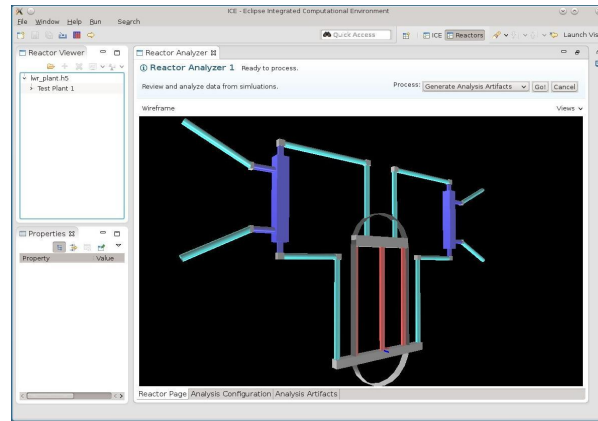
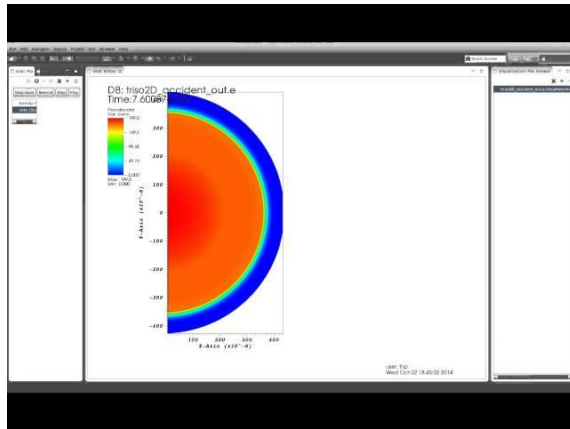
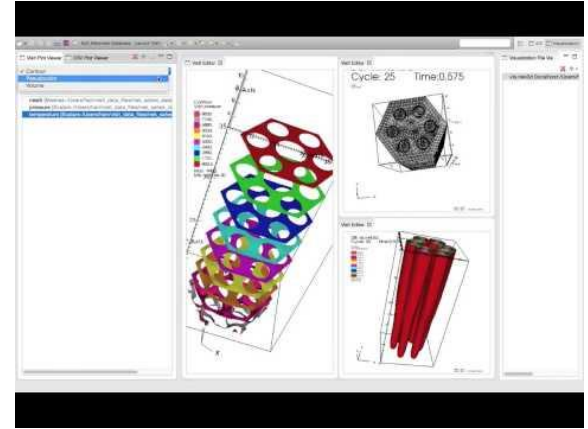
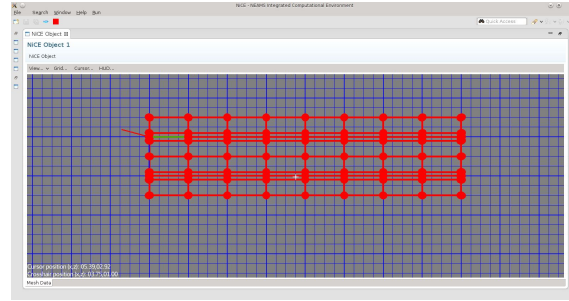
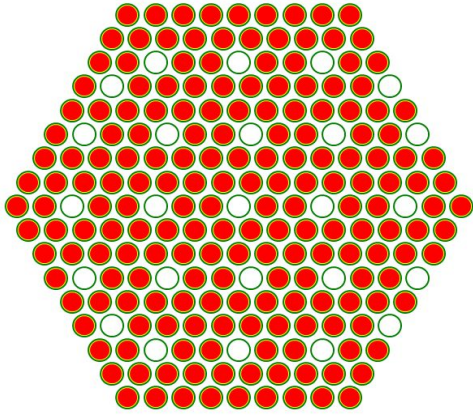
Store data... somewhere!

It would be better to have a computer program handle all of that...



Most of the stuff we need to do can be encapsulated for ease of use and/or automated entirely with improvements.

What can it do in 9 pictures or less?



Where does it work?

Nuclear
Energy

Data
Analysis

Advanced
Manufacturing

More 3rd Party
Tools

Coming in FY16!

Batteries



Advanced
Materials

Quantum
Computing

Astrophysics

Basic 3D Geometry
and 2D Mesh Editing

Usability in Modeling and Simulation

How I send messages around the World:



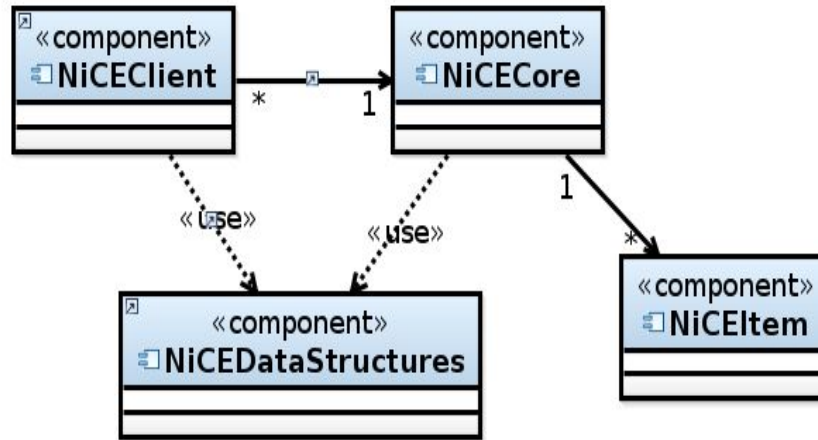
How I send messages to my code:

```
./xolotl ../benchmarks/he-W_2067.txt --  
handlers dummy --petsc -ts_final_time  
1000 -ts_final_time 1000 -  
ts_adapt_dt_max 10 -  
ts_max_snes_failures 200 -pc_type  
fieldsplit -pc_fieldsplit_detect_coupling -  
fieldsplit_0_pc_type redundant -  
fieldsplit_1_pc_type sor -ksp_monitor -  
ts_max_steps 3
```

Really?!

How does it work? Plugins!

Components of NiCE

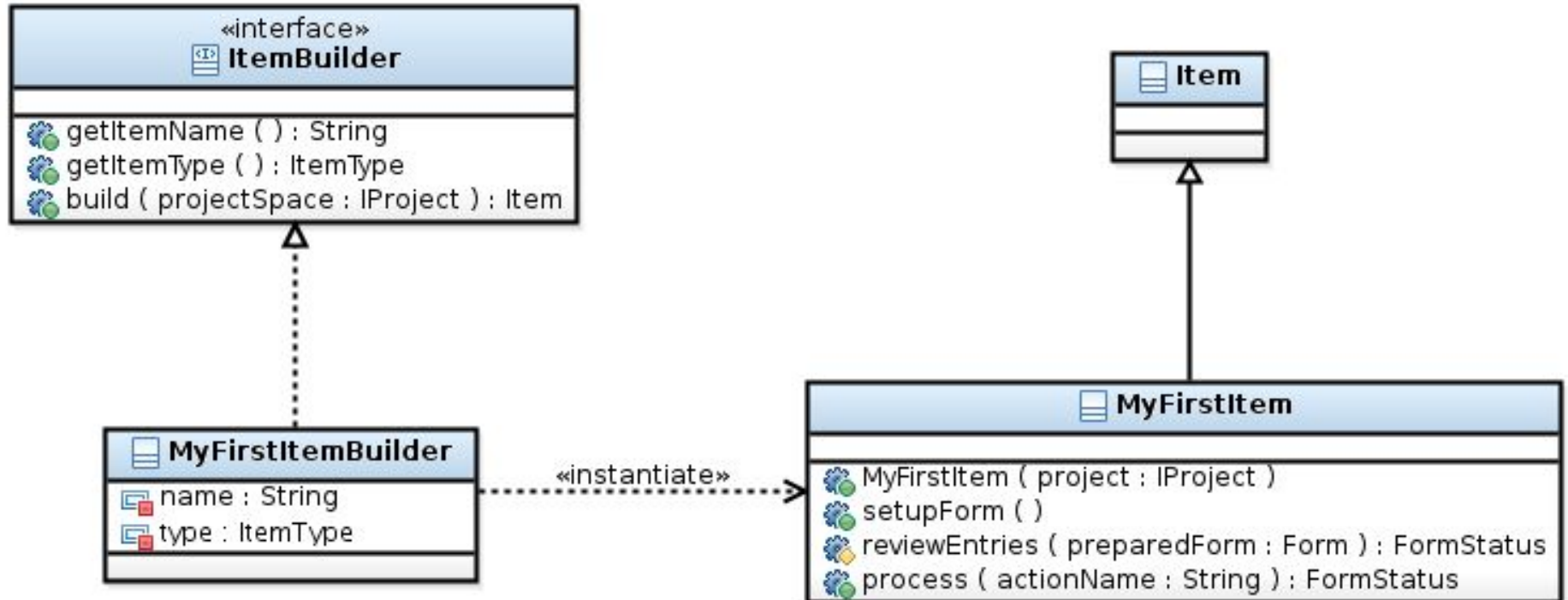


Plugins are:

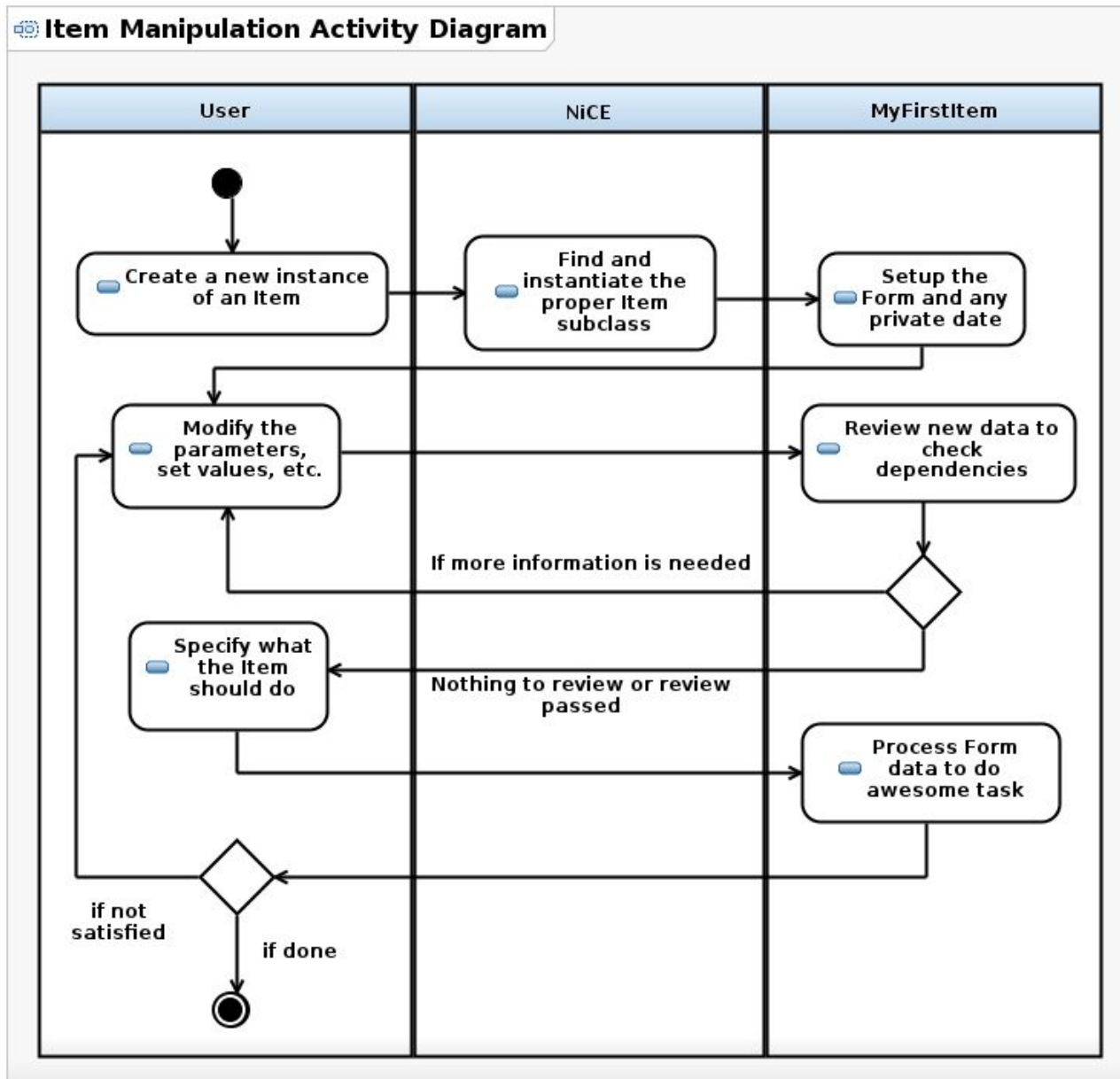
- Dynamic Services - Completely reusable components!
- “Item” Subclasses - Most of the work is already done by the platform
- Self-contained, business logic - **ONLY** your code, not UI, etc.
- Tools - Reusable components, tools, or things other

How does it work? Part 3

MyFirstItem Class Diagram



How does it work? Part 4



How does it work? Part 5

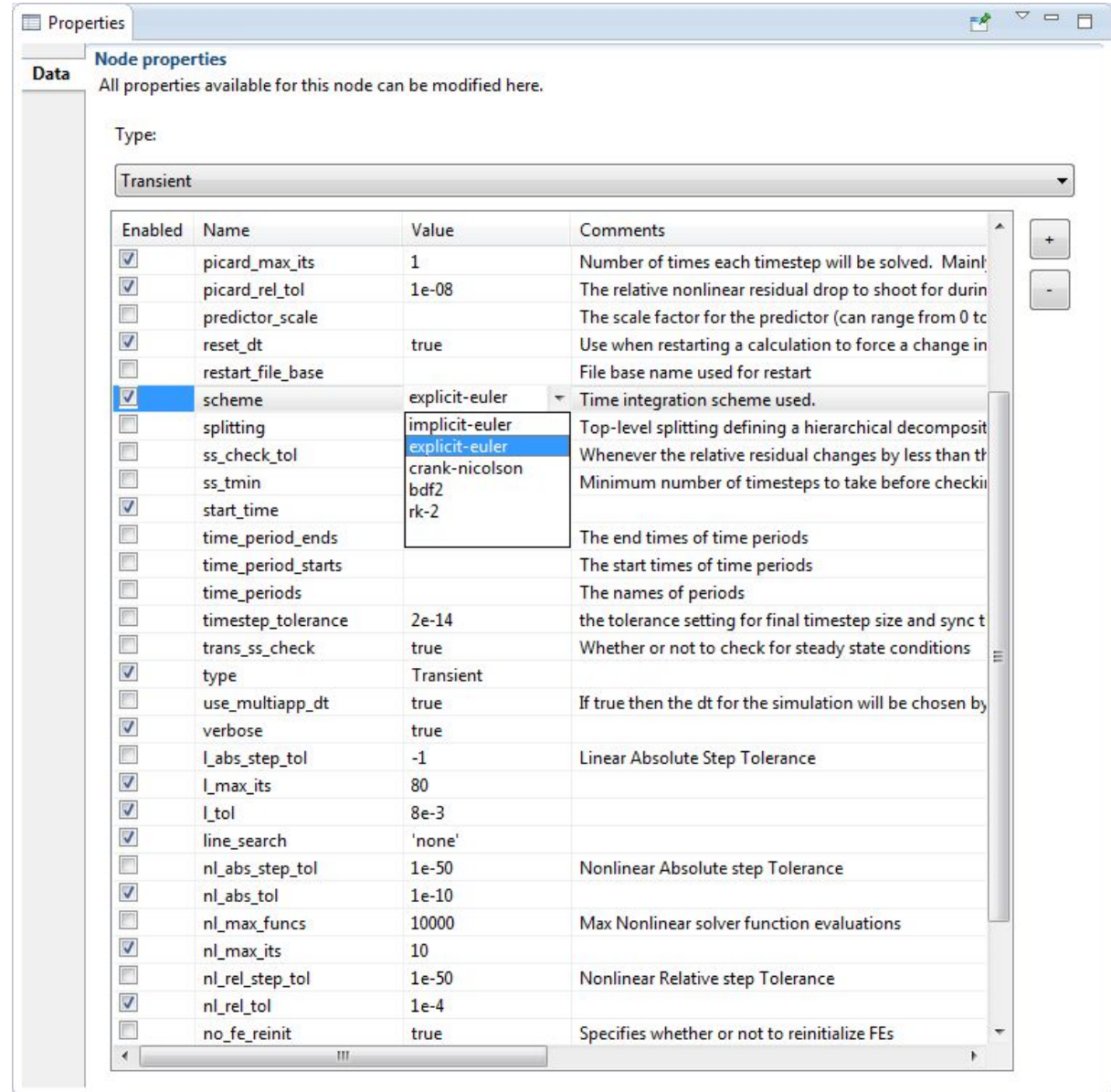
Things to keep in mind:

- You only write business code
- UI and marshalling are provided by the platform (unless you want to extend it)
- Codify only what is needed; reuse what you already have (preprocessors, etc.)

How does it work? Part 6

All of the data structures are backed by sophisticated tools so you deal with your domain.

Standardization for the win!



Different views of the same data

```
<Entry defaultValue="1.7899" ready="true" changeState="false">
  <AllowedValueType>Undefined</AllowedValueType>
</Entry>
```

```
entry1 = new Entry() {
    protected void setup() {
        allowedValues = new ArrayList<String>();
        allowedValues.add("0");
        allowedValues.add("50");
        defaultValue = "1";
        allowedValueType = AllowedValueType.Continuous;
    }
};
entry1.setName("Generic 1");
```

All of these are logically equivalent because of the standardization!

▼ Input File(s)

This section contains the name of the file(s) used by this job.

Input File: Caebat_Model_1.conf

▼ Electrical Properties

Electrical properties and settings

Current Flux:

3602431

Enabled	Name	Value	Comments
<input type="checkbox"/>	predictor_scale		The scale factor for the predictor (can range from 0 to 1)

The Eclipse ICE Item Project Generator

This tutorial will teach you how to create new ICE Item Projects

You will need:

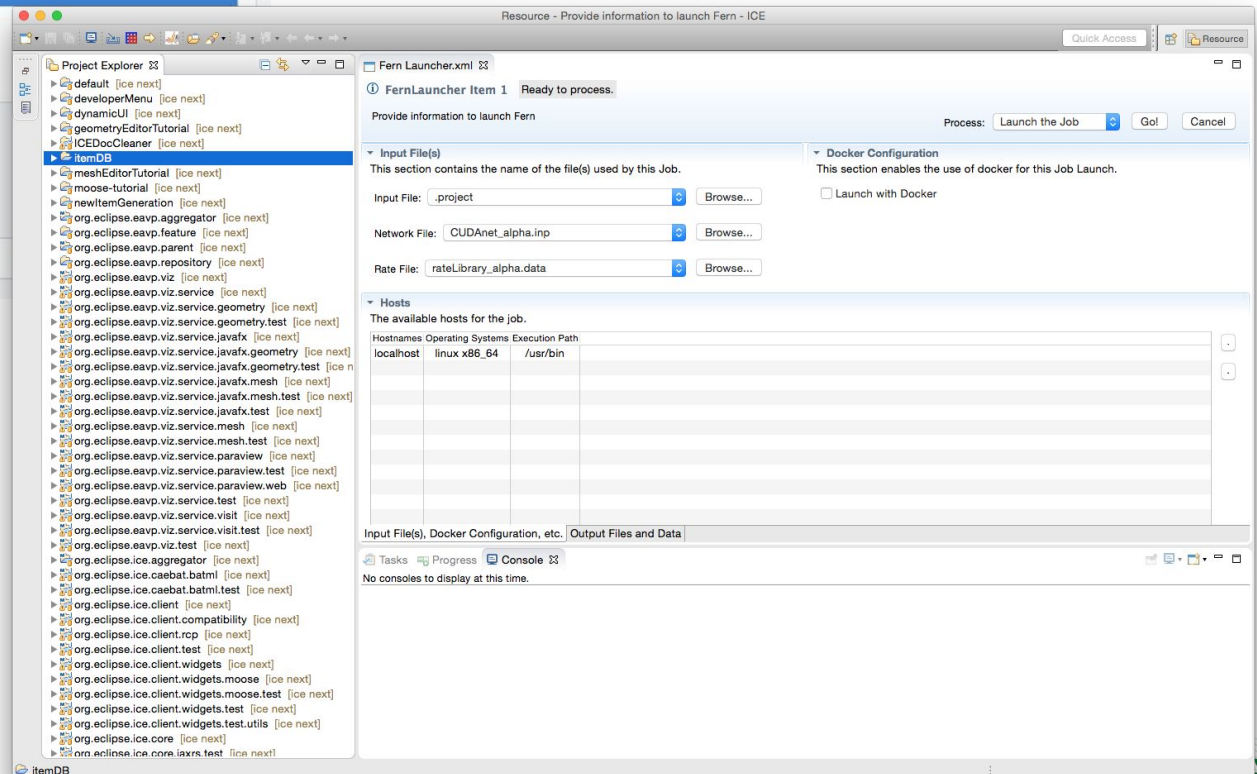
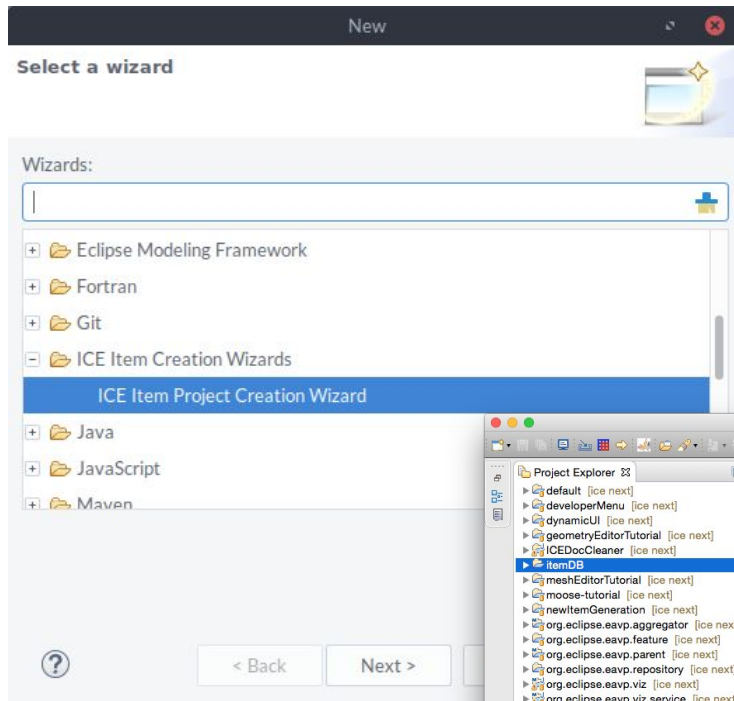
- Experience writing simple Java code
- Docker Image with Fern science code

You will learn:

- How to create an Item project
- How to create Model and JobLauncher Items
- How to use those Items in ICE.

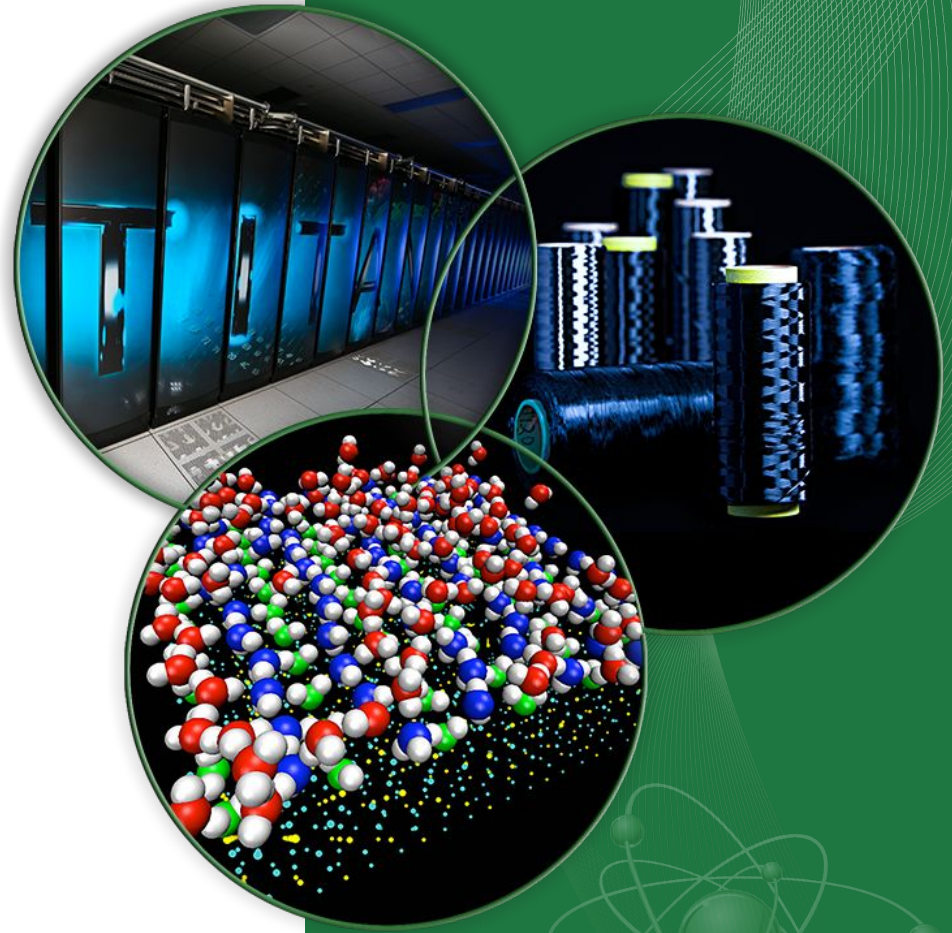
The Eclipse ICE Item Project Generator

*Enable users to go from
no code, to working ICE
Items for their scientific
application*



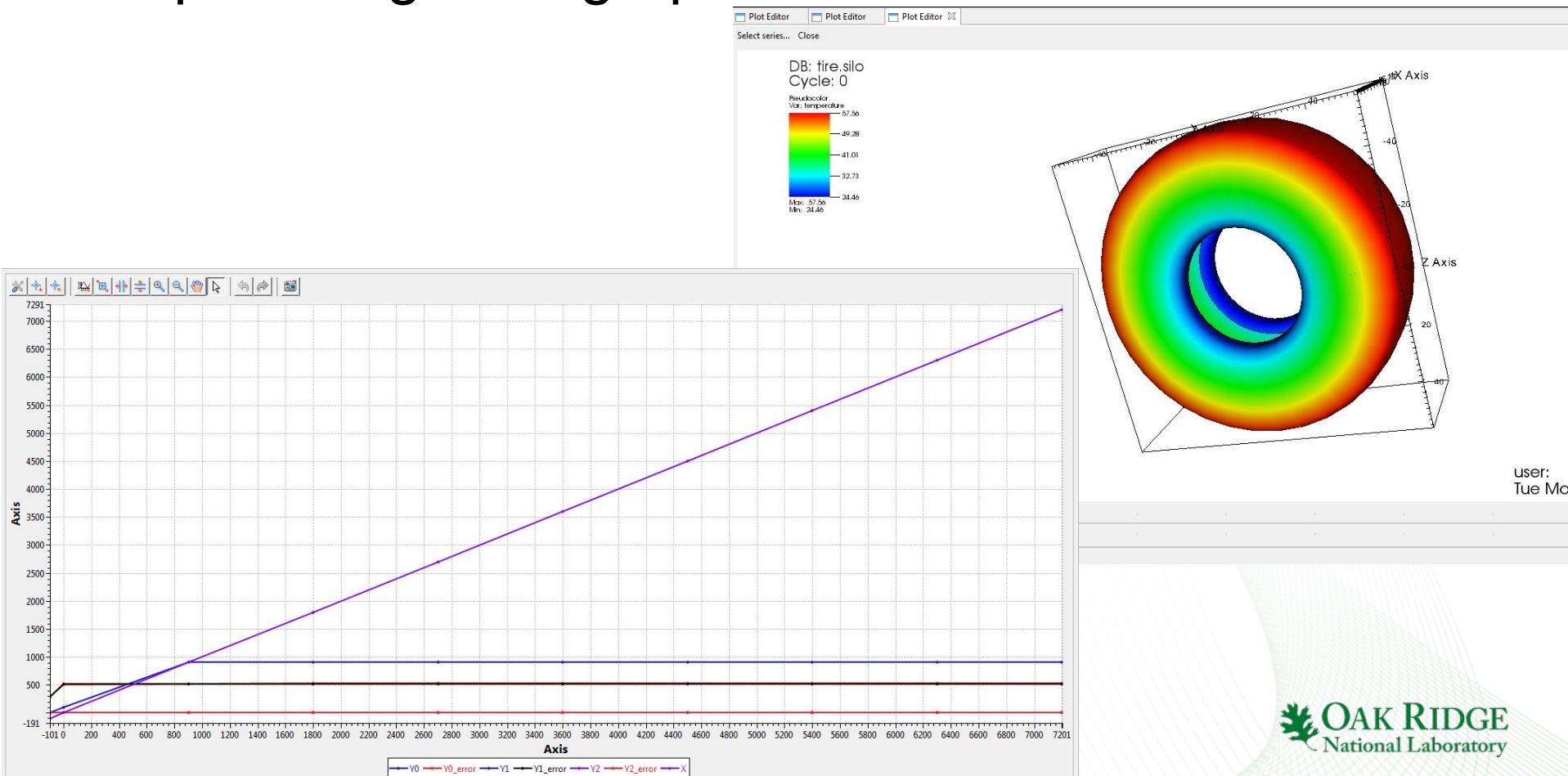
Visualization Components

Resource
Components,
Geometry Editor
Components, and
Mesh Editor
Components



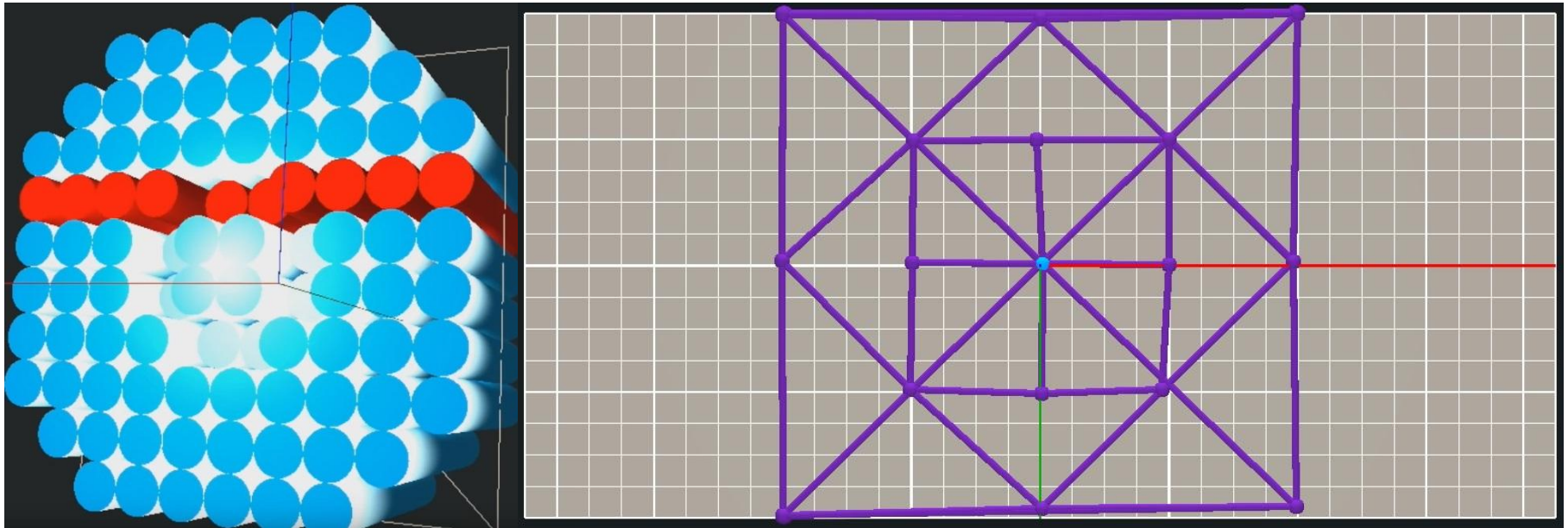
Resource Component

- Add a component to your item which will display output using CSV graphs or VisIt 3D visualization.

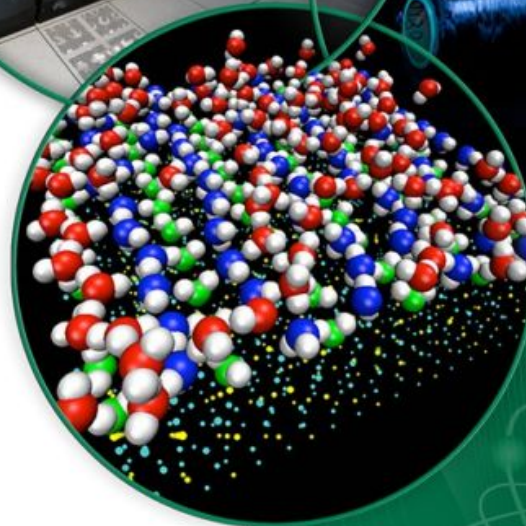
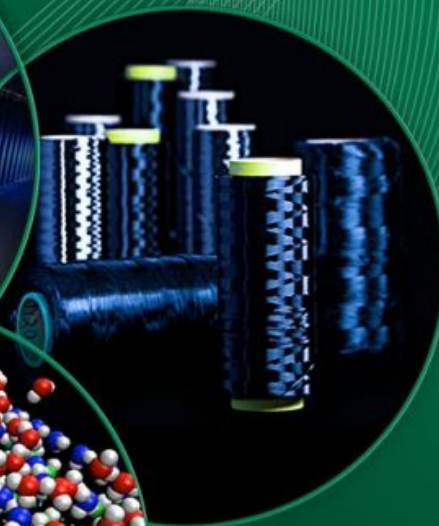


Editing 3D Structures

- Add a Geometry Component to your Item to edit 3D shapes.
- Add a Mesh Component to your Item to create polygonal meshes.



Scripting ICE with EASE

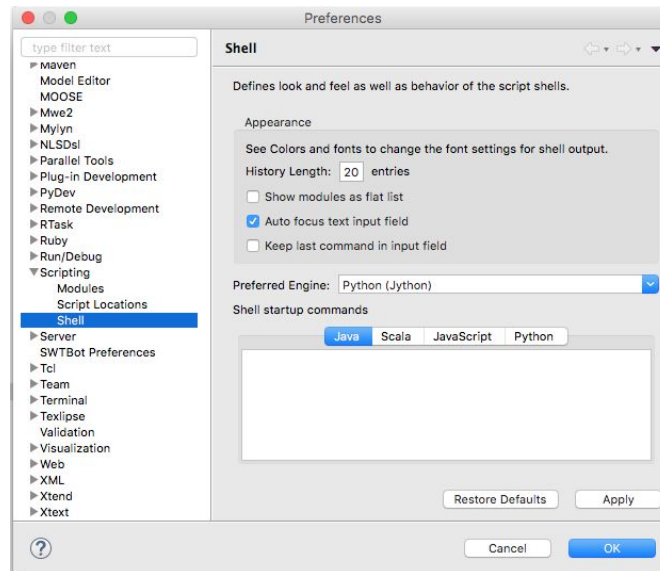
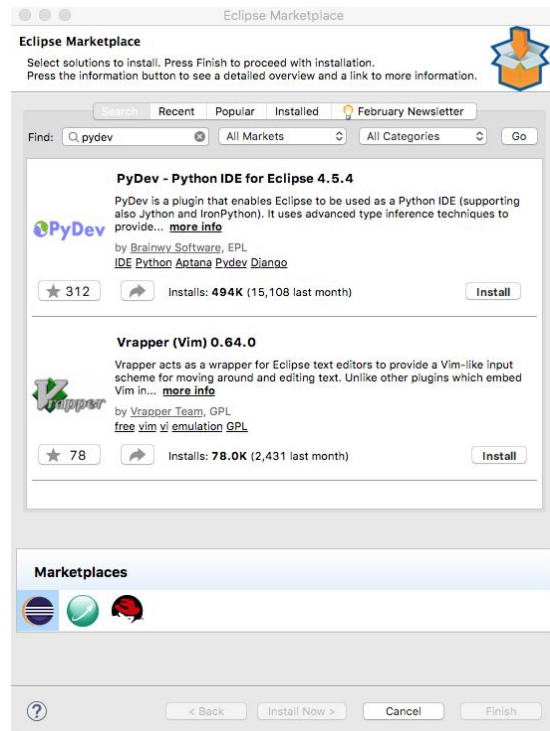
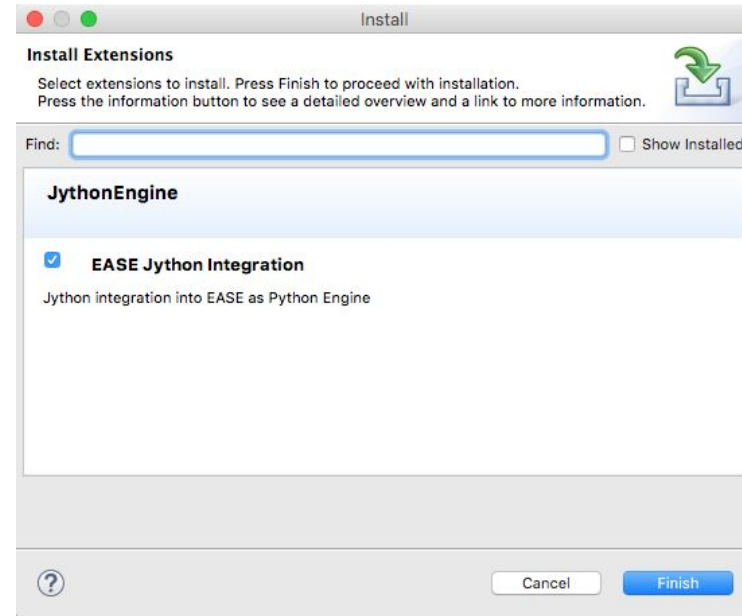


Outline

- Install and Configure EASE
- Creating and Running Scripts
- Using the Sample Scripts

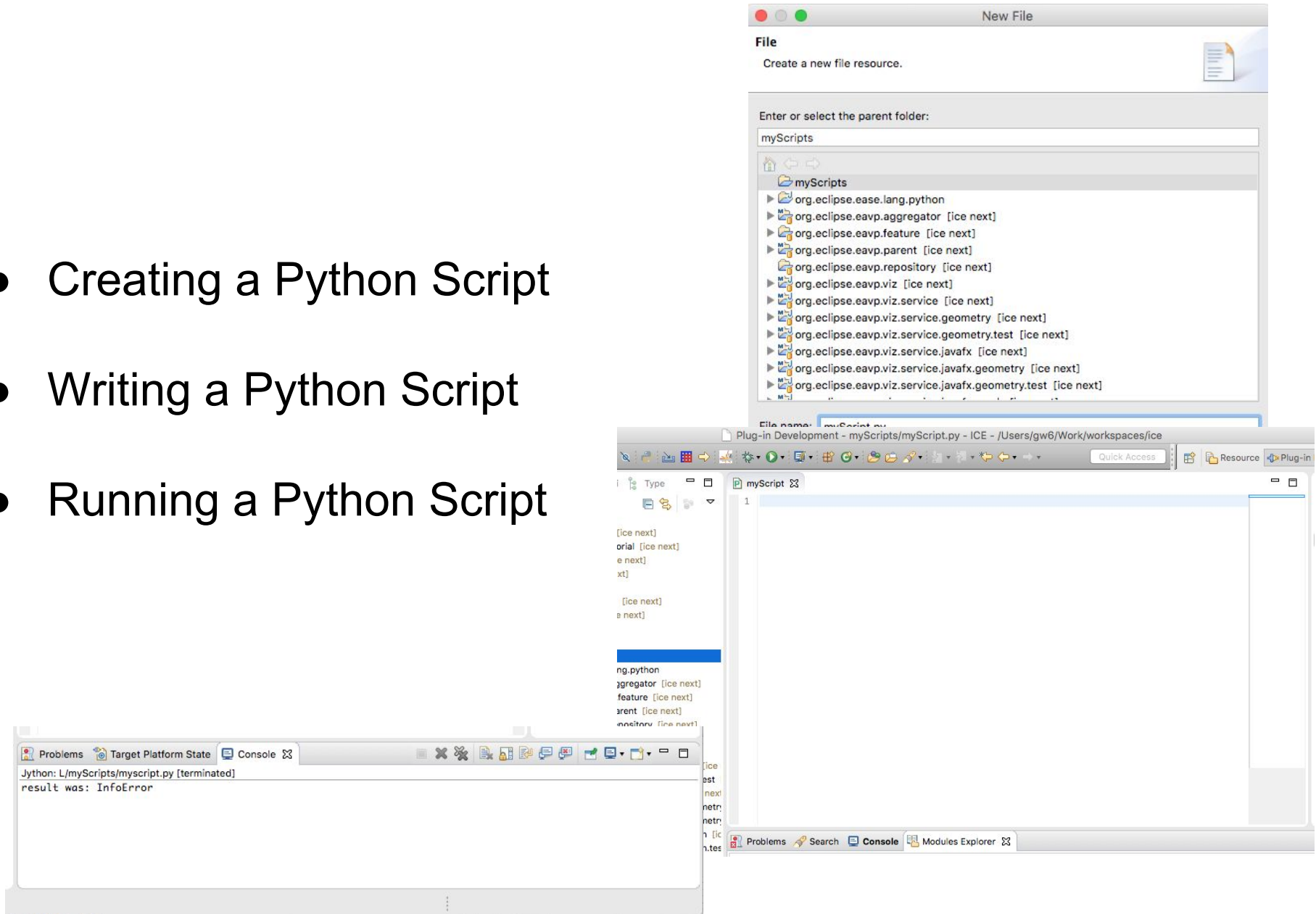
Installing and Configuring EASE

- EASE Jython Installation
- PyDev Installation
- EASE Configuration



Creating and Running Scripts

- Creating a Python Script
- Writing a Python Script
- Running a Python Script



Using the Sample Scripts

- `createAndEditPython.py`
- `createAndProcessPython.py`
- `iterateChangeParameterPython.py`
- `listFromScratchPython.py`

ICE Dynamic UI Tutorial

This tutorial will teach you how to change the ICE UI with only some small amounts of text.

You will need:

- Experience creating Eclipse plugins
- Experience writing UI code with SWT
- Experience creating an ICE Item

You will learn:

- How to create news pages in ICE's FormEditor
- How to create new EntryComposites in ICE's Form editor
- How to publish UI updates to the Eclipse 4 Context

For example...

Change this...



A screenshot of a software interface. It features a light blue header bar with a downward arrow and the text "Data". Below the header, the text "Some Data" is displayed. Underneath, the label "Data Entry:" is followed by a single-line text input field with a vertical cursor at the beginning.

... into this...



A screenshot of the same software interface, but with modifications. The header bar and "Some Data" text remain. However, the "Data Entry:" label is now followed by a button labeled "My button" and the text "in new Entry widget.".

... without changing the way ICE draws the UI.

The Eclipse ICE Developer Menu

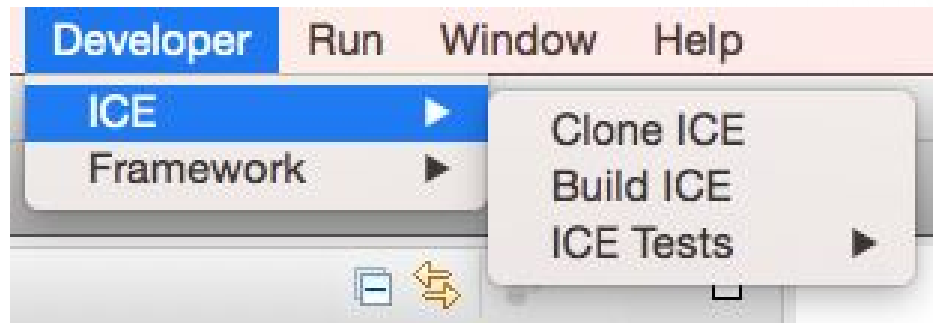
This tutorial will teach you how extend the ICE Developer Menu

You will need:

- Experience creating Eclipse plugins
- Experience adding Extensions

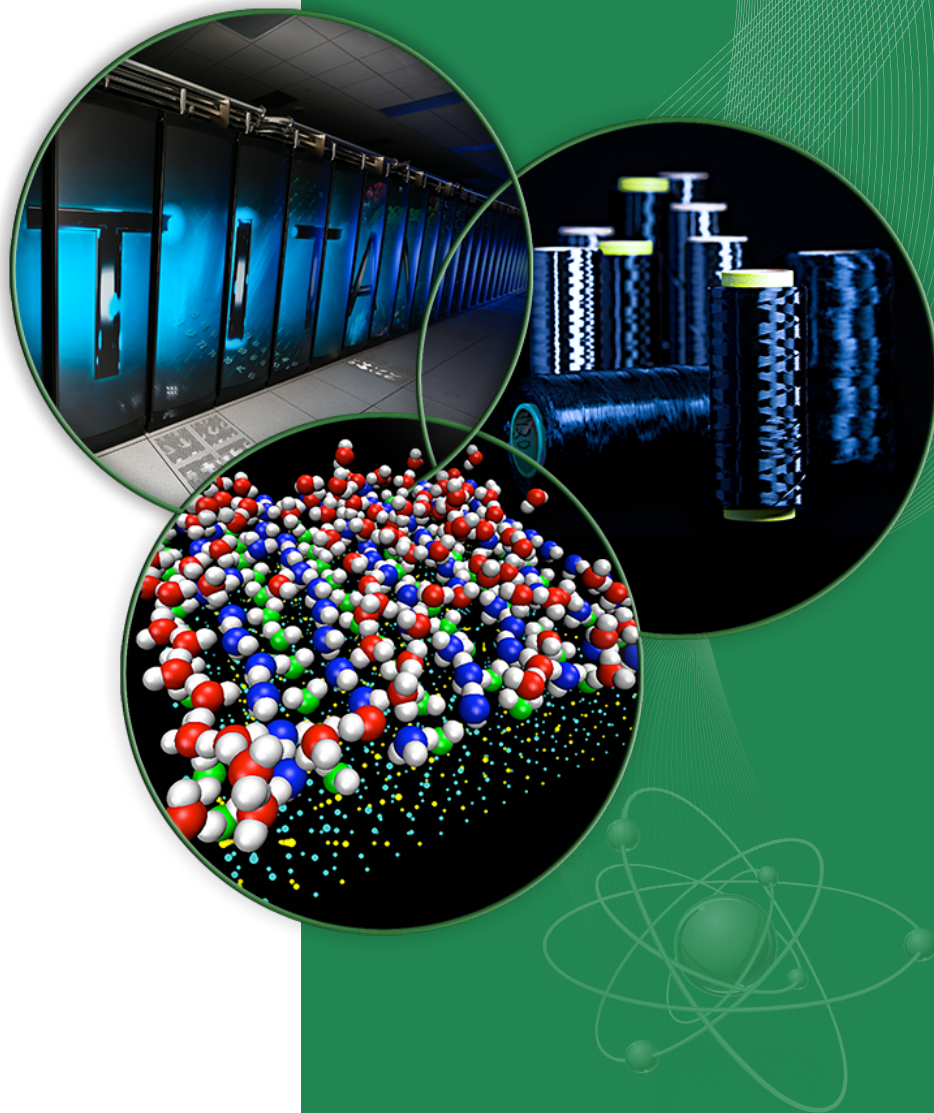
You will learn:

- How extend the Developer Menu with custom actions



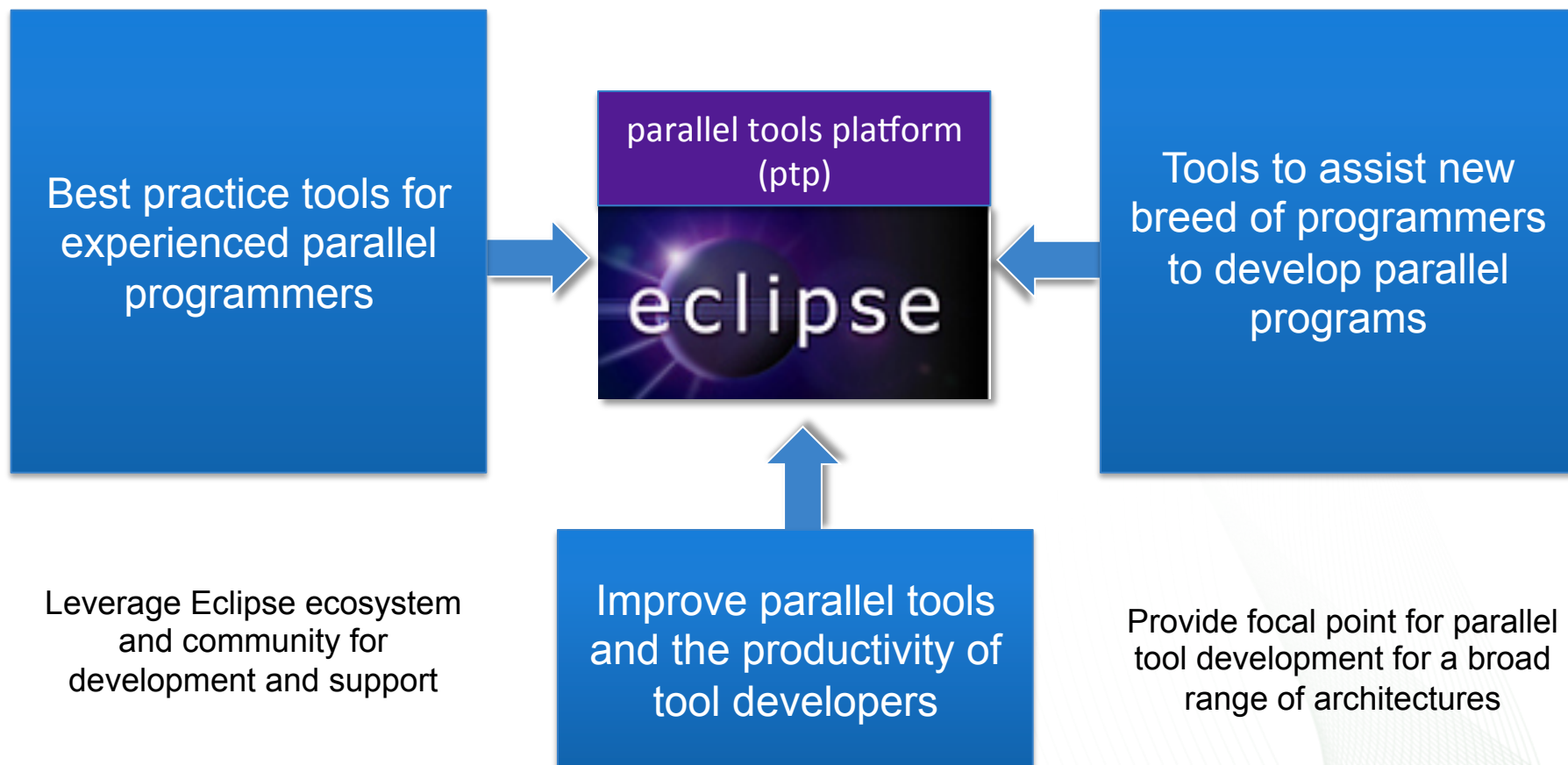
Eclipse for Science

How the Parallel Tools Platform can enhance the development of scientific applications



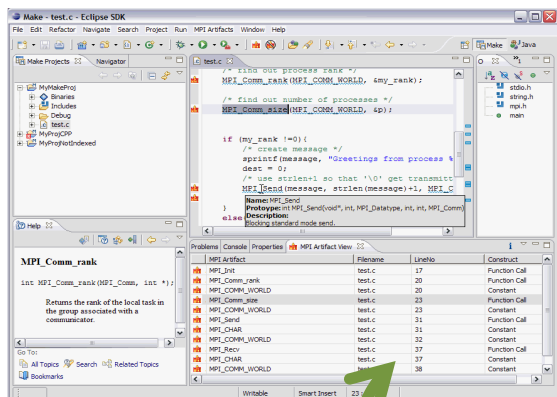
Parallel Tools Platform

Enabling Parallel Application Development

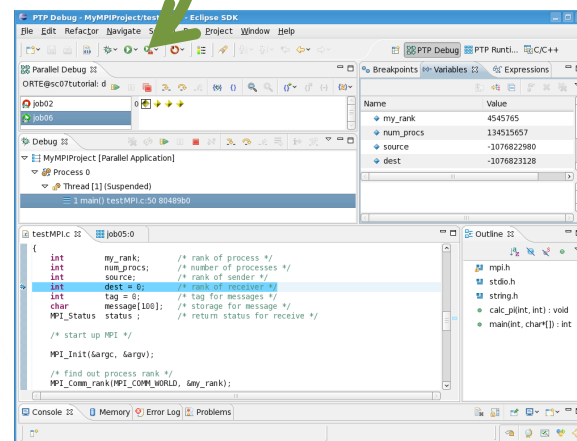
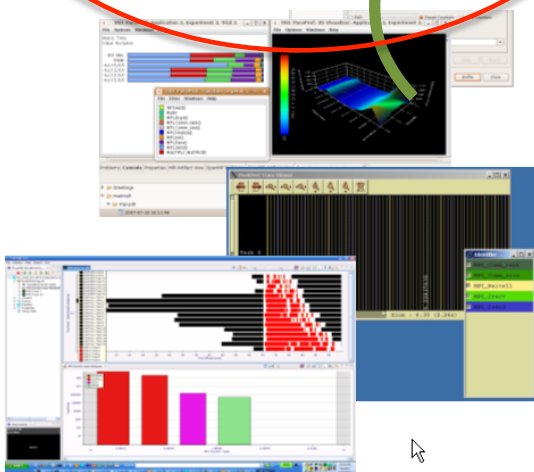
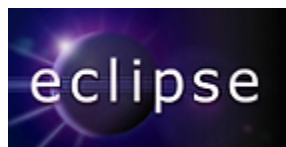
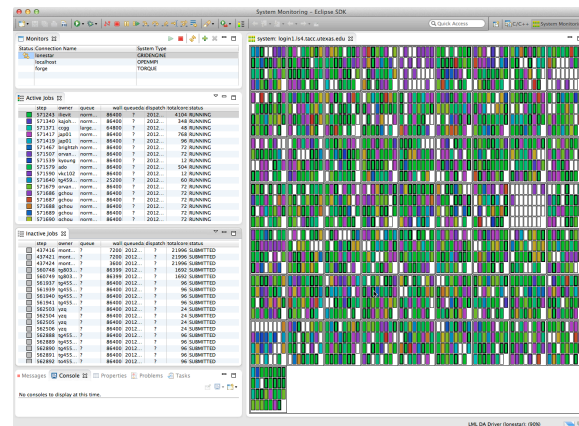


PTP Application Development Cycle

Coding & Static Analysis



Application Execution



Dynamic & Performance Analysis

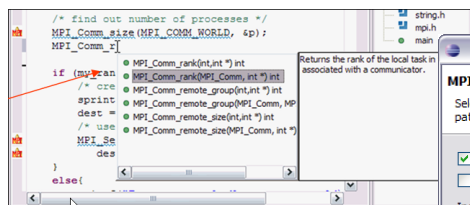
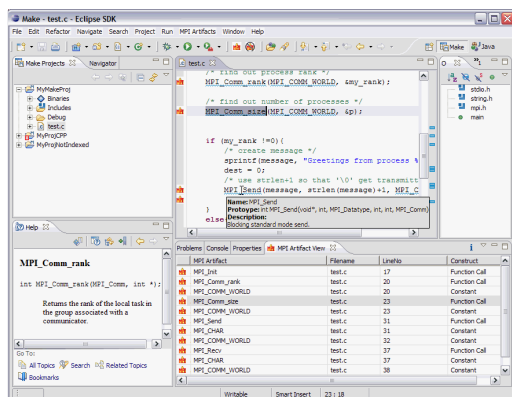
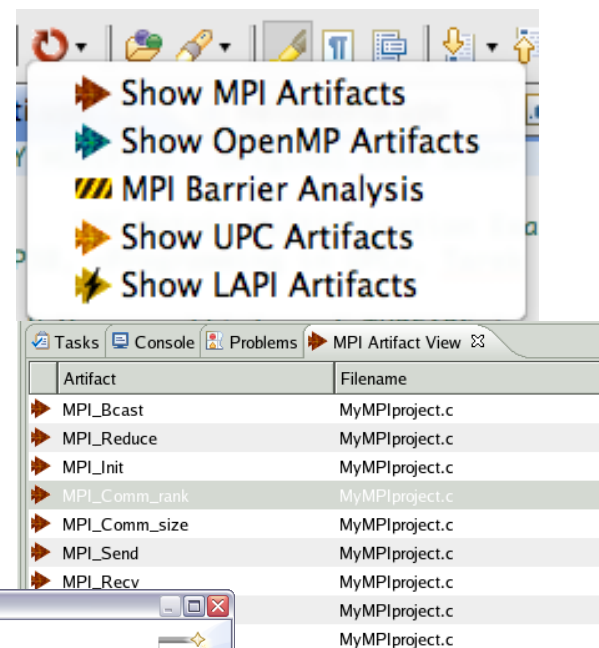
Application Debugging

Coding & Static Analysis

- Eclipse provides a wide variety of coding assistance tools
 - Project management, Editing and formatting, Navigation, Advanced searching, Refactoring, Version control
- C/C++ Development Tools (CDT)
 - Standard (Makefile) and managed builders, Support for arbitrary toolchains, Visual debugging using GDB, High level views (outline view, call hierarchy, type hierarchy, include browser), Refactorings
- Parallel Tools Platform (PTP)
 - Fortran, New project wizards (MPI, OpenMP) Content Assist, Hover help, Built-in API descriptions (MPI, OpenMP, LAPI, UPC), Location of parallel “artifacts” in code (MPI, OpenMP, PAMI, and UPC), Barrier analysis, Deadlock detection
- Python Development (PyDev)
 - Code completion, type hinting, refactoring, debugging, interactive console, unittest, code coverage, Django integration

Coding & Static Analysis

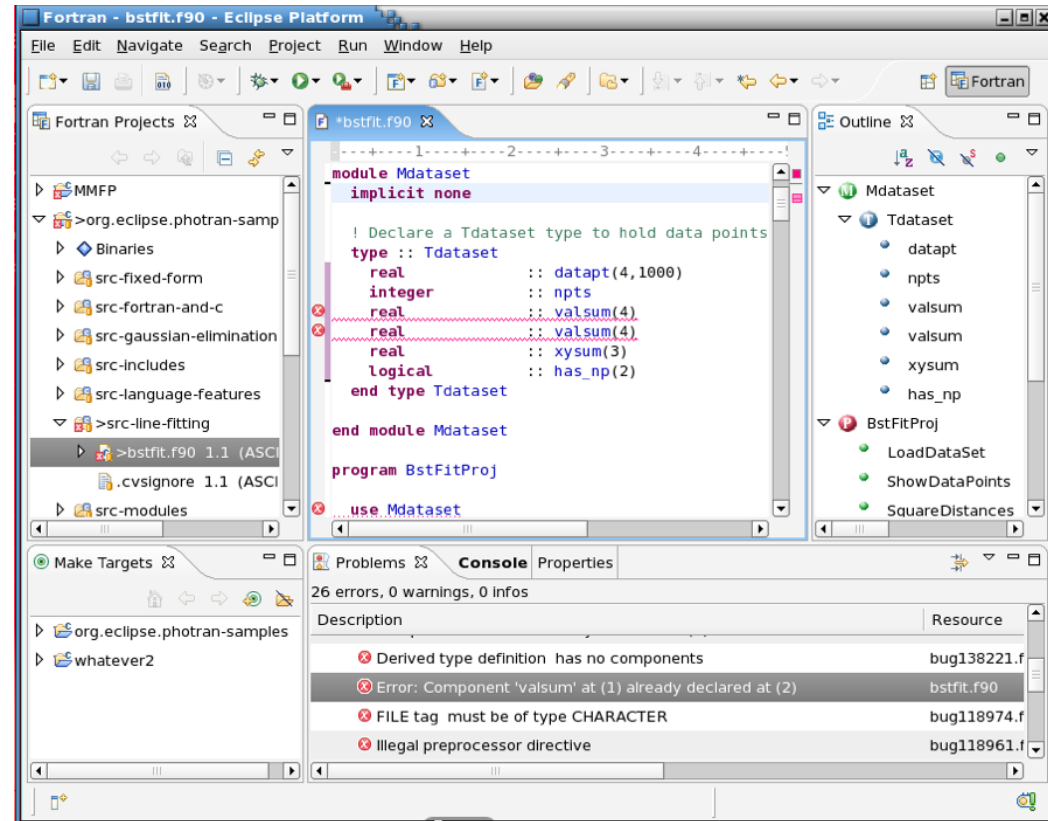
- Assistance tools to increase productivity of parallel programmers
 - New project wizards (MPI, OpenMP)
 - Content Assist (command/API completion), hover help, built-in API help descriptions in an html help “view” (MPI, OpenMP, LAPI, UPC)
 - Location of parallel “artifacts” in code: MPI, OpenMP, and UPC



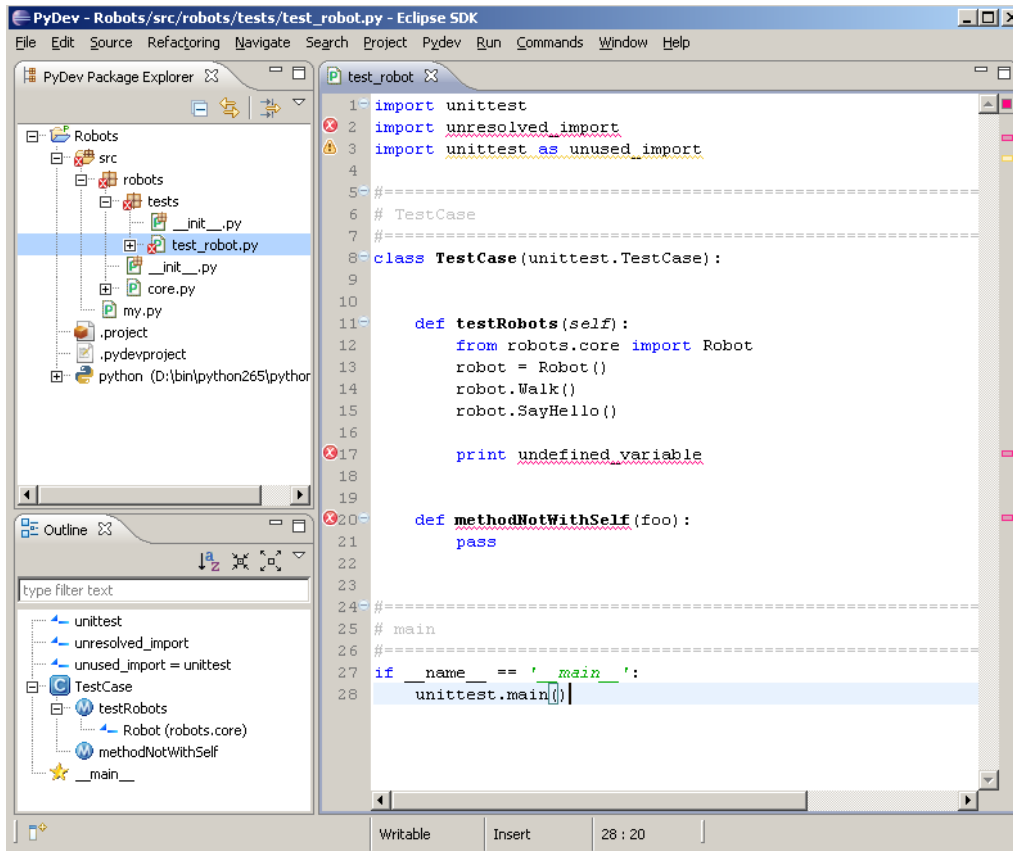
PHASE 3
HPC
PERCS

Fortran Development Tools

- Photran features:
 - Supports Fortran 77-2008
 - Syntax-highlighting editor
 - GUI interface to *gdb*
 - Makefile-based compilation
 - Compiler error extraction
 - Outline view
 - Open declaration
 - Fortran refactorings
 - C preprocessor support



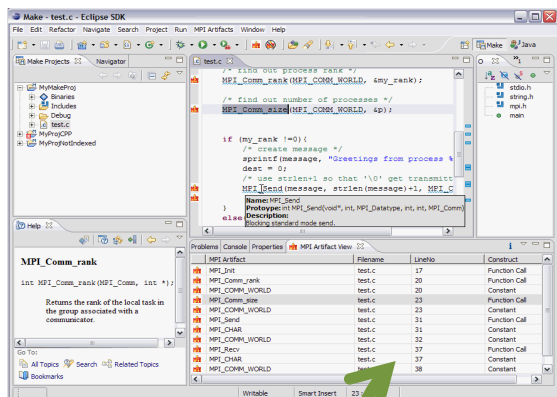
Python Development



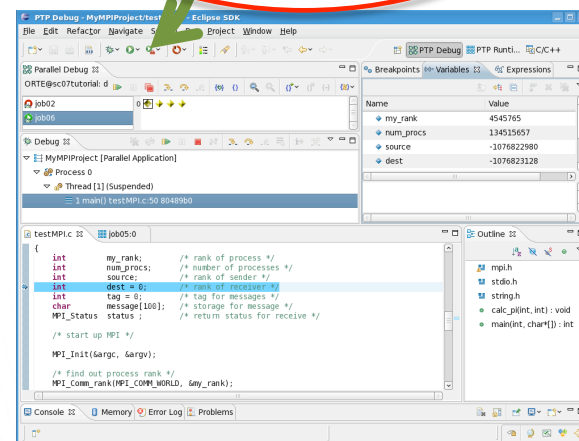
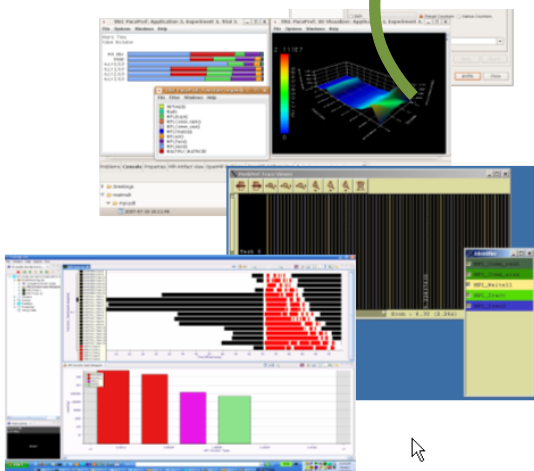
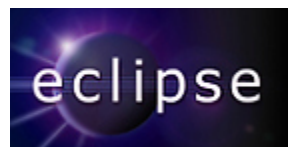
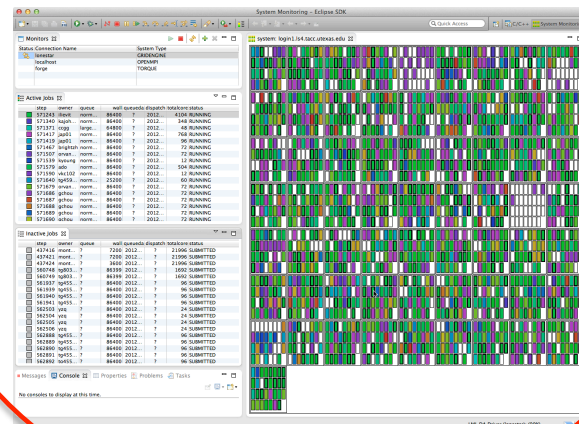
- PyDev is a Python IDE for Eclipse
- Create/manage Python modules
- Full array of Eclipse editing features for Python
- Python debugger
- Interactive console with Python interpreter
- Integration with Python unittest and code coverage modules

PTP Application Development Cycle

Coding & Static Analysis



Application Execution

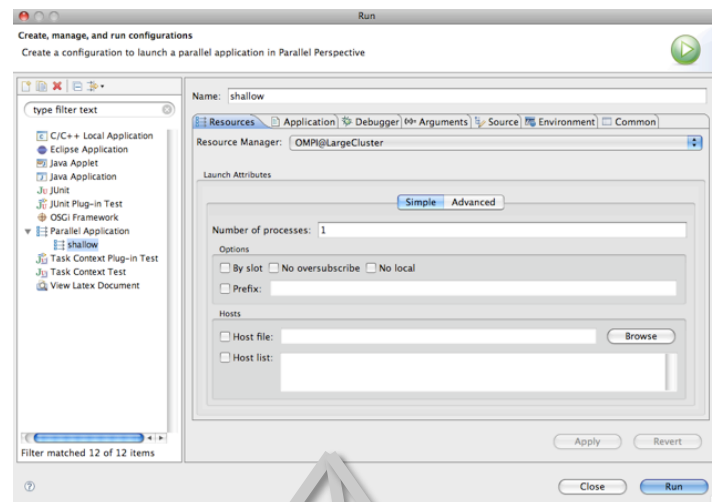
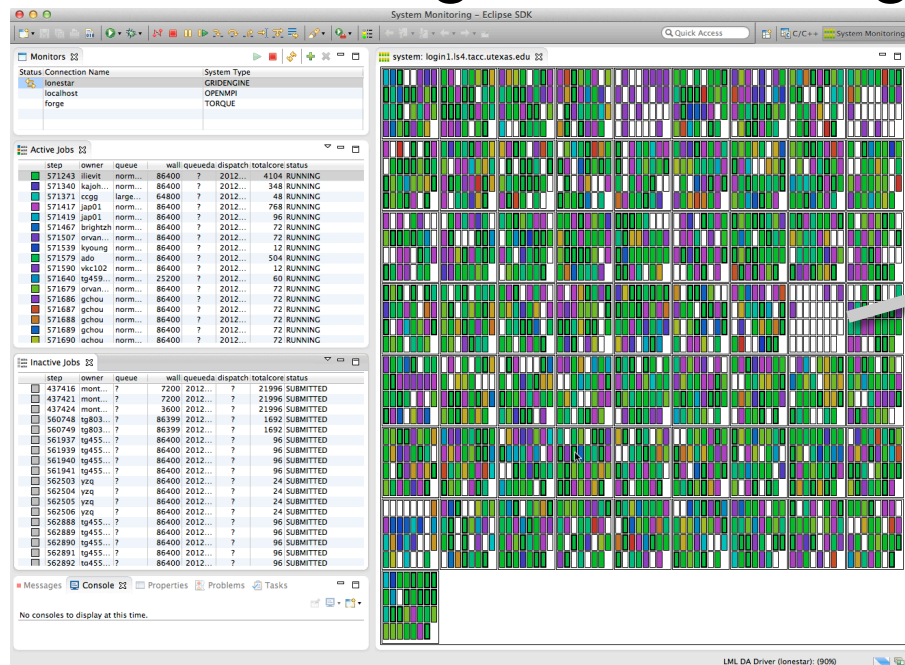


Dynamic & Performance Analysis

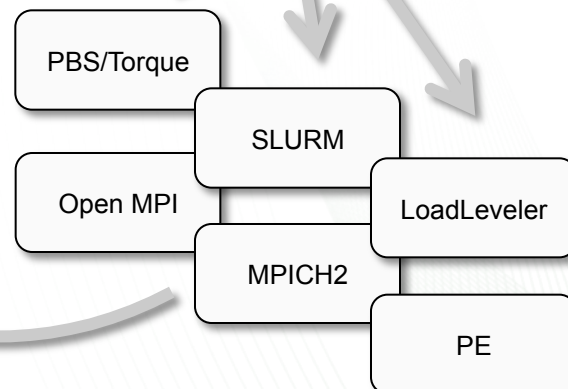
Application Debugging

Application Execution

- Launching & Monitoring



- Improves visibility into target system
- Single point of interface for launching and control
- Manages interaction with different runtime systems and job schedulers

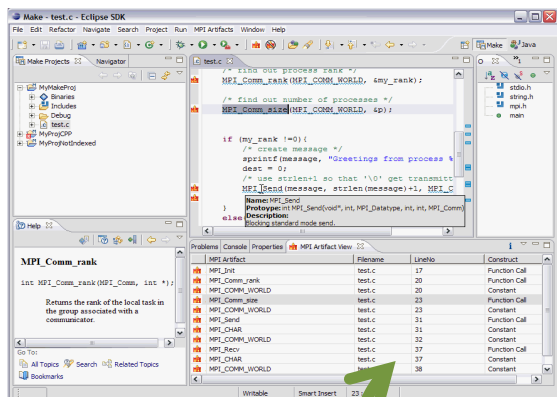


Application Execution

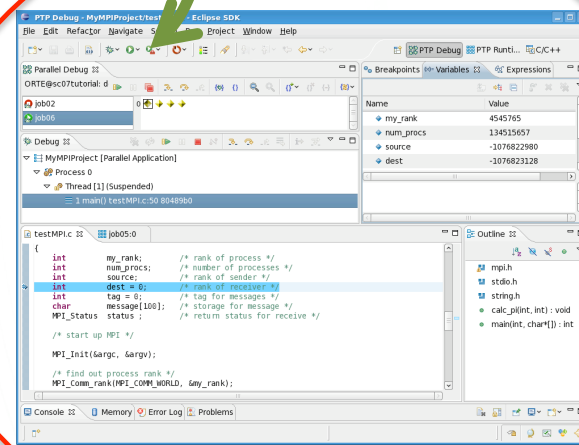
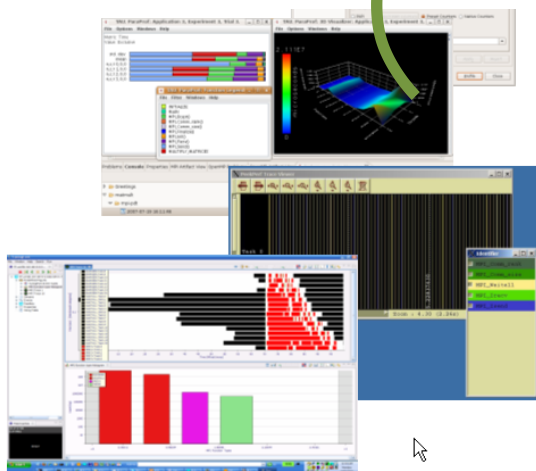
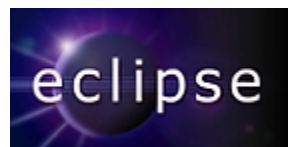
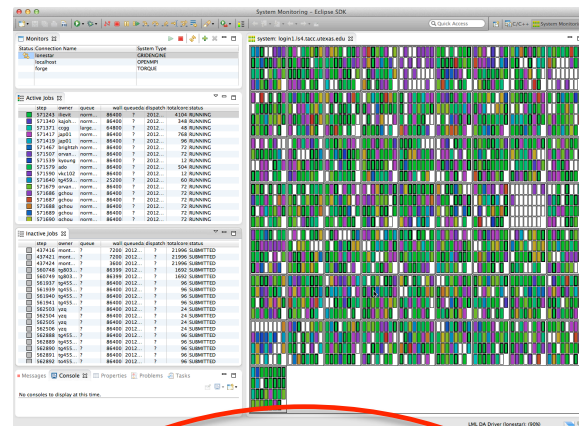
- Target Configuration Framework
 - Extensible framework for launching & monitoring
 - System and node status information
 - Job status (e.g. position in queue) & application status
 - Job submission & control
 - Debugger launch
 - Configuration files to support different resource managers
 - Job schedulers (LoadLeveler, PBS, Torque, SLURM, GridEngine)
 - Interactive runtimes (e.g. PE, Open MPI, MPICH2, MVAPICH)
 - Systems (AIX, Linux, Power, x86, BG/Q, Cray)
 - Local or remote system support
 - Command-line tools executed locally or via ssh connection

PTP Application Development Cycle

Coding & Static Analysis



Application Execution

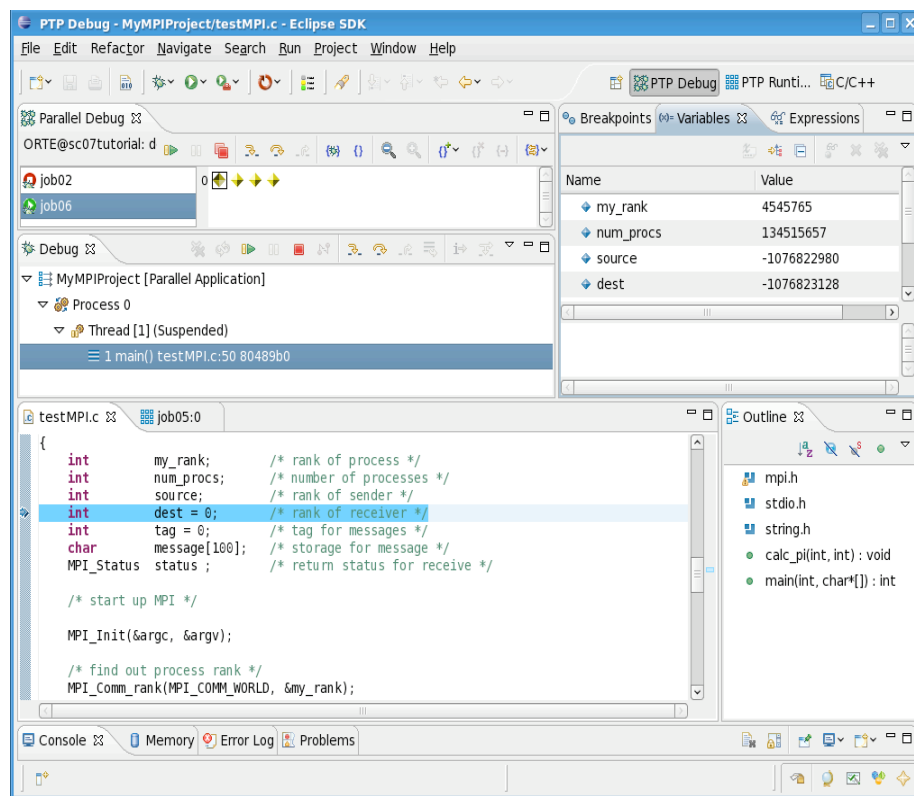


Dynamic & Performance Analysis

Application Debugging

Application Debugging

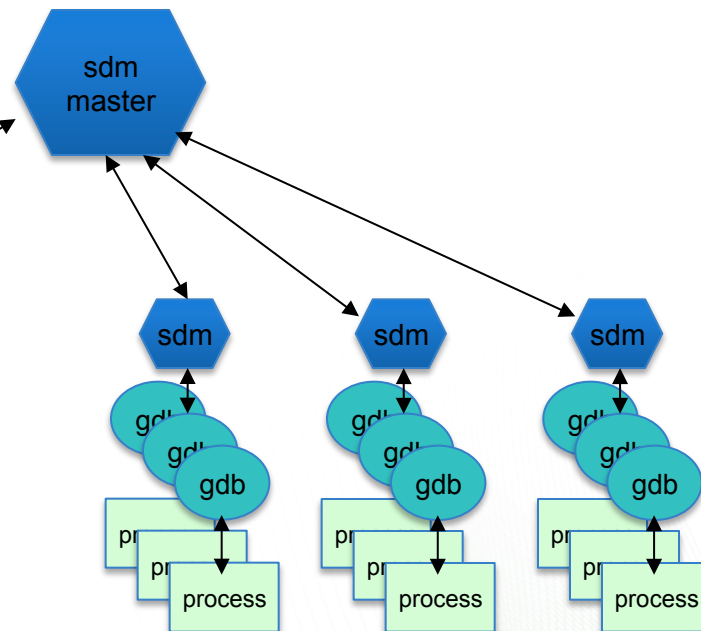
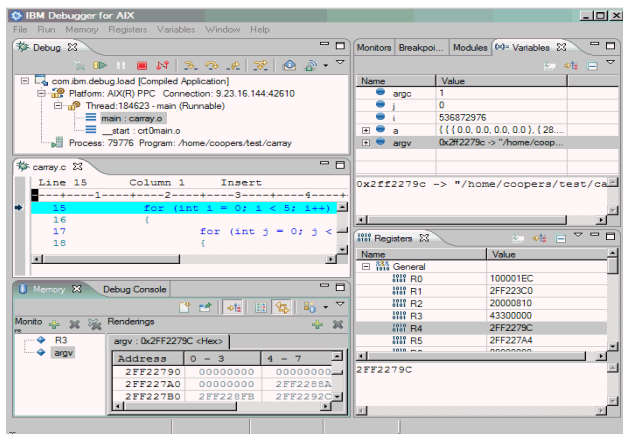
- PTP Parallel Debugger



- Mid-scale integrated debugger
- Tightly integrated with Eclipse
- Supports debugging multiple jobs simultaneously
- Utilizes backend debugger (e.g. gdb) for low level operations
- Targeted at SPMD programming models
- Supports mixed MPI & thread debugging
- Single process and group operations
- Platform for building new debugging paradigms

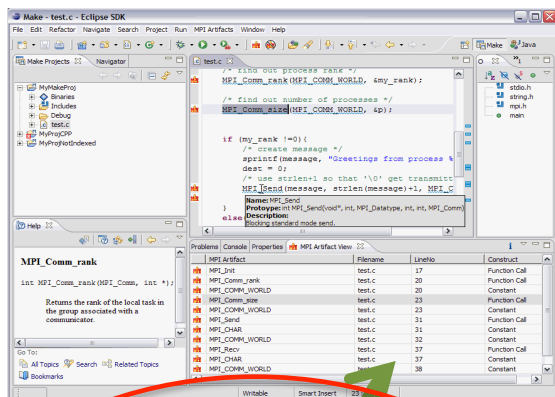
Application Debugging

- Scalable debugger using multicast reduction network
- Integrated with PTP and launched using target configurations
- Supports basic debug commands
- Uses gdb on backend

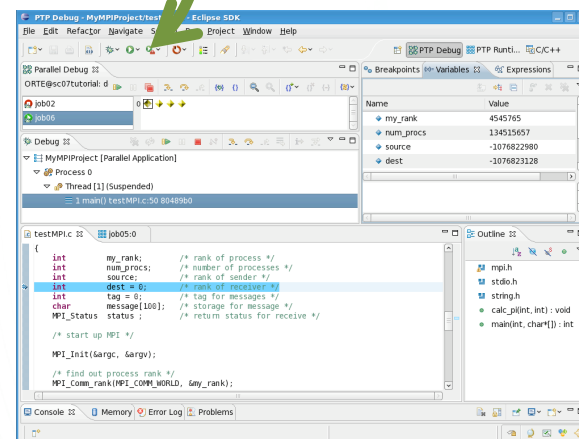
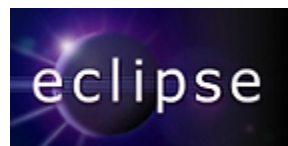
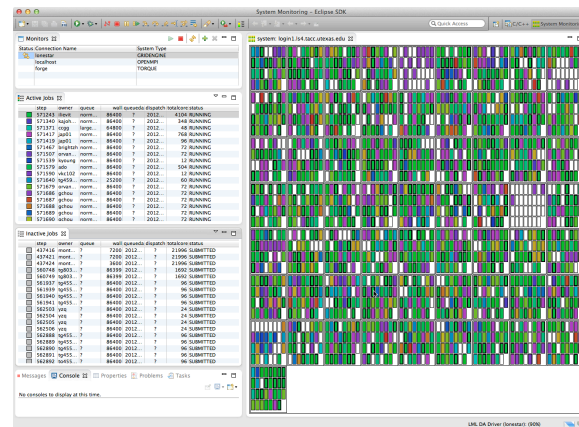


PTP Application Development Cycle=

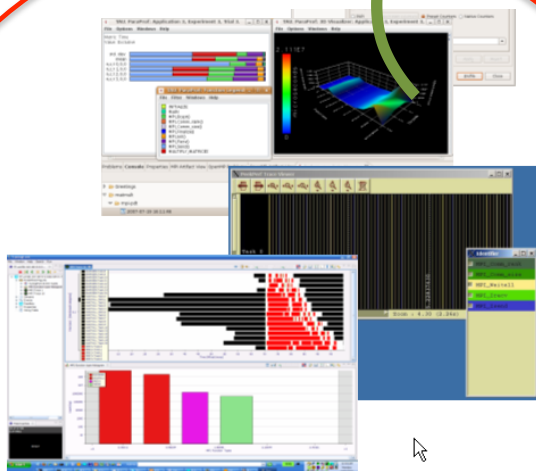
Coding & Static Analysis



Application Execution



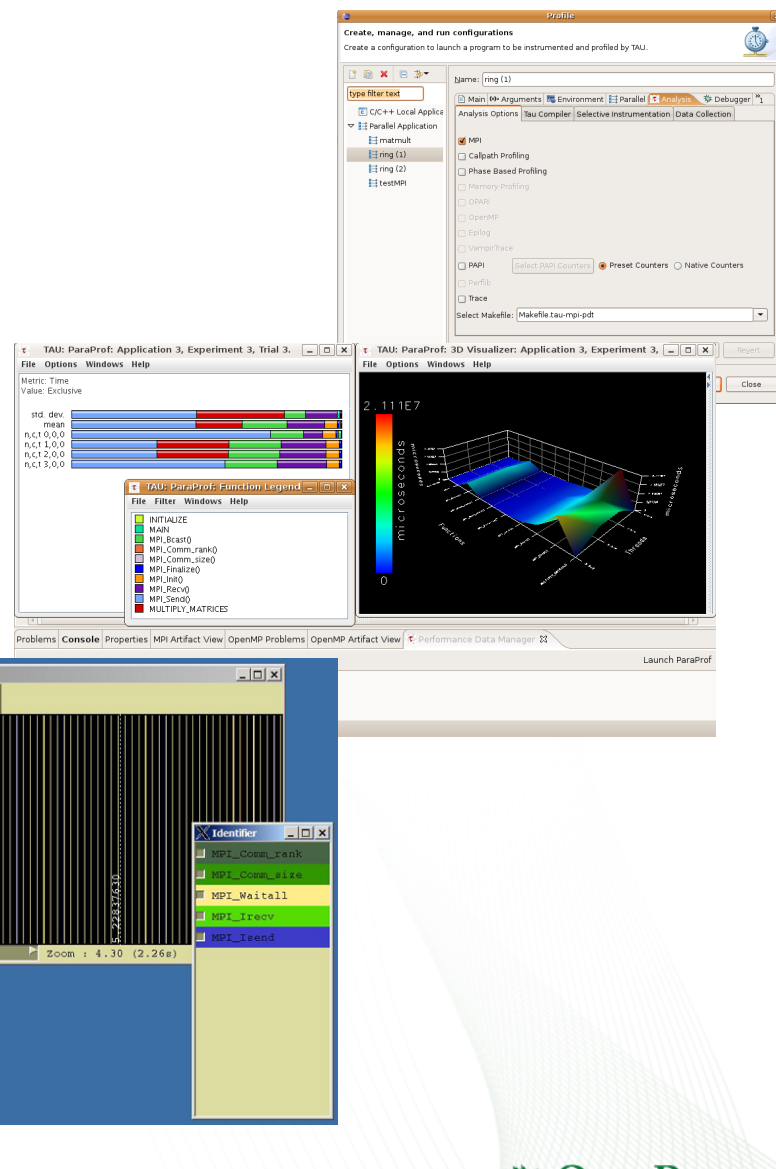
Dynamic & Performance Analysis



Application Debugging

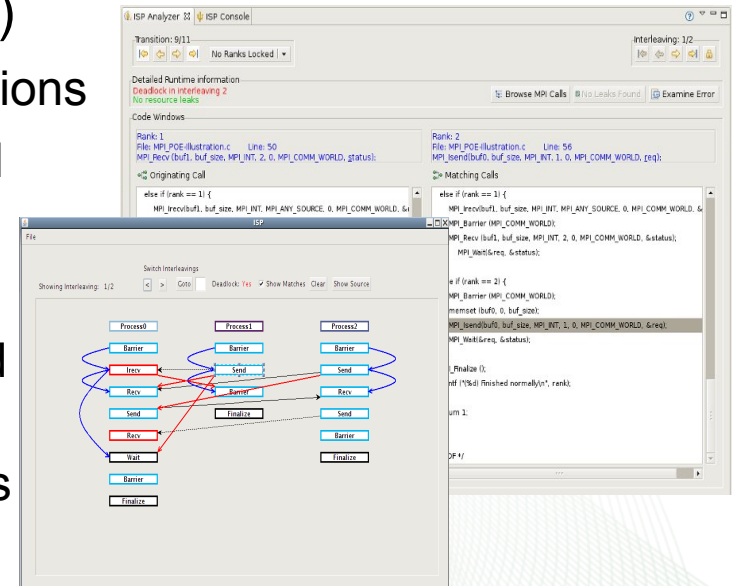
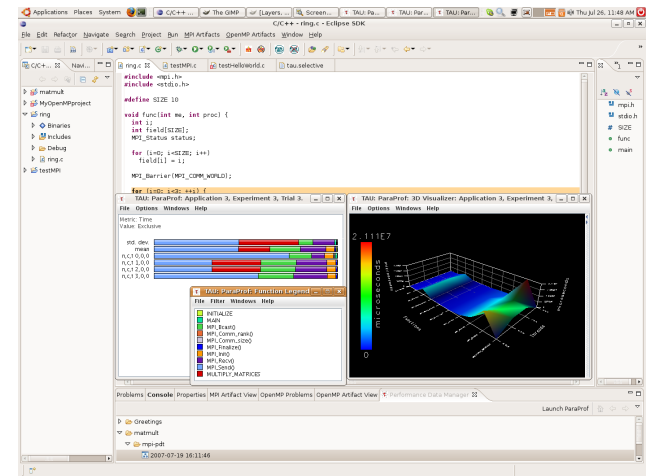
Dynamic & Performance Analysis

- Dynamic Analysis Tools
 - Perform analysis on the running application using external tools
 - Generate results that must be brought back into Eclipse as part of the development workflow
 - May require external tool for visualization or other purposes



Dynamic & Performance Analysis

- Tuning and Analysis Utilities (TAU)
 - Instrumentation and transparent re-build of application executable
 - Execution of profiled application and collect performance data
 - Performance data visible in UI
 - Launches paraparf visualization client from Eclipse
- Graphical Explorer of MPI Programs (GEM)
 - Formal Dynamic Verification of MPI Applications
 - Detects all deadlocks, assert violations, MPI object leaks, and default safety properties
 - Matches sends and receives
 - Allows post-verification review of highlighted bugs
 - Works with a variety of MPI implementations



Online Information

- **Information about PTP**

- Main web site for downloads, documentation, etc.
 - <http://eclipse.org/ptp>
- Developers' wiki for designs, planning, meetings, etc.
 - <http://wiki.eclipse.org/PTP>
- Articles and other documents
 - <http://wiki.eclipse.org/PTP/articles>

Community

- **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 - higher volume
 - <http://dev.eclipse.org/mailman/listinfo/ptp-dev>