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

Jay Alameda[1], Brian F. Jewett[2]

[1]National Center for Supercomputing Applications, University of Illinois, [2]University of Illinois

Overview: Simplifying Complex Problems

- As high-performance computers (HPCs) become more powerful, they become more complex. Changes include moves to multicore, general-purpose graphics processors (GPGPUs), co-processor architectures.
- Optimizing science codes to *efficiently* use current and upcoming petascale machines is– complex!
- Scientific applications *must* be revised to efficiently use these new capabilities to be useful.
- The Workbench for HPC Applications (W-HPC) project is transforming the way computing experts develop, debug, optimize, and run their applications. Their tools are now available to scientific investigators to improve their research applications.
- The Eclipse Parallel Tools Platform (Eclipse PTP) (http://www.eclipse.org/ptp/), part of W-HPC, is being enhanced to aid in working with complex codes:
  - ✓ Eclipse PTP is a robust, open source, portable system providing a *sustainable* development environment.
    - ✓ *Sustainable: many pathways to keep Eclipse PTP a viable development workbench*
  - ✓ It is suitable for use with a broad range of scientific codes (including WRF).
  - ✓ It aids in development, debugging, optimization, and running applications.
- W-HPC includes a targeted education and outreach program that will train new users, explain advantages of using Eclipse-based tools, and encourage user participation in new tool development.

Eclipse and the Eclipse Parallel Tools Platform

Eclipse provides the foundation for our project.

- Eclipse is a vendor-neutral, open-source workbench.
  - •Independent of HPC hardware and software vendors *(though many are contributors to Eclipse PTP)*
- It is designed for multi-language code development including syntax highlighting and specific optimizations for C, C++, and Fortran.
- Eclipse PTP is also an extensible platform for tool integration. It is based on a plug-in framework, as seen in Figure 1a, which is used to create, integrate, and utilize software tools.

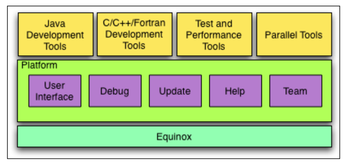

Figure 1a: Schematic of the Eclipse software architecture

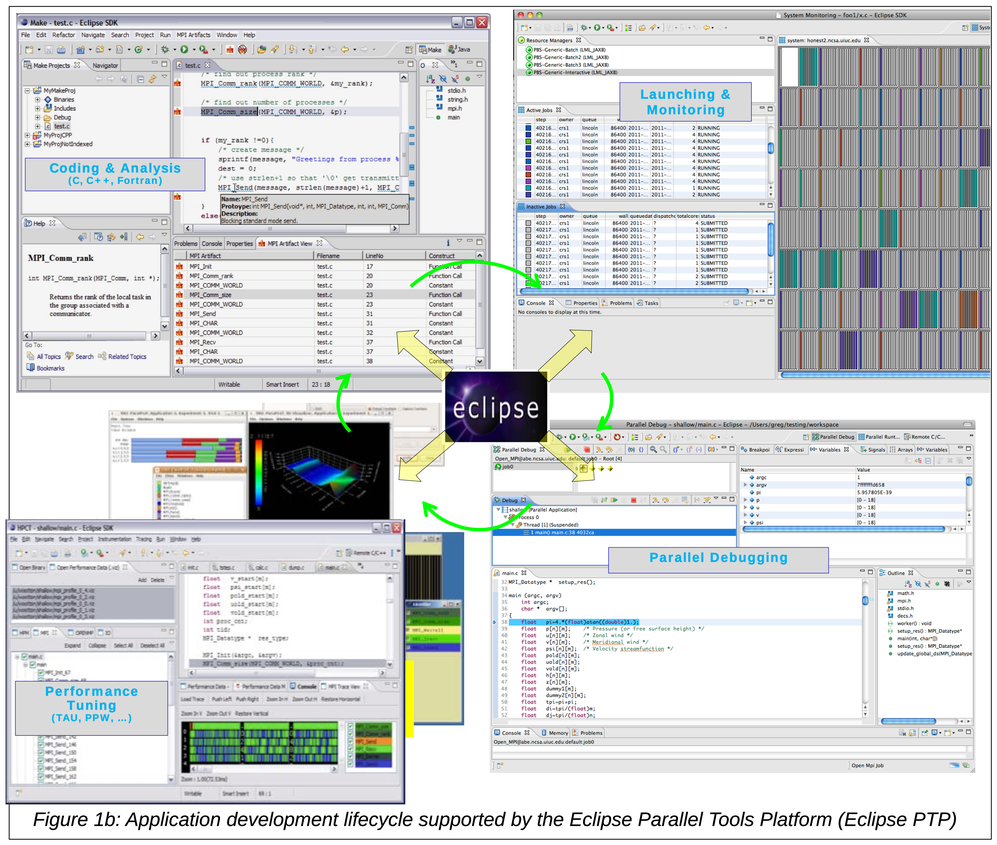

Figure 1b: Application development lifecycle supported by the Eclipse Parallel Tools Platform (Eclipse PTP)

Our **key objectives** in building the Eclipse Parallel Tools include:

- support the *complete scientific application development lifecycle*, as seen in Figure 1b, from coding and analysis, build support, launching and monitoring, debugging, and performance tuning;
- provide the user with as much feedback from compilers and tools to source code as possible;
- provide comprehensive help to guide the scientist in their development and optimization efforts.

To better focus efforts for improving PTP, we are using application codes, on ***production compute resources***, to achieve certain development goals using Eclipse PTP. As we work the development goals, we identify shortcomings in Eclipse PTP that would help us set priorities for improvements to PTP.

We also bridged to **Blue Waters** and **XSEDE** Advanced User Support to assist in their use of Eclipse, and to learn what development and optimization tasks were important to them.

Using Eclipse PTP with WRF

- Eclipse PTP can import WRF and drive remote builds successfully (manually run configure)
- But, WRF uses the C preprocessor extensively, blocking full functionality of Eclipse to be used with WRF
  - Looking for ways to solve this problem for WRF (and other preprocessed codes)

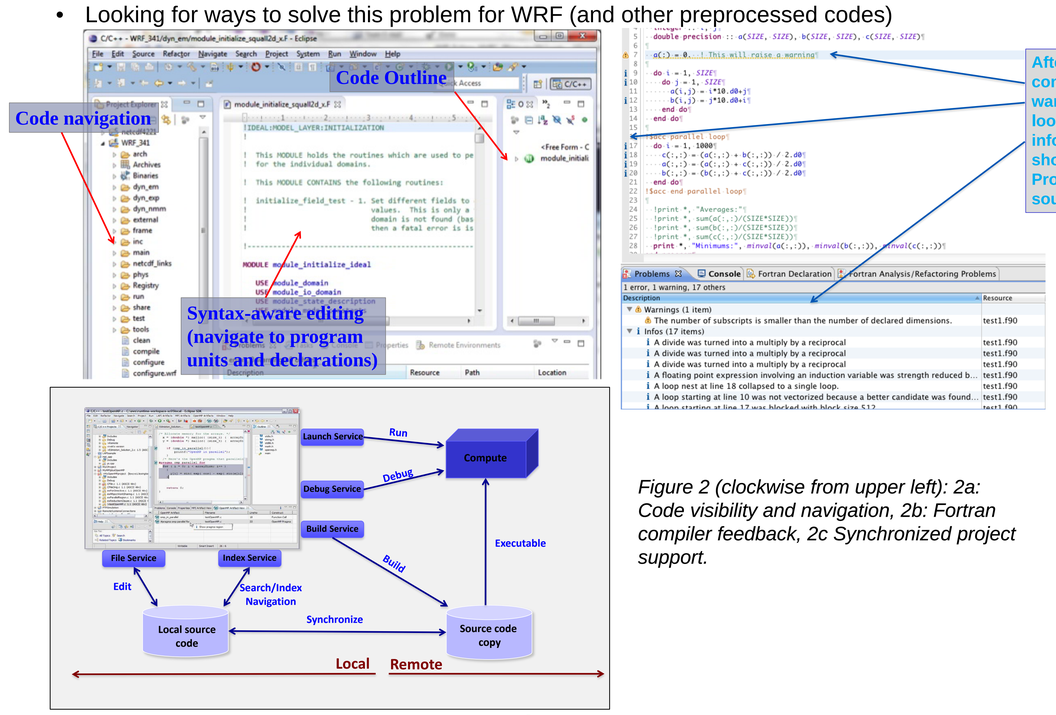

Figure 2 (clockwise from upper left): 2a: Code visibility and navigation, 2b: Fortran compiler feedback, 2c Synchronized project support.

Science Impact

Eclipse PTP has already facilitated scientific research in the following areas:

- *Petascale Earthquake Simulation – Liqiang Wang, U Wyoming.* We have worked with the Wang group to help them configure Eclipse PTP to their local cluster, as well as to support their CUDA code development.
- *Petascale Plasma Physics Simulations Using PIC Codes – Warren Mori, UCLA.* Eclipse was instrumental to assist the Mori group in adapting their code to use GPGPUs via the new openACC directives
- *Cactus Framework, S. Brandt, Louisiana State University.* We made significant improvements to Mojave, which allows Eclipse to support development with the Cactus framework.

Eclipse PTP Project Team, and Recent Accomplishments

Our team has served as the technical integration point, working within the Eclipse foundation, for coordinating and integrating improvements not only from our project, but from other projects improving Eclipse Parallel Tools.

- We coordinate technical developments across many contributing projects.
- We helped our project jump on the Eclipse “release train” and subsequently produced a Parallel Tools Platform bundle, greatly improving ease of installation from the main Eclipse download page.
- Consequently, we have seen tremendous growth of the number of downloads with every Eclipse PTP release.
- Eclipse PTP usability improved with the *configurable target system configuration,* which adapts to batch systems and MPI runtimes (including user interface elements) through an XML configuration. Remote jobs are monitored by a new system monitor contributed by Juelich Supercomputing Center (Figure 3).

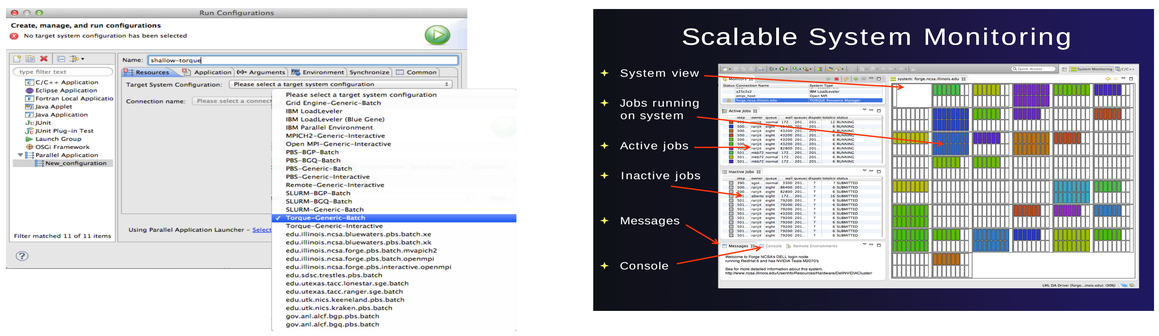

Figure 3a: Target system configurations; Figure 3b: Scalable system monitor

User Support: Outreach and Training

Our project has an extensive outreach, training and education program, to help build our community of users.

- We established a set of monthly user calls, with presentations as well as answering user questions
- We hosted Birds-of-a-Feather sessions at conferences such as XSEDE12, XSEDE13, and SC10, SC11, SC12.
- Tutorials were taught at these conferences and smaller meetings (e.g. UCAR’s Software Engineering Assembly).
- We have hosted an annual user and developer meeting. Our project team has also been involved in using Eclipse in computational science and engineering courses, teaching how to use PTP to complete course assignments.

This work is supported primarily by the National Science Foundation under the following awards: OCI 1047956 and OCI 07-25070. Portions of this material are also 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. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation.