



*2015 Workshop on Software Development Environments for High-Performance Computing*

---

# Toward a Cloud IDE for HPC

---

Jeff Overbey and Mitch Price

Department of Computer Science and Software Engineering

[joverbey@auburn.edu](mailto:joverbey@auburn.edu)

[mcp0038@auburn.edu](mailto:mcp0038@auburn.edu)

# Outline

---

*This is a talk to start discussion, not to present results.*

- Background
  - Why scientific software development is unique
- State of the Art
  - Eclipse Parallel Tools Platform (PTP): An IDE for HPC
  - Eclipse Che: An IDE in the Web Browser
- An Idea: Moving PTP into the Web Browser
  - Really – this is not as stupid as it sounds
  - Why? And how?

# Challenges of Scientific Software

---

- Scientists  $\neq$  Software Engineers
  - Unique languages & APIs
  - Legacy code & community codes
  - Little or no formal training in software engineering
  - Goal = science
  - HPC & changing target architectures
- IDE for software engineers  $\neq$  IDE for scientific programmers

# The Eclipse Parallel Tools Platform

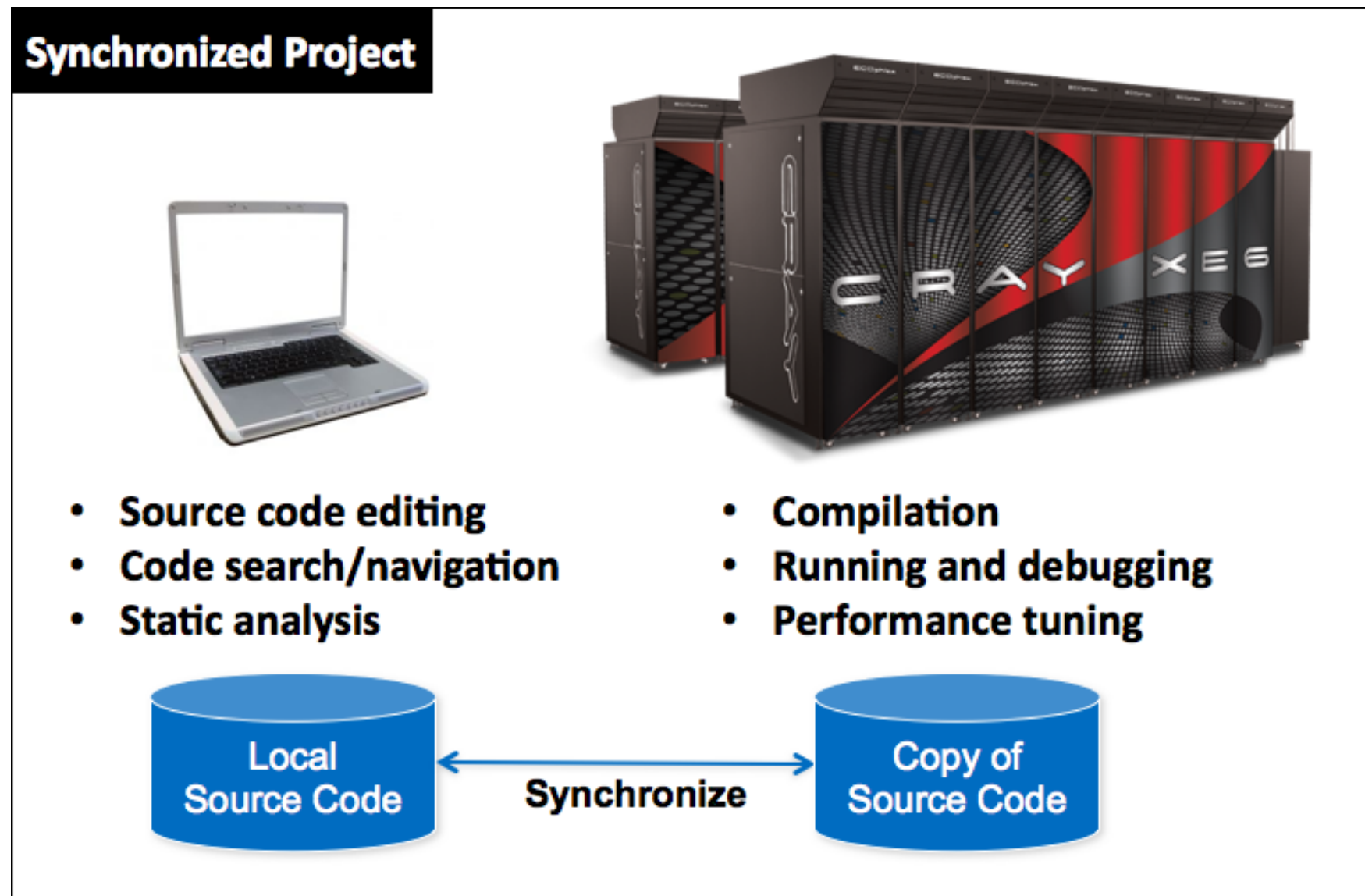
---

- Turns Eclipse into an IDE for high-performance computing

# The Eclipse Parallel Tools Platform

---

- Turns Eclipse into an IDE for high-performance computing
  - Edit locally; compile remotely



# The Eclipse Parallel Tools Platform

- Turns Eclipse into an IDE for high-performance computing
  - Submit and monitor batch jobs

The screenshot displays the Eclipse IDE interface with the Parallel Tools Platform (PTP) configuration and monitoring windows.

**Target System Configuration:** edu.sdsc.trestles.torque.batch

**Connection Type:** Local (selected), Remote (selected), trestles

**Basic Settings:**

| Name      | Value   |
|-----------|---------|
| Job Name: | ptp_job |
| Account:  |         |
| Queue:    |         |

**System Monitoring - Eclipse - /Users/beth/ews/test1106tutor**

**Monitors:**

| Status | Connection Name   | System Type             |
|--------|-------------------|-------------------------|
|        | trestles.sdsc.edu | TORQUE Resource Manager |

**Active Jobs:**

| step     | owner   | queue  | wall  | queued  | dispa |
|----------|---------|--------|-------|---------|-------|
| 10553... | jmondal | normal | 64800 | 2012... | 201   |
| 10553... | jmondal | normal | 64800 | 2012... | 201   |
| 10553... | jmondal | normal | 64800 | 2012... | 201   |
| 10553... | jmondal | normal | 64800 | 2012... | 201   |

**Inactive Jobs:**

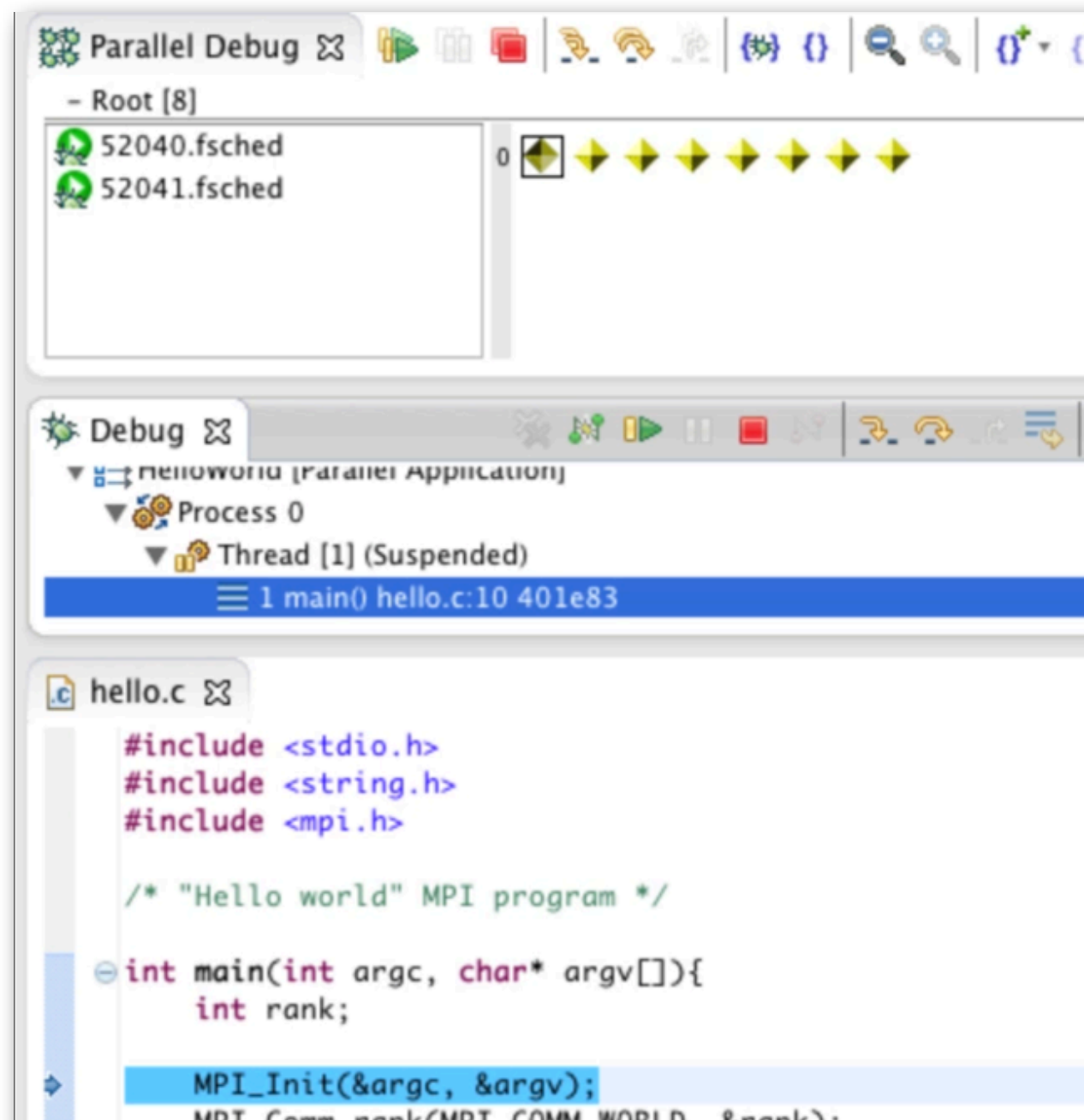
| step    | owner    | queue  | wall | queued | dispatc | total |
|---------|----------|--------|------|--------|---------|-------|
| 1056... | tibbitts | shared | 1800 | 201... | 201...  | 5     |
| 1056... | tibbitts | shared | ?    | ?      | ?       | ?     |
| 1056... | tibbitts | shared | ?    | ?      | ?       | ?     |

**Visual Job Status:** A grid of colored squares representing the status of various jobs across different resources.

# The Eclipse Parallel Tools Platform

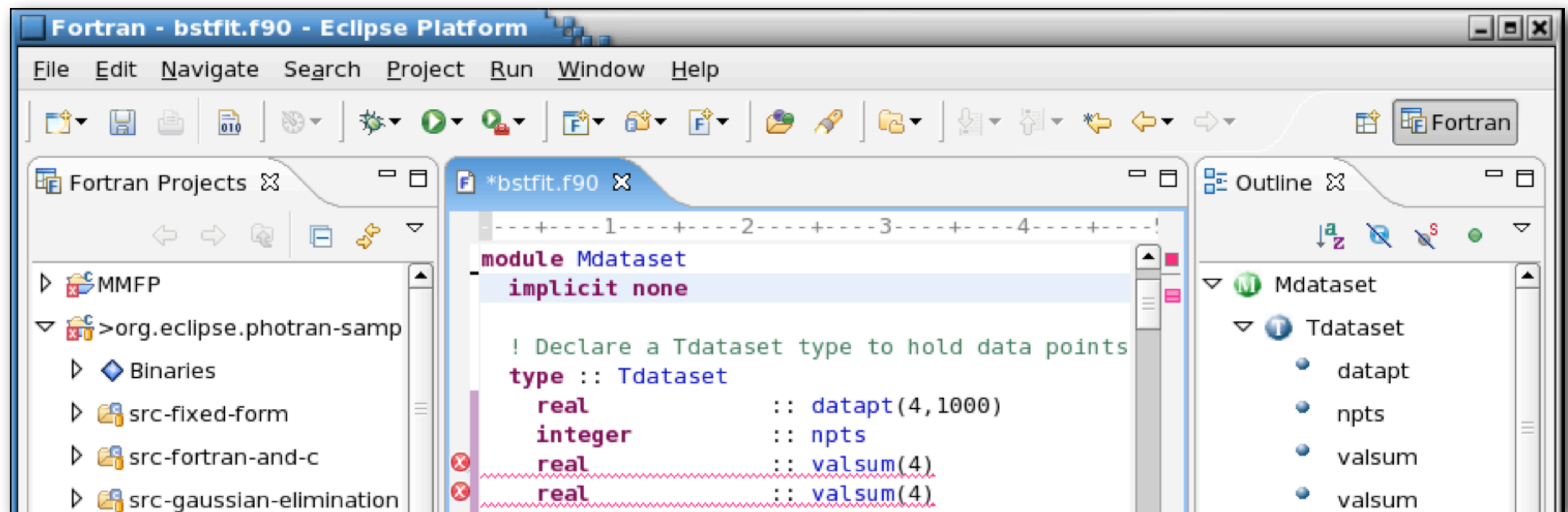
---

- Turns Eclipse into an IDE for high-performance computing
  - Debug MPI programs and tune performance



# The Eclipse Parallel Tools Platform

- Turns Eclipse into an IDE for high-performance computing
- Adds support for editing/analyzing Fortran, MPI, OpenMP, OpenACC





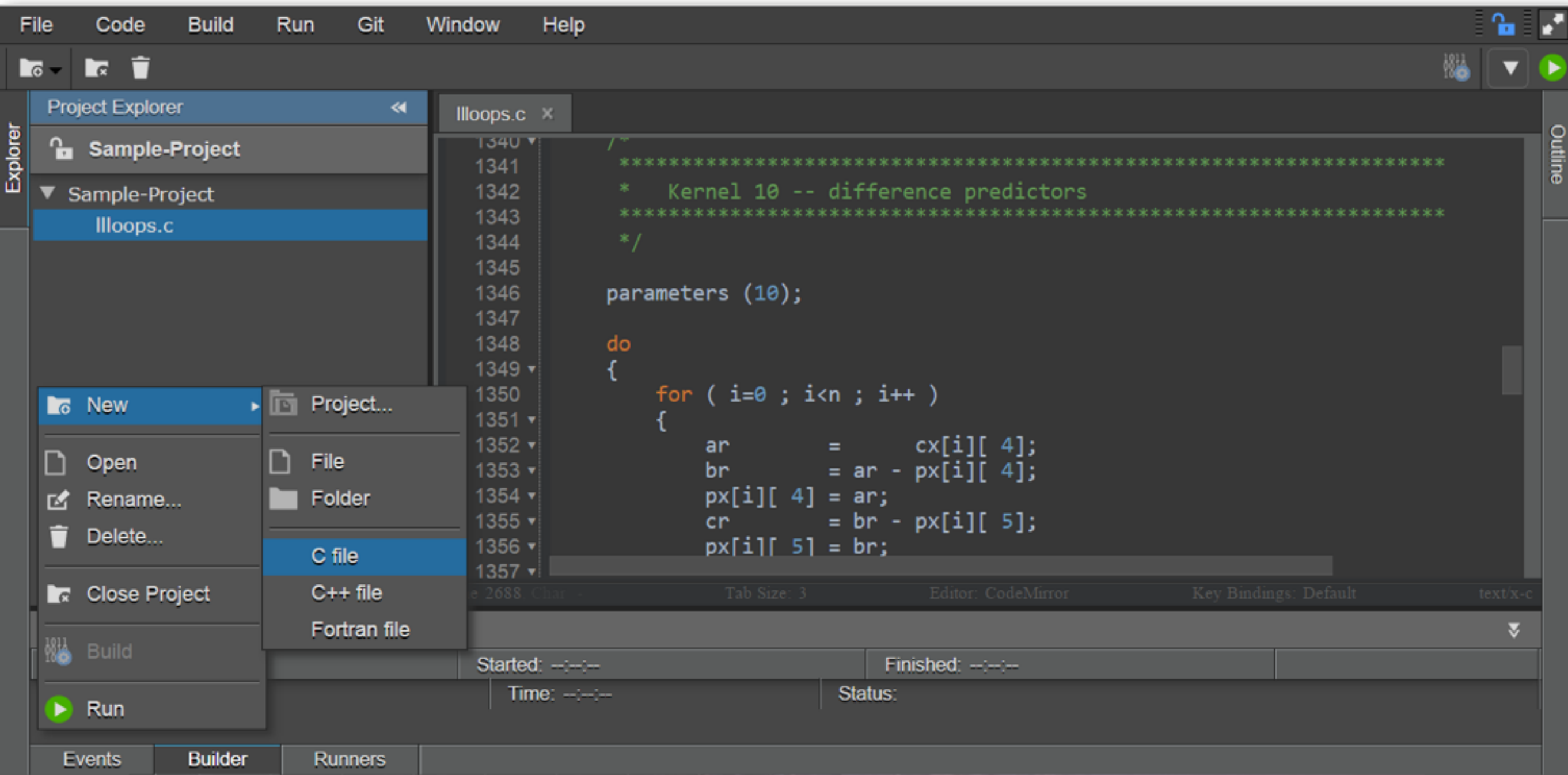
# PTP's Challenges

---

- We don't know how many people use PTP or who they are
- Eclipse's learning curve is steep
- PTP doesn't “just work” out of the box
- Scientists use what their peers use
- Technical support is limited
- HPC vendors are not engaged

# Eclipse Che

- IDE that runs in the Web browser



# Eclipse Che

---

- IDE that runs in the Web browser
  - Open-source contribution to the Eclipse Foundation from Codenvy
  - Supports editing, running, debugging, version control (Git, Subversion)
  - Client-side editor + server-side microservices
  - Written in Java + GWT
  - Currently targeted at Web application development
    - Builds a Docker image of the project; runs in a Docker container
    - But the goal is to move beyond that...

# A Web-based IDE for HPC (1/4)

---

## Concept

- Build on Che to create a Web-based IDE for scientific software
- Start by porting components from PTP to support...
  - Remote connections and authentication
  - Synchronized projects
  - Environment modules and remote build
  - Job submission and monitoring
  - SSH terminal
  - Parallel debugging and performance tuning (maybe in Version 2.0...)

# A Web-based IDE for HPC (2/4)

---

## **Why move to the Web?**

- New users can become productive quickly
  - Minimal start-up time for new developers  
(create an account and go—no compilers/libraries to install, HPC configuration, etc.)
  - Fewer opportunities for client-side problems
  - Minimal user interface
  - Easy to deploy bug fixes and new tools to users easily
- Better opportunities for user study (research)
- Better opportunities for collaboration (next slide)

# A Web-based IDE for HPC (3/4)

---

## **Collaboration & community features are essential**

- *GitHub is partly responsible for Git's popularity. Building on that idea...*
- We believe the key to success is supporting a *community* of users
  - Allow users to share code
  - Make community codes first-class citizens
  - Allow users to share HPC configurations
  - Allow users to share bookmarks/annotations in large codebases
  - Allow conversations/forums
  - Allow collaborative editing and helpdesk access

# A Web-based IDE for HPC (4/4)

---

## **Disadvantages**

- Wide variety of user experience (novices through tuning experts)
- Cost – development & operation
- Loss of developer autonomy
- Security issues are more complex
- Some features are technically difficult to provide
- Codenvy is mature (89,000+ users) but Che is not (collaborate?)
- Che's architecture is quite different from the Eclipse platform

# Some Challenges Resolved?

---

- Eclipse's learning curve is steep
- PTP doesn't "just work" out of the box
- Scientists use what their peers use
- Technical support is limited
- HPC vendors are not engaged
- Che is a lot simpler than Eclipse
- Che runs in a Web browser
- Users in a research group/area could share configurations
- Collaboration features could encourage adoption by peers
- Collaboration features could encourage support from peers (think StackOverflow)
- Can we engage HPC vendors early?



# Summary

---

- PTP turns Eclipse into an IDE for HPC
- Eclipse Che is an IDE that runs in the Web browser
- **Web-based IDE could be ideal for scientific software**
  - Easier for new users to get started
  - Better opportunities for user study
- **Community/collaboration features would be essential**
  - Share code, HPC configurations, etc. within research groups/fields
  - Support collaborative editing, conversations, etc.
  - Incentivize peer support/collaboration