Speculative Analysis

Exploring future states of software development

Yuriy Brun
Reid Holmes
Kivanc Muslu
Tim Vega
Michael Ernst
David Notkin
How shall we use unprecedented compute power?

- Multicore
- Clusters
- Cloud

Typically used to make your program go faster.

Our goal: make the developer go faster.
Program analysis

Informs you about your program
• Type-checking
• Testing
• Profiling
• Bug-finding
• Verification
• Collaboration

Idea: run program analysis on programs you have not yet written
Future programs

A programmer chooses from:
• ~100 keys
• ~200 menu items
Try each one, on one of your 300 cores
• Then: compile, test, other analyses

Advantages:
• If the developer performs the action, results are immediately available
• Information about consequences can help the developer make better choices
Speculative Quick Fix

Status: under development
Speculative version control

Crystal:
A proactive conflict detector
Collaborative Development Info

- Local state
- Relationships with other repositories
- Actions
- Guidance

- same
- ahead
- behind
- merge
- conflict
Crystal
A proactive conflict detector

- Implementation is publicly available: [http://www.cs.washington.edu/homes/brun/research/crystal/](http://www.cs.washington.edu/homes/brun/research/crystal/)
- Works with Mercurial (distributed version control)
- Positive initial feedback from users
“I don’t need Crystal”

• Retrospective study of version control history
  – 8 projects, >3M LOC (non-comment non-blank)

• 18% of merges result in a conflict (572 out of 3514)
  – Textual, compile, and behavioral conflicts
    ⇒ We want to avoid these!

• A conflict lasts 10 days and 12 commits before being resolved
  ⇒ The earlier a conflict is resolved, the easier

• A clean merge relationship lasts 11 days and 12 commits before being resolved
  ⇒ Merging earlier can help to avoid conflicts
How a developer uses Crystal

• Usually, leave minimized in the system tray
• If icon shows action is possible, open Crystal
• Clean merge, push, or pull:
  – Do it now: Avoid potential for developing a conflict
  – Delay: If in the middle of critical work
• Conflict:
  – Talk to the other developer, avoid exacerbation
Developer can predict consequences, avoid wasted work, and act with confidence
Speculative program analysis

• Use excess compute cycles to help the developer
• Analyze programs before they are written
• Guide developer to avoid unproductive work
• Examples:
  – Quick fix (and other IDE actions)
  – Version control
  
    http://www.cs.washington.edu/homes/brun/research/crystal/

• Publications: FoSER 2010; ICSE 2011 submission