Towards an Ecologically Valid Study of Programmer Behavior for Scientific Computing

Christine A. Halverson, Cal Swart, Jonathan Brezin, John Richards & Catalina Danis
Overview

- Motivation
- Previous Studies
- Study Design, Implementation & Lessons Learned
- Preliminary Findings
- Conclusions

? Questions?
Why are we doing this?

- DARPA sponsored High Productivity Computing Systems project
  - Peta-scale computer (with software stack) by 2010
  - High productivity for humans, not just high performance
  - A 10x increase from 2002

- Programmer productivity = ?

- Measuring differences requires setting a baseline

- Where’s the baseline?
Why strive for ecological validity?

- Distinguish empirical versus ecologically valid
- How close do methods, materials, setting and practices mimic the real world?
  - An objective since the late 60s in software studies
  - But...real world is messy
- Quantitative, qualitative or hybrid methods
- Challenges in HP/SC
  - Machine - specificity
  - Busy doing real work -- Niche
  - Problems - big and time consuming

“good enough”
Run on weekends

Our approach: Integrative methodology and ecologically valid
Study Design: Overall

- IRB-Institutional Review Board
  - Two are better than one?
  - It’s different when you’re not teaching students

- Requires -
  - Complete study plan
  - Study materials - consent forms, recruitment mails, screening surveys, pre and post interviews & the exact protocol

- Decisions
  - Subject population
  - Hardware and software
  - How to capture data

Takes longer - not exempt

Details, details, details

Available ≠ attraction?
Design Implementation

- **Novices:** at least 1 parallel programming class and experience parallel programming up to 3 years

- **Experienced:** > 3 years experience

- **Problem write-up - SS CA1 - Smith Waterman Algorithm**
  - Previous (PSC) Edmiston
  - More demanding… but more detailed

" Doesn’t anyone take the easy way? 

**PILOTING**

Iterate.iterate…
Implementation: Hardware and Software

- Circa 2002
  - Command line, emacs or vim, gdb or Totalview, C or Fortran, MPI

- IBM SP3 “Seaborg”
- Laptop
- Hackystat on both
- Istanbul, Slogger on Laptop

- IBM Power5 “Bassi”
- Revised version of Hackystat

PILOTING
Recruitment and Experience

- No Silver Bullet
  - Management buy in
  - Commitment of individuals to recruit

- Difficulties finding subjects
  - Experience means often too busy

- Do parallel programmers skip from classes to >3 years experience?
- Is MPI becoming a dead language?
Murphy’s Law is alive and well

- IRBs are touchy, and two are twice as

- Old habits are hard to break

- A Matter of Control?
Where are the findings???

- So far only 4 subjects

- LOTS O’ DATA

- Long painstaking process for analysis - at least at this time

- Cross validate the data
  - Against other sources
  - Between subjects

Not enough to report findings

270 pages (~5.4k entries) Hackystat data, 8-12 hours per subject screen capture, 4-8 hours of video

Determining patterns from the data, then coding them (50+ in 9 categories) and calculating dev time (571 instances in 2 subjects)

Still to do
Key Issues

- Is a retrospective baseline an oxymoron?
  - Consistent POE
  - Slow change in tools
  - Problem size is not impacted by machine specificity

- Ecological vs External validity
  - 20 subjects - looking for details not generalizable statistically

- Natural Science phase
Questions?

Contact:
Christine Halverson
kryss@us.ibm.com