Xpert Network Panel: Best practices for Computational and Data-intensive (CDI) Research

Karen Tomko
Ohio Supercomputer Center
Email me: ktomko@osc.edu

RSE Team: Samuel Khuvis, Evan Danish, Shameema Oottikkal
Best Practices

• Build a strong foundation
  • Computing: computer organization, operating systems, programming languages, software engineering, AI
  • Math: linear algebra, numerical analysis, statistics
  • The scientific method: computational science or data analytics

• Be curious
  • Learn about
    • New technologies, tools, languages and standards
    • What others are doing
  • Try things, experiment
    • How does this work? Is a faster or is b? etc.

• Be disciplined / methodical
  • Particularly important for debugging
  • Tools can help with this
For RSE work specifically

• **Communicate**
  • Expectations / project goals
  • Ask questions
  • Meet regularly

• **Corral the code**
  • What version of the code am I starting with?
  • How do I build and run it?
  • How do I know if it’s running correctly?
  • Choose a license (if there isn’t one)

• **Use modern tools**
  • Source code repos (branch, merge, …)
  • Build systems / package managers
  • Compilers, debuggers, profilers
  • CI pipelines

• **Automate** frequent tasks
  • Regression & benchmark tests
  • Debugging runs
  • Treat scripts as code, they are

• **Document …**
  • The code (including scripts)
  • The install procedures
  • Issues and what you’ve tried and learned

• **Nudge, don’t push**
  • Introduce new tools and processes gradually
  • Don’t make code unrecognizable to the science team (without consent)

• **Get a quick win** (if you can)
  • Performance gain or bug fix
Training new team members

• Foundation gaps
  • Direct to resources (classes, tutorials, books, …)
  • Provide time for learning

• HPC specific skills
  • Direct to tutorials, bootcamps or online learning

• RSE mentoring
  • Coaching with respect to communication
  • Encourage use of appropriate tools
  • Guidance during debugging & troubleshooting
  • Be a resource (e.g. Fortran, netlib)

• Support participation in the RCD community by encouraging …
  • joining mailing lists and slack channels
  • attending seminars, webinars, community chats
  • attending workshops, tutorials, and conferences
Backup Slides
Current RSE Project Portfolio

- **SETSM Terrain Generation Code**
  - C/C++, MPI+OpenMP
  - Code is in active development and used in a production environment
  - Software engineering, performance improvement

- **FDL3Dlv2 CFD software**
  - Fortran, MPI+OpenMP
  - Performance improvement, modernize for new architectures

- **Soil Spectroscopy**
  - R
  - Software engineering, parallelization, development

- **MVAPICH2 MPI**
  - C/C++, Fortran, CUDA, new programming models (PGAS, tasks), Python
  - Performance analysis, case studies

- **GPS Gear Simulation**
  - Fortran, OpenMP
  - Port to Linux, add MPI parallelization
Tools we currently use

• Git / Gitlab, Cmake/make
• Docker and singularity
• Compiler reports and diagnostics: Intel, Gnu, Cray
• Debuggers: gdb, DDT
• Memory Debuggers: Valgrind, Asan (via gcc)
• Performance Analysis: Intel Advisor & ITAC, ARM Map & Performance reports, Tau, mpiP
• Manual Instrumentation: timing, dumping variables, checking memory use
• Reframe test framework, test scripts