COMPUTATIONAL AND DATA-INTENSIVE (CDI) RESEARCH -BEST PRACTICES

Sunita Chandrasekaran

Assistant Professor, University of Delaware

Dept. of Computer & Information Sciences

schandra@udel.edu

Xpert Network, July 15, 2021

BEST PRACTICES - 7 OF MANY!

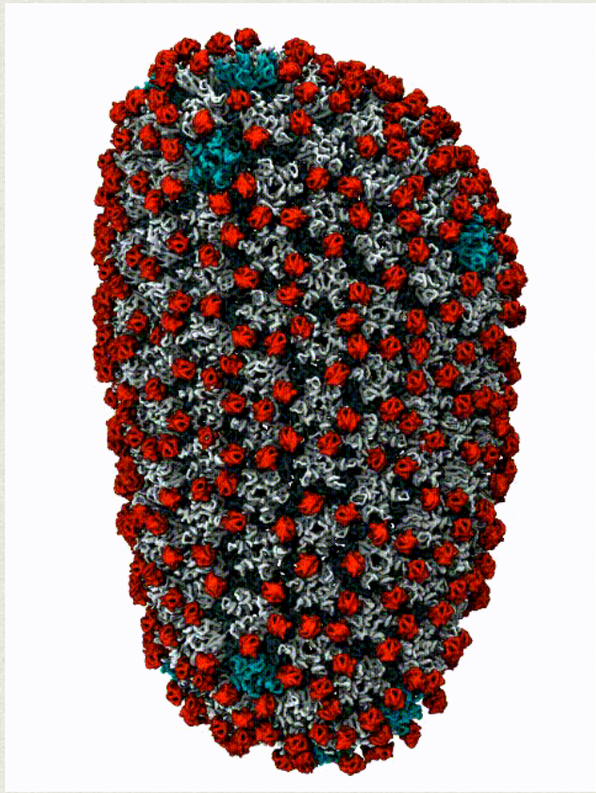
- Best Practice #1 Profiling
- Best Practice #2: Systematic Testing
- Best Practice #3: Report bugs
- Best Practice #4: Automate
- Best Practice #5: Document
- Best Practice #6: Pair Programming
- Best Practice #7: Open Source but…

BEST PRACTICE #1:

PROFILING

ACCELERATING A BIOPHYSICS PROBLEM ON GPUS
(in collaboration with Prof. Juan R. Perilla, UD)

- Nuclear Magnetic Resonance (NMR) is a vital tool in structural biology and biochemistry
- NMR spectroscopy measures chemical shifts
- Predicting chemical shift has important uses in scientific areas such as drug discovery

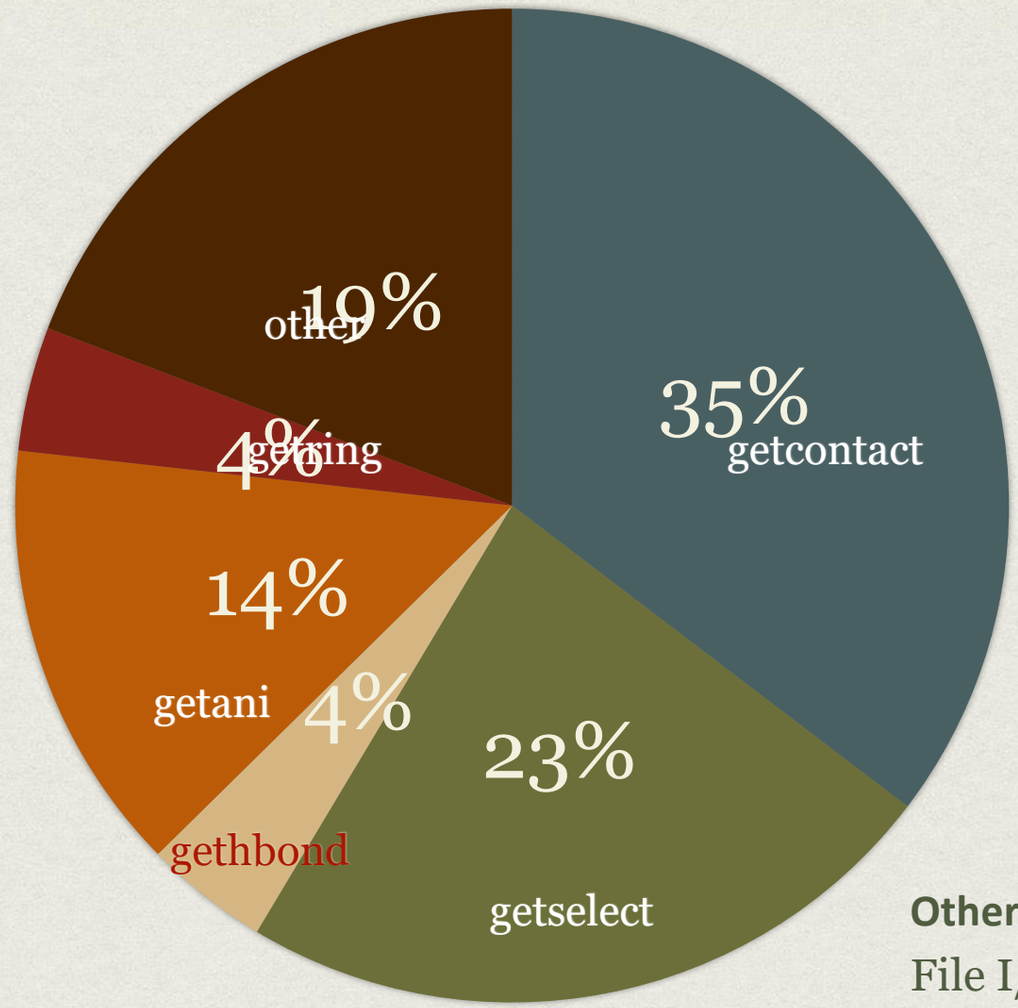


Our goal:

- Accelerate the prediction of chemical shift
  - To enable execution of multiple chemical shift predictions repeatedly
  - To allow chemical shift predictions for larger scale structures

SERIAL PROFILE VISUAL

- Serial code profiling NVProf
- Obtained large overview without needing to read thousands of lines of code
- Identified hotspots within the code
- 2 undergrads - 1 year project

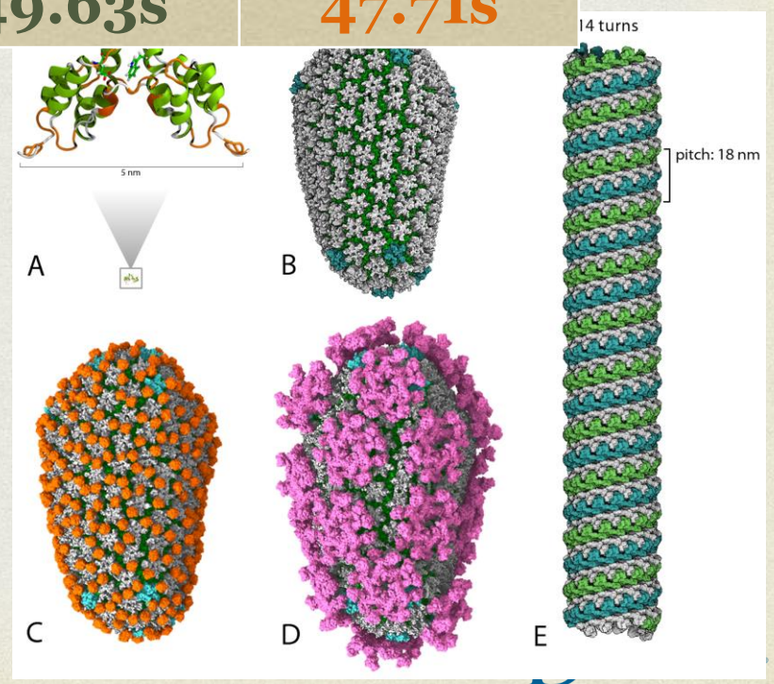


Other Contains:
File I/O
PDB Structure
Initialization

OPENACC-GPU PERFORMANCE RESULTS

	100K atoms	1.5M atoms	5M atoms	6.8M atoms	11.3M atoms
Serial (Unoptimized)	167.11s	572.01s	3547.07s	7 hours	14 hours (estimate)
Serial (Optimized)	53.57s	196.12s	2003.6s	1510.71s	2614.4s
Multicore	4.67s	32.82s	116.66s	153.8s	146.06s
NVIDIA P40	3.47s	17.15s	56.2s	78.57s	72.55s
NVIDIA V100	3.11s	13.62s	39.79s	49.63s	47.71s

PLOS Computational Biology journal:
https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1007877



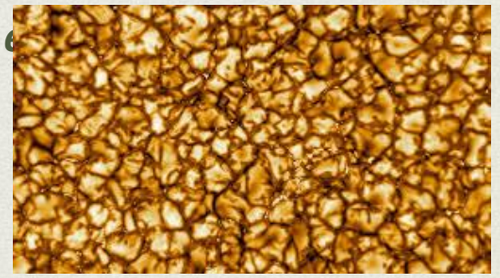
CRPL
Sunita Chandrasekaran

BEST PRACTICE #2:
SYSTEMATIC TESTING

PREPARE MURAM (MAX PLANCK UNIVERSITY OF CHICAGO RADIATIVE MHD) FOR NEXT-GENERATION SYSTEMS

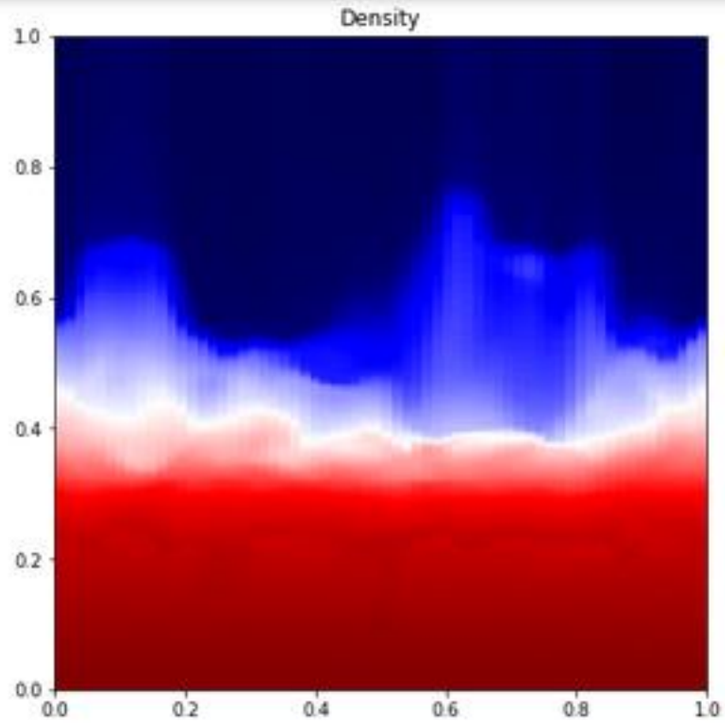
- LOC: 30-40K Lines of Code
- Team: 4 Computer Scientists and 2 Solar physicists
- Tools used: Valgrind, GDB, Python notebooks, PCAST (an NVIDIA/PGI validation tool)
- Test bed: 3 different CPU-GPU compilers and 2 different platforms (NCAR & MaxPlank)
- Version Control: GitLab
- Tim ent: 8 months and the clock was still ticking!!!!

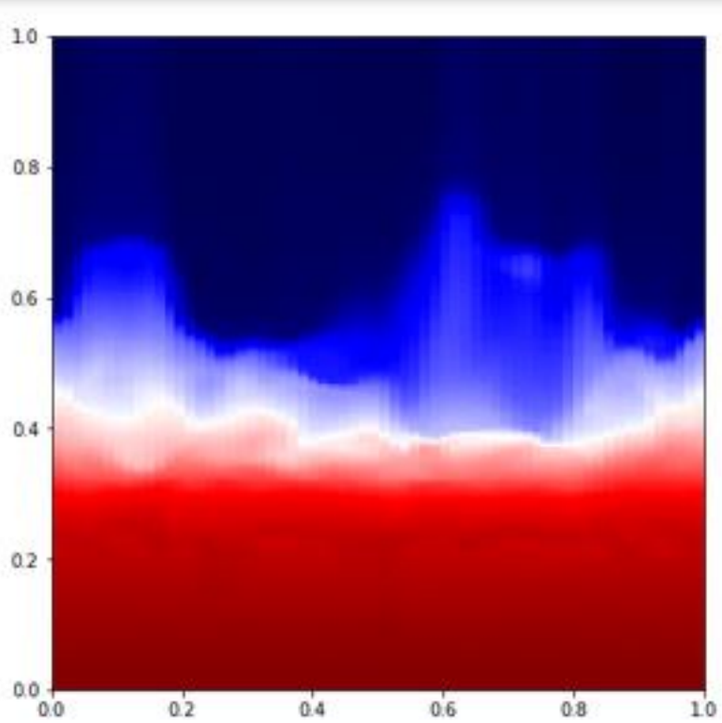
Comprehensive model of entire life cycle of a solar flare

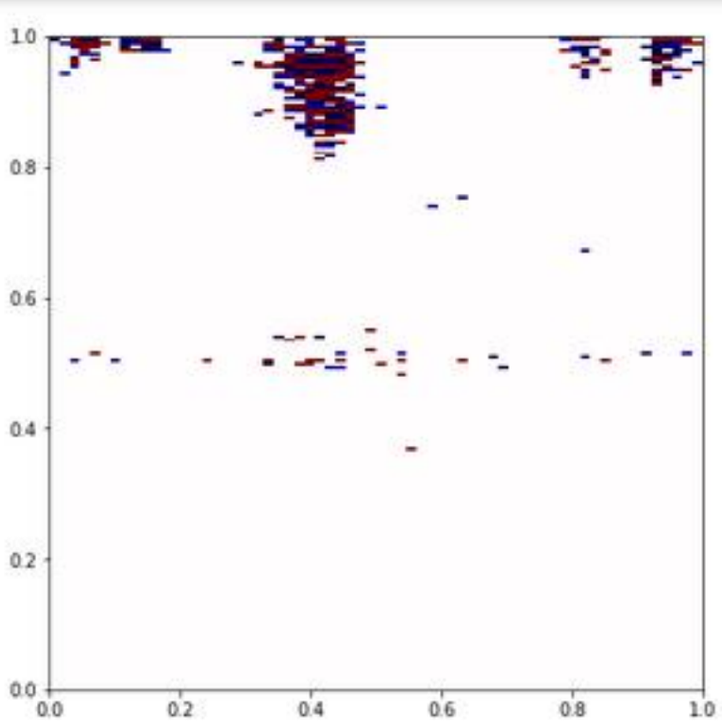


The Daniel K. Inouye Solar Telescope (DKIST), a ~$300M NSF investment, is expected to advance the resolution of ground based observational solar physics by an order of magnitude.

ERROR IN DENSITY CALCULATION







Difference in density between reference and the test runs

CRPL
Sunita Chandrasekaran

UNIVERSITY OF DELAWARE

LESSONS LEARNT

- **Work in small steps:** Make incremental changes
- **Testing:** AFTER every commit BEFORE optimizing/scaling
- DOCUMENT!!! PLEASE :-)
- **Test bed:** 3 diff. compilers, more than 2 hardware setup, more than 2 versions of a compiler (perhaps?)
- **Optimization:** ONLY after a thorough verification of the serial or the base version. VERIFY again
- **Verification:** Build Jupyter notebooks to capture and record divergence of results between serial/multi-core CPU and GPU

YOOHOO - PASC PUBLICATION!

Sunita Chandrasekaran @sunitachandra29 · Dec 14, 2020

So proud of my project team!🥳
Paper submitted 😇
3 yrs 😱
4 talks
1 PhD prelims😤
1 MASSIVE bug took 10 months to fix😳
RIGOROUS testing🤓
Jupyter npbs
GitHub 💪
MANY intense 📞w/ scientists
#InterdisciplinaryScience

The JOY u get doing spell check 4 hrs b4 the deadline!😎

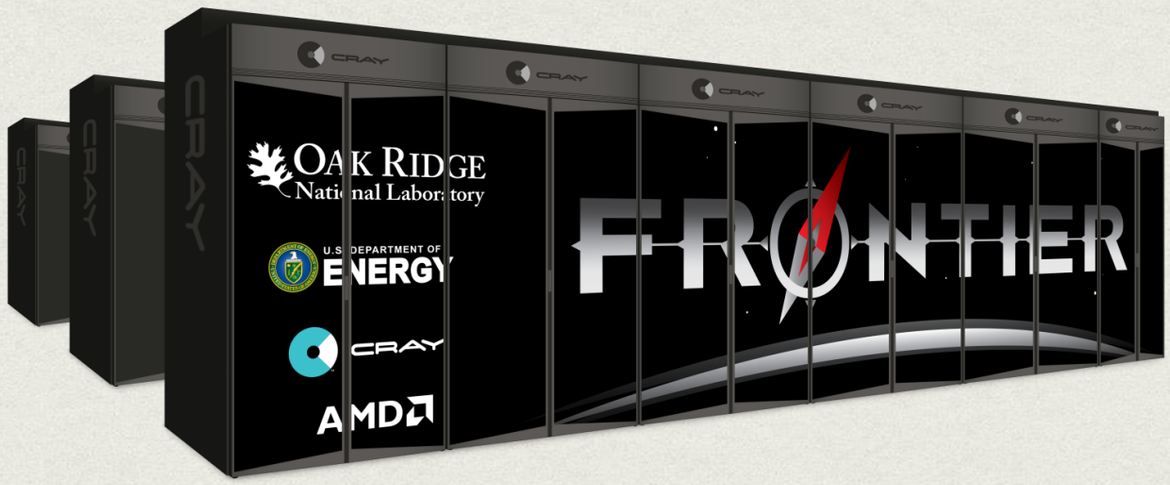
CRPL
Sunita Chandrasekaran

UNIVERSITY OF DELAWARE

BEST PRACTICE #3:
REPORT BUGS

CAAR-ORNL-PICONGPU PROJECT

- Preparing PIConGPU, a plasma Physics application for the upcoming exascale system - Frontier



CAAR Project in Collaboration with COE (AMD + Cray) developers

REPORT BUGS!

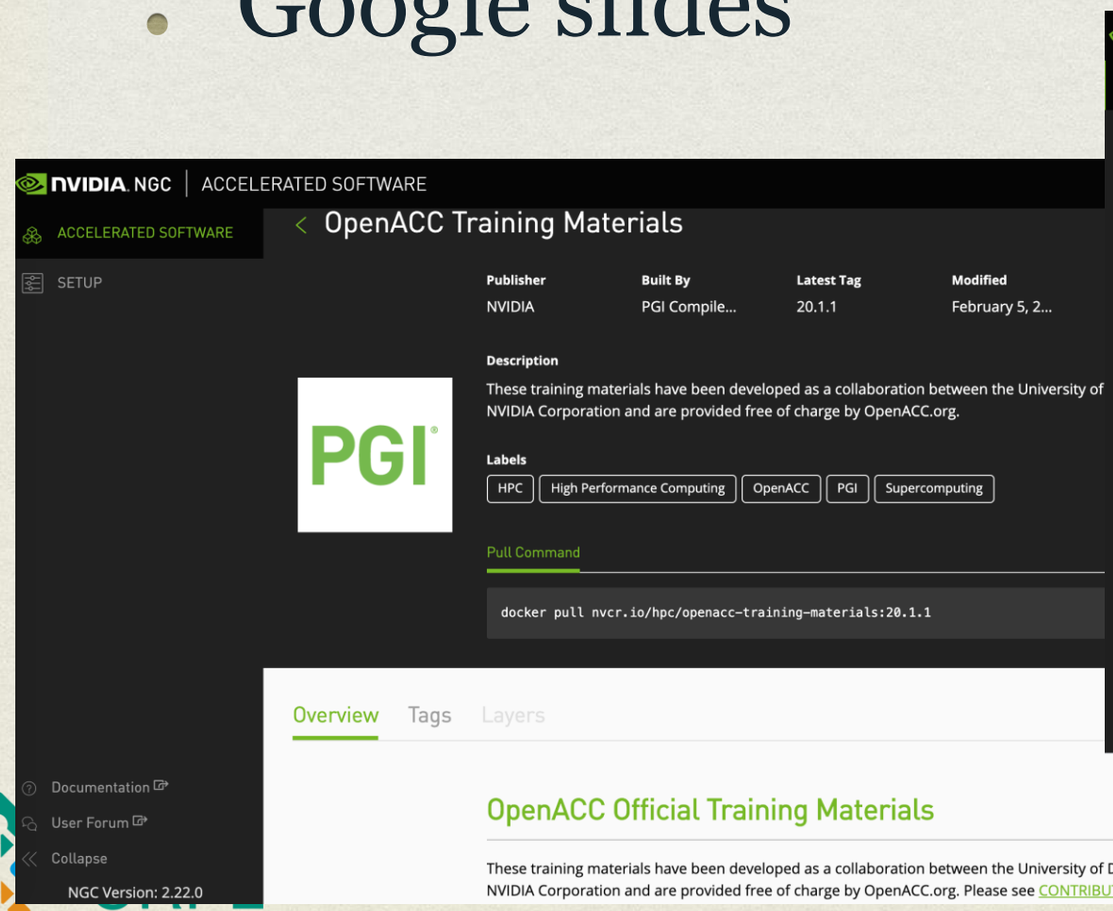
- **Code Review:** author of a PR cannot merge his/her own PR)
- **Report bugs:** help improve compilers
- **Reproducible code:** Useful to debug
- **REPORT:** Workarounds OK but ”REPORT” bugs
- Report bugs via a ticket system (say Trac – wiki + issue tracker) and not via email – PLEASE! 💻
  - The bug and its fix got to be recorded
  - Documented
  - Code changes to be tracked
- **Time critical bugs** - communicate with the developers directly

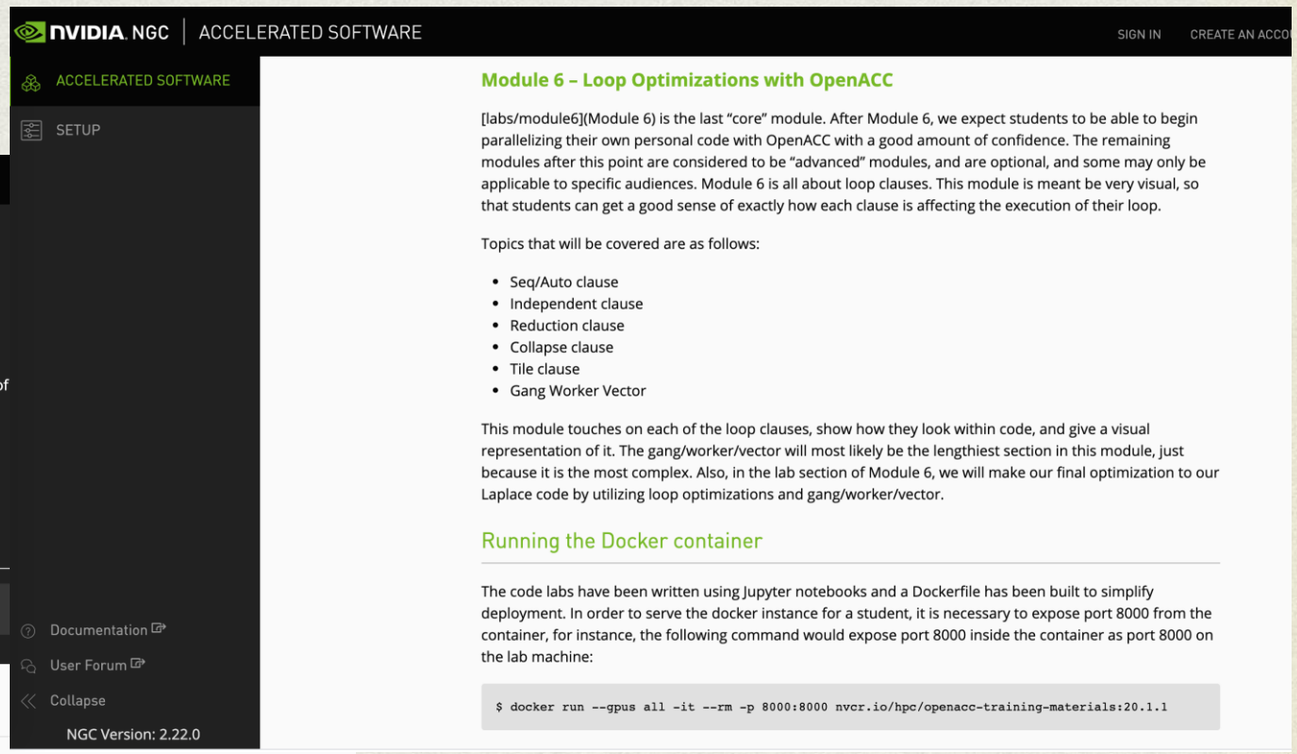
BEST PRACTICE #4:

DOCKER

COLLABORATIVE SOFTWARE DEVELOPMENT

- Tools/Platforms such as Docker, Container, GitHub
  - Dramatically reduces barrier to collaboration
  - Google slides

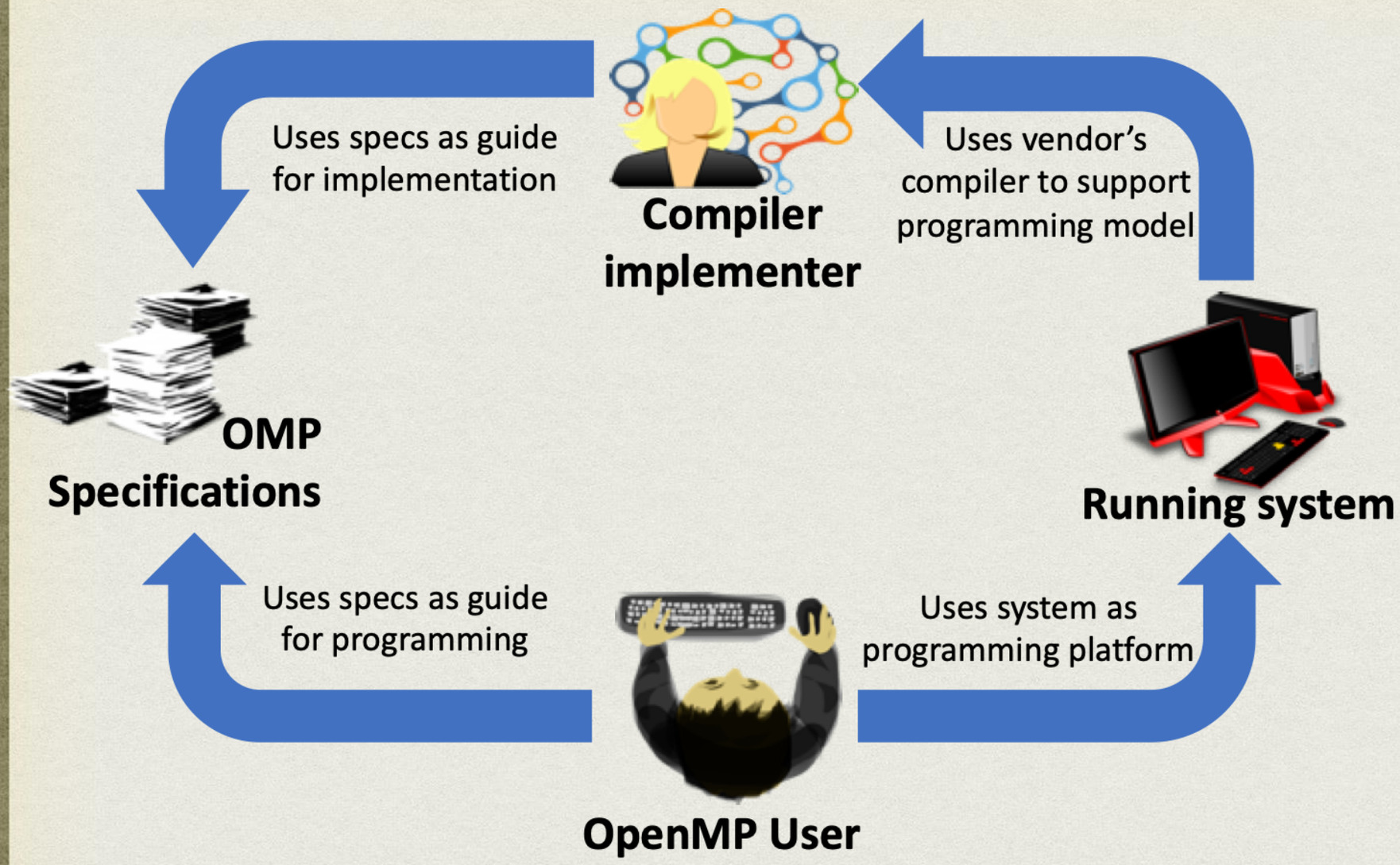




BEST PRACTICE #4:

AUTOMATE

OPENMP VALIDATION & VERIFICATION TESTSUITE



ECP SOLLVE OpenMP V&V

https://github.com/SOLLVE/sollve_vv

https://crpl.cis.udel.edu/ompvvsollve/

OpenMP

U.S. DEPARTMENT OF ENERGY | Office of Science

OAK RIDGE National Laboratory | OAK RIDGE LEADERSHIP COMPUTING FACILITY

ECP EXASCALE COMPUTING PROJECT

CRPL Sunita Chandrasekaran

UNIVERSITY OF DELAWARE

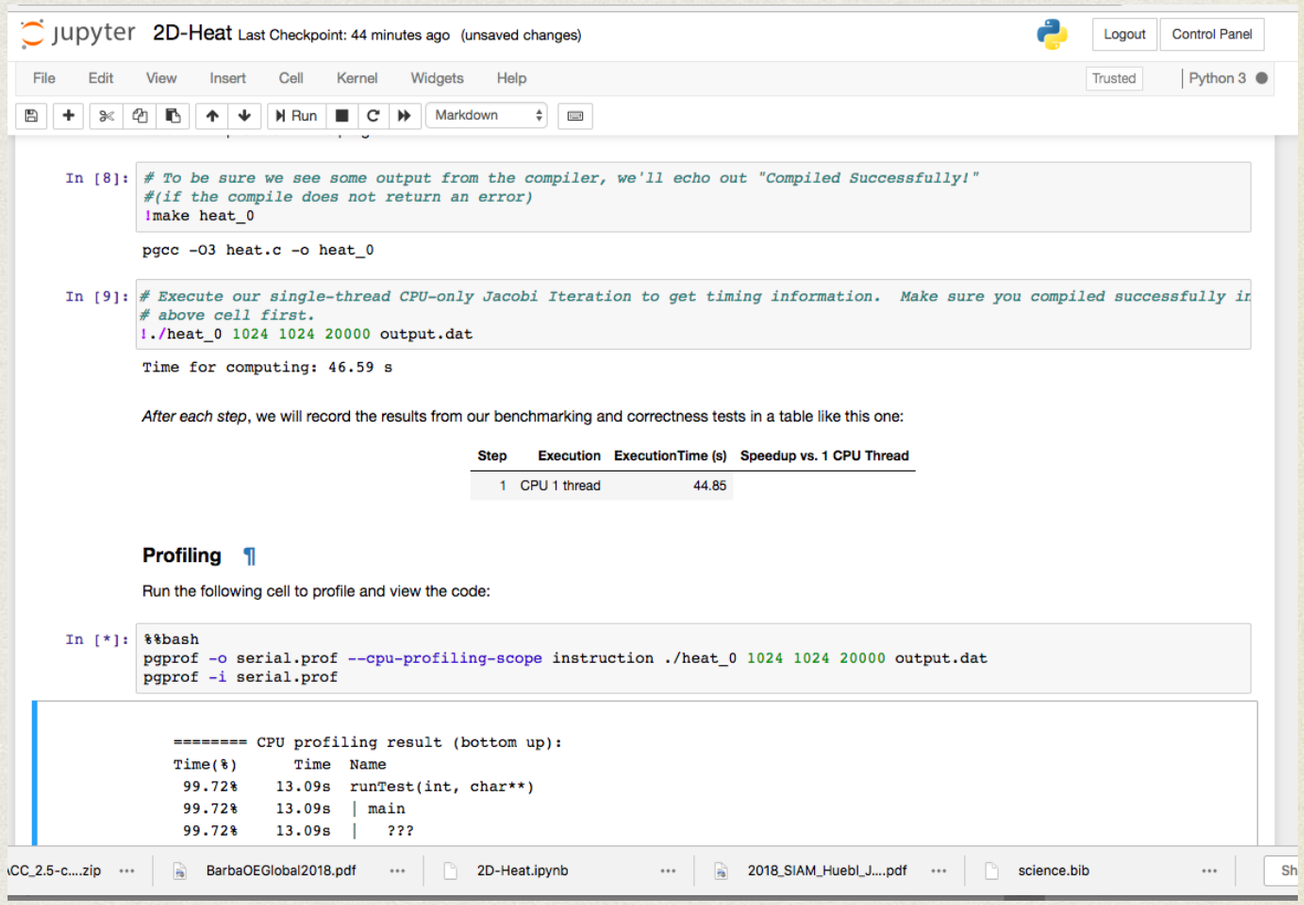
ECP SOLLVE OPENMP V&V

GitHub	Collaborative Environment
Integrated with issue tracker	Keep track of bugs and fixes
Pull request review system	Maintain code history
Website	One stop shop for info
Publication, Presentations	Documentation
Coding standards	Templates - external collaboration

BEST PRACTICE #5: DOCUMENT (pretty pls! :-))

DOCUMENTATION

- IPython notebooks
- Embed code as part of documentation
- Probability of a programmer updating the documentation when the code changes is high!
- Useful for training and education



BEST PRACTICE #6:
PAIR PROGRAMMING

HACKATHONS

- Profile the code, understand performance bottlenecks
- Start with a smaller kernel
- Use a simple system to compile/execute the code for starters
- Identify a good starting point
- Debugging - “when in doubt, comment it out, re-test”
- https://www.gpuhackathons.org/

Chandrasekaran, Sunita, Guido Juckeland, Meifeng Lin, Matthew Otten, Dirk Pleiter, John E. Stone, Juan Lucio-Vega, Michael Zingale, and Fernanda Foertter. "Best Practices in Running Collaborative GPU Hackathons: Advancing Scientific Applications with a Sustained Impact." *Computing in Science & Engineering* 20, no. 4 (2018): 95-106.

BEST PRACTICE #7:
OPEN SOURCE BUT…

OPEN SOURCE IS GREAT BUT….

- Not sustainable unless there is a community that will help with the sustainability

- Need help

  - From sponsors, funding organizations, interested vendors

  - To ensure continuity of software maintenance beyond the funded project period

SUMMARY

- Best Practice #1 Profiling
- Best Practice #2: Systematic Testing
- Best Practice #3: Report bugs
- Best Practice #4: Automate
- Best Practice #5: Document
- Best Practice #6: Pair Programming
- Best Practice #7: Open Source but…

My group in action :-)

CRPL
Sunita Chandrasekaran

Computational Research and Programming Lab