



Hands-On Activities

Better Scientific Software Tutorial

Jared O'Neal
Mathematics and Computer Science Division
Argonne National Laboratory

And the rest of the team

Supercomputing 2018
Dallas, TX
November 12, 2018



See slide 2 for
license details

License, citation, and acknowledgments



License and Citation

- This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/) (CC BY 4.0).
- Requested citation: Jared O'Neal, David E. Bernholdt, Anshu Dubey, and Michael A. Heroux, Hands-On Activities, Better Scientific Software tutorial, in SC '18: International Conference for High Performance Computing, Networking, Storage and Analysis, Dallas, Texas, 2018. DOI: [10.6084/m9.figshare.7309172](https://doi.org/10.6084/m9.figshare.7309172)

Acknowledgements

- **Alicia Klinvex developed some of the hands-on examples**
- This work was supported by the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research (ASCR), and by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration.
- This work was performed in part at the Argonne National Laboratory, which is managed by UChicago Argonne, LLC for the U.S. Department of Energy under Contract No. DE-AC02-06CH11357.
- This work was performed in part at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC for the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.
- This work was performed in part at Sandia National Laboratories. Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. SAND NO SAND2017-5474 PE



Hands-On Activities

Options

- **Kanban using GitHub Issues**
 - See module 03 Teams, slide 23
- **Simplest CI example**
 - https://github.com/jrdoneal/CI_HelloWorld
 - See also module 07 Coverage/CI, slide 14
- **CI example w/ multiple platforms and specific compiler versions**
 - https://github.com/jrdoneal/CI_Multiplatform
 - Instructions in README.md
- **Code coverage, testing and CI tutorial (C++)**
 - <https://github.com/amklinv/morpheus>
 - See following slide, then README.md
- **Code coverage, testing, and CI example (Fortran, C++)**
 - <https://github.com/jrdoneal/infrastructure>
 - Instructions in README.md

Suggested Approach

- **Make sure you are setup appropriately**
 - See module 00 Intro, slide 9
- Work individually or in small groups (2-3)
- Work at your own pace on whichever activities interest you
- Use your yellow sticky note to signal if you need help
 - And take it down afterwards
- Instructors will be circulating and available to help as needed
- **We're also happy to discuss other topics from the tutorial**

Special Notes for Morpheus Tutorial

- A code coverage and testing tutorial can be found at the Morpheus repository doxygen pages
 - <https://amklinux.github.io/morpheus/index.html>
- **STEP 1:** These exercises must be run on your own local machine or on a remote machine that you have access to.
- If you cannot generate your own gcov output, the associated lcov output is online
 - <https://amklinux.github.io/morpheus/lcovFiles/index.html>