

Open Science Grid Software Carpentry Workshop

Duke University
Oct 27-29, 2015

Balamurugan Desinghu, Mats Rynge,
David Champion, Rob Gardner,
Suchandra Thapa, Emelie Harstad

Introduction to Software Carpentry

- Non-profit organization - 1998 (Greg Wilson)
- Mission: Improve basic computing skills and practices of researchers (esp. science, engineering, medicine)
- Materials are open-source
- Instructors are volunteers

<http://software-carpentry.org>

admin@software-carpentry.org

Introduction to Software Carpentry

Problem:

- Most scientists' computing skills are self-taught
- Learn computing skills on need-to-know basis, limited time

Solution:

- 2-day “bootcamp”-style workshop
- Hands-on training, no slides
- Core curriculum:
 - Unix Shell programming (automating tasks) – *Emelie (day 1)*
 - Version control (Git and GitHub) – *Bala (day 1)*
 - Python (building modular, robust code) – *David, Suchandra (day 2)*

Results:

- Good programming practices
- Reproducible research, time-saving, efficient, re-usable

Open Science Grid - DHTC

❖ **Wednesday Night Special Event: What is High Throughput Computing and how can it speed up my research?**

6:00-8:00 p.m.

Gross Hall, 2nd floor Connection Space

Thursday Morning:

- Job Scheduling with HTCondor - *Mats*
- Troubleshooting OSG Job Submissions - *Mats*
- Submitting to OSG from local campus resources - *David*

Thursday Afternoon:

- Handling Data - *Suchandra*
- Scaling up Computing Resources - *Bala*
- Handling Job Dependencies - *Bala*
- Large Scale Computation with Pegasus - *Mats*

Logistics

- Location: Duke University – The Edge Workshop Room, Bostock Library
- Workshop website (with setup instructions and schedule/lessons)
<http://swc-osg-workshop.github.io/2015-10-27-duke/>
- Sticky-notes:
Red = need help Green = all good

Unix Shell

- What is the Unix Shell?
 - A program
 - Interact using CLI instead of GUI
 - Interprets commands, orders computer to execute, and prints out result
- Why use the Unix Shell?
 - Powerful toolset - automate repetitive tasks quickly
 - Easiest way to interact with remote machines
 - Unix-based operating systems are used on most High Performance Computing (HPC) systems

Unix Shell

- Shell commands can look cryptic at first
- Only need to learn about 10-15 basic commands
- We will be using the Bash shell (most popular)

Filesystem

