HPC Carpentry at Supercomputing ‘16

Abstract

In the pursuit of scientific research that is more correct, more widely applicable, or simply needs to be done *faster*, many researchers will run into

- data that is too big to fit on their personal machines or lab workstations, and/or
- code that takes too long to run

The next step for these researchers is generally to get access to a local HPC resource. However, *before* they can use multi-threaded software, MPI, GPUs, or run high-throughput analyses on large datasets, they need to learn a different set of skills to do so *effectively*, such as task automation, cluster computing, and managing parallel workflows. This training is generally missing from coursework, but offered by various groups including HPC center staff, system administrators, and research computing "facilitators".

HPC Carpentry is an effort to bring together these groups to design and deliver a full day workshop that teaches novice users these skills so that they can quickly become productive in an HPC environment. It is modeled on the lesson design and workshop practices of the Software Carpentry and Data Carpentry projects, which have found much success in teaching novice users "basic lab skills" and best practices in software development and data science.

Detailed description of the proposed tutorial

General description of tutorial content

The aim of HPC Carpentry is to teach novice HPC users (defined below) "basic lab skills for high-performance computing". The lesson material is designed to be covered in a full day, and includes

- an introduction to the command line and scripting,
- a background in remote computing, cluster structure and batch submission, and
- management of parallel (high-throughput and high-performance) workflows.

Although the workshop will focus primarily on a single HPC site (the Oklahoma State cluster at SC ‘16), it is designed with the objective to teach concepts more generally, and to enable users to easily translate concepts to their local sites.

Target audience

The target audience for HPC Carpentry at SC '16 may fall into two categories:

- Novice HPC users: attendees that are entering the field of high-performance computing, or high-throughput computing (HTC) but have little to no experience with the command line or remote computing.
- HPC Trainers: attendees that offer or are interested in offering training in high-performance computing for novice HPC users.

Tutorial goals and benefit to audience

- Provide entry-level audiences the skills necessary to begin using HPC resources: learners will be able to prepare and submit HPC and HTC jobs in a more educated, organized and efficient way, and analyze and report problems more effectively.
- Demonstrate the "Software Carpentry" workshop layout and practices and their applicability to HPC training: HPC trainers will be introduced to the Software Carpentry project, which aims to grow the network of trainers interested in teaching productive tools and best practices in scientific computing.

Relevance to SC '16 attendees

HPC Carpentry is primarily directed at community and outreach, and harmonious with some other tutorials that may be offered at SC '16.

Like the "Parallel Computing 101" tutorial that has been offered at Supercomputing for the past few years, HPC Carpentry aims to bring inexperienced users into the HPC community. The scope of the workshop is intentionally limited to focus on foundational concepts and respond to common interests of new users.

The "Best Practices for HPC Training" tutorial, where trainers are invited to present their teaching experience, is an excellent complementary tutorial with significant community overlap. Our hope is that these workshops do not overlap on the schedule, so that attendees may be part of both, and there will be scope for ideas to be exchanged.

Content level

This is an entry-level workshop with the following content level distribution:

Beginner: 100%

Intermediate: 0%

Advanced: 0%

Audience prerequisites

This workshop assumes very few prerequisites:

- Little to no experience with the command line or programming
- Domain expertise or developing domain expertise (users know *what* they want to use HPC for, but not *how* to use it)

The audience will be expected to bring a bring a laptop computer with wireless or wired internet for cluster SSH access.

Lesson material and presenters

As with all Software Carpentry lessons, the material for HPC Carpentry is developed collaboratively on GitHub. The presenters at SC '16 will be active contributors to the lesson material, and to discussions about how it should be delivered. Further, instructors are experienced teaching in teams and creating an interactive, comfortable learning environment. This will ensure that the workshop is presented as a cohesive whole, rather than a series of disparate talks.

Detailed course outline

The workshop will be divided in three sections, covering 6 hours with room for two 10 minute breaks.

The Unix Shell (90 min)

The aim of this section is not to teach learners basic unix commands, or syntax for bash scripting, but rather to motivate them to *automate tasks and develop pipelines*. Topics covered will include filesystem navigation and manipulation, redirection and piping of standard input and output, file permissions, and shell scripts.

Cluster structure and scheduling (80 min)

This section will help learners develop a strong mental model of the cluster, and how tasks get assigned and executed on the cluster. Learners will understand how nodes, CPUs, memory, etc., are organized, and "how it looks like one remote computer when it has a bazillion CPUs and several Terabytes". At the end of the section, they will submit a number of batch jobs for serial, parallel and task array workloads, trace their execution, and examine their results. Throughout, it will be stressed that different HPC sites have different setups, schedulers, policies and resources, and how learners can translate the concepts accordingly.

Parallel workflows (170 min)

This section will guide learners through the process of performing a large HPC simulation, and a high-throughput analysis of the resulting data. For this, they will develop the workflow for a specific research scenario, for example:

> “Lola was hired by a research lab to help prepare the purchase of a multi-million dollar experiment. The experiment is known to fail at temperatures that are too low or too high. She knows that the temperature changes follow a daily pattern, and she's written some code to simulate these temperature changes. After running this code and generating the temperature predictions, she determines how closely her predictions match the actual temperature readings she has for every day in the last year. The simulation would take too long to run and generate too much data for her lab workstation, so she will use the local University's HPC facility for this work.”

Learners will perform every step involved, including importing code and data, loading/setting up the required software, testing with small datasets, and job preparation and submission. They will work with parallelism-*enabled* software, so this section will not discuss any particular parallel platform - instead it will focus on understanding *how* and *why* parallelism works, and how to choose resources effectively.

Description of hands-on demos/exercises

The pedagogical approach of this workshop will be modeled on that of the Software and Data Carpentry communities and will utilize the following interactive components:

Live Coding: The content will be presented through a combination of speaking and live coding in front of tutorial participants. Presenters will both talk about HPC concepts as described in the outline, and demonstrate them in front of participants, with the expectation that participants follow along and run the same commands as the presenter.

Example: Logging onto a large-scale computing system
Presenter draws a diagram of a user's local computer and the login node of a remote computing system, and describes what is happening when users log in. She then demonstrates logging into the provided cluster using the "ssh" command; all participants use the same command to log in to the same cluster.

Exercises: In addition to the content presentation, the tutorial will contain regular exercises where participants are expected to work individually or in groups on a set of challenges. There will be an exercise or independent activity for about every 15-20 minutes of instruction.

Example Exercise: Submitting jobs to a batch scheduler
`submit-job.pbs` submits a request to run our script on the test file `test.csv`:

```
python analyze.py test.csv
```

Modify the script `submit-job.pbs` to instead submit a job to run the analysis on our actual input data, `temperatures.csv`. This job will need more than 1GB, so you will also need to change the amount of memory that `submit-job.pbs` is requesting.

Support: Throughout the workshop, participants will be provided with two colors of sticky notes to serve as a visual indicator of progress: one color to indicate success or understanding, the other to indicate confusion or a question. This allows the lead presenter to respond to the audience at a glance. In addition, when presenters are not leading a session, they will circulate among participants to answer questions or address immediate issues. Finally, an online note-taking page (etherpad) will be set up for participants to take collaborative notes and chat with each other.

Exercises and demonstrations will be fully tested on the planned resources before the workshop.

Lesson Materials: The materials developed for this workshop will be similar in format to those developed by Software Carpentry. For an example of what the final materials will look like, see:

- Software Carpentry lesson on the Unix Shell: http://swcarpentry.github.io/shell-novice/

To see an example of how lessons will incorporate live commands, diagrams, and follow-on exercises, see:

- http://swcarpentry.github.io/shell-novice/03-pipefilter.html

Computing Resources

This tutorial requires access to a large scale computing system that supports remote login and uses a batch scheduler. We will have temporary accounts created for all participants on the Oklahoma State University computing cluster for this purpose. We are also applying for an education allocation on XSEDE which, if granted, will complement the resources available through Oklahoma State.

Participants will need to bring their own laptop or portable computing device that can use WiFi. The tutorial will require using a bash shell (or similar tool); installation instructions will be part of the workshop materials.