HPC Carpentry - practical, hands-on HPC training

Report from Birds-of-a-Feather session, 14 Nov 2017, SC17, Denver, USA
https://hpc-uk.github.io/sc17-hpccarpentry-bof

Session Leaders

Chris Bording
Chris.Bording@ibm.com
IBM Research

Martin Callaghan
m.callaghan@leeds.ac.uk
University of Leeds, UK

Bob Freeman
rfreeman@hbs.edu
Harvard Business School, USA

Christina Koch
ckoch5@wisc.edu
University of Wisconsin - Madison, USA

Tracy Teal
tkteal@datacarpentry.org
Data Carpentry, USA

Andy Turner
a.turner@epcc.ed.ac.uk
EPCC, The University of Edinburgh, UK

Introduction

As more and more research comes to depend on the use of high performance computing to progress, there is a clear need to equip researchers with the basic, practical skills so they can exploit HPC resources in an effective way. These skills are particularly important for people just getting started with high-throughput and high-performance computing, as they make researchers more effective and make the most efficient use of resources. Given the success of the Carpentry series in equipping researchers with the basic, practical skills required to manage and develop their software and scripts (Software Carpentry) and data analysis (Data Carpentry) it is natural to take the Carpentry approach and apply it to HPC.

As for Software and Data Carpentry, we wish this to be a community effort with the worldwide HPC community coming together to develop HPC Carpentry so it as useful as possible. With this in mind we ran a Birds-of-a-Feather (BoF) session at Supercomputing '17 to raise awareness of HPC Carpentry within the community; share information on work already done; and to gather input from the experts in the HPC community on HPC Carpentry. Within the BoF session we covered the following topics:

- BoF Goals
    - What do attendees want to get out of this BoF?
- HPC Carpentry Learner Profile
    - Who are the target audiences for HPC Carpentry?
- HPC Carpentry Course Design
    - What should the learning outcomes be?
    - What lessons should be included?
    - How (if any) much site/facility specific material would be needed?
    - How long should the course last? Should we support different lengths?

The BoF session took the format of three sections matching the three main topics in the list above. Each of the sections started with a short (5 min) scene-setting presentation from one of the session organisers followed by an activity for the BoF participants. These activities are described below along with the results from the activities.

The remainder of this report takes the following form. In the next section, we describe the outputs from each of the BoF activities. After this we compare the suggested topics and structure from our BoF to those in the HPC Carpentry implementation already being used by Compute Canada. Finally, we draw some broad conclusions from the BoF activities and outline the next steps in creating and running an Open Source, community HPC Carpentry course. The Appendices to this report include the full results from the BoF activities.

Outputs from Activities

BoF Goals

In this activity we used the SC17 voting tool (Q&A Live) to allow attendees to suggest reasons for attending and their goals for the BoF and also vote (both positively and negatively) on reasons that other people had added. This lead to a ranked list of goals for the BoF session. Fifteen goals were suggested, these are listed below in their ranked order according to votes:

1. Votes: 33 Understand requirements for HPC Carpentry training from HPC community
2. Votes: 19 Come to some sort of consensus on a curriculum for HPC Carpentry
3. Votes: 18 What is HPC Carpentry?
4. Votes: 17 What audience are you designing your curriculum for? e.g. Very introductory like Software Carpentry (i.e. what is git? –> what is MPI?)
5. Votes: 15 Determine best practices for HPC carpentry sessions (tools or resources that can be used)
6. Votes: 13 What is the current state of HPC carpentry?
7. Votes: 12 Build links with interested people to help build the most useful HPC Carpentry training
8. Votes: 7 Engage wider HPC community with HPC Carpentry activities
9. Votes: 6 What is HPC carpentry ?
10. Votes: 5 Provide input on HPC Carpentry curriculum.
11. Votes: 3 Propose HPC carpentry as a home for links to vetted and useful training, such as that published by TACC or Shodor.
12. Votes: 2 What ideas, decisions and processes can I walk away from the BoF with that will improve my code craftsmanship going forward?
13. Votes: 1 The software carpentry learner profile example: https://software-carpentry.org/audience/
14. Votes: 1 Understand what HPC Programing actually is? Is it multi-threaded, multi-node, GPU/Accelerators ? A hybrid of all three?
15. Votes: -4 What languages should we focus on for HPC Carpentry?

As we can see, some of these raw goals correspond to the same thing. If we merge the similar goals we end up with the following list of ten goals for the session provided by the attendees:

1. (33 Votes) Understand requirements for HPC Carpentry training from HPC community
2. (24 Votes) Find out what HPC Carpentry is
    a. Votes: 18 What is HPC Carpentry?
    b. Votes: 6 What is HPC carpentry ?
3. (24 Votes) Help design the HPC Carpentry curriculum
    a. Votes: 19 Come to some sort of consensus on a curriculum for HPC Carpentry
    b. Votes: 5 Provide input on HPC Carpentry curriculum
4. (19 Votes) Build HPC community engagement with HPC Carpentry
    a. Votes: 12 Build links with interested people to help build the most useful HPC Carpentry training
    b. Votes: 7 Engage wider HPC community with HPC Carpentry activities
5. (18 Votes) Understand the target audience for HPC Carpentry
    a. Votes: 17 What audience are you designing your curriculum for? e.g. Very introductory like Software Carpentry (i.e. what is git? –> what is MPI?)
    b. Votes: 1 The software carpentry learner profile example: https://software-carpentry.org/audience/

6. (15 Votes) Understand what tools and resources should be used to help teach HPC Carpenty
    a. Votes: 15 Determine best practices for HPC carpentry sessions (tools or resources that can be used)
7. (3 Votes) Propose HPC carpentry as a home for links to vetted and useful training, such as that published by TACC or Shodor.
8. (2 Votes) What ideas, decisions and processes can I walk away from the BoF with that will improve my code craftsmanship going forward?
9. (1 Votes) Understand what HPC Programing actually is? Is it multi-threaded, multi-node, GPU/Accelerators ? A hybrid of all three?
10. (-4 Votes) What languages should we focus on for HPC Carpentry?

It is unsurprising that many of the goals for attendees match those described in the description of the BoF session provided ahead of the conference as we would expect attendees to have read the description and made a decision to attend based on the fact that the session goals align with their interests. During the session, this gave us (as session leaders) the confidence that the activities and discussion topics we had planned would be of interest and useful for the session attendees - we had plans in place to modify these activities and topics if this exercise revealed different areas of interest from the attendees). Having said this there are a few interesting features:

- One goal for many attendees was to learn what HPC Carpentry actually is. Although this was not an explicit goal in the session it was implicit that the session would give attendees information on this. One of the session leaders (Christina Koch) gave a short presentation outlining the Carpentry approach to training during the session.
- There was less interest in the technical instantiation of HPC Carpentry - the technology used to run examples and teach students - than there was on the high-level parts of the course - target audience, syllabus, etc. This ties in with the idea that there are general HPC concepts and techniques that can be taught that are of use no matter which facility you are actually using.
- The attendee-suggested goal of understanding which [programming] languages HPC Carpentry should focus on received a negative score - a number of attendees thought this was not of interest for the session. This ties in with the previous observation that the session goals were thought to be at a higher level than the technological implementation of HPC Carpentry. It also reflects that most attendees thought that the target audience for an initial HPC Carpentry course would not necessarily have any programming experience (see Learner Profile activity below).

HPC Carpentry Learner Profile

One of the top goals for both ourselves and the BoF attendees was to understand who we are aiming HPC Carpentry courses at (at least in the first instance). Attendees were split into 5 groups (with a session leader acting as a discussion moderator for each group. They were then asked:

1. Brainstorm the characteristics of the novice HPC user from your perspective? Include skills (e.g. I am not comfortable with the command line) and mindset (e.g. I am intimidated learning about HPC). You can produce multiple, distinct profiles if you wish.
2. Once you have the characteristics, discuss and put into a list ordered by your collective evaluation of importance.

All groups managed the brainstorm activity and three out of the five groups managed to produce a rank-ordered list of the most important characteristics.

Brainstormed characteristics

Analysing the feedback from the five groups, there were a number of common characteristic groupings. The individual statements from the activity that gave rise to these groups can be seen in Appendix A below (along with a note of which group produced the statement).

- I do not have the basic technical skills to use HPC
- I am not sure how HPC can help my research?

- Just give me what I need to get on with my research on HPC
- What is an HPC system anyway? How does it differ from the computer I am already using? What about “the Cloud”?
- HPC and programming are awesome, but I do not know much, tell me more!
- How do I get help and/or help myself when I get stuck?
- I feel unsure and out of my depth with HPC
- I do not understand the shared nature of HPC systems
- I am experienced technically, what do I need to know about HPC?
- Broad range of expertise and experience among researchers

It is interesting that these characteristics seem to cover a wide range of different skill levels for a learner who may be attending what was specified as an introductory course. While there are statements here that clearly describe a learner who has not encountered ‘traditional’ HPC use (i.e. command-line access, multi-user systems) there are also statements that would describe a user who has already been using HPC systems but feel they lack the understanding required to get the most out of the resource or support themselves if they run into problems.

Most important characteristics

Three groups produced sets of what they considered the most important characteristics of learners. Analysing these sets leads to 4 main characteristics of a potential learner on a novice HPC Carpentry course:

- Vague idea of what HPC is but not sure how this translates to their research.
- Lack of experience with Linux, command line, text editors, batch systems etc. i.e. the working environment is foreign to me.
- Lack of knowledge about what HPC systems are: how are they put together, how do they enable faster/larger calculations and how are resources shared?
- Want to know how to support themselves when things don’t work and keen to learn more.

From the groups that attempted the second part of the activity we can see that the most important characteristics lean more towards the complete novice - someone who is encountering HPC for the first time and requires a mix of practical skills to help them use the facility and conceptual knowledge to put the skills into context and increase their ability to be self-sufficient on HPC facilities.

HPC Carpentry Course Design

Another key topic for all involved in the session was what should be taught as part of HPC Carpentry.

This session consisted of two activities:

1. Brainstorm the 5 – 10 skills you would like the provided user profile to have by the end of the workshop.
2. An example workshop outline with lesson modules is provided. Assuming a 1 day workshop:, What lessons are missing? What should be dropped or combined? If a longer workshop were given, what could be in Day 2?

All groups provided notes on the first activity but only some of the groups managed to find time to move on to the second activity.

The full notes from each of the groups for this exercise can be seen in Appendix B below.

Brainstormed Skills

Commonly-identified skills and topics

Looking at the notes from each of the groups we can identify a number of key, generic skills or topics that were commonly described.

- Why use HPC?
- Login, interactive access (ssh); transferring data (scp)
- Understanding HPC jargon
- Basic understanding of HPC architectures and practical implications for their use
    - HPC system components: processors, memory, I/O, interconnects
    - Parallelism
    - Performance: strong/weak scaling, speedup, best practice and basic metrics
    - High-level comparison: laptop, HPC, cloud
- File systems
    - Multiple file systems
    - Users and groups
    - File permissions
    - File organisation
- In-terminal text editors
    - Why do I need this skill?
- modules, environment, applications
    - Executing applications
    - Executing parallel applications (OpenMP, MPI)
    - The PATH
    - module load ...
    - Compiling (parallel) applications
- The batch system and writing job submission scripts
    - Basic concepts
    - Batch system options
    - STDOUT/STDERR
    - Different job types: traditional MPI HPC, task farming, ...
    - Best practice and troubleshooting
- Shared system etiquette
    - Being a good HPC citizen
- Troubleshooting strategies

See the “Comparisons with existing HPC Carpentry efforts” section below for an analysis of how these skills and topics compare to the different ways in which HPC Carpentry has evolved so far. This list of skills does not contain any controversial entries; however, during the discussion there were a small number of voices that argued for a larger set of technical skills in the areas of software engineering (e.g. the use of version control and testing) and science/mathematics (e.g. linear algebra, algorithm design). The majority of attendees thought that using an HPC system effectively was not (at least in the first instance) dependent on having these skills. (The analogy used was that many people can drive a car competently without a detailed technical understanding of how it functions.) These skills may be important for particular uses of HPC but were not generally required for someone matching the learner profile described earlier to be able to use an HPC system.

Some skills were thought to be specific to the HPC system being used:

- How to get account and access the system
- System specific information, e.g.
    - Key information on using the system
    - File retention policies
    - How to get help

It would generally be up to local sites to create modules covering these areas to go into their runs of HPC Carpentry. We would hope many sites would contribute their local modules back to the community HPC Carpentry effort so there would be a range of templates that a new site could use as the starting point for this (and we will encourage sites to do this).

Prerequisite skills

A number of groups thought that basic Linux shell use and scripting should be a prerequisite for course attendees. This basic prerequisite could be met by completing the free Codecademy bash track online (https://www.codecademy.com/catalog/language/bash) before attending or attend another course covering this topic, e.g. Software Carpentry.

This is a reasonable course requirement as, without basic Linux shell skills, it will be very difficult to fit the other key HPC skills and topics into a short course of this nature.

More advanced skills

Finally, one skill was seen to belong to more advanced or different courses: *Workflow optimisation*. It was thought that this topic is domain/researcher specific and so it may be difficult to design a general module on this topic. One approach to including this topic would be to have workflow modules for different research communities that could be used if HPC Carpentry was being taught to attendees who all came from a particular group. This has the advantage that the module could be provided by a person who is familiar with both HPC and the research area (such as a Research Software Engineer, RSE: http://www.rse.ac.uk). Another approach could be to add a consultancy to the end of a HPC Carpentry course where the expert trainers could talk to attendees about their research: where HPC fits and how to optimise their workflow.

Other skills and topics

There were a number of ideas that were suggested by single groups rather than appearing across multiple groups suggesting that there was not the same level of agreement that they were important for an HPC Carpentry course:

- Version control (git) in HPC workflow
- Markdown for documentation

These topics could be covered in optional modules provided by the community if there was sufficient interest. The version control module already exists in Software Carpentry and so would be easy to incorporate into a modular HPC Carpentry course if required.

Thoughts on example syllabus

The BoF attendees were then provided with the draft course plan for an HPC Carpentry course from hpc-novice on the Software Carpentry Github organisation (http://swcarpentry.github.io/hpc-novice/) and asked to consider how well the topics matched to the skills they had just identified. In particular:

- If anything should be dropped from the example course plan
- If anything was missing from the example course plan
- If the course was to run over 2 days, which topics should be placed on day 2

Suggestions for topics to drop included the “Introduction to the Shell”. As noted above, some groups thought that this should be a prerequisite for the course and the required skills could easily be obtained through other courses (including free, online options). It was also noted that “Good Citizenship on Shared Systems” may vary a lot from site to site and so maybe this should move into site-specific modules.

Suggestions for additional topics/modules included modules on:

- basic conceptual information (why HPC? How are HPC systems put together? etc.)
- understanding HPC jargon
- basic performance metrics (or, how to understand if your application is working well on the HPC system).
- how to develop version controlled workflows
- how to write documentation using markdown
- how to get support and how to frame requests for help (may be part of local, site-specific modules)

Where these match key skills identified by multiple groups there is a strong case for including these into a future HPC Carpentry course; these include: the *basic conceptual information*, *understanding HPC jargon*, and *how to get support*. The remaining suggestions on workflows and using markdown look out of scope for a generic HPC Carpentry course, but modules on these may be of use for specific communities.

Multiple groups commented that the example course design looked like there was too much for a single day. Suggestions for topics that should be moved to day 2 included:

- Intermediate shell topics
- Testing jobs while running interactively
- Using software modules
- Running parallel tasks
- Running parallel code

One of the key aims for HPC Carpentry is to create a modular structure that allows the material to be used in different orders that make sense for different audiences, systems and course lengths.

Comparisons with existing HPC Carpentry efforts

There have been a number of previous initiatives on HPC Carpentry. In this section, we compare the outcomes from our BoF session with two existing initiatives that are at different stages of development:

- Compute Canada HPC Carpentry - this is already being used to train Compte Canada users.
- HPC Novice - this is a community version of HPC Carpentry with some suggested lesson modules but no implementation yet.

In both cases, we compare:

- Learner profiles: how do the learner profiles for these two versions compare to the profiles produced by the community in the BoF? Where are the similarities and differences?
- Course design: How do the course designs for these two versions compare to the suggestions from the BoF? Are there any topics in the courses that were not highlighted in the BoF or any from the BoF that do not exist in current courses? What is the length of these courses and is this similar to that proposed by BoF attendees?

Learner profiles

- **BoF:** The BoF learner profile focused on the person's background and gaps in knowledge, specifically that a novice learner has some vague idea of what HPC is, but is not sure how it applies to their research. Furthermore, this person likely has limited-to-no experience with the command line, how to use an HPC system, or even how an HPC system works. This type of learner will feel insecure and unsure about HPC.
- **hpc-novice:** This learner profile contains some generalities about background skills that are similar to the BOF profile, including limited-to-no command line experience and no experience of an HPC system. It is assumed that the person believes HPC will help them (but may not be sure how). It also includes a specific example of a researcher who has run a few tools on her own computer using the command line (mostly through trial and error) and wants to use a cluster instead.
- **HPC-Carpentry hpc-intro:** There is no formal learner profile for these lessons, but they were developed for potential users of Compute Canada resources. These would be researchers who know they need to use large-scale computing to tackle their problems but need to learn how to access and use Compute Canada's resources specifically.

Each of these profiles capture different facets of the novice HPC user. What they have in common is the assumption that while motivated about their research, the typically HPC Carpentry learner has a mixture of motivation (wants to get more work done!) and demotivation (intimidated by interface, new paradigms, and technical terms/assumptions). It is therefore extremely important that the HPC Carpentry curriculum address their motivations -- that it will help show them how using an HPC system will accelerate their research and

how to use it -- and also work to dispel their concerns, including definition of terms (and relating terms to what they already understand!) and a supportive, interactive workshop environment, as is typical of Software and Data Carpentry workshops.

Course design

Built on these learner profiles, both partially developed courses contain common components with each other and with the discussions held at the Supercomputing BoF. The hpc-intro curriculum developed by Compute Canada staff has a lengthy introduction to the concept of a large scale remote system, including definition of technical terms and why you might want to use such a system. While the text is not fully developed, the hpc-novice curriculum has a similar opening episode. Both curricula then contain

- an introduction to using a command line interface and logging into a remote system
- how to submit jobs using a batch system, including submission, monitoring the queue, removing jobs, and some basic troubleshooting
- how to use modules to access specific software

Finally, each course as outline contains a section on being a good citizen on a shared system.

These similarities align with the primary topics discussed at the BOF, addressing learner insecurity about using a different interface (command line), performing the logistics of job submission, and understanding what makes a remote, shared system different than running on their normal computer. If anything, even more emphasis could be placed in the two draft curricula on not just what an HPC system is, and how it differs from a single computer, but a clear link for participants as to why an HPC system will benefit them by allowing their work to run faster.

The two draft curricula vary in a few ways:

- hpc-novice has two episodes on scheduling parallel tasks, and running parallel code
- hpc-intro has a section on benchmarking
- hpc-intro has an episode on ways to transfer files

Both of these address in their own way the question of “how do I know it’s running faster?”, which was one of the main questions brought up in the discussion of a learner’s goals. Certainly, the BOF discussion and the inclusion of these topics in hpc-intro and hpc-novice suggest that we want this course to communicate the benefit to researchers of using a large-scale, parallel system, not just how to use it.

The only major omission from both curricula that came up in the BOF was interacting with filesystems and more advanced file manipulation (permissions, users and groups).

In general, a thorough introduction to the concepts described above seems like it will take at least a day, although, one key goal for these curricula is that they be sufficiently modular that different sites can shorten or lengthen them as necessary, based on their audience and system.

Conclusions and Future Work

Through this BoF session we managed to meet many of the goals that the attendees specified at the start of the session; the top five session goals from above:

1. Understand requirements for HPC Carpentry training from HPC community

   This was achieved by the attendees at the BoF (who represented a wide range of different groups from within the HPC community) coming together to discuss and prioritise their requirements in terms of both the learner profile of who would attend such a course and the proposed set of skills/topics that HPC Carpentry should aim to teach.

2. Find out what HPC Carpentry is

   Attendees at the BoF gained a good overview of the current status and direction of HPC Carpentry along with an idea of what the Carpentry approach to training entails (side-along practical training, train the trainers, community design, etc.)

3. Help design the HPC Carpentry curriculum

   One of the main activities at the BoF was to allow attendees to have input into the design of the HPC Carpentry curriculum.

4. Build HPC community engagement with HPC Carpentry

   The BoF brought together a wide range of people from the HPC community and provided them with information on how to engage with HPC Carpentry going forwards (by joining appropriate mailing lists and groups). We have also engaged specifically with Compute Canada following this BoF session to ensure the experience they have already have from an HPC Carpentry-like course is used to build the best community resource possible. We will reach out to the community for further input and to keep them informed of progress as development of HPC Carpentry moves on.

5. Understand the target audience for HPC Carpentry

   One of the main activities of the BoF was to design a learner profile for the HPC Carpentry course.

The learner profile that came out of the session described a researcher who has limited-to-no experience with the command line, how to use an HPC system, or even how an HPC system works and so will feel insecure and unsure about HPC. The profile focused on the person's background and gaps in knowledge, specifically that a novice learner has some vague idea of what HPC is, but is not sure how it applies to their research and wants to find out how to use this tool. When compared to learner profiles for other previous HPC Carpentry initiatives, there are many similarities which reveal a mix of motivations and demotivations (or barriers) for the learners described. The goal of HPC Carpentry should be to equip researchers with the practical skills to overcome the demotivational aspects (be that that the skills required to use HPC facilities, the knowledge to understand why one approach may work better than another, or even the confidence to seek help without feeling like they are asking stupid questions).

The proposed curriculum is non-controversial and contains the elements required to give the learner the practical skills described above so they can use HPC effectively to aid their research. The concept of modularity came out strongly in the discussions of the curriculum. A modular approach would give the flexibility for HPC Carpentry to be molded to meet the requirements of different groups of learners (say from a specific research community) and to add modules describing key facility-specific information that may be required.

The immediate next step is to develop the initial, core set of modules for the first runs of HPC Carpentry. We are already in discussions with Compute Canada on working with them to modularise and update the material they already have in there *hpc-intro* HPC Carpentry course to provide a basis for these initial modules. This initial pass will then be disseminated to the wider HPC community for comments and additions. At this point we will also generate example facility-specific modules.

We plan to run the first instances of HPC Carpentry before SC'18 and will submit a follow-on BoF to continue the community development of HPC Carpentry by sharing experiences from the initial events and gathering further input from the HPC community.

Session Leader Profiles

Chris Bording

Chris.Bording@ibm.com
IBM research

Chris Bording is a Senior Supercomputing specialist at the Pawsey Supercomputing Centre in Perth, Australia. Chris has particular experience in helping researchers from the bioinformatics community exploit advanced computing resources to improve their research. He has also worked on developing training for a wide range of user communities and improving the ease of access to HPC resources.

Martin Callaghan

m.callaghan@leeds.ac.uk
University of Leeds, UK

Martin is a Research Computing Consultant at The University of Leeds, where he manages the University HPC and Research Computing training programme for research staff and students. Martin has a background in training and research support and is a Software Carpentry trainer and instructor trainer. His research interests are in Deep Learning for text analytics and he has a particular interest in encouraging under-represented research communities in the Arts and Humanities to explore the relevance and potential applications of High Performance Computing to their research.

Bob Freeman

rfreeman@hbs.edu
Harvard Business School, USA

Bob Freeman is Director of Research Technology Operations at Harvard Business School. A former EvoDevo bench scientist, his 10 years of bioinformatics work and prior research computing work helped inform his leadership role with the ACI-REF program. Now in its third year, this program and now similar others enable on-campus facilitators to enable researchers across all disciplines to more effectively use high-end computing resources in their research, through outreach, training, & consultations. Bob now currently works to engage social science and business researchers to use advanced computing resources to ask bigger science questions and to accelerate their research outcomes. http://bit.ly/whatAboutBob2

Christina Koch

ckoch5@wisc.edu
University of Wisconsin - Madison, USA

Christina Koch is a research computing facilitator in the Center for High Throughput Computing (CHTC) at UW Madison. She has an academic background in mathematics and diverse experiences in computing, education, and online collaboration, including teaching software and data skills workshops with international non-profits Software and Data Carpentry. As a facilitator with CHTC, she works with researchers to expand and then transform their work using the power of large-scale computing.

Tracy Teal

tkteal@datacarpentry.org
Data Carpentry, USA

Tracy Teal is a co-founder and the Executive Director of Data Carpentry. As an Assistant Professor at Michigan State University in bioinformatics, she saw that effective data skills have become foundational for research and that data training needs to scale along with data production. At MSU, she also worked as a Research Specialist for the Institute for Cyber-Enabled Research to enable research and training for HPC analysis of genomic data.

Andy Turner

a.turner@epcc.ed.ac.uk
EPCC, The University of Edinburgh, UK

Andy Turner leads the application support teams for the UK national HPC services ARCHER and Cirrus. He is also heavily involved in HPC training at EPCC and was course organiser for the online distance learning course: Practical Introduction to HPC that launched in 2016. Andy has a particular interest in enabling new user communities to make use of HPC and the use of novel user engagement to improve the HPC user experience.

Appendix A: Collected responses for Learner Profile exercise

Analysing the feedback from the groups, there were a number of common characteristic groupings:

- I do not have the basic technical skills to use HPC
    - G2: Don't know LINUX, may know Windows
    - G2: Thinks a computer is a box under a desk or a laptop
    - G2: May be most comfortable with GUI
    - G2: No familiarity with command line
    - G3: I am unsure what the terminal is, how it works, etc
    - G3: I do not know what version control is
    - G3: I am a student and I do not know how to get access to the remote system I have been granted
    - G4: I don't know Linux but HPC systems are typically running Linux
    - G4: I don't know how to create and edit code
    - G4: Comfort with command line environment in a linux environment.
    - G4: Bash.
    - G4: I prefer graphical or web-based interfaces.
    - G5: I've never used a command line.
    - G5: I've never used a compiler.
- I am not sure how HPC can help my research?
    - G1: Why do I care?
    - G2: New graduate student in sciences/engineering
    - G2: Are only using HPC because their Advising professor told them to
    - G2: Biologists becoming bioinformaticians
    - G2: Usually stuck or bogged down due to limitation of current machine/resources
    - G4: Do I have a use for HPC?
- Just give me what I need to get on with my research on HPC
    - G1: I just want to do my science
    - G1: I want it to be interactive
    - G1: Just show me what I need to do to get my results
    - G3: I have a code with a new dataset that's too big to fit in memory/needs multiple cores now
    - G4: My supervisor wants me to run his favourite modelling code. How do I do that?
    - G4: I know my science, I have code, how do I use this tool to accelerate my science?
    - G4: How can I run my analytics on the data from my instrument using the cluster?
    - G4: I have Malab code that is too slow now. Can I speed it up or use something else easy to get results faster
    - G4: I want to try using a GPU for my problem.
- What is an HPC system anyway? How does it differ from the computer I am already using? What about "the Cloud"?
    - G1: Why aren't things just faster automatically?
    - G1: Is this the fastest it can get ?
    - G1: Why is running more slowly?
    - G1: Why does my program not run on multiple nodes? And what do I have to do to have it run on multiple nodes?
    - G1: How much processing power is overkill ?
    - G1: Users have little idea of the resource they need (memory / CPU) so for safety go for exclusive sessions everywhere
    - G2: Does not know what parallel programming models are, parallel vs. SMP models, multicore vs multinode
    - G2: Thinks performance is how fast they can run Doom 4
    - G2: Bases notions of storage cost on $200 8TB drives at Frys or Microcenter
    - G2: Expect programs to run magically faster on a cluster
    - G3: I do not understand the difference between a cluster, supercomputer, cloud, etc
    - G4: Overview of HPC technologies - eg what's a GPU and what is it good for?

    - G4: Concepts of why we use HPC systems and what they are
- HPC and programming are awesome, but I do not know much, tell me more!
    - G1: I just want to learn the “cool” stuff
    - G1: I’ve never programmed, how do I make cool things?
    - G3: Signed up for Software carpentry but I know there are things that it won’t cover
    - G5: Already have Software Carpentry and now want to apply that to HPC.
    - G5: I’ve just started working on a complex software problem with no computer science skills. How do I get help and/or help my self when I get stuck?
    - G1: Why does it take so long to get things done? (software installation assistance; head node compiling/bottleneck; single core desktop vs single core HPC)
- I feel unsure and out of my depth with HPC
    - G1: My novice HPC user is intimidated by HPC
    - G2: Researcher in soft/social sciences, scared by the thought of programming/scripting...
    - G2: ...but, brave enough to show up to an HPC Carpentry session
    - G4: I don’t have mentor with HPC experience (pioneer in group)
- I do not understand the shared nature of HPC systems
    - G1: What is a scheduler and why do I need to use it?
    - G1: Why can’t I have root or administrator privileges
    - G1: Why do I have to wait
    - G1: Obviously I can run on the login node.
    - G1: Novice users don’t understand that resources are competed for with the other users\
    - G2: Multi-user system is foreign to them
    - G3: I don’t know how to use my system scheduler
    - G4: Understanding of job submission in batch environment.
- I am experienced technically, what do I need to know about HPC?
    - G1: How to debug my program on multi threads ?
    - G1: I’ve been programming for years, just run me through the basics.
    - G2: Somewhat comfortable, but need help with optimization
    - G3: I do not know how the linker works, I don’t understand why my dependencies are linking to different shared libraries, and what that means
    - G3: I am trying to use a profiler for the first time with MPI code and I don’t understand what normal MPI behavior is
    - G5: Informal education in programming (self taught)
    - G5: Lacks knowledge or experience of exploiting parallelism.
    - G5: I lack knowledge of optimisation or low level performance.
- Broad range of expertise and experience among researchers
    - G2: Broad spectrum – some quite good programmers, just not familiar with HPC
    - G2: All are good scientists in their own field with computing or data needs
    - G4: Generally have a science background, but not programming or comp.sci. Have taken 1 or 2 basic programming courses.
    - G4: Have run my apps on my laptop, but now want to go to the next level.

Appendix B: Collected responses for Course and Module Design exercise

Group 1

Brainstormed skills

- How do I get an account? How to get access? What is the status of my account?
- How to sign on to the system - portal? Ssh? Off campus?
- Cluster Architecture (building a mental model of an HPC system starting from their desktop/laptop, highlight the system components and translate them to HPC system architecture)
    - What is a compute node
    - What is a storage node
    - Login vs compute node

    - Network and it's importance
    - File system (shared)
- Basic linux -
    - what is a shell?
    - Basic commands, navigation, creating and saving files
- Basic scripting
- Editor
- How to be a user
    - File management
    - being in a group
    - Center policies - how long can I keep my files, backup policies, scheduled downtime
    - Being a good citizen
    - How to get help
    - How to write up your request for help...what to include (error message), system name...
- Batch scheduler:
    - What is it?
    - why do we need it?
    - how do I use it?
    - How do I run a job?
    - What is a network?
    - Understanding exit codes - why jobs fail, stop, etc.

Group 2

Brainstormed skills

- Ability to submit jobs to the cluster
- Ability to schedule Jobs
- Understand resource limits in scheduler
- File system and transfer basics (Globus?)
- Understand environment variables/execution environment
- Understand Remote access to systems
- Appreciation for the meaning of Speed-Up
- Process automation
- Understand process control
- Understand pipes, background/foreground, redirection, stdio/sterr
- Understand impact of data placement, understand parallel file system and data locality, understand the differences between traditional HPC programming model on PFS and Big Data models (MapReduce)

Thoughts on example syllabus

- Drop workflow optimization? (too advanced)
- Drop/condense login/shell/basic Linux and assume prerequisite
- Good Citizenship may vary too much from site to site
- Maybe add Basic Performance Metrics

Group 3

Brainstormed skills

- Log in to remote cluster
- Remote file transfer, access
- Software module environments
- Run "Pleasantly"/"Delightfully"/"Embarrassingly" Parallel interactively

- Submit a batch job for ^
- Use git/version control for workflow management (batch files, directory structure)
- Make a change to ^ batch script and commit change
- How to compile (and run) a code

Thoughts on example syllabus

- Missing
    - Version controlled workflow(s)
    - Markdown (for docs)
- Other resources:
    - Oakridge national lab trainings etc. (TACC, XSEDE, git scm)
    - Communities to join: SIGHPC, SIGHPC-ED
- Vocabulary/jargon for/in HPC:
    - What is QoS?
- Interacting with your support community:
    - How (and who) to ask for help
    - How to find/read bug reports for software you are using
    - How to file a (helpful) bug ticket

Group 4

Brainstormed skills

Practical (technical)

- ssh and scp: data transfer
- Command line basics
- File system: creating, organising, navigating, file permissions
- Executing applications (either your own or centrally installed)
- Being able to run batch jobs (and find your output), checking status of jobs

Practical (non-technical)

- Where to get help when you get stuck (e.g. documentation)
- Etiquette on a multi-user system and implications for you
- Troubleshootling strategies

Conceptual:

- Basic HPC architecture and implications
- Differences from your laptop/workstation
- Why should I use HPC? What can it give me? When should should I use HPC?
- Parallelism and parallel processing
- Understand terminology and jargon

Thoughts on example syllabus

- Missing:
    - Conceptual content missing: why HPC? Basic HPC architecture
- What should be on day 2:
    - Intermediate shell topics
    - Testing jobs while running interactively
    - Using software modules

    - Running parallel tasks
    - Running parallel code
    - Optimizing workflows

Group 5

Brainstormed skills

- Understand how to access the system.
    - What is SSH? (or alternative)
- Understand the basic software environment.
    - Basic system environment.
    - Editing files.
    - Compilers
- Understand the scheduler
    - What is a scheduler?
    - What is a job?
    - How to submit a job
    - How to check the status of a job
    - How to kill a job
    - What is the wall clock
    - Why hasn't my job started yet? Why is my job queued?
    - Where's my output?
- Understand that simply running on a supercomputer will not speed up my code
    - Users should understand they will have to modify/parallelize their code to get an improvement