

MIT Open Access Articles

*Building Experience and Confidence in HPC Practitioners
through the Project-Based, Hands-On Practical HPC Course*

The MIT Faculty has made this article openly available. **Please share**
how this access benefits you. Your story matters.

Citation: Milechin, Lauren, Mullen, Julia and Hill, Chris. 2022. "Building Experience and Confidence in HPC Practitioners through the Project-Based, Hands-On Practical HPC Course."

As Published: <https://doi.org/10.1145/3491418.3535140>

Publisher: ACM|Practice and Experience in Advanced Research Computing

Persistent URL: <https://hdl.handle.net/1721.1/146282>

Version: Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

Terms of use: Creative Commons Attribution 4.0 International license



Building Experience and Confidence in HPC Practitioners through the Project-Based, Hands-On Practical HPC Course

Lauren Milechin
lauren.milechin@mit.edu
MIT
Cambridge, Massachusetts, USA

Julia Mullen*
MIT Lincoln Laboratory
Lexington, MA, USA
jsm@ll.mit.edu

Chris Hill
MIT EAPS
Cambridge, Massachusetts, USA

ABSTRACT

The MIT SuperCloud and Lincoln Laboratory Supercomputing Center have been introducing High Performance Computing (HPC) to a new audience through the "Practical High Performance Computing: Scaling Beyond your Laptop" class for the past four years. This informal class, open to the entire MIT community, introduces HPC, identifies canonical HPC workflows, and provides hands-on activities to explore the challenges encountered in the HPC environment. The students use their own research applications as project work to apply the class concepts to gain experience and confidence in using an HPC system and throughout the scaling process. Survey data collected before and after each class demonstrate that students feel they gain familiarity and experience in the concepts taught in the course and confidence in their own ability to apply those concepts.

CCS CONCEPTS

• **Social and professional topics** → **Informal education; Computational thinking; Adult education; Computational science and engineering education.**

KEYWORDS

HPC training and education, active learning, hands-on learning, flipped classroom

ACM Reference Format:

Lauren Milechin, Julia Mullen, and Chris Hill. 2022. Building Experience and Confidence in HPC Practitioners through the Project-Based, Hands-On Practical HPC Course. In *Practice and Experience in Advanced Research Computing (PEARC '22)*, July 10–14, 2022, Boston, MA, USA. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3491418.3535140>

*DISTRIBUTION STATEMENT A. DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited. This material is based upon work supported by the United States Air Force under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force. © 2022 Massachusetts Institute of Technology. Delivered to the U.S. Government with Unlimited Rights, as defined in DFARS Part 252.227-7013 or 7014 (Feb 2014). Notwithstanding any copyright notice, U.S. Government rights in this work are defined by DFARS 252.227-7013 or DFARS 252.227-7014 as detailed above. Use of this work other than as specifically authorized by the U.S. Government may violate any copyrights that exist in this work.



This work is licensed under a Creative Commons Attribution International 4.0 License.

PEARC '22, July 10–14, 2022, Boston, MA, USA
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9161-0/22/07.
<https://doi.org/10.1145/3491418.3535140>

1 INTRODUCTION

To extend the HPC community at MIT the authors host the course, Practical High Performance Computing: Scaling Beyond Your Laptop (PHPC), every year during MIT's Independent Activities Period (IAP), which occurs in the break between the fall and spring semesters. As part of IAP, the course is advertised and open to the entire MIT community: undergraduates, graduates, alumni, staff, and faculty. There are no prerequisites for the course and participants include new members of research groups that currently use the MIT SuperCloud system and potential users whose applications have grown too large for their desktops or those who have not yet begun their HPC journey.

Understanding that adult learners are motivated by solving a self-defined problem [2], the course is designed around hands-on learning that actively engages participants in scaling their research problem. Furthermore, to engage a group that includes developers, package users, and novices, the course uses a theory and practice model to scaffold the student understanding and develop intuition. Drawing from years of meeting with users to discuss applications and suggest scaling strategies, we uncovered a set of canonical workflows [3] which we use to teach users how to analyze their applications. Like many HPC centers we begin by teaching users how to time their code to determine where bottlenecks exist. Taking a step further, we introduce the students to other concerns that can impact performance, including the need for extra memory, excessive I/O, uncontrolled threading, or accelerator support. The goal is to provide the participants with the tools to scale their current application and understand how to analyze and scale future applications.

In the following sections we describe the approach used in the course, the modifications made based on feedback, our lessons learned, and the next steps. While our system may be unique in the support of interactive capabilities, the HPC community is expanding to include more users who match our student population and we expect our approaches and lessons learned will be valuable to the wider HPC community.

2 THE PRACTICAL HIGH PERFORMANCE COMPUTING COURSE

The PHPC course was designed to introduce HPC to a new audience and to develop greater understanding among existing HPC users. The course is offered as part of MIT's IAP session during January, which is traditionally a time for students to explore and experience new topics and skills outside of their major requirements. In the spirit of IAP, the only prerequisites are knowledge of a scripting language and a project in need of scaling. The learning objectives of the course are:

- Understand what HPC is and when to use it.
- Understand how a supercomputer differs from a personal computer:
 - Which components are the same? Different? New?
 - Understand how these differences affect application code.
- Know the canonical HPC workflows and understand how to map an application to a workflow and efficiently implement the workflow on a parallel and distributed system.
- Have experience running applications using the canonical HPC programming models.

Central to the course is the concept of the canonical HPC workflows. We roughly identify and cover three workflows: Throughput, Loosely Coupled (or MapReduce), and Fully Parallel (or Tightly Coupled). The workflow that fits an application dictates both what tools can be used to scale and how the application should be submitted as a job.

As with any audience new to HPC, we introduce its role in research and talk about what makes an HPC system unique. We then identify and describe in detail the canonical HPC workflows. As a group we work through thought experiments and case studies to help the students become confident in identifying these workflows. We play two collaborative games that demonstrate how parallel programs run and how a scheduler allocates resources [5]. Throughout we hold discussions on the student projects and help plan and debug parallel implementations. The students identify which workflow fits their application, analyze their serial code, and begin to work toward scaling up. Students are given MIT SuperCloud access for the hands-on sessions and project work.

The course has been offered four times in two different formats: twice fully in-person and twice in a flipped virtual format. The class has been led by two main instructors, with one or two additional instructors providing content or assisting with activities. These instructors comfortably supported 12 students in person and 26 virtually. Each year we have made slight changes to the schedule and expanded and improved content based on feedback, but the central goals and learning objectives of the course have not changed. Below we describe the course as it was taught in the first two and second two years.

2.1 Initial Course Design

The PHPC course began as a non-credit in-person course in January 2019. It was presented over two full days and included lecture interspersed with hands-on activities and discussion. The first day focused on introducing HPC, including a deep dive on HPC systems, HPC workflows, identifying independence and communication patterns, and measuring performance. Halfway through the morning we played the Puzzle Game [5], which demonstrates data distributions, shared and distributed memory applications, communication, and communication latency. The afternoon centered around case studies that included hands-on work using Jupyter Notebooks running on SuperCloud. These notebooks demonstrated simple implementations of Throughput, Loosely Coupled, and Parallel workflows. The second day began with a focus on submitting jobs through the scheduler. During this time, the students played The Scheduler Game to develop some intuition about how resources are allocated [5]. The afternoon was reserved for work

on student projects with group discussion. Students went to the board, sketched their research problem, described their application, identified a workflow, walked through their scaling plan, and asked questions. Based on feedback from year one, an afternoon session on MPI and an extra half-day for project work and questions was added in January 2020 (year two).

2.2 Current Course Design

In January 2021 we moved toward a flipped classroom approach, where lecture material was presented online and class time was reserved for hands-on activities, discussion, and demonstrations. Much of the video content had already been produced as part of a fully asynchronous online course, and the rest was completed to accommodate a fully virtual course [4]. The move to a virtual course afforded the option of shorter class meetings spread over a longer time frame. The combination of the scheduling differences, the flipped mode, and additional material resulted in more time for guided project work. The course remained roughly the same in 2021 and 2022 barring some minor schedule differences and the conversion of a live virtual lecture into an asynchronous module, and our discussion will focus on the 2022 version. The class was split into four days of content with the themes:

- Day 1: Introduction to HPC and HPC Workflows
- Day 2: Parallel Application Development Process
- Day 3: Implementing Throughput and Loosely Coupled Applications and Running Jobs
- Day 4: Developing Parallel Programs

In preparation each day the students were required to complete the corresponding asynchronous lecture content and laboratory exercises that were designed to help the students apply the lecture material to their projects. Class sessions were two hours and focused on interactive content. Each live session began with a Q&A for the asynchronous material and help troubleshooting issues from the homework laboratory exercises. To get the pulse of the learning and uncover misunderstandings or technical issues, Zoom polls were used to assess the students' ability to complete the exercises. A poll was also conducted to capture the amount of time required for them to complete the asynchronous material, the result was roughly 2-4 hours for each day.

The remaining agenda for each virtual in-person session roughly maps to the hands-on activities and discussions in the original physical in-person format of the course. The virtual environment lead to modifications that we feel improved the course. The Puzzle Game was replaced with the Processor Game, which had similar learning objectives and was more amenable to a virtual format [5]. Based on feedback, the range of case studies was expanded and spread out over two days to provide more time for students to identify and explore each canonical workflow and it's applicability to the research projects of different course participants.

In 2022, to help students analyze their applications, we added a live demo to show how to estimate memory requirements with Slurm's `sacct` command, explore their CPU and GPU utilization through `htop` and `nvidia-smi`, and time their code in a variety of scripting languages. On Day 3 we played a virtual version of the

Scheduler Game, and added a tutorial and live demo of LLMaPReduce, a language-agnostic tool for setting up and submitting Throughput and Loosely Coupled applications on SuperCloud [1]. To complete the detailed view of the workflows with distributed computing, we provided an MPI demo in Python on Day 4. The revised schedule allowed more flexibility and we took advantage of it by incorporating HPC research talks into the course. In 2022, we included a guest speaker who presented emerging work on creating greener HPC processing by evaluating the trade-off between the time to solution versus the consumption of power associated with GPU computing. This presentation was meant to give students insight into the HPC research domain beyond operational use of HPC systems.

The flipped format and longer class time allowed us to put more focus on student projects than in the past. In 2021, during the first week students created two slides that included an overview of their research question, provided initial serial timing for the application and identified their workflow and approach to scaling. For the second week, students presented their progress with the scaling approach as well as any bottlenecks or issues that they encountered. In 2022, the presentation was not as formal. On the first day small groups of students were assigned to break-out rooms with an instructor and each student briefly introduced themselves, their projects, and stated their goals for the class. Project work time was provided during an additional hour after the live session and students presented their scaling approach verbally during the final class meeting. There was also a for-credit option in 2021 and 2022. To earn credit, students were required to write a short paper on their project, in addition to participating in the class sessions, completing the asynchronous assignments, and presenting their work.

3 SURVEY RESULTS

A key concern with new users is developing not only an understanding of how HPC workflows are created or the skills required to submit jobs but to build confidence in their ability to effectively use HPC systems. It is standard practice for HPC researchers to work remotely from the center and support teams, and to succeed a researcher must be self-sufficient. Researchers with more confidence tend to be more self-sufficient in researching new techniques and trying out solutions.

In order to determine the efficacy of the course in developing confidence and achieving the learning objectives we have conducted surveys before and after each course. Since questions in the survey have not varied much from year to year we have combined the results in Figure 1.

Before each class we ask the students to rate their familiarity with HPC on a scale of 1-5 (5 being very familiar). The upper left plot in Figure 1 highlights our typical audience: the majority of students are unfamiliar with HPC, with a few somewhat familiar looking to learn more and ask questions.

In each survey we ask the students to rate their familiarity and experience with different workflows on the following scale: No knowledge, Have heard of it, Have tried to implement, Have implemented, and Use regularly. Most students either have no knowledge or have only heard of each workflow before the class starts. After the class the students have at least heard of each workflow,

and many have tried to implement or implemented each workflow. These results are showed in the plot in on the top right in Figure 1.

Perhaps the most important goal for the class is the ability to map an application to a workflow. The class is too short to go into detail on each workflow, but if a student is confident that they can map a given application, they know what direction to go in and the tools required to be successful. We ask students to rate their confidence in mapping an application to a workflow before and after the class. As shown in the bottom left plot of Figure 1, students enter the class with little confidence, and most leave with high confidence. More than half of the students who take the post-course survey rate their confidence at a 5, the highest level.

Scaling up also involves being able to run jobs on an HPC system, which requires some ability to use a scheduler. We therefore ask students to rate their confidence in batch processing and scheduler commands. The plot in the bottom right of Figure 1 shows the shift in confidence of the students. Around 40% of students rate their confidence at a 1 before the class, and more than half of students rate their confidence at 4 or 5 at the end.

4 LESSONS LEARNED AND NEXT STEPS

Between the first and fourth offerings of the course, many of the lessons that we learned were integrated into the course. For example, we modified the schedule because we found it is important to hold separate sessions for project questions and troubleshooting. We also found that designating the last hour of a class session for project work means that any class overtime leaves less time to work on projects. Having a separate session allows students to think about what they learned in class, apply it to their project, and come back with more questions.

Formal project presentations and paper templates helps focus students during the application analysis phase. Having a presentation before and after attempting to scale focuses on the importance of analyzing an application in its serial form, and helps catch misconceptions before scaling begins. Presentations and accompanying discussion helps other students in the class see the wide range of application types and strategies. Furthermore, since students are working on their own research projects, they are more invested in the outcome and develop greater confidence in continuing their research on the system. Finally, providing lecture material in an asynchronous format prior to class meant class time could be used more productively, with a focus on active learning, demonstration, and practice.

As with each iteration of the course, we will continue to improve and augment our existing content based on the questions we receive from our users throughout the year. Moving forward, we plan to adjust the schedule to fill out the full 4 weeks of the IAP term. Each week will have one class session and one project work session, which will include troubleshooting help and student project presentations. By spreading the course over four weeks, we give students the time needed to reflect on the material, experiment with the concepts and tools, and work on their course projects while balancing their own research demands. We also hope this format will be less time consuming for instructors.

Using the asynchronous content of this class, we have created a fully self-paced online version. The Self-Paced PHPC course has

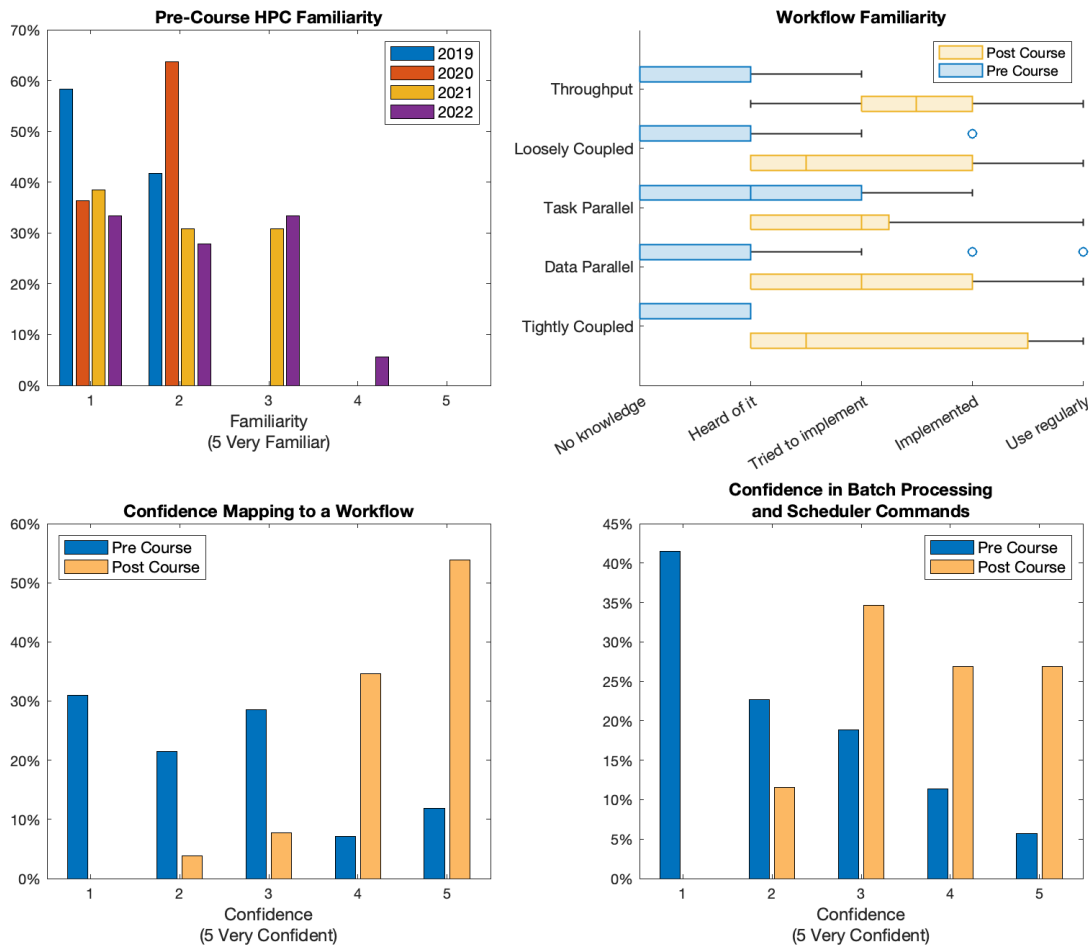


Figure 1: Survey results from the PHPC Course from 2019-2022. Top left shows the students’ familiarity with HPC going into the class for each year. Top right shows the students’ familiarity and experience with different workflows before and after the class. Bottom plots show the students’ confidence in mapping their application to a workflow (left) and in batch processing and scheduler commands (right). A total of 54 and 26 students answered the Pre-Course and Post-Course surveys, respectively.

been incorporated into the training for new MIT SuperCloud users and completion is now required to obtain an increase from a small startup resource allocation to the standard allocation. To address the hands-on portion of the PHPC course, the self-paced course walks students through the first steps of setting up their new SuperCloud accounts, analyzing their application, and testing potential scaling strategies. Overall, the self-paced course provides the knowledge and practice needed to scale efficiently on a larger allocation. As part of our next steps, we are exploring how to best evaluate the efficacy of this new onboarding process.

5 SUMMARY

We presented a project-based, hands-on approach to HPC education and training for researchers who use or will need to use HPC

resources. By providing lecture material in an online format prior to class meetings we were able to use class time for active learning centered around HPC concepts, skills, and student research projects. Survey results indicate that project work provides context for HPC knowledge and skills, resulting in increased confidence in using HPC systems. Based on the success of the course we are evaluating ways that we can scale the learning experience while retaining the project-based approach.

REFERENCES

- [1] C. Byun, J. Kepner, W. Arcand, D. Bestor, B. Bergeron, V. Gadepally, M. Hubbell, P. Michaleas, J. Mullen, A. Prout, A. Rosa, C. Yee, and A. Reuther. 2016. LLMapReduce: Multi-level map-reduce for high performance data analysis. In *2016 IEEE High Performance Extreme Computing Conference (HPEC)*. 1–8. <https://doi.org/10.1109/HPEC.2016.7761618>

- [2] M. Knowles. 1980. *The Modern Practice of Adult Education, From Pedagogy to Andragogy* (2nd. ed.). Cambridge Books.
- [3] Julia Mullen, Chansup Byun, Vijay Gadepally, Siddharth Samsi, Albert Reuther, and Jeremy Kepner. 2017. Learning by doing, High Performance Computing education in the MOOC era. *J. Parallel and Distrib. Comput.* 105 (2017), 105–115. <https://doi.org/10.1016/j.jpdc.2017.01.015> Keeping up with Technology: Teaching Parallel, Distributed and High-Performance Computing.
- [4] Julia Mullen, Weronika Filinger, Lauren Milechin, and David Henty. 2018. The impact of MOOC methodology on the scalability, accessibility and development of HPC education and training. In *Fifth SC Workshop on Best Practices for HPC Training and Education at SC18*.
- [5] Julia Mullen, Lauren Milechin, and Dennis Milechin. 2021. Teaching and learning HPC through serious games. *J. Parallel and Distrib. Comput.* 158 (2021), 115–125. <https://doi.org/10.1016/j.jpdc.2021.07.014>