Leveraging Teaching on Demand: Approaching HPC to Undergrads

Sandra Catalán[a,*], Rocío Carratalá-Sáez[b], Sergio Iserte[b]

[a] *Universidad Complutense de Madrid (UCM), Madrid, Spain*
[b] *Universitat Jaume I (UJI), Castelló, Spain*

Abstract

High Performance Computing (HPC) is a highly demanded discipline in companies and institutions. However, as students and also afterwards as professors, we observed a lack of HPC related content in the engineering degrees at our university, including Computer Science. Thus, we designed and offered the engineering students a non-mandatory course entitled "Build you own Raspberry Pi cluster employing Raspberry Pi" to provide the students with HPC skills. With this course, we covered the basics of supercomputing (hardware, networking, software tools, performance evaluation, cluster management, etc.). This was possible thanks to leveraging the flexibility and versatility of Raspberry Pi devices, and the students' motivation that arose from the hands-on experience. Moreover, the course included a "Teaching on demand" component to let the attendees choose a field to explore, based on their own interests. In this paper, we offer all the details to let anyone fully reproduce the course. Besides, we analyze and evaluate the methodology that let us fulfill our objectives: increase the students' HPC skills and knowledge in such a way that they feel capable of utilizing it in their mid-term professional career.

Keywords: Computational Cluster, Undergrad Teaching, System Administration, Parallel and Distributed Computing, Raspberry Pi

*Corresponding author.
Email addresses: `scatalan@ucm.es` (Sandra Catalán), `rcarrata@uji.es` (Rocío Carratalá-Sáez), `siserte@uji.es` (Sergio Iserte)

1. INTRODUCTION

The mathematician and computer scientist Alan Turing said “We can only see a short distance ahead, but we can see plenty there that needs to be done”. High-Performance Computing (onward, HPC) is nowadays vital for almost every researcher and company, covering from data analysis to computer simulations and artificial intelligence studies. By analyzing *the short distance ahead* of us, as graduates from Universitat Jaume I (UJI), we observed that there exists an indisputable gap between the syllabus of the degrees and the HPC needs of the professional environment. We realized that there is a lack of HPC contents in Computer Science studies, and also in other engineering degrees. For this reason we decided to offer a non-mandatory introductory course entitled “Build your own supercomputer with Raspberry Pi”[1]. It serves as an introduction to HPC, starting from the basics and using Raspberry Pi devices to build a dummy supercomputer.

In this paper, we present our experience with the two editions of the course we have offered in 2018 [1] and 2019, respectively. It is common to find courses that focus on a specific HPC area, such as parallel programming, machine learning, cluster design from the perspective of selecting specific hardware or recycling it, etc. However, from our understanding, a complementary course of introduction to clusters of computers should show all the possibilities that a supercomputer may provide and let the students decide which aspect appeals more to them. For this purpose, in the course, we opted for covering a wider scope of HPC-related fields, providing the attendees with general basics.

Three main reasons motivated us to create and design the course: 1) HPC society interest and need is increasing; 2) There exists a lack of HPC related content among the Computer Science syllabus at UJI; and 3) HPC self-learning is complicated. The obvious contradiction that 1) and 2) expose, together with

[1]Course website (in Spanish): `https://sites.google.com/uji.es/supercomputadorraspberrypi`

the difficulties associated to autonomously start in this field, evidenced that such a course could be useful for our students. Particularly at UJI, the engineering degrees are studied in the School of Technology and Experimental Sciences (ESTCE), which offers 11 degrees, including Computer Science, and 15 masters. The proposed course was offered to all the students in ESTCE since they are expected to have a technical background. Basic Linux command line knowledge was highly recommended, although it was not a strict prerequisite.

The main goal of the course is to provide the students with HPC knowledge. This includes not only understanding the basics but also being capable of identifying when supercomputers are needed and how they are built, as well as where HPC is applied. To soften the entrance barrier to this field, the course avoids classical teaching methodologies to let students experiment with actual hardware while enjoying the learning process. It is also part of the course's purpose to show the importance of monitoring and data analysis, as well as showcasing the interaction with a real supercomputer. Besides, a part of the course is left open to adapt its content to the particular interests of the students. To this end, a ”teaching on-demand” methodology is leveraged, allowing the attendees to choose in what they prefer to invest this specific time slot.

The main contributions of this work are:

- Description, test and evaluation of the “teaching on demand” approach.
- Implementation of a hands-on experience based on the usage of Raspberry Pi devices to motivate the students.
- Increase in the HPC knowledge and interest among engineering students.
- Detailed description of the course in such a way that the community can reproduce it.
- Analysis of the mid-term impact of the course on the attendees academic/professional development.

The rest of the paper is structured as follows: in Section 2 we give some HPC background, focused on describing the terminology, tools, applications, and hardware that are employed in the course; in Section 3 we describe different works that reflect the effort from the HPC community to provide students with skills in the field; in Section 4 we explain the course methodology, providing details about its motivation, goals, curriculum, design, and the students recruitment and selection criteria, as well as describing its structure and the evaluation of the curriculum; in Section 5 we discuss the results extracted from the evaluation of the course curriculum and the lessons learnt that derive; in Section 6 we present the performed follow-up to evaluate the mid-term impact of the course on the attendees; in Section 7 we provide the conclusions extracted from this work; and in Section 8 we explain the open lines that can be explored as future work.

2. BACKGROUND

In this section, some background to facilitate the understanding of the paper is provided following. More detailed descriptions of the employed terms can be found in [2, 3, 4, 5].

2.1. Terminology

A *cluster* is understood in the context of the course as a set of independent computers (namely *nodes*) which can work collectively. Related to cluster, we present the definition of *supercomputer*: a system (usually a cluster) that delivers high computational power. Besides, HPC is presented as the field in charge of solving scientific and/or engineering problems, which are typically complex and costly in terms of time and resources. To achieve high performance and, thus, solve large problems, it is necessary to leverage the coordinated and simultaneous work of different devices. This way, solutions can be computed in a reasonable time, much faster than in personal computers. Precisely, the joint execution performed by several processes concurrently is named *parallel processing*, and in case those processes do not share the memory, then it is said to be *distributed processing*.

The term *performance* refers to the computational power provided by the cluster. Likewise, *scalability* is presented as the property that evaluates the ratio between two measurements of the time, each of them associated with different hardware/software configurations.

2.2. Tools and applications

The basic setup of the cluster includes the network configuration among the nodes, to enable the remote access through *Secure SHell (SSH)*; and installing the *Network File System (NFS)*, which provides the cluster with a shared file system. Moreover, parallel and distributed programming will rely on *OpenMP* and *MPI* to respectively coordinate the execution of several processes within each node, and through different nodes in the system.

2.3. Hardware

As mentioned before, to let the students build their cluster, they are provided with *Raspberry Pi* devices. These can be seen as small computers composed of a motherboard that integrates the processor (with ARM architecture), RAM, several ports for input/output signals, and a slot for a memory card. The cheap price, together with its versatility, made us opt for this device.

Of course, production HPC clusters are not based on Raspberry Pi, but more powerful components, although there exist actual low-cost energy-aware clusters based on these devices [6, 7, 8]. However, in rough outlines, there exists straightforward parallelism between what students can observe when building the cluster employing these simpler devices and what forms actual HPC supercomputers that allows learning the HPC basics [9, 10]. The similarities and differences between production HPC clusters and Raspberry Pi based clusters are explained to students along the course.

3. RELATED WORK

In the recent past, we can find many efforts from the HPC community to convey the supercomputing philosophy to CS and other engineering students.

Within these efforts, we can find platforms to simulate HPC environments and help students to understand the different needs and infrastructures [11], and extra courses to provide them with essential HPC knowledge that is not officially included in their curricula [12]. Moreover, there are also remarkable efforts such as [13] that aim to establish what a CS student should know about HPC, setting some "core topics".

Usually, courses involving supercomputing topics focus on parallel programming languages or paradigms [14, 15]. Others [16], explain the process of designing a cluster for educational purposes. Commonly, the resulting cluster is based on Beowulf distributed computing system [17], useful for recycling deprecated machines in the center. In [18], the authors combine the hardware and software experience involving students in the cluster set-up, employing Odroid instead of Raspberry Pi. However, that course follows a strict program addressing the implementation of a fixed suggested problem. A thorough review of experiences using micro-clusters for educational purposes can be read in [19]. In that paper, the authors compile a series of low-cost clusters using different hardware such as Parallela, Odroid, NVidia Jetson, and Raspberry Pi. Furthermore, they enumerate several strategies for engaging students.

Apart from scientific publications, several related project descriptions and experiences are available online, where different authors set up and configure a Raspberry Pi cluster for performing distributed tasks. An effort in building a Raspberry Pi cluster is reported in [20], where High-Performance Linpack (HPL) is executed over 32 nodes. In [21], the authors implement a bingo where each node corresponds to a cell in a bingo card, and all the nodes collaborate to check if the new number completes a match. Projects like [22] provide an image for a Raspberry Pi 3 system with pre-installed support for clusters and other attractive parallel suites. A more recent project [23] leverages a new generation of Raspberries, the Raspberry Pi model 4 B [24] to build a cluster and execute small distributed Python codes.

From our understanding, a complementary course of introduction to clusters of computers should show all the possibilities that a supercomputer may provide and let the students decide which aspects appeal more to them. For this purpose, we have included a wide variety of topics in the course, providing simplified versions of the ones classically addressed in professional and more complex HPC related manuals [25, 26, 27], such as software performance, cluster scalability, networking, CPU frequency, cooling, benchmarking, and scientific applications.

The main difference concerning all the other approaches we have analyzed is the fact that we do not focus on any specific aspect (such as parallelism, hardware description, performance evaluation, users management, etc.), but we try to cover the whole scope of subsystems and functionalities while building and utilizing an HPC system, to provide a general HPC overlook.

4. METHODOLOGY

In this section, we describe the proposed course. First, we present the course goals, a high-level curriculum description, the course design, and the recruitment and selection criteria; then, details regarding the contents of the course are provided and discussed; lastly, we include an evaluation of the syllabus.

4.1. Overview

Broadly speaking, the course is a 10-hour workshop highly focused on a hands-on approach to bring HPC to students. The main idea is that students, organized in groups, build and configure their cluster. That prototype is used to run HPC applications, to face first-hand issues of supercomputer management, and to experience how to test and characterize a system.

4.1.1. Course motivation

The main reason that inspired us to create this course is to bring HPC to undergraduate students. Although it is a cross-cutting subject that is present in a wide variety of fields (for instance physics, economics, or medicine), HPC is far away to be well known by society and, especially, CS and engineering students. In this scenario, three reasons drove us to the creation of the course:

HPC society interest and need are increasing.

The amount of CS professionals that require HPC skills is increasing due to the importance of this area in many social and professional fields. In fact, European and American governments and institutions have increased the funding of HPC related projects [28, 29, 30].

On the other hand, the effort performed by institutions to bring HPC and supercomputers to society, in general, are more common every day. For instance, documents [31] and [32] are good examples of research centers opening their facilities to the public, to increase the awareness of HPC, supercomputers, and science.

There exists a lack of HPC-related content among the Computer Science syllabus.

There are several previous works in which this fact is also highlighted and addressed [33, 34]. The CS degree of UJI is four years long and includes, basic training (BT), compulsory (C), and optional (OP) subjects, as well as a final degree project (FDP), but the presence of HPC is marginal in the curriculum. The set of subjects included in UJI's Bachelor in CS is described in Table 1. It can be observed that only 11 out of 62 subjects include HPC-related content. Moreover, three of them are optional, which means that part of the students will conclude their studies having dealt with HPC in less than 13% of the subjects. Besides, six of the subjects studied during the fourth year correspond to the specialty chosen by the student, and the only two that include HPC content belong to a single specialty.

HPC self-learning is complicated.

Subjects such as "Computer Structure" and "Computer Architecture", in which the basis of HPC are established, typically present the lowest marks among the students, and their failure rates are high. Besides, even though there exist platforms to supplement conventional lectures [35], or detailed and well-described manuals [36], they often turn out to be complex to understand

[2] It could happen that an HPC related FDP is developed. However, this is not the most common scenario.

Year	Type	#Subjects	#HPC Subjects
1	C	9	1
	BT	1	1
2	C	3	0
	BT	7	3
3	C	9	1
	OP	7	3
4	C	24	2
	OP	1	0
	FDP[2]	1	0

Table 1: Subjects of the UJI's Bachelor in Computer Science syllabus summarized by year, specifying their type, the total number of subjects which are studied that year, and the number of subjects that include HPC related content (according to those offered in years 2018/2019 and 2019/2020).

from the students perspective. Furthermore, a holistic view of HPC requires knowledge from different fields that are unlikely taught along with the CS/OE syllabus, such as configuring applications setup, or job management. All in all, we consider that self-learning HPC can be a challenging endeavor.

4.1.2. Course goals

When designing the course, we chose its content to guide the students to reach the following main learning objectives:

- *Objective 1: Gain general HPC knowledge.* We consider that it is crucial to know not only what HPC and supercomputers are, but also what are cutting-edge trends in this field.
- *Objective 2: HPC not only in supercomputers.* Change their mind about thinking of HPC restricted to huge supercomputers that belong to powerful companies. HPC can be carried out on small infrastructures, such as a personal computer with dedicated hardware and specific software.

- *Objective 3: A better grasp of supercomputers.* Be able to understand the needs (both in terms of hardware and software) of a supercomputer employed in HPC.
- *Objective 4: When to use HPC.* Recognize current applications where HPC is necessary.
- *Objective 5: Enjoy the learning process.* To avoid traditional lessons pressure feelings, flexible methodologies are employed.
- *Objective 6: Relevance of monitoring.* Emphasize the importance of monitoring metrics. For instance, temperature and its impact on the cooling requirements.
- *Objective 7: Analyze performance.* Learn how to conduct and understand performance analysis and scalability.
- *Objective 8: Real-world HPC experience.* Comprehend how an actual HPC cluster/system is built and how users interact with it.

4.1.3. High-level curriculum overview and its design

The aim of the course was not only to bring HPC to students but also to do it in a practical way. The idea was trying to deviate from traditional lectures to implement a hands-on approach while fulfilling the personal interests of the students. Combining all these goals led us to design the course divided into four stages:

- Theoretical introduction.
- Cluster assembling, configuration, and test.
- Learning on demand.
- Showcasing a supercomputer.

Figure 1 illustrates the four stages. Firstly, we provide a theoretical introduction supported by a set of slides. After that, the students are explained how to assemble and configure their cluster. Once the basic configuration and tests are performed, which closes the first part of the course, each group of students can choose on which topic (or topics) wants to focus. Each group works on its chosen topic and, at the end of this stage, results are shared and discussed among the students from all groups. The last part of the course consists of the showcase of an actual supercomputer to give the students a realistic experience as supercomputer users.

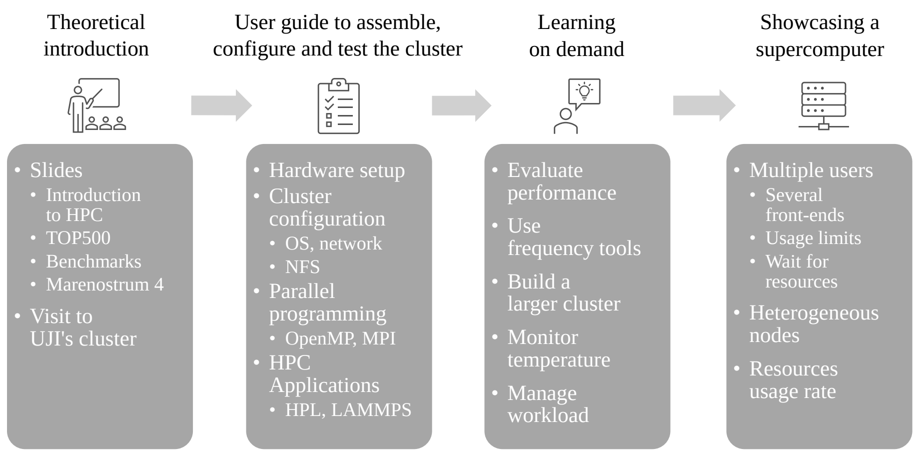

Figure 1: Course curriculum overview.

4.1.4. *Design of the course*

A crucial aspect when creating a course is the number of available resources in terms of material, budget, and people. In this course, we propose using a low-cost cluster to guarantee a hands-on experience to the attendees, while keeping a reasonable budget. The required hardware to assemble each of the clusters is described in Table 2. Hardware peripheral devices such as screens, keyboards, and mice (and the associated wires) were already available in the classroom. This is why they are not included in the cluster budget, although they were employed.

Component	Quantity	Unitary price
Raspberry Pi 3 Model B+	4	29.47€
USB Hub (4 ports)	1	11.99€
USB 2.0 wires	4	0.94€
Micro SD Class 10 (16GB)	4	7.77€
Switch Ethernet (5 ports)	1	16.50€
Ethernet wires	4	1.14€
Total price		185.77€

Table 2: Description of the components required for each cluster detailing their prices and quantities (based on year 2018 prices).

In our case, the presented hardware was acquired by the HPC&A[3] research group from UJI. Given the available budget and the total cost of each cluster, we could offer 20 vacancies in the course in its first edition, and 24 in the second. The attendees were grouped in a maximum of four-people teams to guarantee the participation of all the members as much as possible.

On the other hand, the amount of human resources is essential in this course, especially to ensure an enriching experience in the on-demand learning part. According to our experience, we consider that one person can be in charge of doing the theoretical introduction. However, the hands-on part requires at least two people in order to help all the groups, given that the duration of the course is restricted to 10 hours. Nevertheless, although the on-demand learning part can be managed by two instructors, we consider that three people are the most recommendable number, since progression pace may vary among the different groups. One lecturer takes care of the groups that need more time to complete the user's guide configuration. (Note that is important to take into account that some students are not from CS and also that there are "early-stage" CS students that do not count with the background related to building and configuring the cluster.) Meanwhile, another lecturer coordinates

[3]http://www.hpca.uji.es/

all the groups that are interested in creating a larger cluster.The third instructor guides the groups interested in the performance-energy consumption and helps them using specific tools.This distribution of the workload among the lecturers is based on our personal experience of the course and can be of course adjusted to the different needs, agendas, budgets, and course vacancies in other contexts.

Regarding the agenda, it must be noted that, for the first edition, we proposed the course as a 2-day workshop (of five hours duration each day), while for the second edition we decided to offer a single day experience that occupied a Saturday morning and afternoon[4]. The reason to change the format of the course was the students' availability. The second edition was conducted closer to the exams period and, from our experience, we considered it easier for students to find a time slot for one (longer) session than for two (shorter) sessions divided into two different days.

4.1.5. Recruitment and selection criteria

Participants enrolled voluntarily in the course "Build your own supercomputer with Raspberry Pi". The moderate number of vacancies in the course was constrained by the budget and available facilities. However, the interest in the course was reasonable. In the first edition, 26 students applied for the course in total, 20 were selected as participants, and 18 attended the course. In the second edition, the number of applicants raised to 48, 24 were selected to participate, and 20 attended the course.

HPC inherently belongs to the CS field, however, its interdisciplinary nature and the general approach considered in the course made us offer it to all the engineering students. To select attendees, we followed a simple idea: as we believe the lack of HPC and supercomputing knowledge should be corrected as soon as possible, we prioritized the students who were in an early stage of their studies (first and the second year). Since the course begins on the basics, only basic Unix and Shell knowledge prerequisites were recommended. Once early-

[4]Remark: at UJI, there are no lectures on Saturdays.

stage students were selected, if there were still vacancies, they were assigned to third and fourth-year students.

To analyze the course's findings and outcomes, we divide the attendees into CS and *other engineering* (OE) students. The distribution of students, according to their studies, is shown in Table 3. Note that, to maintain the proportion in terms of applicants' interest and background, we selected a similar amount of participants from CS and OE, always following first-come, first-served criteria regarding their application.

	First edition		Second edition	
	CS	OE	CS	OE
Applicants	15 (58%)	11 (42%)	24 (50%)	24 (50%)
Selected	12 (60%)	8 (40%)	13 (54%)	11 (46%)
Attendees	12 (66%)	6 (33%)	10 (50%)	10 (50%)

Table 3: Participants classification per field (Computer Science - CS, Other Engineering - OE), degree of participation (Applicants, Selected applicants, Attendees), and course edition (First, Second).

A well-known and widely studied concern in engineering and CS students is gender unbalance. UJI is not an exception and, unfortunately, there is a small percentage of female students. While in the first edition of the course, we had a female attendee (out of the two applicant women), in the second edition, the number of interested women in the course raised to five. Three of them were selected to participate, but none attended it.

4.2. Course Curriculum

As it has already been stated, the main goal of the presented course is to introduce HPC to CS and OE students at UJI to complement the syllabus of their degrees.

For this purpose, a specialized curriculum has been tailored to provide the expertise needed to fill the gap of knowledge between HPC and other related disciplines. In this sub-section, the curriculum of the course is detailed, and linked to the course objectives introduced in Section 4.1.2.

4.2.1. Step 1: What's HPC?

The course starts with a brief description of the concept *HPC*. Supporting the explanation with schemes and videos, in this part, the following questions are addressed:

- *Why is supercomputing necessary?* A series of well-known examples such as weather prediction computations, social network analysis to personalize adverts, voice recognition in smartphones, or traffic information in Google Maps application, are raised to make the students think about the need for extremely fast operations processing. Besides, a wide discussion about treating vast amounts of data or generating and manipulating real-time results is promoted.

- *How is supercomputing different from computation?* Following, the most relevant terms related to HPC are defined by giving general ideas and fostering students' discussions. It is worth noticing that in-depth explanations are out of the course's scope. In order to ease the understanding of the terms and technologies, as long as it is possible, they are compared to more familiar concepts such as the ones associated with the components in a personal computer.

- *Where does supercomputing take place?* The introduction concludes with the presentation of the TOP500 list [37], and the HPL (a Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers) benchmark [38, 39]. In more detail, we leverage this moment to expose the system's features occupying the first position in the list and compare it to the Spanish system *Marenostrum 4* [40], which also appears in the TOP500. Finally, intending to give a more down–to–earth approach, the course includes a visit to the university computing data center, so the students can see an actual HPC system.

This first part of the course aims to convey the big picture of HPC (*Objective 1: Gain general HPC knowledge*), prove that HPC can be implemented at a small-scale (*Objective 2: HPC not only in supercomputers*), and illustrate which are the target applications of HPC (*Objetive 4: Where to apply HPC*). Thus, a global vision of HPC is presented to the students in the context of nowadays facilities and systems.

4.2.2. Step 2: Cluster set-up

In the second part of the course, the students are grouped in teams of 3-4 people, and the groups are given the hardware components of the cluster, and the student's guide (see Appendix A). This document describes the necessary steps to end up with a fully functional cluster with support for distributed computation. In this regard, the cluster set-up is divided into these three parts:

- *Cluster assembly.* Each group is in charge of assembling its own cluster composed of four Raspberry Pi 3 Model B+ devices (Figure 2 reflects a sample of an already assembled cluster). Besides, a switch, the corresponding Ethernet wires, four SD cards, and the power supplies are provided.

Figure 2: A Raspberry Pi cluster assembled during the course. Note that the 3D-printed structure that holds the Raspberry Pi devices is only used for aesthetics purposes.

Although the guide has already been handed out, the instructors can provide tips and short explanations of the hardware components, especially describing the Raspberry Pi features. Because of the different students' backgrounds or knowledge, with these intermissions, balanced progress among groups is pursued, preventing frustration to the less skilled groups, or boredom to the most advanced ones.

- *Cluster configuration.* The basic software configuration of a cluster comprises the installation of the operating system[5], the network configuration, and the file system set-up. The operating system is pre-installed in bootable SD cards handed out to the attendees to save time and not lose the perspective of the course. Thus, the students can focus on the most relevant steps of the configuration. Once all the components are assembled, the students configure the network and the shared file system. The network configuration is essential to make all the devices work collaboratively. On the other hand, a shared file system (Network File System, NFS [42]) is enabled through the cluster to facilitate access to the files within the cluster.

  With more detail, students are driven to complete the next steps:

  - Configure the DHCP service to accept a dynamic IP in the wireless network interface of one single node, which will act as the main node or front-end. Furthermore, the WiFi interface is set as the front-end default gateway. The Ethernet interfaces of all the nodes are configured with a static IP.
  - Configure SSH by creating public-private pair of keys in the front-end node and then sending the SSH key to the other nodes.
  - Install and configure an NFS server and create a directory in the front-end shared with all the nodes. It is highly recommended that, at this point, the students check that each node can reach the other

[5] We opted for using Raspbian OS [41].

nodes, as well as use the NFS properly. For instance, this can be verified by creating a text file located in the shared directory and then editing it from different nodes.

Because part of the listed items takes some time to be downloaded, configured, and/or installed, that "idle time" is used to explain the corresponding usage and features to the students.

- *Performance evaluation.* To provide a realistic HPC experience, students are introduced to some of the most common software used in a supercomputer. For this purpose, students will learn how to compile, install and configure:

  - HPL [39], a performance benchmark leveraged to rank the supercomputers in the TOP500 list.
  - LAMMPS [43], a classical application for molecular dynamics modeling.

  Both applications require the use of the OpenMP programming model [44] and Message Passing Interface (MPI) [45] to exploit all the processing elements in the cluster. OpenMP is natively integrated into the GCC package and, consequently, no extra installations are required. However, MPI needs to be installed. Concretely, we opted for using MPICH [46], since it is one of the most popular open-source libraries employed in nowadays clusters, along with OpenMPI [47]. As we presented both MPICH and OpenMPI to the students, some thought of installing and comparing both implementations, as we will explain in Section 4.2.3.

  The activities proposed for this section are:

  - Configure an execution of HPL to achieve the maximum performance of the cluster. (Note that all nodes should be leveraged.)
  - Perform a scalability analysis of LAMMPS using different configurations of threads and processes.

– After discussing with the students the importance of also leveraging threads and not only available nodes, we suggest checking the official website, where the way to configure LAMMPS with OpenMP is explained.

Objective 3: A better grasp of supercomputers and *Objective 4: When to use HPC* are mainly covered during this second part of the course, because the architecture of a supercomputer and how software makes use of its components are explained and related. Moreover, by means of running benchmarks and helping the students to understand the provided performance results, *Objective 7: Analyze performance* is addressed.

4.2.3. Step 3: Learning on-demand

Once the cluster is fully functional, the course is planned to have a free hands-on section, where the students can put into practice the acquired knowledge and/or do research in a specific aspect of their interest, with the help of the instructors.

Although the students can explore any field, they are provided the following on-demand ideas as guidance:

- **Performance and frequency tools**. This activity aims to widen the knowledge about the relationship between performance and energy consumption, the impact of power consumption on HPC facilities, and which tools or strategies can be applied to control these facts. In this regard, *cpufreq* utilities [48] are introduced and tested on their clusters. In this way, students can experience the effect of frequency changes on the performance of executions.

  Furthermore, the activity can be complemented with studies of power consumption and the associated economic costs on current supercomputers to illustrate the relevance of the problem. For instance, instructors can provide more information about how performance affects power consumption; dynamic and static power, and their interaction.

- **Creating a larger cluster**. Different teams can decide to merge their clusters to scale their resources up to 8, 12, or more nodes.

  The proposed activity requires to reconfigure both the network and the distributed file system in order to ensure the appropriate functioning of the new cluster. With this first step, the knowledge acquired by following the user's guide was reinforced thanks to reproducing that in a new scenario. Once the students check that the new cluster is working correctly, they have the opportunity to try different configurations for LAMMPS and analyze which of them delivers the best performance.

- **Real-time measuring of the device temperature**. Thanks to an incorporated sensor on the board, the temperature can be monitored using the command `vcgencmd measure_temp`.

  This activity requires developing a script that periodically checks the temperature. The data acquired is saved for further analysis.

  For instance, Figure 3 shows an actual output of this activity provided by one of the groups from the second edition of the course. Concretely, the plot reflects the temperature variations along the seconds required to complete LAMMPS execution setting two different frequencies: 1.40 GHz and 600 MHz. Through this analysis, the students in the group also checked the differences in terms of total execution time when varying the frequency (355 seconds with the higher frequency, in contrast to 661 seconds with the lower one).

- **Workload management in an HPC system**. Many users compete for the resources available at the supercomputers. When introducing this notion, students are taught that, with workload management software, resources and users' requests are orchestrated. For this purpose, Slurm Workload Manager [49] is presented. In this activity, following Slurm's administrator guide and helped by the instructors, students face the installation and configuration of Slurm.

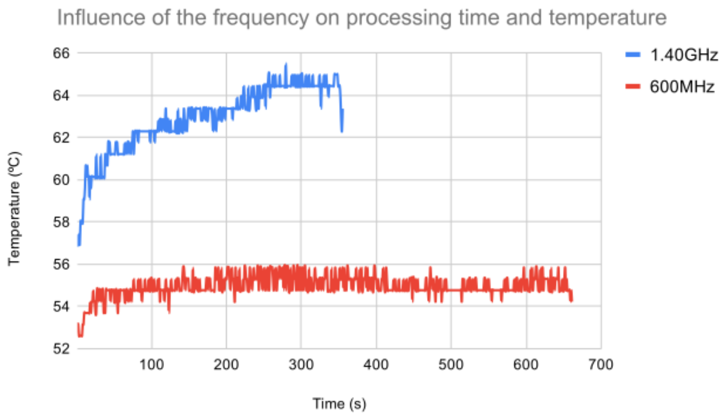

Figure 3: Temperature during a LAMMPS execution with different frequencies.

Before ending this part, each group is encouraged to briefly share with the others their conclusions on the topic explored and the work performed.

Objective 5: Enjoy the learning process is attained by promoting this free hands-on approach. The fact that the students propose the work they want to develop during this part of the course lets them feel relaxed (note that the students are not given a final mark at the end of the course) and free to experience whatever they feel curious about. Also, *Objective 6: Relevance of monitoring* and *Objective 7: Analyze performance* are targeted by showcasing the importance of monitoring tools and how to understand them.

However, some activities may take more time than expected and the students could not finish them on time. Even in this case, students are encouraged to learn the valuable lesson of *easier said than done*. For instance, in the second edition of the course, after installing and configuring Slurm, a team found the following error:

```
slurmctld: error: High latency for 1000 calls to
gettimeofday(): 1824 microseconds.
```

The solution seemed straightforward for the course purposes: recompile the source code increasing the latency tolerance. This is translated into an amount of time that is not available in the proposed course. A *failed* activity can be considered a very interesting example to reinforce the experience of the existent difficulties that HPC system administrators, or users, need to solve, and reinforces *Objective 8: Real-world HPC experience.*

4.2.4. Step 4: How to use a supercomputer?

The course wraps up showcasing a real user experience when logging into a top-class supercomputer. This activity demonstrates that large-scale systems have more complexity than the cluster configured in the course. However, the students realize that the cluster philosophy is the same they have learned in the course. It is crucial that, before ending the course, students apprehend the big-picture of a production supercomputer, since they are likely to be the next generation of users, developers, and/or administrators. In this regard, the lecturer in charge logs in a production supercomputer via SSH and explains that:

- Unlike the cluster set up in this course, which only has one user and a front-end that, in turn, is a compute node, it is usual to have several dedicated front-ends to provide fault tolerance while supporting hundreds of users interacting with the system simultaneously. For this purpose, the lecturer accesses some of the front-ends and counts the logged users at that moment with the command `w | wc -l`.
- All these users share the supercomputer, but they cannot use the hardware in their own free will. We introduce here that there is a management software responsible for temporarily assigning resources to users. Furthermore, it is important to note that users have quotas of usage, which limit the computation time or disk space that a user can utilize.
- Resources are not usually idle waiting for us. In production supercomputers, there is a waiting time before assigning resources to the users' jobs. This time varies depending on how saturated the system is, in other words, the number of pending jobs in the queue which are not being executed.

In general terms, jobs requesting more resources for more time experience higher pending times.

- The nodes of the supercomputer do not need to be identical. Supercomputers can be heterogeneous to meet more necessities. In this regard, there exist different partitions of hardware for different types of users. Another important concept to bear in mind is that there usually exists a partition with shared machines that are likely to be assigned earlier to the jobs.
- The employment of the hardware tends to 100%. For this purpose, we show utilization rate statistics through time. These statistics reflect not only the constant high utilization but also total and partial outages that may indicate maintenance or unexpected failures.

All in all, with this holistic instruction, we are confident that students understand that what they have done is the cornerstone of an HPC system. Hopefully, after leaving the classroom, they will be eager for a second part of the course to learn more about jobs, queues, shared resources, or distributed computation.

A more realistic experience has been given covering *Objective 8: Real-world HPC experience* in this part. This is one of the significant improvements of the second edition in contrast to the first one, made thanks to the change of format between editions, allowing us to include more material that enriches the experience of the attendees by setting them in front of a real cluster in production.

4.3. Evaluation of the curriculum

The new teaching approach presented in this paper to let students become closer to HPC implied risks and challenges. We wanted to be able to judge the impact of this "custom-tailored" teaching methodology on students. For this purpose, we collected qualitative information at the beginning and the end of the course through two anonymous surveys. In this section, the questions of the surveys are presented, and those related to the impact of the course on the attendees are analyzed.

4.3.1. Initial and final surveys

The initial and final surveys targeted the potential changes in the attendees' knowledge about HPC. Note that, in the second edition of the course, new questions were included to better justify some of the observations after the first edition. Those questions are marked with the letter "N".

We have categorized all students' survey answers to analyze the impact of the course in the short-term (the categories are presented in Figures 4-10). The questions were designed to find out how students felt and what they knew about HPC before the course, and what was the progress in terms of HPC knowledge after it.

Regarding the surveys, the initial one (IS) consisted of the following questions:

- ISQ1: Why have you signed up for this course?
- ISQ2: Do you think that HPC has an influence on your day to day?
- ISQ3: How would you define HPC?
- ISQ4: What do you think about supercomputers?
- ISQ5 (N): In which field do you see yourself developing your professional career when you finish your studies? (Multi-answers were accepted in this question.)
- ISQ6 (N): Do you believe that HPC could be applied in your desired professional career field?

The final survey (FS) consisted of the following questions:

- FSQ1: Do you feel more/same/less interested in HPC now?
- FSQ2: Do you think that HPC influences your day-to-day?

- FSQ3: How would you define HPC?
- FSQ4: What do you think about supercomputers?
- FSQ5 (N): In which field do you see yourself developing your professional career when you finish your studies?
- FSQ6 (N): Do you believe that HPC could be applied in your desired professional career field?
- FSQ7 (N): Evaluate (from “0 - Very Bad” to “10 - Excellent”) the quality of the theoretical explanations previous to the practical activities.
- FSQ8 (N): Evaluate (from “0 - Very Bad” to “10 - Excellent”) the empathy degree and quality of the help provided during the practical activities resolution.

4.3.2. Results of the surveys

The already categorized answers collected from the students through the surveys are reflected in Figures 5-9. Each plot shows the Initial (IS) and Final Survey (FS) answers to a given question (specified in the title). Data is shown in full colored bars (for IS) and line-filled bars (for FS), including Computer Science (CS) students in blue and Other Engineering (OE) students in red. Differentiation between the first and the second edition answers is made using, respectively, darker and lighter color-schemes.

Next, we analyze those survey questions that refer to the curriculum evaluation (Questions 1 to 4, and Question 6). In general, results show that students increase their awareness about HPC in daily life. Moreover, they show good comprehension of usual terms in the HPC field. The main conclusions extracted from the curriculum evaluation are now highlighted. The remaining questions (regarding the course evaluation) are addressed in Section 5.

There exists a lack of HPC knowledge among Engineering students.

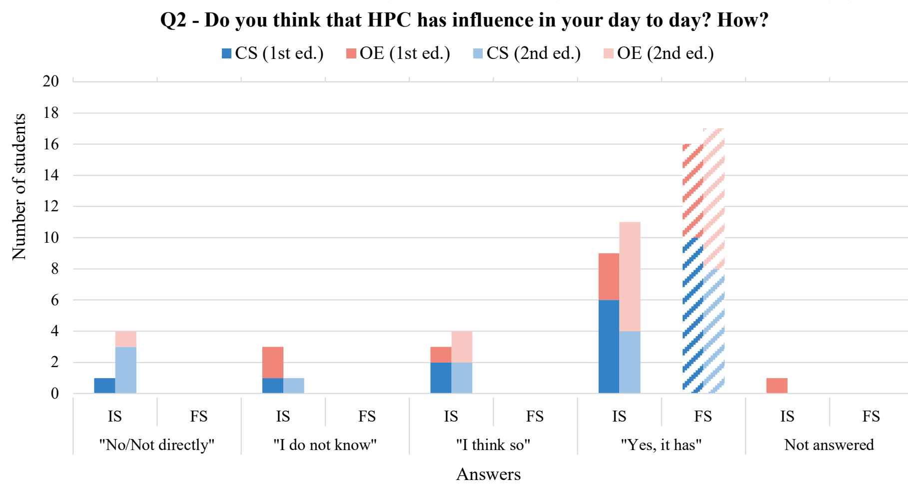

Figure 4: Answers of ISQ2 and FSQ2.

From our teaching experience through the past years, we observed a lack of HPC knowledge among the students. This was an opinion before starting the course, but now we have evidence that exposes that it was true, taking into account what can be observed in Figure 4. According to this plot, before conducting the course, only around half of the students considered that HPC has an obvious influence daily. It is remarkable that in both editions, around 20% of the students in total did not know if HPC has that impact (particularly in the first edition, 3 out of 17 attendees), or even negated it (especially in the second edition, 4 out of 20).

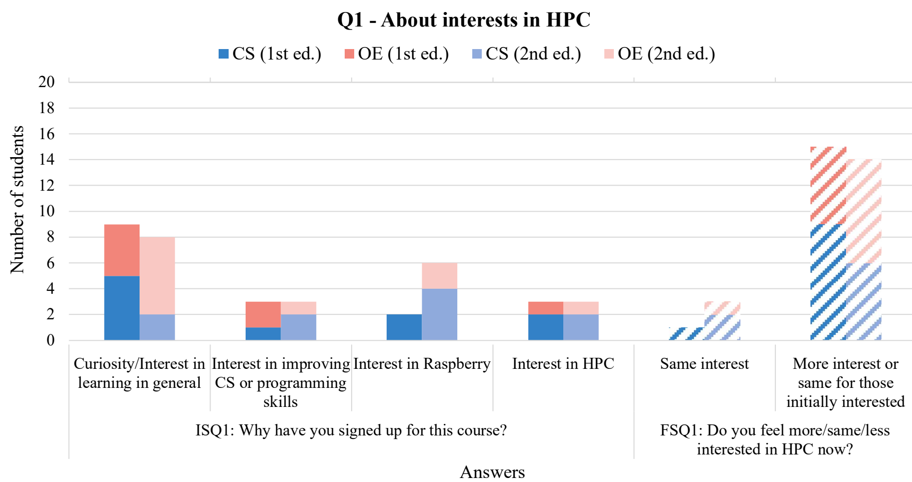

Figure 5: Answers of ISQ1 and FSQ1.

HPC interest has increased among attendees.

Regarding the HPC interest and awareness, Figure 5 (FSQ1) shows that, in both editions, the vast majority of students affirm having more interest in HPC than they had before taking the course, regardless of their studies. This is possibly justified by the fact that all of the students in both editions end up the course believing that HPC has an impact on their daily life, which is shown in Figure 4.

We consider that in case HPC impact in their daily life was more valued, HPC interest in taking the course would have been higher (see Figure 5 (ISQ1)). With that in mind, it seems reasonable that a low number of students applied to the course because of their interest in HPC. In particular, 18% of the students in the first edition (3 out of 17) and 15% (3 out of 20) in the second.

HPC knowledge has increased among attendees.

Related to the already mentioned lack of HPC knowledge, we analyze what the students believe HPC and supercomputers are. Figures 6 and 7, which refer to questions 3 and 4 respectively, illustrate their thoughts. The answers on these plots reflect that defining "HPC" or "supercomputer" terms implied associating them to “complex computations” or “optimizations” of the software or the hardware. Those results show that, although their awareness of the daily impact of HPC in our lives is moderate, they have a feeling about what HPC and a supercomputer are even before the course.

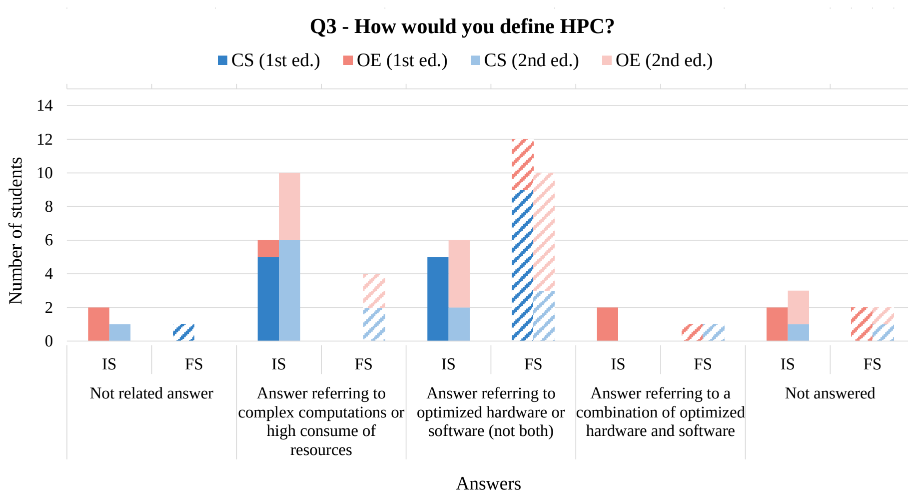

Figure 6: Answers of ISQ3 and FSQ3.

Overall HPC knowledge has increased after the course. Figure 6 reflects that in both editions, most of the students abandon their original idea of identifying HPC only with “solving very complex calculations using lots of resources” in favor of “optimizing available software and/or hardware resources”. In the first edition, this percentage increases from 41% (7 out of 17 students) to 81% (13 out of 16 attendees), while in the second edition the percentage changes from 30% (6 out of 20) to 65% (11 out of 17). This knowledge acquisition is also observed in Figure 7: the great majority of the students finalize the course relating “supercomputers” to systems equipped with optimized software and/or hardware. This fact is especially remarkable in the second edition, when 70% of the students (specifically 12 out of 17) show this final opinion, in contrast to the initial 50% (10 out of 20).

HPC will be taken into account in the professional career path.

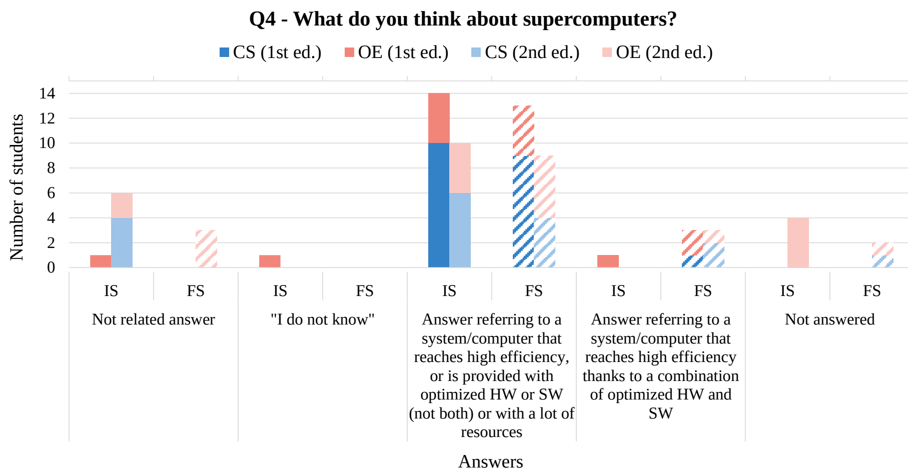

Figure 7: Answers of ISQ4 and FSQ4.

If students had known more about HPC, all of them could have answered that of course it could be applied in their professional career field in the beginning, even though maybe they would need more training, but this is only initially stated by approximately half of the students (see IS answers in Figure 8). The other half of the students, initially thought that HPC could not be applied in their areas or they did not know how it could be done.

5. COURSE EVALUATION: DISCUSSION AND LESSONS LEARNT

The remaining results extracted from the surveys and the course experience itself are summarized and discussed in this section.

Are students motivated to learn?

Motivation is vital for learning, and university students have it. However, we (as lecturers) do not always find a way to keep it alive. In Figure 5, we observe that (in both editions) approximately half of the students attending the course signed up for it because they were interested in learning and felt curious. Appealing their motivation to learn and their curiosity was our intention when

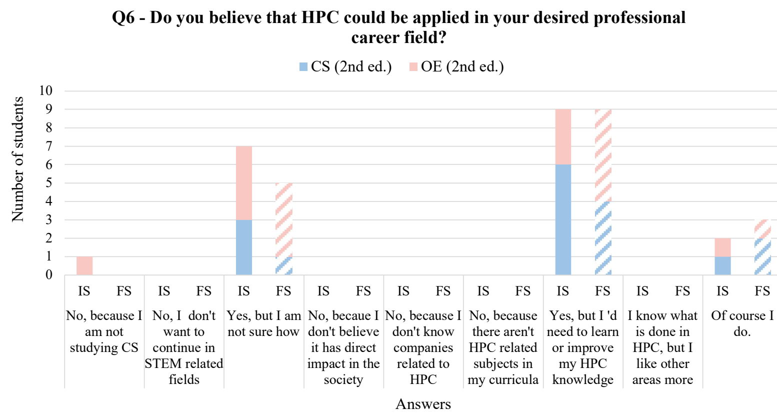

Figure 8: Answers of ISQ6 and FSQ6. Note that only 2^{nd} edition answers are provided, as this question was newly included in it.

considering how to bring HPC to students. We believe that an in-place "hands-on" experience is crucial, as stated by other authors [50, 51].

Choosing Raspberry Pi devices for the course and advertising their usage is attractive.

Figure 5 reflects that there is an interest in Raspberry Pi that motivates part of the students to attend the course. In fact, in the second edition, 30% of the students (6 out of 20) express that knowing more about Raspberry Pi is why they decided to participate in the course. It is curious that, in the first edition, the number of students interested in Raspberry Pi was much lower, around 12% (2 out of 17) and all of them were CS students. Despite the differences in terms of percentage between both editions, results show that finding an attractive way to present HPC is essential to initiate students in the field.

Raspberry Pi components provide sufficient flexibility and versatility.

Reasonable prices of Raspberry Pi and all the other components employed for the clusters offer the possibility of enabling students to group and develop their clusters independently. This lays good foundations for creativity and favors that, after completing the basic cluster configuration, they focus on what they are more interested in. For instance, hardware experiments such as interconnecting different clusters, understanding temperature consequences, or analyzing performance differences between shared and distributed memory executions.

The approach and programming of the course are appropriate to establish fundamental knowledge about HPC.

After the second edition of the course, we can reaffirm that a 10-hour program lets the students learn the basics of HPC, motivates them to keep/raise their HPC interest and curiosity, and encourages them to experiment with their cluster by applying it their proposals.

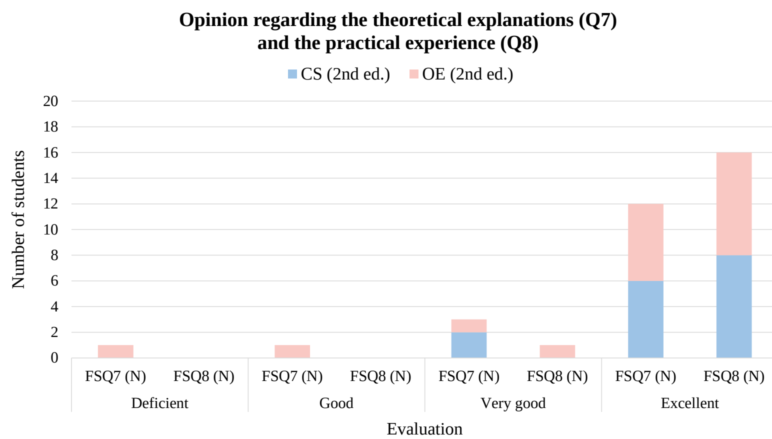

Figure 9: Answers of FSQ7 and FSQ8. Note that only second edition answers are provided, as these questions were newly included in it.

In the second edition of the course, we included two questions (Q7 and Q8) in the final survey to evaluate the degree of satisfaction with the course, whose answers are reflected in Figure 9. In general, we can conclude that students liked both theoretical explanations and practical activities. There are small differences between CS and OE students and we consider that it might be caused by the lack of CS knowledge that OE students have, compared to the CS ones. Besides, only one OE student considered that our empathy degree and quality of the help during practical activities were very good, instead of excellent (as all the other students stated), which makes us believe that what was not *very* good or *excellent* for certain students during the theoretical explanations was compensated afterward during the practical section of the course. Thus, we consider that the theoretical explanations should be kept as they prevent 1) the risk of losing the attention of CS students with too basic explanations, and 2) reducing the on-demand time slot because of providing longer explanations.

Students' original professional interests are kept, but now HPC is considered in those contexts.

We can observe in Figure 10 that future professional interests are very different from one to other students, and taking the course does not modify their preferences. However, Figure 8 reflects that, after the course, fewer students consider HPC is not applicable in their field or ignore how, and more believe that they could use it, although most express they would (naturally) need to improve their skills to be able to apply it.

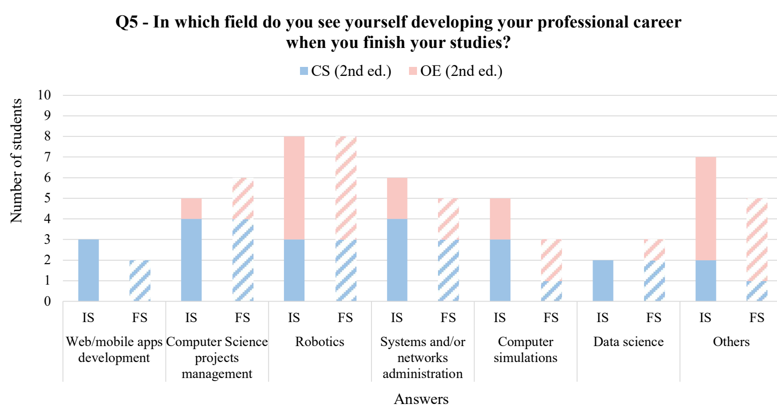

Figure 10: Answers of ISQ5 and FSQ5. Note that multiple answers where accepted for Q5.

For us, it is very important to ensure and remark that acquiring basic HPC knowledge does not modify their professional interest, but enriches their perspective of how HPC can be employed to improve several applications, even though they belong to very different fields.

Low interest in HPC among women.

After the two editions of the course, we observe a low number of women applying to it. We consider the main reason is the fact that the number of female students in the degrees which this course was offered is reasonably low, and consequently this imbalance is also naturally present in the course. The latest data regarding the number of women enrolling in STEM degrees in the university where the course took place [52] show that female students represent only 11%. In the first edition, female applicants were 7.7% (2 out of 26), while in the second edition were 10.5% (5 out of 48). Consequently, the number of female applicants seems to be consistent with the existing number of female students in these fields.

On the other hand, the number of final attendees does not keep consistent between both editions. While in the first edition the selected woman attended the course, in the second one none of them finally participated. We consider that we do not have enough information to make a strong statement about the reasons that lead to these results. In future editions, more information will be gathered regarding female interest.

One-day vs. two-day course.

One of the main differences between the first and second editions of the course was its time distribution; the first edition was split into two days of five hours each, while the second edition was carried out in one day for ten hours. This change in the format of the course was because the exam period was close by. Before the second edition of the course was published, we guessed that fewer students would apply because it was taking place on Saturday (a non-school day), and the personal effort from students would be greater, spending all day in the workshop. Surprisingly, quantitative results did not differ much from the previous edition. However, we consider them qualitatively better after analyzing the following three aspects:

- The number of applicants raised significantly from the previous edition to the second, going from 26 to 48.
- The number of applicants belonging to CS or OE slightly varied between the two editions. In the first edition, 58% of the applicants were from CS, while this percentage in the second edition turned to be 50%.

- The number of attendees who enrolled in the course versus the number of them that finished it was similar. In the first edition, 11% of the attendees dropped the course before its end. In the second edition, this amount rose to 15%.

The interest in the course increased drastically from one edition to the next one. There could be different reasons for that, but we consider that the previous knowledge about the course existence, together with offering it during a non-school day, has made an important difference.

Regarding the major of the applicants, similar numbers are obtained in both editions. In the first edition, CS students seemed to be a bit more interested than in the second one. We see this fact as another evidence that HPC is a transverse field that is widely used by many different sciences and engineering nowadays. Consequently, students from all areas may be interested in it.

The dropout rate is similar in both editions, being a bit higher in the second. We think that this may be caused by the duration of the course. Spending 10 hours on a day may be difficult for some of the attendees that may have other obligations to take care of.

In terms of the course contents, we observed a remarkable difference that was very satisfying from our point of view. In contrast to the first edition, in the second one, some of the students that progressed faster thanks to their previous CS knowledge were able to experiment and develop more activities (see Section 4.2.3). This was especially observable during the on-demand phase and gives a non-negligible value that, in our humble opinion, enriches the students' experience a lot. For this reason, we are determined to keep offering the course in this "whole day experience" format instead of splitting it into two sessions.

6. FOLLOW-UP: IMPACT OF ATTENDING THE COURSE

In the first quarter of the year 2021 (respectively two and three years after the first and second edition), the authors sent a short questionnaire to the participants of both editions of the course to follow-up on the impact and influence of attending it. The questionnaire was composed of two questions that the students rated from 0 ("*Not at all*") to 10 ("*Absolutely*"):

- Did attending the course influence any subsequent training choices? For example, in the selection of specialization pathways for undergraduate studies, or specific training courses.

- Do you think that the training obtained in the course has any impact on the field in which you are currently studying or working?

Figure 11 reports the questionnaire answers, showing in orange the influence over posterior training selection by the attendees, and in green, the impact in their current studies and/or the professional environment; the light tonalities refer to the first edition and darker tones to the second one.

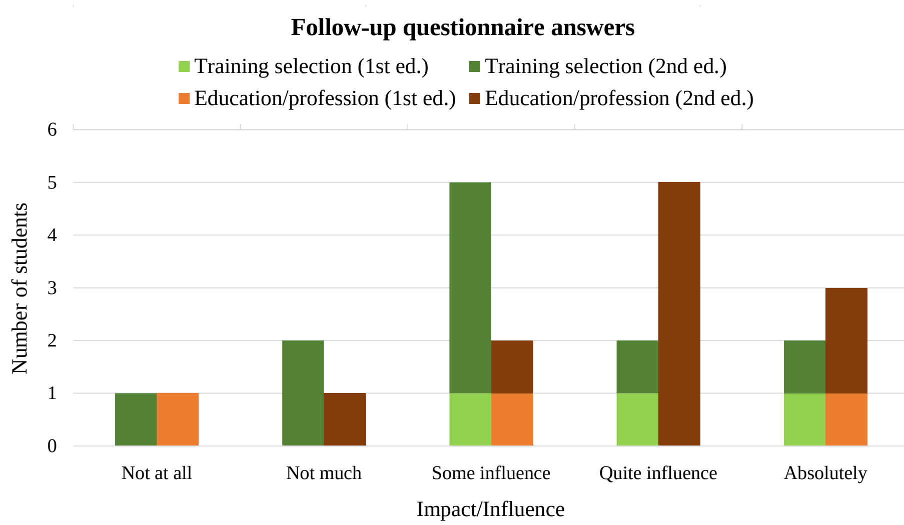

Figure 11: Answers of the follow-up questionnaire.

In total, 12 students out of 38, answered the follow-up survey. Three of them attended the first edition, which means that only 16% of that edition members are represented; the other nine answers belong to students from the second edition, which is a much more representative opinion, reflecting 45% of the original attendance. Most of the students lose or stop checking their university email inbox when they conclude their studies, and that can be the reason why some of them did not reply.

In general, there is a moderate influence of the course attendance over training selection and a high impact of the acquired skills in the studies and/or professional development. The lack of HPC content in the engineering syllabus (already explained in this paper) can be the reason why the attendees did not vary much in their afterward training selection, as there are not many options related to what they learned in the course. However, as it was also stated in the introduction, nowadays engineering-related professions often require generalist HPC knowledge, and that is reflected in the high impact of the course HPC apprenticeship in the attendees' education and/or profession.

The already mentioned low representation of the first edition can justify the *polarized* opinions, while a most representative scope like that regarding the second edition homogenizes the data and is closer to what can be observed globally without differentiating editions. For this reason, we consider that the main conclusion that can be extracted regarding the different editions is that the attendees of the second one show a clear impact on the course in terms of training selection and mostly regarding educational/professional development.

7. CONCLUSIONS

HPC necessity and importance are unquestionable, and we have realized that there is a lack of related content in the engineering syllabus of the UJI, particularly in CS subjects. Motivated students and an increasing necessity of HPC knowledge in the job market, moved us to propose the course. Combining the "hands-on" experience and the "on-demand" approach works in favor of creativity and motivation, and results show that the students valued positively those aspects. Regarding HPC skills and knowledge, it is seen that the number of students that define appropriately HPC and supercomputer has increased.

On the other hand, the number of students that show more interest in HPC raises after the course.

In terms of motivating the students, employing Raspberry Pi devices and giving them some freedom to experiment (in our case, through the "on-demand" part of the course) has been largely positive and has helped to achieve the objectives. In fact, the use of Raspberry Pi attracted the attention of the students to enroll in the course. Moreover, offering a single-day course instead of splitting it into two afternoons helps to maintain their interest. This new format eases the understanding while linking all the new concepts we present. Besides, it facilitates to dedicate more time to the "on-demand" section, as no recaps are needed, and fewer questions regarding previously explained things are asked. It is also remarkable the higher number of activities that the most advanced groups can experience on the cluster, as no extra reboots or extra reconfigurations are done due to restarting everything one day after.

Regarding the students' awareness about HPC and its applicability in the real world, we can conclude that it has increased. Students know more about the impact of HPC on daily life after the course. Moreover, students seem more confident about the possibility of applying HPC in their fields, although their professional interests have not changed after the course.

Finally, we can conclude that the course had an impact not only in the short-term but also in the mid-/long-term. The follow-up questionnaire results show that, in some cases, attendees chose HPC-related subjects and courses after the proposed course. Moreover, a reasonable percentage of them are applying their HPC knowledge in their current job or field of study.

Notice that these conclusions are extracted from a small population, so they may be less accurate in a scaled-up environment. However, the course could not be carried out with more students due to the tight budget available for this activity.

8. FUTURE WORK

We are satisfied with the results derived from completing the course and will offer subsequent editions of it. We plan to look for funding from technological companies so we can include a small competition at the end of the course in which a general challenge is proposed, following the spirit of student cluster competitions.

Moreover, we are working on including Slurm installation and usage, as well as defining an extended collection of HPC applications arising from the different students' specific fields of interest.

Considering the current health crisis due to COVID and the limitations we are experiencing in terms of planning in-person courses, it is unavoidable for us to think of designing an extension that allows us to run everything virtually. Although the hands-on is one of the highlights of this course, the new pandemic situation is affecting and will possibly affect further editions, so we believe it is worth putting a big effort into designing ways to still allow this "hands-on" (though virtual) and maintaining the degree of freedom the students get in this course, which derives from the "on-demand" approach.

One of the authors' concerns is promoting HPC not only in university environments but also in high schools. We believe that teenagers would earlier discover CS in general and HPC in particular through experiencing a properly adapted version of this course. As a consequence, their corresponding perspective could be more adjusted to reality and less limited than nowadays. We also consider that discovering and experiencing CS at earlier ages could increase interest in this field among girls, and hopefully contribute to reducing the gender imbalance.

Lastly, we are also considering coupling all the materials that make this course possible and publishing them (extended with complementary explanations, exercises, and developments) in book format.

Acknowledgment

The researcher Sandra Catalán was supported by MICINN under the project Heterogeneidad y Especialización en la Era Post-Moore, RTI2018-093684-B-I0. Rocío Carratalá-Sáez was supported by projects CICYT TIN2014-53495-R and TIN2017-82972-R of MINECO and FEDER, project UJI-B2017-46 of UJI, and the FPU program of MECD. Sergio Iserte was supported by a post-doctoral fellowship from Valencian Region Government and European Social Fund, APOSTD/2020/026. This course was possible thanks to the funding provided by the High Performance Computing and Architectures Research Group (HPC&A), and Computer Science Department (DICC) from UJI. The authors also want to thank the anonymous reviewers of the work presented in [1], whose comments were very helpful to enrich the course. Likewise, authors appreciate the anonymous reviews and suggestions of the current manuscript which helped to improve the quality of the paper.

[1] R. Carratalá-Sáez, S. Iserte, S. Catalán, Teaching on Demand: an HPC Experience, in: Proceedings of EduHPC 2019: Workshop on Education for High Performance Computing - Held in conjunction with SC19: The International Conference for High Performance Computing, Networking, Storage and Analysis, 2019, pp. 32–41.

[2] G. V. Wilson, A glossary of parallel computing terminology, IEEE Concurrency (out of print) 1 (01) (1993) 52–67. doi:10.1109/88.219862.

[3] HPC Wiki, https://hpc-wiki.info/hpc/HPC-Dictionary, accessed: March 2021.

[4] HPC Glossary (The University of Tennessee), https://www.nics.tennessee.edu/hpc-glossary, accessed: March 2021.

[5] HPC Glossary of Terms (Pomona College), https://www.pomona.edu/administration/hpc/access-guidelines/glossary, accessed: March 2021.

[6] J. Saffran, G. Garcia, M. A. Souza, P. H. Penna, M. Castro, L. F. W. Góes, H. C. Freitas, A low-cost energy-efficient raspberry pi cluster for data mining algorithms, in: F. Desprez, P.-F. Dutot, C. Kaklamanis, L. Marchal, K. Molitorisz, L. Ricci, V. Scarano, M. A. Vega-Rodríguez, A. L. Varbanescu, S. Hunold, S. L. Scott, S. Lankes, J. Weidendorfer (Eds.), Euro-Par 2016: Parallel Processing Workshops, Springer International Publishing, Cham, 2017, pp. 788–799.

[7] M. F. Cloutier, C. Paradis, V. M. Weaver, A raspberry pi cluster instrumented for fine-grained power measurement, Electronics 5 (4). `doi:10.3390/electronics5040061`.
URL `https://www.mdpi.com/2079-9292/5/4/61`

[8] W. Hajji, F. P. Tso, Understanding the performance of low power raspberry pi cloud for big data, Electronics 5 (2). `doi:10.3390/electronics5020029`.
URL `https://www.mdpi.com/2079-9292/5/2/29`

[9] K. Doucet, J. Zhang, Learning cluster computing by creating a raspberry pi cluster, in: Proceedings of the SouthEast Conference, ACM SE '17, Association for Computing Machinery, New York, NY, USA, 2017, p. 191–194. `doi:10.1145/3077286.3077324`.
URL `https://doi.org/10.1145/3077286.3077324`

[10] A. Mappuji, N. Effendy, M. Mustaghfirin, F. Sondok, R. P. Yuniar, S. P. Pangesti, Study of raspberry pi 2 quad-core cortex-a7 cpu cluster as a mini supercomputer, in: 2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE), 2016, pp. 1–4. `doi:10.1109/ICITEED.2016.7863250`.

[11] A. Lasserre, R. Namyst, P.-A. Wacrenier, Easypap: a framework for learning parallel programming, in: EduPar-20: 10th NSF/TCPP Workshop on Parallel and Distributed Computing Education (EduPar-20), 2020, pp. 276–283.

[12] L. Marzulo, C. Bianchini, L. Santiago, V. Ferreira, B. Goldstein, F. França, Teaching high performance computing through parallel programming marathons, in: 2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), 2019, pp. 296–303.

[13] S. Prasad, K. Kant, Y. Robert, A. Chtchelkanova, S. Das, A. La Salle, R. Le Blanc, A. Rosenberg, F. Dehne, M. Lumsdaine, S. Sahni, M. Gouda, D. Padua, B. Shirazi, A. Gupta, M. Parashar, A. Sussman, J. Jaja, V. Prasanna, C. Weems, J. Wu, Nsf/ieee-tcpp curriculum initiative on parallel and distributed computing - core topics for undergraduates, in: SIGCSE'11 - Proceedings of the 42nd ACM Technical Symposium on Computer Science Education, SIGCSE'11 - Proceedings of the 42nd ACM Technical Symposium on Computer Science Education, 2011, pp. 617–618, 42nd ACM Technical Symposium on Computer Science Education, SIGCSE 2011 ; Conference date: 09-03-2011 Through 12-03-2011. doi:10.1145/1953163.1953336.

[14] V. Holmes, I. Kureshi, Developing High Performance Computing Resources for Teaching Cluster and Grid Computing Courses, Procedia Computer Science 51 (2015) 1714–1723.

[15] P. López, E. Baydal, Teaching high-performance service in a cluster computing course, Journal of Parallel and Distributed Computing 117 (2018) 138–147.

[16] A. A. Datti, H. A. Umar, J. Galadanci, A Beowulf Cluster for Teaching and Learning, Procedia Computer Science 70 (2015) 62–68.

[17] T. Sterling, T. Sterling, D. J. Becker, D. Savarese, J. E. Dorband, U. A. Ranawake, C. V. Packer, Beowulf: A Parallel Workstation For Scientific Computation, in Proceedings of the 24th International Conference on Parallel Processing 1 (1995) 11—-14.

[18] L. Alvarez, E. Ayguade, F. Mantovani, Teaching HPC Systems and Parallel

Programming with Small Scale Clusters of Embedded SoCs, in: Workshop on Education for High Performance Computing (EduHPC), 2018, pp. 1–10.

[19] J. Adams, S. J. Matthews, E. Shoop, D. Toth, J. Wolfer, Using Inexpensive Microclusters and Accessible Materials for Cost-Effective Parallel and Distributed Computing Education, The Journal of Computational Science Education 8 (3) (2017) 2–10.

[20] Steven J. Vaughan-Nichols, Build your own supercomputer out of Raspberry Pi boards (2013).
URL https://www.zdnet.com/article/build-your-own-supercomputer-out-of-raspberry-pi-boards/

[21] Eva Havelkova, Parallel computing demonstrations on Wee Archie (2018).
URL https://youtu.be/DN_cqiSg8jU

[22] Teaching PDC Using the Raspberry Pi.
URL https://csinparallel.org/csinparallel/raspberry_pi.html

[23] PJ Evans, Build a Raspberry Pi cluster computer (2020).
URL https://magpi.raspberrypi.org/articles/build-a-raspberry-pi-cluster-computer

[24] R. P. Foundation, Raspberry pi model 4b, https://www.raspberrypi.org/products/raspberry-pi-4-model-b/.

[25] D. Eadline, High Performance Computing for Dummies, Wiley Publishing, 2009.

[26] C. Severance, K. Dowd, High Performance Computing, O'Reilly Media, Inc, 1998.

[27] ASC Community, The Student Supercomputer Challenge Guide, Springer Singapore, Singapore, 2018.
URL http://link.springer.com/10.1007/978-981-10-3731-3

[28] A. C. : Björn-Sören Gigler, A. Verbeek, Financing the future of supercomputing: How to increase the investments in high performance computing in Europe, European Investment Bank, 2018.

[29] B. Obama, Executive Order - Creating a National Strategic Computing Initiative, Vol. 29, The White House, 2015.

[30] E. Joseph, J. Wu, S. Conway, S. Tichenor, Benchmarking Industrial Use of High Performance Computing for Innovation, The Council on competitiveness, 2008.

[31] BSC - Computing with you, `https://www.bsc.es/discover-bsc/computing-with-you`.

[32] TACC - Tours, `https://www.tacc.utexas.edu/education/tours`.

[33] F. Banchelli, F. Mantovani, Filling the gap between education and industry: evidence-based methods for introducing undergraduate students to hpc, in: 2018 IEEE/ACM Workshop on Education for High-Performance Computing (EduHPC), 2018, pp. 41–50.

[34] B. Neelima, J. Li, Introducing high performance computing concepts into engineering undergraduate curriculum: A success story, in: Proceedings of the Workshop on Education for High-Performance Computing, EduHPC '15, ACM, New York, NY, USA, 2015, pp. 6:1–6:8.

[35] B. Chaudhury, A. Varma, Y. Keswani, Y. Bhatnagar, S. Parikh, Let's HPC: A web-based platform to aid parallel, distributed and high performance computing education, Journal of Parallel and Distributed Computing 118 (2018) 213–232.

[36] J. L. Hennessy, D. A. Patterson, Computer Architecture: A Quantitative Approach, 6th Edition, Morgan Kaufmann Publishers Inc, 2017.

[37] TOP500 List, `https://www.top500.org`.

[38] LINPACK library, `https://www.netlib.org/linpack/`.

[39] A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers, https://www.netlib.org/benchmark/hpl/.

[40] Marenostrum Supercomputer from the Barcelona Supercomputing Center (BSC), https://www.bsc.es/marenostrum/marenostrum, accessed: March 2021.

[41] Raspbian free operating system based on debian optimized for the raspberry pi hardware., https://raspbian.org.

[42] NFS protocol, http://nfs.sourceforge.net/.

[43] S. Plimpton, Short-Range Molecular Dynamics, Journal of Computational Physics 117 (6) (1997) 1–42.

[44] OpenMP, https://www.openmp.org/.

[45] W. D. Gropp, W. Gropp, E. Lusk, A. Skjellum, A. D. F. E. E. Lusk, Using MPI: portable parallel programming with the message-passing interface, Vol. 1, MIT press, 1999.

[46] Mpich, https://www.mpich.org/.

[47] Openmpi, https://www.open-mpi.org/.

[48] L. Zhou, S. Guo, Thermal management of arm socs using linux cpufreq as cooling device, in: COMPUTER MODELLING & NEW TECHNOLOGIES, 2015, pp. 162–167.

[49] A. B. Yoo, M. A. Jette, M. Grondona, SLURM: Simple Linux Utility for Resource Management, in: D. Feitelson, L. Rudolph, U. Schwiegelshohn (Eds.), Job Scheduling Strategies for Parallel Processing: 9th International Workshop, JSSPP 2003, Seattle, WA, USA, June 24, 2003. Revised Paper, Springer Berlin Heidelberg, Berlin, Heidelberg, 2003, pp. 44–60. doi:10.1007/10968987_3.

[50] N. J. Nersessian, Conceptual change in science and in science education, Synthese 80 (1) (1989) 163–183.

[51] M. P. Clough, Using the laboratory to enhance student learning, Learning science and the science of learning (2002) 85–94.

[52] C. García, Los grados de Ciencias solo captan a una de cada 10 alumnas en la UJI, El Periódico Mediterráneo.
URL https://www.elperiodicomediterraneo.com/noticias/castellon/grados-ciencias-solo-captan-10-alumnas-uji_1204005.html

Appendix A. User guide

In this appendix, we present a "step by step" guide detailing the tasks proposed in the course. Besides, it is also specified if the instructions must be executed on every Raspberry Pi device or only in the front-end.

Appendix A.1. Assemble the cluster

- Insert SD cards in Raspberry Pi devices (instructors need to previously prepare them to be bootable with Raspbian OS).
- Connect the screen, the mouse and keyboard to one of the Raspberry Pi devices.
- Connect all the Raspberry Pi devices to the switch using Ethernet cables.
- Connect each Raspberry Pi to the energy.

Appendix A.2. Basic configuration of Raspbian OS (in all the devices)

- Run each Raspberry Pi and perform the basic OS configurations, such as setting the right clock time, and the proper keyboard and system language. To do this, alternate the screen, the mouse and keyboard from one to the other Raspberry Pi devices.

Appendix A.3. Further configuration of Raspbian OS (in all the devices)

- Assign a hostname to each device (for example, use `nodeX` where `X` is substituted by 1, 2, 3, and 4 for each device).
- Enable SSH. This can be done through system interfaces configuration.
- It is highly recommended to check that geographical location is correctly set.
- After rebooting, configure DHCP using:
  - interface: `eth0`

    - static ip_address: `192.168.0.x/24` where `x` is substituted by 1, 2, 3, and 4 for each device.
    - static routers: `192.168.0.1`
    - static domain_name_severs: `192.168.0.1`

- Reboot the devices.

Appendix A.4. Configure the front-end (`node1`)

- In the system preferences, set hostname and enable SSH:
    - Hostname: `node1`
    - Interfaces: SSH
- Reboot the system and then configure DHCP by modifying the `/etc/dhcpcd.conf` file to add:
    - interface eth0
    - static ip_address=192.168.0.1/24
    - static routers=192.168.0.1
    - static domain_name_severs=192.168.0.1 8.8.8.8
- Reboot the system or type `sudo ifconfig eth0 down & sudo ifconfig eth0 up`.
- Enable WiFi.
- Create the file `/lib/dhcpcd/dhcpcd-hooks/60-gw` and write `route del default gw 192.168.0.1` on it.
- Reboot the system and overwrite the file `/etc/hosts` with:
    - 192.168.0.1 node1
    - 192.168.0.2 node2
    - 192.168.0.3 node3

  - 192.168.0.4 node4

- Reboot the device and generate SSH keys typing `ssh-keygen`.
- Configure SSH in the other nodes, from `node1` by repeating three times (substituting `X` by 2, 3, and 4):
  - `ssh-copy-id nodeX`
  - `scp /etc/hosts nodeX:`
  - `ssh nodeX sudo mv hosts /etc/hosts`
- Update the system (`sudo apt-get update`) and install NFS server by typing `sudo apt-get install nfs-kernel-server`.
- Create a shared directory:
  - `sudo mkdir /SHARED`
  - `sudo chmod 777 /SHARED`
  - Modify the file `/etc/exports` by adding:
    - `/SHARED node2(rw, sync, no_subtree_check)`
    - `/SHARED node3(rw, sync, no_subtree_check)`
    - `/SHARED node4(rw, sync, no_subtree_check)`
  - `sudo exportfs -a`
- Make the shared directory accessible for the other nodes (these steps can be done by entering each node using SSH, and need to be done three times):
  - `sudo mkdir /SHARED`
  - `sudo chmod 777 /SHARED`
  - Modify the file `/etc/fstab` by adding `node1:/SHARED /SHARED nfs`
  - `sudo mount -a`

Appendix A.5. Install OpenMPI

- Download OpenMPI from `open-mpi.org`.
- Decompress downloaded files.
- Configure OpenMPI by typping `./configure --prefix=/SHARED/OpenMPI --enable-mpirun-prefix-by-default`.
- Install it by typing `make && make install`.
- Only in `node1`, modify the file `bash.rc` by adding `export PATH=/SHARED/openmpi/bin:$PATH`.
- Update `bash.rc` in the remaining nodes by typping `scp .bashrc nodeX:`.
- It is recommended to check that OpenMPI has been successfully installed by running `mpiexec hostname`.

Appendix A.6. Alternatively, to check different performance rates, MPICH can also be installed following these steps:

- Download MPICH from `https://www.mpich.org/`.
- Decompress downloaded files.
- Configure it by typping `./configure --prefix=/SHARED/mpich --disable-f77 --disable-fc --disable-fortran`.
- Install it by typing `make && make install`.
- Only in `node1`, modify the file `bash.rc` by adding `export PATH=/SHARED/mpich:$PATH`.
- Update `bash.rc` in the remaining nodes by typping `scp .bashrc nodeX:`.

Appendix A.7. Install LINPACK

In order to install, configure and execute LINPACK, students are encourage to follow the LINPACK installation guide, and so they get familiar with following an installation guide by themselves and facing system administrators issues. However, instructors will help them with the use of configuration wizards such as `https://www.advancedclustering.com/act_kb/tune-hpl-dat-file` or `http://hpl-calculator.sourceforge.net`.

Appendix A.8. Install LAMMPS

In the case of LAMMPS installation, students are led to use the official user guide. Furthermore, apart from installing the basic version of LAMMPS, students are also motivated to go further and install the OpenMP extensions from `https://lammps.sandia.gov/doc/Build_extras.html#user-omp`.

With LAMMPS building versions (MPI and OpenMP) students will be asked to carry out the scalability and performance evaluation using different approaches such as threads, processes and their combination.