Teaching on Demand: an HPC Experience

Rocío Carratalá-Sáez
Dept. of Computer Science and Engineering
Universitat Jaume I
Castelló de la Plana, Spain
rcarrata@uji.es

Sergio Iserte
Dept. of Mechanical and Eng. Construction
Universitat Jaume I
Castelló de la Plana, Spain
Dept. of Computer Science
Universitat de València
Valencia, Spain
siserte@uji.es

Sandra Catalán
Dept. of Computer Sciences
Barcelona Supercomputing Center
Barcelona, Spain
sandra.catalan@bsc.es

***Abstract*—In this work we present the experience of the course "Build your own supercomputer with Raspberry Pi", offered as a non-mandatory workshop with the purpose of bringing High Performance Computing (HPC) closer to bachelor students of Universitat Jaume I (UJI, Spain). The intention of the course is twofold; on the one hand, we target towards increasing the knowledge of Computer Science and Engineering students about the labor performed by the HPC community; on the other hand, we aim to create a personalized experience for each student by fulfilling their curiosity about the topics presented and discussed in the class. In order to evaluate the impact and learning, we analyze two surveys filled out by the students respectively before and after the course, where HPC interest and knowledge are exposed.**

***Index Terms*—Computational Cluster, Undergrad Teaching, System Administration, Parallel Distributed Computing, Raspberry Pi.**

I. INTRODUCTION

The term "supercomputer" is usually unfamiliar to college students, not only from Computer Science (onward, CS) but also from other disciplines that base their developments on this kind of machines, such as mechanical engineering, biology, finances, etc. Although they are aware of the need for better computers that perform faster computations, they may know nothing about the underlying infrastructure responsible for providing High Performance Computing (HPC) to real-world applications which they may use, or develop, in a near future.

This lack of a down-to-earth experience, motivated us to design and conduct a 10 hours introductory course to clusters of computers, where students experience the whole process of setting up and configuring a cluster. Besides, the UJI's course, entitled "Build your own supercomputer with Raspberry Pi", goes further and challenges students to install and execute a series of HPC applications, widespread in production systems, in their own built cluster.

Moreover, the course does not only target CS students, but any student from a scientific or engineering discipline. For this reason, in this paper, we present a partially on-demand method that allows students to learn the very basics of computer clusters depending on the interests and/or needs of each student. Thus, we cover from a system administration perspective, focused on the cluster management, to a data scientific approach, interested in applications. For this purpose, the course presents the big picture of supercomputers and their implication on day-to-day issues.

Our course aims to involve the students in the cluster set up process, promoting its development from scratch and illustrating its use while compiling and testing complex HPC applications such as LINPACK [14] or LAMMPS [8]. This way, we fulfill what we consider a non-negligible knowledge gap in the current CS and, in general, engineering syllabus. Finally, we adapt the last part of the course to each of the particular interests of the students, leveraging a "teaching on demand" methodology which allows them to choose in what do they prefer to invest the remaining time of the course.

II. RELATED WORK

Usually, courses involving supercomputing issues focus on parallel programming languages or paradigms [5], [6]. Others [7], explain the process of designing a cluster for educational purposes. Commonly, the resulting cluster is based on Beowulf distributed computing system [9], useful for recycling deprecated machines in the center.

In [4], 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 [29]. 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. So, depending on the course scope and the syllabus, they group strategies by one cluster per student, one cluster per group, and one cluster per class. Projects like [30], provide an image for a Raspberry Pi 3 system with pre-installed support for clusters and other interesting parallel suites.

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 [1]–[3] in the course, such as software performance, cluster scalability, networking, CPU frequency, cooling, benchmarking, and scientific applications.

III. METHODOLOGY

A. Population

Participants enrolled voluntarily in the course "Build your own supercomputer with Raspberry Pi". In total, 26 students applied for the course; among them, 20 were selected as participants, and 18 attended the course. The moderate amount of places in the course is constrained by the budget and available facilities (see Section III-D).

Regarding the topic, although it naturally belongs to the CS field, it was offered to all engineering students in the university given the general approach considered in the course and the transverse content. To select the attendants, we followed a simple idea: as we believe the lack of HPC and supercomputing knowledge should be corrected as soon as possible, we gave priority to the students that were in an early stage of their studies. Due to the fact that the course begins from the basis, only basic Unix and Shell knowledge prerequisites were recommended.

In order to analyze the findings and outcomes of the course, we divide the attendants in CS and other engineering (OE) students. The distribution of students according to their studies is shown in Table I. Note that, in order to maintain a balanced distribution in terms of applicants' interest and background, we selected a similar amount of participants from CS and OE.

	CS	OE
Applicants	58%	42%
Selected	60%	40%
Attendants	66%	33%

TABLE I
PARTICIPANTS CLASSIFICATION PER FIELD (COMPUTER SCIENCE - CS, OTHER ENGINEERING - OE), AND DEGREE OF PARTICIPATION (APPLICANTS, PARTICIPANTS, ATTENDANTS).

B. Objectives

When designing the course, we chose its content in order to guide the students in such a way that allowed them 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 which ones are nowadays trends in this field.
- *Objective 2)* Change their mind about thinking of HPC only in huge supercomputers that belong to huge companies.
- *Objective 3)* Be able to understand the needs (both in terms of hardware and software) of a supercomputer employed in HPC.
- *Objective 4)* Recognize current applications where HPC is necessary.
- *Objective 5)* Enjoy the learning process and avoid classical lessons pressure feelings.

C. Teaching methodology

The target of this course was introducing HPC to CS and OE students in our university, while covering the exposed objectives. From our experience, we detected a lack of knowledge about what HPC is and how it helps people in their daily life. For this reason, we designed this course in order to increase the interest of the attendants in the field by providing first-hand experience, while building a small scale cluster based on Raspberry Pi 3B+ devices [10].

We considered that building a cluster from scratch by assembling the hardware, and configuring the software to ensure the proper functioning, was the best way to experience HPC. During the training part of the course, attendants built their own real-wise HPC environment facing actual problems that are found when those systems are set up and discovering the skills that HPC system administrators require. Moreover, a whole vision of the HPC environment was provided in order to make attendants aware of the relevance of the field. For that reason, the course included information about different companies, institutions, and facilities in the field. In this way, attendants got a more complete vision of how HPC systems are built, the different approaches taken by those companies, and the economic impact of their products.

On the methodology side, the course was divided into three phases distributed in two sessions. The first part followed a traditional approach; a theoretical introduction was provided in order to set the basis and contextualize the course. The second part combined theoretical expositions with "hands-on" experiences; students followed a guide with the required steps in order to assemble the provided hardware, and also to install the basic software to configure the cluster. The third part of the course was approached as an on-demand course; each group of students, according to its interests, either varied the configuration of the cluster or received extra information about tools that were interesting in that specific case.

All these three phases are exposed in detail in Sections III-C1, III-C2, and III-C3.

1) First part - Theoretical introduction: the course starts with a brief introduction about the concept of HPC, the facilities used by the community, and also some relevant information regarding this field. This means that, at the beginning, we ask the students if they can define terms such as *supercomputer* or *HPC*. Then, we use well known examples such as weather prediction computations, social networks analysis to personalize adverts, voice recognition in smartphones, or traffic information in Google Maps application, to make them think about the need of speed in certain operations, about treating huge amounts of information, about well trained services, or about real-time updated information.

When all those situations where HPC is crucial have been exposed and discussed, we also present the existence of the Top500 list [16], and the LINPACK benchmark [14]. We leverage this moment to expose the features of the system occupying the first position in the list, and compare it to the Spanish system *Marenostrum 4*, which also appears in the Top500.

Each concept is simply defined, giving a general idea to the audience, promoting students' discussions, and avoiding deeper explanations that are out of the scope of the course. This introduction is supported by a set of slides. Moreover, we also include at this point a visit to the university computing data center, so they can see an actual HPC system.

In this first part of the course, we mainly cover objectives 1, 2, and 4.

2) Second part - User guide: the second part of the course consists of three guided modules that provide the basic information in order to assemble the cluster, configure it, and install and configure some HPC applications. To this end, a user's guide[1] is provided to the students. This document includes detailed information about the concepts explained in the theoretical introduction, and also about the hardware that is going to be used. This is described in the following paragraphs.

a) Hardware setup: each group is in charge of assembling its own cluster composed of 4 Raspberry Pi Model 3B+ devices (See Figure 1). In addition, a router, the corresponding Ethernet wires, 4 SD cards, and the power supplies are provided. Although the guide has already been provided and the students progress on their own, depending on their previous knowledge and agility, the instructors leverage this moment to explain the functions of each hardware component, especially describing the Raspberry Pi features.

Fig. 1. A Raspberry Pi cluster assembled during the course.

b) Cluster configuration: The basic software configuration of a cluster comprises the installation of the operating system[2], the network configuration, and the file system set up. Given the time restrictions of the sessions, the operating system is pre-installed in bootable SD cards handed out to the attendants. Thus, 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.

[1]To see the manual and the slides employed during the course, check the course repository https://github.com/rociocarratalasaez/BuildYourOwnSupercomputer.

[2]We opted for using Raspbian OS [17], concretely the version released in March 2018 (kernel version 4.9).

The network configuration is essential to make all the devices work collaboratively. On the other hand, a shared file system (Network File System, NFS [15]) is enabled through the cluster in order to facilitate the access to the files within the cluster.

The following list details the applications, models, and services installed by the students, in the order they do it:

- Chromium web browser and Leafpad text editor. These applications are already installed, but we recommend their usage to the students, as most of them are not familiarized with console commands (and this is out of the scope of the course).
- Configure DHCP using the terminal in one of the nodes, which will act as the main node.
- Remove default gateway in all the nodes, enable WiFi, and provide each Raspberry Pi with an IP direction.
- Configure SSH by creating public-private pair of keys in the main node, and then sending the SSH key to the other nodes.
- Install the NFS, and create a directory shared by all the nodes.
- It is highly recommended that, at this point, students check that each node can reach the other nodes, as well as use the NFS properly (simply by creating a text file located in the shared directory, and then editing it from different nodes).

Due to the fact that part of the listed items take some time to download and also to configure and/or install, those "waiting moments" are used to explain the usage and features of the different tools to the students.

c) HPC applications: in order to provide a realistic experience about currently used HPC applications, LINPACK and LAMMPS are compiled and configured. LINPACK is used to rank the supercomputers gathered in the Top500 list, while LAMMPS is a classical application for molecular dynamics modeling. Both applications require the use of OpenMP programming model [11] and Message Passing Interface (MPI) [18] in order to exploit all the cores in the cluster. OpenMP is natively integrated into GCC, consequently, no extra installations are required. However, MPI needs to be installed. Concretely, we opted for using MPICH [12], since it is one of the most popular open-source libraries employed in nowadays clusters, along with OpenMPI [13].

The following list details the steps that the students need to complete at this part of the course:

- Install MPICH (only the main node; afterwards copy `.bashrc` file to the other nodes).
- Download LAMMPS, and compile its MPI version.
- After discussing with the students the importance of leveraging also threads and not only available nodes, it is suggested to check the official website, where the way to configure LAMPSS with OpenMP is explained.
- Install OpenMPI and compare LAMMPS performance results when using it instead of MPICH.
- Install LINPACK.

Objectives 3 and 4 are mainly covered during this second part of the course.

3) Third part - Learning on demand: the third part of the course comprised the second session. In this phase, after ensuring that LAMMPs application is installed in all the clusters, each group proposes a target according to its interests and preferences. Concretely, the students expressed interest in checking different LAMMPs configurations for analyzing various behaviors, evaluating the workload of the machine and testing frequency-related tools, and also on creating a larger cluster by joining two clusters in order to check the scalability of LAMMPs in this new scenario. More details about these tests are exposed in the following paragraphs.

a) Performance and frequency tools: one of the interests expressed by part of the attendants was to widen their knowledge about the relationship between performance and energy consumption, the impact of power consumption in HPC facilities, and which tools or strategies can be applied in order to control these facts.

In order to fulfill the lack of knowledge about this topic, extra information was provided. We presented a few numbers about power consumption and the associated economic costs on current supercomputers to illustrate the relevance of the problem. In addition, a rough idea about how performance affects power consumption was given; dynamic and static power concepts were introduced and their interaction was explained. Finally, to give a real experience about the topic, *cpufreq* utilities [20] were presented and the attendants tried them on the cluster. In this way, they could retrieve information about the actual processors' frequency and change it to check how performance is affected.

b) Creating a larger cluster: the other part of the attendants were curious about the scalability of LAMMPs when running it on a larger cluster. They opted for linking two of the assembled clusters to create a bigger one with eight nodes.

The proposed change required 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 checked that the new cluster was working properly, they had the opportunity to try different configurations for LAMMPs and analyze which of them delivered the best performance.

We believe that the Objective 5 is attained by promoting this "hands-on" approach, and the fact that the students propose the work they want to develop during the third part of the course. Note that the students are not given a final mark at the end of the course, and this lets them feel relaxed and free to experience whatever they feel curious about.

A detailed "step-by-step" guide can be found in the Appendix.

D. Human resources, material and budget

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

Component	**Quantity**	**Individual 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 II
DESCRIPTION OF THE COMPONENTS REQUIRED FOR EACH CLUSTER DETAILING THEIR PRICES AND QUANTITIES.

Given the available budget and the total cost of each cluster, we could offer 20 places in the course. In this way, each group has a maximum of 4 members and we ensure the participation of all the members in the process of setting up the cluster.

On the other hand, the amount of human resources is essential in this course, especially to guarantee 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 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. In this way, one person can take care of these groups that need more time to complete the user's guide configuration (it is important to take into account that some students are not from CS and also that there are "early-stage" CS students that are not yet provided with a wide background in all the *cluster building and configuring* related fields). Another person coordinates all the groups that are interested in creating a larger cluster, either combining the existing groups to create larger clusters or creating a unique cluster by combining all the assembled clusters. The third person can manage the performance-energy consumption groups by given a general explanation about the topic and helping the groups following this line with the use of performance tools. Moreover, there could appear new interests in future editions of the course, so it would be even more necessary to keep three lecturers available to cover all the demands.

E. Data collection procedure and instrument

The new teaching approach presented in this paper to let students became closer to HPC implied risks and challenges, and we wanted to be able to judge the impact of this "custom tailored" teaching methodology on students. For this purpose, we collected qualitative information both at the beginning of the course and at the end. Data were collected through two anonymous surveys (an initial one right at the beginning of the course and a final one at the very end).

The initial survey (IS) consisted of the following four questions:

- ISQ1: Why have you signed up for this course?
- ISQ2: Do you think that HPC has influence on your day to day? How?
- ISQ3: How would you define HPC?
- ISQ4: What do you think about supercomputers?

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

- FSQ1: Do you feel more/same/less interested in HPC now?
- FSQ2: Do you think that HPC has an influence on your day to day? If so, how?
- FSQ3: How would you define HPC?
- FSQ4: What do you think about supercomputers?

We have categorized all students' survey answers to be able to analyze how they felt and what they knew about HPC before the course, and which was the progress in terms of HPC knowledge after the course. A detailed categorization of each answer can be found in Table III.

ISQ1 Categories	
Description	**Number**
Curiosity/Interest in learning in general.	1
Interest in improving CS or programming skills.	2
Interest in Raspberry.	3
Interest in HPC.	4
ISQ2 and FSQ2 Categories	
Description	**Number**
"No."/"Not directly."	1
"I do not know."	2
"I suppose/believe it has influence."	3
"Yes, it has influence."	4
ISQ3 and FSQ3 Categories	
Description	**Number**
Not related answer.	1
Answer referring to complex computations or high consume of resources.	2
Answer referring to optimized hardware (or software).	3
Answer referring to a combination of optimized software and hardware.	4
ISQ4 and FSQ4 Categories	
Description	**Number**
Not related answer.	1
"I do not know."	2
Answer referring to a system/computer that reaches high efficiency, or is provided with optimized hardware (or software) or with a lot of resources.	3
Answer referring to a system/computer that reaches high efficiency thanks to a combination of optimized hardware and software.	4
FSQ1 Categories	
Description	**Number**
Same interest.	2
More interest or same interest for those initially interested.	3

TABLE III
CATEGORIES OF THE INITIAL AND FINAL SURVEY ANSWERS.

IV. MOTIVATIONS

The initiative (and subsequent proposal) of this course is based on several hypotheses and observations that are presented in this section. These motivations are divided in two categories: Motivations that impulsed the creation of the course (and not addressed in the surveys), and motivations that constitute the objectives to be improved. All the proposed hypotheses are supported by the analysis of related bibliography; moreover, for those motivations that are targeted in this course, we present how they are improved through the results obtained in the surveys.

A. The reasons to create this course.

HPC society interest and need is increasing.

It is evident that the amount of Computer Science professionals which require HPC skills is increasing due to the importance of this area in many social [24], [25] and professional [23] fields.

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

There are several previous works in which this fact is also highlighted and addressed [21], [22]. The set of subjects that are included in UJI's Bachelor in Computer Science degree is described in Table IV. It can be observed that only 11 out of 62 subjects include HPC related content. Moreover, three of them are optional (OP), which means that part of the students will conclude their studies having dealt with HPC in less than 13% of the subjects. Besides, the only two subjects of the fourth year that include HPC content, belong to only one of the four available specialities, each with its specific set of 6 concrete subjects.

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	1	0

TABLE IV
UJI'S BACHELOR IN COMPUTER SCIENCE SYLLABUS SET OF SUBJECTS SUMMARY BY YEAR SPECIFYING THEIR TYPE (*Basic Training* - BT, *Compulsory* - C, *Optional* - OP, *Final Degree Project* - FDP), 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 YEAR 2018/2019).

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 classroom [28], detailed and well described manuals such as [19] often turn out to be complex to understand from students perspective. Joining this with the fact that, most of the time, HPC requires knowledge from different fields that have not been previously correlated, we consider that being HPC self-taught is complex for someone that has just finished Bachelor studies (or is still studying).

B. Targeted flaws.

Are students motivated to learn?

Motivation is vital for learning and university students have it, although we (as lecturers) do not always find a way to keep it alive. Regarding Figure 2, which shows the answers to the first question in the IS, more than half of the students attending the course signed up for it because they were interested in learning and felt curious. Appealing to their motivation to learn and their curiosity was our intention when considering how to bring HPC to students, and we believe "hands-on" experience is crucial, as stated by other authors [26], [27].

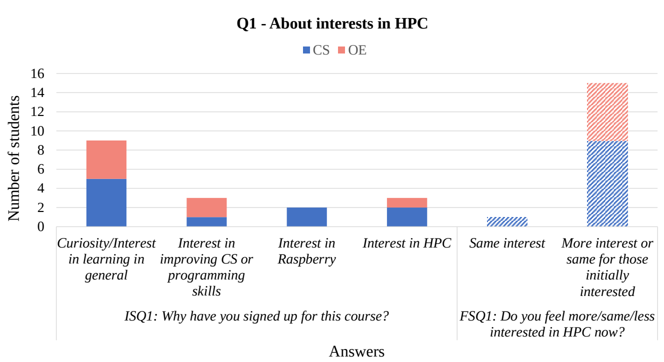

Fig. 2. Initial Survey - Question 1 (ISQ1) and Final Survey - Question 1 (FSQ1) answers categorization, respectively in full colored bars and line-filled bars, including Computer Science (CS) students in blue and Other Engineering (OE) students in red.

There exists a lack of HPC knowledge among Engineering students.

During the last years in which we taught, we observed a lack of knowledge about HPC by 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 Figures 3, 4 and 5, which respectively compare their answers to questions 2, 3, and 4 in the IS and the FS.

It can be observed in Figure 3 that, before conducting the course, only around half of the students considered that HPC presents a clear influence in their day to day. Moreover, Figures 4 and 5 reflect that defining "HPC" or "supercomputer" terms implied associating them to "complex computations" or "optimizations" of the Software or the Hardware.

V. Discussion and Lessons Learnt

Results extracted from the surveys and the course experience itself are summarized and discussed in this section.

HPC interest has increased among students.

Regarding the HPC interest and awareness, Figure 2 shows that only one student affirms to feel the same interest about HPC that he/she had before taking the course, although he/she

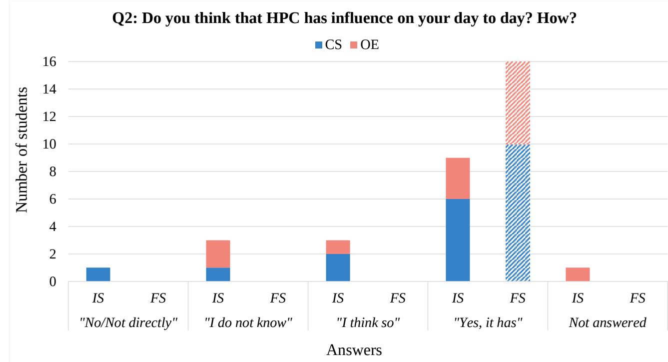

Fig. 3. Initial Survey (IS) and Final Survey (FS) Question 2 answers categorization, respectively in full colored bars and line-filled bars, including Computer Science (CS) students in blue and Other Engineering (OE) students in red.

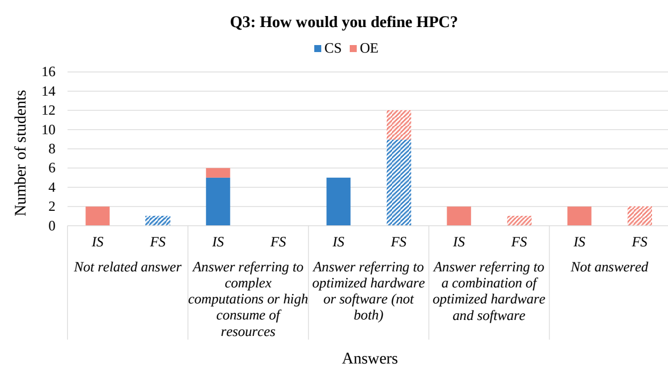

Fig. 4. Initial Survey (IS) and Final Survey (FS) Question 3 answers categorization, respectively in full colored bars and line-filled bars, including Computer Science (CS) students in blue and Other Engineering (OE) students in red.

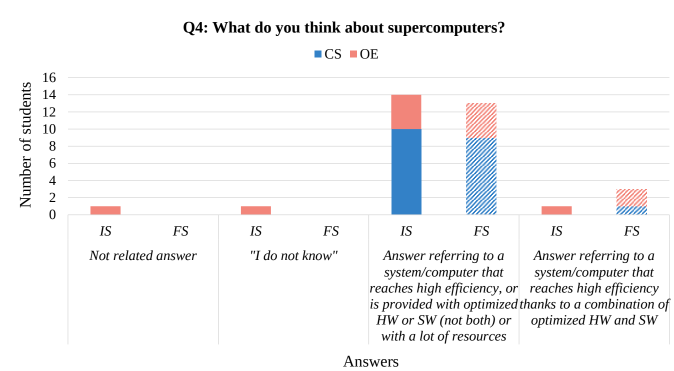

Fig. 5. Initial Survey (IS) and Final Survey (FS) Question 4 answers categorization, , respectively in full colored bars and line-filled bars, including Computer Science (CS) students in blue and Other Engineering (OE) students in red.

was already interested, and Figure 3 evidences that all the students afterward believe that HPC has impact on their day to day (only half of the students had that feeling before the course).

HPC knowledge has increased among students.

HPC knowledge (see Figures 4 and 5) is visibly better after the course. Discarding approximately 20% of the students that provide not related answers in both surveys, Figure 4 reflects that, in the IS, around 35% of the students thought that HPC was equivalent to "complex computations and a high consumption of resources", and around 40% of them somehow established a relationship between HPC and optimization of SW and/or HW. However, after the course, they abandoned the idea of identifying HPC only with "solving very complex calculations using lots of resources" in favor of optimizing available resources, and this last percentage increased to 81%. Even better improvements are observed analyzing Figure 5; 12% of the students were initially not able to define "supercomputer" term in an *appropriate* manner but, after the course, 13 out of 16 related it to optimizations on SW or HW, and the remaining 3 provided definitions exposing the importance of optimizing both SW and HW at the same time.

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 own cluster 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, namely HW experiments such as interconnecting different clusters or cooling temperature consequences, or SW tests like shared-memory vs. distributed memory executions performance analysis.

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

After this first edition of the described course, we consider that a 10-hour program lets the students learn the basics of HPC and motivate them not only to keep their HPC interest and curiosity, but also to encourage them to experiment with their cluster and apply on it their own proposals.

VI. CONCLUSIONS

HPC need and importance is unquestionable and we have realized that there is a lack of related contents in Engineering syllabus, and particularly in Computer Science subjects. In general, students are motivated to learn and this, combined with the fact that a lot of professional positions require HPC knowledge, moved us to propose the course. Combining "hands-on" experience and "on-demand" approach works in favor of creativity and motivation, and we can confirm that students increased their HPC skills and interest.

We have also included this course in a series of CS complimentary courses, at UJI, focused on HPC. This series involves: C language programming [31], parallel distributed and accelerators programming and supercomputer facilities.

VII. FUTURE WORK

We are satisfied with the results that derive from completing the course and will offer subsequent editions of it (we have already obtained foundings for the following edition). 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.

Regarding the motivations to create the course (IV-A), we plan to include related questions in the surveys. In this way, students' opinions about the presented three observations could be analyzed in order to back up our hypotheses, and not exclusively through the bibliography.

ACKNOWLEDGMENT

The researcher Sandra Catalán was supported by Fujitsu under the BSC-Fujitsu joint project: Math Libraries Migration and Optimization; 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. This course was possible thanks to the funding provided by the High Performance Computing and Architectures Research Group (HPC&A) from UJI. The authors also want to thank the anonymous reviewers, whose comments were very helpful in finalizing this paper.

REFERENCES

[1] D. Eadline, "High Performance Computing for Dummies", Springer, 2018.
[2] C. Severance, and K. Dowd, "High Performance Computing", Springer, 2018.
[3] ASC Community, "The Student Supercomputer Challenge Guide", Springer Singapore, 2018.
[4] L. Alvarez, E. Ayguade, and F. Mantovani, "Teaching HPC Systems and Parallel Programming with Small Scale Clusters of Embedded SoCs", Workshop on Education for High Performance Computing (EduHPC), 2018.
[5] V. Holmes, and I. Kureshi, "Developing High Performance Computing Resources for Teaching Cluster and Grid Computing Courses", Procedia Computer Science, Elsevier, vol. 51, pp. 1714–1723, January 2015.
[6] P. López, and E. Baydal, "Teaching high-performance service in a cluster computing course", Journal of Parallel and Distributed Computing, Academic Press, vol. 117, pp. 138–147, July 2018.
[7] A. A. Datti, H. A. Umar, and J. Galadanci, "A Beowulf Cluster for Teaching and Learning", Procedia Computer Science, Elsevier, vol. 70, pp. 62–68, January 2015.
[8] S. Plimpton, "Fast Parallel Algorithms for Short-Range Molecular Dynamics", Journal of Computational Physics, Academic Press, vol. 117, pp. 1–19, March 1995.
[9] T. Sterling, D. J. Becker, D. Savarese, J. E. Dorband, U. A. Ranawake, and C. V. Packer, "Beowulf: A Parallel Workstation For Scientific Computation", Proceedings of the 24th International Conference on Parallel Processing, vol. 1, pp. 11–14, 1995.
[10] Raspberry Pi Model 3B+, Raspberry Pi Foundation, https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/.
[11] OpenMP, https://www.openmp.org/.
[12] MPICH, https://www.mpich.org/.
[13] OpenMPI, https://www.open-mpi.org/.
[14] LINPACK Library, https://www.netlib.org/linpack/.
[15] NFS protocol, http://nfs.sourceforge.net/.

[16] Top 500 List, https://www.top500.org.
[17] Raspbian free Operating System based on Debian optimized for the Raspberry Pi hardware, https://raspbian.org.
[18] W. D. Gropp, E. Lusk, and A. Skjellum, "Using MPI: portable parallel programming with the message-passing interface", MIT Press, vol. 1, 1999.
[19] J. L. Hennessy, and D. A. Patterson, "Computer Architecture: A Quantitative Approach", Morgan Kaufmann Publishers Inc, 2017.
[20] L. Zhou, and G. Shengchao, "Thermal management of ARM SoCs using Linux CPUFreq as cooling device", 2015.
[21] F. Banchelli, and F. Mantovani, "Filling the gap between education and industry: evidence-based methods for introducing undergraduate students to HPC", pp. 41–50, November 2018.
[22] B. Neelima, and J. Li, "Introducing High Performance Computing Concepts into Engineering Undergraduate Curriculum: A Success Story", Proceedings of the Workshop on Education for High-Performance Computing (EduHPC-15), Austin, 2015.
[23] The Council on comptetitiveness, "Benchmarking Industrial Use of High Performance Computing for Innovation", 2008.
[24] B. Obama, "Executive Order - Creating a National Strategic Computing Initiative", The White House, vol. 29, 2015.
[25] European Investment Bank, "Financing the future of supercomputing: How to increase the investments in high performance computing in Europe", 2018.
[26] N. J. Nersessian, "Conceptual change in science and in science education", Synthese, vol. 80, pp. 163–183, July 1989.
[27] M. P. Clough, "Using the laboratory to enhance student learning", Learning science and the science of learning, National Science Teachers Association Washington, DC, pp. 85–94, 2002.
[28] B. Chaudhury, A. Varma, Y. Keswani, Y. Bhatnagar, and S. Parikh, "Let's HPC: A web-based platform to aid parallel, distributed and high performance computing education", Journal of Parallel and Distributed Computing, Academic Press Inc., vol. 118, pp. 213–232, August 2018.
[29] Adams, Joel and Matthews, Suzanne J. and Shoop, Elizabeth and Toth, David and Wolfer, James, "Using Inexpensive Microclusters and Accessible Materials for Cost-Effective Parallel and Distributed Computing Education", The Journal of Computational Science Education, vol. 8, pp. 2–10, Dec. 2017.
[30] Teaching PDC Using the Raspberry Pi, https://csinparallel.org/csinparallel/raspberry_pi.html
[31] A. Castelló, S. Iserte, A. J. Belloch, "Accessible C-programming course from scratch using a MOOC platform without limitations", 4th International Conference on Higher Education Advances (HEAD), Valencia (Spain), Jun. 2018.

APPENDIX

In this appendix, we present a "step by step" list detailing everything that needs to be done to fully complete the course. For each detailed set of steps, it is specified if it needs to be done in all the Raspberry Pi devices or only in the one that acts as the "main one".

This part of the course is fully done by following the slides that are available in the GitHub course repository (see footnote 1).

A. 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.

B. 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.

C. 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.

D. Configure the main node (*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`

E. 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`.

F. 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:`.

G. Install LINPACK

To this point, we let the students follow the LINPACK installation guide, and so they get familiar with following an installation guide by themselves and facing system administrators issues.