H2020-FETHPC-3-2017 - Exascale HPC ecosystem development

EXDCI-2

European eXtreme Data and Computing Initiative - 2

Grant Agreement Number: 800957

D4.8

HPC Outreach report
Final

Version: 1.0
Author(s): François Bodin, Univ. Rennes 1
Laurent Morin, Univ. Rennes 1
Date: 24/02/2019

Project and Deliverable Information Sheet

EXDCI Project	Project Ref. №: 800957	
	Project Title: European eXtreme Data and Computing Initiative - 2	
	Project Web Site: http://www.exdci.eu	
	Deliverable ID: D4.8	
	Deliverable Nature: Report	
	Dissemination Level: PU	Contractual Date of Delivery: 28/02/2020
		Actual Date of Delivery: 24 / 02 / 2020
	EC Project Officer: Evangelia Markidou	

* - The dissemination level are indicated as follows: **PU** – Public, **CO** – Confidential, only for members of the consortium (including the Commission Services) **CL** – Classified, as referred to in Commission Decision 2991/844/EC.

Document Control Sheet

Document	Title: HPC Outreach Report	
	ID: D4.8	
	Version: 1.0	Status: Final
	Available at: http://www.exdci.eu	
	Software Tool: Microsoft Word 2013	
	File(s): D4.8-Deliverable.docx	
Authorship	Written by:	François Bodin, Univ. Rennes 1 Laurent Morin, Univ. Rennes 1
	Contributors:	Alistair Grant, EPCC
	Reviewed by:	Katia Soto, PRACE Jean-Claude André
	Approved by:	MB/TB

Document Status Sheet

Version	Date	Status	Comments
0.1	15/01/2020	Draft	Outline, structure, Into
0.2	12/02/2020	Draft	PMO review
1.0	24/02/2020	Final	Update according to review

Document Keywords

Keywords:	HPC at schools, STEM, primary school, secondary school.

Table of Contents

List of Acronyms and Abbreviations

AAI	Authentication and Authorisation Infrastructure
AI	Artificial Intelligence
AIOTI	The Alliance of Internet of Things Innovation
BDEC	Big Data and Extreme-scale Computing
BDVA	Big Data Value Association
CDN	Content Delivery Network
CoE	Centres of Excellence for Computing Applications
CMOS	Complementary Metal Oxide Semiconductor
CSA	Coordination and Support Action
CPU	Central Processing Unit
D	Deliverable
DNN	Deep Neural Networks
EC	European Commission
ECMWF	European Centre for Medium-range Weather Forecasts
EOSC	European Open Science Cloud
FET	Future and Emerging Technologies
FPGA	Field Programmable Gate Arrays
H2020	Horizon 2020: EC Research and Innovation Programme in Europe
HPC	High Performance Computing
ISV	Independent Software Vendor
KPI	Key-Performance Indicator
ML	Machine Learning
PCM	Phase Change Memory
R&D	Research and Development
R&I	Research and Innovation
SME	Small and Medium Enterprise
SRA	Strategic Research Agenda
TPU	Tensor Processing Unit
TRL	Technology Readiness Level
US	United States
WG	Working Group
WP	Work Package

Executive Summary

The workshop, held the 8 and 9 of July 2019 at EPCC, aims at understanding how career in the STEM/HPC can be encouraged. The workshop gathered 16 workshop attendees provided experiences ranging from primary to university practices. They have been invited thanks to their experience in teaching cross-curricular STEM, programming or promoting computer science. The reflexions provided by the attendees led to the following recommendations:

1. To promote HPC/scientific related careers STEM vocations have to be promoted starting in primary school. All exchanges during the workshop do show that this is possible and well-suited assuming that the teachers are supported by scientists.
2. Students from the age of 16 begin should have a minimum knowledge of writing programs for supercomputers.
3. Teaching programming at school works much better if they can apply it at what they are studying otherwise.
4. Python is popular. Pedagogically interpreted languages avoid dead time that helps the students to focus. It is recommended that from the age of 11 pupils begin to learn Python and that every year gradually increase the level of complexity with the aim of reaching a level of data management or automatic learning and supercomputing before the university level.
5. It is necessary to emphasize about the benefits of mathematics and consequently the computer skills personally and in all careers. Children must be able to explain parents.
6. We must implement the knowledge of programming in all the STEM subjects and evaluate them in the examinations from the age of 10.
7. Nano computer based platform (e.g. Raspberry PI) can provide a very rich support for children to experiment computer science and in particular in parallel programming.
8. Integrating data, AI and IoT related knowledge could be a great support in primary school to develop environment related pupils projects. Collaborations between universities and primary school to set up such initiative.
9. The sequence STEM at primary school, programming in high school, HPC at the university must be carefully staged and coordinated.

1 Introduction

In order to expand Europe's pool of talent, promoting computing, HPC and Data Science to schoolchildren is extremely important as it can influence their future choice of school courses and University degree. Many organisations have programmes to work with local schools but there is neither high-level coordination of these activities nor a common understanding of best practices.

This document reports the testimonies of the face-to-face workshop that has brought together practitioners in HPC & Data Science, together with schoolteachers. It is not intended to provide here a full coverage of the issues. It is a starting point to explore the practices and point of views between the HPC community and the schoolteachers. We have chosen a very pragmatic approach in order to start building from the ground-up accounting that HPC outreach approach are frequently too theoretical and too far from the schoolteachers reality.

The discussed elements are not restricted to HPC but cover STEM (science, technology, engineering, and mathematics) since a restriction to the HPC domain is not relevant while considering children from primary schools. We use depending on the context STEM, programming or HPC.

The document is organised as follow. Section 2 provides a short description of the workshop organization and attendees. Section 3 states the current situation in the three countries. The next section reports the testimonies and point of views collected during the workshop. We conclude with a set of recommendations.

2 Workshop Organisation

The workshop was held the 8 and 9 of July 2019 at EPCC. It was a two-day discursive workshop including presentations on activities and existing work, problems in computer science education and issues with long-term support and education in this area. There were 16 workshop attendees:

- 5 from EPCC
  - Alistair Grant, EPCC
  - Bianca Prodan, EPCC
  - Jane Kennedy, EPCC
  - Darren White, EPCC
  - Ben Morse, EPCC
- 4 from France (2 High Schools, 2 Universities)
  - François Bodin, University of Rennes 1
  - Laurent Morin, University of Rennes 1
  - Jean-Baptiste Faure, Lycée Renée Cassin, Montfort
  - Jean-Louis Jarry, Lycée Renée Cassin, Montfort
- 1 from Spain (University)
  - Joan Verdaguer-Codina, Inst. Menendez y Pelayo
- 6 from Scotland/UK (2 from Primary School, 2 High School, 1 Research Center, 1 University)
  - Jacqueline Campbell, St. Mungo's High School
  - Frances Collins, St. Mary's RC Primary School
  - Darren Grant, SSERC (https://www.sserc.org.uk/)
  - Majorie Kellas, St. Mary's RC Primary School

- Shona McAlpine, Stirling High School
- Jeremy Singer, University of Glasgow

The attendees provided experiences ranging from primary to university practices. They have been invited thanks to their experience in teaching cross-curricular STEM, programming or promoting computer science.

The workshop agenda has been the following:

1) Exchange of experiences and current situation
2) Identifying directions to promote STEM/HPC at schools
3) Demonstration of prototypes used for parallel computing at school (Wee Archie and Archie Frog)

The presentation of the discussion happening during the workshops is presented as testimonies. The testimonies format was chosen because it was not intended here to report the workshop findings as general rules or statements. We had the feeling that many elements are specific to a country and the way schools are organized. It would have been better to cover more countries, however this was not possible budget-wise.

We have not tried to pinpoint these testimonies to a given person as each of these is a synthesis of the discussion that happened around each reported experience.

3 Current situation

The workshop started by exchanging the point of views on practices. The three countries have very different educational systems that are reflecting in the testimonies presented in the remainder of this section. It should be noted that the point of views exposed here are from people already very familiar with programming as a discipline.

3.1 Scotland

[**Testimony 1**] *In a Scottish secondary school Computer Sciences (CS) is taught as a discrete subject. Depending on the school, pupils are exposed to some CS education in S1 and S2 or none at all. Pupils can continue their CS education through the national awards i.e. from levels 3-7. There are also a variety of NPAs (National Progression Award) available in some schools. In my school we teach an NPA in Cyber Security and an NPA in Games Development.*

[**Testimony 2**] *There are a variety of coding languages used in Scottish Schools including Java and Visual Basic. My experience is starting with younger pupils in Scratch or coding the microbits in using MakeCode [https://makecode.microbit.org/]. Once they have learned the basic coding concepts we progress onto using Python. Pupils also learn HTML/CSS for web design and SQL for database. Currently in my school there is no cross-curricular work between CS and other areas of the curriculum.*

3.2 France

[**Testimony 3**] *In France, from the start of the 2017 academic year, algorithmic and Python programming are on the program of the mathematics course in general subjects in lycée (15 to 18 year old). Teaching programming from 15 years is too late, I think it must be much earlier.* ***Students will learn Python if they can apply it at what they are studying***.

[**Testimony 4**] *We teach Python using Jupyter Notebook, which is an Integrated Development Environment (IDE). We start from the age of 15 and before that age students learn Scratch. Raspberry Pi uses Trinket which is another IDE. Trinket has a limitation; it can not execute graphics programs that require matplotlib [https://matplotlib.org/].*

3.3 Spain

[**Testimony 5**] *I use Gedit as editor and Terminal to run Python programs, which all students have on Linkat. Linkat is another layer of Ubuntu developed by the Department of Education of the Generalitat de Catalunya [https://ca.wikipedia.org/wiki/Linkat]. Another advantage of using the Gedit editor and the Terminal is that the students not only learn the basic instructions of Linux also learn that exists another operating system. They are in learning period and Linux is part of their learning process. The common factor among us is that we have beginners in programming.*

Zed Shaw is the author of "Learn Python3 The Hard Way" and his books emphasizes instruction and making things as the best way to get started in many computer science topics. In the book mentioned above and in the paragraph "Warnings for beginners" he wrote: all you need is an editor, a terminal, and Python; in the paragraph "Alternative text editors" he wrote: do not use an IDE while you go through this book, and, in the last paragraph he wrote: you should also not use IDE. It has serious limitations in how it works and isn't a very good piece of software. All you need is a simple text editor, a shell, and Python.

In Spain, my experience is that students who have learned Python before learning the first-degree equation (in Catalonia at 14 year old), they have no problem understanding that x is a variable and that it can have whatever value. At this age they are sponges of knowledge, another issue is that they like programming.

I follow the CAS (Computing at School) guidelines. At Key Stage 3 (https://en.wikipedia.org/wiki/Key_Stage_3) should be the transition using block to use text. https://www.computingatschool.org.uk/

[**Testimony 6**] *Texas Instrument published a report in 1997 titled "The role of calculators in math education" [https://education.ti.com/sites/US/downloads/pdf/therole.pdf]. Twenty years later (2017) the Math Geek Mama [https://mathgeekmama.com/the-great-calculator-debate/] published in the web "The Great Debate: The Role of Calculators in Math Education". In July 2019 the web Latest Bulletins [http://latestbulletins.com/category/education/] published a document titled "What Is The Role Of Calculators In Education?*

All of them find beneficial to use calculators in schools, but the reality is quite the opposite. The use of calculators is very poor, quite exclusively as a mere substitutes of tables of logarithms, tables of trigonometry and to calculate a square root. In sociology, the phrase "resistance to change" is known as the movement of workers from factories and companies against the use of computers in the early 80s. A similar phenomenon occurs in schools, the battle lost to use calculators has as the main cause the resistance from teachers to change, and this resistance exists because many teachers have never learned the functions of a calculator and how to apply them in a classroom.

If it is a huge problem that many students enter university ignoring the powerful functions of their calculator, the same frequently happens with their ignorance of any programming language. The university is not the place where the students have to have the first contact with the programming. For instance, in Catalonia, the engineering schools of the UPC and the Faculty of Physics of the UB teach Python the first year. More than 95% of new students in these universities have null knowledge of the functions of their calculators and of programming as well. Portugal solved partially this problem, the exams to enter the university must be done with a graphical calculator and in the exams there are problems that require to students a good knowledge of the functions of the calculator, so, teachers are obliged to teach with calculator.

The point of view is that we must learn the failure of the past in the use of calculators. For all of us, the lesson is that we cannot spend another twenty years talking about how to implement programming, how to integrate it into the transversal curricula and how to increase the use of programming to solve various tasks in different subjects at secondary level. The politicians must consider the solution of Portugal; they must implement the knowledge of programming in all the STEM subjects and evaluate them in the examinations from the age of 10.

4 Topics of concerns across countries

In this section we report the main exchanges.

4.1 Gender gap

It is generally accepted that girls have better marks in STEM than boys in secondary and high school and we make a question: why it is not reflected in the number of girls accessing to STEM subjects?

[**Testimony 7**] *I talked with teenagers about this question, they said: nowadays we can decided what we want to study, so, if some girls don't want to study STEM is their decision, that's all.*

I search about this problem deeper and in my point of view there are several facts that don't arise, so, I found in Sian Beilock [https://www.apa.org/science/about/psa/2011/09/academic-stress] studies some explanations: family fear about science; the negative influence from many elementary school teachers (they are afraid to mathematics and science), see Figure 3. *I found more information, some girls don't believe in themselves and I explain the case published in the IEEE Spectrum that a teenager won a prize for an app to diagnose retinopathy. She said: "I don't think the problem is a lack of passion. It's more, I don't feel like I'm good enough".*

Many girls found STEM careers too difficult and decided to choose other easier (perceived as) careers.

4.2 Which programming language

Programming language is at the core of teaching STEM/HPC at school. The choice of the language is perceive as a key vector to introduce the computer science concepts.

[**Testimony 8**] *I started to use Python in 2011 because it requires to students less capacity of abstraction than other programming languages and it helps teacher in the classroom management. To manage a teenagers classroom is not easy and a teacher must avoid dead times. Any programming language in order for code to be executable, or usable, needs to be compiled or interpreted. The difference between a compiled and an interpreted language is the steps involved in its execution.*

Python is an interpreted language, this means the source code is being interpreted line by line, executing the line before moving onto the next line. There isn't a need to compile the code in order for it to be executed, consequently there aren't dead times[1]*.*

Compiling the code is an extra step incurred whilst developing your code. Each time you want to run the application, the code needs to be compiled, this is time consuming, during this period, the student is waiting. ***The compiling time is the worse for a teacher, because it is the dead time****.*

4.3 The age when to start

Following the programming language discussion, the match between age of pupils and teaching approach has been considered during the workshop.

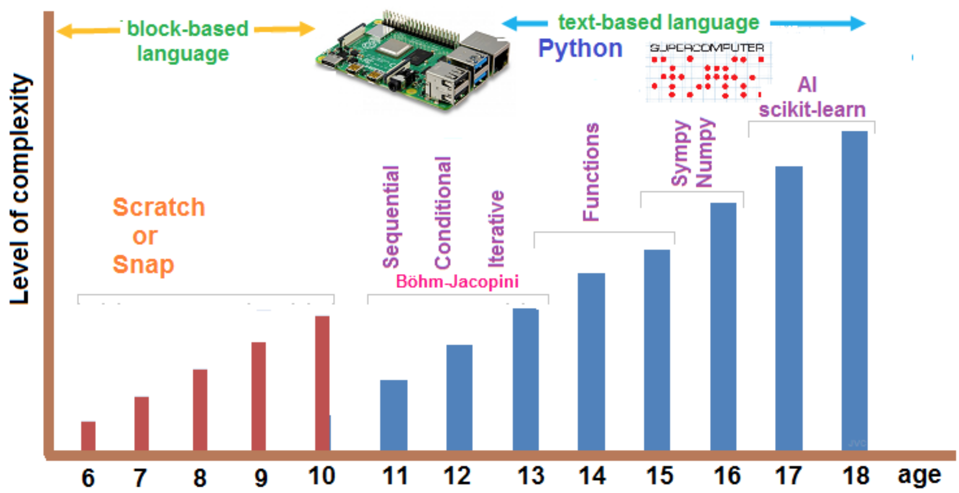

Figure 1 Ages versus programming languages

[1] It should be noted that Python is extremely popular in the programmers community: https://spectrum.ieee.org/computing/software/the-top-programming-languages-2019

From the age of 11, students must start to learn how to write lines of code, understand an algorithm, and gradually introduce the necessary elements to achieve the set objective. As shown in the Figure 1, the student gradually reaches the necessary levels to start working with supercomputing and machine learning.

As in a scaffolding, all these steps are possible if there is a cross curricula among subjects and a shared understanding of the approaches and interests between the school and academic professors.

Learning programming by block-based language is an excellent choice at early ages in primary school and at less than 11 year old, but at the same time is important to show students that any icon is, in fact, an amount of lines of code. Showing text-based language students understand that the icon or block and the code are the same is a good way of teaching programming.

It is recommended that from the age of 11, students begin to learn Python and that every year the level of complexity is gradually increased with the aim of reaching a level of data management or automatic learning and supercomputing before the university level [**General agreement**].

4.4 Education Partners Roles

Figure 2 gives an articulation of the main ecosystem components. The unusual connection here is the one linking universities with primary schools. In the exchange between the participants it turned out the universities are not involved enough with primary school projects. This link is counter-intuitive, it is a consequence of the introduction of "data" (e.g. from sensors) in many scientific fields. Primary school projects frequently requires broad competencies that are available mainly at the universities; For instance, a project about the environment may require Internet of Things devices or data that are available to research teams and can be, set in the proper form, help the primary school to define projects that include STEM / HPC knowledge (i.e. the use of statistical/numerical models).

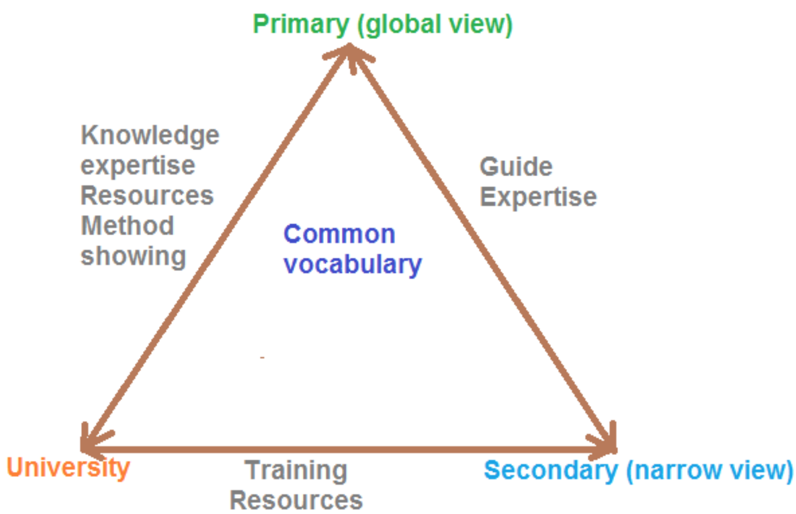

Figure 2 Relationships between partners in Education.

[**Testimony 9**] *Currently, in my experience there is no relationship between secondary schools and universities. We supply a good number of pupils to Computing Science (and other STEM* *subject) undergraduate courses each year but we do not know what they would like in the way of prior education/skills. There is a professor at Napier University who publicly condemns our subject on Twitter calling it irrelevant and stating that we teach it in a 'dull and boring' way. However, we have not had any input from University staff on how to solve this issue or if in fact this is a widely held view.*

In my experience, we have a relationship with primary schools. We have visited primary schools to deliver coding workshops. We have also supported primary colleagues to learn digital applications although no specifically Computing Science applications.

In my opinion it would be helpful to have a relationship between the three sectors. The relationship requires working in both directions i.e. from tertiary down to primary and vice-versa. There requires being a progression in learning (helped by Experiences and Outcomes) and a sharing of resources and expertise by subject specialists. Ultimately, we need to know where Computing Science education leads; what knowledge, skills and expertise is required and then work out how to provide that progression in an engaging way.

Many parents have a bad memory of their school time and never understood mathematics, she continues saying that they don't understand 1+1 because they didn't it in they childhood. It is necessary to emphasize about the benefits of mathematics and consequently the computer skills personally and in all careers. Children must be able to explain parents. A level of knowledge is necessary in mathematics in any field and learning programming helps them.

4.5 Influence on STEM/HPC Careers

Figure 3 summarizes the influences on STEM careers. Building a strong connection is important. In particular, if the families do not grasp the interest of STEM career (projects at primary school involving universities are a must have tool for this), this is an important roadblock.

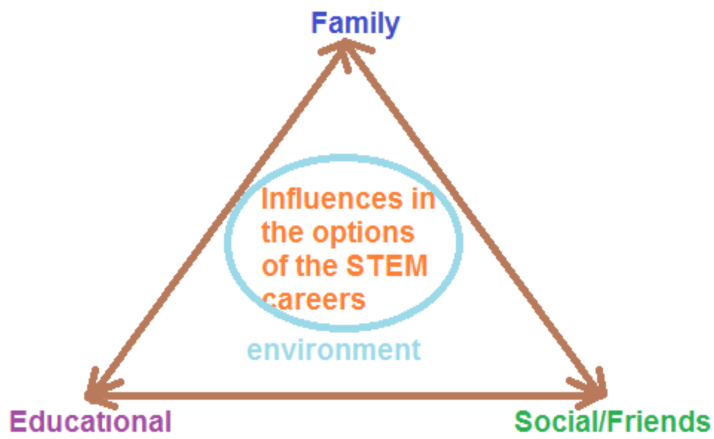

Figure 3 Influences on STEM Career Choice.

[**Testimony 10**] *There are two concerns; firstly, falling uptake in STEM areas particularly Computing Science (in our school this is not an issue but it is nationally). Secondly, gender imbalance.*

Traditionally, bright pupils and their parents choose to continue with three sciences i.e. Biology, Chemistry and Physics. I believe that if parents and pupils were aware of the importance of programming in each of these areas they would be more inclined to choose to study it. There are careers today that did not exist when parents were at school that may be a more attractive/suitable option. I think education/information is key to addressing this issue. The gender issue is a difficult one which again may be helped by better information/education. However, there is a need for CS to lose its reputation as a geeky subject particularly around computer gaming.

4.6 Cross-Curricular Projects

Cross-curricular projects have been perceived as an important mean to connect HPC/STEM activities to primary and secondary schools.

[**Testimony 11**] *I cannot think of specific projects but I liked the look of the project presented during the workshop that showed using coding in Biology to work with DNA data. I think using data sets is a good idea with code being used to manipulate the data.*

The benefits would be putting coding into a context. Pupils often lose sight of why they are writing a program - what is its purpose? As we learned at the workshop from Bianca and others first exposure to coding determines uptake/engagement. Therefore, in my opinion, learning programming in a context where it has a specific purpose would improve both engagement and ultimately uptake.

The drawback is the workload and the need to change attitudes in order to design projects across curricular areas that have traditionally been discrete.

Primary teachers say that they work with a wide spectrum of subjects assembled together without pressure of exams and carried by one or two teachers, on the contrary. Secondary teachers are one teacher for each subject, have a tight spectrum and students are evaluated by exams. So, the key question is how to include computer science in this tight spectrum. Or how to manage cross curricula.

I explain the example of Solar system; at first course of secondary level in Catalonia, age 13, students learn astronomical unit (AU) and km and how to convert between them using Google. So, a technology context is: the Voyager I reoriented the antenna to Earth and the time to receive the signal was 19h 35', giving the formula $x=v*t$, searching on Google the speed of light (v) they must write a program that find the distance (x) of the artificial satellite in AU and km. It is a sequential program for beginners.

[Example] Projects have been carried out between EPCC and St Mary's Primary School over several years which explore cross-curricular projects including computer programming, computers supporting science and development of new materials for introducing computers in an applied fashion to primary age children ranging from 9-12. One of these projects obtained support from the Royal Society to obtain equipment and materials. These included Raspberry Pi boards and Micro-bits. Using these and other electronics, explorations of physics and the human body were carried out. This involved pupils designing sensors and dealing with data interpretation.

4.7 To be in the trench

It is important to not map university pedagogical approaches onto schools. Teacher's methods are quite dissimilar. Decisions to embrace a STEM career can happen very young; it is sensitive to the family and friend perception.

[**Testimony 12**] *It is not the same to teach primary students, than secondary students, than university students.*

A primary class is usually easy to manage; but it is an important period to define options for some students. Some open the brain to STEM careers, for example: Sandra B., who decided to be an engineer at age 12 when she visited a telescope facility, or Sara I. who decided to be an astronomer at age 8 after with her classmates worked on the solar system. Both are scientists involved in the EHT project [https://eventhorizontelescope.org/].

In the Figure 3 *is shown how three different factors that create a personal environment have an influence in the decision making to find a STEM career.*

The management of a secondary class is other level, especially when the mood in the teenager period is like a roller coaster. This period of personal changes affects them in their studies, personal decisions that involve their future way of life and influence in their vocational decision to STEM. Here also STEM vocations are defined. However, if programming is not taught, students have fewer choices to choose from; and when the decision making time arrives they find careers requiring programming knowledge as an personal obstacle saying "it is to much difficult" and choose a non STEM career or a soft STEM careers that they believe will not need technology knowledge.

In Catalonia for many years it has been explained by university educational managers that the change of name of "faculty of informatics" to "computer engineering" has been the cause of the loss of female students in computing. I think that Ted Nelson [https://fr.wikipedia.org/wiki/Ted_Nelson] is right when he says that the breakup occurred when computers were associated with video games in the 70s. The movie "War Games" [https://en.wikipedia.org/wiki/WarGames] shows this turning point. It has been necessary over two generations when girls to start considering computer science with a wide spectrum and not a tool that only serves to play video games.

4.8 Encouraging supercomputing: The role of RPi

The use of Raspberry PI technology is considered as many as a convincing tool, while being cheap, to create pedagogical experimental platform. Especially, it is possible to help children to understand parallel computing using simple parallel algorithms (e.g. drawing the Batman logo using a Monte-Carlo parallel method).

[Testimony 13] *The Raspberry Pi board is a great embedded system, cost efficient, that helps educational system. It is useful as small home computer as well as industrial computer.*

It is clear that prior to work in a supercomputer, people must have a good knowledge in programming language and a basic knowledge in Linux as OS. Supercomputing has new rules inside the programming world.

It is know that with several RPi is possible to have a small Beowulf [https://en.wikipedia.org/wiki/Beowulf_cluster] which is in a small scale an approximation of a supercomputer.

The example done in this workshop by Rennes-1 IRISA University to be applied in a classroom is a clear example that a training period of supercomputing to students from upper secondary level (from the age of 15) is possible and also desirable, in order to find hidden talent among students (See Section 4.8.2).

Also local or regional maker faire or code clubs could be a good place to teach and encourage young people in the supercomputing programming world.

4.8.1 Wee Archie

Wee Archie[2] was created by EPCC at the University of Edinburgh to as part of the ARCHER supercomputing service outreach and education programme. The aim of Wee Archie was to introduce the concept of supercomputers and parallelism in a tangible and accessible way is to wide and non-traditional audiences. Core HPC elements are reproduced at a smaller scale to illustrate different system elements while providing a real parallel programming environment.

Wee Archie was designed to fulfil the following criteria:

- Transportability - the design is robust and portable requiring only a network and power cable to be attached.

- Functionality - Wee Archie will demonstrate parallel concepts through running parallel simulation applications.

- Visibility – complex systems are often hidden from the general user, Wee Archie is designed to expose the network and component connections and hardware safely.

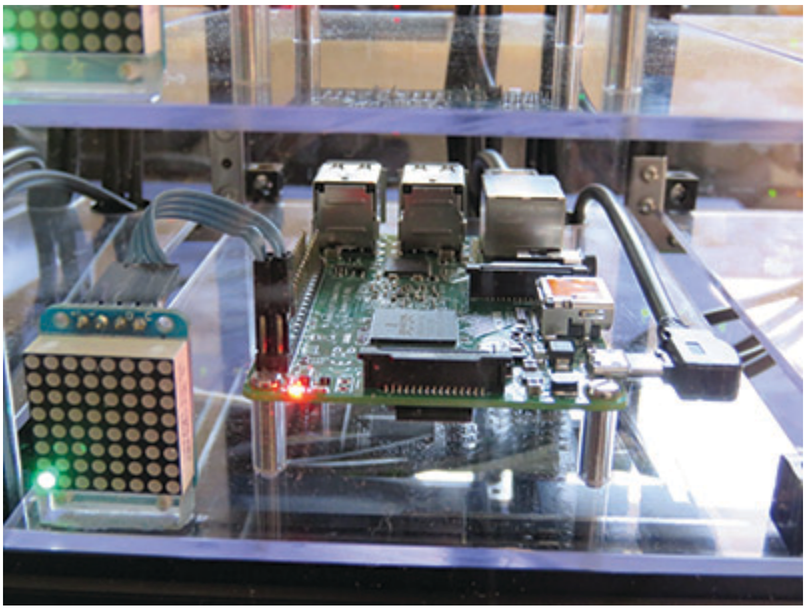

Figure 4 A computation node – board and matrix indicator

2 https://www.epcc.ed.ac.uk/discover-and-learn/resources-and-activities/what-is-a-supercomputer/wee-archie

Wee Archie consists of 18 Raspberry Pi (link is external) 2 Model B boards with three network switches housed in a custom design casing (See Figure 4). Sixteen Raspberry Pi boards are for dedicated to computation; the last two form a control, file management and monitoring system. Each board has an associated LED matrix used to show live operational metrics, including CPU Core load, memory usage, network traffic and temperature.

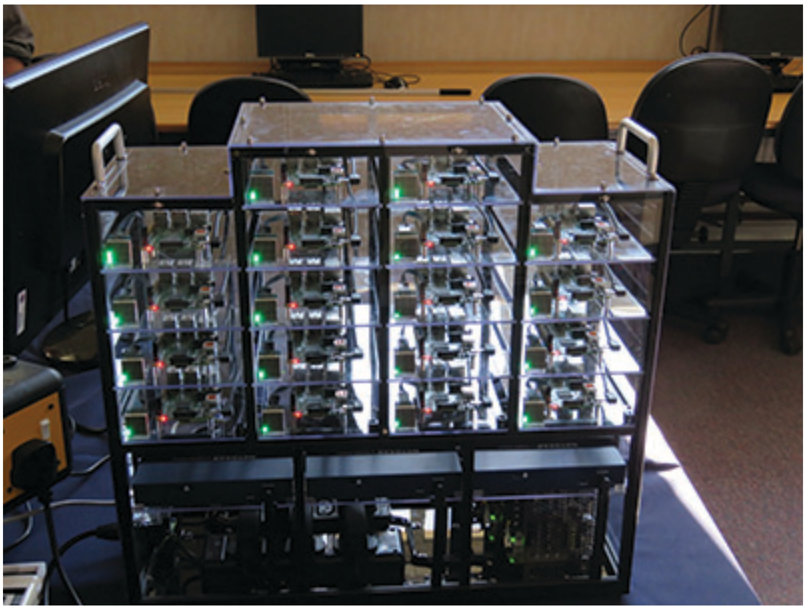

Figure 4 Wee Archie RPI cluster.

Wee Archie has been successfully used in major science engagement events, such Edinburgh Science Festival, The Big Bang Fair and New Scientist Live and has been taken to community groups, science centres and schools in Scotland and the United Kingdom. The demand for the use of Wee Archie led to a second unit being commissioned. Both units have travelled to conferences in America, Europe and China. The units are undergoing redevelopment in early 2020 to upgrade the hardware and refine the software demonstrations.

4.8.2 Archie Frog Prototype

The Archie Frog project was directly inspired by the Wee Archie initiative but targets a different objective: provide an easy-to use and compact nano super computer for educational purpose. Like the Archie Wee machine, the prototype is made of a grid of Raspberry Pi linked by an Ethernet switch and charged by the HPC programming tools (mainly MPI). Beyond this base, three main criteria drove the design of the prototype.

The concept is to be compact enough to be transportable easily from one room to another with only one electric plug.

The core hardware system is a custom PCB able to hold 9 Raspberry Pi on one side, and 9 displays on the other side, each display associated to one cluster node. This design enables the monitoring of the node, or the restitution of graphical result per node, and finaly show how a parallel tasks work (see Figure 5) A control node based on an Intel NUC (a mini PC), makes the interface between the users and the platform, and the last version includes a external wifi access-point able to manage several tens of users.

Figure 5 The Archie Frog prototype

The system must offer a realistic view of the software stack of an HPC cluster.

The prototype is equipped by the software systems typically deployed to manage and operate parallel computing on a real-life cluster. It includes the Slurm job manager, a distributed file system, and the traditional OpenMP/MPI programming models. The control node based on a mini-PC manages the shared file-system, the long-term storage, a web interface, the job scheduling, and includes all features needed to mitigate then cyber-security risks when deployed on the field.

The system shall require the smallest expertise possible, be administration free, and easy to deploy and managed for primary teachers.

Over the supercomputer, a simple web interface is accessible via an open-wifi. A full classroom can get connected to the machine and start a course using a Python notebook (See Figure 6). This interface includes an administration section for creating sessions, managing the cluster status, bring and control an Internet access (firewall limited to a set of services, logged, and authenticated), and follow the status of on-going work. At the end of a session, each user gets its session backed-up for a later access, and the system is cleaned for a new course.

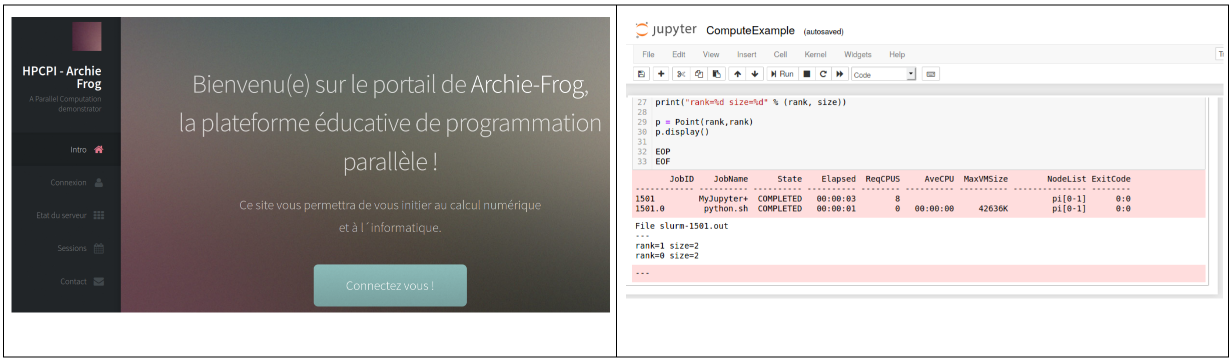

Figure 6 Archie Frog Jupyter Interface

The Archie Frog has yet been used in several courses, for math with secondary school students, and for teaching parallelism in a master module at the University of Rennes 1.

4.9 From now to future: the Internet of Things (IoT)

[Testimony 14] *It was in the IT Summit in Berlin on November 2015 when Dr. Struth, now Bosch former member of board of management, said: "Industry 4.0 needs an education offensive"*[3].

He requires an education offensive for the connected world considering that a broad-based education offensive aimed at children could help in overcoming this challenge, and adding; "we have to lay the groundwork for confidently navigating the digital world at a young age. Young people have to be capable of doing more than just using the apps on their smartphones. They should also know a programming language, because that's the only tool that will allow them to make their ideas reality".

Also, in the Bosch document you can read: "to be able to teach these fundamental skills, schools and teachers need to be equipped with the necessary know-how and the right technical infrastructure".

We must have in mind that we are preparing students to find a job for the next decade to have competitive European factories as Dr. Struth claims.

5 Conclusion

All testimonies during the workshop have pointed out that programming is a key knowledge to better teach and promote STEM. The consensus build at the workshop allows us to draw a few recommendations:

[3] http://www.automotiveworld.com/news-releases/bosch-board-management-member-struth-summit-industry-4-0-needs-education-offensive/

1. To promote HPC/scientific related careers STEM vocations have to be promoted starting in primary school. All exchanges during the workshop do show that this is possible and well-suited assuming that the teachers are supported by scientists.
2. Students from the age of 16 begin should have a minimum knowledge of writing programs for supercomputers.
3. Teaching programming at school works much better if they can apply it at what they are studying otherwise.
4. Python is popular. Pedagogically interpreted languages avoid dead time that helps the students to focus. It is recommended that from the age of 11 pupils begin to learn Python and that every year gradually increase the level of complexity with the aim of reaching a level of data management or automatic learning and supercomputing before the university level.
5. It is necessary to emphasize about the benefits of mathematics and consequently the computer skills personally and in all careers. Children must be able to explain parents.
6. We must implement the knowledge of programming in all the STEM subjects and evaluate them in the examinations from the age of 10.
7. Nano computer based platform (e.g. Raspberry PI) can provide a very rich support for children to experiment computer science and in particular in parallel programming.
8. Integrating data, AI and IoT related knowledge could be a great support in primary school to develop environment related pupils projects. Collaborations between universities and primary school to set up such initiative.
9. The sequence STEM at primary school, programming in high school, HPC at the university must be carefully staged and coordinated.

While it would have been desirable to get opinions from practitioners from more countries we believe that the recommendations are general enough to apply to many European schools. However, each school curriculum being specific, the implementation needs to be localized to ensure success.