Attracting HPC-talent: Ideas & Experiences

Anne C. Elster

Dept. of of Computer Science
Norwegian University of Science and Technology
Trondheim, Norway
elster@ntnu.no / anne.elster@gmail.com

***Abstract*—Since the world was forced into parallelism in the mid 2000s as we hit the Power Wall, Parallel and Distributed Computing (PDC) and High Performance Computing (HPC) in particular, is no longer a specialty seen at the US government labs and weather forecasting centers, but is now everywhere from embedded systems in smartphones that typically have one or more multicore processor and 3 or more GPUs, to laptops, workstations, large server and server farms to supercomputers. This has created a huge demand for technical talent with HPC-skils. Big Data, AI and associated Data Analytics has put even higher pressure on the demand.**

In this short-paper/lightning talk, I will highlight some of my personal experiences in attracting students into my parallel computing class, my experience with advising 80+ graduate students (most of them master students), and current efforts and ideas for introducing HPC to first-year students. This includes why I believe in teaching MPI before OpenMP, threading or CUDA, why optimizing code/performance engineering is so important, the importance of group meeting and team building, and how and why parallel computing concepts should also be taught in the very first introduction to computers classed.

Finally, I will believe we need to have good mentors early in the curriculum and do out-reach to not only high school students, but also grade schools and middle schools. (*Abstract*)

Keywords—HPC, teaching, methods, ideas, *(key words)*

I. Introduction

Since we hit the Power Wall around 2005, we have not been able to get much more performance out of a single processor core by just increasing the clock speed. Instead, we have since had to use vectorization and/or parallelism in systems from embedded chips for smart phones to workstations and large compute clusters, in order to meet computational demands. As a result, the demand for people with HPC skills is just growing, and it has become a real concern how we meet this demand, both in attracting people to the field as well as how do we best train and retain them. As the systems become more heterogenous and complex, the complexity of these systems also grows. As a result, we need better ideas on how to train enough people to build the needed complex tools that tailor and target applications to these developing platforms that are not even necessarily targeting HPC.

The author would like to thank the Excited Project for their recent mini-grant support for developing my Parallel Computing course.
https://www.ntnu.edu/excited

II. Teaching Paralell Computing

A. *Why start with MPI?*

When I was part of the MPI 1 & 2 Committees in the early-mid 1990s, we had no ideas that MPI would still be in use today. The MPI Forum was modelled after the HPF Forum, which people barely have heard of now. The beauty (and challenge) of MPI is that it explicitly forces the user to think about data locality – where is the data and were will it be in the next simulation step. If a process does not have the required data, one has to use an MPI call to get it via MPI_Send, MP_Bcast or several other more targeted and optimized calls. The key here, however, is that the user is forced to think about how the data is distributed among the processes. I always tell my students to remember that the 3 main challenges in HPC is much like in real estate. It is all about location!

1) ***Location*** *– as in feeding the registers*
2) ***Location*** *– as in bringing compute to data to avoid time and energy associated with data transfers*
3) ***Location*** *– as in picking the right algorithm that avaoid data locality issues.*

Data centers thus often see that MPI program scale a lot better than OpenMP programs for cases where more than, say, 8-16 processor are used. Once the students understand MPI, it is a lot easier to teach them about code optimization, as well as efficient OpenMP, Pthread and CUDA programming. MPI is also useful for scaling up codes across larger systems in conjunction with these other environments, but that is not my main motivation for teaching MPI first.

B. *The importance of assignments*

I always tell my students that programming is much like mathematics: You do not become a good mathematician by just reading a book. You have to work on problems. It is the same with programming and parallel programming: You will not get good at it by just reading texts and manuals -- you need to struggle and actually do the programming. However, it is very rewarding when your program works.

To get all the students up to about the same level, I start the course with a crash-course in C. In know a lot of colleagues prefer C++, but for people who really care about performance and will go towards cutting edge architectures, including the embedded systems, C skills are very valuable. Trondheim; Norway, we were are located have several companies such as ARM, Microchip and Nordic, that all praise us for teaching them C, compilers and parallel programming. When one talks to these companies once quickly gets support for teaching C.

As far as teaching CUDA rather than the now more fading OpenCL or upcoming OpenAcc, my reason for using CUDA is tjhat it has a lot of nice tools around it, and one cannot ignore how dominant Nvidia CPUs still are.

Finally, we were given resources to not only record lectures, but edit them and have a student assistant record some assignment-related videos. We are still evaluating this part of our efforts.

III. Advising and Mentoring Graduate Students

Most western countries struggle with attracting enough local students into not only their PhD studiers, but getting them through their Bachelor degrees, little less Masters or PhDs.

Norwegian Univ. of Science and Technology (a.k.a. NTNU), sprung out from Norwegian Institute of Technology which originally had a 4.5 year "diploma engineer" program that grew into our current 5-year direct to Master in Technology (MTech) program I "datateknologi", where there is no option of leaving with a Bachelor after 3 years. We do also offer a 3-year Bachelors in Informatics degree and a 2-year Master degree in informatics (less math requirements) as a result of the merger in 1996 with another college, and a certain number ofn students with BSc degrees that have taken enough mathematics and have strong records, are let in the 4th year into the MTech program, and also server as filling up slots from the students that drop out.

As a result, we produce more master students that Stanford and Univ. of California at Berkey combined. However, our resources are a lot more limited, so several of the most active professors will advise more than 5 master students per year, and often with a co-supervisor from industry.

PhD students are harder to come by, as our attractive students are heavily recruited by industry. For example, one of my master students that just started his 5th year, has already signed a job contract (in early September 2019) for a job starting after he graduates in August 2020.

On the brighter side, this year is the first time I have had 3 of the currently 12 master students I am advising interested in our integrated PhD program! In this program, students do their final thesis year over two years while taking PhD course and starting looking into their thesis topic. They also get ½ a PhD stipend, which at NTNU is quite generous as PhD students are considered temporary employees who should be compensated adequately, although not as much as currently is the case in industry.

So how did I and others suddenly recruit so well? We had 9 applicants for 4 slots, we normally have a struggle filling even 1 slot per year with a qualified Norwegian candidate. It is still har to be certain, but the Department did a lot more advertising this year had a Catch-IDI event where 3rd and 4th year students were invited to research presentation and encouraged to apply if interested. I also think once one strong student wanted to apply, so then did his/her peer. At least I believe that is the case of the 3 students I found and the 3 that are from our AI group. I am still not sure if all will go through with it. However, knowing how attractive these students are to industry and how hard companies try to get them before finishing, I know it will be important to admit preferably all 3 so that the can support each other throughout their studies. Like the US, most of the other PhD students are not Norway born…

A. *MITx MicroMasters program – an idea for HPC?*

MIT has developed an online MITx MicroMasters Program in Data, Economics, and Development Policy (DEDP) to "equip learners with the practical skills and theoretical knowledge to tackle some of the most pressing challenges facing developing countries and the world's poor. Through a series of five online courses and in-person exams, learners will ..". See [4].

.

IV. First Year student & Final thoughts

During the last few years I have realized it is very hard for me to recruit female master students because they are not even taking my parallel programming class. In addition, I found that more women than men drop out during the 2 first years. I also found out that sine we are so few women, and I am the only female native Norwegian full professor in our department, and the Department was desperately looking for faculty members to help with our CS introduction course, I decided to step up. I am hence part of the teaching team for that course this semester. We have two courses, one for the CS and other IT students with about 900 students that is run in two parallells, and the other which has about 1800 students and is run in 4 parallels. I am teaching the "theory" part of the former. I am now hence working out ideas for adding some very basic parallelism into the final lectures. What I end up doing and how it will go, I will be able to report at SC19. Stay tuned!

Also, I have seen how important it was for my own daughter to be exposed to IT and Scratch already as an 8-year old. We also need to start recruiting future HPCers a lot earlier early than at the high school or college level!

AcknowledgmentS

I would like to thank the hundreds, now more than thousands of students that have taken my classes, my 80+ graduate students that chose me as their advisor, and my current and former PhD and Post Docs, for their inspiration, enthusiasm and feedback. I am quite sure I have learned a lot more from you than I have been able to teach you. You are the main reason I love academia.

References

[1] Parallel Computing course at NTNU taught by the author https://www.ntnu.edu/studies/courses/TDT4200#tab=omEmnet

[2] Introduction couse at NTNU for comptuer science and realted IT studies, https://www.ntnu.no/studier/emner/TDT4109/2019/1#tab=omEmnet

[3] Author´s research group at NTNU, with 12+ master students: https://www.ntnu.edu/idi/hpc-lab/people

[4] MITx MicroMaster on-line Program that can be used as part of a Master degree program: https://micromasters.mit.edu/dedp/

[5] The Excited Project: https://www.ntnu.edu/excited