Grainer IT Governance Research Working Group, 2022-2023

Report and Recommendations

Members:

Paris Smaragdis (paris@illinois.edu, CS – Chair)
Mark Smylie Hart (mshart@illinois.edu, Engineering IT – Ex Officio)
Mark Anastasio (maa@illinois.edu, BioE)
Lawrence Angrave (angrave@illinois.edu, CS)
Mattia Gazzola (mgazzola@illinois.edu, Aero)
Lavanya Marla (lavanyam@illinois.edu, ISE)
George Chacko (chackoge@illinois.edu, CS/GCOE)

Summary

The group help a number of meetings during the year, and also discussed multiple items via email and 1:1 discussions. The group meetings focused on the following items:

- Introductory meeting and schedule for the rest of the year
- Discussion on research support and steps towards effective computation startup packages
- Joint meeting with the Education Governance Working Group
- Discussion and refinement of a proposal for GCoE's 6th round of investment in ICCP

The main pain points that were identified during our meetings were:

- (seemingly) Lacking support for GPU computing on campus
  - In number of GPUs
  - In software flexibility
  - In obtaining rapid GPU access when in a pinch
  - In research-oriented IT support
- Lack of enticing options for computational resources in startup packages
- Difficulty in readily obtaining usage pattern data to guide future investments
- Delays in ordering and setting up equipment
- Bridging the gap between education and research

Some improvements achieved over the last year:

- Simplified the process to gain compute time on the Delta system within ACCESS
- Worked towards a computational startup options consulting service for department heads and prospective faculty
- Made progress on speeding up ordering and installing new hardware
- Tighter collaboration between EngrIT and NCSA on research computing operations
- Steps towards a unified file system that will help users use more systems with less hassle

Computational resources for research on campus

A main pain point that has been constant over the last few years is the availability of GPU computing on campus. Despite our campus' leadership in computing infrastructure, and the ample availability of such resources locally, we have not been as successful as other top-engineering schools in completing high-visibility large experiments.

There is no shortage of news articles from MIT, Berkeley, and Stanford, having trained large AI language models, computer vision models, robotics models, or speech recognition systems that necessitated large numbers of GPUs over long windows of time. As has been the case over the last few years, we see that massive data sets used to train massive AI models result in breakthrough performance over all disciplines (from medicine to language processing). However, initiating such endeavors on our campus has been difficult. Throughout our discussions we have identified a couple of reasons, mainly ease of access and usability.

Access

There are a number of systems on campus which can be used for large high-visibility projects. However, getting access to these systems has is not straightforward which can often deter potential users. For example, the Delta system had proposal calls every six months which is an unreasonably long window to wait for an experiment. Other systems used a variety of methods; some fast, some slow, but certainly all different adding unnecessary overhead. The situation is improving (the Delta allocation process was revised as a result of our discussions), but there is still work to do on many of the other systems on campus (each having a different sign up process).

***Recommendation**:* Simplifying and unifying compute system access can encourage more use

Usability

Having a number of systems on campus means that users need to learn how to use each one, and potentially reimplement their experiments. It is quite common to see researchers stick to their own smaller compute clusters given the considerable overhead of porting their code to a new system. The group discussed the need for a more uniform experience and the need for more research IT that would not just address infrastructure issues, but would help researchers with porting and optimizing their computational experiments. An additional issue that was identified was the lack of a unified file system that would allow easy access to data sets from any system on campus. That would not only minimize duplication of data storage over multiple systems and users, but would also help simplify transitioning between systems. Finally, the computational model that is supported by most systems is currently inadequate for a lot of modern uses. Systems on campus use a batch-based model (a legacy of HPC computing), which is not suitable for many modern workflows that require increased interactivity and faster turnover of requests.

Recommendations: i) Move towards a unified file system to simplify system usage, ii) Maintain a campus-wide curated collection of read-only data sets to minimize storage duplication, iii) Establish a research support group that can help improve system utilization and porting pains, and iv) Provide additional computational interfaces suited for modern research workflows.

The issues above probably create a chicken-and-egg problem. Feedback from faculty seems to imply that there are not enough GPU resources on campus, whereas feedback from NCSA implies that utilization is not up to full capacity. This is likely due to the fact that many faculty prefer the simplicity of using their own smaller systems over dealing with the hassle of a shared resource.

Computational Startups

Our group devoted a significant amount of time discussing incoming faculty startup allocations. This is an important subject, since a lot of engineering nowadays is computationally-driven and providing the necessary startup infrastructure is key to recruiting faculty. We identified a couple of problems on this front, which are probably easy to solve with the right technical support.

Lots of options, little guidance

A key problem with computational startups is the number of options. At the moment incoming faculty can be i) offered resources to build their own system, ii) given resources on a shared system, or iii) given resources on commercial cloud systems. Depending on the work being done, either of these options can be a good choice. However, it is clear that the negotiating parties do not always have a full picture to help them make the right choices, resulting in undue stress and potentially poor choices. Based on these discussions, EngrIT has now developed a consulting service for both dept. heads and prospective faculty, presenting them with the available options for computational startup resources on campus, and their respective pros and cons.

Recommendations: This service should become an integral part of incoming faculty negotiations that rely on computational resources (vs. the current option of being an optional call). It should also be used by EngrIT/NCSA to gather data about the ever-changing needs of our faculty body. Doing so will help us respond faster to emerging computational needs and be better equipped to discuss future requests. EngrIT/NCSA could develop a living document outlining the options and their differences, at a level where faculty can get the high-level picture and make the best choice based on their needs.

The need for research engineers

Faculty that opt for building their own computational resources are often getting in more trouble than they expected. Building and maintaining such systems is not straightforward, and although the flexibility of having one's own system is great, it is often offset by wrong design choices and maintenance needs. Our group discussed the difficulties that incoming faculty might face on this front, and agreed that more support from EngrIT would be help helpful. At the moment, many group systems are managed by grad students, which is not a good idea in many ways. It would be natural to have EngrIT help with this process, however this will likely stretch their obligations even more. This problem connects to the recommendation in the previous section, discussing the need for research IT engineers that can help optimize and streamline experiments. Such engineers would be able to provide consultations on what systems would work best for different problems, and help maintain group-owned computational systems.

***Recommendation*:** Consider the creation of research engineers that can help groups to optimally deploy and use their own systems (in addition to the roles mentioned in the previous section).

The gray area between research and education

During our joint meeting with the Education Governance Working Group we discussed a number of problems that seem to be common across research and education computational needs. Perhaps an interesting problem that we are facing is bridging the continuum between education and research.

As we all know, quite often research explorations start from class projects. Many of our students write conference papers based on class final projects, or get started with their research from homework assignments. This creates a usage pattern that is hard to categorize as being either research or teaching and has the potential of falling between the cracks and not being addressed by either of the working groups. We feel that it is important for students to be able to have the resources that will help them make that transition from learning to research.

A different model

Currently, the dominant computational model on campus is using queue-based systems. Usually (potentially after a proposal stage), an experiment is setup and is launched on a system and when enough resources are available it runs in due course. This process requires deliberate planning and justification, and unfortunately some waiting. One could argue that experienced faculty can play this "long game", but this is not a model that suits students who are experimenting with lots of (potentially dead-end) ideas, trying to find something that "sticks". In both the research and education group, we discussed the need for readily providing our students with the latitude to work like that. At the moment we do not have a good model, and having students compete for time on the ICCP or Delta is cumbersome and probably disruptive to research users.

Two usage models that instead work well for this kind of work, are tools like Google's Colab, and Amazon's AWS. They both provide a reasonably powerful computational shared resource for students to experiment with; but do so in a way that doesn't tie up resources the same way we currently do with our queue-based systems. We have started some discussions with NCSA on potentially using systems like Radiant to provide such a service, but at the moment there are not enough GPUs on that system to even start being useful to students. Any solution on that front (whether tin-house or commercial) will likely cost money, but if implemented correctly it will result in a much more shareable resource that is better utilized and would be applicable for both research and education.

Recommendations: The space between education and research should not be neglected, we need the right computational resources for it. We can certainly try to outsource this, for example having departments (or GCoE) purchase Google Colab Pro licenses for all students[1], or broker a deal with AWS to provide such resources. Or we can find ways to build our own system. A potential way to scale this as an educational tool would be to, e.g. have GCoE to purchase a mid-range GPU (~$1,300) for each N incoming students (N=1,2,3,4?) to be hosted in a system like Radiant. That can help build a large enough (self-replenishing) resource for students, and will likely provide more than enough resources for homework and pre-research experimenting. Not being explicitly tied to research will help ensure no resource competition.

---

[1] *Just kidding, we'll never do that with our unique approach to FERPA enforcement!*

Data, Data, Data!

Throughout our discussions this year it was quite apparent that we do not have enough raw data to provide optimal recommendations, or even quantify how big perceived problems are. A good example of this was the deliberation on a proposal for GCoE's 6th round of investment in ICCP. After the initial round of comments, Mark Smylie Hart took some time to gather and share data on CPU vs. GPU usage patterns that was very helpful in (re)shaping our feedback. Having easy access to such data is crucial when it comes to making recommendations on investments and future directions of our computational infrastructure for research. Similarly, a lot of the usage patterns on Delta, ICCP, and other systems is still unclear to many of us, and even when we see the data we often can interpret it in conflicting ways (e.g. is the high CPU usage attributed to a particular need for CPUs, or have CPU users not made the effort to reimplement on GPUs instead? Is current GPU utilization representative of actual need, or does it mirror some discontent wrt access or usability).

Recommendation: It would be very useful to institute a formal data gathering program with usage dashboards, and utilization data. That would be useful for both users, and decision makers. It will not be easy, nor reliable, but it will be better than operating on gut feelings. Likewise, we need to engage more with faculty and students to understand their computational needs and their thoughts on using shared resources. There is certainly a disconnect between what we offer and what people need, but this has not been quantified by raw data yet.

Keeping the group informed

A key task in the charter of this working group was to review new policies, purchases, proposals, etc. that relate to campus computing. However, we were frustratingly in the dark when it came to a lot of the related activity. Through hearsay our members would learn of new initiatives, potential investments, and plans, but there has been no formal way to keep up informed.

Recommendation: It would help immensely if this group was regularly informed of any developments in campus computational infrastructure. Although we are happy to chime in with our concerns, doing so in a vacuum is not as impactful. Having the broader context and the opportunity to see potentially concerning actions would make us much more effective in our charter.