Research Computing Infrastructure and Researcher Engagement: Notes from Neuroscience

RAJENDRA BOSE, Zuckerman Institute, Columbia University, USA
ALEXANDER ANTONIADES, Zuckerman Institute, Columbia University, USA
JOHN PELLMAN, Zuckerman Institute, Columbia University, USA

The advent of the Mortimer B. Zuckerman Mind Brain Behavior Institute (Zuckerman Institute) at Columbia University over a decade ago presented the opportunity to design a discipline-focused Research Computing (RC) group allowing for close collaboration with a relatively fixed number of neuroscience laboratories to enhance discovery. Experiences and observations related to customizing Zuckerman computing infrastructure, creating "task-based" services and systems, and engaging with researchers in our Institute are shared to inform others about establishing discipline-focused research computing teams. Case studies related to providing a GPU cluster service tailored to Institute needs and the evolution of infrastructure choices to hybrid designs allowing bursting to vendor-provided cloud services are reviewed. Future directions involving research software engineering and sharing whole data analysis pipelines are noted.

CCS Concepts: • **Applied computing** → **Life and medical sciences**; • **Social and professional topics** → *Management of computing and information systems.*

Additional Key Words and Phrases: research computing, scientific computing, research support, GPU

ACM Reference Format:
Rajendra Bose, Alexander Antoniades, and John Pellman. 2020. Research Computing Infrastructure and Researcher Engagement: Notes from Neuroscience. In *Practice and Experience in Advanced Research Computing (PEARC '20), July 26–30, 2020, Portland, OR, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3311790.3396644

1 INTRODUCTION

The advent of the Mortimer B. Zuckerman Mind Brain Behavior Institute (Zuckerman Institute) at Columbia University in the late 2000s presented an opportunity to design a discipline-focused Research Computing (RC) group[1] for a population size of researchers (today, over 600 people across 49 laboratories) allowing for creative collaboration. The Zuckerman RC team seeks to engage and partner with Institute researchers to create customized computing solutions, beyond what may be possible with broader, university-scale computing support.

Zuckerman researchers explore how the brain develops, performs, endures, and recovers to gain critical insights into human health and behavior. The Institute's labs feature diverse scientific activity, encompassing experimental observations, advanced brain and cellular imaging techniques, and modeling and simulation of brain function. Early concepts behind the development of the Zuckerman RC team included:

(1) supplementing existing, centrally-provided research computing services by focusing on local infrastructure and training at Institute-scale, and

---

[1] Areas critical to the functioning of any enterprise including email, standard network connections, Windows and MacOS troubleshooting and imaging, and basic network security—often grouped as "IT" or "Helpdesk" functions—are delegated outside of the Zuckerman RC group.

(2) fostering partnership and direct interaction with faculty and their lab members to serve the labs most effectively and help drive their science to new levels.

Institute researchers supported by the RC team range from newly minted faculty to established principal investigators running large (30-50 person) labs comprising graduate students and postdoctoral and other research staff. The following sections share experiences and observations in tailoring Zuckerman computing infrastructure services, creating "task-based" systems and filtering technical "noise" while engaging with researchers in our Institute. The aim of this paper is to inform others about establishing discipline-focused research computing teams.

2 INSTITUTE-SCALE COMPUTING INFRASTRUCTURE

The Zuckerman Institute's RC planning centers on creating a technical infrastructure flexible enough to integrate new storage technologies and leading edge computational approaches.

Strategic planning for the Zuckerman RC group commenced in 2012. Early interviews were held with select faculty labs at other Columbia locations that would eventually move to the Institute, and these discussions about potential future needs informed the scale of planned initial RC data storage and computational systems. A Visiting Advisory Committee formed in January 2013 included leadership from eight established regional scientific/research computing groups and provided additional planning material for infrastructure and services. Construction of the Jerome L. Greene Science Center proceeded during the next several years.

Inputs to planning included the combined decades of experience of RC team members, direction from Institute leadership, and the content of periodic regional, national and international meetings with research computing counterparts from peer institutions. The overarching aim in planning efforts was to complement the immediate, time-sensitive needs of the research labs with a longer-term, larger-scale Institute perspective.

Current strategic goals of the RC group are to:

- Scale support to many hundreds of Institute members with limited staffing through careful growth of RC-managed infrastructure, featuring self-service, automation where possible, and rate models appropriate for labs;
- Create a transition path to RC services for the many labs that manage their own equipment, by:
  1. supplying a robust, backed up storage foundation to enable collaboration and fulfill university retention policies;
  2. adding computation services (such as virtual machines and specialized GPU capabilities) connected to the storage system; and
  3. achieving standard software configuration on computation services and allowing bursting to vendor-provided cloud services when required.
- Work proactively with labs and the pre-award/research development group on the computing- and data management-related portions of proposal creation and planning; and
- Continuously track and compare infrastructure based on-premise with vendor-provided cloud service pricing and capabilities. (The use of cloud services are discussed further in Section 3.2.)

Strategies for research data storage include progress toward a storage infrastructure that will maintain a relatively small footprint of local hardware with performance suitable for active, day-to-day research data storage for labs, while investigating a "deep archive" tier for data storage, possibly including a tape component [3].

3 ENGAGING WITH RESEARCHERS FOR TAILORED COMPUTING SOLUTIONS

The RC team grew to 3–4 people as the completion of the Greene Science Center housing the Institute proceeded and support of Institute labs began. Team members have had diverse backgrounds, often including some professional system administration, and sometimes including familiarity with research or experience working with researchers in functional imaging, data science and other areas. Over time, in both scheduled and informal meetings with faculty and their labs, RC team members have sought to avoid an overly prescriptive approach by working directly with individuals to learn the details of their current workflow and practice, then offering suggestions to improve or streamline processes when possible by incorporating use of our services and systems into their procedures.

When engaging with labs, RC staff attempt to condense highly "noisy," fast-moving technology trends to succinct talking points crafted for busy researchers steeped in the details of their own highly specific scientific procedures and publications. The goal is to remove barriers to the pursuit of science and minimize the time lost on computing-related distractions.

As technology development in software and hardware accelerates, researchers involved with procuring powerful workstations and modifying and running specialized programs are often faced with rapidly multiplying computing-related choices and decisions. The value an RC group can provide includes helping scientists choose the right technology path, potentially including managed services and automation, to fit their needs. Graphics processing unit (GPU) adoption by Zuckerman researchers serves as a case study.

3.1 Case Study 1: GPU Needs and Resources

An early scientific computing specialist role on the RC team started to document the workflows and data flows within a first set of Institute labs during 2015-16. By mid-2018 nearly the full complement of current labs had arrived under one roof in the new Greene Science Center, initiating community-building for the Institute. Additional RC staff with a mix of system- and researcher-facing responsibilities scaled up understanding of scientific computing within individual labs by compiling an up-to-date inventory of actively used software.

The inventory helped to identify common computing environments shared across the Institute. A pattern emerged that many labs were using high end consumer NVIDIA GPU cards in their workstations. Several labs were using *spike sorting algorithms* (grouping the detected firing of neurons—spikes—into clusters based on the similarity of their shapes) and *machine learning applications* that could take advantage of GPU cards.

Many labs transitioned to the Institute from other areas of the university where, of necessity, their computational environments were purchased, configured and managed by one or more lab members: faculty principal investigators, Ph.D. students, postdocs or other, possibly shared lab staff. The customary approach for labs to acquire and access GPUs has been for them to research and specify the technical requirements, then purchase, configure and self-manage their own GPU workstations, including compiling and installing specialized software. The tradition of self-managed servers and workstations can introduce challenges to labs and reduce time spent on science, as in the first two cases below.

Case a: One postdoc who transitioned to the Institute spent over a week grappling with GPU computing compiler tooling in order to build a GPU-enabled spike sorting MATLAB package, and through trial and error had created a non-functioning build environment. They were unable to compile the program, and could not correctly identify the tainted build environment as the primary impediment. After consultation with the RC group, they were able to build the program after creating a clean build environment.

Case b: In another instance, a researcher built an elaborate NVIDIA 1080ti workstation, typically seen in the PC gaming market, out of parts ordered separately and assembled in the lab. Unfortunately the resulting PC wouldn't start and the researcher entered a troubleshooting period of three months, during which various parts of the workstation (power supply, motherboard) were tested in turn. A lack of both standardized parts and a support plan common to most enterprise part replacement plans (which usually include overnight shipping of replacement parts) complicated matters.

Case c: Other Zuckerman labs with the need for GPUs participated in Columbia's most recent (2018-2019) central high performance computing (HPC) cluster through the purchase of GPU servers with a five-year lifetime from the central IT research services group. The shared nature of the university-scale HPC system has been advantageous in many cases, providing access to larger computational power than the lab's purchased share when available. However the enterprise grade hardware featured 2 NVIDIA V100 GPUs per server, which was expensive in comparison to the computing power offered by individual workstations. The Zuckerman labs felt that the advantages the enterprise V100 cards offered (primarily more, error correcting, memory) didn't offset the difference in cost.

The RC group ultimately seeks to move from the more time-consuming, individual help provided in case *a*, to emphasize "task based" services and systems. The team was able to do this in providing a solution to cases *b* and *c*, as described below.

In mid-2018 the team had discussions with targeted labs with a known interest in GPUs and began a limited pilot effort involving an RC-managed GPU cluster using consumer grade NVIDIA 1080ti GPUs. This involved selecting a trusted specialty vendor which specialized in building servers which could reliably accommodate 8 GPUs in a standard chassis, and leveraging our national network of peer group contacts to specify leading edge processors.

The success of the pilot resulted in the RC group's launch of a GPU cluster service in 2019 which, for now, follows the model of the university's larger HPC cluster service: Institute faculty have the ability to purchase standard hardware specified by the RC team; the RC staff installs, configures and manages the cluster; and cluster policies are defined by an RC-moderated operations committee with faculty members and their technical and administrative delegates.

One key differentiation between Zuckerman's GPU cluster and the broader university HPC cluster service is that the Zuckerman GPU cluster is structured around machine learning libraries and the scientific Python stack. The RC group found in talking to researchers that the jobs they were running did not make use of MPI or more traditional elements of HPC clusters. This led to a software environment that was more built around the *conda* package manager, with very little, if any communication between compute nodes. In many ways, Zuckerman's GPU cluster is more akin to a many-task computing (MTC) or big data cluster, which is attributable in large part to the fact that neuroinformatics as a distinct field is still relatively young (with the International Neuroinformatics Coordinating Facility founded in 2005) and as such is more influenced by broader computing trends that have occurred over the past decade.

An advantage of the GPU cluster is that, unlike the scenario described in case *a*, the *conda* package manager eliminates most of the toil involved in installing software and even allows cluster users to self-provision software libraries. When custom software builds have been necessary, the RC group has been able to cooperate with cluster users to produce working installs in a matter of one or two days, as opposed to one week. Some custom-compiled software is even provided within a local *conda* repository maintained by the RC group. In this way, the burden of navigating a traditional software build process by non-software engineers has been reduced and even eliminated.

In contrast to case *b* above, hardware support for Zuckerman's GPU cluster is directly provided by the RC group. Rather than resolution times spanning the course of multiple months, many hardware issues have been resolved relatively quickly, within a few weeks at most, maximizing researcher productivity and moving the burden of having to troubleshoot from the researchers to dedicated technical staff. For instance, a researcher who had recently contributed new compute nodes to the GPU cluster recently had a hardware issue requiring that an internal component on their node's motherboard be reconfigured. Instead of needing to go through the steps of determining which component needed to be reconfigured, opening up a computer chassis, making a change, and then testing that change, the RC group completed this task for them in under one hour.

3.2 Case Study 2: Evolving Infrastructure Choices

The advantages and shortcomings of on-premises infrastructure compared to vendor-provided cloud capabilities have been a source of continuing and wide-ranging discourse between entities involved with *overseeing and funding* and entities involved with *directly providing* computational and data-related services to researchers at major research universities and computing centers. Several reports addressing this issue are available [5] [4]. A brief synopsis of some points related to Institute planning follows.

Relying on on-premise infrastructure, assuming local expert administrator staff, features:

- the ability to more finely control, customize and tailor systems and data storage to an institution's needs; and
- predictable and stable costs over the long-term, but involving capital expenditure.

Relying on vendor-provided cloud services, also assuming local expert administrator—and "cloud specialist"—staff, features:

- the ability to "burst" computation to practically limitless resources when needed;
- a remote environment that others control, designed in many cases for business and not research applications;
- harder-to-predict costs requiring ongoing operational subscriptions or expenditure (thus introducing limitations to the availability of resources that could impact discovery); and
- no need for capital expenditure.

Our planning, likely similar to other institutions, incorporates a hybrid strategy to use the advantages of both approaches, extending our local computational services with the capability to burst to vendor-provided cloud services when required. At least one lab is expecting to require occasional MATLAB computational needs beyond the capability of the university HPC clusters or Institute resources. Other labs are using credits or other funding to work directly with vendor-provided cloud services to host software tools and data sets.

RC services include consultation and guidance for selecting computing environments to fit particular lab needs, including with accounts set up with the university's AWS agreement. Part of this assistance could include, for example, ensuring use of available private network links between the university's infrastructure and the cloud service provider to speed data transfer, providing an interconnection to the existing RC-managed storage system if warranted, and potential participation in the NIH STRIDES Initiative [1] which could allow for deeper technical interaction with cloud service vendor staff.

One senior researcher in the neuroscience community recently shared informal comments on "Computing models for a neuroimaging lab" [6] based on the various lab computing paradigms tried during his career at different stages and institutions. Concurring with the Zuckerman RC goals, he advocates for partnerships between researchers and computing professionals when possible with hybrid or other approaches to preserve the limited time and effort of lab members for scientific research.

3.3 Case Study 3: Self-service Software Provisioning

A perennial issue in the administration of scientific computing resources has been finding ways to install research software in a timely and sustainable way. Many researchers are hampered by the discrepancy between the time when they request a new software package and the time it is installed on a shared computing resource. Systems administrators have often been regarded as gatekeepers rather than facilitators, much to the frustration of researchers that want to rapidly iterate on their analyses.

To be consistent with the RC team's general philosophy of guiding researchers towards technical solutions rather than imposing requirements upon them, the RC team adopted a self-service model for installing software on the GPU cluster. Researchers can use the *conda* package manager to install copies of common software packages (such as the GNU Compiler Collection and ffmpeg) within their home directories, eliminating the need for administrative access, and reducing friction as they develop their analytic pipelines. For more tailored needs, we offer to build software using the Spack package manager, which boasts substantially more customizability than *conda* at the cost of needing to compile libraries and binaries from scratch. The bulk of the software used by researchers on the GPU cluster, however, is rooted in the machine learning community, which produces mature and generally well-tested packages (such as Tensorflow and PyTorch) for *conda*.

This approach has freed our team up to pursue more complex issues in software packaging and software compatibility. For example, a researcher once encountered difficulty installing Jax, a GPU-accelerated numerical computing library that reimplements the NumPy interface. An issue with the standard build process used by the Jax maintainers prevented the library from working under CentOS 7. The RC team was able to determine a workaround and then incorporate that workaround into a custom package saved to an Institute-specific *conda* package repository that researchers could install from. With the time saved from not having to attend to software installation tickets, we were able to create a reusable package for other researchers at the Institute and provide feedback to the Jax maintainers directly via their GitHub project.

4 FUTURE DIRECTIONS

A logical extension of the Zuckerman RC goal to achieve standard software configuration on RC-managed infrastructure computation services mentioned in Section 2 could be assisting with software development at the lab and researcher level, something that current staffing does not provide for. Areas that have been pursued by research computing groups elsewhere include software consulting, which could encompass software integration or systems testing and debugging of researcher code at a high level, and offering the service of a research software engineer (See [2]). This would involve having a dedicated software engineer on staff with domain expertise who could create well-crafted research software according to best practices in software engineering.

An overarching vision for the Institute's RC group—connected with an NIH-funded Data Science Consortium initiative fostering the collaboration of a number of national, multi-year (NIH U19) awards—is to improve scheduling, monitoring and visualization for lab researchers completing many sequential experimental pipeline runs or dealing with many varieties of workflows, and to create a suitable, stable platform for long-term Institute use which may eventually inform the wider community. This comprehensive Institute-wide effort will be built on the foundation Zuckerman Research Computing has established to date.

5 CONCLUSION

The creation of Columbia's Zuckerman Institute provided a mandate to build a discipline-focused RC team to engage proactively with researchers on computing solutions to further the fundamental science mission to decipher the brain. Strategic planning for RC infrastructure focused on achieving the capability to scale support to hundreds of Institute members with a small number of staff, partly by creating a transition path to centrally provided RC services for data storage and computation appropriate for the labs. When engaging with labs, the RC staff aims to filter highly "noisy" and rapidly advancing technology directions to allow time-limited researchers to advance their science. The team also seeks to move from individual help on day-to-day details of running hardware and software to emphasize "task based" services and systems, including a successful GPU cluster service for neuroscience-specific applications. Our evolving infrastructure plans include a hybrid strategy to use the advantages of both an on-premise environment, enhanced with the ability to burst to vendor-provided cloud services when required. Future directions could include more involvement with software development at the lab level, and improved tracking and sharing of lab workflows and data analysis pipelines.

ACKNOWLEDGMENTS

We would like to acknowledge the other Zuckerman Institute Research Computing team members, past and present, including Bruno Scap, Dennis Shushunov, Lokke Highstein, Yaki Stern, Roslyn Hui, Vlad Bouchev, Jochen Weber, Tim Lumley and Ben Silver, as well as all others involved with planning and operations from 2012-present, including Alan Crosswell, Barbara Han, Halayn Hescock, and George Garrett.

REFERENCES

[1] 2019. *NIH Science and Technology Research Infrastructure for Discovery, Experimentation, and Sustainability (STRIDES) Initiative.* Retrieved May 14, 2020 from https://datascience.nih.gov/strides

[2] 2020. US Research Software Engineer Association (US-RSE). Retrieved May 14, 2020 from https://us-rse.org

[3] Rajendra Bose and Bruno Scap. 2017. *Research Storage for Columbia's Zuckerman Mind Brain Behavior Institute.* an experience/position paper for the Workshop on Research Data Management Implementations. Arlington, VA. https://rdmi.uchicago.edu/papers/08212017163548_NSF_RDMI_CU_paper_20170821_final.pdf

[4] Jim Bottum et al. 2017. *The Future of Cloud for Academic Research Computing.* Results of an NSF-Supported Workshop, Entitled "Cloud Forward" Supported by NSF ACI/CSE Award 1632037. https://www.microsoft.com/en-us/research/publication/the-future-of-cloud-for-academic-research-computing/

[5] Dennis Gannon. 2018. *The State of the Cloud for Science-2018.* Technical Report. https://doi.org/10.13140/RG.2.2.27664.30729

[6] Russ Poldrack. 2019. *Computing models for a neuroimaging lab.* Retrieved May 14, 2020 from http://www.russpoldrack.org/2019/12/computing-models-for-neuroimaging-lab.html