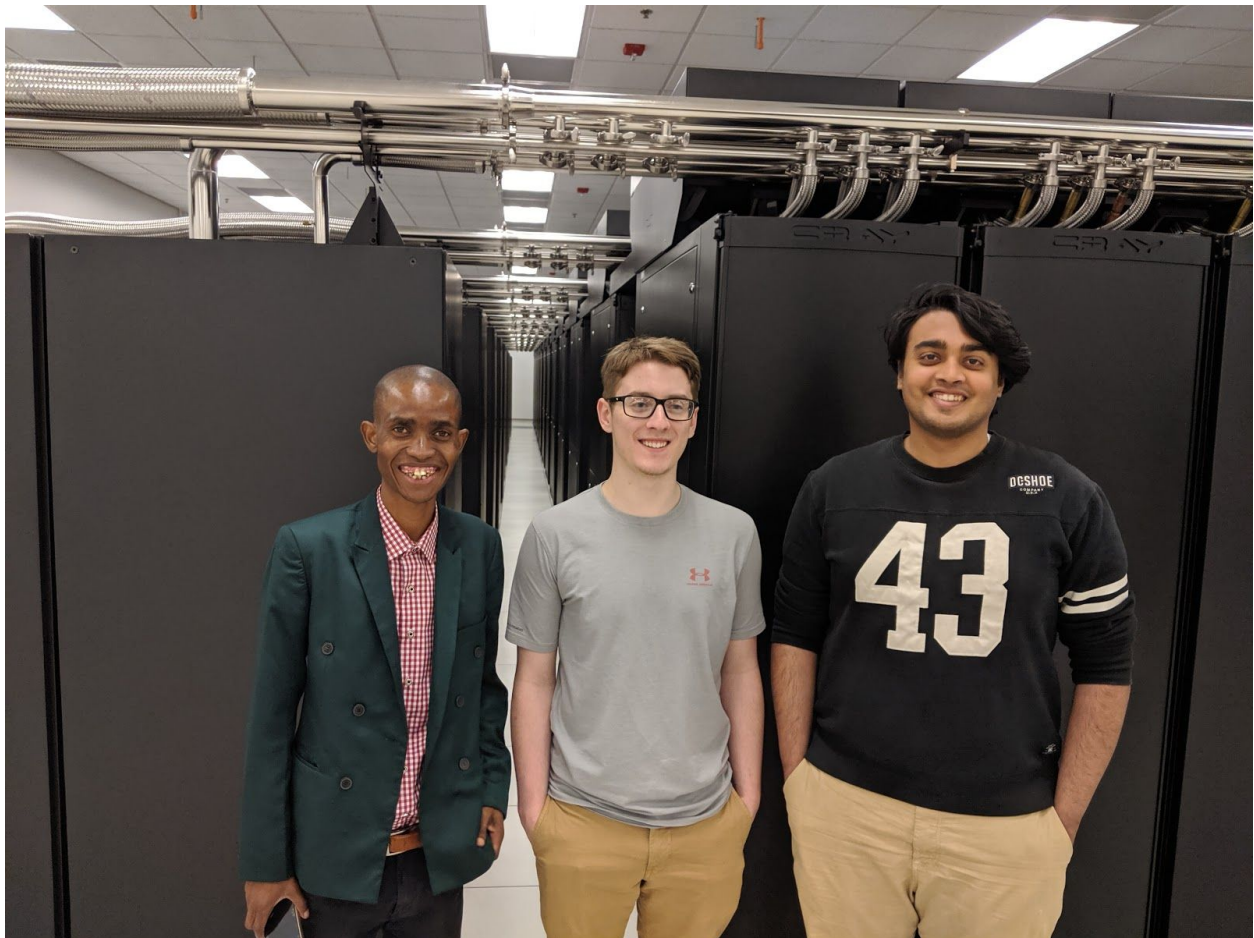


## My Experience as a CyberInfrastructure Professional Intern at the National Center for Supercomputing Applications - Badisa Mosesane

The National Center for Supercomputing Applications (NCSA) hosted the third cohort of the CyberInfrastructure Professional Intern program (CIP) which I was truly honored to participate in from May 20 to August 16, 2019.

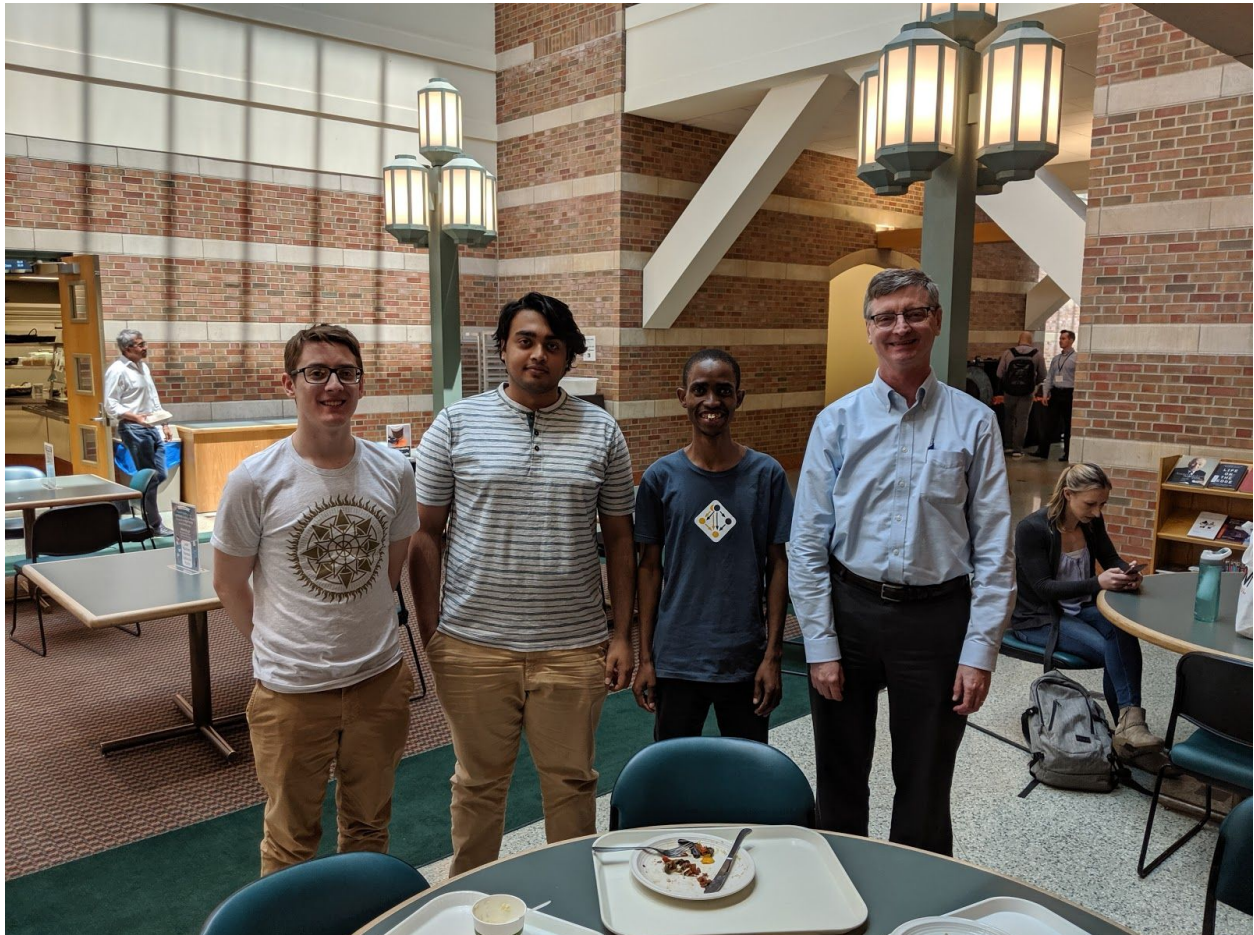
*The CIP program is NCSA's internship program for those interested in pursuing a career in Cyberinfrastructure (CI) and is designed to address the shortage of a workforce with the specialized skills needed to support advanced CI operations. During the program, I worked directly with engineers to gain hands-on experience in the CyberInfrastructure operations of a major leading supercomputing center.*



2019 CIP interns (left to right): Badisa Mosesane, Jacob Gallion, Nishant Dash

The first week culminated in a series of presentations from the different CI groups at NCSA giving a talk on their day-to-day work and research efforts. Some of the CI groups include the HPC Systems group, Storage Enabling Technologies, Advanced Visualization Lab, Data Analysis

and Visualization, Innovative Software and Data Analysis, Incident Response and Security Operations (IRST), Network Engineering, Integrated Data and Databases services and Innovative Systems Lab.



*CIP interns luncheon with NCSA Director Bill Gropp*

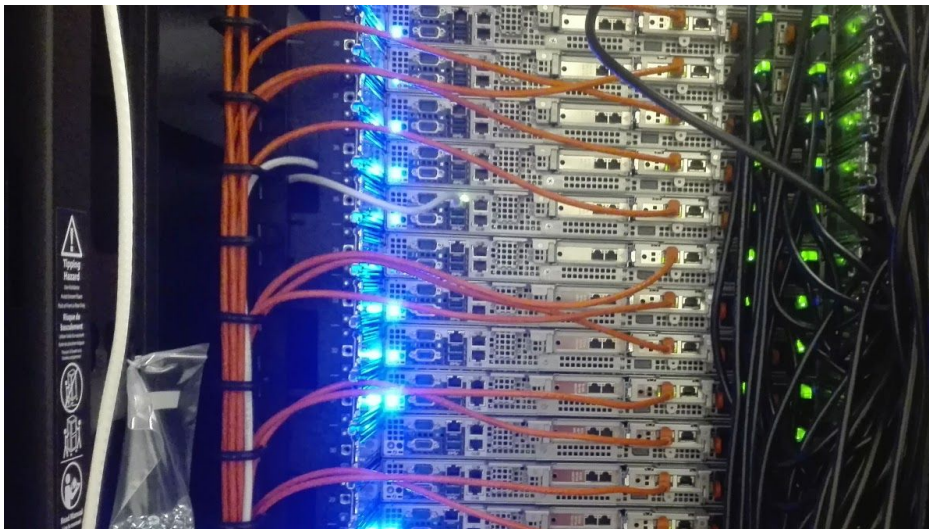
## One day embeds

After a week of presentations from each CI group, I selected 4 of the groups I wanted to do one day embed with. Embedding with a group meant, joining a team for one day to observe, interact, attend meetings for that group and learn the kind of work, tools, workflow they use and ask questions to get the most of what the group is focused on. My one day embeds were with Systems group, Innovative Systems Lab (ISL) and Incident Response and Security Operations respectively.



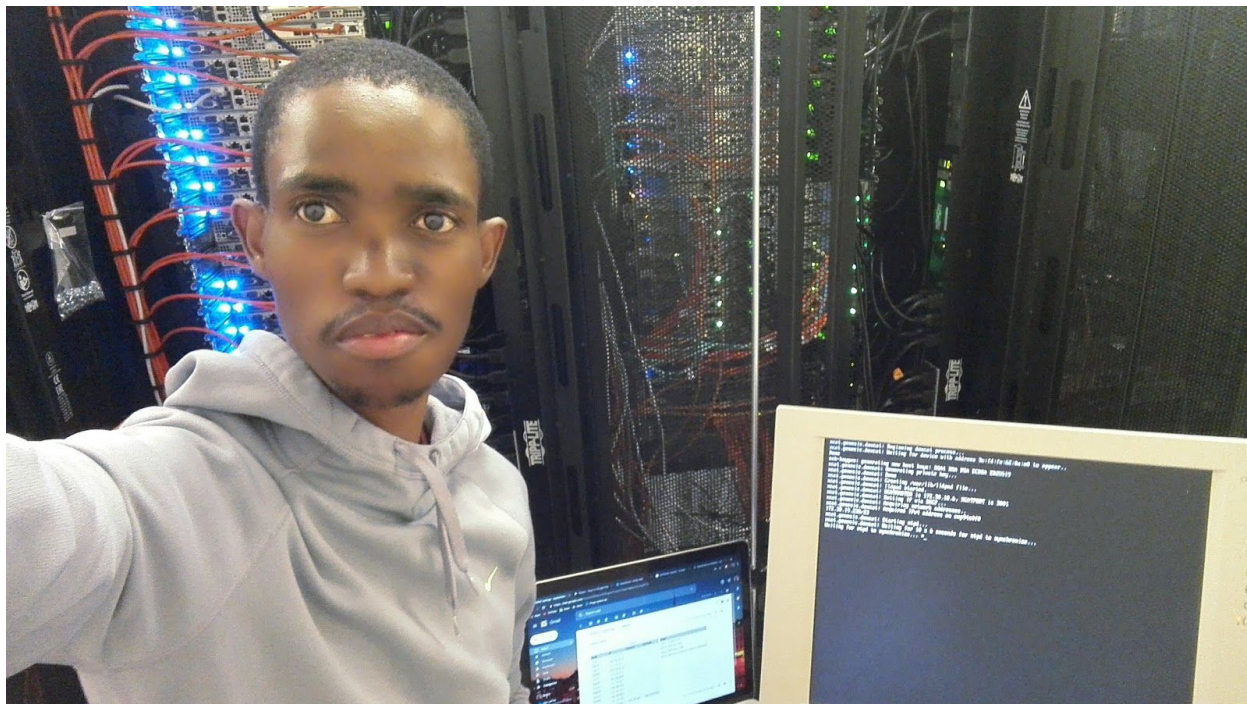
Summary of each group I embedded with is as follows:

- Systems: tasked with the deployment and effective operations of NCSA's HPC and large data resources.
- ISL: focus is on advanced GPU and possibly FPGA resources for researchers to work on optimizing and developing efficient methods of machine learning algorithms
- IRST: performs active threat hunting on the entire NCSAnet to detect misconfigurations and general system weaknesses. IRST also detects issues more broadly for all networked assets and to investigate more deeply than simple checklists for NCSA's most critical infrastructure.



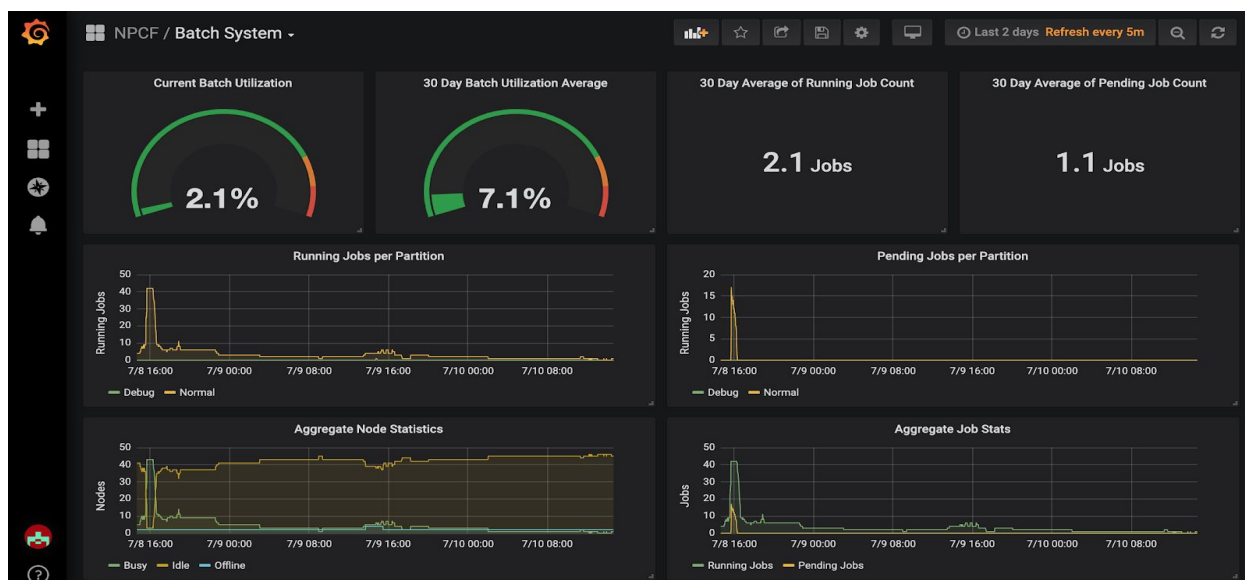
## Selecting the Primary CI group

Following one day embeds with the various CI groups, we had to make a choice on our preferred groups we wanted to intern with. As an HPC enthusiast with interests in administration & operational monitoring of Supercomputers and Deep Learning Applications on HPC environments, selecting which group I will be joining for the rest of the CIP internship was not an arduous task. I joined the Systems group based on my interests and the diverse systems and projects the team was involved with.



## Why am I interested in HPC monitoring?

High Performance Computing monitoring is one of my favorite areas. The reason I like it is because HPC clusters are complex systems that can fail due to various causes. Performance and reliability are one of the two metrics that are critical in operating HPC clusters that both SysAdmins and managers devote a special attention to.



Monitoring helps to visualize system health and utilization, visualize usage patterns and enable preemptive enhancements or fixes. It gives some foresight into the health of the cluster. On the



user side, it helps with better understanding of the utilization of the system and verifying correct execution of workloads.

## My work on NCSA HPC Systems team

The systems team is responsible for deploying and maintaining about 5 systems namely: BlueWaters (BW), Industry systems, Large Synoptic Survey Telescope (LSST), Campus Cluster (CC) and Deep Learning MRI (HAL) - Hardware Accelerated Learning.

During the internship I rotated across BW system, LSST, Industry, CC and then Industry systems respectively working on each system for 2 weeks. Working on all these systems has helped me get a broader sense of the different HPC stacks, applications & use cases and the 'why' and 'why not' of using certain solutions and why some workflows work and some don't. It also helped me validate and identify the important skillset I need to develop and 'must-know' tools for automating repetitive admin tasks.



Working on the systems team gave me hands-on experience deploying and maintaining HPC and big data systems, analyze and optimize HPC systems for more effective and efficient operations in concert with other CI groups at NCSA.

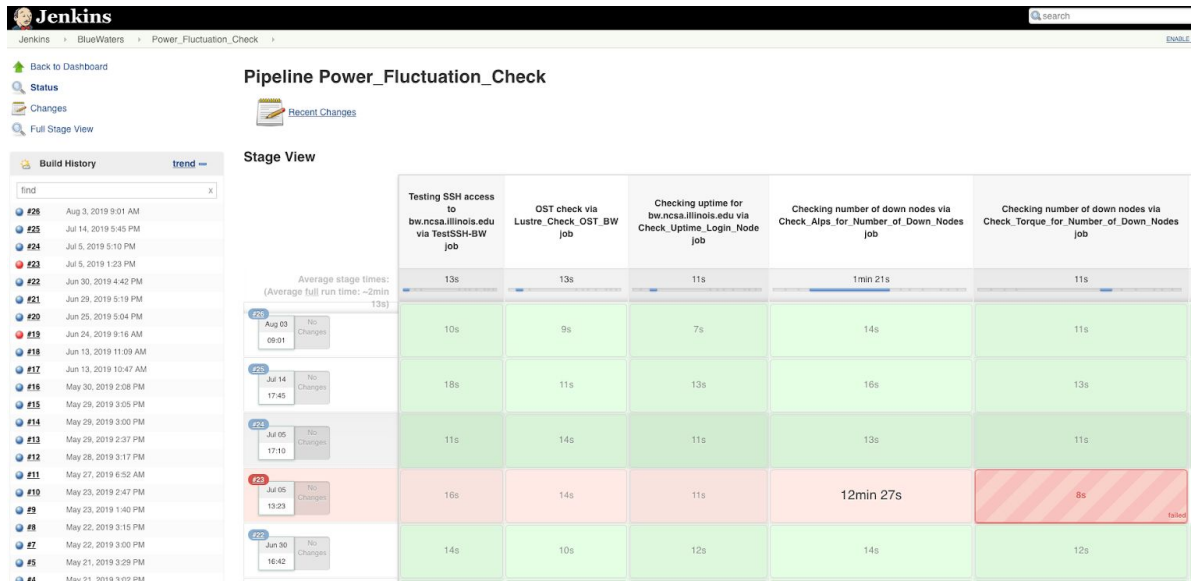
The many team meetings I attended across different systems opened my eyes to effective approaches to solving problems including from cyber security standpoint, which is often an after-thought in many development work. I appreciated the beauty of having diverse teams working together to advance & support computational work of many researchers.

While working on the tasks I was assigned, I maintained and kept the documentation up-to-date so that the same procedures I used can be followed by other sysAdmins to achieve a similar result.

## Skills and Knowledge Acquired

### Working on BlueWaters Supercomputer

- For 2 weeks I worked on the Bluewaters system which is NCSA's first Petascale system with over 27,000 nodes capable of doing sustained science calculations in excess of 1 PetaFlops/s. The primary users are scientists with a demonstrated need for large scale calculations. It's peak speed is almost 3 million times faster than an average laptop.
- I spent most of the time investigating on this supercomputers' machine architecture, how it balances processing speed with data storage, memory and communication within itself and to the outside world and the kinds of Scientific projects it supports.
- I also had a chance to look at BW-jenkins - a continuous integration tool for source code repositories using pipelines and automating other routine development tasks. Jenkins runs multiple tests on BlueWaters like power fluctuation checks, check if some machines can be SSH'ed into (SSH check) and display the results of these automated tests in a console output. Jenkins can perform all tests in production and tests currently in development. Some of the BW-jenkins tests include: ssh access to BW, check uptime for Bw via check\_Uptime\_Login\_node\_job, check number of down nodes, check Lustre availability, LDAP power fluctuation checks, check for scheduler paused duration frequency and power feed tests.



The challenging task was wrapping my head around understanding some of the Bluewaters monitoring scripts written by experienced engineers some time ago - obviously with less documented code. After spending some time on the bash and perl scripts, monitoring.jobs.pl, monitoring.nodes.pl and monitoring.user.pl I then understood what each script is doing, useful for monitoring the Bluewaters system, coupled with jenkins.test.

### Working on Large Synoptic Survey Telescope (LSST) System

With a little background on Astrophysics & Particle Physics from [CERN](#), I was so excited to get my hands dirty on the Large Synoptic Survey Telescope. The LSST project will use a 3.2-Gigapixel camera to photograph a giant swath of the heavens for 10 years. Each day it will capture more than 20 Terabytes of data, capturing every night with a clear sky, creating the world’s largest astronomical stop-motion movie.



*Credit: Symmetry magazine, sandbox studio, chicago with Ana Kova*

<https://www.symmetrymagazine.org/article/an-astronomical-data-challenge>

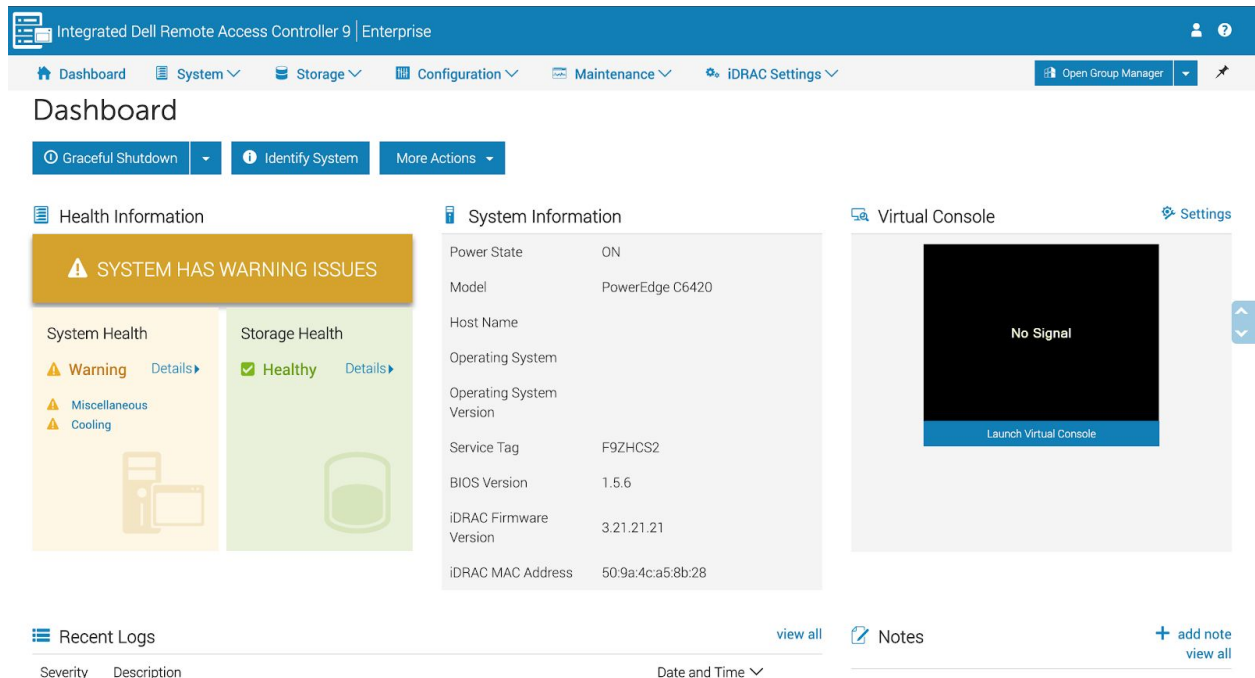
The LSST project will give unprecedented real-time streams to help scientists look at billions of celestial objects and study changes in the sky, enabling them to respond to transient events like sudden flares from active galaxies, exploding supernovas, discoveries for new asteroids and receive updates about anomalous activities happening in the skies.





I was fascinated to learn about the computational complexities of this project. The collection of enormous volumes of data each day, processing it for use by scientists and distributing it among institutions all over the globe through data highways proves a task that needs a dedicated multidisciplinary team.

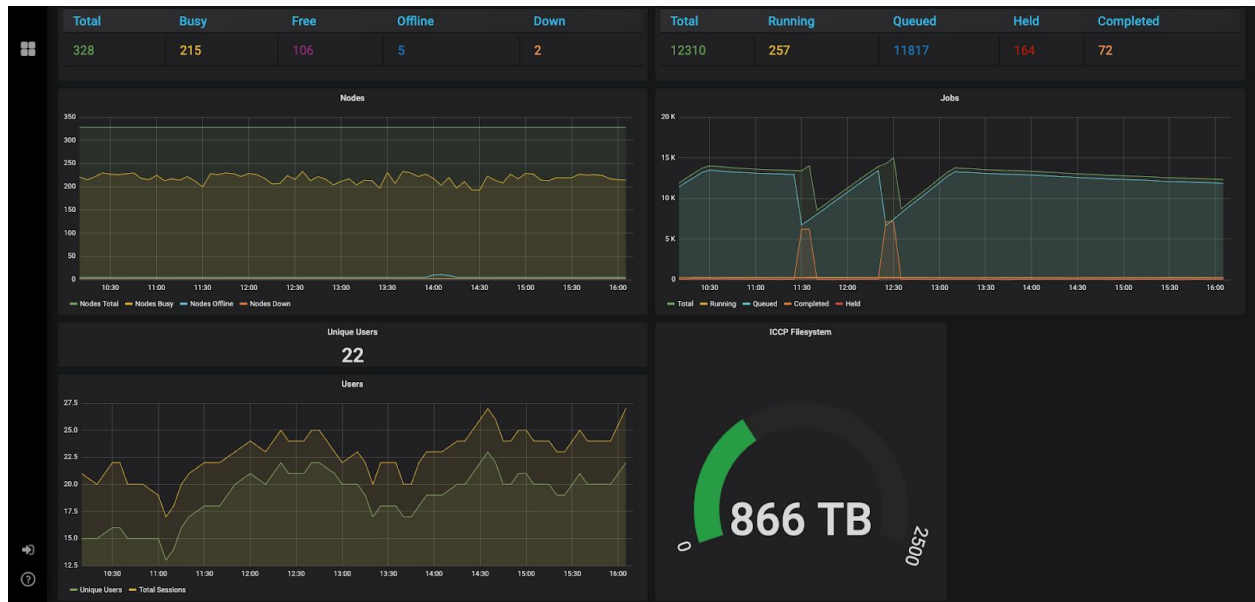
Alongside engineers at NCSA, I worked on fundamental tasks that help in batch processing for real time analysis and reporting. One of my tasks was to deploy xCAT service on LSST test cluster of 2 nodes, frwd24 and frwd25. xCAT is an open-source tool for automating deployment, scaling, and management of bare metal servers and virtual machines.



Some of my work aimed at being used as a test-bed for deploying machines onsite in Chile. To achieve this I had to work on a couple of tasks such as:

- xCAT software install
- configuring new xCAT master node running CentOS 7 (testVM31)
- start Puppet on xCAT master node - Puppet is a very useful tool for configuration management & maintaining server machines.
- install xCAT tools
- set site attributes
- setup DNS, DHCP, VM management, physical hardware management, create routes and groups for public networks, setup postscripts and install OS images and setup NTPD

I also worked on the Telegraf-InfluxDB-Grafana stack to create a monitoring dashboard for LSST. The dashboard displays the most important metrics of the cluster and sends alerts via slack to engineers on anomalous activities in the cluster.



Grafana monitoring dashboard

## Understanding the Telegraf-InfluxDB-Grafana (TIG stack) Monitoring Workflow

I used the TIG stack to build a high-level LSST dashboard with panels for displaying information on number of total unique users, number of total nodes (busy, free, offline, down) and number of jobs (total jobs, running, queued, held, completed).

In simple terms, telegraf collects metrics data from tonnes of nodes and populates the influxDB database and Grafana builds queries and display the metrics in a real-time dashboard.

In summary;

**Telegraf:** is the agent for collecting and reporting metrics and data. It does automatic metrics collection on a flexible plugin-based system. It has built-in plugins for common metrics (cpu, disk, network, etc) and can collect a variety of metrics, events and logs directly from the systems.

**InfluxDB:** is an open-source time series database optimized for fast high-availability storage and retrieval of time series data. It ingests data through the InfluxDB Line Protocol, SQL-like query language. It is great for operations monitoring, application metrics, and real-time analytics.

**Grafana:** is an open-source lightweight web dashboard for metrics analysis and visualization suite. It visualizes time series data for infrastructure and applications. It has Powerful built-in visualizations (panels) with plugin support for threshold-based alerts and email notifications.



It was rewarding and a sense of achievement to work on smaller parts of the project that will then fit into the bigger scientific project.

### Working on Campus Cluster (CC)

Campus cluster is operated by NCSA for the Illinois Campus Cluster project. It is designed to support various campus researchers in a 'condo' model of computing - investors buy the nodes and CC SysAdmins install and run them in the common cluster infrastructure, common network and storage backend.

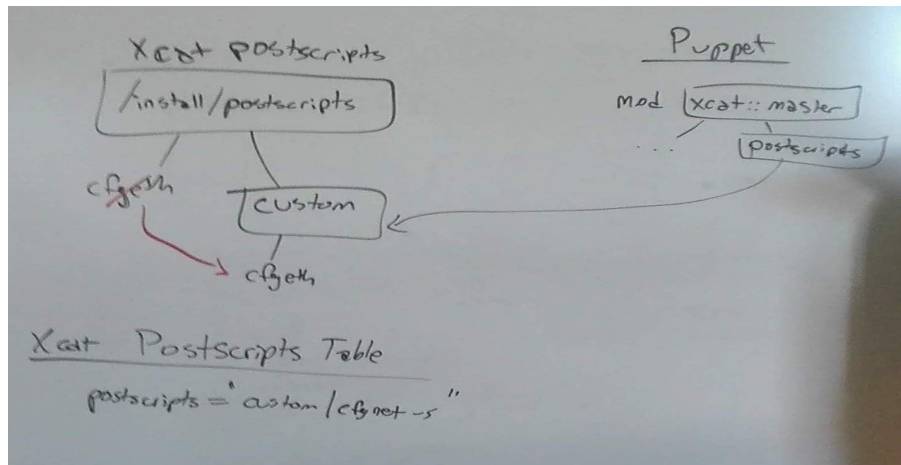
My tasks included assessing General Parallel File System (GPFS) race condition on campus cluster. GPFS is an IBM file system for distributing and managing data across multiple nodes, implemented in many HPC & large-scale storage environments. I helped in investigating campus cluster nodes that were down due to GPFS up with auto load using 10G in an effort to bring the nodes up.

### Deploying Puppet (Master) Service + xCAT and managing postscripts

I worked on a test campus cluster machine (tmg-01) to upgrade Puppet server, PuppetDB and Puppet agent to adopt new features and functionalities of newer versions. Puppet is an open source configuration management tool and server automation framework.

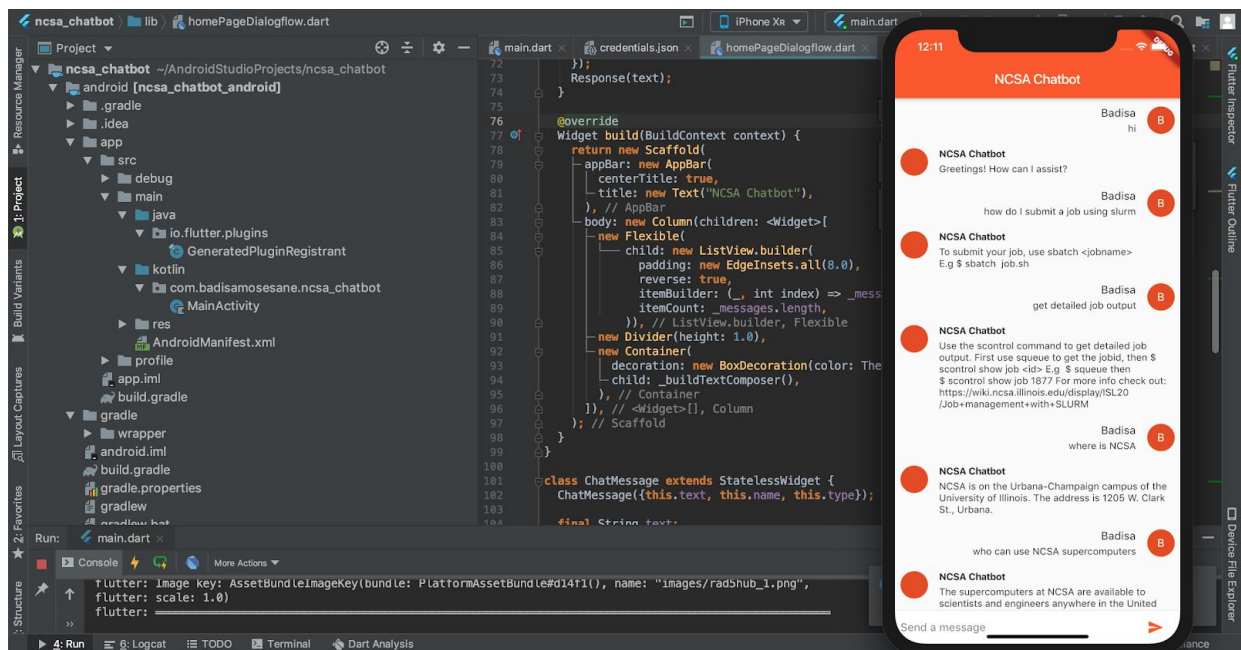
An interesting problem I worked on CC was understanding and testing out puppet to manage xCAT postscripts. Postscripts are scripts that should be run on each node after installation or diskless boot.

\*The challenge is when modifying one of these postscripts in /install/postscripts and then update xCAT, it overrides the changes and the previous modifications in these scripts are lost. The workaround is to have Puppet managing these postscripts, create a symlink in /install/postscripts that points to /custom directory then have a puppet module with r10k pulling any changes from the GitHub repository and applying the changes and installing to custom directory. Demonstrated in the image below;

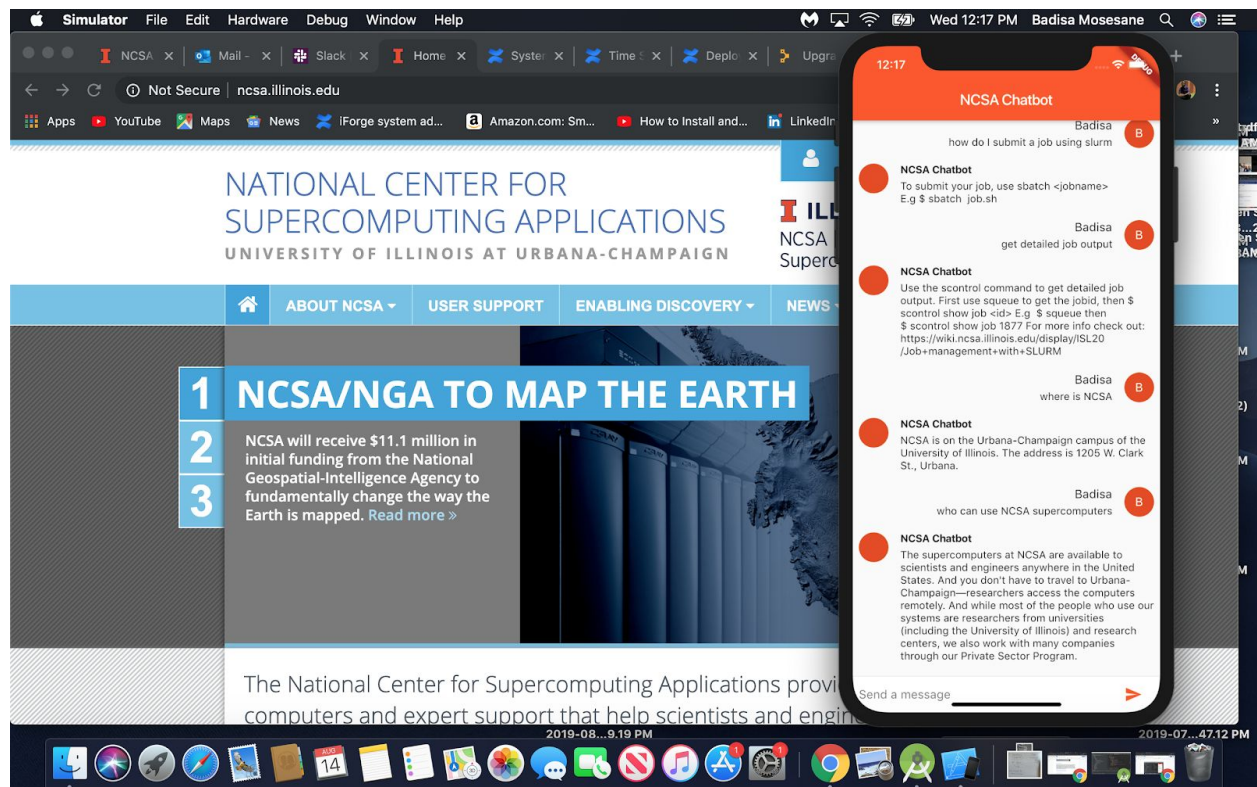


## CC user support Chatbot

Motivated by the Illinois campus cluster project resources page, I saw a need for developing a cognitive application prototype that can help CC users to provide automated responses to some of their questions relating to utilizing the cluster. I developed the chatbot using Flutter - a mobile app development framework for building Android and iOS apps in a single codebase, integrated with Dialogflow- an end-to-end development suite with powerful Natural Language Processing for building rich conversational experiences. Users can use voice, text or call the chatbot to ask questions like “how do I submit a job to the cluster”, “how to install R packages” or query any errors that encounter to get help.



Chatbot interface on the flutter iOS simulator



## Working on Industry systems

NCSA's Industry systems has general and special purposes HPC systems. The primary users for these systems are large industry firms with local (Midwest) roots looking to expand their HPC knowledge.

Much of my work on these systems ranged from creating queries from Slurm metrics influxDB, looking at IPTables setup, assess the impact of Qualys scans on the nodes, investigate nodes that were down due to 100% memory use by a particular user, job accounting on Industry systems, Syslogs, fix boot issues via iDRAC, learn the Slurm workload manager, de-racking and removing InfiniBand cards from a system that reached End of life. Some of my time was spent on the hardware tasks, pulling off rails, ethernet cables, Infiniband, sorting power cables and ethernet, replace power supplies and Protocol Data Units (PDUs).





*Deracking EOL cluster*

I also participated in the Planned maintenance (PM) of one of the Industry systems and got to see how the PM procedures and status page is maintained to make sure everything that was planned goes as planned. It was a great experience to see first hand PM operations and running Nagios tests after the services have been brought up online.

### Activities and events attended

I attended a couple of events which were beneficial to my professional development and made a couple of good contacts in HPC & AI communities.

Scaling for AI talk by Dell technologies: Sizing, solutioning, tuning, and supporting Deep Learning models at scale; infrastructure for models that don't starve GPUs waiting for resources as they train.



*Dell CTO presenting on Scaling AI*

PEARC19: presenting at the AI4Good session

<https://twitter.com/SciNode/status/1155852540006817792>

NCSA All staff meetings: lightning talks on monitoring stacks for the Storage and Enabling Technology group and plugins for sharing data.

IT Professional Forum: theme focused on ways technology empowers IT professionals to break down all kinds of barriers. Barriers between people, barriers to good service, barriers to learning, and barriers to efficient research.

<https://www.itproforum.illinois.edu/home>

Ice cream social: take a break from the busy day and get to know co-workers at NCSA over an ice cream.

## Applications and Tools learned

- Puppet & xCAT tools
- Slurm workload manager
- GPFS
- Telegraf-InfluxDB-Grafana time series monitoring and analytics
- Perl & Bash scripting
- iDRAC & IPMI
- Lua-based environment modules (Lmod)
- Containers (Singularity) and Kubernetes basics

## Future Work

I will continue working on some of these projects to develop my CI skills especially automating sysAdmin tasks. I intend to further learn Puppet software deployments, linux scripting, containers support in concert with versioning systems like github. I also would like to work on projects cutting across HPC and Machine learning e.g. applying machine learning models to influxDB on nodes metrics data to predict time series and foretell undesirable events on machines before they happen.

I would highly recommend the CIP program to anyone who is interested in developing a career around CyberInfrastructure operations. The experience has been an exciting and insightful one, both professionally and social wise.

## References

<https://github.com/ncsa/puppet-xcat>

[https://puppet.com/docs/puppet/6.7/upgrade\\_minor.html](https://puppet.com/docs/puppet/6.7/upgrade_minor.html)

<https://xcat.org/>

[https://www.ibm.com/support/knowledgecenter/en/SSPT3X\\_3.0.0/com.ibm.swg.im.infosphere.biginsights.product.doc/doc/bi\\_gpfs\\_overview.html](https://www.ibm.com/support/knowledgecenter/en/SSPT3X_3.0.0/com.ibm.swg.im.infosphere.biginsights.product.doc/doc/bi_gpfs_overview.html)

<https://www.symmetrismagazine.org/article/an-astronomical-data-challenge>

<https://www.influxdata.com/blog/applying-machine-learning-models-to-influxdb-with-loud-ml-docker-for-time-series-predictions/>