Team Ada Six

Student Cluster Competition Team

Students: Nicole Brewer, Jessi Lanum, Claudia Li, Jia Lin Cheoh, HyeJin Kim, Heidi Anderson
Advisors: Elizabett Hillery and Jenett Tillotson

PURDUE UNIVERSITY

DELL

Introduction

Purdue University has assembled a diverse team of six undergraduates with highly differentiable academic backgrounds. Collectively we have research, internship, teaching assistantship, and mentorship experience. As the first all-female cluster competition team, we hope to promote the inclusion of women in the field of HPC. Our namesake is Ada Lovelace who is often regarded as the first female computer programmer.

Nicole, *Senior*
Mathematics with CS

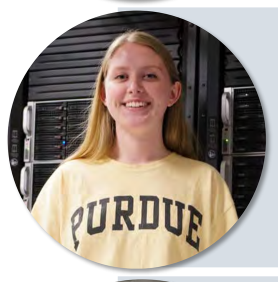
- **Student Software Developer** for Scientific Solutions in Purdue's RCAC
- **Undergraduate Researcher** creating data on the Purdue cluster, Halstead

Jessi, *Sophomore*
Cybersecurity

- **Project Assistant** for Dark Enterprises Inc.
- **Student Cybersecurity Analyst** for Purdue's RCAC

HyeJin, *Senior*
Mathematics, Statistics, and CS

- **Software Engineering Intern** at a search engine company, NAVER
- **Undergraduate Researcher** studying convex optimization in machine learning

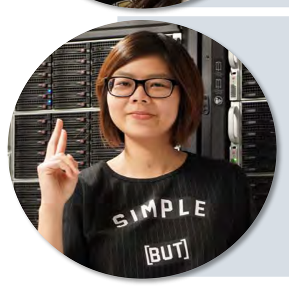


Heidi, *Junior*
Computer Science

- **Mentor** for local middle and high school girls in computing
- **Technology and Computer Chair** of Gamma Phi Beta sorority

Claudia, *Sophomore*
Neurobiology

- **Undergraduate Researcher** performing data analytics in a neurobiology lab
- **Statistics Minor** with experience manipulating brain images in R



JiaLin, *Senior*
Computer Science

- **Undergraduate Researcher** in computational science
- **Commercial pilot** for Singapore Airlines

Haynes Cluster

Our cluster houses four **Dell PowerEdge R740** servers with the following configuration:

Node	
CPU	2x Intel Xeon Gold 6148
Memory	192GB
Storage	2TB SSD
GPU	3x Tesla Nvidia V100
Interconnect	Mellanox EDR 100Gbps

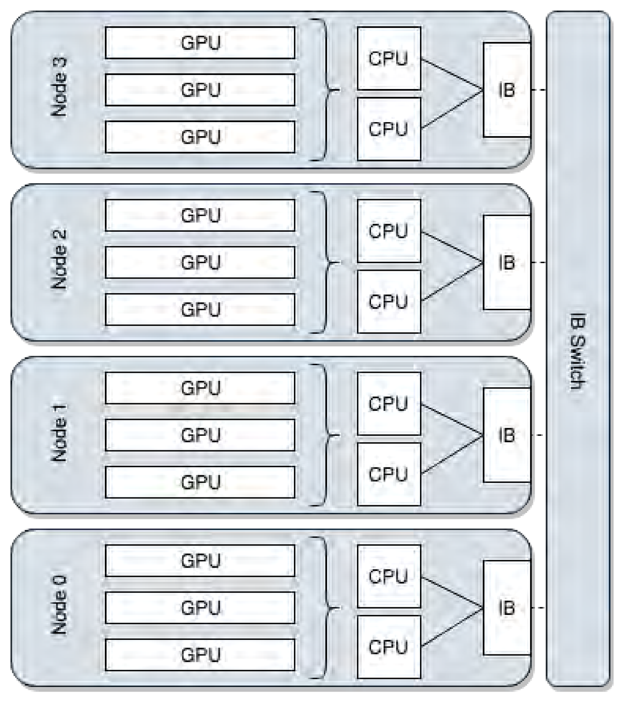


Figure 1. Diagram of cluster architecture. Haynes is named after Purdue's first female faculty member, Sarah Allen Oren Haynes

Hardware

Haynes was built with applications in mind. We have chosen to use all of the GPUs available to us because applications such as HPL, HPCG, and Horovod, run well on them. Trivially parallel applications such OpenMC weren't taken into as much consideration because they can be inexpensively offloaded to the cloud. Having four nodes allows our team to simultaneously share compute resources, without wasting too much power.

Software

Our operating system was chosen to match the OS on the cluster at we used to practice building applications before our cluster arrived. We chose to use SLURM as our batch system because it is open source and it includes a resource manager. The software and development tools we chose were necessitated by compatibility with the applications.

System Software	
Operating System	CentOS 7.5
Resource Manager	SLURM 18.08.0
Development Tools	
C and Fortran Compilers	GCC 4.8.5, Intel Cluster Studio XE 2018
Python Interpreters	Versions 2.7 and 3.6
Nvidia API	CUDA 9.2
MPI	Intel MPI 2018.3.222 OpenMPI 1.10.2 and 3.1.2
Optimized Math Libraries	Intel MKL
Nvidia Libraries	NCCL2

Time Constraints

Our choice to use four nodes and a resource manager will cost us some of our power allotment. However, this choice will save us the effort of strictly coordinating limited resources. This also prevents us from wasting time in between job runs, and the flexibility to queue jobs and physically leave our booth.

Power Constraints

We chose to include GPUs in our cluster's architecture, because even though they are power-hungry, they typically provide more FLOPs per watt than CPUs will, and many of the applications can take advantage of them." We plan to monitor our power consumption via the competition PDU. We will be able to manually throttle our CPUs and GPUs

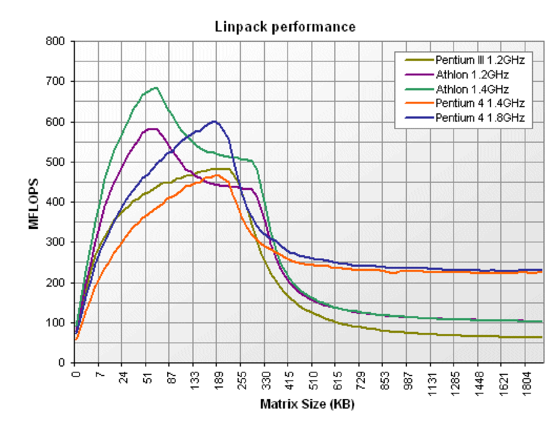
Azure Cloud

We have chosen to use cloud resources for applications aren't best suited to our cluster's GPU heavy architecture. OpenMC is a good candidate for several reasons:

- There isn't a GPU port for this application, so it is important that our cluster's GPUs are dedicated to applications that run more efficiently on them
- As a Monte-Carlo simulation we expect it to be trivially parallel, and therefore a good candidate to be run on inexpensive nodes that don't have components such as GPUs or InfiniBand interconnects

If the mystery app has analogous performance requirements, we may choose to offload it to the cloud also.

Applications



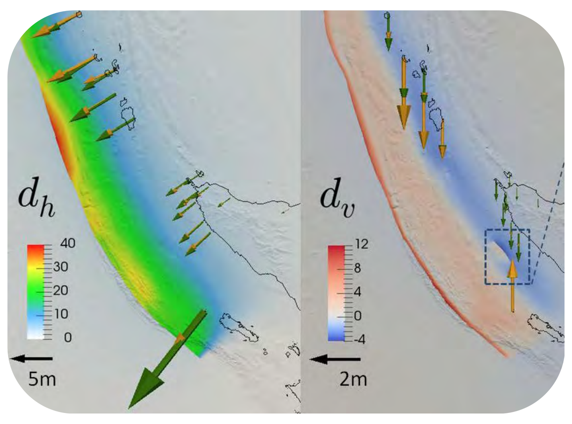
HPL & HPCG

We will run HPL and HPCG on the GPUs, as they handle the linear systems algorithms well. We executed test runs with different parameter combinations in the .dat files to tune each application. We also experimented with the latest GPU-optimized binaries from Nvidia, changing the number of threads to further optimize the performance. In order to run HPL and HPCG on our GPUS while staying under the power threshold, we will modulate the GPU frequency during the benchmark run to adjust the power while maximizing the performance.

HOROVOD

Horovod

Since Horovod utilizes TensorFlow, we built TensorFlow for GPUs from source instead of using the tensorflow-gpu pip package. This way TensorFlow at the base level is running as optimally as possible for our cluster. Additionally, we have written benchmarking and power meter scripts to test optimal performance across nodes and GPUs and monitor power usage at the competition.



SeisSol

SeisSol is used to simulate dynamic rupture earthquakes. This app is takes advantage of several optimizations including local timestepping, code generation to reduce the cost of matrix multiplication, and asynchronous output to prevent I/O bottlenecks. We have done single node performance runs on both the Shaking Corals and the BaseLine version of SeisSol to observe runtime speed-up. We have also performed string scaling studies to extrapolate the runtime of the entire simulation.

OpenMC

OpenMC

OpenMC is the app with the highest flexibility compare to all the other apps in terms of the environment that it can run on. We have prepared for the challenge on OpenMC by mastering the strategies to run on the cluster, interface and cloud. Depending on the nature of the Mystery app given to us, we will make a choice between running OpenMC on cloud or Mystery app on cloud. We also built OpenMC from source in the event that the datasets given to us will perform better when they are ran from OpenMC that is built from source. Since OpenMC is a scientific heavy application that has the ability to accommodate different file formats with scripting, we also prepared for the challenge of OpenMC by mastering a variety of data formats that can be churned out from OpenMC.

Mystery Application

We have prepared for the mystery application by building and running applications from previous cluster competitions. Depending on the nature of the application, we are prepared to build and run the app on the Azure Cloud.

Why We Will Win

- We have been building applications inside and outside of class time since the spring of last year, so we have accrued problem solving experience
- We have had access to Purdue's supercomputing cluster, Brown, in order to practice building and running applications in parallel well before the arrival of our Dell cluster
- We have a wealth of support from many experienced individuals in Purdue's Rosen Center for Research Computing
- We have more members in our group than we have team members and we've built a strong community of women in tech

WHPC WOMEN IN HIGH PERFORMANCE COMPUTING PURDUE UNIVERSITY

DELL EMC

NVIDIA

Acknowledgements

We owe a special thanks to our program manager Lev Gorenstein for his effervescence and continual availability day or night. We would also like to thank Chuck Schwartz, Steven Harrell, Trinity Overmyer, Pat Finnegan, and Professor Alina Alexeenko for their contributions to our program.