Daemon Deacons

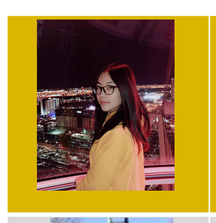
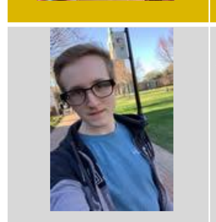
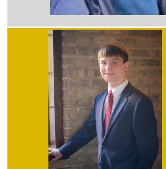
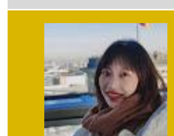
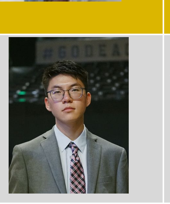
Sophia Fang, Paul Kefer, Robert Langefeld, Hao Tong, Eva Wu, and Yixin Zhang
Department of Computer Science, Wake Forest University, Winston-Salem, NC, 27109, USA

Primary Advisor: Samuel S. Cho (Depts. Computer Science and Physics)

Secondary Advisors: Sean Anderson (Information System), Adam Carlson (Information Systems) and Cody Stevens (Information Systems)

About our Team

Left to right, up to down: Paul Kefer, Prof. Cho (Computer Science & Physics), Sean Anderson (Information Systems), Cody Stevens (Information Systems), Robert Langefeld, Adam Carlson (Information Systems), Jeff Squyres (Cisco), Hao Tong, Eva Wu, Sophia Fang, Yixin Zhang.

	**Name:** Sophia Fang **Year:** Junior **Major:** Computer Science and Statistics **Programming Languages:** C, C++, Java, Matlab, R **Fluent Spoken Languages:** English, Mandarin
	**Name:** Paul Kefer **Year:** Senior **Major:** Computer Science **Programming Languages:** C, C++, CUDA, Java, Javascript, Matlab, Python **Fluent Spoken Languages:** English and German
	**Name:** Robert Langefeld **Year:** Junior **Major:** Computer Science and Mathematics **Programming Languages:** C, C++, CUDA, Java, Matlab, R, Python, Javascript **Fluent Spoken Languages:** English and Spanish
	**Name:** Hao Tong **Year:** Senior **Major:** Engineering **Programming Languages:** C, C++, Java, Python, MATLAB, **Fluent Spoken Languages:** English, Mandarin
	**Name:** Eva Wu **Year:** Senior **Major:** Computer Science and Statistics **Programming Languages:** Java, C, C++, Python, R **Fluent Spoken Languages:** English, Mandarin
	**Name:** Yixin Zhang **Year:** Senior **Major:** Computer Science and Statistics **Programming Languages:** C, C++, CUDA, Java, MIPS, Python, R **Fluent Spoken Languages:** English, Mandarin

Our team consists of Computer Science, Mathematics/Mathematical Statistics, Applied Mathematics, and Engineering majors (half are double-majors) with minors in Biology, Computer Science, Economics, Japanese, and Neuroscience. Two of the students are returning SC19 SCC veterans, and four are new. We have taken a wide range of relevant prerequisite classes, including Data Structures I and II, Databases, Parallel Computing, GPU Programming, and Machine Learning. Moreover, we are proficient in C/C++, CUDA, Java, Javascript, MATLAB, MIPS, Python, R, and Swift. In addition, we have taken a wide array of upper-level mathematics and statistics courses, allowing us to leverage new and modern techniques for high performance computing, as well as adapt to applications from a wide range of fields. Our diverse background and wide range of skills will enable us to perform even better in the SC20 SCC.

While we tried to be as inclusive as possible without targeting any specific group for increased representation, our team's diversity is manifested in several ways. In total, our "American" team has representatives from three different nations. Aside from English, our team members fluently speak German, Japanese, Mandarin Chinese, and Spanish. In addition to computer science, all team members have performed interdisciplinary computational research in medicine, biophysics, computer vision, and German culture and linguistics. By having diverse backgrounds that focus on many different fields, our team can use different ideas from different fields to improve our approaches to solve problems in new, exciting ways.

Our team name is the "Daemon Deacons", a play on our university's mascot, the "Demon Deacons". In computing, a daemon is a process that works in the background and carries out critical processes.

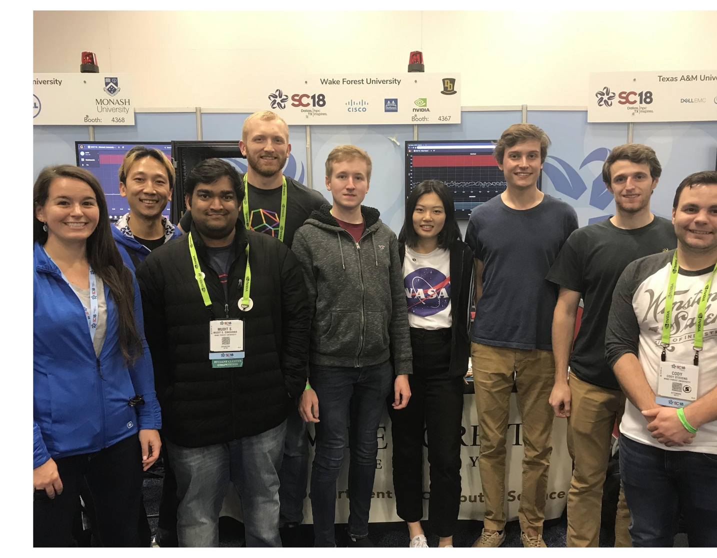
Wake Forest University is a liberal arts university with a predominantly undergraduate population that is currently listed on US World News Reports with a National Rank of #28. Located in North Carolina, Wake Forest University has about 5,000 undergraduate and 3,000 graduate students and prides itself on its low faculty student ratio (1:10). Two years ago, we were the first school from North Carolina to ever participate in the SC Student Cluster Competition, and we finished 2nd among the 6 first time-competitors. Last year, we came 1st in the reproducibility challenge. This year, we will be the first to win (see below, right).

Team Preparation

To prepare for the competition, the Computer Science Dept. created a HPC class in Fall 2017. It has been taught for 6 consecutive semesters so far by Prof. Samuel Cho (Associate Prof., Depts. Computer Science & Physics), who has extensive experience performing research and teaching classes in GPU Programming, Parallel Algorithms, Computational Biophysics, and Machine Learning. In addition, our team has three secondary advisors, Adam Carlson, Cody Stevens, and Dr. Sean Anderson who comprise the High Performance Computing Team at our university's DEAC Supercomputing Facilities (3.5k+ x86_64 compute and 12k+ GPU cores). All four advisors have extensive practical experience with compiling, running, and benchmarking applications in a high performance computing facility, as well as maintaining it.

To date, 24 unique undergraduate students have taken the High Performance Computing class across four (4) semesters, and many of them are repeat students. The last semester had 15 students, which is unusually high for an elective 1-credit course in our department (typical range is around 5-10 students). We hope that it is clear that we have a strong institutional investment and a strong set of instructors and a growing number of undergraduate students who are dedicated to the success of our team.

Since Summer 2020, team members have worked independently or in groups on building the run-time environment for each application and communicated their progress to the rest of the team on a weekly basis. The installation and test-drive for each application was first done on local virtual machines, then moved to DEAC before being tested on Azure.



Application Installation and Optimization

SPACK

Spack is a tool for package management that aims to simplify and automate the process of installing and setting up HPC applications and their dependencies. This allows us to quickly deploy our software on various SKU types to test application performance in an automated fashion. It also enables the automated swapping between various compilers and dependencies to automate optimization of compilers and dependency choices.

BEEGFS

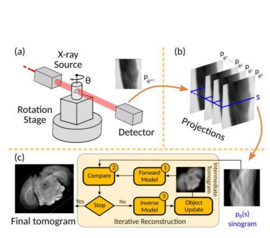
BeeGFS is a parallel file system developed and optimized for high performance computation. According to PVFS on Azure Guide, in addition to an easy installation procedure, BeeGFS also provides an IO-performance comparable to Lustre, normally the optimal file system, in terms of R/W throughput. Simple is the best.

Applications

Gromacs
fast
flexible
free

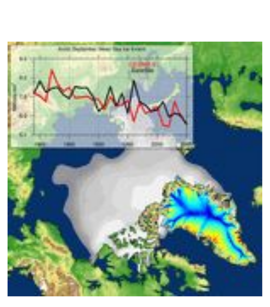
Gromacs

Gromacs is a program used for running molecular dynamics (MD) simulations and modeling protein dynamics and folding. Due to its design, Gromacs will benefit from parallelization. We plan to leverage GPU virtual machines like Azure's NC and NV-series, in addition to optimizations stemming from the nature of MD simulations and based off of recent research and trends in the field. We plan to utilize single precision over double precision, as is standard in the field, since publications have demonstrated a minimal drop in accuracy when using single precision. We also plan to utilize the Intel compiler and related libraries to compile and run Gromacs and CUDA to tune our build to the Intel hardware on Azure.



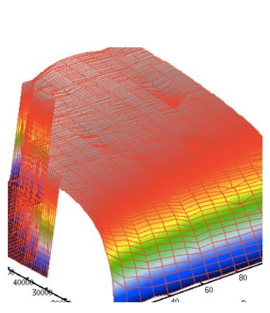
MemXCT

MemXCT is a memory-centric approach of X-ray computed tomography, which is used to computationally reconstruct 3D images from samples of 2D images taken at multiple projections. Since there is additional memory complexity associated with MemXCT, we choose to use memory-optimized virtual machines from Azure like the Ed v4 series. There are six tuning parameters, which we tune using the Bayesian optimization process. Since the purpose of the reproducibility report is to attempt to replicate findings, we explore cost efficiency as an alternative objective function to computation speed.



CESM2

CESM is a open source, fully-developed climate model that generates computer simulations of the Earth's past, present, and future climate states, including atmosphere, sea-ice, land, river, ocean land-ice and ocean wave. In the competition, we will choose to use Da V4 virtual machine from Azure to build and run the CESM2 Model. In the competition, we plan to find the bottleneck model and devoting more resource to that model for higher performance.



HPL

The High Performance Linpack is the de-facto standard for comparing HPC systems. The computation it benchmarks is the solving of large dense linear algebra systems in double precision. Since we have received an optimized GPU version of linpack, we prioritize SKUs with GPUs. This favors the NC and NV-series because of their NVIDIA Tesla K80 and NVIDIA Tesla M60 GPUs, respectively. Since we want to obtain the highest score possible, the measured floating point operations per second can be used directly as the objective function for the Bayesian optimizer.

HPCG

Short for High Performance Conjugate Gradients, this benchmark aims to provide context to HPL results. It tests access patterns that more closely reflect those of common HPC tasks, and in this way, incentivizes investment into their acceleration. As with HPL, HPCG can make use of GPU accelerators, thus indicating the use of NC and NV-series SKUs. Compared to HPL, the parameter space of HPCG is modest at 2 parameters, allowing us to use grid search to find the optimal parameter combination.

IO-500

IO-500 is a benchmark measuring I/O characteristics consists of multiple subcomponents. It measure both the performance of I/O bandwidth and metadata with IOR and MDTEST respectively. In this competition, we reviewed different file systems, including Lustre, BeeGFS, GlusterFS and then turned the parameters with different backends(posix, netcdf, hdf5, mpiio, etc.) for optimal IO-500 overall score. The hardware for IO-500, however, is going to give ways to other computationally-intensive applications as long as premium SSDs are accessible.

Optimization Strategies

On Azure, we have a vast parameter space of possible hardware configurations but also a limited budget for testing. We narrowed the options by analyzing application types and their access patterns. We have preliminarily chosen SKUs for each of the applications: HPL/HPCG (NCv3-series or NDv2-series), Gromacs (HC-series/NC-series), and CESM2(NC-series) are compute-intensive, IO-500 is storage-intensive (DSv2 series with premium SSD), and MemXTC is memory-intensive, requiring TBs of storage (E-series, HB-series).

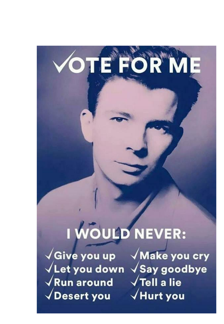
To further optimize our hardware configuration parameter search space, we employ a Bayesian analysis method. The algorithm forms a posterior distribution of gaussian processes to approximate the performance space and decide the next point in the parameter space to explore to maximize performance within the computing cost constraint. This approach is designed to require minimal sample runs, thus arriving at the maximum score with fewer Azure credits. We also use this approach to tune both the parameters for our applications, as well as the compilers and dependencies for their environment using Spack.

We have chosen SLURM as our resource manager because we have used it successfully in past competitions and because we are familiar with it from using it on Wake's DEAC supercomputing facility. We also utilized DEAC to prepare to build, test, and practice with our applications. Because DEAC features an array of CPU architectures and Nvidia GPU models, we have been able to get a feel for how the different applications respond to different environments, even before having access to Azure. Additionally, thanks to the flexibility and speed of Spack, we have been able to build several application software-dependency configurations on DEAC and test them to gauge performance benefits from specific configurations. As a result, we have been able to observe how applications perform under a range of conditions, tested on relevant input files and data, so we know these results are representative of how the applications will work during the competition, as opposed to just under testing conditions.

As for our choice in OS, we have chosen to go with Ubuntu as we have compared it with CentOS previously and found no discernable performance drop. We are also comfortable with it due to our experience using it in both our preparation during the academic year, as well as in prior competitions.

Due to the nature of Azure, we plan to utilize Spack for the vast majority of our software installation. Spack allows us to easily set options when we build software and export Spack recipes that allow us to save our configuration and duplicate the build on any other machine, down to every compiler and dependency used. On DEAC, we were able to utilize the shared file system to build our software and take advantage of the overlap in application dependencies. We plan to utilize Spack to duplicate this environment on Azure with libraries where it is applicable, storing it in a shared file system, and mounting that file system to each of our nodes. This will allow us to save time and credits, as we will not be required to build identical dependencies for identical architectures when we can build them once and share them.

Why We Will Win



One of our secret sauces is **preparation**. We have invested a lot of time and resources to make sure that we have the strongest possible team. Two years ago, we finished 2nd out of the 6 first-time competitors. Last year, we came 1st in the reproducibility challenge. After the SC19 SCC, the five undergraduate students who competed performed a SWOT (Strength, Weakness, Opportunity, Threats) analysis of their experience under the supervision of the primary advisor. From their SWOT analysis results, they identified a set of "action items" to be better prepared for the next competition. We spent the last year implementing them.

We are not only planning to win this year, we are also taking careful notes of the things that work and do not work so that we can learn to optimize ourselves too. We will overcome every failure and **persevere**. Why? <u>We are already planning on winning next year too.</u>

Acknowledgements

In addition to our team advisors and the Wake Forest University Computer Science Dept., we would like to thank Christine Harvey (Mitre Corp., Student Programs @ SC) who first introduced the Student Cluster Competition (SCC) to us. Also, we are grateful to Damian Valles for early support for our first application to the SCC. We are especially grateful for the generous anonymous alumni who funded our travel to the SCC.