SCC2021 : Team Death Valley Computing

Team Introductions:

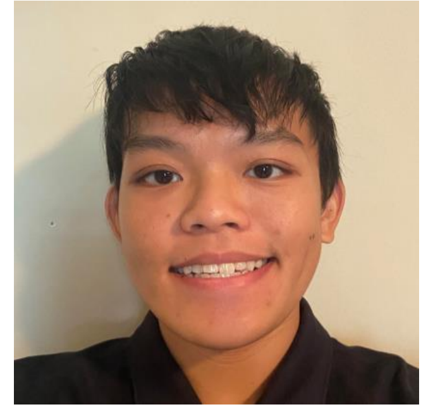
High-performance computing (HPC) has long been a priority and topic of research across several departments and colleges at Clemson University. Clemson students and faculty regularly use HPC to revolutionize their field of study. Clemson has assembled a diverse and highly competitive team for this year's Student Cluster Competition under the mentorship of Dr. Jon C. Calhoun.

Cooper Sanders

Majors: Computer Engineering, Electrical Engineering
Application Lead: HPCG
NSF REU Student and CUDA programmer. Multiple research internships in supercomputing and artificial intelligence.

Steven Lam

Major: Computer Engineering
Application Lead: Cardioid
SWE at Itron developing WebAPIs in C#. Heavily interested in SWE as well as DS&A and Systems Design



Alex Pendris

Major: Physics
Application Lead: Reproducibility
Involved in applying search algorithms to cellular automata. Looking to apply HPC knowledge to physics.

David Krasowska

Major: Computer Engineering
Application Lead: Quantum Espresso
Research in HPC lossy compression with Argonne National Laboratories. First generation Polish American college student.

Ethan Gindlesperger

Major: Computer Engineering
Application Lead: LINPACK
Research in data sampling and interning with Intel. Interested in computer architecture and electronics. System power management skills.

Sansriti Ranjan

Major: Computer Engineering
Application Lead: IO-500
Involved in Clemson IEEE Robotic team. Experienced in running IO-500 benchmarks. Learning about HPC and ML.

Team Preparation:

As a team, we:

- Attend weekly webinars to learn more about the applications we are running and become more familiar with the cluster.
- Test the competition applications in different configurations to obtain the best configuration for running.
- Hold weekly meetings to share our progress on application optimization and competition strategy.
- Utilize Clemson's Palmetto cluster to quickly iterate and test applications and environments

Hardware Configuration and Software Description:

	Model	Specifications
CPU	Intel Xeon Platinum 8352S	Clock Speed: 2.2G Core/Threads: 32C/64T UPI Speed: 11.2 GT/s Memory Type: DDR4 - 3200 MHz Cache: 48 MB TDP: 205 W
GPU	Nvidia A100	Clock Speed: 765 MHz Boost Clock: 1410 MHz Memory Clock: 1215 MHz Bus Width: 5120 bits Memory Size: 40 GB Total Board Power: 250W
Memory	512 GB DDR4-3200	Speed: 3200 MT/s Dual Rank 16 GB Base
Storage	960GB SSD	Speed: 6Gbps Size: 2.5in
NIC	Rockports	100 Gbps bandwidth

Software	Versions(s)
OpenMPI	3.1.6, 4.1.1
CUDA	10.2.89, 11
GCC	8.3.1
CMake	3.20.2
FFTW	3.3.8
MPICH	3.3.2
Python	3.6.8

Software Stack	
Rocky Linux	OS
ZFS	File system
OpenPBS	Scheduler

- Our software architecture resembles our Raspberry Pi clusters we built to practice for this competition and that of Clemson's Palmetto Cluster.
- This choice of software allows us to get assistance from CCIT and use systems and methods with which we are familiar.
- There is a large similarity with the Xeon CPU skews between our cluster and Georgia Tech's cluster used in the reproducibility paper. This allows for added comparability between the results.

Power Management Configuration:

- Power management tools will be controlled with **Intel PowerStat/cpufreqset** for CPUs and **nvidia-smi** for GPUs to handle the varying power supply limits. Estimate each node's base TDP is around 1600W and the full cluster to consume 4800W. We do not expect both the CPUs and the GPUs to be 100% active at any time.
- Our plan is to schedule when different benchmarks/applications are to be run based on the power budget at the current time. For example, if the power budget is low, we would likely schedule the IO-500 benchmark. This is because IO-500 is a CPU heavy benchmark, and CPUs are more power efficient when it comes to calculation. On the other hand, when the power budget is high, we will take full advantage of our slew of many high wattage graphics cards and run GPU heavy benchmarks like HPL and HPGC.

Strategizes for Running and Optimization:

Benchmarks:

LINPACK – Ethan Gindlesperger

- A dense matrix multiplication benchmark that makes heavy use of GPU and memory resources.
- Highly tunable for a variety of hardware.
- The homogeneous nature of cloud resources allows simpler tuning and faster end results (better optimization of P and Q).
- Utilize high performance GPU nodes with large GPU Memory capacity to reduce data transfer costs and provide the optimal environment.
- Optimize the split of CPU and GPU to have efficient parallelization to not hinder performance.
- Test performance of optimized binaries from Nvidia and Intel against building from standard source.
- Support from CCIT who has run HPL on the Palmetto cluster for TOP500 runs.

HPCG – Cooper Sanders

- Developed as a more relevant benchmark [compared to HPL] for practical HPC applications as HPCG has a memory access rate proportional to N, meaning performance is strongly influenced by memory bandwidth.
- Solves a system of sparse linear equations arising from a three-dimensional partial differential equation model using the conjugate gradient method
- The first CLI argument determines the local memory mesh for each rank (nx, ny, nz), which represents X*Y*Z points per MPI rank. The second determines ranks in the processor mesh (npx, npy, npz). This determines how effectively the ranks communicate
- For optimization, we explore both Jagged Diagonal and ELLPACK on GPUs to achieve efficient memory accesses.
- Explore message aggregation to avoid multiple sends between nodes

IO-500 - Sansriti Ranjan

- The IO-500 has been created to compare storage systems. It relies on community benchmarks while tracking storage performance.
- The community benchmarks are simulated based on IOEasy, IOHard, MdEasy, MdHard and Find.
- There are four phases for calculating the IOR and Mdtest scores - produce phase, consume phase, find phase and calculating the bandwidth and iops.
- Produce phase consists of write and create tests whereas the consume phase consists of read and stats test. Read and write is performed on IOR while create and stats is performed on MD. Find phase only runs the find operation. The Bandwidth comprises the four IOR Easy/Hard read or writes while iops comprises the mdtest, find.
- Using Lsv2-series instances with direct-mapped MVMe SSD storage for the fastest storage.

Applications:

Cardioid – Steven Lam

- Cardioid simulation is heavily parallelized, so it is a great application to take full advantage of HPC hardware like ours.
- Cardioid can very efficiently scale on GPUs using CUDA, and multiple nodes using MPI.
- We have a total of 12 A100 GPUs across 3 nodes, so we can leverage both MPI and CUDA.
- Cardioid can run efficiently on CPUs while leveraging OpenMP and SIMD registers, which can be useful when co-scheduling GPU computation or when power bounded.

Quantum ESPRESSO – David Krasowska

- Quantum Espresso is a suite for first-principles electronic-structure calculations and materials modeling requiring high computational power.
- Allows for the implementation of either CPU or GPU loads depending on the given configuration.
- We will focus on using our Nvidia Ampere A100 GPUs to efficiently demonstrate electronic-structure calculations and materials modeling at the nanoscale within Quantum Espresso, we will focus on using our Nvidia Ampere A100 GPUs.
- Having thousands of CUDA cores provided by the A100 GPUs will show us best performance for Quantum Espresso.

Reproducibility - Alexander Pendris

- ramBLe [1] is designed for learning on Bayesian Networks. For the reproducibility part specifically, finding Markov Blankets on the structure for given variables is the goal.
- There is only a CPU implementation, so the use of Xeon CPUs matches the CPUs used to test ramBLe. This allows for greater accuracy within the reproducibility of results. The constant factor time difference due to difference in CPU performances is not very relevant, as the focus is scalability, and our CPUs scale similarly.

Why will we win?

- We have a diverse group with complimentary experience in fields such as HPC and cloud computing.
- Our application leads have backgrounds suitable for handling the unique aspects of each competition application.
- Prior to the competition, the team will have local and remote SSH access to the cluster allowing our team to familiarize ourselves with the applications and benchmarks ahead of time.
- As a first-year in person team, we are prepared to be "All In" (a Clemson motto) and bring our best skills to win.
- We will use our power plan to effectively control our cluster at all points in the competition.

[1] Ankit Srivastava, Sriram Chockalingam, and Srinivas Aluru. "A Parallel Framework for Constraint-Based Bayesian Network Learning via Markov Blanket Discovery." In 2020 SC20: International Conference for High Performance Computing, Networking, Storage and Analysis (SC), IEEE Computer Society, 2020.