Final Architecture Proposal:

Hardware Configuration:

For the competition, our team has recruited two vendors (Cisco Systems, Inc. and NVIDIA Corp.) to lend us equipment from June to December 2019, which will span the summer and fall until the end of the competition in November at SC19 (see next section for details).

Cisco has provided one (1) C480ML-M5 server with 192GB DDR4 RAM, 980GB SATA SSDs, and (4) 1600W Power Supplies in 3+1 redundancy. The Cisco C480ML-M5 server comes with 8 Nvidia V100 GPUs with 32G memory in SXP2 form factor. In addition, they provided three (3) C240 M5 servers, each with 192GB DDR4 RAM, 980GB SATA SSD, and 770 W AC Power Supply.

The main motivation for this configuration is that the C480 server should be capable of performing serial and parallel computations with the *fastest communication possible*, depending on the types of applications provided. Regarding the communication, we believe intra-node will be much faster than inter-node communication. We are hoping to maximize all 8 Nvidia V100 GPUs for parallel processing and leverage NVLinks among the GPU communications.

Upon learning the software requirements for this year's competition, which were not as GPU focused as last year, we determined that our original single-node solution would not be advantageous during the competition. After further collaboration with our vendor, Cisco, we arrived at our final four-node solution, complete with a three smaller compute nodes, a single GPU node, and a 100G Cisco Switch. Our three Compute nodes are Cisco C240-M5's with 384GB of DDR4 memory, and two 18-core CPU's each (Intel Xeon Gold 6150 @ 2.70 GHz). The GPU node is a Cisco C480ML-M5 with eight nVidia V100 SXM2 built-in GPU cards, 768GB of memory, two 18-core CPU's (Intel Xeon Gold 6140 2.30 GHz), and 7TB of shared storage for the cluster. Across all four nodes, there is a total of 1.875TB of memory and 128 CPU Cores.

For highly parallelized applications, we will use the NVIDIA V100 SXM2 GPUs this year. This platform offers ~10% performance increase (TFLOPs) over the PCIe V100's and nearly 12x faster achievable bandwidth thanks to built-in NVLink 2.0 interconnects. With a single node with 8 NVIDIA GPUs, we will have thousands of cores that can perform parallel single-precision and double-precision computations for applications that are written in CUDA and can take advantage of the GPU architecture. Our approach is to assign a single MPI instance to single CPU process and the process is mapped to individual GPUs.

Software Configuration:

Our node will run the Ubuntu 18.04.3 LTS Server operating system. We also explored the Centos 8.0 operating system, and this was in fact our initial choice. However, the application installation was very cumbersome because it did not come with new and updated applications and libraries. As such, we constantly had to install prerequisite programs and libraries from source. For example, the GNU C++ compiler, gcc, is currently on v.4.8.2, but the latest is v9.2. We learned that this arose from different philosophies between Centos, which values stability and will therefore keep using older version software that are well-tested. Ubuntu, however, values the newest versions of applications more. Of course, we were also concerned about performance. When we ran HPL on Ubuntu and Centos, we found the performance to be negligibly different. Therefore, we decided to save installation time by going with the Ubuntu 18.04.3 LTS.

We will also have SLURM scheduler and resource manager to distribute the running of our applications so that we can allocate exclusive resources and prevent one application from taking over any one node. Further, we will monitor the execution of the application and limit the resources used by the applications to prevent our servers from exceeding the power usage limit. For monitoring, we plan to utilize the built in CIMC in combination with RRDtool for storage and graphing.

The undergraduate team members, as well as the primary and secondary advisors, have physical and remote access to the servers so that they can practice installing and benchmarking the applications from previous Student Cluster Competitions. Once an application installation is finished, we created a script that installs it so that 1) we have documentation of how the installation occurred, and 2) environment module files are generated so that anyone can load and unload the application as necessary while minimizing library conflicts. We set up a Network File System (NFS) so that we can share the same set of files across all four servers. All applications are

Communication will primarily be conducted through Slack (which was the main mode of communication for the students during the high performance computing class). Responsibilities and task assignment will be organized using Trello, a project management software, and students will either self-assign themselves to tasks or the primary advisor will assign them. Just prior to and during the competition, one of the veteran undergraduate students will be designated the team leader and take on this role.