UIUC Student Cluster Competition Proposal

About our team

The Student Cluster Competition (SCC) has brought together a highly diverse team of students from our school, joined by our mutual interest in high performance computing. While we are all first time competitors, we jointly possess knowledge of every facet of HPC. Justin has built his own mini-cluster out of Raspberry Pis and improves the Charm++ framework as part of his research. Nihar is a software developer with system programming and multithreading exposure. Edward develops MPI implementations for Charm++ as part of his research, and is well versed in relevant mathematics such as optimization. Yan has experience benchmarking systems and is very comfortable administering Linux. Wei's work on the Large Synoptic Survey Telescope familiarized him with real-time data processing and visualization. Jonathan has used the Blue Waters supercomputer in the past for a large scale data mining project, and learned CUDA for GPU programming for his current research. Our team also includes world-class professors who are currently performing research at the cutting edge of parallel computing and High Performance Computing and are offering mentorship. Finally, we have generous sponsors, Jump Trading, LLC and National Center of Supercomputing Applications (NCSA), who will financially support our team and provide us with the necessary hardware to build our best design.

Our team includes students majoring in *electrical engineering*, *computer engineering*, *computer science*, and *engineering physics*. The team was selected from a pool of over 30 applicants based on their coursework, prior experience, and interests and includes members of multiple ethnicities coming from three countries. One of our alternate team members is a computer engineering female student. We also approached some female students directly in order to involve qualified members from the underrepresented communities and asked ACM and IEEE student organizations to advertise about this competition. In conjunction with the preparation for this competition, we will create a registered student organization focused on HPC to attract students from a wide variety of backgrounds for future competitions.

Why we're participating

Our interest in the Student Cluster Competition stems from the role HPC plays in the modern world. We believe that HPC is and will always be a pivotal method of accomplishing the most demanding, challenging, and rewarding tasks. We hope to use this opportunity to improve our knowledge of the industry, both from applying what we learn in the competition and from connecting with experts in the field. Above all, designing a system to compete in this competition provides the opportunity to understand and solve the challenges of HPC, particularly in comparing hardware components and designing load-balancing schemes. For the competition, we will implement solutions to actual problems, providing both use cases and experience in real-world environments. Between researching and applying HPC techniques to optimize our cluster's ability to solve the applications, we hope to materialize and grow our proficiency in HPC.

Looking into the future, our members have high hopes of employing what we learn into research and design of HPC systems, as well as working in the field as system architects and computer scientists. Half of our team is interested in attending graduate school to improve the breadth of their HPC knowledge. Other members are looking to enter the workforce as hardware designers, systems engineers, and HPC practitioners. Attending this conference and participating in the competition perfectly aligns with our interests and long-term goals.

Why we will succeed

Our team has a number of advantages. Aside from our knowledge and abilities, we have an exceedingly diverse set of skills and experience that complement each other perfectly for the competition. We represent electrical engineering, computer engineering, computer science, and engineering physics majors, and the cumulation of these provides us with a comprehensive understanding of systems engineering. Our faculty advisors are highly knowledgeable about both the hardware and software side of HPC. Dr. Kindratenko is a Senior Research Scientist at NCSA with nearly two decades of experience with high performance computing. Professor Gropp develops algorithms for HPC to solve computationally difficult problems and is one of major MPI contributors. Professor Hwu is internationally renown for his work on parallelism, particularly on GPUs. Professor Kramer directs the Blue Waters supercomputing project and researches extreme-scale computing. Lastly we also have a graduate student advisor, Jiahui Yu, a participant in the 2014 SCC who is very familiar with all aspects of the on-site competition.

Aside from our advisors, we also have strong connections with our sponsors. Jump Labs, the research and development arm of Jump Trading at our campus's Research Park, is providing funds for the purchase of the system as well as for our travel expenses, training, and tools to help us succeed. Jump Trading will also provide guidance on the system design and training on cluster deployment based on its industry expertise in running large clusters for financial market research. The NCSA is providing space to work and deploy the cluster prior to the competition, as well as training in HPC and access to HPC professionals that can guide us in application development and code tuning. Questions about the architecture and NCSA sponsorship should be directed to Dr. Kindratenko and questions related to the sponsorship by Jump Trading should be directed to Mr. Tod Courtney.

While we are working with the vendors on the system design and acquisition, we will be using experimental HPC systems in the NCSA's Innovative Systems Lab (ISL) run by Dr. Kindratenko. One such system consists of eight QDR IB-interconnected nodes equipped with 8 AMD GPUs each. Another system includes both Xeon Phi accelerators and NVIDIA GPUs. These systems, as well as other equipment available in the ISL, will be sufficient to start investigating the characteristics of various architectures and parallelization strategies for the challenge applications. Each team member will be assigned to work on a subset of the challenge applications with other team members that will best utilize his/her prior knowledge and interests.

By the time of this competition, we will have extensive training and practice on developing for our own cluster. We will have experience in GPU programming from our Applied Parallel Programming course (ECE 408), MPI and OpenMP from Parallel Programming (CS 484), distributed memory systems from Distributed Systems (ECE 428) and Communication Networks (ECE 438), and experience in developing systems from Systems Engineering (ECE 391). Furthermore, we have taken Graph Theory coursework (part of CS 374) to aid in the ParConnect application, research in data analysis and visualization for Paraview, and Security coursework for Password Auditing/Cracking. Additionally, the university is providing us with a 3- or 4-credit hour individual study course (ECE 397: Individual Study in ECE, or CS 397: Individual Study) which we will utilize to work with advisors and to prepare for the competition. This time will be spent on-site at NCSA, building and benchmarking the system as well as developing or optimizing each of the applications. Ultimately, we believe we possess the connections, resources, and background to succeed in the competition and enjoy the conference.

About our system

Hardware

We have analyzed the competition applications requirements and concluded that no single compute hardware can provide best performance for all of them. Therefore, we challenge ourselves to build a hybrid system based on several accelerator types and use a part of it for each of the applications, depending on the strength of each type of accelerators. Our tentative plan is to use a combination of Intel Xeon E5-2699v4 CPUs and NVIDIA K80 GPUs for the HPL benchmark, due to their high double-precision performance (theoretical double-precision peak over 600 GFLOPS at 145W TDP and 2.9 TFLOPS at 300W TDP, respectively); AMD GPUs for visualization applications and for a hybrid implementation of the HPCG benchmark, due to their high single-precision performance (theoretical single-precision peak over 8 TFLOPS at 175W TDP); and CPUs for the graph application. The password recovery application can be best addressed by the multi-core CPUs and potentially by the AMD GPUs due to the high bandwidth memory (HBM). We will use a series of tests to determine the actual performance of the accelerators and CPUs, and select the best combination to use for each application.

We are planning to build a system with the following hybrid AMD/NVIDIA configuration:

Chassis	Colfax CX1450s-T-X6 1U Rackmount Server Base Platform	4x
Processor	Intel Xeon E5-2699V4 22C/44T 2.2Ghz 9.6GT/s 55MB 145W	8x (2 per node)
Memory	16384MB 2400MHz DR x 8 Registered ECC DDR4	32x (8 per node)
GPU	Nvidia Tesla Kepler K80 Computing Processor	4x (1 per node)
GPU	AMD XFX R9-NANO-4SF6	8x (2 per node)
Storage	Intel DC S3710 Series 400gb SSD SATA 6.0Gb/s	4x (1 per node)
InfiniBand	Mellanox ConnectX-4 VPI EDR Adapter and cable	4x (1 per node)
InfiniBand	Mellanox SB7790 EDR 100Gb/s InfiniBand Switch	1x
Ethernet	Built-in dual-port 10GbE	4x (1 per node)
Ethernet	Netgear XS712T 12 Port 10Gbase-T Switch	1x

For this configuration, we have calculated that either the CPU-NVIDIA GPU subsystem or the CPU-AMD GPU subsystem can draw power just under the limit of 2x1560 Watt. Since we will always be running only a part of the system at full load, we will use power management tools to keep unused components at their lowest power state in order to keep the total power below the power limit and will carefully transition from one subsystem to another. For example, we measured that it takes about 10s from the time the application exits for the single K80 GPU to go all the way down to the lowest power level (~28W). Also, four idle K80 cards, for example, will still consume about 7% of the total power. We will determine the power consumption of this system under various loads and develop the best strategy to use power efficiently.

The other option that we have been counter playing with is a Xeon Phi (Knights Landing)-based system. The Xeon Phi processors have very high performance for both floating-point operations (over 3 TFLOPS DP and 6 TFLOPS SP) and integer arithmetics, making them suitable for a wide range of applications. The challenge is that there are fewer libraries and applications optimized for the Xeon Phi architecture than

NVIDIA GPUs. We are planning to order pre-production hardware from Intel through their Developer Access Program to use for testing and training.

We will also consider using a high-performance, low-power FPGA PCIe-based accelerator for the password recovery application. Several team members have already taken Digital Systems Lab (ECE 385) and are proficient with FPGA design using SystemVerilog.

Software

We are looking to build a software stack that matches our system and delivers the best performance. For operating system, we will use CentOS, due to its good stability, well-written documentation, support for the GPU and interconnect hardware, and wide use at NCSA.

Some applications, such as Paraview, may require to work with a dataset of a considerable size. Since our cluster does not have a dedicated shared file system, we will consider combining SSDs across the nodes using a filesystem, such as OrangeFS.

We will also need an MPI implementation in order to run parallel jobs. Based on our experience, we prefer MVAPICH2 due to its better performance on InfiniBand fabric. Intel MPI is another option, especially if we chose the Xeon Phi-based system, since it is well-tuned for Intel products. We will also test OpenMPI and choose the best one based on performance.

In order to utilize the computing power of different types of hardware, we will use matching software for each of them. For Intel CPU (or Xeon Phi processors), we will use Intel's Parallel Studio XE suite, which includes Intel MPI, Intel Math Kernel Library (MKL) and the ICC compiler. Intel MKL provides optimized BLAS routines and other math functions for Intel hardware, and the ICC compiler can produce binaries better optimized for Intel hardware than the standard GCC compiler. For NVIDIA GPUs, we will use NVIDIA's CUDA Toolkit, which includes NVIDIA CUDA Compiler Driver (NVCC) and optimized versions of various scientific computing libraries. For AMD GPUs, we will use AMD's APP SDK, which provides OpenCL functionality to the GPUs. AMD's GPUOpen initiative also provides the HCC compiler, a C++ compiler specially designed for heterogeneous computing; and the HIP API, which converts CUDA code to C++/OpenCL code, making it easier to port CUDA applications to AMD GPUs. The team also has access to a collection of highly optimized CPU and GPU library functions produced by Prof. Hwu's research group.

We recognize that resolving conflicts between OpenGL and OpenCL libraries and kernel drivers with both NVIDIA and AMD GPUs in a single system could be a challenge, but the performance of the proposed system is very attractive.

To better understand the workings of our hardware and software and optimize the system for best performance and power efficiency, we will utilize a variety of monitoring and tuning software. We have Intel VTune Amplifier for Intel hardware, NVIDIA Nsight and NVIDIA Visual Profiler for NVIDIA GPUs, and CodeXL for AMD GPUs. We will work with our advisors, Jump Labs, faculty from the university and NCSA, and people from Intel, NVIDIA, and AMD to make the best use of our hardware and software resources.