Student Cluster Competition 2019

Final Architecture Proposal

Warsaw Team

Team Members:

- Dominik Psujek (captain)
- Łukasz Kondraciuk
- Tomasz Cheda
- Iwona Kotlarska
- Aleksandra Księżny
- Marek Masiak

Team advisor: Marcin Semeniuk (ICM at the University of Warsaw)

The team's proposal on hardware configuration is based on the requirements of the five competition applications and an overall analysis of various architectures' design. This time, a significant number of applications are CPU-intensive, so our setup relies on Intel Cascade Lake architecture.

Having learned from previous competitions, we decided to explore the parallel performance of competition applications and build our hardware configuration on five relatively strong servers. Two of them are accelerated with four NVIDIA GPUs each, three with lightning-fast NVMe drives instead. This configuration is a reasonable trade-off between performance and power requirements, especially considering that only benchmarks and, potentially, VPIC rely heavily on GPU performance.

Regarding the communication, we are using Mellanox EDR Interconnect - it is especially important in the SST and Reproducibility challenge, as these applications are particularly communication-intensive.

Application Descriptions

- **Reproducibility challenge: Computation of Planetary Interior Normal Mode:** This year's paper is focused on developing a new algorithm for calculating Normal Modes of elastic-gravitational systems. The Finite Element Method is used to approximate and reduce this task to a generalized eigenvalue problem. Finally, Lanczos approach along with polynomial filtering is used to extract the exact eigenfrequencies.
  Speaking of the program: among dependencies, we can find ParMetis, Blas/MKL and MPI, which is a pretty standard set for a physics simulation. Unfortunately, because some parts of the source code are written in Fortran, a deep analysis of how the application works internally is a bit hard and time-consuming.

The main challenge of the task is to present scalability of a large number of parallel Sparse Matrix Vector multiplications (SpMV) in a distributed memory model. This operation is communication-intensive, so we had to keep an eye on the performance of the cluster interconnect and MPI implementation while choosing final hardware and software stack.

- **VPIC** is an application that uses a particle-in-cell model to simulate movements of particles in electric and magnetic fields. It helps scientists understand many plasma physics phenomena. The simulation volume is divided into a voxel mesh, and plasma is represented as multiple particles that are acted on by the fields, and those fields are updated using electrical currents accumulated from particle motions.
  Because of the massively parallel nature of performed operations, we believe that there is a great potential in porting VPIC simulation to CUDA. Or at least in optimizing it in regards of SIMD instructions available on modern Intel Xeon Skylake processors.
- **SST** is a simulation framework designed in a way that allows it to be used for simulations in various domains, as long as it can be represented by elements interacting with each other by sending messages via predefined links. This flexible design allows it to be used for modelling systems from the level of detailed processor and cache interaction simulations to whole HPC systems with a wide choice of granularity. Our analysis and early experiments have shown that it scales very well with multiple processes. By profiling it, we have determined that a common feature of some very different workloads is that they are communication-intensive, and this has influenced our choice of interconnect hardware and overall design.
- **HPL and HPCG Benchmarks** both require a lot of computational power, but only in terms of resource usage. Unlike IO-500 benchmark, they do not rely on IO, but only on computing performance and data transfer rate. Since we want these benchmarks to achieve the best results, we require a lot of GPU power for running linear algebra computations and also CPU power to supply the accelerators with enough data to attain their full performance. For the competition, we are going to use executables optimized for Volta architecture provided by NVIDIA.
- **IO-500 Benchmark** consists of data and metadata benchmarks. Its goal is to identify performance boundaries for optimized and suboptimal applications. For identifying the first ones responsible are so called "easy" tests and as for the second ones they are called "hard" tests. They are meant to show worst-case scenarios. The final score consists of the results of bandwidth score (geometric mean of easy and hard IO tests) and metadata score (geometric mean of easy and hard metadata tests). We intend to use several fast NVMe disks, and as such we hope to get a high score on this benchmark since its results depend on chosen storage.

Hardware Configuration

- The aim of our cluster design is to achieve the best possible performance while balancing on the power usage limit, keeping also in mind various needs of the applications as well as the power shutoff activity.
- The cluster consists of five compute nodes, 2 with 4 NVIDIA Tesla V100 GPU

accelerators, 3 without, each one with 2 Intel Xeon Cascade Lake multiple-core processors as well as 384GB of DDR4 RAM. This setup gives us a lot of performance (about 55 TFLOPS peak) to compete with all the other teams in terms of computational speeds in all applications, as well as balanced power efficiency.

Software Configuration

- The cluster runs CentOS 7.7 as the operating system. The reason for this choice is stable support for all the hardware and provides a lot of useful tools and libraries.
- We also considered recently released CentOS 8, but it's freshness also means there may not be enough support from libraries and utilities we may need for the competition.
- Compilers and libraries: we use the latest version of Intel Parallel Studio XE 2019, and a set of GNU Compilers, the 8.3 release. This well-tested combo lets us get the best possible performance in every application task.
- For the communication we are going to use OpenMPI/HPC-X libraries and for the GPU accelerated ones - CUDA 10.1 framework, which supports the NVIDIA Volta architecture.
- Because the competition is long, we also use SLURM Workload Manager, 19.05 edition, to queue and smoothly run all our tasks.
- To run applications and benchmarks on our cluster, we use the MooseFS shared file system. It has proven itself resilient and failure tolerant in previous competitions, and this is exactly what is required especially considering Power Shut-off activity.

Final Architecture Design:

- 2x Lenovo SR670
  - 2x Intel Xeon Gold 6230 20C 125W 2.1GHz Processor
  - 4x NVIDIA Tesla V100 32GB
  - 384GB RAM
  - Mellanox ConnectX-4 VPI Card

- 3x Lenovo SR630
  - 2x Intel Xeon Gold 6230 20C 125W 2.1GHz Processor
  - 384GB RAM
  - 2x U.2 NVMe 1TB
  - Mellanox ConnectX-4 VPI Card

Power budget estimates

	Item	Specification	Power	Qty
Server	Lenovo SR670	- 2x Intel Xeon Gold 6230 - 384 GB DDR4	Stress: 400 W Idle: 200 W	2
	Lenovo SR630	- 2x Intel Xeon Gold 6230 - 384 GB DDR4 - 2x NVMe 1TB	Stress: 350 W Idle: 200 W	3
GPU Card	NVIDIA Tesla V100 PCIe	- 5120 CUDA cores @1455MHz - 32 GB memory	Stress: 250 W Idle: 15 W	8
Network	Mellanox ConnectX-4 VPI Card	InfiniBand EDR 100Gb/s interconnect	15 W	5
	Mellanox EDR IB Switch	36 port Mellanox InfiniBand EDR 100Gb/s switch	135 W	1
	1 GE Switch	Gigabit Ethernet switch	30 W	1
Estimated Power	CPU mode	5 nodes: 2x CPU + idle GPU	2*400 + 3*350 + 8 *15 + 5*15 + 135 + 30 = 2210	
	GPU mode	2 nodes: 2x CPU + 4x GPU 3 nodes: 2x idle CPU	2*400+8*250 + 3*200+5*15+135+30 = 3640	