Warsaw Team
ICM UW

SC17 – Student Cluster Competition

ICM – University of Warsaw

Interdisciplinary Centre for Mathematical and Computational Modelling (ICM) is one of the first High Performance Computing Centres in Poland, founded more than 25 years ago as a non-departmental unit of the University of Warsaw, located in the capital of Poland, Warsaw. ICM is also a research centre for interdisciplinary sciences. As a HPC centre ICM operates supercomputing infrastructure with combined computational power of more than 2 PetaFLOPs.

Team introduction

Marcin Semeniuk – ICM, University of Warsaw

I received my MSc in Computer Science at the University of Warsaw in 2006. Paradoxically, while working for 10 years as System Administrator in the HPC center, I had almost no contact with HPC. This changed last year, when Warsaw Team was formed and I started helping Maciej Cytowski to prepare them for ASC17. Now for SC17 I'm co-mentoring Team with Maciej Szpindler and we hope we will have a chance to continue this adventure next year as well.

Adam Sobecki – University of Warsaw

I am a sophomore studying Computer Science at the University of Warsaw. I am proficient in programming languages such as C/C++, CUDA, Java, Python, Haskell, OCaml and really love problem solving. My major areas of interest in Computer Science are High Performance Computing, Machine Learning with Deep Learning and Algorithms and Data Structures. I also try to keep up-to-date with new technologies. My hobby is scuba diving and swimming.

Dominik Psujek – Warsaw University of Technology

I'm a 2nd year Computer Science student from Warsaw University of Technology. I enjoy reading books of any kind, but mostly science fiction, fantasy, crime novels and thrillers, as well as watching movies. I also like playing platformer and arcade games. But my passion is tinkering with everything possible, hardware, software, even mechanical things not connected in any way to computers or microprocessors. From IT field, I really like administrating networks, servers and systems (even small ones).

Jarosław Ławnicki – University of Warsaw

I'm a third year Computer Science and Mathematics Bachelor's student at the University of Warsaw. I really enjoy problem solving especially in interesting areas which are a close mix of programming and Math such as HPC, machine learning and algorithms. My main programming languages are C++ and Python but I also enjoy coding in functional languages.

Marcin Mielniczuk – University of Warsaw

I'm a 3rd year Computer Science and Mathematics Bachelor's student from the University of Warsaw. I really love problem solving. My favorite programming language is Rust and I'd love to see more of it in HPC. I'm a great Linux and open-source enthusiast. I'm really fond of the idea of encryption and I'd be really disappointed if it were P = NP! I told about my scientific interest and what about my spare time? I really love reading books and often play the guitar. Besides, I'm really fond of long walks in the wild, off the beaten track.

Piotr R. Konopelko – Łódź University of Technology

Apart from my work (which consumes most of my day time having fun with the best Distributed File System on the world, MooseFS :) I love driving a car. I enjoy long trips outside Warsaw when my eyes rest, away from the computer. In summer I often sail with my friends and we definitely have great time. I also have many other activities, including volunteering in one of student organizations ran by Warsaw University of Technology - Student Internet Television TVPW. In TVPW we produce video podcasts about student life in Warsaw and live broadcasts from university events. We do it non-profit, but we get lot of fun and share our experiences with each other.

Szymon Pajzert – University of Warsaw

I'm a third year student studying at the University of Warsaw. My main ares of interest are programming languages and optimizing compilers. I've got interested with them when introduced to functional programming and now I'm working on how to combine efficiency with descriptiveness and composition in a language. Joining SC17 is part of this effort.

Architecture Overview

	Item	Specification	Power
Server	Supermicro–based	2x Intel Xeon E5-26xx v4 8 x 16 GB DDR4 2x 240 GB Solid State Drive 1x NVMe Solid State Drive	Stress: 350 W Idle: 120 W
GPU	4x2 - NVIDIA Tesla V100 PCIe	5120 CUDA cores @1455 MHz 16 GB memory	Stress: 250 W Idle: 15 W
Network	InfiniBand 100 Gb/s interconnect 1 Gigabit Ethernet switch Mellanox InfiniBand 100 Gb/s switch		15 W + 30 W + 135 W

Hardware

Our cluster architecture consists of 8 NVIDIA Tesla V100 PCIe GPU accelerators distributed across 4 compute nodes cooled with air. Theoretical total performance of the GPU cards is $8 \cdot 7 = 56$ TFLOPs. The number of used cards delivers compromise between performance and power consumption, but could be slightly better. We tried to obtain another V100 to be able to get up to 63 TFLOPS with 3x3 architecture, but we are not as good in finding free GPUs as we are in writing posters. Our goal was to create water cooled cluster with 9 NVIDIA Tesla V100 SMX2 GPUs, with one cluster node less providing us with enough power efficiency to not need to underclock GPUs.

Power Consumption

- Estimated power consumption in CPU mode
  4 nodes: 2x CPU
  4*(350+15) + 8*15 + 30 + 135 = 1395 W
- Estimated power consumption in GPU mode (1 CPU core per node)
  4 nodes: 2x CPU + 2x GPU

  4*(120+15) + 8*250 + 30 + 135 = 2820 W

To meet power consumption limits we're going to slightly underclock the GPUs. We also removed redundant fans from our server to gain around 25 Watt from each node.

Performance

We base our architecture on GPUs, because both benchmarks, MrBayes, LAMMPS and hopefully the Mystery Application can be run easily on those units. We've done our best to efficiently port Born to GPU as well, but the final results will depend on provided datasets. GPUs provide a great ratio of speed to power consumption. In order to maintain all potential we've equipped our cluster with 2 CPU units per node as well InfiniBand EDR 100 Gb/s interconnect

Competition Applications

LINPACK and HPCG Benchmarks

We use binaries provided by NVIDIA to utilize our GPU oriented architecture. Since we weren't able to access V100 before the competition, we had to optimize the benchmarks while setting up the cluster.

Reproducibility Challenge - Vectorization of Tersoff

The reproducibility challenge demands from us to try to replicate the results on a slightly different architecture than described in the paper. In our case we are making use of NVIDIA Tesla V100 GPUs and Intel Xeon E5-2680v4 processors. Our approach We had to compile the code of the vectorized version of Tersoff potential for our architecture, which was challenging, because we also used newer versions of the compilers (both Intel MPI and CUDA). We also put an effort on providing a scalability so that we could run the application on many nodes instead of just one. We also considered using a newer version of LAMMPS, but we resigned this idea because the code had changed a lot during a year and merging changes of the vectorized version with the current version wasn't the best choice and it would go beyond the task of the results replication. However, we strongly believe that it could have a good influence on the performance of the application.

MrBayes

With strong GPU architecture we aimed from the beginning to use beagle library for GPU devices. To handle many test cases at once, our computations are running on single nodes instead of whole cluster. This allows us to adjust load per node and possibly share it with smaller workload from another application.

Born

First we tried to port Born to openACC to utilize our experience with it from ASC17. It turned out fast that it won't be that easy because of lack of compatibility with object oriented code. Having problems with handling even very simple code snippets we switched to CUDA. We optimized Born on both CPU and GPU in case optimizations of the latter were not going to work.

Figure 1: Alternative power source

Cloud Component

Since our architecture is suitable for most of the applications, we're going to treat this component as subsidiary to our server. In case of inefficiency of Mystery Application or either Born or MrBayes we can then move blocking workloads to the cluster.

Power Shutoff Activity

We have attached NVME SSDs to each node to be able to save checkpoints of the calculations in case of power shutdown. Since Tersoff and MrBayes already use checkpointing to save results of computations, only Born and Mystery Application are vulnerable to power shutdown. However, after considering advantages and disadvantages we decided to not add an overhead to our cluster and in need run bigger workloads on provided cloud architecture.

This project is co-financed by the European Union's European Social Fund. The sole responsibility of this publication lies with the author.
The European Union is not responsible for any use that may be made of the information contained therein.