TEAM dEFAULT GERMANY

Our Team

We are representing the Friedrich-Alexander University Erlangen-Nuremberg. To assemble the best possible team, we contacted students from all over the technical faculty to join us. Now we have a solid team with members of different areas, genders and origins, that together will provide all necessary skills for cluster computing.

Our team members are studying computer science, which provides a solid foundation for programming, cluster computing, and systems administration, and computional engineering, a degree course that combines computer science, numerical and applied mathematics.

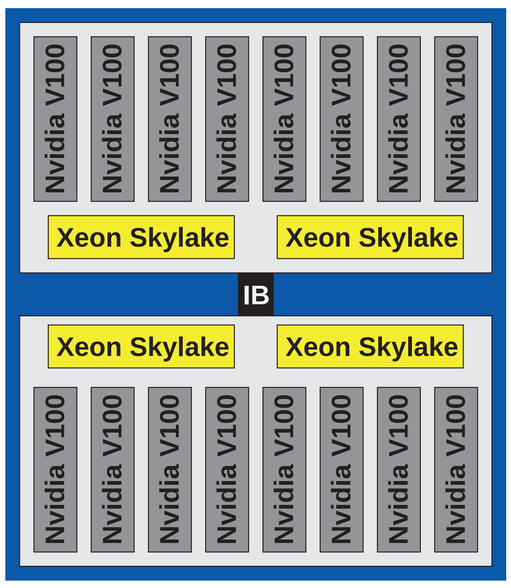
Although women are typically underrepresented at the technical faculty of our university, we are proud to have two in our team. Also one of us is an international student from Ukraine, which makes our team even more unique.






name:	David Sauerwein	Eva Dengler	Oleksandr Bannov	Meike Bloecher	Marcel Pabst	Manuel Peschel
subject:	computer science	computer science mathematics,	computer science information	comp. engineering simulations,	computer science industrial elec-	computer science computer
skillset:	UNIX systems administration	theories	theory, NVIDIA fan	mathematics	tronics technician	architecture
tasks:	OpenMC Horovod	OpenMC Horovod	Reproducibility Horovod	HPL/HPCG Reproducibility	HPL/HPCG Reproducibility	OpenMC HPL/HPCG

Hardware and Software



Hardware

We selected Intel Skylake CPUs because they are the latest generation and have the advantage of six memory channels. Compared to the older versions with only four channels it provides faster memory access. In addition Skylake CPUs also support AVX512 which leads to higher performance.

To keep the node power overhead low, we decided to use as many GPUs in one node as possible, thus using as few nodes as possible. This consideration leads to eight V100 GPUs in each of our two nodes. Furthermore we use the SXM2 version of the card, due to its higher interconnect bandwidth which is 16 times higher than the PCIe version. For using PCIe there had to be more than two CPUs per node to use the full potential of the V100 cards because two Xeon Skylake CPUs only have two times 48 PCIe lanes and the eight GPUs need eight times 16 lanes.

Additionaly, the HBM2 technology provides superior memory bandwidth, which we will need in the HPCG benchmark. The Nvidia V100 provides us with a very high double precision performance for the HPL benchmark and its tensor cores will be an advantage for the deep learning application.

Software

As far as software is concerned, we plan to go with software that has worked for other teams in the past.

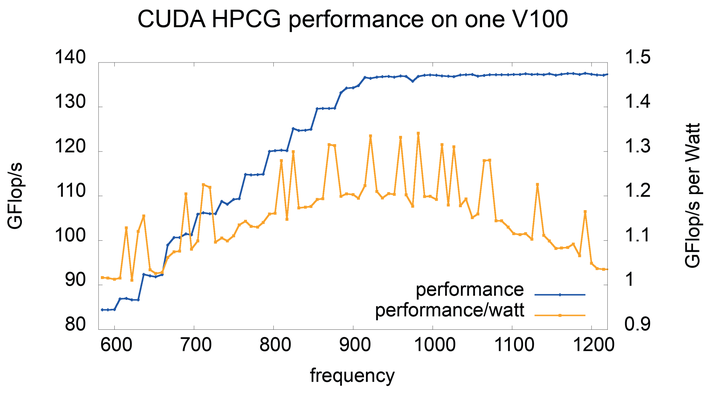
This includes using:

- CentOS as free RedHat-compatible UNIX based operating system using package management with RPM
- Ansible to manage packages and operating system configuration on the cluster nodes
- HPE's CMU and iLO for the cluster and node management
- SLURM as batch scheduling system to handle the application workflow
- the Module system to manage different software versions, such as different versions of the Intel and GNU C compilers and different MPI runtime environments
- git for version management

Preparations

HPL and HPCG

Over the last six months we learned a lot about the HPCG and HPL benchmarks. Our findings of HPCG are distilled into the graph below. Do not hesistate to ask us about the graph - we are glad to explain the results to you.



Horovod

Horovod is not just a Russian circle dance, but also a distributed training framework for Tensorflow, which implements allreduce to achieve more efficient inter-GPU communication via ring reduction. Because of this, the application scales extremely good on multiple GPUs. In preparation for the competition we read about the principles of deep learning, tested various neural networks like ResNet50 and ResNet100 and compared the different performance of the networks to gain a solid background knowledge.

This will help us to make the right decision when it comes to choosing a network for a specific task of the competition.

OpenMC

OpenMC is a Monte-Carlo particle transport simulation code focused on neutron-critically calculations.

To optimize the outcome, we did more than just use the prebuilt version from the repositories. We tried to compile the source with many different compilers, MPI variants and parallel HDF5 versions.

To make sure our results are correct, we compared our results to the official binaries.

Because of the compiler optimizations of the Intel C and Fortran compilers the test suite failed, although the results are within the error range of the probability distribution.

Reproducibility

The reproducibility challenge is quite a difficult application for us, due to the fact that the results in the paper are based on a large number of nodes. Definitely we do not have enough nodes to reproduce all results.

Hence we need to extrapolate some of our results to match those in the paper. To save us time at the competition itself, we decided in advance which exact results we want to reproduce and wrote suitable scripts.

This gives us the advantage that we only need to start them at the competition, receive the results and spend more time on solving other problems. We know our cluster well so we can explain all of our achieved results in consideration of our configuration.

Power Management

Because our task is to get the maximum performance while not using more than 3000 W, we have to keep an eye on the power consumption of our system.

First of all, we tested the applications to determine the parameters which leads to the highest performance while staying below the power limit. For the worst case we will set a power cap in our BIOS, so we will hopefully never get over the hard power limit and be punished.

Moreover, we need to schedule when which application should run for how long. E.g. if Horovod takes longer than expected for the needed accuracy, we will give it more runtime, which has to be taken away from another application.

So we will prevent running two applications at the same time, as neither application can achieve decent results as it can not access the systems resources exclusively.

Why We Will Win

Our goal in this competition is to win and we think we can accomplish that because we have great support from the chair of computer architecture at the FAU, who gave us a general introduction to the setup of a cluster and high performance benchmarking. Another reason for us to win is that the Regional Computiong Center Erlangen gave us access to their clusters so that we could install and try out some of the applications before we get our own cluster. This way we already know how to work with a bigger cluster. Then of course we have a cluster, which is optimal for GPU applications. And as there are more GPU than CPU appliations we will profit from the higher flop rates of the GPUs and our high memory bandwidth.

Our team also has a broad knowledge in very different subjects throughout the field of computer science and cluster computing.

Every team member is highly motivated to work hard at the competition and use the knowledge we gained in the preparations over the last half year to win this competition.

Cloud Resources

We mainly want to use the cloud recources to compensate the few CPUs we have in our own cluster.

Because OpenMc does not have a GPU accelerated version, we plan to run the calculations of the neutron transfer simulation in the cloud, using nodes with a high amount of CPUs in it.

Since we do not know whether the mystery application will perform well on our system or not (because of the lack of CPU power in our system), we also consider running it in the cloud for better solutions.

In preparation for the contest we wrote a script which makes the set up of the cloud as fast and convenient as possible. So creating a new cluster instance in the cloud will only take a few clicks.