Optimization of Ultrasound Simulations on Multi-GPU Servers

Filip Vaverka
Brno University of Technology, Faculty of Information Technology, Centre of Excellence IT4Innovations
Brno, Czech Republic
ivaverka@fit.vutbr.cz

Matej Spetko
Brno University of Technology, Faculty of Information Technology
Brno, Czech Republic
mspetko@gmail.com

Bradley E. Treeby
University College London, Medical Physics and Biomedical Engineering, Biomedical Ultrasound Group
London., United Kingdom.
b.treeby@ucl.ac.uk

Jiri Jaros*
Brno University of Technology, Faculty of Information Technology, Centre of Excellence IT4Innovations
Brno, Czech Republic
jarosjir@fit.vutbr.cz

KEYWORDS

Ultrasound simulations, Local Fourier basis decomposition, k-Wave toolbox, Multi-GPU systems, CUDA, MPI.

1 INTRODUCTION

Realistic ultrasound simulations have found a broad area of applications in preoperative ultrasound and photoacoustic screening [7, 8] as well as non-invasive ultrasound treatment planing [1, 10]. However, a typical medical simulation requires a set of partial differential equations to be solved over a domain with more than 1024^3 grid points for tens of thousands of simulation steps. Five years ago, the only architecture offering a sufficient amount of compute power, and more importantly, sufficient main memory, was CPU-based clusters[3]. At SC 2017, we presented a poster on a GPU accelerated simulation code reaching almost linear scaling on a cluster of 512 single-GPU nodes of the Piz Daint supercomputer.

The current trend in GPU accelerated computing is towards the use of fat nodes with multiple GPUs per node, as recently seen in Sierra[1] and Summit[2] supercomputers. The performance of such systems is stunning. However, the complex node architecture with multi-level NUMA places high demands on the developers to properly orchestrate the intra-node communication.

This paper investigates the benefits of CUDA-Aware MPI [5] and CUDA peer-to-peer transfers [6] on such a multi-GPU node. Our system is based on a dual socket PNY server equipped with Intel CPU E5-2620v4 processors, 2×256 GB of main memory, and 8 Nvidia Tesla P40 Pascal GPUs, each with 3840 CUDA cores and 24 GB of memory. The 8 GPUs are divided into two quads connected to particular CPUs by 32 line root PCI-Express hubs. The quads are further split into pairs connected via second level PCI-Express hubs. The communication between CPU sockets/root hubs is enabled by the Intel QPI technology.

Combined, the server has a total GPU memory of 192 GB, and a theoretical single-precision performance of 96 Tflops. This is comparable to 1,200 Intel Haswell cores at 2.4GHz.

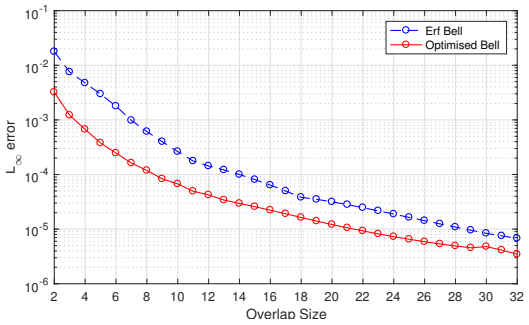

Figure 1: Numerical error introduced by a single interface between two subdomains for various overlap sizes and bell shapes.

2 METHOD AND RESULTS

The multi-GPU version of the k-Wave acoustic toolbox is based on local Fourier basis domain decomposition [2] where the 3D simulation domain is partitioned into rectangular cuboid blocks assigned to particular GPUs [4]. The partitioning can be done in one, two or three dimensions. The communication is done over the nearest neighbors on a 1/2/3D torus by exchanging the overlaps including slabs, edges and corners. The size of the overlaps is determined by the required precision and the shape of the bell function enforcing periodicity on local subdomains, see Fig. 1. For medical applications, the error on the order of 10^{-3} is acceptable.

The whole simulation is executed on GPUs using the CUDA FFT library [9] and custom CUDA kernels. The CPU is only responsible for controlling the simulation, progress reporting and storing simulation data. The simulation code uses a non-blocking MPI framework to exchange overlaps amongst subdomains, fast DMA transfers to download/upload overlaps from/to a particular GPU, and CUDA kernels to pack/unpack the overlaps into/from linear buffers. With this framework, we have demonstrated almost linear scaling on GPU clusters with a single GPU per node, e.g., Piz Daint[3] or Anselm[4].

*corresponding author.
[1] 4 Volta GPUs per node, Lawrence Livermore National Lab, USA
[2] 6 Volta GPUs per node, Oak Ridge National Lab, USA
[3] Up to 512 Nvidia P100 GPUs, CSCS, CH
[4] Up to 20 Nvidia K20m GPUs, IT4Innovations, CZ

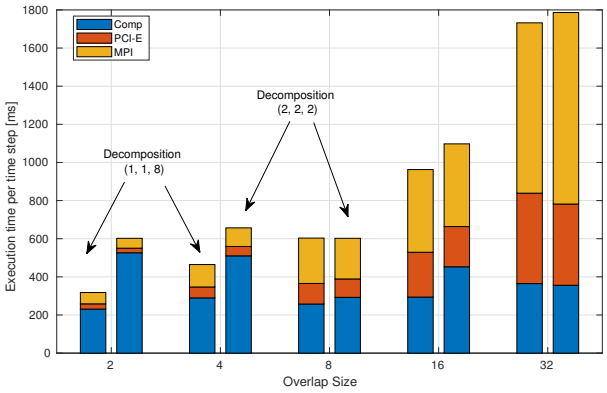

Figure 2: Simulation time breakdown of a single time steps for a domain size of 1024^3 grid points (dataset of approx. 120 GB). The left bar is for a 1D decomposition while the right one is for a 3D one.

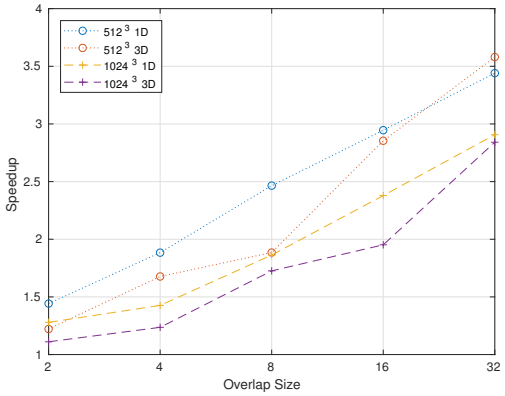

Figure 3: Speedup achieved by replacing standard MPI by CUDA-Aware MPI with peer-to-peer transfers for different domain sizes and decompositions.

When testing this framework on an 8-GPU server, we observed rather poor performance, see Fig. 2. Here, a 1D decomposition over the Z dimension into $1 \times 1 \times 8$ blocks and a 3D decomposition into $2 \times 2 \times 2$ blocks are investigated. We can see a significant increase in the MPI and PCI communication for bigger overlap sizes, especially for 16 and 32 grid points. Here, the communication takes between 66 and 80% of the simulation time. The difference in communication time between the two decompositions is given by a different number of neighbors (2 vs 27) and the size of the overlaps. The variation of the computation time between different decompositions and overlap sizes is caused by the size of the subdomains and their corresponding highest prime factors, which strongly affects the performance of the FFT.

The key for good communication performance is to minimize the number of message copies during the overlap exchange (GPU $\rightarrow$ CPU $\rightarrow$ QPI/Net $\rightarrow$ CPU $\rightarrow$ GPU). The first technique investigated employs CUDA-Aware MPI which reduces the number of data copies by directly taking the overlaps from the GPU memory and sending them to the other GPU without staging in the main memory. In our case, this technique has the potential to reduce the number of copies by a factor of two. Moreover, it is generally applicable in cluster environment even on single GPU nodes. However, in multi-GPU configuration, a proper CPU and GPU binding is crucial. For example, a favorable binding under 1D toroidal decomposition only requires 4 overlaps to travel over the Intel QPI. However, all 32 overlaps have to do so in the worst case when the neighboring domains are distributed over the PCI-E hubs in a round robin fashion. In our implementation, the subdomains are topologically distributed over GPUs in a way that minimizes the data transfers over the Intel QPI.

The second technique is direct peer-to-peer transfer between GPUs bypassing MPI. This allows overlaps to be transferred under the same PCI-Express hub by means of CUDA direct memory transfers without the assistance of the CPU. In this case, a proper binding is a must.

The replacement of the standard MPI[5] by a CUDA-Aware MPI has a great impact on the communication time, see Fig. 3. The reduction of the simulation times depends on the domain size, decomposition type, and the overlap size. Since the computation dominates for small overlap sizes of 2 and 4 grid points, the reduction in the overall simulation time is small, yet the speed-up reaches a factor of 1.8. When running realistic simulations, a typical overlap size is between 8 and 16, and in simulations with many reflections even 32 grid points. Under these circumstances, the reduction of communication overhead is eminent. The overall simulation time is reduced by a factor of 2–3.6. The combination of CUDA-Aware MPI and the peer-to-peer transfer only leads to a marginal reduction in the communication overhead. The speedup over the CUDA-Aware MPI in the most favorable configuration is only 6%. This is very likely given by the CUDA-Aware MPI already using the peer-to-peer transfers when applicable.

3 CONCLUSIONS

The main contribution of this paper has been to experimentally evaluate the benefits of CUDA-Aware MPI and peer-to-peer transfers on an 8-GPU node. Under a proper process binding and domain distribution, our experiments with 1D and 3D data decompositions have shown a significant speed up which linearly grows with the overlap size. This allows higher precision to be achieved with similar computational requirements when bigger overlaps are used. The highest speed up observed attains a factor of 3.6, which far exceeds the expected values of about two (half of the message copies were removed). One dimensional decomposition reached higher acceleration than three dimensional one. This is caused by fewer but bigger messages, which better utilize the PCI-Express and QPI bandwidth. Peer-to-peer transfers show only a marginal difference to the CUDA-Aware MPI.

[5] OpenMPI 1.10.7 without CUDA 9.2 support

4 ACKNOWLEDGEMENT

This work was supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project IT4Innovations excellence in science - LQ1602" and by the IT4Innovations infrastructure which is supported from the Large Infrastructures for Research, Experimental Development and Innovations project IT4Innovations National Supercomputing Center - LM2015070". This project has received funding from the European Union's Horizon 2020 research and innovation programme H2020 ICT 2016-2017 under grant agreement No 732411 and is an initiative of the Photonics Public Private Partnership. This work was also supported by the Engineering and Physical Sciences Research Council, UK, grant numbers EP/L020262/1.

REFERENCES

[1] Osama Al-Bataineh, Jürgen Jenne, and Peter Huber. 2012. Clinical and future applications of high intensity focused ultrasound in cancer. *Cancer Treatment Reviews* 38, 5 (2012), 346–353. https://doi.org/10.1016/j.ctrv.2011.08.004

[2] M. Israeli, L. Vozovoi, and A. Averbuch. 1993. Spectral multidomain technique with Local Fourier Basis. *Journal of Scientific Computing* 8, 2 (jun 1993), 135–149. https://doi.org/10.1007/BF01060869

[3] Jiri Jaros, Alistair P. Rendell, and Bradley E. Treeby. 2015. Full-wave nonlinear ultrasound simulation on distributed clusters with applications in high-intensity focused ultrasound. *International Journal of High Performance Computing Applications* 30, 2 (may 2015), 1094342015581024–. https://doi.org/10.1177/1094342015581024 arXiv:arXiv:1408.4675v1

[4] Jiri Jaros, Filip Vaverka, and Bradley E Treeby. 2016. Spectral Domain Decomposition Using Local Fourier Basis: Application to Ultrasound Simulation on a Cluster of GPUs. *Supercomputing Frontiers and Innovations* 3, 3 (nov 2016), 39–54. https://doi.org/10.14529/jsfi160305

[5] Jiri (Nvidia Corporation) Kraus. 2013. An Introduction to CUDA-Aware MPI. (2013).

[6] Cook Shane. 2013. *CUDA Programming*. Elsevier. https://doi.org/10.1016/C2011-0-00029-7 arXiv:arXiv:1011.1669v3

[7] Bradley E. Treeby, Jiri Jaros, and Ben T. Cox. 2016. Advanced photoacoustic image reconstruction using the k-Wave toolbox. In *SPIE Photons Plus Ultrasound: Imaging and Sensing*, Alexander A. Oraevsky and Lihong V. Wang (Eds.), Vol. 9708. 97082P. https://doi.org/10.1117/12.2209254

[8] Sheng Wang, Jing Lin, Tianfu Wang, Xiaoyuan Chen, and Peng Huang. 2016. Recent advances in photoacoustic imaging for deep-tissue biomedical applications. *Theranostics* 6, 13 (2016), 2394–2413. https://doi.org/10.7150/thno.16715

[9] Xueqin Zhang, Kai Shen, Chengguang Xu, and Kaifang Wang. 2013. Design and Implementation of Parallel FFT on CUDA. In *2013 IEEE 11th International Conference on Dependable, Autonomic and Secure Computing*. IEEE, 583–589. https://doi.org/10.1109/DASC.2013.130

[10] Yu-Feng Zhou, Ali Syed Arbab, and Ronald Xiaorong Xu. 2011. High intensity focused ultrasound in clinical tumor ablation. *World journal of clinical oncology* 2, 1 (2011), 8–27. https://doi.org/10.5306/wjco.v2.i1.8