SAND2011-0421C

The Development of Mellanox - NVIDIA GPUDirect over InfiniBand – a New Model for GPU to GPU Communications

Gilad Shainer[1], Ali Ayoub[2], Pak Lui[2], Tong Liu[2], Christian R. Trott[3], Greg Scantlen[4], Paul S. Crozier[5]

{[1]HPC Advisory Council, [2]Mellanox Technologies, [3]Institut für Physik, Technische Universität at Ilmenau Germany, [4]Creative Consultants, [5]Sandia National Laboratories}

Abstract

The usage and adoption of General Purpose GPUs (GPGPU) in HPC systems is increasing due to the unparallel performance advantage of the GPUs and the ability to fulfill the ever-increasing demands for floating points operations. While the GPU can offload many of the application parallel computations, the system architecture of a GPU-CPU-InfiniBand server does require the CPU to initiate and manage memory transfers between remote GPUs via the high speed InfiniBand network. In this paper we introduce for the first time a new innovative technology - GPUDirect that enables Tesla GPUs to transfer data via InfiniBand without the involvement of the CPU or buffer copies, hence dramatically reducing the GPU communication time and increasing overall system performance and efficiency. We also explore for the first time the performance benefits of GPUDirect using Amber and LAMMPS applications.

1. Introduction

The rapid increase in the performance of graphics hardware, coupled with recent improvements in its programmability, has made graphics accelerators a compelling platform for computationally demanding tasks in a wide variety of application domains. Due to the great computational power of the GPU, the GPGPU method has proven valuable in various areas of science and technology. The modern GPU is a highly data-parallel processor, optimized to provide very high floating point arithmetic throughput for problems suitable to solve with a single program multiple data model. On a GPU, this model works by launching thousands of threads running the same program working on different data. The ability of the GPU to rapidly switch between threads in combination with the high number of threads ensures the hardware is busy at all times. This ability effectively hides memory latency, and in combination with the several layers of very high bandwidth memory available in modern GPUs also improves GPU performance. Figure 1 shows the performance and memory bandwidth advantage of GPUs versus X86 CPUs. This advantage enables HPC systems to achieve the needed performance capabilities mandated by the ever increasing simulation complexities [1], [2], [3].

There are two main classes of programming interfaces for GPUs - graphics interfaces (such as OpenGL or DirectX), and general-purpose interfaces, such as NVIDIA's Compute Unified Device Interface (CUDA). These interfaces provide a natural programming environment, in particular allowing integer variables, pointer manipulation, and arbitrary memory reads and writes on the GPU cores. CUDA is emerging as

the dominant interface for scientific GPGPU programming [4]. CUDA programs are based on the C programming language with certain extensions to utilize the parallelism of the GPU. These extensions also provide very fast implementations of standard mathematical functions such as trigonometric functions, floating point divisions, logarithms, etc. The code defining the kernel looks almost like an ordinary scalar C function. The only major difference is that some effort must be made to optimize the function to work as efficiently as possible in a parallel environment, e.g., to minimize divergence among the threads and synchronize thread memory read/writes.

GPU-based clusters, as shown in Figure 2, are being used to perform compute intensive tasks, like finite element computations, gas dispersion simulations, heat shimmering simulations, accurate nuclear explosion simulations, Monte-Carlo simulations, etc. Five out of the seven world's Petascale systems, according to the Nov 2010 release of the world's TOP500 supercomputers [5], are using GPUs or accelerators in order to achieve the desired performance while reducing the total system power consumption and real-estate. Since the GPUs provide the highest core count and the floating point operations capability, a high-speed network is required to connect between the GPU-CPU platforms or servers. In many cases, InfiniBand has been chosen as the high-speed networking of choice for such systems.

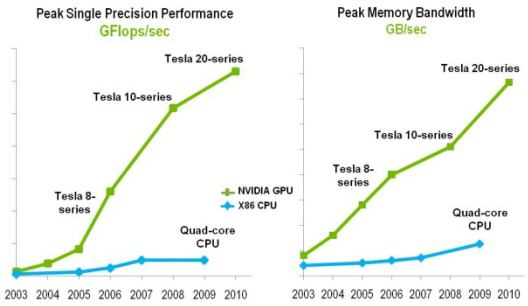

Figure 1 - The performance comparison of NVIDIA GPUs versus X86 multi-core CPUs

By providing low-latency, high-bandwidth and extremely low CPU overhead, InfiniBand [6] has become the most deployed high-speed interconnect for high-performance computing, replacing proprietary or low-performance solutions. The InfiniBand Architecture (IBA) is an industry-standard fabric designed to provide high bandwidth, low-latency computing, scalability for ten-thousand nodes and multiple CPU/GPU cores per server platform and efficient utilization of compute processing resources. Mellanox ConnectX-2 InfiniBand adapters and IS5000 switches [7] provide up to 40Gb/s of bandwidth between servers and up to 120Gb/s between switches. This high-performance bandwidth is matched with ultra-low application latency of 1µsec, and switch latencies under 100ns that enable efficient scale out of compute systems.

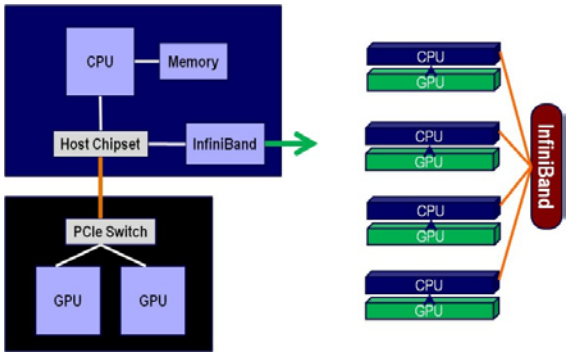

Figure 2 – A diagram of a GPU-CPU-InfiniBand cluster

2. The GPU-GPU communication model

While GPUs have been shown to provide worthwhile performance acceleration yielding benefits to price/performance and power/performance, several areas of GPU-based clusters could be improved in order to provide higher performance and efficiency. One issue with deploying clusters consisting of multi-GPU nodes involves the interaction between the GPU and the high speed InfiniBand network - in particular, the way GPUs use the network to transfer data between them. Before the GPUDirect technology, a performance issue existed with user-mode DMA mechanisms used by GPU devices and the InfiniBand RDMA technology. The issue involved the lack of a software/hardware mechanism of "pinning" pages of virtual memory to physical pages that can be shared by both the GPU devices and the networking devices. In general, GPUs use pinned memory in the host memory to increase DMA performance by eliminating the need for intermediate buffers, or to pin and unpin regions of memory on-the-fly. The use of pinned memory buffers can allow a well-written code to achieve zero-copy message passing semantics via RDMA. The lack of a mechanism for managing memory pinning among user-mode accelerator and InfiniBand message passing libraries creates performance issues due to the need of having a third device, the host CPU, be responsible for moving the data between the different GPU and InfiniBand pinned memory regions. The issue is described in Figure 3.

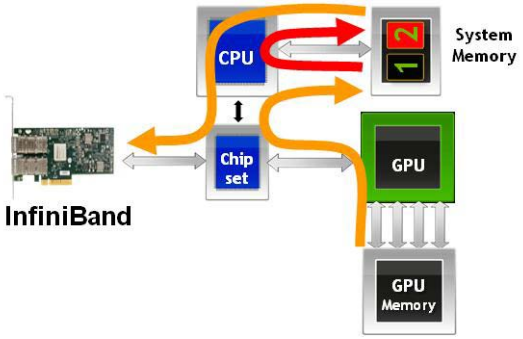

Figure 3 – The non-efficient GPU-InfiniBand data transfer mechanism

As shown in Figure 3, data transfer between remote GPUs requires three steps:

1. The GPU writes data to a host pinned memory, marked as system memory 1
2. The host CPU copies the data from system memory 1 to system memory 2
3. The InfiniBand device reads data from its pinned memory - system memory 2, and sends it to the InfiniBand pinned memory on the remote node

Step 2 not only requires the host CPU involvement, which in turn reduces CPU efficiency, introduces CPU overhead and CPU noise (CPU interrupt), but it also increases the latency for the GPU data communications. Such overhead can be counted for 30% of the communication time, which can dramatically

reduce high-performance, latency-sensitive application performance.

3. GPUDirect, the new model for GPU communications

The ultimate communication mechanism between GPUs and InfiniBand devices would involve the development of a mechanism for performing DMA and RDMA operations directly between GPUs and bypass the host entirely. Such an interface could conceivably allow RDMAs from one GPU device directly to another GPU on a remote host. An intermediate solution can use the host memory for the data transactions, but requires elimination of the host CPU's involvement by having the acceleration devices and the InfiniBand adapters share the same pinned memory as described in Figure 4.

The new hardware/software mechanism called GPUDirect eliminates the need for the CPU to be involved in the data movement, and essentially enables not only higher GPU-based cluster efficiency, but sets the way for the creation of "floating point services". GPUDirect is based on a new interface between the GPU and the InfiniBand device that enables both devices to share pinned memory buffers, and for the GPU to notify the network device to stop using the pinned memory so it can be destroyed. This new communication interface allows the GPU to maintain control of the user-space pinned memory, and eliminates the issues of data reliability.

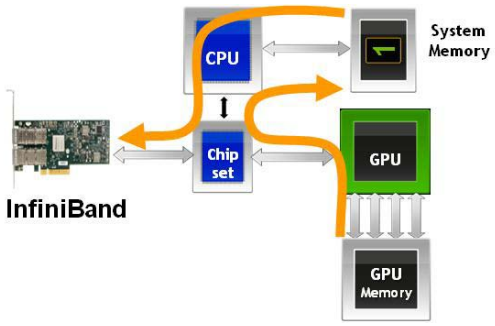

Figure 4 – An efficient GPU InfiniBand data transfer mechanism – GPUDirect

4. The GPUDirect Elements

The development of the GPUDirect solution required software modification in three areas – the Linux kernel, the Mellanox InfiniBand drivers and the Tesla GPU drivers. No hardware changes have been required as all of the hardware capabilities were already included in the Tesla GPUs and within the Mellanox ConnectX-2 adapters (such as native RDMA etc.).

Linux Kernel modifications: support was added for sharing pinned pages between different drivers. With the GPUDirect capability the Linux Kernel Memory Manager (MM) allows NVIDIA and Mellanox drivers to share the host memory and provides direct access for the latter to the buffers allocated by the NVIDIA CUDA library, and thus, providing Zero Copy of data and better performance.

NVIDIA Driver: allocated buffers by the CUDA library are managed by the NVIDIA Tesla driver. We have added the modifications to mark these pages to be shared so the Kernel MM will allow the Mellanox InfiniBand drivers to access them and use them for transportation without the need for copying or re-pinning them.

Mellanox OFED Drivers: The InfiniBand application running over Mellanox InfiniBand adapters uses the OpenFabrics Enterprise Distribution (OFED) API to send the data that resides on the buffers allocated by the NVIDIA Tesla driver. We have modified the Mellanox InfiniBand driver to query this memory and to be able to share it with the NVIDIA Tesla drivers using the new Linux Kernel MM API. In addition, the Mellanox driver registers special callbacks to allow other drivers sharing the memory to notify any changes performed during run time in the shared buffers state, in order for the Mellanox driver to use the memory accordingly and to avoid invalid access to any shared pinned buffers.

5. Performance evaluation

In order to evaluate the performance advantage of GPUDirect, we have decided to use Amber and LAMMPS [8].

5.1 Amber Performance Evaluation

Amber is a molecular dynamics software package, one of the most widely used programs for bimolecular studies with an extensive user base and is developed in an active collaboration of David Case at Rutgers University, Tom Cheatham at the University of Utah, Tom Darden at NIEHS (now at OpenEye), Ken Merz at Florida, Carlos Simmerling at SUNY-Stony Brook, Ray Luo at UC Irvine, and Junmei Wang at Encysive Pharmaceuticals. Amber was originally developed under the leadership of Peter Kollman.

One of the new features of Amber 11 is the ability to use NVIDIA GPUs to accelerate both explicit solvent PME and implicit solvent GB simulations. We have used the HPC Advisory Council Computing Center for the performance testing, which included 8 compute nodes, connected via Mellanox ConnectX-2 InfiniBand adapters and switches. Each node includes one NVIDIA Fermi c2050 GPU.

GPUDirect accelerates GPU communications over InfiniBand by 30% which in turn can provide up to 42% theoretical peak performance improvement, assuming the application does nothing but communications, as shows in the function below.

$$Performance\ Increase = \frac{T}{(T - T_{boost})} \text{-1=0.428}$$

In practice, we expect the performance improvement to be dependent on the system size and be in the range of 20-35%. The performance improvement can vary based on the amount of GPU communications being used, or the extent of GPU parallel computations. The Amber performance results with and without GPUDirect are presented in Figures 5 and 6. Figure 5 presents the performance results with the GPU ECC mode enabled (for data protection). Figure 6 presents the performance results with the GPU ECC mode disabled. The reason for testing with ECC mode disabled was to achieve the highest performance possible. Furthermore we wanted to explore if GPUDirect would provide performance benefits for both modes. We have tested the Cellulose (408,609 atoms) case on four nodes and eight nodes, using the PMEMD simulation program. The cellulose benchmark simulates a cellulose fiber and it consists of 408,609 atoms including water.

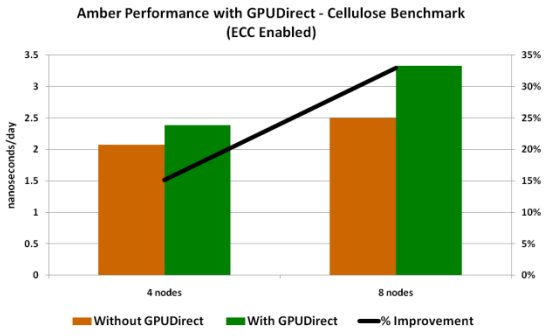

Figure 5 – Cellulose benchmark results with ECC enabled

When ECC was enabled, the performance benefit with GPUDirect as shown in graph 5 is 15% with 4 nodes and 33% with 8 nodes. When ECC was disabled, the performance benefit with GPUDirect as shown in graph 6 is 16% with 4 nodes and 29% with 8 nodes for the Cellulose case. When more nodes are added to the system, the performance improvement with GPUDirect increases due to the fact that more communications are performed between the nodes as part of the parallel computations. As such, we expect the performance benefit to continue and increase beyond our 8 node test bed numbers.

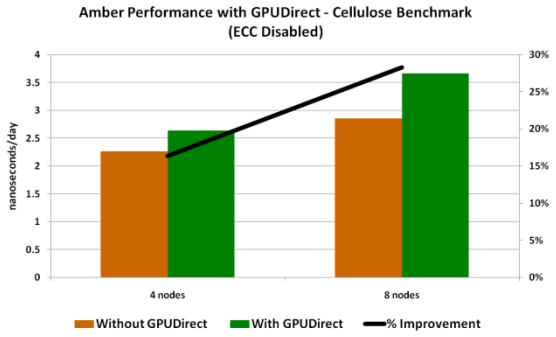

Figure 6 – Cellulose benchmark results with ECC disabled

5.2 LAMMPS Performance Evaluation

LAMMPS is a Large-scale Atomic/Molecular Massively Parallel Simulator. It is a classical molecular dynamics code which can model atomic, polymeric, biological, metallic, granular, and coarse-grained systems.

In order to leverage the compute power of GPUs the USER-CUDA package has been developed for LAMMPS at the University of Technology Ilmenau. It currently supports 26 different force fields and is able to use GPUDirect for further speed increases. By allowing the pinning of a communication buffer by both the GPU and the network card, GPUDirect enables a partial overlap of force calculations with communication routines by exploiting asynchronous data transfers. The latest development version of the package is available at the GPU-LAMMPS repository [8] while stable releases can also be downloaded at www.tu-ilmenau.de/lammpscuda.

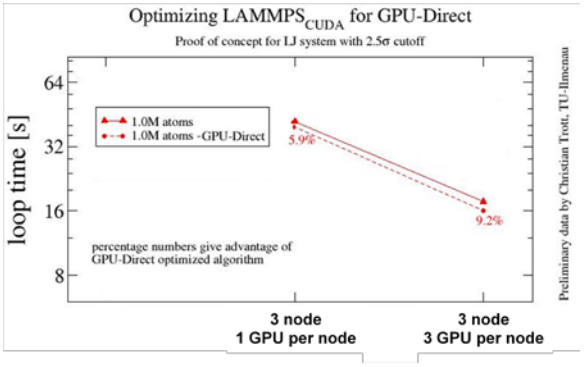

Figure 7 – LAMMPS benchmark results

For testing LAMMPS we have used the eStella benchmarking center of Creative Consultants, using the same GPUs and InfiniBand cards that were used in the Amber testing. LAMMPS

performance results are shown in Figure 7. GPUDirect accelerates LAMMPS by nearly 10% at 3 nodes/9 GPUs. The performance increases with cluster size and we expect it to reach 35% at larger system size. We plan to benchmark the code on larger system in the near future and to report the performance results.

5.3 The usage of RDMA

Achieving higher performance results relies on the ability to utilize RDMA options within the MPI library in order to take full advantage of the GPUDirect technology. GPUDirect enables the InfiniBand adapter to directly access the host memory used by the GPU, hence using RDMA operations will results in no buffer copies and the highest performance gain. We have modified the MPI library used for the testing (MVAPICH) to use RDMA for a message size used by the applications. To make it simpler, we have set the MPI to use RDMA for every message size. One of course can profile a given application to determine the most used message sizes and to set the MPI parameters accordingly.

6. Summary and future plans

HPC demonstrates ever-increasing demands for more computing power. We see those ever increasing demands on the TOP500 supercomputers list and from the worldwide Petascale programs. GPU-based compute clusters are becoming the most cost-effective way to provide the next level of compute resources. For example, building a Petascale Cray proprietary system requires 20,000 nodes, while achieving the same level of performance using InfiniBand and GPU-based clusters requires only 5,000 nodes. The advantages of the second option are clear - space, management and affordability.

As GPU-based computing becomes popular, there is a need to create direct communications between GPUs using the world's fastest interconnect solutions such as InfiniBand, and increase the modifications necessary for applications to utilize GPU and parallel GPU computations more effectively. In this paper we have reviewed the latest GPUDirect technology and demonstrated up to 33% performance improvement (with translates to the capability to run 33% more jobs per day) on only 8 nodes, each with a single NVIDIA Fermi GPU and Mellanox ConnectX-2 InfiniBand adapter. Furthermore, the performance results presented in the paper are considered to be a world record compared to published data when this paper was written. These results met our initial expectations, and we expect to see higher benefits on larger systems.

We plan to continue testing different applications and difference usage models with GPUDirect in the near future, as part of the HPC Advisory Council's education activities (www.hpcadvisorycouncil.com). We also plan to explore further optimizations in the GPU-IB communications to enable faster and even more efficient data transfers between multiple GPUs.

Acknowledgement

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract No. DE-AC04-94AL85000.

References

[1] V. Kindratenko, J. Enos, Guochun Shi, M. Showerman, G. Arnold, J. Stone, J. Phillips, Wen-mei Hwu; "GPU clusters for high-performance computing"; Cluster Computing and Workshops, 2009.

[2] Enhua Wu, Youquan Liu; "Emerging technology about GPGPU"; Circuits and Systems, 2008.

[3] Gang Chen, Guobo Li, Songwen Pei, Baifeng Wu; "High Performance Computing via a GPU"; Information Science and Engineering (ICISE), 2009.

[4] M. Garland ; "Parallel computing with CUDA"; Parallel & Distributed Processing (IPDPS), 2010

[5] The TOP500 list – www.top500.org

[6] InfiniBand Trade Association - www.infinibandta.org/

[7] Mellanox Technologies – www.mellanox.com

[8] LAMMPS and the USER-CUDA package - http://lammps.sandia.gov/, http://code.google.com/p/gpulammps/