2012 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 7-12, 2012. Vilamoura, Algarve, Portugal

A master-slave robotic simulator based on GPUDirect

Jianying. Li, Yu. Guo, Heye. Zhang, Yongming. Xie

Abstract—**The same as in traditional surgery, surgeons in telerobotic surgery need extensive training to achieve experience and highly accurate instrument manipulation. Traditional training methods like practice in operating room have major drawbacks such as high risk and limited opportunity for which virtual reality (VR) and computer technologies can offer solutions. To accelerate the data transmission speed in our master-slave robotic simulator, GPUDirect was applied to ensure the synchronization and display rate of three computers. By using GPUDirect with InfiniBand card we realized up to 247% performance improvement in data transmission speed on NVIDIA Tesla™ products on different computers compared to that without GPUDirect, which shows that GPUDirect enables better communication between remote GPUs over InfiniBand.**

I. INTRODUCTION

IN recent years, medical robots have been developed to assist surgeons in performing a variety of surgeries [1][2][3][4]and some of them have been commercialized successfully, such as the Da Vinci Surgical Robotic System (Intuitive Surgical, Inc, Mountain View, California, USA) and Sensei™ (Hansen Medical, Inc, Mountain View, CA, USA) [5]. Such systems allow surgeons to operate on patents remotely. This is usually done through a master-slave robot, with imaging supplied through video cameras configured to provide a stereoscopic view. The same as in traditional surgery, surgeons in robotic surgeries need extensive training to achieve experience and highly accurate manipulations of instruments. Traditional training methods like practice in operating room have major drawbacks such as high risk and less chance for which virtual reality (VR) and computer technologies can offer solutions [6][7][8]. Early results suggest that virtual reality simulators have played an important role in surgical training [9][10].

In our previous work, a master-slave robotic simulator was developed for telerobotic spine surgery training [11][12].Our simulator's goal is high-fidelity presentation of the visual and haptic cues present in a surgical environment. The simulator has a similar structure with the Da Vinci Surgical Robotic System, a maser console for the operator, a slave robot following the motion of the operator and a stereo TV for assistants. Stereoscopic view and haptic interface in the master console can provide the operator visual and tactile feedback respectively in real-time. The slave robot can interact with a touch screen virtually, following the motions of the operator at the master side via a master-slave control scheme [12], which simulated telerobotic operations successfully. For the visual feedback in the simulator, virtual surgery scene needs to be displayed on the computer at the master side, the computer connected with the touch screen at the slave side and the computer with a stereo TV for the assistants simultaneously, which means it must be transferred and synchronized between several computers. When the interaction between the virtual surgical tools and the virtual object on the computer at the master side occurs, update of the display at the slave side and on the stereo TV immediately is necessary for living simulation of teleoperation. In our original prototype, we used Gigabit Ethernet for the communication among the three computers, and found the transmission speed was not satisfactory and latency of the update of the display on the slave computer and the stereo TV.

Based on traditional data transfer methods such as TCP/IP protocol, due to the limit of bandwidth, the update rate in the slave computer reduces greatly when large amounts of data are transferred. Computation-intensive tasks such as dense matrix multiplication can benefit from GPU's higher floating-point performance [13][14][15]. Another category of memory-bandwidth-intensive tasks suitable for GPU clusters are those with high degree of data locality, such as finite-difference simulation [16]. The source data are decomposed spatially and distributed over the device memory (or dmem in short) of the GPUs. GPU devices only communicate with each other about the state updates at the boundaries of the spatial decomposition. With limited communications in and out of each GPU device, the overall performance benefits considerably from the high dmem bandwidth. If the data must be transferred from/to dmem every time, then the PCI bandwidth between hmem and dmem becomes the bottleneck [17]. Luckily, the problem can be solved [18][19][20][21] by using GPUDirect, the newly released technology by NVIDIA.

In this paper, GPUDirect is applied and tested in our master-slave robotic simulator to improve the synchronization and display rate. Transmission speeds with and without GPUDirect in the simulator are compared. The paper is organized as follows: in Section 2 the background of CUDA and GPUDirect is introduced and in Section 3 the telerobotic simulator is described briefly. Then we detail the virtual surgical data transfer process and present the results in Section 4. In Section 5 we conclude the paper.

Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences Shenzhen, PR China, 518055
The Chinese University of Hong Kong, PR China
Guangdong Province Key Laboratory of Robotics and Intelligent System, PR China
{ Jy.li & yu.guo & hy.zhang & ym.xie} @siat.ac.cn

I. CUDA AND GPUDIRECT

A. CUDA

In recent years, the performance of GPU has been greatly improved, and its great potential and power as a general-purpose processing unit has been found. CUDA (Compute Unified Device Architecture) introduced in November 2006 by NVIDIA can effectively use the great processing power and huge memory bandwidth of GPU in general-purpose processing, which is widely used in video image processing, pattern recognition, computational fluid dynamics, bio-computing and other fields [22]. CUDA is a new parallel programming model and instruction set for NVIDIA GPUs that can be used for performing general purpose computations [23][24]. It comes with a software environment that allows developers to use C as a high level programming language, which makes it accessible to the non-expert.

The architecture offers support for massively multithreaded applications and provides mechanisms for inter-thread communication and memory access. The API distinguishes between host and device domains and offers access to fast caches on the device side. The implemented method of thread partitioning allows for the execution of multiple CUDA applications (kernels) on one GPU [25]. Each kernel is executed by a grid of thread blocks. A block consists of a batch of threads that can cooperate together by efficiently sharing data through some fast shared memory and synchronizing their execution to coordinate memory accesses. The threads in the block are addressed through one-, two- or three-dimensional IDs. This allows uniquely identifying and assigning tasks to each thread. Another feature of the CUDA architecture is the interoperability with graphic APIs (OpenGL and Direct3D) which allows to use, for example, rendered images as input to CUDA kernels. Since this data already resides on the graphics device it only needs to be copied on the device to be processed by CUDA [26][27][28].

B. GPUDirect v1.0

First released in June 2010 by NVIDIA, GPUDirect v1.0 allows third party network adapters, solid-state drives (SSDs) and other devices directly read and write CUDA host memory, eliminating unnecessary system memory copies and CPU overhead, resulting in significant performance improvements in data transfer times on NVIDIA Tesla™ and Quadro™ products. GPUDirect v1.0 is based on a new interface between the GPU and the InfiniBand device that enables both devices to share the same pinned memory and for the GPU to notify the network device to stop using the pinned memory so it could 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. The InfiniBand device uses pinned memory for RDMA transfers and CUDA also uses pinned memory for fast DMA transfers to/from card, avoiding one extra memory copy between GPU and Infiniband pinned memory.

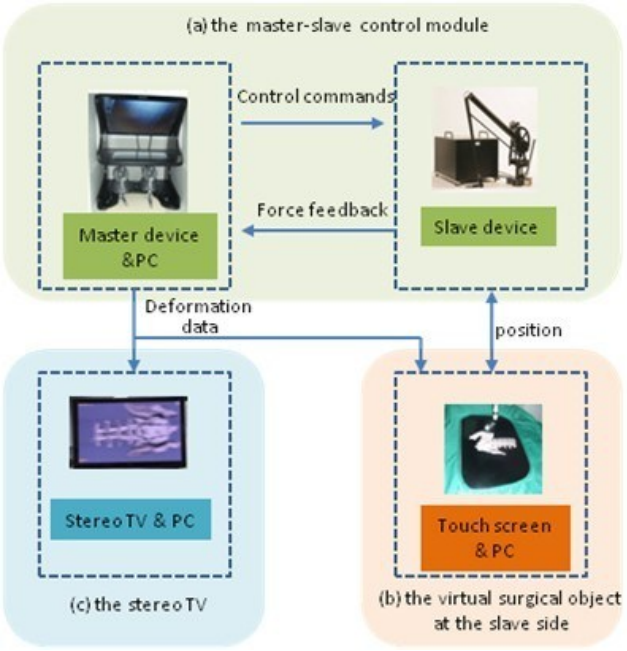

Fig. 1. System architecture of the virtual reality simulator for telerobotic surgery.

GPUDirect version 1 is for accelerating GPU communications between GPUs located on separate computers over InfiniBand, supported by Mellanox and Qlogic and no changes to user code are necessary.

GPUDirect v1.0 supports deployment on Tesla M-class and Tesla S-class datacenter products running Red Hat Enterprise Linux (RHEL). The development of the GPUDirect v1.0 solution required software modification in three areas: the Linux kernel, the Mellanox InfiniBand drivers and the Tesla GPU drivers, for more information please reference [25] and follow the installation instructions in the README file.

II. THE MASTER-SLAVE ROBOTIC SIMULATOR

The master-slave robotic simulator consists of three parts, as shown in Fig.1: (a) the master-slave control module which simulates the telerobotics; (b) the touch screen set in the back of a human body model which provides the virtual surgical object at the slave side to provide an immersive telerobotic environment; (c) the stereo TV to display the surgery scene the same as that at the master side for the assistants. The prototype of the simulator is shown in Fig.2.

In the master-slave control module, Display 300 (SenseGraphics, Inc, Stockholm, Sweden) incorporating two PHANTOM Omni haptic devices (Sensable, Inc, USA) as the master side and a Phantom Premium 3.0 haptic device (Sensable, Inc, USA) as the slave side, with a proper master-slave control algorithm [12], the operator can handle the Phantom Omni devices to control the movements of the slave device and treat the Phantom Omni devices as surgical tools to interact with the virtual spine in the virtual surgical environment on the computer at the master side.

In order to enhance the immersion of telerobotics, a touch

Fig. 2. The prototype of the master-slave robotic simulator.

screen (WACOM UX21) set in the back of a human body model facing down on a surgical bed is used for the slave haptic device to interact with. To show the operator's operations at master side to more trainees, a stereo TV controlled by a computer is offered too. The virtual surgery scene such as the interaction between the virtual surgical tools and the spine can be seen on the computer at the master side, on the touch screen at the slave side and the stereo TV simultaneously.

The teleoperator system is primarily a computer-based system with a master computer controlling the two Phantom Omni haptic devices, the Display 300 and the Phantom Premium 3.0, another computer controlling the touch screen, a third computer connected to the stereo TV, and the communication among the three computers based on GPUDirect is focused on in the following section.

III. Virtual Surgery Data Transfer Based on GPUDirect

A. Data Preparation

In our master-slave robotic simulator, the virtual surgery scene need to be displayed on the Display 300 at the master side, on the touch screen at the slave side and the stereo TV simultaneously. For the virtual surgery, the interactive calculation process is a context-sensitive vector calculation process, so when the location and status of the tool are transformed, it must ensure that every location is sent successfully. Otherwise it will lead to inconsistency if the computer that needs synchronization updates the spine model according to an error, and the inconsistency will increase over time. Therefore, our proposed transmission method does not directly transfer the location of the tool and the whole volume of the spine at the master side, but the deformation data, i.e. the vertexes whose positions change as time and the rotation matrix of the surgical tool. Thus the data transferred is context-free, which facilitates the real-time transmission on the network.

B. Hardware Deployment for Data Transmission

GPUs have been shown to provide worthwhile performance acceleration yielding benefits to price/performance and power/performance. In our master-slave robotic simulator, GPUs are applied to accelerate not only the calculation of the deformation of interaction between virtual surgical tools and objects but also the display of the updated virtual surgery scene. The computer at the master side is HP Workstation xw4600 mounted with Tesla S1070, the computer for the touch screen is Dell Precision T5500 installed with Tesla S1070, and the computer connected to the stereo TV is Dell Precision T5500 with Tesla S1070. All the computers are connected with high speed InfiniBand network [29]. The above deployment satisfies the requirements of GPUDirect v1.0. The structure of the hardware deployment and data transmission process is illustrated in Fig. 3. Software modification is completed according to [29].

During the training, when the interaction at the master side occurs, the shape of the virtual surgical object may change, then the rotation matrix of the surgical tool and the deformation data of the object computed on the computer (HP workstation) at the master need to be transmitted to the graphic cards (Tesla S1070), the computer on the slave side and the computer for the stereo TV via InfiniBand network cards. At first, the volume data of surgical object and tools are transferred from the CPU memory to the memory of the Tesla S1070 on the HP workstation to do the calculation of the deformation.

Secondly, the deformation data was transferred back to the pinned host memory shared by the InfiniBand network cards, then send to the host memory of the computer at the slave side and the computer for stereo TV respectively by InfiniBand. At last the Tesla S1070 on the computer at the slave side and that on the computer for stereo TV read the data from the pinned host memory shared by the IB-cards on the corresponding computer and update their display. In this way, Infiniband network cards and Tesla S1070 share the same pinned memory and Infiniband uses pinned memory for RDMA transfers; CUDA also uses pinned memory for fast DMA transfers to/from card, avoiding one extra memory copy between GPU and Infiniband pinned memory.

To test the efficiency of GPUDirect v1.0 in our simulator, transmission speeds with GPUDirect and without it are recorded. Transmission speed with Gigabit Ethernet is also computed.

C. Results and Analysis

The tested results are listed in Table 1. Compared to the transmission speed of 115MB/s by Gigabit Ethernet, we got a transmission speed of 750MB/s without GPUDirect and 1.85GB/s with GPUDirect by InfiniBand. The transmission speed by InfiniBand with GPUDirect is 5.5 times more than that by Gigabit Ethernet because the InfiniBand is designed with PCI-Express slots instead of the traditional network card slot and GPUDirect technology has raised the speed to 1.85GB/s, leading to 247% performance improvement compared to 750MB/s by InfiniBand without GPUDirect. One existed issue with a system 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

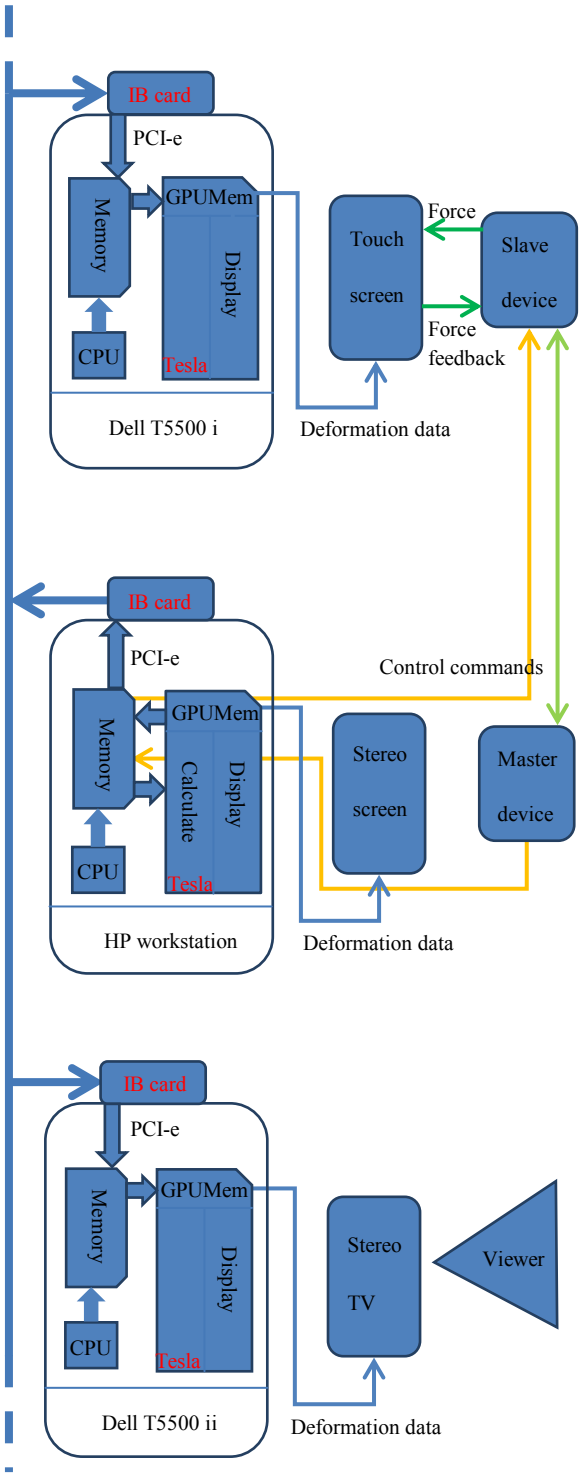

Fig. 3. The structure of the hardware deployment and data transmission process with GPUDirect.

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. With GPUDirect technology, Infiniband network card and GPU can share the same pinned memory, avoiding one extra memory copy between GPU and Infiniband pinned memory and eliminating unnecessary system memory copies and CPU overhead. At the same time, two GPU-related factors lead to better performance: firstly the use of GPU devices improves the sustained memory bandwidth for processing large-size data; secondly GPU device memory allows larger subtasks to be processed in whole and hence reduces repeated data transfers between memory and processors.

TABLE I
COMPARE OF TRANSMISSION SPEED IN OUR MASTER-SLAVE ROBOTIC SIMULATOR

Network	With GPU-Direct	Without GPUDirect
InfiniBand	1.85GB/s	750MB/s
Gigabit Ethernet	x	115MB/s

Different memory has different bandwidth for data transfer. For example, the GPU device memory (or dmem) can be up to 20 times faster than the main host memory (or hmem). To achieve optimal performance, source and temporary data must be held in the right hardware memory. Bandwidth-intensive tasks are harder to accelerate, as the bottleneck often lies with the PCI between main memory and GPU device memory or the communication network between workstation nodes. Data in different memory devices with different bandwidth must be properly allocated. To reach the peak bandwidth between different memory devices, the interface hardware and low-level software often require data to be transferred in certain restricted patterns, which affects the overall performance. With limited communications in and out of each GPU device, the overall performance benefits considerably from the high device memory (dmem) bandwidth. Thanks to the optimization of CUDA memory operation in GPUDirect, data transmission speed on the slave computer and the stereo TV in our master-slave robotic simulator is increased.

IV. CONCLUSION

Virtual reality robotic simulators have played an important role in robotic surgical training. To provide a highly immersive experience in our master-slave robotic simulator, updates of the display on the slave server and the server for the stereo TV in real time are necessary. GPUDirect v1.0 was applied in our master-slave robotic simulator to accelerate virtual surgery data transmission speed between three computers mounted with Tesla S1070. By using GPUDirect v1.0 with InfiniBand card we realized up to 247% (1.85 /0.75) performance improvement in data transfer times on NVIDIA Tesla™ products on different computers, which shows that GPUDirect version 1 enables better communication between remote GPUs over InfiniBand.

REFERENCES

[1] H. Tanaka, K. Ohnishi, H. Nishi, T. Kawai, "Bilateral Control in Multi DOF Haptic Surgical Robotic System Utilizing FPGA", GASTROINTESTINAL ENDOSCOPY, vol. 72, pp. 593-599, 2010.

[2] K. Ho, S. Phee, A. Shabbir, S. Low , V . Huynh , A . Kencana , K. Yang , D . Lomanto , B . So , Y . Wong , S .Chung, "Endoscopic submucosal dissection of gastric lesions by using a Master and Slave Transluminal Endoscopic Robot (MASTER)",Gastrointest Endosc,vol. 72, pp.593-599, 2010.

[3] G. Mylonas, K. Kwok, "Haptic Constraints and associated Cognitive Demand for Robotic MIS", Medical Image Analysis (2010), doi: 10.1016/j.media.2010.07.007.

[4] S. Low, L. Phee, "A Review of Master-Slave Robotic Systems for Surgery", Proceedings of the 2004 IEEE Conference on Robotics, Automation and Mechatronics, Singapore, pp. 37-42, October, 2004.

[5] P. Gomes, "Surgical robotics: Reviewing the past, analysing the present, imagining the future." Robotics and Computer-Integrated Manufacturing,, 2011, 27(2): 261-266.

[6] H.C. Peter,C.C. Patrick, J.M. Christopher, A. John, "Virtual reality simulators: Current status in acquisition and assessment of surgical skills", Cartmill Division of Surgery, Nepean Hospital, Sydney, New South Wales, Australia, vol 72, pp. 30–34 ,14 Mar 2002.

[7] E.S. Neal, G.G. Anthony, A.R. Sanziana, K.O. Michael, K.B Vipin, K.A. Dana, M.S. Richard. "Virtual Reality Training Improves Operating Room Performance." ANNALS OF SURGERY, Vol. 236, No. 4, 458–464, 2002.

[8] H. Berkenstadt, D. Erez, Y. Munz, D. Simon, A. Ziv, "Training and Assessment of Trauma Management: The Role of Simulation-Based Medical Education." Anesthesiology Clinics, vol. 25, pp. 65–74, March 2007.

[9] S. Tsuda, D. Scott, J. Doyle,D.B. Jones, "Surgical Skills Training and Simulation." Curr Probl Surg 2009; 46:271-370.

[10] S. Suebnukarn,N. Phatthanasathiankul,S. Sombatweroje, P. Rhienmora, P. Haddawy, "Process and outcome measures of expert/novice performance on a haptic virtual reality system." Journal of dentistry, 2009,37, pp :658-665.

[11] Y. M. Xie, P. A. Heng, J. S. Zhang, P. Liu, T. C. Zhu, J. Y. Li, "A simulation system for training telerobotic spine surgery," The 2011 International Conference on Electric Information and Control Engineering, 2011, pp. 102 - 105.

[12] P. Liu, Y. M. Xie, T. C. Zhu, J. Y. Li, J. S. Zhang, P. A. Heng, "A master-slave telesurgery simulator with force-feedback," Lecture Notes in Electrical Engineering, 1, Volume 126, Recent Advances in Computer Science and Information Engineering, Pages 493-499.

[13] X. Cui, Y. Chen, and H. Mei. "Improving performance of matrix multiplication and FFT on GPU." In International Conference on Parallel and Distributed Systems, pages 42–48. IEEE Computer Society, 2009.

[14] M. Fatica, "Accelerating linpack with CUDA on heterogenous clusters." GPGPU'09, June 2009.

[15] V. Volkov, J. W. Demmel, "Benchmarking GPUs to tune dense linear algebra." SC'08, November 2008.

[16] P. Micikevicius, "3D finite difference computation on GPUs using CUDA." GPGPU2, March 2009.

[17] PCISIG, "*PCI Express 3.0 Frequently Asked Questions*."http://www.pcisig.com/news_room/faqs/pcie3.0_faq.

[18] H. Wang, S. Potluri, M. Luo, et al., "Optimized Non-contiguous MPI Datatype Communication for GPU Clusters: Design, Implementation and Evaluation with MVAPICH2," in IEEE International Conference On Cluster Computing (Cluster' 11), 2011.

[19] *Mellanox: NVIDIA GPUDirect technology—accelerating GPUbased systems.* http://www.mellanox.com/pdf/whitepapers/TB_GPU_Direct.pdf

[20] H. Wang, S. Potluri, M. Luo, A. K. Singh, S. Sur, and D. K. Panda, "MVAPICH2-GPU: Optimized GPU to GPU Communication for InfiniBand Clusters," in Computer Science - R&D 26(3-4), pp. 257 – 266, 2011.

[21] InfiniBand Trade Association. http://www.infinibandta.com

[22] J.A. Stratton, S.S. Stone, et al., "M-CUDA: An efficient implementation of CUDA kernels on multicores." IMPACT Technical Report 08-01, University of Illinois at Urbana-Champaign, 2008.

[23] S.S. Stone, H. Yi, W.W. Hwu, et al., "How GPUs can improve the quality of magnetic resonance imaging. The First Workshop on General-Purpose Processing" Graphics Processing Units (October).

[24] J. Nickolls et al., "Scalable Parallel Programming with CUDA," ACM Queue, vol. 6, no. 2, Mar./Apr. 2008, pp. 40-53.

[25] E. Lindholm et al., "NVIDIA Tesla: A Unified Graphics and Computing Architecture," IEEE Micro, vol. 28, no. 2, Mar.Apr. 2008, pp. 39-55.

[26] NVIDIA, "*CUDA Toolkit 4.0 Overview*." http://developer.download.nvidia.com/compute/cuda/4_0/CUDA_Toolkit_4.0_Overview.pdf.

[27] J.E Stone, J.C. Phillips, et al., "Accelerating molecular modeling applications with graphics processors." Journal of Computational Chemistry 28(16): 2618 – 2640, 2007.

[28] L. Nyland, M. Harris, "Fast n-body simulation with CUDA." In GPU Gems 3. H. Nguyen, ed. Addison-Wesley 2007.

[29] G. Shainer, A. Ayoub et al., "The development of Mellanox/NVIDIA GPUDirect over InfiniBand—a new model for GPU to GPU communications." Computer Science - Research and Development Volume 26, Numbers 3-4, pp. 267-273, 2011.