2012 IEEE 18th International Conference on Parallel and Distributed Systems

GPGPU for Real-Time Data Analytics

Bingsheng He[1] Huynh Phung Huynh[2], Rick Goh Siow Mong[2]
[1]Nanyang Technological University, Singapore
[2]A*STAR Institute of High Performance Computing, Singapore

The demand for real-time data analytics (**RTDA**) has been on the rise in the past decades and is ever-growing with the proliferation of different data collection devices (like various sensors, camera and mobiles) and our application requirements (such as monitoring, visualization and interactive explorations). This field has been identified as one of the most exciting and promising areas for both academia and industry. In this field, we are facing the challenges at all levels ranging from sophisticated algorithms and procedures to mine the gold from massive data to high-performance computing (HPC) techniques and systems to get the useful data in time. The high-performance requirements come from the ever growing data and time-consuming analytics processes. There has been a tremendous amount of research work on data mining and processing algorithms. Instead, this tutorial focuses on the research on HPC techniques and systems.

GPGPU (General-Purpose computation on Graphics Processing Units) is an emerging research area in HPC. With the massive computation power and high memory bandwidth, GPUs have become a sharp weapon to address the performance requirement of RTDA. Designed as co-processors, GPUs pose a number of technical challenges for RTDA in terms of efficiency and programmability. On the one hand, while new-generation GPUs can have over an order of magnitude higher memory bandwidth and higher computation power (in terms of GFLOPS) than CPUs, novel GPGPU algorithmic design and implementation are a must to unleash the hardware power. On the other hand, writing a correct and efficient GPU program is still challenging in general, and even more difficult for RTDA with streaming updates and real-time multi-tasking.

In response to this situation, a number of GPGPU systems and tools (e.g., [4], [6], [5], [9], [10], [2], [12]) have been developed recently by leveraging GPGPU for (real-time) data analytics. Some studies have developed a full-fledged system for a particular RTDA application, e.g., PacketShader [4] is a high-performance PC-based software router platform that accelerates the core packet processing in Internet. In relational databases, online transactions and analytics have been accelerated with GPUs [6], [7], [8]. There have been some tools to ease the programmability of data analytics. For example, the presenters have developed GPGPU systems and tools including (1) Mars [5], [1], a MapReduce framework, (2) Medusa [11], a graph processing programming framework, and (3) automatic mapping stream programs to the GPU [9], [3]. Those systems and tools have greatly improved the programmability of GPGPU for data analytics. Users can focus on their application logic, and the details on GPGPU implementations and optimizations are hidden from users. Due to the advancement of RTDA, more GPGPU systems and tools are likely to be (re-)invented.

In this tutorial, we will discuss the open problems and challenging issues in RTDA, and urge the design and development of common systems and tools optimized for GPUs. Next, we will have an extensive review and comparative study on representative GPGPU systems and tools in detail. Still, the major focus of this tutorial is not just about introducing a wide range of systems and techniques to our audience. Rather, we endeavor to offer perspectives from a variety of different angles of looking at the common patterns in improving the efficiency and programmability of RTDA systems on GPUs. We will also demonstrate our homegrown tools on how they can support RTDA applications.

The goal of this tutorial is to provide a comprehensive introduction to current GPGPU research for RTDA to an audience with GPU computing background, interested in participating in research and/or applications of GPGPU to RTDA. We believe that this tutorial will stimulate the discussions from audience and call for further actions to address the open problems.

More details about this tutorial can be found at http://www3.ntu.edu.sg/home/bshe/GPGPUTut.html.

REFERENCES

[1] W. Fang, B. He, Q. Luo, and N. K. Govindaraju. Mars: Accelerating mapreduce with graphics processors. *IEEE Trans. Parallel Distrib. Syst.*, 22(4):608–620, Apr. 2011.

[2] A. Gharaibeh, S. Al-Kiswany, S. Gopalakrishnan, and M. Ripeanu. A gpu accelerated storage system. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, HPDC '10, pages 167–178, New York, NY, USA, 2010. ACM.

[3] A. Hagiescu, H. P. Huynh, W.-F. Wong, and R. S. M. Goh. Automated architecture-aware mapping of streaming applications onto gpus. In *IPDPS'11*, pages 467–478, 2011.

[4] S. Han, K. Jang, K. Park, and S. Moon. Packetshader: a gpu-accelerated software router. In *Proceedings of the ACM SIGCOMM 2010 conference*, SIGCOMM '10, pages 195–206, New York, NY, USA, 2010. ACM.

[5] B. He, W. Fang, Q. Luo, N. K. Govindaraju, and T. Wang. Mars: a mapreduce framework on graphics processors. In *Proceedings of the 17th international conference on Parallel architectures and compilation techniques*, PACT '08, pages 260–269, New York, NY, USA, 2008. ACM.

[6] B. He, M. Lu, K. Yang, R. Fang, N. K. Govindaraju, Q. Luo, and P. V. Sander. Relational query coprocessing on graphics processors. *ACM Trans. Database Syst.*, 34(4):21:1–21:39, Dec. 2009.

[7] B. He, K. Yang, R. Fang, M. Lu, N. Govindaraju, Q. Luo, and P. Sander. Relational joins on graphics processors. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, SIGMOD '08, pages 511–524, New York, NY, USA, 2008. ACM.

[8] B. He and J. X. Yu. High-throughput transaction executions on graphics processors. *Proc. VLDB Endow.*, 4(5):314–325, Feb. 2011.

1521-9097/12 $26.00 © 2012 IEEE
DOI 10.1109/ICPADS.2012.156

[9] H. P. Huynh, A. Hagiescu, W.-F. Wong, and R. S. M. Goh. Scalable framework for mapping streaming applications onto multi-gpu systems. In *Proceedings of the 17th ACM SIGPLAN symposium on Principles and Practice of Parallel Programming*, PPoPP '12, pages 1–10, New York, NY, USA, 2012. ACM.

[10] G. Vasiliadis, M. Polychronakis, and S. Ioannidis. Midea: a multi-parallel intrusion detection architecture. In *Proceedings of the 18th ACM conference on Computer and communications security*, CCS '11, pages 297–308, New York, NY, USA, 2011. ACM.

[11] J. Zhong and B. He. An overview of medusa: simplified graph processing on gpus. In *Proceedings of the 17th ACM SIGPLAN symposium on Principles and Practice of Parallel Programming*, PPoPP '12, pages 283–284, New York, NY, USA, 2012. ACM.

[12] Y. Zu, M. Yang, Z. Xu, L. Wang, X. Tian, K. Peng, and Q. Dong. Gpu-based nfa implementation for memory efficient high speed regular expression matching. In *Proceedings of the 17th ACM SIGPLAN symposium on Principles and Practice of Parallel Programming*, PPoPP '12, pages 129–140, New York, NY, USA, 2012. ACM.