A SURVEY ON VISUALIZATION FOR EXPLAINABLE CLASSIFIERS

by

YAO MING

Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Supervised by Prof. Huamin Qu

October 2017, Hong Kong

TABLE OF CONTENTS

ABSTRACT

CHAPTER 1

INTRODUCTION

Placeholder for introduction.

CHAPTER 2

CONCLUSION

Placeholder for Conclusion.

REFERENCES

[1] *AMD Brook+. http://ati.amd.com/technology/streamcomputing/.*

[2] *CUDA - Tutorial 5 - Performance of atomics. http://supercomputingblog.com/cuda/cuda-tutorial-5-performance-of-atomics*.

[3] *CUDPP. http://gpgpu.org/developer/cudpp/.*

[4] *Hadoop. http://ati.amd.com/technology/streamcomputing/.*

[5] *NVIDIA CUDA. http://www.nvidia.com/cuda*, 2006.

[6] *OpenCL. http://www.khronos.org/opencl/*, 2008.

[7] Anastassia Ailamaki, Naga K. Govindaraju, Stavros Harizopoulos, and Dinesh Manocha. Query co-processing on commodity processors. *VLDB*, 2006.

[8] Fischer Black and Myron S Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–54, May-June 1973.

[9] Phelim P. Boyle. Options: A monte carlo approach. *Journal of Financial Economics*, 4, 1977.

[10] Ian Buck, Tim Foley, Daniel Horn, Jeremy Sugerman, Kayvon Fatahalian, Mike Houston, and Pat Hanrahan. Brook for gpus: stream computing on graphics hardware. *SIGGRAPH*, 2004.

[11] Bryan Catanzaro, Narayanan Sundaram, and Kurt Keutzer. A map reduce framework for programming gpus. In *STMCS*, 2008.

[12] Maria Charalambous, Pedro Trancoso, and Ros Stamatakis. Initial experiences porting a bioinformatics application to a graphics processor. *Lecture notes in computer science*, 2005.

[13] Cheng-Tao Chu, Sang Kyun Kim, Yi-An Lin, Yuan Yuan Yu, Gary Bradski, Andrew Y. Ng, and Kunle Olukotun. Map-reduce for machine learning on multicore. In *NIPS*, 2006.

[14] Marc de Kruijf and Karthikeyan Sankaralingam. Mapreduce for the cell b.e. architecture. Technical report, University of WisconsinâĂŞMadison, 2007.

[15] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. *OSDI*, 2004.

[16] Jimin Feng, Samarjit Chakraborty, Bertil Schmidt, Weiguo Liu, and Unmesh D. Bordoloi. Fast schedulability analysis using commodity graphics hardware. *RTCSA*, 2007.

[17] Naga Govindaraju, Jim Gray, Ritesh Kumar, and Dinesh Manocha. Gputerasort: high performance graphics co-processor sorting for large database management. *SIGMOD*, 2006.

[18] Naga K. Govindaraju, Brandon Lloyd, Wei Wang, Ming Lin, and Dinesh Manocha. Fast computation of database operations using graphics processors. *SIGMOD*, 2004.

[19] Bingsheng He, Wenbin Fang, Qiong Luo, Naga K. Govindaraju, and Tuyong Wang. Mars: a mapreduce framework on graphics processors. In *PACT*, 2008.

[20] Bingsheng He, Naga K. Govindaraju, Qiong Luo, and Burton Smith. Efficient gather and scatter operations on graphics processors. *Supercomputing*, 2007.

[21] Bingsheng He, Ke Yang, Rui Fang, Mian Lu, Naga Govindaraju, Qiong Luo, and Pedro Sander. Relational joins on graphics processors. *SIGMOD*, 2008.

[22] Changhao Jiang and Marc Snir. Automatic tuning matrix multiplication performance on graphics hardware. *PACT*, 2005.

[23] Andrew Kerr, Gregory Diamos, and Sudakhar Yalamanchili. Modeling gpu-cpu workloads and systems. In *GPGPU-3*, 2010.

[24] Michael D. Linderman, Jamison D. Collins, Hong Wang, and Teresa H. Meng. Merge: a programming model for heterogeneous multi-core systems. *ASPLOS*, 2008.

[25] Michael D. McCool. Data-parallel programming on the cell be and the gpu using the rapidmind development platform. In *GSPx Multicore Applications Conference*, 2006.

[26] NVIDIA corp. *NVIDIA CUDA Programming Guide 2.0*, 2008.

[27] John D. Owens, David Luebke, Naga Govindaraju, Mark Harris, Jens KrÃijger, Aaron E. Lefohn, and Timothy J. Purcell. A survey of general-purpose computation on graphics hardware. *Computer Graphics Forum*, 2007.

[28] Colby Ranger, Ramanan Raghuraman, Arun Penmetsa, Gary Bradski, and Christos Kozyrakis. Evaluating mapreduce for multi-core and multiprocessor systems. *HPCA*, 2007.

[29] Shubhabrata Sengupta, Mark Harris, Yao Zhang, and John D. Owens. Scan primitives for gpu computing. *Graphics Hardware*, 2007.

[30] David Tarditi, Sidd Puri, and Jose Oglesby. Accelerator: using data parallelism to program gpus for general-purpose uses. *ASPLOS*, 2006.

[31] Vasily Volkov and James W. Demmel. Benchmarking gpus to tune dense linear algebra. *Supercomputing*, 2008.

[32] Hungchih Yang, Ali Dasdan, Ruey-Lung Hsiao, and D. Stott Parker. Map-reduce-merge: simplified relational data processing on large clusters. *SIGMOD*, 2007.

[33] Jackson H.C. Yeung, C.C. Tsang, K.H. Tsoi, Bill S.H. Kwan, Chris C.C. Cheung, Anthony P.C. Chan, and Philip H.W. Leong. Map-reduce as a programming model for custom computing machines. In *FCCM*, 2008.

[34] Richard Yoo, Anthony Romano, and Christos Kozyrakis. Phoenix rebirth: Scalable mapreduce on a numa system. In *IISWC*, 2009.