

Contents

Contents	i
List of Figures	iii
List of Tables	v
1 Introduction	1
1.1 Motivation	1
1.1.1 Goals of Research	5
1.1.2 Thesis Organisation	8
References	13

List of Figures

Thesis is available at: <http://dmacssite.github.com/Thesis.pdf>

List of Tables

Thesis is available at: <http://dmacssite.github.com/Thesis.pdf>

1. Introduction

High performance computing power is believed to be the key to process all the scientific and data intensive applications, industry research, big data analysis, data science research etc. Major scientific discoveries and analyses are accelerated by utilizing these high end computing facilities. These escalated computational requirements of these applications has motivated to develop accelerators like CellBE processor, GPU, Xeon Phi architectures. Parallelism at the scale of teraflop to petaflop can be seen in these accelerators. We can exploit these inherent parallelism in these accelerators to improve the overall performance of the given class of scientific applications.

1.1 Motivation

In recent years, one of the major challenges for computational scientists is to deal with applications manipulating large data sets running on the parallel machines. Scientific applications with large data sets will necessarily involve a great deal of transfers from memory to the disk. Computationally intensive challenges can be resolved by using machines with large number of processors or cores. This has lead to the revolution of multi core, many core processors and also massively

Thesis is available at: <http://dmacssite.github.com/Thesis.pdf>

parallel systems based on such processors. However, it is observed that the performances of I/O subsystems of these machines are not improving in the same proportion. Due to this increasing gap between the power of computation and ability to handle I/O in modern machines, I/O bottleneck issues have become one of the major areas of research concerning the High Performance Computing Community.

Some of the applications like Geographical Information System (GIS) [10] [11], Digital Elevation Model (DEM)[12], financial forecast model simulations [13], geometrical topology data processing, Medical Imaging Tomography calculation [14] and visualization [15] are some of the major illustrations of such data intensive applications where I/O data transfer rate has significant impact on their overall performance.

As parallel machines are becoming faster, these are increasingly being used to solve problems requiring large amount of computing power. High performance computing is playing an important role in science and engineering and is enabling highly accurate simulations of complex phenomenon. Previously, the scientists used to rely upon experimental evidences to ascertain their theoretical findings. In modern times, the types of systems scientists study and the kind of methodology used have pushed the paradigm towards simulation based research. Nowadays, scientists prefer simulation based verification over experimental and empirical studies to validate their theoretical models in order to arrive at conclusion. The availability of these massively parallel machines has facilitated new ways of doing scientific research.

We see that some of the data intensive applications like DNA sequencing in human genome, climate change model from National Center for Atmospheric Research (NCAR) [3] and the Center for Ocean-

Atmospheric Prediction Studies (COAPS) at Florida State University [4], run projects that monitor and model the hurricanes due to various ocean-atmospheric variations. These applications are based on simulations performed using massively parallel systems. It is reported that more than petabytes of data needs to be processed requiring massively parallel machines for their computation.

In a typical data intensive application, the I/O activities encountered are of the following types: reading data from disk to memory, writing the results back to the disk, checkpointing in case of fault tolerant applications. Since I/O operations are cost intensive, in comparison to the cost of computation it becomes the bottleneck. To make efficient use of the massively parallel machines, we need to overcome the relatively poor performance of I/O subsystems.

Over the years there has been a lot of improvements in the computation and communication performance of parallel computers. Titan-Cray XK7 at Oak Ridge National Laboratory, Sequoia- BlueGene/Q at Lawrence Livermore National Laboratory, and Mira- BlueGene/Q at Argonne National Laboratory [7] have systems which are achieving speed of more than Petaflops with Linpack benchmark. Similar improvements are not seen in the I/O performance of parallel machines. Even though the computing power with peak performance is now exceeded to peta flops, the real data intensive applications running on these high end systems usually achieve an I/O bandwidth of at most in the range of Giga Bytes only.

I/O has been a somewhat neglected area in high end computing, relative to computational speeds, and has not improved with time. For example, the fastest supercomputer in 1999 [5] was the ASCI Red with a

total theoretical peak compute performance of 3.2 teraflops/s (Tflop/s) and bandwidth to disk being 4 GB/s. The total bandwidth to disk on the Stampede at Texas Advanced Computing Center, which ranked 7 in the recent top500 supercomputer list, is about a hundred times more, at 352 GB/s. However, the total compute performance is about 1500 times faster at 5168.1 Tflop/s [8]. This gap is expected to increase further as the computing community is actively planning to have exaflop machine in the near future.

In BlueGene/Q [6], the ratio of compute node and I/O node is 128:1. I/O cores are one or two magnitudes lesser than that of the computation cores. This points out that the I/O communication overheads need to be reduced or I/O performance needs to be enhanced for improving the overall computation efficiency of these machines.

These I/O Bottleneck issues can be addressed by considering various methods, like reducing the communication cost, filtering of the data in I/O servers and prefetching of data using various prediction algorithms. In the recent times, the availability of multi-cores and multiple processor based systems provides an ample opportunity for parallelization. The fact that data can be stored on separate physical disks, and processor units have separate main memories implies that minimizing the communication cost will lead to considerable improvement in overall performance.

In order to minimize the communication cost one can compress the given data, which will increase the effective bandwidth available, thereby reducing the communication cost. Subsequently, enabling query processing over the compressed data offers a greater opportunity to reduce the cost of I/O. In the literature a number of prediction algorithms

to prefetch the data in order to reduce this I/O bottleneck are proposed.

In this work we have shown how compression algorithms can increase the performance of the computing system for the various scientific applications.

1.1.1 Goals of Research

In the first part of the thesis, we discuss about how the parallel compression algorithms using predictions can affect the performance of generic applications using accelerators. Accelerators such as Graphics Processing Units (GPU) and multi-core/ many-core processors provide high computational power. We show that the computational power of these accelerators can be harnessed to provide sufficiently fast compression and decompression. Adding to it, we have latest technology like nvlink [1] from Nvidia and IBMs CAPI [2] for improving CPU-GPU communication which increases the performance of the overall computation. In this work, we have shown that compression is improving the overall performance of the high end computing system. We have highlighted two parameters, throughput and compression ratio for the I/O performance. Compression has been proposed by various other researchers in the past[9] but we have focused more on compression ratio along with throughput to reduce the computation overhead.

In particular, we used simple mathematical techniques, time series analysis, to focus more on throughput than compression ratio. We show that it can be implemented efficiently on multicore/many-cores and GPUs to provide an effective compression scheme for the purpose of saving on data transfer overheads.

Initially we observed prediction is based on linear fitting prediction, quadratic fitting algorithms [we need to site here], last valued based technique for compressing the data. It predicts the next value, based on previous observed values. If the prediction is accurate, then an exclusive-OR of predicted value with the original value is stored. It yields several zeros in most significant bits. These can be compressed efficiently. Then we have applied time series based prediction algorithm called holt's linear exponential smoothing [30] for our floating point compression. We have particularly used to holt's algorithm because it requires much less computation than that of the other prediction model in the literature with good degree of prediction. Our main focus is to have high network throughput so that we need to faster compression than the better compression ratio.

We have implemented these parallel prediction algorithms in various accelerators including GPUs, CellBE Processor and Intel MIC. These algorithms are:

- Holt's exponential smoothening algorithm
- Last Value based algorithm
- Parallel Linear fitting based compression algorithm

The other area we consider to demonstrate a similar benefit of compression compression is the DNA Sequencing in the domain of Bioinformatics. We show that compression methods can be developed with high throughput that can yield substantial improvement in performance of applications in high end computing systems which has I/O bottleneck issues. The issue of sequence analysis is a core problem in Bioinformatics. Various strategies and implementations have been

used to address the issue. But the amount of data that is generated and transferred remains a big challenge.

To overcome this challenges, we propose compression technique on the DNA-Sequences. We discuss compression algorithms on DNA sequencing in human genomes. The genome consists of all the hereditary information for running and maintaining an organism. This information contained in genome is encoded in the form of DNA. DNA chain is made of four bases: Adenine (A), Guanine (G), Thymine (T), and Cytosine (C). When these cells divide to grow, every new cell which emerges from the parent cell needs a copy of the DNA to function properly. For this sake, DNA replicates itself before the cell divides into the child. Because of this process, the genomic data increases constantly which leads to doubling of the DNA sequences. DNA also has many repeats in its sequence of AGTC or combination of these four bases. This property of the DNA can be used to compress the data using different technique. Moreover these DNA sequence application will be one of the prospective big data application in near future. In future we need to focus more on sequencing in compressed domain.

Unless this is achieved, the task remains half done at best in the future. Any of these activities can be carried out on any hardware architecture. However, the recent revolution in this area leads us to multi-core processors and GPUs as the preferred choice for the hardware platform to use for high performance at low cost and power.

Pattern matching, subsequence matching, and alignment of DNA Sequences in the compressed form are built on top of our compression techniques of these DNA sequences. Results demonstrate a performance on par with standard results available in the literature which works with

the original sequences and not in the compressed form. We also show that the performance of our implementations scale well on multi-core and GPUs.

In this thesis, we target the implementation details of our technique for GPU and many core processor, which is used as an accelerator in the high end computing systems. With the abundance of computing power provided by accelerators, such as GPUs and many core processors, we hope that our work will further stimulate research in high throughput floating point compression techniques to ameliorate bandwidth bottlenecks.

1.1.2 Thesis Organisation

This thesis is organized in the following manner. Chapter 2 provides an overview of the existing parallel accelerators and its performance. This study of accelerators is required for understanding of the architecture and also to find the underling parallelism. We also studied the different programming models for these architectures optimizing the overall performance of our algorithms. Subsequently, in Chapter 3 we describe the related work in the literature which tackle the I/O subsystem bottleneck. In chapter 4, we describe the various parallel algorithms we have implemented to solve I/O issues. Chapter 5, presents the results for each of the algorithm we have implemented and finally in Chapter 6 presents the conclusion and scope for the future study in this domain.

Bibliography

- [1] Nvidia, "Nvidia," Nvidia, November 2014. [Online]. Available: [info.nvidianews.com/rs/nvidia/images/NVIDIA NVLink High-Speed Interconnect Application Performance Brief.pdf](http://info.nvidianews.com/rs/nvidia/images/NVIDIA_NVLink_High-Speed_Interconnect_Application_Performance_Brief.pdf).
- [2] IBM, "IBM," 29 September 2014. [Online]
- [3] <http://www.ncar.ucar.edu/research/climate/>: example for Climate model and Weather forecasting.
- [4] COAPS. The Center for Ocean-Atmospheric Prediction Studies at Florida State University. <http://www.coaps.fsu.edu/>.
- [5] GETTING UPTO THE SPEED, The Future of Computing by natural research council of the national academies. Editors are Susan L. Graham, Marc Snir, and Cynthia A. Patterson.
- [6] <https://computing.llnl.gov/tutorials/bgq/#Software>
- [7] <https://www.alcf.anl.gov/mira>
- [8] <https://portal.tacc.utexas.edu/user-guides/stampede>
- [9] Martin Burstscher et. al. "Floating-point data compression at 75 Gb/s on a GPU" Published in ACM article 2011 [<http://dl.acm.org/citation.cfm?id=1964189>]
- [10] Andrew Danner , Thomas Mlhave , Ke Yi , Pankaj K. Agarwal , Lars Arge , Helena Mitasova TerraStream: from elevation data to watershed hierarchies, Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems, November 07-09, 2007, Seattle, Washington.
- [11] Andrew Danner, Pankj Agarwal, Lars Arge "I/O efficient algorithms and applications in geographic information systems", 2006 ISBN: 978-0-549-02200-8

Thesis is available at: <http://dmacssite.github.com/Thesis.pdf>

- [12] <http://www.terrainmap.com/>
- [13] <http://pcbunn.cithec.caltech.edu/results/model/model.html>
- [14] V. Messereli, O. Figueiredo, B. Gennart, et.al "Parallelizing I/O Intensive Image Access & Processing Applications". IEEE concurrency, Vol. 7, No. 2, April-June 1999, 28-37.
- [15] Tom Peterka, Hongfeng Yu, Robert Ross, Kwan Liu Ma "Parallel Volume Rendering on the IBM Blue GeneP", submitted to Eurographics Symposium on Parallel Graphics and Visualization, 2008.
- [16] Ajith Padyana, T.V. Siva Kumar and P.K. Baruah, Fast Floating Point Compression on the Cell BE Processor, In Proceedings of the 15th IEEE International Conference on High Performance Computing (HiPC), 2008.
- [17] V.Engelson, D.Fritzson and P.Fritzson. Loseless Compression of highvolume numerical data from simulations. In Proceedings of the IEEE Data Compressions Conference (DCC), pages 574-586,2000
- [18] P Ratanaworabhan, J .Ke and M . Burtscher .Fast Loseless Compression of scientific floating point data. In IEEE Data Compression Conference (DCC) pages 133-142,2006
- [19] P. Lindstrom and M. Isenburg. Fast and Efficient Compression of Floating-Point Data. IEEE Transaction on Visualization and Computer Graphics, Vol. 12, No. 5. September 2006.
- [20] L.Ibarria, P. Lindstrom, J. Rossignac and A. Szymczak. Out of Core Compression and Decompression of Large n-Dimensional Scalar Fields. Eurographics, pp.343-348. September 2003.
- [21] E Ghido An Efficient Algorithm for Lossless Compression of IEE Float Audio Data Compression Conference, pp 429-438. March 2004.
- [22] M.Burtscher and P.Ratanaworabhan. High throughput compression of double-precision floating point data. In Proceedings of the IEEE Data Compression Conference (DCC),2007

- [23] M. Burstcher, P. Ratanaworabhan pFPC: A Parallel Compressor for floating point data, Data Compression Conference, pp 43-52. March 2009.
- [24] D. Chen, N. Memon. Lossless Compression of point based 3D models Pacifics Graphics. Pp 124-126. October 2005.
- [25] Time Series forecasting using Holt- Winters exponential smoothening. by Prajakta S. Kalekar, kanwal Rekhi School of Inforamtion Technology, under the guidance of Prof.Bernard.
- [26] http://en.wikipedia.org/wiki/Time_series # Prediction and forecasting
- [27] Martin Burtscher and Paruj Ratanaworabhan, High Throughput Compression of Double-Precision Floating Point Data, Cornell University, Ithaca.
- [28] <http://www.nvidia.com/object/io1258360868914.html>
- [29] Ajith Padyana, T.V. Siva Kumar and P.K. Baruah, *Fast Floating Point Compression on the Cell BE Processor*, In Proceedings of the 15th IEEE International Conference on High Performance Computing (HiPC), 2008.
- [30] [http://www.personal.soton.ac.uk/rchc/Teaching/MATH6011/Forecasting Chap3.htm](http://www.personal.soton.ac.uk/rchc/Teaching/MATH6011/ForecastingChap3.htm)
- [31] <http://www.ncsa.illinois.edu/News/11/0518NCSAinstalling.html>
- [32] University of Florida Sparse Matrix Collection, <http://www.cise.ufl.edu/research/sparse/matrices/>
- [33] developer.download.nvidia.com/
- [34] S. Williams, L. Oliker, R. Vuduc, J. Shalf, K. Yelick, J. Demmel, Optimization of Sparse Matrix-Vector Multiplication on Emerging Multicore Platforms, Supercomputing (SC), 2007.
- [35] V.Engelson, D.Fritzson and P.Fritzson. Loseless Compression of high-volume numerical data from simulations. In Proceedings of the IEEE Data Compressions Conference (DCC), pages 574-586,2000

- [36] M.Burtscher and P.Ratanaworabhan. High throughput compression of double-precision floating point data. In Proceedings of the IEEE Data Compression Conference (DCC),2007
- [37] P Ratanaworabhan, J .Ke and M . Burtscher .Fast Loseless Compression of scientific floating point data. In IEEE Data Compression Conference (DCC) pages 133-142,2006
- [38] <http://www.ncsa.illinois.edu/News/11/0518NCSAinstalling.html>
- [39] “DNA Sequence Compression using the Burrows-Wheeler Transform”, IEEE, 2002.
- [40] TC Bell et al., Prentice Hall, 1990.
- [41] A Grumbach & F Tahi, “Compression of DNA sequences.”, Proceedings of the “IEEE Data Compression Conference, Snowbird, UT, USA.”,March 30April 2. 1993.
- [42] Xin Chen, Sam Kwong, Ming Li, “A Compression Algorithm for DNA Sequences.”,Proceedings of the Fourth Annual International Conference on Computational Molecular Biology, Tokyo, Japan,April 8-11, 2000.
- [43] Xin Chen 1, Ming Li, Bin Ma, and John Tromp. “DNACompress: fast and effective DNA sequence compression”,volume 18, pages 1696-1698, 2002.
- [44] P. Raja Rajeswari & Dr Allam AppaRao, “DNABIT Compress Genome compression algorithm”, Book on Bioinformation, 2011, volume 5, pages 350-360.
- [45] P Raja Rajeswari & Dr Allam AppaRao,GENBIT Compress-Algorithm for repetitive and non repetitive DNA sequences,International Journal of Computer Science and Information Technology,2010 ,volume 2,pages 25-29.
- [46] Matsumoto,T., Sadakane,K. and Imai,H, “Biological sequence compression algorithms”, Genome Informatics Workshop, Universal Academy Press, 2002,43-52.
- [47] J. Ziv and A. Lempel,“A universal algorithm for sequential data compression ”,IEEE Trans. Inform. Theory,1977,volume -23, pages 337-343.

- [48] Ziv and Lempel, "Compression of individual sequences via variable-rate coding.", IEEE Transactions on Information Theory, September 1978, volume 24, pages 530-536.
- [49] Ma, B., Tromp, J. and Li, M. "PatternHunter : faster and more sensitive homology search.", Bioinformatics, 2002, pages 440-445.
- [50] Jackson, Peter, Book Chapter "Introduction To Expert Systems", Addison Wesley, 1998 , 3rd edition, ISBN 978-0-201-87686-4.
- [51] Regina Barzilay and Daryl McCullough and Owen Rambow and Jonathan Decristofaro and Tanya Korelsky and Benoit Lavoie and Cogentex Inc , "A New Approach to Expert System Explanations", In 9th International Workshop on Natural Language Generation, 1998, pages 78-87.
- [52] Russell, Stuart J.; Norvig, Peter, "Artificial Intelligence: A modern approach", Prentice Hall, 2003, 2nd Edition.
- [53] Nils J. Nilsson, "Artificial Intelligence: A New Synthesis", Morgan Kaufmann Publishers, San Francisco, 1998.
- [54] McCorduck, Pamela, "Machines Who Think", A. K. Peters, Ltd., 2004.
- [55] Koch, C.G., Isle, B.A. ; Butler, A.W., "Intelligent user interface for expert systems applied to power plant maintenance and troubleshooting", IEEE Transactions, March 1988, 3rd volume.