



A Survey on Accelerated Mapreduce for Hadoop

JYOTINDRA TIWARI^{1*}, DR. MAHESH PAWAR² and DR. ANJANA PANDEY³

¹School of Information Technology, RGPV, bhopal, India.

²Department of IT, UIT RGPV, bhopal, India.

³Department of IT, UIT RGPV, bhopal, India.

Abstract

Big Data is defined by 3Vs which stands for variety, volume and velocity. The volume of data is very huge, data exists in variety of file types and data grows very rapidly. Big data storage and processing has always been a big issue. Big data has become even more challenging to handle these days. To handle big data high performance techniques have been introduced. Several frameworks like Apache Hadoop has been introduced to process big data. Apache Hadoop provides map/reduce to process big data. But this map/reduce can be further accelerated. In this paper a survey has been performed for map/reduce acceleration and energy efficient computation in quick time.



Article History

Received: 13 May 2017

Accepted: 23 June 2017

Keywords

Map Reduce,
GPU computation
and Open CL.

Introduction


The amount of data generated in the digital world has grown rapidly in last few years. Storage and processing of this huge amount of data is a difficult task, yet an essential one. Various efforts are being made to store and process this huge information in a quick and efficient way. Thus, there is a need of an efficient and robust framework to handle such a huge quantity of data. One such framework is Hadoop Map Reduce which has picked up importance in the recent couple of years for storage and processing of big data. It is a framework for parallel processing of big data in distributed environment. It is open source, written in java, uses commodity hardware and supports Map Reduce for distributed processing. It is the framework which

provides the environment where data is stored and process across the cluster of computational units. It gives scalable, economical and less demanding approach for parallel processing of information on large computational units.

Hadoop was motivated by Google File System (GFS) and Map Reduce paradigm in which input data is broken down into smaller size blocks and to process these data blocks that is stored across the cluster, mapper and reducer task are created. Hadoop framework supports the Map Reduce Processing paradigm and is designed to support the storage and processing of huge information data sets.

CONTACT Jyotindra Tiwari ✉ tiwarijyotindra@gmail.com 📍 School of Information Technology, RGPV, bhopal, India.

© 2017 The Author(s). Published by Enviro Research Publishers

This is an  Open Access article licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License (<https://creativecommons.org/licenses/by-nc-sa/4.0/>), which permits unrestricted NonCommercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

To link to this article: <http://dx.doi.org/10.13005/ojcs/10.03.07>

It gives scalable, economical and less demanding approach for parallel processing of information on large computational units. It also supports distributed architecture where the data is stored across the cluster of commodity hardware. It is best suitable for batch oriented processing like log processing and text processing etc.

Hadoop framework is composed of two parts:

- Hadoop Distributed File System – for storing the data
- Map Reduce – for processing the data.

Introduction To Gpu Computing By Open cl

Big Data tasks are computationally intensive thus require a huge amount of computational resources. Traditionally CPUs were used for computing and if the task was too computationally intensive then a CPU cluster could be used. For such tasks GPUs have proved to be better when compared with a CPU cluster. A GPU has hundreds of simple cores whereas a CPU has very few but complex cores. GPUs can process thousands of concurrent hardware threads but a CPU has single-thread performance optimization. The cost of performing a computationally intensive task such as training a neural network is cheaper on a GPU when compared with a CPU cluster having same number of cores. Graphic processing units or GPUs are built for graphical computations. A special GPUs are built for non-graphic processing tasks, they are called GPGPUs- general purpose graphic processing unit.

A lots of heterogeneous platforms are available these days such as multi-core central processing units (CPUs), field programmable gate arrays (FPGAs), digital signal processors (DSPs), graphics processing units (GPUs) and heterogeneous accelerated processing unit (APUs). A standard is needed to support the processing over these heterogeneous systems. Open CL is defined as such a standard which helps in program execution over different heterogeneous devices to extract parallelism.

Open CL hides the hardware complexity and makes the program portable between different heterogeneous devices.

Hadoop CL combines the strengths of Hadoop and Open CL and provides a high performance distributed system. Hadoop CL execute inherently parallel Map Reduce code written by user on heterogeneous devices. The processing of data is done at thread-level (Intra-node) at each cluster (Inter- node) of Hadoop CL using Open CL kernels code. So, these two level parallelisms produce the result much faster and the recommendations provided to the users are much faster, efficiently and more accurately. The utilization of resources is improved and it hides the hardware complexities from the users.

Features Of Hadoop Cl

- It is time efficient with a overall speedup of three times.
- Delivers high performance and is also energy efficient.
- A reliable and robust distributed system and performance better than Hadoop.
- Hardware complexity is hidden from the user to code easily and efficiently.
- Allow tuning experts to manipulate platform configuration in order to optimize performance, energy efficiency and reliability.
- Integration of Hadoop and Open CL is done by APARAPI to deliver a heterogeneous distributed system.
- Asynchronous communication and dedicated communication thread is used.

Literature Review

Dumitrel Loghin *et al.*, 2015¹

Propose an energy efficient approach for parallel programming using a hybrid application of MPI and Open MP. When the execution time is fixed then the most energy efficient configuration use the minimum energy over all given configurations. They have used a measurement-driven analytical model to determine the energy efficient approach. The proposed approach gives a energy-efficient Pareto-optimal configurations in terms of the number of nodes, core clock frequency and number of cores per node. The given configurations use minimum energy for a fixed execution time or execute in minimum time for fixed energy consumption.

Sung Ye Kim *et al.*, 2015²

have tried to make a custom Hadoop framework to accelerate Map Reduce programming by powering with GPUs. A better performance and power gains are expected if current integrated GPU are used for compute intensive code blocks because of their high performance per watt. Intel GPUs are used via Open CL and K-means clustering algorithm is applied using mahout machine learning library of Hadoop. The benchmarking is performed through Hi Bench benchmarking suite. A good speedup and significant power reduction is observed. A speedup of 45x for map tasks and 4.37x for K-means clustering is observed.

Motahar Reza *et al.*, 2015³

presents a model and implementation to perform Sp MV using Hadoop-CUDA Hybrid approach. Sparse Matrix Vector Multiplication (SpMV) is a very important part of various scientific computations. The SCOO format is used, which is the best format on GPU for the computation on CUDA based on performance. Input matrix is split into smaller sub-matrices using Hadoop. Individual data nodes store these sub-matrices. Required CUDA kernels are then invoked on the individual GPU-possessing cluster nodes. Experiments performed to compare the performance of Hadoop-CUDA cluster and non-Hadoop CUDA system shows that hybrid cluster is performing better and a speedup of around 1.4 is observed.

Jie Zhu *et al.*, 2014⁴

proposed an integration of Hadoop computation with CUDA to exploit resources of CPU and GPU. CUDA

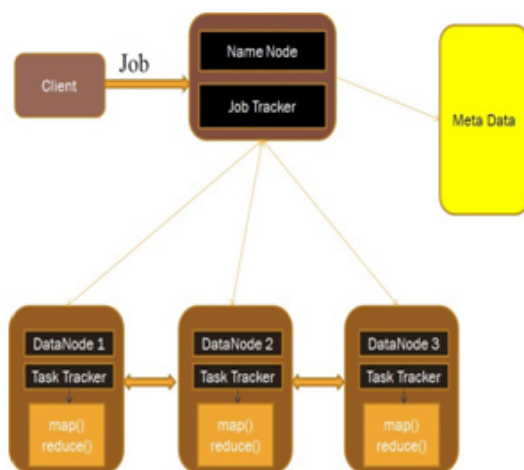
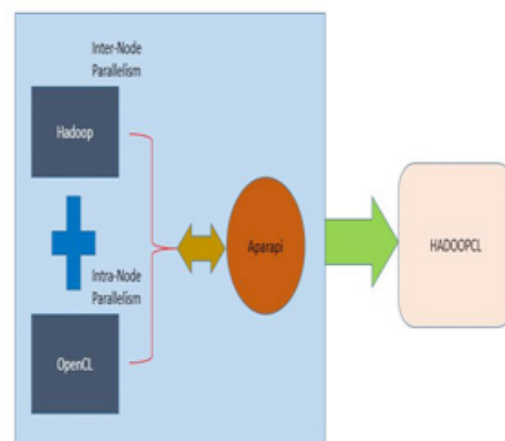
helps in execution of user-written Hadoop code on GPU native threads which are heterogeneous and consumes low power. Hadoop integration with GPU is possible by using any of the four approaches which are JCUDA, JNI, Hadoop Streaming and Hadoop Pipes. Authors have implemented all the four approaches and done a detailed analysis. A comparison of these four approaches shows that JCudais the highest performing and Hadoop Streaming is worst performing. It is difficult to test JNI. but it is also the most expensive for development and translation. Hadoop Pipes show above average performance in all criteria and have not shortcomings. Hadoop Streaming shows best results in all criteria except High performance.

Sufeng Niu *et al.*, 2014⁵

have worked on combining Hadoop and GPGPU for effective processing for microarray data. A new set of tools is developed to process microarray data. Hadoop is used for data intensive tasks and General-Purpose Graphics Processing Units (GPGPUs) for compute-intensive tasks. The proposed approach is effective for complex scientific tasks that contain data and compute intensive tasks. Microarray data is gathered by biologists and valuable information can be mined from this data. Microarray data is huge amount of data and requires huge amount of resources to process. The tools developed by authors guarantee a superior performance when evaluated on a large microarray data sets.

Mayank Tiwary *et al.*, 2014⁶

propose an algorithm by which time efficient

**Fig.1: Hadoop architecture****Fig. 2: HadoopCL architecture**

solution to Apriori data mining technique is given by integrating GPU in Map Reduce programming model. The part of algorithm which requires intensive computation is off loaded to the GPU. For comparing the serial and parallel Apriori algorithm 4 node cluster is used and to each node NVIDIA's GPU was attached. Results of the experiments show that a maximum speedup of 18x is seen on GPU cluster.

Can Basaran *et al.*, 2013⁷

present Grex which is a GPU based Map Reduce framework to boost parallel data processing. A parallel split phase is used in place of split phase of Map Reduce to efficiently handle variable size data. To avoid any data partitioning skews Grex performs an even distribution of data. A new scheme for memory management is also provided by Grex. All these capabilities make Grex much faster than current GPU-based Map Reduce frame work.

Ranajoy Malakar *et al.*, 2013⁸

build a high performance system for image processing on heterogeneous devices by integrating CUDA acceleration into Hadoop framework. Experiments were performed using a face processing algorithm which is based on the Adaboost learning system. Experimental evaluations indicate that CUDA enabled Hadoop cluster gives an improvement of 25% in processing throughput even with a low end GPU. The scalable system build by integration of these two technologies delivers high throughput and is power efficient as well as provides cost-efficient computing.

Yanlong Zhai *et al.*, 2013⁹

have presented a new big data framework which is called Lit and it delivers high performance. Lit leverages the power of Hadoop by using Hadoop with GPU cluster. Authors have designed and implemented a basic architecture of Lit and have stressed a lot on optimizing communications between GPUs and Hadoop. Lit has been designed to automatically produce CUDA codes using Hadoop codes. Lit improves the computational capability of each node thus enhancing overall computation capability of cluster using GPU. Lit hides the programming complexity by extending the optimizer and compiler. The communications between Hadoop and CPU-GPU provide a major bottleneck in co-processing systems. Experiments show a speedup of 1x to 3x on three applications on Hadoop: Matrix Multiplication, Scan and Fast Fourier Transform. A 16 percent performance gain is also observed using data flow optimization.

Hao Li *et al.*, 2012¹⁰

present a new approach of Map Reduce acceleration with GPU. The implementation is done using Hadoop and integrating it with Open CL. To achieve a better seamless-integration Hadoop is integrated with Open CL using Open CL Java language binding (JOCL). Authors have tried to extend a multi-machine alone parallelism of Map Reduce model to multi-machine along with multi-core. It is aimed at both data and compute-intensive applications. The results show a comparison between the time taken executing the same job on Map Reduce (MR) and MRCL (Map Reduce with Open CL). A speedup of more than 1.6 is observed for all scale of data.

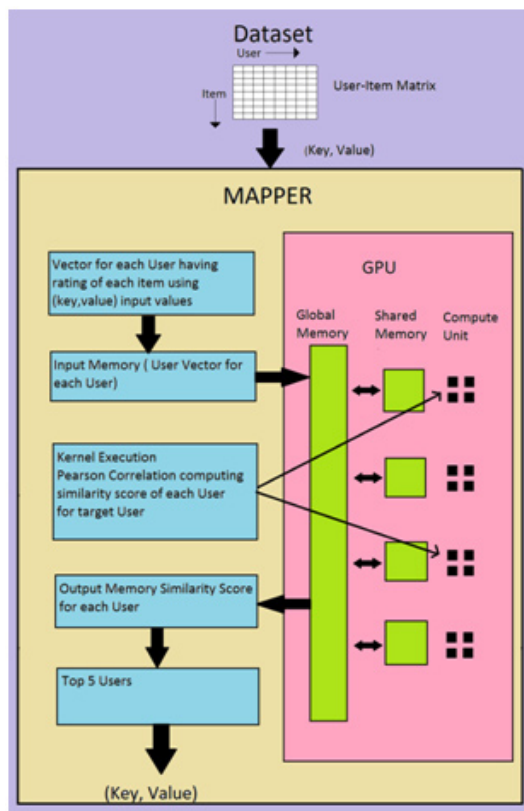


Fig. 3 Recommender System on HadoopCL

Wenbin Fang *et al.*, 2011¹¹

proposed a runtime system on Map Reduce called MARS which is accelerated with GPU. Authors implemented Mars on several heterogeneous platforms. The design of Mars was aimed at enhancing programmability, flexibility and high performance. The aim of MARS was Single machine implementation of Mars called Mars CUDA was developed to integrate Mars Map Reduce into NVIDIA's GPU. Mars Brook was developed for integrating Hadoop with AMD GPU and similarly Mars CPU for a multi core CPU. An implementation of Mars outperformed the state-of-the-art Map Reduce, The results have shown that an implementation of Mars outperformed Phoenix which is a state-of-the-art Map Reduce framework on the multi core CPU with a speedup to 72 times and 24 times on average depending on the application.

Koichi Shirahata *et al.*, 2010¹²

presents a hybrid map task execution technique for GPU-based heterogeneous computer clusters. The prototype of proposed scheduling technique is implemented by extending Hadoop Map Reduce framework. Their approach minimizes the execution time of a submitted Map Reduce job by using dynamic monitoring of mapper tasks elapsed time and other such behavior running on CPU cores and GPU clusters. For evaluation the proposed technique is applied on a GPU-based supercomputer. K-means clustering algorithm is used as a benchmark. K-means clustering using proposed technique is 1.93 times faster than the Hadoop original scheduling.

Proposed Work

Recommender systems are widely used these days in areas including movies, music, news, ecommerce websites, social networking etc. Recommender systems are used to predict the 'rating' or 'preference' that a user would give to an item. Two most commonly used type of recommender systems are collaborative and content-based filtering. Collaborative filtering method generates recommendations for a user by collecting preferences from many users. The basic

idea is that if two users have same opinion on an issue then it is more likely that they have same opinion on a different issue than that of a randomly chosen person.

Collaborative filtering recommendations are done in two ways either user-user collaborative filtering or item-item collaborative filtering. In user-user collaborative filtering based recommender systems the top users who share the same rating patterns with the active user are considered. The rating of only these users is used to make recommendations for the active user. The recommender systems use Big Data processing framework like Hadoop. Hadoop is scalable and does efficient processing of Big Data. Every computer system comes with multiple cores of CPU, GPUs, APUs, and FPGAs etc. The efficiency of the system improves if we utilize these resources properly. Hadoop doesn't utilize these resources which may lead to poor computational performance. Another problem with Hadoop is it consumes high energy. Considering these problems in mind, a recommender system is implemented on Hadoop CL¹³. Hadoop CL uses Open CL to utilize the resources like cores of CPUs, GPUs, APUs, FPGAs, etc. Hadoop provides parallelism in distributed environment whereas Open CL provides parallelism in heterogeneous environment. Because of this integration, the two level parallelism: Inter-Node and Intra-Node parallelism is achieved. A data set is distributed block-wise on each system of cluster of Hadoop in distributed manner and further distributed on cores of CPU and threads of GPUs in heterogeneous manner. It makes the complete system faster and improves the computational performance of the system.

Conclusion

In this paper a survey has been done on the map/reduce acceleration techniques. It is found that these techniques provide a energy efficient and quick time computation for big data. It is concluded in this paper that map/reduce acceleration can be achieved using GPU computation also. GPU computation when coupled with Apache Hadoop utilizes heterogeneous environment.

References

1. Dumitrel Loghin, Lavanya Ramapantulu, Yong Meng Teo "An Approach for Energy Efficient Execution of Hybrid Parallel Programs" in IEEE International Parallel and Distributed Processing Symposium 2015.
2. SungYe Kim, Jeremy Bottleson, Jingyi Jin, PreetiBindu "Power Efficient MapReduce Workload Acceleration Using Integrated-GPU", in IEEE First International Conference on Big Data Computing Service and Applications (Big Data Service), pp.162-169. 2015.
3. Motahar Reza, Aman Sinha, Rajkumar Nag, Prasant Mohanty "CUDA-enabled Hadoop cluster for Sparse Matrix Vector Multiplication" in IEEE 2nd International Conference on Recent Trends in Information Systems (ReTIS) 2015.
4. J. Zhu, Li Juanjuan, E. Hardesty, H. Jiang and L. Kuan-Ching, "GPU-in-Hadoop: Enabling Map Reduce across distributed heterogeneous platforms", IEEE/ACIS 13th International Conference of Computer and Information Science (ICIS), pp.321-326, 2014.
5. Sufeng Niu, Guangyu Yang, Nilim Sarma, PengfeiXuan, Melissa C. Smith, PardipSrimani, Feng Luo, "Combining Hadoop and GPU to preprocess large Affymetrix microarray data" in IEEE International Conference on Big Data (Big Data) 2014.
6. Mayank Tiwary, Abhaya Kumar Sahoo, RachitaMisra "Efficient implementation of apriori algorithm on HDFS using GPU" in International Conference on High Performance Computing and Applications (ICHPCA) 2014.
7. MengjunXie, Kyoung-Don Kang, Can Basaran" Moim: A Multi-GPU Map Reduce Framework" in IEEE 16th International Conference on Computational Science and Engineering 2013.
8. Malakar, R.; Vydyanathan, N., "ACUD Aenabled Hadoop cluster for fast distributed image processing, Parallel Computing Technologies (PARCOMPTECH), 2013 National Conference on , vol.1, pp.21-23, Feb. 2013.
9. ZhaiYanlong, Guo Ying, Chen Qiurui, Yang Kai and E. Mbarushimana, "Design and Optimization of a Big Data Computing Framework Based on CPU/GPU Cluster", pp.1039-1046, 2013.
10. M. Xin and H. Li, "An implementation of gpu accelerated mapreduce: Using hadoop with opencl for data- and compute-intensive jobs", International Joint Conference on Service Sciences, pp.6-11.
11. Wenbin Fang, Bingsheng He, Qiong Luo, Naga K Govindaraju "Mars: Accelerating MapReduce with Graphics Processors" in IEEE Transactions on Parallel and Distributed Systems.
12. Koichi Shirahata, Hitoshi Sato, and Satoshi Matsuoka." Hybrid Map Task Scheduling for GPU-Based Heterogeneous Clusters" In Proceedings of CloudCom, pp.733-740, 2010.
13. Max Grossman, Mauricio Breternitz, VivekSarkar "HadoopCL: MapReduce on Distributed Heterogeneous Platforms through Seamless Integration of Hadoop and OpenCL" in IEEE 27th International Symposium on Parallel & Distributed Processing Workshops and PhD Forum 2013.