INCOPRATING CUDA IN HADOOP IMAGE PROCESSSING INTERFACE FOR DISTRIBUTED IMAGE PROCESSING

Helly Patel[1], Mr. Krunal Panchal[2]

[1] PG Student, Computer Engineering, LJIET, Ahmedabad, Gujarat, India
[2] Assistant Professor, Computer Engineering, LJIET, Ahmedabad, Gujarat, India

ABSTRACT

This paper presents parallel processing based on integrated approach of Hadoop and CUDA for large scale image processing. It makes the use of high reliability, scalability and fault tolerance capability of Hadoop system and high computing power of CUDA for processing huge amount of images in highly efficient manner.so the main aim is to improve performance of image processing task by using features of both, Hadoop and CUDA and to overcome the problem that are occur while processing large no of images in customary sequential manner. The proposed model serves as a good candidate solution for both type of applications i.e. data intensive application and compute intensive application. As Hadoop performs well for data intensive application through the use of HDFS and CUDA serves best in case of compute intensive application, integration of both the framework provides faster execution for image processing task. Image storage is provided through Hadoop Distributed File System and Map and Reduce primitive of Hadoop Mapreduce will be performed using CUDA on GPU.

Keyword : - *Hadoop, CUDA, GPU, GPGPU,* Distributed system, Parallel system

---

1. INTRODUCTION

Multimedia database generated in today's world is Real time database and it keeps on increasing due to social media, computer graphics and vision, satellite database, video surveillance, medical Image database . So it is challenging task to process and analyze such vast amount of database. Technology, algorithm and framework developed up till now are both resource and compute intensive and its execution with larger number of images take much more time for in case of single node implementation. Parallel and Distributed systems are having potential to do such kind of task with more scalability, reliability and with faster execution. Which makes it emerging technology for the image processing domain. Apache Hadoop is widely used in distributed processing as it manage data partitioning, data distribution, fault tolerance, result aggregation, etc. which in turn provide programming simplicity to the developers. CUDA provide efficient medium for parallel execution on Graphics Processing units.

2. RELATED WORK

Lots of research has been done on in the field of image processing and still going on to efficiently process images using parallel and distributed architecture. Fang et. al. developed Mars, a MapReduce runtime system accelerated with graphics processing units which is executed only on single node of Hadoop [1]. Gongrong Zhang et. al. suggested parallel processing model based on Hadoop platform for large-scale images processing [2]. Ramani Duraiswami et. Al. had performed canny edge detection on NVIDIA CUDA performance is tested[3]. Jie Zhu had integrated Hadoop with GPU for word count application [4]. Peter Bajcsy et. al. had executed image processing on Hadoop to lower the barrier of executing spatial image computations in a computer cluster/cloud environment instead of in a desktop/laptop computing environment[5].

3. BACKGROUND

Apache Hadoop is open source, scalable framework for processing data in distributed manner. Hadoop is having its own storage system i.e. HDFS (Hadoop Distributed File System) and its own computing system i.e. Mapreduce. HDFS provide distributed storage with scalability, reliability, automatic data distribution, and aggregation and fault tolerance. Map reduce work with two primitives i.e. Map and Reduce. The Map function takes an input key/value pair (k_1,v_1)and outputs a list of intermediate key/value pairs (k_2, v_2) The Reduce function takes all values associated with the same key and produces a list of key/value pairs.

GPU (Graphics Processing Unit) are special processors that perform graphical tasks in a massively parallel manner and thus supplied high processing power. It is most powerful and inexpensive computational hardware which is widely used in the field of Image processing. Massively-parallel threaded GPUs provide more efficiency and speed up. A GPU card include number of cores which can execute multiple tasks in parallel. They are known as a massively parallel processors which are 10 times more rapidly computation and 10x greater memory bandwidth than CPUs. At present, they are used as co-processors for the CPU. The programming languages include NVIDIA CUDA, OPENMP, etc. Programmers write two kinds of code while performing GPU programming, the kernel and the host code [13]. The kernel code is executed in parallel on the GPU. The host code running on the CPU controls the data transfer between the GPU and the main memory, and starts kernels on the GPU.

CUDA (Compute Unified Device Architecture) is a programming model which is used for leveraging the high compute-intensive processing power of the Graphical Processing Unit (GPUs) to perform general, non-graphical tasks in a massively parallel manner [10]. It is a C-based programming model suggested by NVIDIA for leveraging the parallel computing capabilities of the GPU for general purpose computations [10]. CUDA allows software developers to use CUDA-enabled GPU for general purpose processing – an approach known as GPGPU .In the CUDA context, the GPU is known as a device, whereas the CPU is known as host. A kernel includes set of computations that is offloaded by the CPU to be executed on the GPU. A CUDA kernel is used to perform execution on the GPU by a grid of thread blocks, each consisting of a set of threads. Compute-intensive data-parallel part of applications is allowed to be executed as a kernel on GPU by CUDA as kernels, to the GPU [10].

HIPI(Hadoop image Processing Interface) is an image processing library designed to be used with the Apache Hadoop Mapreduce parallel programming framework and provide support for processing Images at larger extent [11]. HIPI removes the highly technical details of Hadoop's system and give users with the familiar sense of an image based library by allowing users to access to the resources of a distributed system [11]. The goal is to providing a platform specific to all image and graphics based applications. It is able to perform even with repetitive modifications and enhancements within Hadoop [11]. HIPI abstract functionality of Hadoop into an image-centric system and providing an efficient tool to researchers.

4. IMAGE PROCESSING FOR CANNY EDGE DETECTION

Identification and extraction of edges from images is considered as an Edge detection function in image processing. It is applicable in fields such as object recognition, image segmentation, data compression, land-water border etc. Edges in an image are signified by a significant image intensity change which represents important object features and boundaries between objects in an image. This multistep algorithm is considered as a standard and optimal detector among all edge detector algorithm.

Canny's algorithm consists of five major steps:

I. Image smoothing
II. Gradient computation
III. Edge direction computation
IV. Nonmaximum suppression
V. Hysteresis.

5. EXPRIMENTATION

In this work, we implemented CPU and GPU based canny edge detection algorithm using HIPI and CUDA framework respectively. Another is integrated framework of CUDA with HIPI execution. Canny Edge detection algorithm is performed to evaluate execution speedup. Performance comparison is done for different for different image dimension.

Table -1: Software configuration of node

CUDA	7.5
No of node in Hadoop	2
Components	*Configurations and Releases*
OS	Ubuntu 15.04 LTS
JDK	1.7.0_79
Hadoop	2.4.0

Table -2: GPU key parameters

CUDA/GPU Specification	***Value / Description***
Name	GeForce GTX 750 TI
Number of Streaming Processors (SMs)	640 CUDA core
Core speed	1020 MHz
Memory	2 GB of GDDR5
Memory clock	5.4 GBPS
Standard Memory	2048 MB

6. RESULT ANALYSIS

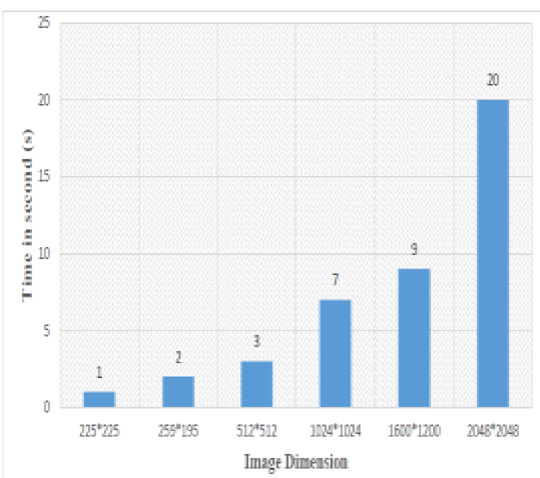

Chart -1: Performance comparison of HIPI program executed on CPU

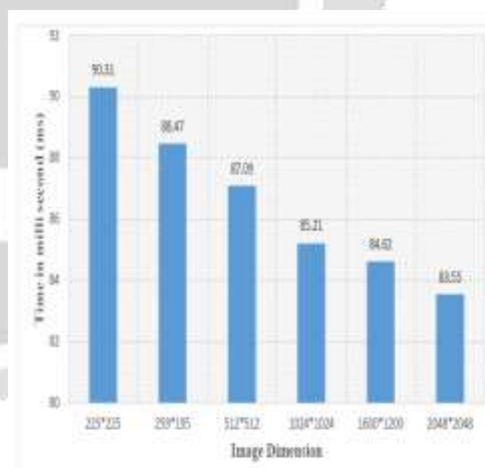

Chart -2: Performance comparison of CUDA program executed on CPU

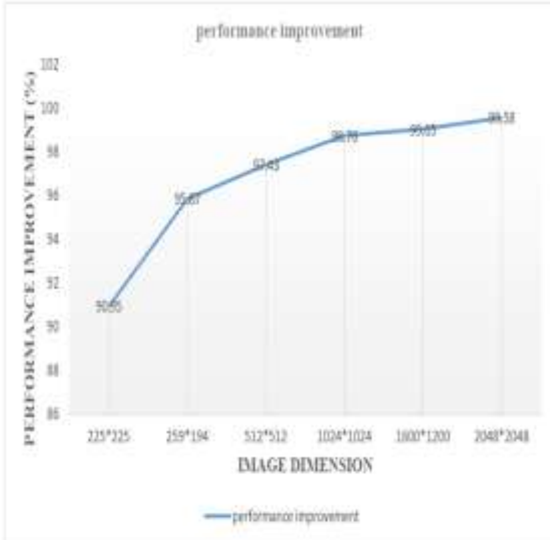

Chart -3: Performance improvement in execution time(%) in GPU(CUDA) compare to CPU(HIPI) execution

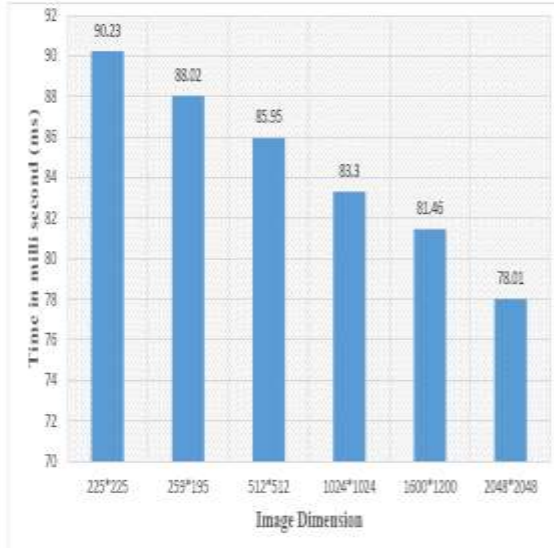

Chart -4: Performance comparison of integrated program executed in GPU

Chart -4: Performance Improvement in execution Time (%) in GPU (CUDA) compare to Integrated program execution

Canny edge detection algorithm is performed on different image size. As shown in Chart 1 execution time for HIPI program increase with increase in image size. Chart-2 shows that for CUDA program execution time decrease with increase in image dimension. Chart -3 shows Performance Improvement in execution Time (%) in GPU (CUDA) compare to CPU (HIPI) execution. Chart -4 shows performance comparison of integrated program which is less than standalone CUDA and HIPI program. Chart -5 shows Performance Improvement in execution Time (%) in GPU (CUDA) compare to integrated program execution. Canny Edge Detection algorithm performance is similar to

CUDA execution i.e. percentage improvement in execution time is around 1 to two percent. For larger size Images of dimension 1600*1200and 2048*2048 , performance is improved up to 3.73 to 6.3 percentage which shows that for large scale image processing Integrated system is having very good potential to execute task in parallel and Distributed system.

Table -1: Execution time on different platform

Image Dimension	**Execution time**		
	HIPI (second)	**CUDA (millisecond)**	**Integrated Program (millisecond)**
225*225	1	90.31	90.23
259*194	2	88.47	88.02
512*512	3	87.09	85.95
1024*1024	7	85.21	83.30
1600*1200	9	84.62	81.46
2048*2048	20	83.55	78.01

7. CONCLUSION

By using GPU based parallel processing mechanism the computing power of GPU and CPU is fully utilized. CUDA accelerated Hadoop image processing interface is having a lot of potential as a platform for processing computationally intensive image database with faster speed in distributed environment. The entire image detection algorithm performed faster for every size input image. For all image sizes, the performance increases gradually. For the portions of the algorithm performed entirely with the GPU (image smoothing, gradient computation, edge direction computation, and edge classification), the improvement was much larger. The smallest input image was processed 95.4 percent faster by the GPU and the larger input images were processed between 99.3 and 99.7 percent faster. Proposed system is suitable for both resource intensive and compute intensive applications. Framework exhibit higher scalability, reliability, and performance.

8. REFERENCES

[1] Fang, Wenbin, Bingsheng He, Qiong Luo, and Naga K. Govindaraju. "Mars: Accelerating mapreduce with graphics processors." IEEE Transactions on Parallel and Distributed Systems 22, no. 4 (2011): 608-620.

[2] Zhang, Gongrong, Qingxiang Wu, Zhiqiang Zhuo, Xiaowei Wang, and Xiaojin Lin. "A Large-scale Images Processing Model Based on Hadoop Platform." In Proceedings of the Second International Conference on Innovative Computing and Cloud Computing, p. 51. ACM, 2013.

[3] Luo, Yuancheng, and Ramani Duraiswami. "Canny edge detection on NVIDIA CUDA." In Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on, pp. 1-8. IEEE, 2008.

[4] Zhu, Jie, Juanjuan Li, Erikson Hardesty, Hai Jiang, and Kuan-Ching Li. "GPU-in-Hadoop: Enabling MapReduce across distributed heterogeneous platforms." In Computer and Information Science (ICIS), 2014 IEEE/ACIS 13th International Conference on, pp. 321-326. IEEE, 2014.

[5] Bajcsy, Peter, Phuong Nguyen, Antoine Vandecreme, and Mary Brady. "Spatial computations over terabyte-sized images on hadoop platforms." In Big Data (Big Data), 2014 IEEE International Conference on, pp. 816-824. IEEE, 2014.

[6] Bandre, Sanraj, and Jyoti Nandimath. "A Network Intrusion Detection System Framework based on Hadoop and GPGPU."

[7] Lourenço, Luis HA, Daniel Weingaertner, and Eduardo Todt. "Efficient implementation of canny edge detection filter for ITK using CUDA." In Computer Systems (WSCAD-SSC), 2012 13th Symposium on, pp. 33-40. IEEE, 2012.

[8] Sarade Shrikant D.,Disale Swapnil P., "LARGE SCALE SATELLITE IMAGE PROCESSING USING HADOOP DISTRIBUTED SYSTEM ", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 3 Issue 3, March 2014

[9] Sozykin, Andrey, and Timofei Epanchintsev. "MIPr-a framework for distributed image processing using Hadoop." In Application of Information and Communication Technologies (AICT), 2015 9th International Conference on, pp. 35-39. IEEE, 2015.

[10] Yang, Zhiyi, Yating Zhu, and Yong Pu. "Parallel image processing based on CUDA." In Computer Science and Software Engineering, 2008 International Conference on, vol. 3, pp. 198-201. IEEE, 2008.

[11] Vemula, Sridhar, and Christopher Crick, "Hadoop Image Processing Framework."