ISSN 2395-1621

Low Power Consumption for Increasing Speed of Map-Reduce on CPU-GPU Architecture

#1Mr.Saurabh Pawar, #2Mr. Akshay More, #3Mr.Sanket Shirse, #4Mr.Ravi Kumar

[1]saurabh.pawar47@gmail.com
[2]akshaymore64@gmail.com

JSPM's Imperial Collage Of Engineering And Research,
Pune, Maharashtra India

ABSTRACT

In this paper, by using the hadoop map-reduce framework represent a promising advance in adding cost and energy efficient compute parallel multi-node environment. Map-reduce introduces additional costs in disk IO and data transfer, caused by streaming intermediate outputs to disk. By using the k-mean clustering algorithm and job scheduling algorithm we are going to experiment that shows the integrated GPU in the hadoop Map-reduce framework environment. The priority based algorithm, in this algorithm CPU chooses the process of higher priority. It help us to increase power efficiency or work should be done in minimum power consumption.

Keywords: **Hadoop, MapReduce, K-means algorithm, Priority based scheduling algorithm, GPU.**

ARTICLE INFO

Article History

Received: 31st May 2016

Received in revised form :

31st May 2016

Accepted: 3rd June 2016

Published online :

3rd June 2016

I. INTRODUCTION

Now a days use of high speed internet grow day by days. Very large amount of data are generated every days. It stores data logs and various type of information are store in different files on different computers and different format. It may be in petabytes or hexabytes per month. This huge amount of data mainly known as Big data.

To handle such kind of big data the general system is used known Apache Hadoop. Apache Hadoop is union of HDFS and Map-Reduce compute model. These can help to easily store and process huge amount of data. Hadoop consist of Name node and Data node. Name node work as master it's main objective to manage resources and assign Map or Reduce tasks to the child node known as Data node. Both Map and Reduce task are perform separately on Data nodes by performing reading operation on input and generate desired output when processing is done. As Map and Reduce work separately it takes much more time and power.

GPU having efficient computing speed as well as it consume minimum power with this greatest feature of GPU we accelerate Map-Reduce work load. Due to integrated and high performance per watt nature of current integrated GPU, this gives much better performance as compare to the existence of CPU with minimum power consumption our work is to by using the k-means clustering algorithm and job scheduling algorithm now we experiment to show that the integrated GPU in the hadoop framework culture. The priority dependent algorithm, in this algorithm CPU selects the process higher priority.

II. PROPOSED SYSTEM

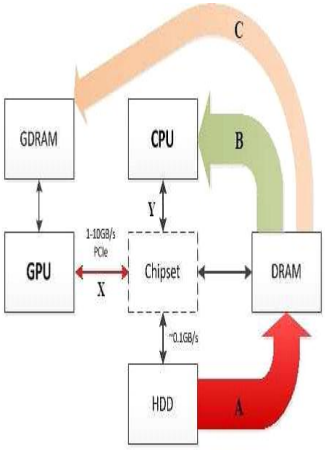


Fig 1. System architecture

Description about Components:

1. GPU - Graphics Processing Unit

- It increases the transistor density.
- Reduction power consumption.
- Enable to supply power for all logic chip calling.
- GPU enable to access the virtual address.
- Its not able to access the physical memory.

2. CPU-Central Processing Unit

- CPU only read physical memory.
- CPU not able to access virtual memory.

3. DRAM - Dynamic Random Access Machine

- It have input/output tape.
- Each processor have its local memory.
- Each processor can access shared memory.
- Extended version of Ram.
- DRAM is a type of random access memory that stores each bit of data in a septate capacitor within an integrated circuit.

4. HDD - Hard Device Drive

- Hard Disk is data storage device use for storing and retrieving digital information.
- It use one or more rigid rapidly rotating disk with magnetic material.
- Data is accessed random access memory.
- HDD retain store data even power o_.

5. GDRAM- Graphics Dynamic Random Access Machine

- Graphics Dynamic Random Access Machine is type of dynamic random access memory that store each DRAM (MDRAL, Synchronous graphics RAM (SGRAN).

6. Chipset

- Chipset is a set of electronic component in an integrated circuit that man-age data flow between processor, memory and peripherals.
- It is usually found on motherboard.
- Chip set are design to work with special family of microprocessor.
- Because it controls between the processor and external device.

Description about Architecture

Data processing implies data exchange between HDD, DRAM, CPU and GPU. Figure shows how data is transferred when a commodity machine performs computations with a CPU and a GPU. Data exchange between components of a commodity computer when executing a task

1. Arrow A: Transferring data from an HDD to DRAM (a common initial step for both CPU and GPU computing)

2. Arrow B: Processing data with a CPU (transferring data: DRAM chipset CPU)

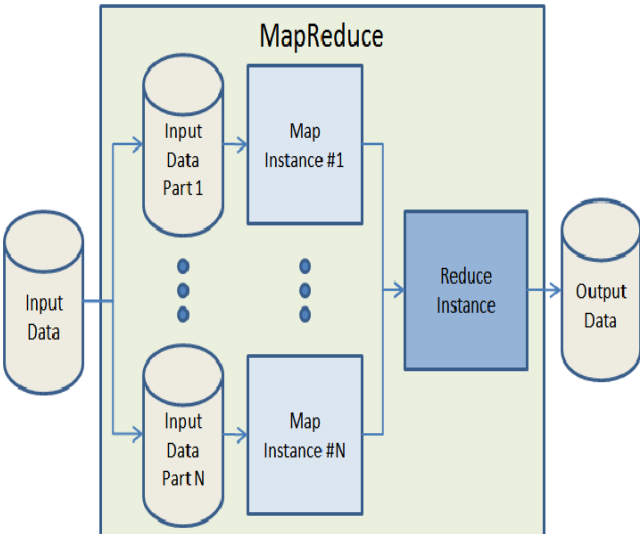
3. Arrow C: Processing data with a GPU (transferring data: DRAM chipset CPU chipset GPU GDRAM GPU)

III. IMPLEMENTATION

MapReduce :

MapReduce is one of the most important programming model which produce and process the big data set with a parallel, distributed algorithm on a cluster.

A MapReduce program is union of a Map() which conduct filtering and sorting and a reduce that conduct a summery operation. The MapReduce system set a order in such a way that the processing by collecting or assembling distributed server, running the multiple tasks in parallel , handle all communication and data transfer between different part of system, and provide for redundancy and fault tolerance. MapReduce may be take advantage of locality of data processing it on or near the storage assets in such a way that to reduce the distance over which it must be transmitted.



K-Means Algorithm:

K-Means is one of the simplest unsupervised learning method among all partitioning based clustering methods .It classifies given set of data object in k cluster , where k is the number of desired cluster and it is required in advance. A centroid is defined for each cluster. ALL the data object are placed in a cluster having centroid nearest to that data object. After processing all data object, k-means, or centroid, are recalculated and the entire process is repeated. ALL data object are bound to the cluster based on the new centroid. In each iteration centroid change their location step by step. This process is continue until no any centroid move.

As a result, k-cluster are found representing a set of n data object. .

ALGORITHM:

Step:

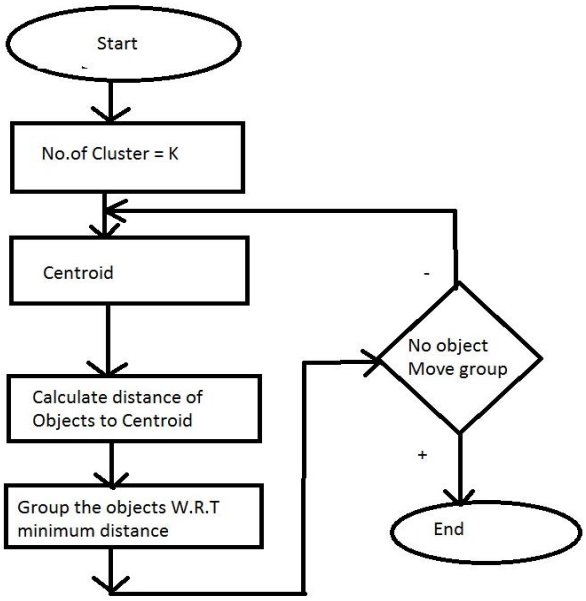
(i) Arbitrarily choose "K" object as the initial cluster center.

(ii) Repeat,

(a) (Re)assign each object to the cluster to which the object is the most similar, based on given similarity function;

(b) Update the centroid i.e. calculate the mean value of the object for each cluster;

(iii) Until no change.



Flowchart: K-means Algoritham

Algorithm Priority Based Scheduling :

One of the main reason behind the priority i.e, each and every process assign a priority and priority allow to run. Priority process can be schedule first come first serve (FCFS) unique order. The another scheduling algorithm is Shortest-Job-First (SJF) algorithm is one of the special case of general priority scheduling algorithm. In SJF algorithm is the simple priority algorithm in which the priority is the inverse of CPU brust. That means it represent longer the CPU brust the priority is lower and vice-versa. The priority can be defined into two different modes, modes are also represent as state

1. Internally
2. Externally

1) Internally :

Internally define unique properties use some measurable quantities to compute priority of process. These are some common example of internal priorities :

- Time limit.
- Memory Requirement.
- File. For example, number of open files requirement

2) Externally :

Externally define unique properties are set by certain criteria that are external to operating system such as

- The important of process.
- Type or amount of funds being paid for computer use.
- The department sponsoring the work.
- Politic

Priority scheduling can be either non-preemptive or preemptive

- A preemptive priority algorithm will be preemptive the CPU if the priority of new arrival process is maximum than priority of currently running process .
- Non priority algorithm will simply put the new process at the head of ready queue.

IV. RESULT

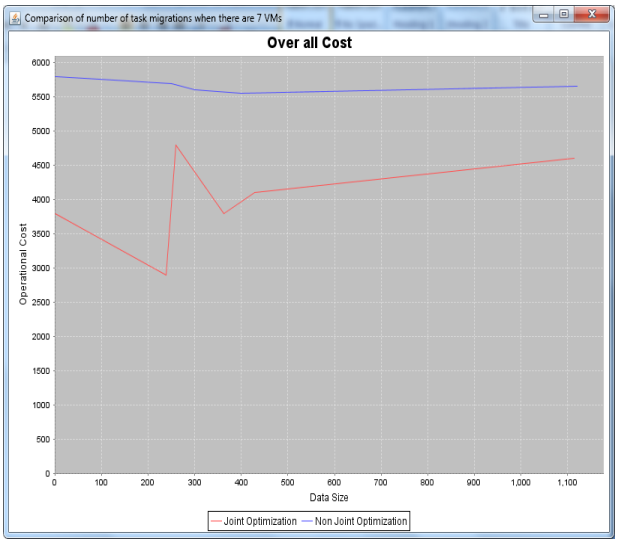


Fig2: Overall cost of executing job on GPU

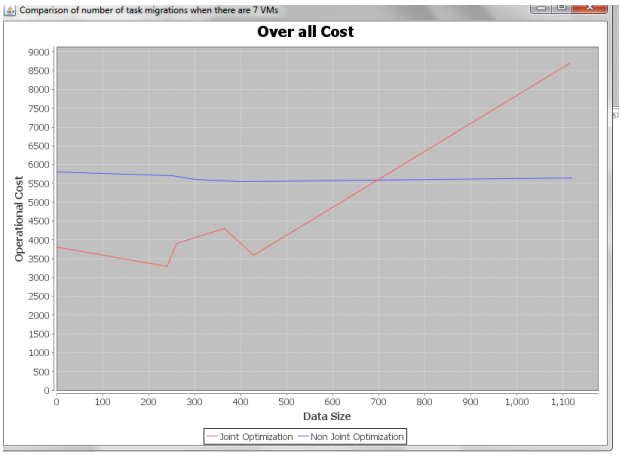


Fig3: Overall cost of executing job on CPU

As the main aim of this system to increase power efficiency and decrease the processing time it result in decreasing the operational cost. Fig2 shows the overall cost of executing job on GPU as we compare it with the fig3 which shows the overall cost of executing job on CPU the operational cost of GPU is less than the CPU. At the point where data size is 500 and above in the CPU system Operational cost goes higher than the GPU system operational cost at the point where data size is 500 and above.
Fig2 and Fig3 shows this comparison by using that we can easily evaluate it.

V. CONCLUSION

Here we mainly decrease the processing time and increase power efficiency of system as compare to existing system. In digital world the system is depend on two unique factor. So we work on these two factor with Priority based scheduling algorithm to make system have better efficiency and short processing time with respect to minimum power consumption.

REFERENCE

[1]M. Xin and H. Li, “An implementation of gpu accelerated mapreduce: Using hadoop with opencl fordata- and compute-intensive jobs,” in International Joint Conference on Service Sciences, pp. 6–11, May 2012.

[2] S. Kim, I. Roy, and V. Talwar, “Evaluating integrated graphics processors for data center workloads,” in Proceedings of the Workshop on Power-Aware Computing and Systems, ser. HotPower ’13. New York, NY, USA: ACM, pp. 8:1–8:5, 2013

[3] https://mahout.apache.org/, accessed: Nov. 21, 2014.

[4] M. Hammoud and M. F. Sakr, “Locality-aware reduce task scheduling for mapreduce,” in Proceedings of the IEEE Third International Conference on Cloud Computing Technology and Science, ser. CLOUDCOM ’11. Washington, DC, USA: IEEE Computer Society, pp. 570–576, 2011.

[5] L. Chen, X. Huo, and G. Agrawal, “Accelerating mapreduce on a coupled cpu-gpu architecture,” in Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, ser. SC ’12. Los Alamitos, CA, USA: IEEE Computer Society Press, pp. 25:1– 25:11, 2012.

[6] Mr.Saurabh Pawar, Mr. Akshay More, Mr.Sanket Shirse, Mr.Ravi Kumar “Minimum Power Consumption For Accelerating Map-Reduce On CPU-GPU Architecture” in (IJAFRSE) Volume 2, Issue 4 , September 2015. Science Central Value: 26.54

