

Distinction of Discrete Transformations Applied to Hadoop's MapReduce

Bhavini Kanoongo
Computer Dept., Mumbai
University, D.J.Sanghvi COE
Vile Parle (W) Mumbai, India

Puja Jagani
Computer Dept., Mumbai
University, D.J.Sanghvi COE
Vile Parle (W) Mumbai, India

Chetashri Bhadane
Computer Dept., Mumbai
University, D.J.Sanghvi COE
Vile Parle (W) Mumbai, India

ABSTRACT

Hadoop MapReduce is an effective data processing platform for both commercial as well as academic applications. It intends the simplification of vast quantities of data as well as ease of processing in parallel on enormous clusters of hardware in a fault-tolerant and dependable approach. There are many modifications possible in the MapReduce to increase the performance along with increasing the simplicity of job tuning. Three of the adaptive run-time techniques namely, HIPI (Hadoop Image Processing Interface), HOG (Distributed Hadoop MapReduce on Grid) and using SAM (Situation Aware Mappers) are described and compared in the following paper. There is a rapid increase in the amount of images uploaded on the internet; however the applications which utilize this data are severely inadequate. Large and computational distributed processing can be done by employing HIPI.

Another Hadoop transformation that we study is the HOG which provides a complimentary, adaptable and dynamic MapReduce environment on the resources of the grid, reforms Hadoop's fault tolerance for wide area data analysis. All the modifications to the Hadoop framework are transparent to the existing Hadoop MapReduce applications.

General Terms

Data Mining, optimization techniques.

Keywords

Hadoop, MapReduce, Grid, Parallel data systems, Distributed data systems.

1. INTRODUCTION

MapReduce is not only a programming model, but also a related employment for generation and processing of huge data sets. The map function is specified by the user to process a key pair so that a set of intermediary key pairs can be generated and the reduce function is used to coalesce intermediate values correlated with the identical intermediate key. Apart from being highly scalable, MapReduce also runs on enormous cluster of product machines. A quintessential computation process of MapReduce involves heaps of terabytes of data being processed on tons of machines. Profitable and extensible execution of resource-exhaustive applications is possible due to the ability of the MapReduce to parallelize tasks and hence it supplies a platform for such applications.

The vision applications which make use of the framework of Hadoops MapReduce need an overwhelming learning curve along with staggering complexity. The advancement of researchers is severely crippled by the large overhead of such applications. The extreme technical specifications of Hadoops

system are removed by the HIPI. The users are only provided with the feel similar to that of an image library which can access the cutting-edge resources of distributed system. Driven by the overreaching goals of separating the functionality of Hadoop to an image-centric system, HIPI is basically a software model. Being an extendible system, HIPI provides the researchers a tool to efficiently utilize MapReduce for computer vision along with image processing.

Users generally need a Hadoop platform that runs on dedicated surroundings such as the cluster or cloud, to employ Hadoop MapReduce. HOG is a Hadoop platform based on the OSG, which supplies extensible and open services for users that want to use Hadoops Mapreduce. With not many changes, it can be transformed to other distributed grid systems. Basically Hadoop has two failure levels for fault tolerance and replication is employed to avoid loss of data. The two levels are the node level and the rack level. Node level is the first level and is responsible for maintain the data integrity of the cluster. Rack level secures the data even if all the nodes in a rack fail. A new level called the site failure level is introduced in this method. This extension of a third level provides data locality benefits.

The performance of MapReduce can be enhanced by making the runtime environment adaptive so as to avert creating the performance tuning tougher than present. The flexibility and adaptivity of MapReduce can be enhanced by crumbling a crucial supposition that the mappers are entirely isolated. An asynchronous interaction channel amongst the mappers is introduced, which uses a transactional and Distributed Meta-Data Store (DMDS). This empowered the mappers to inform meta-data regarding their state along with states of all mappers. These Situation Aware Mappers (SAMs) have a combined aspect of the job state and are eligible to take global integrated optimization arrangements. On the whole, the SAM tasks can modify the execution, during runtime, relying on the global state.

Section 2 discusses about the various modifications in MapReduce for various applications and they can be implemented. Section 3 deals with the review of the three transformations.

2. REVIEW OF LITERATURE

2.1 HIPI

The design of HIPI was made considering few goals in mind. The MapReduce framework should provide an extensible and open library for various applications of computer vision. It should allow simple filtering along with efficient storage of images. Furthermore, the complex details of the MapReduce framework should be hidden and the tasks must be parallelized as well as balanced.

A HIPI Image Bundle data type [7] is created to store huge amount of images in a single file so as to facilitate a more efficient performance. A HIPI Image Bundle consists of two files: a data file containing concatenated images and an index file containing information about the offsets of images in the data file. [7] This permits convenient access to images from anywhere in the bundle without the need to read each and every image. The main focus of the library is to bring similar data types to the user directly. The user's only job is to provide an input of HIPI Image Bundle. Making the tasks parallel and dispatching the float images, is taken care of by the HIPI. The InputFormat and RecordReaders which would have to be taken care of by the user, is included in the HIPI framework. This specification works on HIPI Image Bundles for various image types, sizes, and varying amounts of header and exif information.[7] The various permutations of images is handled at the backend. A culling stage is introduced to filter the images according to image properties before the start of the map tasks. The images that are distributed to map tasks are those which pass the culling stage.

This process is often very efficient because culling often occurs based on image header information, so it is not required to read the entire image. [7]

2.2 HOG

The infrastructure of HOG comprises of three components, namely grid submission and execution, Hadoop distributed file system (HDFS) and MapReduce framework.

At the beginning of HOG, Hadoop worker nodes can be requested to run on the grid by the user. The submission and removal of worker node jobs can shrink as well as grow. The Hadoop worker node includes both the datanode and the tasktracker processes. [6]

A simple shell wrapper script will initialize the Hadoop worker node environment. It initializes OSG operating environment, downloads Hadoop worker node executables, extracts worker node executables and set late binding configurations, start the Hadoop daemons and cleans up the working directory when daemons shut down. [6] The single point failures of the HOG system, the Job Tracker for MapReduce and Namenode for HDFS dwell on a stable central server. As soon as master server turns unavailable the MapReduce jobs stop their execution and there is also a lack of availability of HDFS file system. However, there is no loss of data. The slave servers notify to individual master server at the beginning of the grid jobs. One of the requirements of Hadoop is that the worker nodes should be attainable by each other. Some clusters in the Open Science Grid are designed to be behind one or more machines that provide Network Address Translation (NAT) access to the Internet. [6] NAT obstructs direct access because at the rear of a distant NAT, Hadoop can't communicate with nodes. Hence, in the OSG there is a restriction to the sites which have public IPs on their respective worker nodes.

2.3 SAMs

There are hundreds of adaptive optimization techniques that drastically enhance the performance of MapReduce framework and specially stability of performance. Three techniques we used are adaptive mapping, adaptive combining and adaptive partitioning and sampling. MapReduce processes splits which have one-to-one mapping of partitions as well as tasks of input data. There can be more map tasks in comparison with the map slots available, to ensure balance of workload. Unfortunately, increase in the number of mappers also leads to an increase in the starting overhead and task scheduling. The startup overhead may include running user code to perform job-specific setup tasks, such as loading reference data, which can become a significant portion of the running time. [5] However, as MapReduce checkpoints outcomes of each mapper, simultaneously having shorter splits reduces the assets from applying combiner. The dynamic "stitching" of a number of splits to an individual virtual split allocated to the mapper, changes the checkpoint interval since the mapper is running. The quantity of splits is decoupled from the quantity of mappers by the AM, to achieve the best of both worlds. The mappers at the later phase process the remaining data, if the mappers of the earlier phase determine to stop. Once the processing of all splits occurs, only then the AMs of the last wave halt.

Local accumulation of map outputs by the use of combiner functions is supported by MapReduce to decrease the quantity of data that is required to be shuffled as well as merged in reducers. Hadoop combiners not only need all the map outputs to be sorted and serialized, but also possibly written to disk. However, it is mentioned in database literature that the performance of hash-based aggregation is better than that of sort-based aggregation. The hash-based aggregation is leveraged for combining map outputs with frequent keys, while keeping the sort-based aggregation as a fallback alternative for non-frequent ones. [5] The advantages of shuffling and merging small amount of data in reducers is preserved by the AC, along with removing the overhead needed to apply combiners. The sort is replaced with hashing for often occurrences of map output keys. ACs could be adapted to work in this multi-threaded setup, using ideas that have been explored for hardware caching of hash tables on multi-core machines in [3]. [5] By default, the partitioning in Hadoop is done by hashing and a custom function can be used. Irrespective of being custom or not, statistical decision of the partitioning function is done even before the job begins. At times when a good partitioning banks on the input data, there is often a need for a separate sampling job. The AS technique dynamically takes the decision of when to end sampling, relying on a global sampling condition. The AS produces a global histogram of map output keys early in the map stage. [5] One of the important applications of the histogram is the dynamic decision making about the partitioning function simultaneously, when the job is running.

3. COMPARATIVE STUDY

Table 1. Comparison between HIPI, HOG and SAMs

Parameters for comparison	Hadoop Image Processing Interface for image based MapReduce tasks	Distributed Hadoop MapReduce On the Grid	MapReduce using Situation Aware Mappers
Basic Idea	It aims to create an interface for computer vision with MapReduce technology.	It aims to provide a free, elastic and dynamic MapReduce environment on the opportunistic resources of the grid.	The mappers communicate via a distributed meta-data store and also know the global state of the job.
Working of the Method	HIPI abstracts the technical specifications of the Hadoop's system and it can also implement the various trending techniques of computer vision.	A novel Hadoop platform, Hadoop on the Grid (HOG), based on the Open Science Grid (OSG) is created to provide scalable and free of charge services for users who plan to use Hadoop MapReduce.	Situation Aware Mappers (SAMs) are employed by a number of adaptive optimization techniques for MapReduce framework so as to cooperatively make global optimization decisions.
Main purpose of the Method	Present users with an intuitive interface for image-based operations and hide the details of the MapReduce framework.	Hadoop's fault tolerance is improved for wide area data analysis by mapping datacenters to virtual racks and creating multi-institution failure domains.	To achieve minimum task startup overhead and dynamic load balancing.
Architectural modifications	A HIPI Image Bundle data type is introduced which stores large amount of images on files for increasing the efficiency of MapReduce. This bundle comprises of a data file which consists of concatenated images and an index file which includes information regarding corresponding offset of images in the data file.	HOG has three main components namely, grid submission and execution component, Hadoop distributed File System (HDFS) and MapReduce Framework. The unique feature is the first component which is responsible for resource allocation and management.	They develop a set of techniques that make MapReduce more dynamic: (a) Adaptive Mappers dynamically take multiple data partitions (splits) to amortize map-per start-up costs. (b) Adaptive Combiners improve local aggregation by maintaining a cache of partial aggregates for the frequent keys. (c) Adaptive Sampling and Partitioning sample the mapper outputs and use the obtained statistics to produce balanced partitions for the reducers.
How does the method help users	HIPI serves as an excellent tool for researchers in computer vision because it allows development of large-scale computer vision applications to be more accessible than ever.	A new level called the site failure level is introduced apart from the node level and the rack level to improve Hadoop's fault tolerance system. This also brings data locality benefits.	The user does not have to make the decisions before each job starts as that is taken care of by the SAMs.
Various operations provided to the user	<ul style="list-style-type: none"> • Encoding and decoding of images to present the user with float images within the MapReduce pipeline. • Accessing of pixel values for image processing and vision operations. • Filtering of images. • Cropping operations. 	To run MapReduce, users need a Hadoop platform such as the cloud or cluster. HOG is an open platform created for the ease of access by the users. HOG is not only free but also scalable and hence beneficial to the users.	Users can implement system-level enhancements using the SAM mechanism. Also the user can control the optimization techniques.

Advantages over the existing techniques	Rather than involving large overhead for obtaining float image representation, this method uses a library that brings familiar image-based data types to the user directly.	HOG's rapid scalability can provide comparable performance to a dedicated Hadoop cluster. Traditional rack awareness is extended to site awareness to improve fault tolerance. Default replication factor was also increased from 3 to 10. Only underlying changes were made to MapReduce, no API changes.	It supports general intra-task adaptivity. The adaptive algorithms exhibit superior performance stability and do not depend on cardinality and cost estimation.
Disadvantages/Drawbacks	HIPI cannot be performed on extremely large image sets and the problem of OCR persists.	High node failure rate is the largest barrier faced by HOG addresses. Also abandoned data nodes and disk overflow are major drawbacks.	An extra serialization or deserialization step before the aggregate function is required, unless the partial aggregation function can be done on the serialized objects.

4. CONCLUSION

Google uses MapReduce model successfully for varied purposes. This was achievable due to various reasons. Not only is the model user-friendly, but also can be used by programmers who are not experienced in the field of parallel and distributed systems. This is possible because various operations such as fault-tolerance, parallelization, locality optimization and load-balancing are hidden from the user.

One of the applications of MapReduce discussed in this paper is the image processing framework called the Hadoop Image Processing Interface (HIPI). The design of the library is successful in hiding the composite particulars of MapReduce framework. It brings ahead the part which the users care most about, that is, images. This interface has implemented certain features so as to empower teachers along with researchers with an efficiency tool. The second application was that of Hadoop on Grid (HOG) whose infrastructure depends on Open Science Grid (OSG). However, the unreliability of grid causes the implementation of HOG challenging. Although difficult to develop, HOG is able to achieve similar performance as that of a dedicated cluster. The last application discussed in this paper was adaptive techniques that utilize Situation Aware Mappers (SAM). SAMs not only configure automatically but also accordingly take global optimization decisions. Many essential system-level improvements can be made by making use of these different applications.

5. REFERENCES

- [1] Jeffrey Dean and Sanjay Ghemawat. 2004. MapReduce: Simplified Data Processing on Large Clusters. Google inc.
- [2] R.Pordes, D.Petravick, B.Kramer, D.Olson, M.Livny, A.Roy, P.Avery, K.Blackburn, T.Wenaus, F.W'urthweinetal. 2007. The open science grid. Journal of Physics: Conference Series, vol.78.
- [3] J. Cieslewicz and K. A. Ross. 2007. Adaptive aggregation on chip multiprocessors. VLDB, 2007, pp. 339–350.
- [4] I. Raicu, I. Foster, and Y. Zhao. 2008. Many-task computing for grids and supercomputers. Many-Task Computing on Grids and Supercomputers, 2008. MTAGS 2008. Workshop on. IEEE, 2008, pp. 1–11.
- [5] Andrey Balmin, Kevin S. Beyer, Vuk Ercegovac. 2012. Adaptive MapReduce Using Situation Aware Mappers. IBM Research Report.
- [6] Chen He, Derek Weitzel, David Swanson and Ying Lu. HOG: Distributed Hadoop MapReduce on the Grid. University of Nebraska – Lincoln.
- [7] ChrisSweeney, LiuLiu and SeanArietta. HIPI: A Hadoop Image Processing Interface for Image-based MapReduce Tasks. University of Virginia.
- [8] Casey McTaggart. Hadoop/MapReduce. Object-oriented framework presentation.