Affordable Cluster-Based Context for Multimedia Big Data Extraction

Pampati Nagaraju, Etyala Ajith Kumar

Abstract: Most of the data being created than ever before and that can be textual, multimedia, spatial information. To process this data, several data processing platforms have been developed including Hadoop, based on the Map Reduce model and HPCC systems. The HPCC System analysis provides a framework for multimedia data processing. Moreover, Multimedia data encompasses a wide variety of data which is not limited to image data, video data, audio data and even textual data, while developing a unified framework for such wide variety of data to consider computational difficulty in it. Preliminary results show that HPCC can potentially reduce the computational complexity significantly.

Keywords: Hadoop Big Data Analysis, HPCC framework, HDFS, Feature Extraction, Multimedia Big Data

I. INTRODUCTION

Big data analytics may be applied on huge amount of datasets that include differing types like structured / unstructured and streaming/batch, and different sizes from terabytes to petabytes. Big data may be a term applied to data sets whose size or type is beyond the power of traditional relational databases to load, manage, and process the info with low-latency. Generally Big data is defined using the 3V (volume, variety and velocity). To process these several big data processing platforms were developed with Hadoop based on the MapReduce model and HPCC systems.

A. Motivation

In the emerging era of technology filled with multimedia data, which makes up about two thirds of internet traffic, it is important to have systems capable of handling this data. Most of the research in the Big Data field has been concentrated to text-based analytics.

The purpose of this study is that the optimized integration of image processing algorithms into High Performance Computing Cluster (HPCC) Systems and to design a data pipeline capable of efficient pre-processing and processing of multimedia data. HPCC is an open sourced massively parallel processing computing platform used for solving Big Data problems.

This paper aims to compare performance and architectures of HPCC with Hadoop platform while running image processing tasks. The Apache Hadoop platform is widely used for Big-Data analytics and is also being used for multimedia applications. Selecting the right platform for the problem at hand is critical to achieve all possible performance efficiencies. The objective is compare HPCC and Hadoop by considering the system architecture, programming model, and benchmarking in evaluating the two platforms for processing Multimedia Big Data. Feature extraction and multimedia content retrieval were used for benchmarking the platforms.

B. Key Contributions

Multimedia big data processing differs from other kinds of big data tasks like textor log file processing as multimedia content is very diverse and resources used for task completion vary for each individual image or video. With this in mind, it becomes important to have efficient usage of resources with load balancing techniques.

The important contributions to this paper include:

- Identify challenges in optimizing multimedia big data frameworks
- Performance evaluation of framework with respect to multimedia data
- Complexity reduction with optimization.

II. BACKGROUND AND RELATED WORK

Multimedia data has been growing at an amazing speed and accounts for more than 70% of unstructured. It can be considered as “big data” not just on the basis of its huge volume, but also because it can be analyzed to gain insight and information in a wide range of applications. This data can be extracted to gain useful knowledge and understand the semantics by analysis.

A. Hadoop Ecosystem

The Hadoop ecosystem has evolved from the Apache Hadoop implementation of the MapReduce paradigm. A MapReduce job may be a unit of labor that consists of the computer file, the associated Map and Reduce programs, and user-specified configuration information. MapReduce may be a programming model for processing and generating large data sets. Hadoop provides the Hadoop Distributed classification system (HDFS) for storing data. HDFS may be a classification system designed for storing large files with provisions for streaming data, access patterns and run on commodity hardware.

Revised Manuscript Received on March 18, 2020.

Pampati Nagaraju, Department of Computer Science and Engineering, Balaji Institute of Technology & Science, Warangal, Telangana, India· Email- nagaraju.pampati@gmail.com

Etyala Ajith Kumar, Department of Computer Science and Engineering, Talla Padmavathi College of Engineering, Kazipet, Warangal, Telangana, India.Email- ajithkumaretyala@gmail.com

B. HPCC System

HPCC is the solution to data intensive cloud computing. To meet all requirements that such a platform would have required the design and implementation off two distinct cluster processing environments, each of which could be optimized independently for its parallel data processing purpose.

The first of these platforms is called a Data Refinery or THOR. The Thor system cluster is implemented employing a master/slave approach with one master process and multiple slave processes which give a parallel job execution environment. Thor employs a record-oriented based storage system. These records can be of fixed or variable length, and support a variety of formats like XML, CSV etc. Records can also contain nested child datasets. Record I/O is buffered in large blocks to scale back latency and improve data I/O. Files to be loaded to a Thor cluster are typically first transferred to a landing zone from some external location, then a process called spraying is employed to separate the file and cargo it to the nodes of a Thor cluster. The initial spraying process divides the file on user-specified record boundaries and distributes the data as evenly as possible with records in sequential order across the available processes in the cluster.

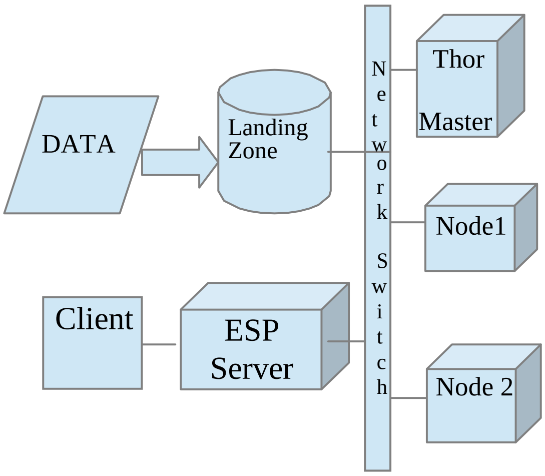

Figure 2.1 Clusters Based Hpcc Thor

These characteristics make Thor ideal for processing huge volumes of data for tasks like ETL processing, record linking and entity resolution, large-scale ad-hoc complex analytics, and to support high-performance queries and data warehouse applications. Figure 2.1 shows an HPCC System multi-cluster setup. It shows a THOR processing cluster which is similar to Google and Hadoop MapReduce platforms with respect to its function, file system, execution, and capabilities but offers higher performance.

The Roxie system could also be an internet base platform for data warehouse to deliver the multiprocessing requirements of online applications through Web service interfaces by supporting thousands of simultaneous queries and users. Enterprise Control Language (ECL) is the language used in HPCC Systems. It has been designed to be Data-flow oriented, declarative, and non-procedural, parallel language for data- intensive computing. The compiler generates C++ code that's highly optimized. A key feature of ECL and HPCC is control over distribution of knowledge across the computing cluster which helps avoid data skew and allows for execution of both local operations that are performed on data local to node and global operations performed across nodes. ECL is compiled into optimized C++ code for execution on the HPCC Systems platform, and can be used for complex data processing and analysis jobs on a Thor cluster. ECL allows inline C++ functions to be incorporated into ECL programs, and external programs in other languages are often incorporated and paralleled through a PIPE.

The use of HPCC Systems is proposed because it enables users to leverage a multi-cluster environment to speed up the running time of any computationally intensive algorithm. It is open source and easy to setup. It provides programming abstraction and parallel run time to hide complexities of fault tolerance and data parallelism. The HPCC system is the ideal system to run resource intensive image processing tasks on massive scale.

C. HIPI – Hadoop Image Processing Interface

HIPI is a library designed to be used with the Apache Hadoop MapReduce parallel programming framework. Figure 2.2 shows an affordable image processing with MapReduce style parallel programs typically executed on a cluster. HIPI also provides integration with OpenCV. The system manages storage using HIPI image bundles which stores many images in one large file.

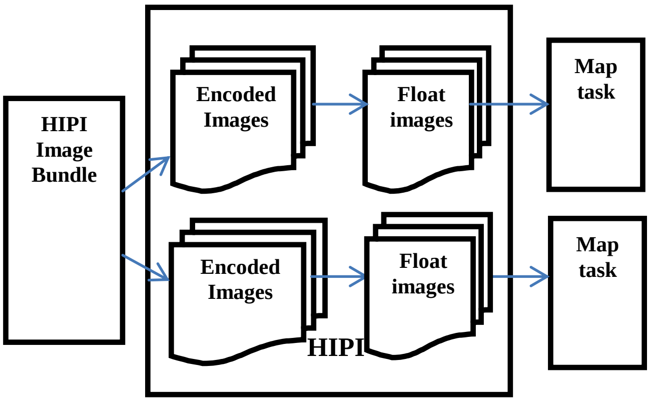

Figure 2.2 HIPI image processing

A HIPI image bundle consists of two files: a data file containing concatenated images and an index file which has information about images and their offsets. While executing jobs, these images in a bundle are distributed across map nodes.

D. Multiple Platforms for Big Data Analysis

The objective of this to realize a multiple big data processing platform with high performance, high availability and high scalability that will be compatible with any existing business intelligence and analysis tool.

The system monitors state of the cluster and dynamically adjusts the VMs resource allocation. A caching mechanism is more necessary in some environments which have many duplicate SQL commands. Every search query will require time and resources. These can be reduced while the cache hits with a high-speed In-memory and a large-capacity In-disk. In addition, the design of cache can reduce retrieval time.

III. PROBLEM DESCRIPTION

Multimedia data processing poses challenges different than those faced in processing structured, textual data. Since multimedia data is highly heterogeneous, there is a semantic gap between the data points that can be extracted for analysis and this poses a great challenge when confronted with multimedia data analysis tasks. The other major challenge when dealing with multimedia big data is intent expression. In multimedia data processing, the query intent generally can be represented only by text; however, the text can only express very limited query intent.

A. Multimedia Big Data in HPCC

The main problem of this thesis aims to address is to leverage HPCC systems to process large scale image databases. Distributed processing of large scale image datasets is still a relatively new concept and to the best of our knowledge there is no single system capable of optimized processing of multimedia in a distributed computing cluster.

B. Parallel Cluster System Framework

In figure 3.1, the first component of the framework is the data processing component where the input datasets are cleaned, preprocessed and processed so that data points can be extracted and analyzed to execute queries. Once the data is processed, the extracted data can be broadly categorized into metadata and actual image data. The metadata consists of information about geo-location, if data and user tags among other points of data about the images in the dataset. This metadata can be used to optimize queries and improve performance of framework.

To run queries on image data, images to be decoded and features of importance must be extracted. This is done in the feature extraction phase where multiple nodes of the THOR cluster parallel perform this task in a highly efficient and optimized manner. The feature extraction phase can be executed multiple times based on input queries and the features required. The application layer is where the user interacts with the system and can provide parameters for queries and visualize results to obtain meaningful information.

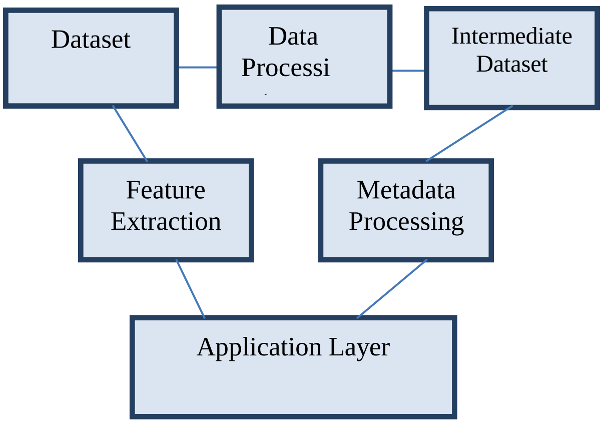

Figure 3.1 Cluster-based Framework

IV. PERFOMANCE EVALUATION SETUP

The framework consists of processing the metadata and reading images in the HPCC platform, decoding the JPEG image data and using the decoded JPEG data and metadata to execute queries that retrieve the images. For applicable queries, visualizations can be generated to make the huge amounts of data easier to understand. The implementation consists of the following steps:

- Initially, reads the metadata files and image datasets into the HPCC cluster.
- Decode image data to compute RGB values which can be used for extracting features.
- Retrieve a set of images based on metadata query parameters and extracted features.
- Finally, visualize the query results in a web-based application.

The first step of the implementation was to set up HPCC systems computational cluster with nodes. Each node has 20 Intel Xeon® CPU’s and 128 GBs of memory.

A. Dataset with Processing

The dataset contains a tab separated list of image and video records. Each record of the list has fields describing various attributes like Image / video identifier, Image/video hash, Date taken, Date uploaded, Capture device, Title, Description, User tags, Machine tags amongst others. Three expansion packs have recently been added to the dataset with in the Exif metadata pack, each line refers to an image/video and its set of Exif metadata item. Similarly, the Auto tag and location metadata pack describes presence and location of a variety of concepts, such as people, animals, objects, food, events, architecture, scenery etc. using deep learning principles for each image/video where it was taken in a human readable format instead of geographic coordinates.

Due to resource constrains, the YFCC100m was not used for image processing tasks. Instead the Yahoo! Flickr Creative Common Images tagged with ten concepts which consist of 200,000 images and categorized into groups.

There are different options when spraying files onto the HPCC systems cluster. The delimited option is the perfect fit for the metadata files onto the cluster as logical files; the next step is to read them using ECL code.

Integration of HPCC with R uses image processing packages and running shell scripts in a parallelized manner with ECL commands. A simple open source implementation of the JPEG decoder was chosen and modified to fulfill our requirements. The records of image Data DATASETs are generated, and passed as a parameter to the JPEG decoder. It decodes the image data and returns average R, G & B values for each image which are added as columns to the dataset. These additional data points can be used for feature extraction tasks.

B. Query Processing

In multimedia content retrieval requires processing of semantic as well as textual information. In this framework, the metadata processing component of system executes ROXIE queries to process the textual information. Semantic data processing is handled by the Data Analysis component where THOR queries executed.

C. Metadata Processing

The dataset includes extensive metadata. Queries are often executed on this metadata as a primary step to retrieving multimedia content. The ROXIE queries are executed on the tables generated to return records based on query parameters. The following query is used to return a subset of images.

Imags: =ImagesPlaces (STD.Str.Find (field2, 'United + States: Country', 1) != 0);

In next step, a table is generated in which each record contains the ImageID and zip code where the image was taken.

ZipImgs := PROJECT (Imgs, TRANSFORM (ResultTable,ZipPos:=Std.Str.FIND(LEFT.field2,'Zip'); SELF.field1 := LEFT.field1; SELF.field2 :=LEFT.field2 [ZipPos-6..ZipPos-2]));

In a similar manner, a table can be generated with other query parameters and JOIN operation between the resultant tables generates records that meet all query parameter conditions. The following queries generate a table SortedResults which sort images based on Zip code where the image was taken.

ResultImages: = JOIN (s1, ZipImgs, LEFT.field2 = RIGHT.field1, TRANSFORM (MyOutRec, SELF.ID:= LEFT.field2; SELF. Camera: = LEFT.field8; SELF.Zip:= RIGHT.field2));

SortedResults: = SORT (Table (ResultImgs, {ID, Camera, Zip}), Camera);

When a query is executed in ECL, an execution graph is generated in the form of graphical representation of the data flow for the query. In the data analysis component of the system, image features are extracted based on the RGB values calculated for each image. Queries are executed in the THOR cluster to extract the image features. For example, the RGB color space is converted to HSV and the huge component is used to compute the color.

V. CONCLUSION

In this paper we have proposed a simplified framework for efficient processing of multimedia big data. With the help of HPCC Systems, we were able to configure a multi-node cluster capable of processing a dataset of hundred million images. Some of the challenges we faced were lack of any image processing libraries in HPCC Systems or ECL. The metadata files were analyzed and simple, efficient image processing algorithms were used to extract features. Because of HPCC Systems' unique multi-cluster design, these jobs could be executed in parallel. Data points extracted from both were used to execute sample queries to retrieve images. HPCC systems guarantee optimized and efficient execution of these queries.

VI. FUTUREWORK

The core contributions of this paper can be extended to suit a variety of applications other than just multimedia content retrieval. By analyzing spatiotemporal points of image data and running a processing algorithm. We can improve accuracy of many prediction engines that previously did not use image data. For example, systems that predict weather patterns can analyze images from past and use spatial-temporal data to improve accuracy. Due to resource and time constraints, we were not able to develop the system to its full capabilities. We would like to extend the framework to support the processing of videos as well. It would be useful to compare the system to an implementation in a different distributed computing framework and computing clusters with GPU power.

REFERENCES

1. Chang, B. R., Tsai, H. F., & Wang, Y. A. (2016, April). Optimized Multiple Platforms for Big Data Analysis In Multimedia Big Data (BigMM), 2016 IEEE Second International Conference on (pp. 155-158).
2. "HPCC Systems: ECL Programmers Guide. Boca Raton Documentation Team," 2015.
3. "HPCC Systems: HPCC Client Tools. Boca Raton Documentation Team," 2014.
4. "HPCC System: Using ECL Watch. Boca Raton Documentation Team," Huang, T. C., Chu, K. C., Zeng, X. Y., Chen, J. R., & Shieh, C. K. (2016, April).
5. CURT MapReduce: Caching and Utilizing Results of Tasks for MapReduce on Cloud Computing. In Multimedia Big Data (BigMM), 2016 IEEE Second International Conference on (pp. 149-154).
6. Chang, B. R., Tsai, H. F., & Wang, Y. A. (2016, April). Optimized Multiple Platforms for Big Data Analysis In Multimedia Big Data (BigMM), 2016 IEEE Second International Conference on (pp. 155-158).
7. "HPCC Systems: ECL Programmers Guide. Boca Raton Documentation Team," 2015.
8. "HPCC Systems: HPCC Client Tools. Boca Raton Documentation Team," 2014.
9. "HPCC System: Using ECL Watch. Boca Raton Documentation Team," Huang, T. C., Chu, K. C., Zeng, X. Y., Chen, J. R., & Shieh, C. K. (2016, April).
10. CURT MapReduce: Caching and Utilizing Results of Tasks for MapReduce on Cloud Computing. In Multimedia Big Data (BigMM), 2016 IEEE Second International Conference on (pp. 149-154).
11. "Distributed processing of big data across clusters in cloud computing", Narne Devender, and Pampati Nagaraju, International Conference on Cloud Computing at JITS at Narsampet, Telangana Dist., in month of November 2013.
12. Ryu, Chungmo, et al. "Extensible video processing framework in apache hadoop." Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on. Vol. 2.

AUTHORS PROFILE

Mr. Pampati Nagaraju is working as Assistant Professor in CSE Department at BITS, Warangal affiliated to JNTU Hyderabad. He has more than 15 years of teaching experience and published around 10 papers. Qualified in APSET-2012 (July). NPTEL Online Course Certifications in Problem Solving Through Programming in C, Java Programming, and Cloud Computing. His academic interest includes Web Programming, Computer networks, Operating Systems, and Linux. He is a Member in CSI, ISTE, and ISCA.

Mr. Etyala Ajith Kumar is working as Assistant Professor in CSE Department at TPCE, Kazipet. He has 4 years of teaching experience and published 3 papers.