Contents

INTRODUCTION

The mapping of archive images into different classes where each class provides the same information about the image archive as the entire image collection can be referred to as image clustering.

The basis for clustering in the project is an information theoretic principle called ‘Information Bottleneck’ which automates the clustering based on information loss when merging the clusters to form a new higher level cluster. The type of clustering is unsupervised and agglomerative in nature. To avoid expensive pixel manipulation during clustering, the clustering is performed over Gaussian Mixture Model of every image.

This type of clustering facilitates the image retrieval through inexpensive search for closest cluster based on GMM and application of optimized image search algorithm on limited set of images.

The limitation of hardware resources and the tolerance of time consuming present a bottleneck in processing a large amount of images. The techniques of parallel computing and distributed systems are no doubt the suitable choices.

The development environment is a distributed system managed by Hadoop over LINUX file system where inexpensive commodity hardware is used. Map Reduce is a framework that allows certain kinds of problems particularly those involving large data sets to be computed using many computers.

We try to parallelize the image clustering algorithm on Hadoop, an open source system that implements the Map Reduce programming model to reduce the time complexity.

Overview

The structure of document is developed so as to give brief overview of the system to be developed. The document gives a brief introduction along with scope, methodologies and architecture of the system. The document is organized as follows: the current section explains the scope and background of the system. The following section presents system definition along with software and hardware architecture. The next section presents an overview of the methodology of the system implementation. The document concludes with project team along with roles and responsibilities.

The proposed system maps a collection of images into a set of classes that contains similar information, which actually is clustering. Image clustering is a heavy task in terms of time and space complexity when the individual image size is large and the number of images is large. The system takes a number of images as input, simply creates a model of every image called Gaussian Mixture Model (GMM), and stores it into Distributed File System (HDFS) as Hbase file along with the original image and its metadata. Agglomerative Information Bottleneck (AIB) is used for clustering process which takes GMM of every image as input from HBase, performs a map reduce job to cluster and stores output in Hbase file. User can access the system through web interface where he/she can see the clustered images.

Scope

The document addresses a brief overview of the system to be developed. However, this document does not address the details of the implementations.

Background

There has been a growing interest in developing effective methods for searching large image databases based on image content. Most approaches are focused on (search by query) and development of image browsing environment. A key step for structuring a given database and for efficient search is image content clustering. The goal is to find a mapping of the archived images into classes such that the set of classes provides essentially the same prediction, or information, about the image archives' as the entire image set collection. The generated classes provide a concise summarization and visualization of image content. The similarity between the images in the database is determined by selected feature space representation (e.g., color, texture).

Image clustering is a heavy task in terms of time and space complexity when the individual image size is large and the number of images is large. The limitation of hardware resources and the tolerance of time consuming present a bottleneck in processing a large amount of images. The techniques of parallel computing and distributed systems are no doubt the suitable choices.

The system is being developed for Department of Electronics and Computer Engineering, IOE Pulchowk campus as a Major project for the partial fulfillment of the requirement for the bachelor's degree in Computer Engineering.

SYSTEM DEFINITION

The system is an image clustering application that distributes the task of clustering based on information content using Distributed computing over HADOOP file system (HDFS) to reduce time and space complexity. The system is scalable in both space (to high extent) and time (to some extent).

The basis for clustering is an information theoretic principle called 'Information Bottleneck' which automates the clustering based on information loss when merging the clusters to form a new higher level cluster. The type of clustering is unsupervised and agglomerative in nature. To avoid expensive pixel manipulation during clustering, the clustering is performed over Gaussian Mixture Model of every image.

Overview

Initial Phase

The system takes a collection of images as input, creates a model of every image called Gaussian Mixture Model (GMM), and stores it into Distributed File System (HDFS) as Hbase file along with the original image and its metadata.

Clustering Phase

Agglomerative Information Bottleneck (AIB) is used for clustering process which takes GMM of every image as input from HBase, performs a map reduce job to cluster and stores output in Hbase file.

Output Phase

User can access the system through web interface where he/she can see the clustered images. The user interface communicates with HBase database and HDFS through REST interface provided by HBase and the file system API of HADOOP.

Scope

Image archive clustering is important for efficient handling (search and retrieval) in large image databases. In the retrieval process, the query image is initially compared with all the cluster centers. The subset of clusters that has the largest similarity to the query image is chosen. The query image is next compared with all the images within the selected subset of clusters. Search efficiency is improved due to the fact that the query image is not compared exhaustively to all the images in the database.

User often requires a browsing mechanism (for example, to extract a good query image). A high quality browsing environment enables the users to find images by navigating through the database in a structured manner. Foe example, hierarchally clustering the database into a tree structure, imposes a coarse to fine representation of image content within clusters and enables the users to navigate up and down the tree levels. A key step for structuring a given database and for efficient search is image clustering.

System architecture

The system consists of a web server, a distributed file system and a client application. The web server is the source of images. The images are downloaded and distributed over HDFS file system as HBase File. The client application is a Hbase client that communicates with Hbase through REST interface.

The data downloaded from the web server is processed by a map reduce job which generates the Gaussian Mixture Model of the downloaded images and inserts into Hbase table. The schema for the Hbase Table is shown below:

RowKey	ImageData			ImageInfo	
	Split1	Split2	Split3	ImageHeader	GMM

Agglomerative Information Bottleneck (AIB) Algorithm is applied taking GMM as input. The distance measure for clustering is obtained by calculating the information loss when merging two clusters, which is obtained from information bottleneck principle.

System Technical Architecture

The input source to the system is a web server that hosts collection of images. A list of images is created and stored in HDFS. A map reduce job downloads each image from the web server and creates GMM. The image is spitted into chunks of 3MB each and stored in Hbase along with its image header and GMM. Every image is resized to fit width of 200 pixels before creating GMM.

The clustering is done using AIB algorithm which takes GMM as input, merges two similar clusters so that information loss between the clusters is minimum and creates a central GMM for the resulting cluster.

The user interface is a web interface which accesses data from Hbase through REST interface. The expected output is the set of cluster with similar images.

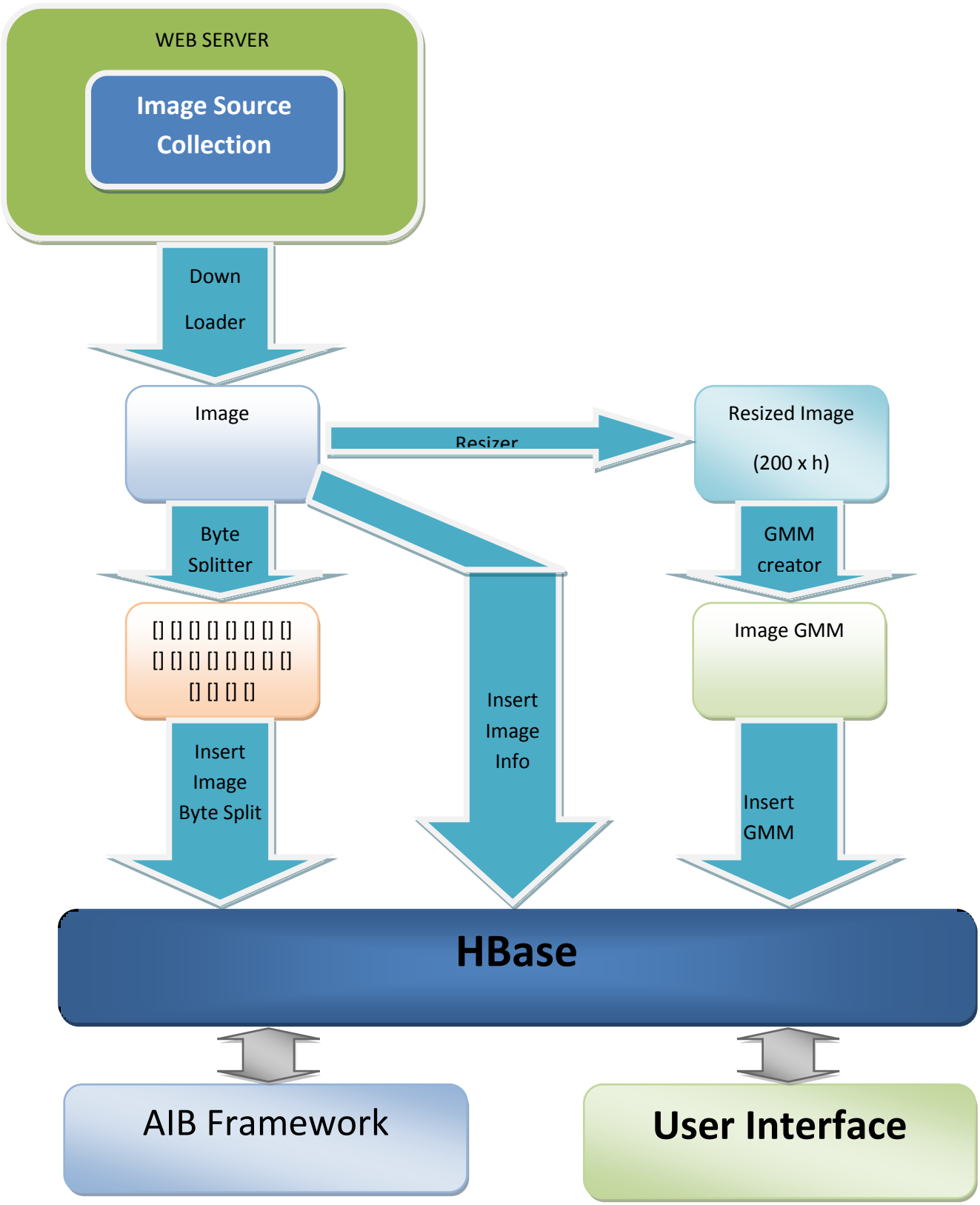

Fig: System technical architecture

System Hardware Architecture

The hardware architecture consists of a remote client, HADOOP file system and a web server. The figure below shows the general hardware architecture. Each rectangle in the figure represents a single computer. HADOOP file system consists of a single name node and multiple data nodes which is scalable as per requirement.

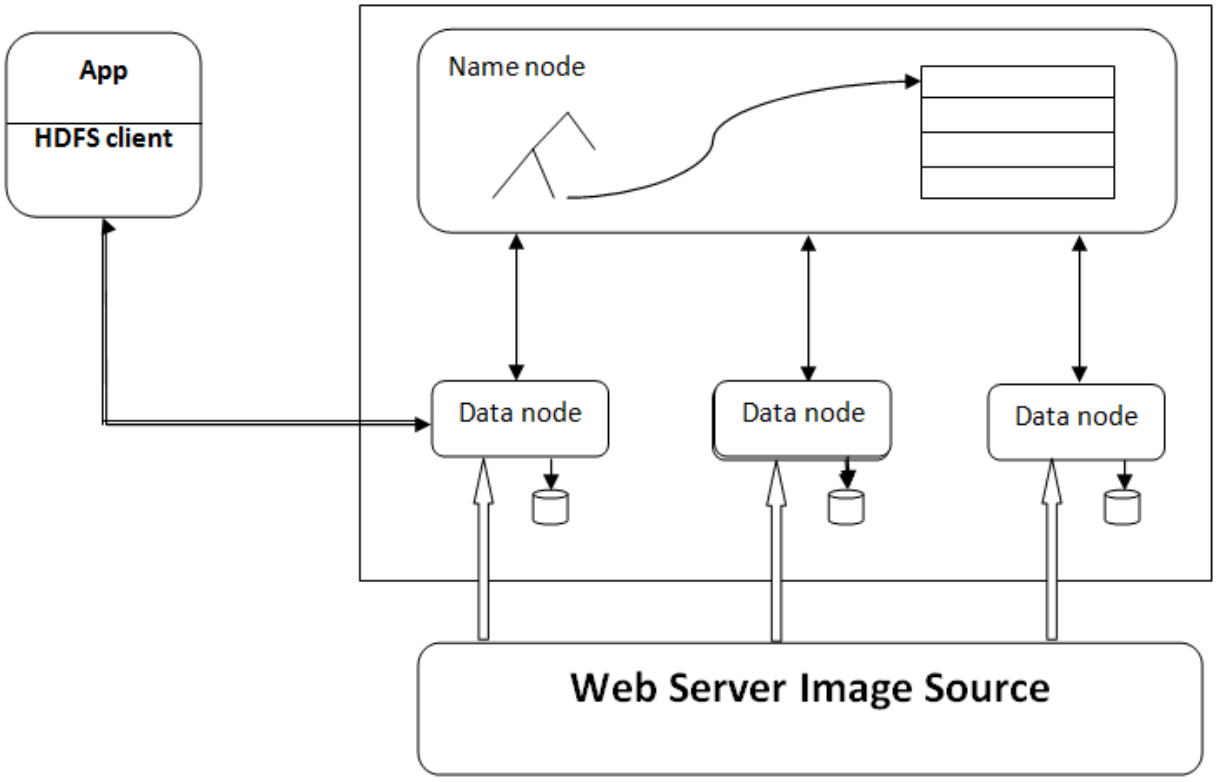

Fig: System Hardware Architecture

	Client	Web Server	Name Node	Data Node
Memory	1GB	4GB	4GB	2GB
Processor	Pentium IV or above	Core 2 Duo 2GHz	Core I5	Core 2 Duo 2GHz
NOS.	1	1	1	10

METHODOLOGY

The clustering method is based on hierarchical grouping: Utilizing a Gaussian mixture model, each image in a given archive is first represented as a set of coherent regions in a selected feature space. Images are next grouped such that the mutual information between the clusters and the image content is maximally preserved. The appropriate number of clusters can be determined directly from the IB principle.

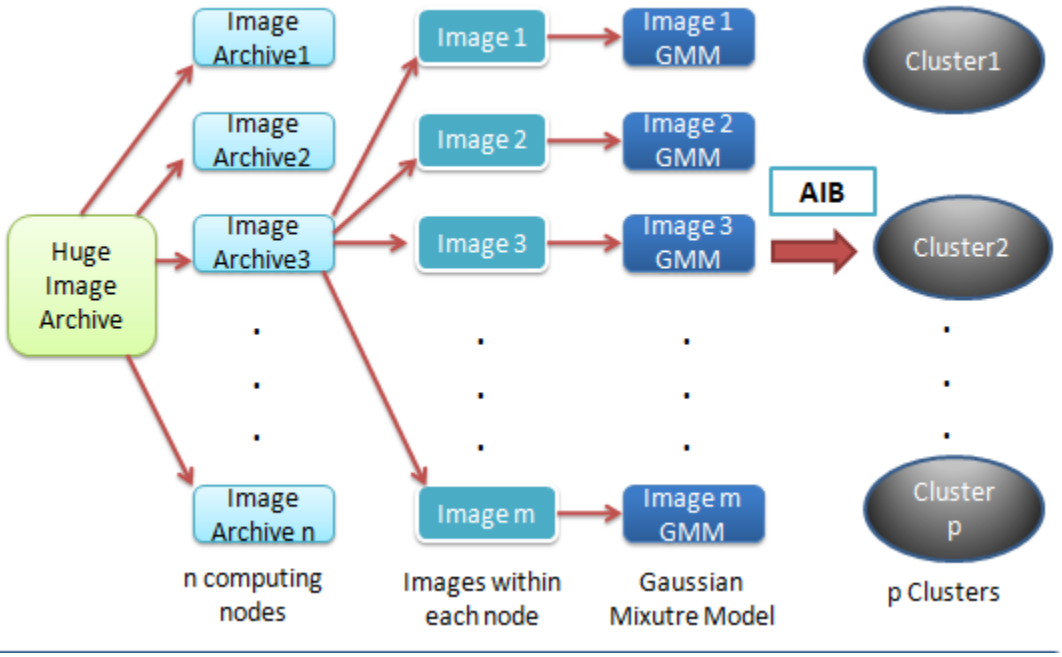

Figure: General Methodology

Implementation Plan

Image representation

- Pixels -> feature vectors -> regions

Feature space = color (CIE-lab), Spatial(x, y)

- Grouping the feature vectors in a 5-dimensional space
- Image is modeled as a Gaussian mixture distribution in feature space

Image representation via Gaussian mixture modeling(GMM)

$$f(y) = \sum_{j=1}^{k} \alpha_j \frac{1}{\sqrt{(2\pi)^d |\Sigma_j|}} \exp\{-\frac{1}{2}(y-\mu_j)^T \Sigma_j^{-1}(y-\mu_j)\}.$$

The Agglomerative IB algorithm for image clustering is:

1. Start with the trivial clustering where each image is a cluster.

2. In each step, merge cluster C1 and C2 such that information loss d(c1,c2) is minimum.

$$d(c_1, c_2) = \sum_{i=1,2} \frac{|c_i|}{|X|} D_{KL}(f(y|c_i)||f(y|c_1 \cup c_2))$$

Where | X| is the size of image database.

$$D_{KL}(f(y|c_1)||f(y|c_1 \cup c_2)) \cong \frac{1}{n} \sum_{t=1}^{n} \log \frac{f(y_t|c_1)}{f(y_t|c_1 \cup c_2)}.$$

which is Kullback-Liebler divergence.

$$f(y|c) = \frac{1}{|c|}\sum_{x \in c} f(y|x) = \frac{1}{|c|}\sum_{x \in c}\sum_{j=1}^{k(x)} \alpha_{x,j} N(\mu_{x,j}, \Sigma_{x,j}).$$

k(x) is the number of Gaussian components in f(y|x).

f(y|x) is a GMM distribution, the density function per cluster c, f(y|c), is a mixture of GMMs and therefore it is also a GMM.

3. Continue the merging process until the information loss d(c1 ,c2) is more than a pre-defined threshold, indicating that we attempt to merge two non-similar clusters.

PROJECT TEAM

Team member

Bhanu Bhakta Sigdel

Pradeep Bista

Rameshwor Parajuli

Sandip Pandey

Roles and responsibilities

1. Initiation Phase

>Data Collection and Distribution

>GMM creation

2. Processing Phase

>Calculating information content from every GMM (using IB principle)

>Calculation of Information loss in merging two GMMs

>Merging of two GMMs to create resultant GMM

>Application of Agglomerative clustering

>Threshold tuning

>Testing results

3. Output Phase

>Creation of a HTML template

>Accessing Hbase

>Casting Byte Array into images

>Rendering images

REFERENCES

- *Makho N gazimbi* , Data Clustering using Map Reduce Boise State University . March 2009
- *S. Gordon , H. Greenspan, J. Goldberger*. Applying the information bottleneck principle to unsupervised clustering of discrete and continuous image representations .Tel-Aviv University Israel,
- *N. Slonim and N. Tishby*. Agglomerative information bottleneck. In Proc. of Neural Information Processing Systems, pages 617-623, 1999.
- *E. Schneidman, N. Slonim, N. Tishby, R. R. deRuyter van Steveninck, and W. Bialek.*Analysing neural codes using the information bottleneck method. In Advances in Neural Information Processing Systems, NIPS, 2001.
- *Li-shun, J., Ding-sheng*.: Research on K-Means Clustering Parallel Algorithm of Remote Sensing Image. Remote Sensing Information 01, 27–30 (2008)