A GPU-based MapReduce Framework for MSR-Bing Image Retrieval Challenge

Lei Wang†, Hanli Wang†⋆, and Bo Xiao†

† Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai 200092, P. R. China
⋆ Department of Computer Science, Tongji University, Shanghai 201804, P. R. China
Email: {110_wangleixx, hanliwang, 1314xiaobo}@tongji.edu.cn

Abstract—This paper presents a large-scale image retrieval system based on an efficient Graphics Processing Units (GPU)-based MapReduce framework for the MSR-Bing Image Retrieval Challenge. The proposed system is designed for searching images and scoring image-query pairs based on their relevances efficiently and accurately. Unlike the former systems which usually start with text queries to select partial images and then process their visual contents, the proposed system attempts to search similar images directly from the entire dataset through visual content and then compare their text similarities, owing to the powerful computational capabilities of the proposed GPU-based MapReduce framework. It is shown that the proposed system achieves 0.492 in terms of DCG@25 on the final evaluation.

Keywords—MSR-Bing Image Retrieval; Scoring System; Text Similarity; GPU; MapReduce

I. INTRODUCTION

Image retrieval, as a regular but widely used application (e.g., Bing Image, Google Image and Baidu Image, etc), is becoming more and more challenging since huge amounts of multimedia data are produced everyday with the development of social networks. Generally, the task for traditional image retrieval system is to search and return a list of relevant results from billions of images with different backgrounds and viewpoints, and the retrieval process should be fast and accurate. Most of the state-of-the-art image retrieval systems [1] are built upon local features [2], [3] and the Bag-of-Words (BoW) [4] representation because of their discrimination and efficiency. And a number of researches have been made to enhance the retrieval performance.

As a matter of fact, local feature based image retrieval systems suffer from three problems: the locality of features, quantization errors and burstiness [5]. The method of Nested-Scale Invariant Feature Transform (SIFT) [6] solves the problem of feature locality by adding spatial information into matching process. Multiple Assignment (MA) [7] reduces the impact of quantization errors by assigning features into more than one visual words while Hamming Embedding (HE) [8] refines matching by calculating binary signatures. Moreover, several image normalization methods [5] improve retrieval accuracy by handling burstiness.

Recently, for more compact representations, aggregation models are proposed. The Vector of Locally Aggregated Descriptors (VLAD) [9], [10] based methods like Aggregated Selective Match Kernel (ASMK) [11] and democratic aggregation [12] both improve the precision and efficiency. Moreover, spatial re-ranking [13], [14] and query expansion [15], [16] are employed to refine the initial retrieval results.

In order to build a more powerful and robust large-scale image retrieval system, we are engaged in the MSR-Bing Image Retrieval Challenge (MSR-Bing IRC) [17]. Different from traditional image retrieval mentioned above, the goal of MSR-Bing IRC is to encourage the contestants to build efficient image retrieval and scoring systems for assessing the effectiveness of query terms in describing the images crawled from the web for image search purposes. A contesting system is asked to produce a floating-point score on each image-query pair that reflects how relevant the query could be used to describe the given image, with higher numbers indicating higher relevance [18].

Two datasets are available to develop and evaluate the scoring system, including a training set which contains a big table of 23,094,592 different triads of (image, query, click count) and 1M unique images. Moreover, another manually labeled development set contains 1,000 queries and 79,665 images used for self-test. Three levels are defined to measure the relevance between images and queries, including Excellent, Good and Bad. The Discounted Cumulated Gain (DCG@25) [19] is introduced to evaluate the ranking results against the manually labeled ground-truth. To compute DCG@25, an ordered image list for each query is generated with a list of floating-points returned from the retrieval system. DCG@25 for each query is then calculated as

$$DCG_{25} = 0.01757 \sum_{i=1}^{25} \frac{2^{rel_i} - 1}{\log_2(i + 1)},$$

where $rel_i = \{\text{Excellent} = 3, \text{Good} = 2, \text{Bad} = 0\}$ is the manually judged relevance for each image with respect to the query, and 0.01757 is a normalizer to make the score for 25 Excellent results equal to 1. The final metric is the average of all queries in the test set.

In this work, facing MSR-Bing IRC, a GPU-based MapReduce Framework named as GMRF is proposed, aiming to fully investigate the parallel power of the fusion of MapReduce
Fig. 1. Overall structure of GMRF.

and GPU. MapReduce [20] is a parallel programming model originally proposed by Google to process large-scale data. It highly abstracts the process of complex parallelization and distribution of computations, and offers automatic task/data management, inter-machine communication as well as fault tolerance. On the other hand, GPU [21] is initially designed to accelerate computer graphic applications, and is becoming more suitable for data-parallel computations by distributing the data amongst massive computing units to achieve parallelization.

The proposed GMRF is a computational framework to investigate the processing power of MapReduce and GPU in a joint manner for addressing large-scale image retrieval. The contributions of this work are three-fold. First, the mechanism of data loading and transmission is specifically optimized to improve the throughput capacity of GMRF. Second, a novel programming prototype referred as Worker Prototype is designed to exploit jointly the computing power of CPUs+GPUs. Third, GMRF is optimized by removing several unnecessary procedures as compared with the traditional MapReduce framework. Moreover, GMRF is implemented and employed for MSR-Bing IRC for providing powerful computational resources, thus supports sufficient search efficiency. The rest of this paper is organized as follows. The proposed GMRF results are shown in Section IV. Finally, Section V concludes this paper.

II. PROPOSED GMRF FRAMEWORK

The overall structure of the proposed GMRF is shown in Fig. 1. The base of GMRF is composed of a number of computers as a cluster, acting as the hardware infrastructure for distributed computing and storage. Above the ‘Cluster & Distributed System’ is the Hadoop Distributed File System (HDFS) [22] and MapReduce Layer. The functionality of HDFS is to automatically manage all of the data, and MapReduce is responsible for automatic task management, inter-machine communication and fault tolerance. Another layer is the CUDA [23] which is a popular parallel computing platform and programming model invented by NVIDIA. CUDA is employed to enable dramatic increases in computing performance by harnessing the power of GPU. The top layer is the core of the proposed GMRF, including the GMRF Driver and Worker Prototype.

Regarding the GMRF Driver, it is introduced in our proposed GMRF framework as illustrated in Fig. 2, where the modules in dashed boxes are replaced or modified as compared with the traditional MapReduce framework. Moreover, GMRF is implemented and employed for MSR-Bing IRC for providing powerful computational resources, thus supports sufficient search efficiency. The rest of this paper is organized as follows. The proposed GMRF is detailed in Section II. In Section III, the implementation of GMRF for MSR-Bing image retrieval is introduced. Evaluation
saving of network bandwidth. In addition, the data is loaded and processed within the Worker Prototype (which will be discussed next) according to the Filelist received before. The intermediate results generated from the Worker Prototype for each Map are then combined into several fixed-size data blocks and then transmitted directly to HDFS. Inevitably, a small amount of data fragments that cannot form a data block will be left and transmitted to the Reduce module and processed in the traditional way. Therefore, the Reduce module is greatly simplified. Moreover, the modules of Partition and Sort are removed from GMRF Driver to skip unnecessary computations.

As far as the Worker Prototype is concerned, it is another key feature in the proposed GMRF, which is employed to accomplish most computations of each task dispatched from the GMRF Driver. The Worker Prototype should be implemented with respect to each specific task, e.g., feature extraction, K-Means clustering, bag of features generation, and so on. It is also well designed to be capable of extending to other multimedia data mining applications besides large-scale image retrieval. In order to obtain high performances, the Worker Prototype fully harnesses both the power of CPU and GPU, and balances the computational load between them in a joint automatic manner.

III. MSR-BING IMAGE RETRIEVAL WITH GMRF

The implementation of GMRF for MSR-Bing image retrieval is briefly described herein, including hardware configuration and integrated scoring algorithms.

A. Hardware Configuration

The proposed GMRF is currently implemented with a 12-node computer cluster equipped with multiple GPUs. The detailed information about these computer nodes is given in Table I. Each node is equipped with 1 CPU and 2 GPUs, and gigabit-nics are used among them for data transfer. The Java 1.7 version and Hadoop 1.0.4 version are set on all the nodes with the GNU/Linux Ubuntu 12.04 operating system.

B. Offline Training System

The process of offline data training will be discussed herein. For MSR-Bing image retrieval, a large-scale dataset is provided as the training set to build a retrieval and scoring system, which contains a big table of about 23M different triads of (image, query, click count) and 1M unique images. All of these data are collected from the MSR-Bing search engine, and it is very desired that the queries are correctly corresponding to the related images. In order to achieve this, a post-processing procedure is performed to re-generate a more credible text data pool. Moreover, for searching similar images from such a big dataset efficiently and accurately, the standard BoW model appended with two advanced algorithms including Hamming Embedding (HE) and Weak Geometric Consistency (WGC) [8] is employed to achieve a visual data pool.

1) Text Data Training: As a matter of fact, the training dataset contains about 23M raw user clicks from Bing search log. It suffers from three kinds of problems. First, some of the queries in the triads are misspelled due to the inattention of users. Second, there are a few meaningful words like preposition or article in a portion of triads. Third, a number of queries share one word of the same meaning but in different grammatical form. The aforementioned problems will cause redundancy and bias when counting the real click count of one triad and need to be corrected.

For a given query text, two powerful toolkits are engaged to enhance the textural processing performance, including NLTK [24] and word2vec [25]. Regarding NLTK, it is a leading platform for building Python programs to work with human language data. As far as word2vec is concerned, it is the state-of-the-art toolkit for building a neural network model to map words to vectors, which provides an efficient implementation of the continuous BoW and skip-gram architectures for computing vector representations of words.

As shown in Fig. 3, the query in each triad is firstly filtered by a spelling checker, and then, all the meaningless words are removed by a stop remover. After that, a standard EN-lemmatizer is utilized to change all the words to noun form. By means of these steps, a certain number of different queries
can be combined, which improves the reliability of training data. Moreover, we use the pre-trained vectors generated on part of the Google News dataset (about 100 billion words) as our major corpus released at the homepage of word2vec project. The model contains 300-dimensional vectors for 3 million words and phrases which are obtained using a simple data-driven approach as described in [26].

2) Visual Data Training: When training visual content, the standard BoW model is employed as shown in Fig. 3. The Hessian-Affine detector [27] and SIFT descriptor are applied for feature extraction and description. Moreover, a 100k-size visual dictionary is generated by using all of the descriptors obtained from the entire 1M images and employing the flat K-Means clustering algorithm. To enhance the retrieval accuracy, two extra methods including HE and WGC [8] are implemented based on the technique of inverted file index [28]. And for each node in inverted files, it contains ImageID, HE signature, feature scale and dominant orientation.

C. Online Evaluation System

As illustrated in Fig. 4, the online evaluation system is composed of two modules: image retrieval and text comparison. The image retrieval module mainly focuses on the search of near-duplicated copy of images through visual features, with HE and WGC employed to refine the matching process in BoW and improve the overall retrieval performance. Given an image, its SIFT descriptors are extracted at first, and then, each feature is assigned to the nearest visual word in the visual dictionary pre-trained. Meanwhile, relevant attributes of the features are set into the inverted files. After that, GMRF scans all of the inverted files and finds the most similar ones with highest matching scores with HE and WGC employed.

Once given an image-query pair, the system firstly processes the image and retrieves \( n \) \((n = 10)\) in our implementation) most similar images from the test image set. In fact, each image corresponds to several different texts, and each text is required to be compared with the cleaned test query, which is filtered by spelling checker, stop word remover and word lemmatizer as mentioned above. Then, the final score is calculated by multiplying the click count which is considered as a confidence factor. The final scoring function is computed as

\[
rel(I,Q) = \max \left( T(Q,t_i) \times \log_{10}(\text{clickcount} + 1) \right), \tag{2}
\]

where \((I,Q)\) is a test image-query pair posted by users, and \(t_i\) is defined as a text that is related to the retrieved images.

It should be noted that we do not involve image relevance in the above scoring function here because it has been included implicitly since we only retrieve the texts of the top-\( n \) similar images.

IV. EVALUATION

In this section, the computational performance as well as the retrieval accuracy in terms of DCG@25 on both the DEV set and final evaluation set are presented.

A. Computational Complexity of Offline Training

During the training process, the most time-consuming task is to generate BoW vocabularies by K-Means clustering. With the aid of the proposed GMRF, all of the descriptors \((i.e., 392,528,043)\) extracted from the entire 1M image dataset are employed to generate a 100k-size visual dictionary within a short period of time. Table II shows the training time of clustering with GMRF, as well as the task of HE/WGC generation. All of these tasks are completed in several hours, which achieves an amazing improvement in computational speed of about \(100 \times \sim 1500 \times\) times faster than that of a single machine.

<table>
<thead>
<tr>
<th>Tech</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-Means clustering</td>
<td>21,600</td>
</tr>
<tr>
<td>HE/WGC generation</td>
<td>5,400</td>
</tr>
</tbody>
</table>

B. Evaluation Performance

Table III shows the average DCG@25 conducted on both the DEV set and the final evaluation achieved by GMRF. A random score on the DEV set is measured as the baseline of retrieval performance. By using the proposed system introduced above, 0.01 improvement is achieved, and the average time for scoring an image-query pair is about 1.0 second. In fact, the most time-consuming part in the whole system is feature extraction and description since it is handled on a single node.

V. CONCLUSION

This paper introduces our retrieval system for MSR-Bing image retrieval. A number of efficient computer vision tech-
Fig. 3. Offline training.

TABLE III. RETRIEVAL PERFORMANCE.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>DCG@25</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEV</td>
<td>Random</td>
<td>0.470</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GMRF</td>
<td>0.480</td>
<td></td>
</tr>
<tr>
<td>FINAL</td>
<td>Random</td>
<td>0.486</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>GMRF</td>
<td>0.492</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Techniques and text analysis methods are implemented on the proposed GPU-based MapReduce framework GMRF. The evaluation results demonstrate the effectiveness of our framework for searching and scoring as well as the powerful computing capability of GMRF. Moreover, GMRF is a basic and universal framework that more powerful and advanced techniques can be implemented on it.

REFERENCES


