Universal Journal of Computer Science and Engineering Technology
1 (1), 31-35, Oct. 2010.

Distributing Arabic Handwriting Recognition System Based on the Combination of Grid Meta-Scheduling and P2P Technologies (Omnivore)

Hassen Hamdi
Mir@cl Lab, FSEGS
University of Sfax, BP 1088, 3018 Sfax, Tunisia
hhassen2006@yahoo.fr

Maher Khemakhem
Mir@cl Lab, FSEGS
University of Sfax, BP 1088, 3018 Sfax, Tunisia
maher.khemakhem@fsegs.rnu.tn

Abstract—Character recognition is one of the oldest fields of research. It is the art of automating both the process of reading and keyboard input of text in documents. A major part of information in documents is in the form of alphanumeric text. Significant movement has been made in handwriting recognition technology over the last few years. Up until now, Arabic handwriting recognition systems have been limited to small and medium size of documents to recognize. The facility of dealing with large database (large scale), however, opens up many more applications. Our idea consists to use a strong and complimentary approach which needs enough computing power. We have used a distributed Arabic handwriting system based on the combination of Grid meta-scheduling and Peer–to-Peer (P2P) technologies such as Omnivore. Obtained results confirm that our approach present a very interesting framework to speed up the Arabic optical character recognition process and to integrate (combine) strong complementary approaches which can lead to the implementation of powerful handwriting OCR systems .

Keywords- Large scale handwriting OCR; P2P; Grid Meta-Scheduling; Omnivore; cluster.

I. INTRODUCTION

Optical Character Recognition (OCR) is the mechanical or electronic translation of scanned images of handwritten, typewritten or printed text into machine-encoded text. It is widely used to convert books and documents into electronic files, to computerize a record-keeping system in an office, or to publish the text on a website. The process of optical character recognition of any script can be broadly broken down into five stages: Pre-processing, segmentation, feature extraction, classification and post-processing.

Pre-processing aims to produce data that are easy for the OCR systems to operate accurately. The main objectives of pre-processing are: binarization, noise reduction, Stroke width normalization, Skew correction [1].

Segmentation aims to text lines detection (amongst the most used techniques we can mention: Hough Transform, horizontal projections, smearing) and word extraction (amongst the most used techniques we can mention: vertical projections, connected component analysis).

The objective of the feature extraction stage is to represent each character by an invariant feature vector which eases and maximizes the recognition rate with the least amount of data. Feature extraction methods are based on 3 types of features: statistical, structural and global transforms and moments.

In the classification step, there is no such thing as the "best classifier". The use of classifier depends on many factors, such as an available training set, a number of free parameters etc. such as k-Nearest neighbors (k-NN), Bayes Classifier, Neural Networks (NN), Hidden Markov Models (HMM), Support Vector Machines (SVM), Euclidean distance, and so on…

The post-processing stage, which is the final stage, aims at improving the recognition rate by refining the decisions taken by the previous stage; it can be at least a speller check which uses a set of lexicons.

Handwriting OCR still constitutes a big challenge, especially if we need to computerize a big amount of documents, despite the wide range of proposed approaches and techniques which attempted to solve the inherent problems [2]. Indeed, the complex morphology and the cursive aspect of this writing are behind the weakness of the proposed approaches. A deep observation of the existing proposed approaches and techniques lead to the conclusion that maybe the combination (integration) of some of them, which are very complementary, can lead to the implementation of powerful Handwriting OCR systems. Unfortunately, such combination requires, surely, a huge amount of computing power owing the fact that most of these approaches and techniques are complex in terms of computing. Hopefully, distributed infrastructures such as LAN, clusters and grid computing can provide enough

Corresponding Author: Hassen Hamdi, Mir@cl Lab, FSEGS, University of Sfax, BP 1088, 3018 Sfax, Tunisia.

computing power which can be exploited and used to solve our problem.

Today's Computing Grids are primarily used to connect dedicated compute clusters. Building dedicated compute clusters requires considerable administrative and fiscal resources. Often, necessary compute power is already available in the form of desktop computers - incorporating them into on-demand resource pools prevents investments in additional computer systems, alleviating the problem of resource wastage.

In this paper, we propose a novel approach to distribute the Arabic OCR based on on the combination of Grid meta-scheduling and Peer–to-Peer technologies, namely Omnivore to incorporate unused resource pools. Experimental results prove the validity of our approach to speedup the recognition process.

Our approach uses a Grid meta-scheduling system as a frontend to the user. By that the user experience no difference to standard Grid submission systems and only small differences to cluster scheduling systems. We use GridWay as deployed Grid meta-scheduler because it is widely used in Grid environments.

GridWay is a Service-oriented architecture based on a flexible, secure and coordinated resource sharing infrastructure allowing dynamic service exchange among members of several virtual communities. Allowing the use of a Grid over and beyond of the borders of organizations. The Semantic Grid highlights the information and knowledge dimension of these service exchanges [3].

In our case P2P technologies are distributed systems with auto-adaptive, self-healing, self-configuring and decentralized features. We focus on distributed hash table (DHT) based P2P systems. In these sysytems all participants are equal. By that P2P complements the classic "Client/Server" model; each participant can be either Client or Server [4].

At this point it is necessary to establish the concept of jobs. When we talk about a job we are thinking of an executable combined with some data and describe by a job description. There are some specifications as JSDL (used by GridWay) or RSL used within Grids and some proprietary as used by GridWay internally.

Omnivore is the interface between GridWay and our P2P meta-, meso-scheduler and P2P-scheduler. This means the P2P system can be either used to schedule between Grid sides without using another Grid meta-scheduler such as GridWay (as a meta-scheduler), but also as a meso-scheduler interfacing between a Grid meta-scheduler such as GridWay and Grid sides. At least it could be used as a classic scheduler scheduling between desktop computers, called just P2P scheduling. By that Omnivore and the P2P scheduling system proposed by us is very flexible. To achieve it the system provides a plugin interface. The P2P scheduler supports running jobs directly on desktop computers or in virtual machines on desktop computers. To ease the reading of our paper we subsumed Omnivore and the P2P scheduling system as Omnivore.

Omnivore is mainly thought for integrating unused desktop PCs within a PC pool and teaming them up with Grid and Cloud environments.

In contrast to GridWay our Omnivore supports not only the Linux OS platform but Windows and MacOS platforms too. Therefore it was necessary to extend the job description file with some modifications. It doesn't harm the functionality of GridWay. The additional, for Omnivore necessary information, is hidden within the existing ENVIRONMENT parameter. It is used to define environment variables for the job.

This is an example of a GridWay job description file.

GridWay job template file example:
EXECUTABLE=/does/not/matter
ARGUMENTS=-jar test.jar test
ARGUMENTS=-la /tmp
ENVIROMENT=EXEC=LOCAL,
LOCALBINARY=java.exe

GridWay job template file example:
EXECUTABLE=/bin/ls
ARGUMENTS=-la /tmp

In this description the executable could only be entered as a Linux binary. By that it is not usable for Windows platform. Therefore, Omnivore ignores the EXECUTABLE parameter and reads the real executable from the hidden information LOCALBINARY:

Additionally this information is used to select in which environment a job should be executed. At the moment Omnivore supports LOCAL (for execution directly on the system), GLOBUS (for submitting the job to a running Globus Toolkit Grid environment) and some Virtualization environments. The execution environment is specified by the parameter EXEC. In this paper we only focus on local execution.

The paper is organized as follows: section 2 describes the problem statement. An overview of our approach is presented in section 3. The details of the distribution of the studied application over the cluster computing and Omnivore then corresponding performance evaluation are described and investigated in section 4. Conclusion remarks and future work are presented in section 5.

II. PROBLEM STATEMENT

There are several Arabic teaching, practicing, research centers, but very little digital information is available about their activities and contributions to society. There are several Arabic teachers, instructors; spread all over the

country and abroad but details of their expertise and wisdom are not well known.

In many national libraries, there are several publications in the form of books, journals, research papers, conference proceedings, dissertations, and monographs. But the number of comprehensive documentation centre is limited such as in Australia [5]. Hence there is an urgent need to develop a system for monitoring and facilitate the creation of digital library.

To ease the use of such documents, archive them and make them readable by a bigger audience it is necessary to have them digitalized.

We assume that the documents be there as scanned pages in shape of images.

First it comes to mind to use one computer to recognize the images. Therefore, we started the digitalization of some a document as a sequence of words. First in this case, different Arabic words are recognized sequentially on a PC (3.4 GHZ CPU frequency, 1GB of RAM and running Windows XP-professional). The time of recognition process achieve 5.85 minutes with a single document of 9000 words. Figure 1 presents the results as a graph.

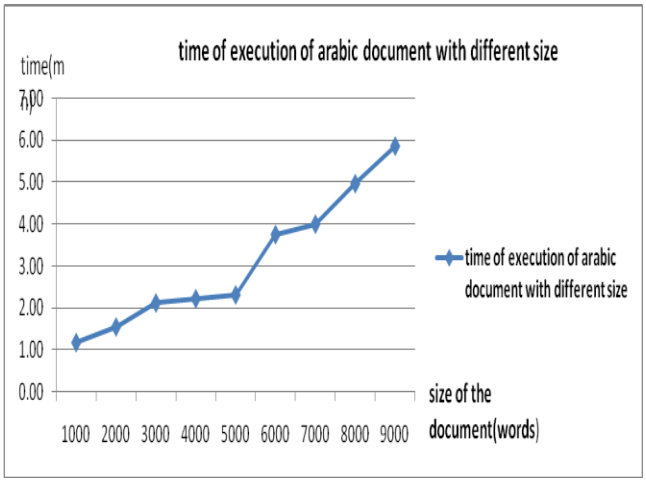

Figure 1: variation of the speedup according to the size of documents.

It is obvious that this solution is not adaptable to a huge amount of documents. Therefore, it is necessary to parallelize the recognition to shorten the used time and higher the throughput. This is possible because the recognition of a word can be seen as an atomic operation without any interconnection to the recognition of another word.

III. THE PROPOSED APPROACH

The idea of the proposed approach is to use Omnivore [6]for the OCR processing to execute the parallelized OCR jobs. To parallelize them it is necessary to create packages containing a part of the data, in our case images, a small data base and an executable combined with a job description.

We propose to split optimally the binary image of a given Arabic text to be recognized into a set of binary sub images and then assign them among some computers interconnected to the GridWay. Our Grid Computing is composed of several institutions heterogeneous computers interconnected trough the Internet. One of these computers is named the coordinator and the remaining one is named workers. The coordinator is responsible of the management of the recognition process and the coordination among workers. The coordinator is working as a web server. If we need to launch on the grid a distributed Arabic recognition process, we have first to log in to the coordinator, ask it about the number, the computing capacity and the Operating System of available workers.

IV. THE EXPERIMENTAL STUDY

In order to improve the influence of Omnivore architecture on the time of execution, we used different corpus with different size (1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000 and 9000 words) randomly chosen from the IFN/ENIT corpus data base formed of handwritten Tunisian town's names. Our application was first tested running on a cluster and then on Omnivore. Both interfaced by Gridway to have the same overhead.

We have considered also a reference library composed of 345 characters representing approximately the totality of the Arabic alphabet (including the characters shape variation according to their position within words and with different position (rotation and translation)),

The character image is divided into NxM zones. From each zone features are extracted to form the feature vector. The goal of zoning is to obtain the local characteristics instead of global characteristics.

In order to analyze our experiments, we define two factors such as the speedup and the efficiency factor.

The speedup factor defined as the ratio of the elapsed time using sequential mode with just one processor to the elapsed time using the distributed architecture and the efficiency factor defined as the ratio of the speedup factor to the number of computers or clusters participating in the work.

A. On a clusters

All jobs are executed on: Compute Nodes with 16 GByte memory, 2xDualCore Opteron 2216 HE 2.4GHz, 250 GByte SATA HD, and the network speed was 1 Gbit/s.

Figures 2 and 3 illustrate the obtained results of our experiment using distributed architecture based on clusters. These figures show in particular that:

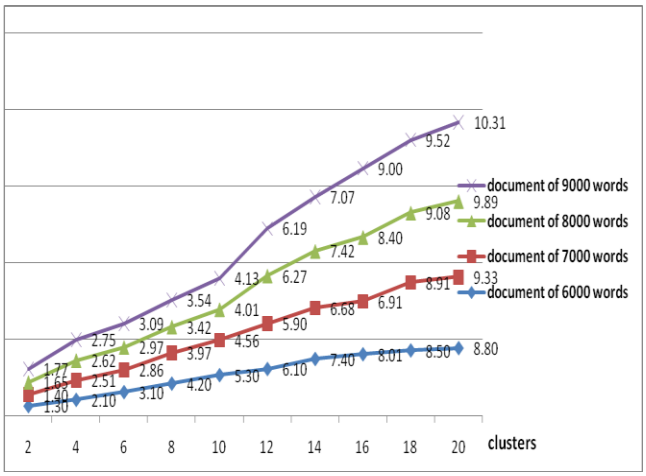

Figure 2. Speedup factor on a clusters

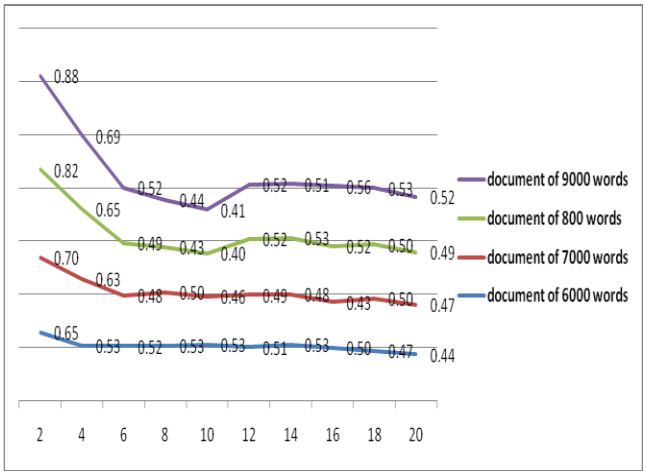

Figure 3.Efficiency factor with clusters

The speedup factor increases with the number of Compute nodes used and the efficiency factor increase with the size of the file to recognize. The efficiency factor is greater than 0.52 which means that the computing power of each dedicated compute node is used for more than 52%.

If we use 20 compute nodes then the speedup factor reaches the value 10.31 which amounts to a recognition rate around 656 characters per second which is a very interesting recognition speed compared to the existing products[7] [8].

B. With Omnivore

- We have used 20 dedicated homogeneous workers having the exact same configuration: 3.4 GHZ CPU frequency, 1GB of RAM and running Windows XP-professional, taken from a PC pool at the University of Marburg.
- The grid network capacity was 100 Mbit/s.

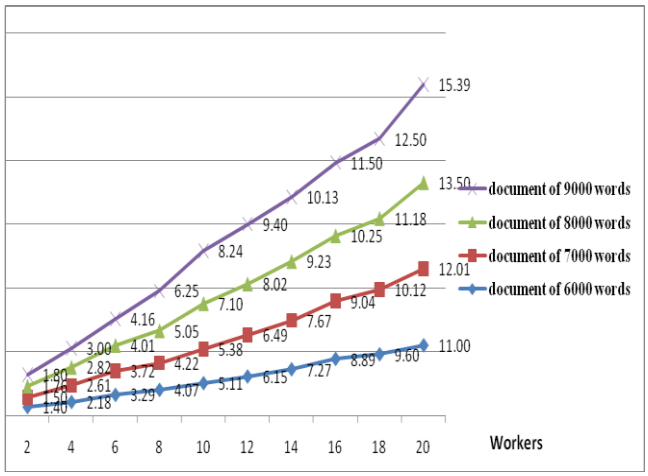

Figure 4. Speedup factor with Omnivore

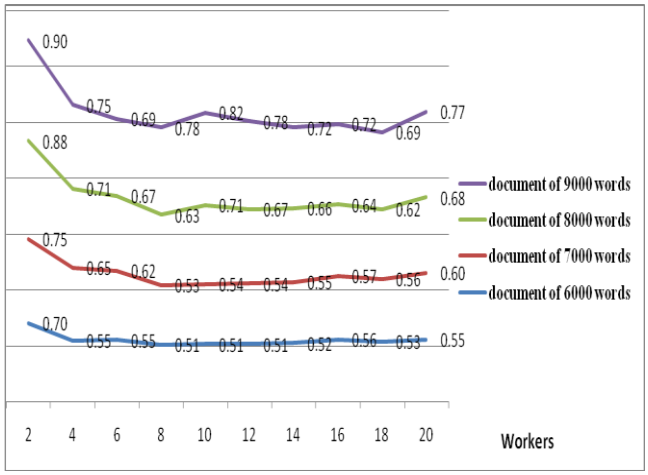

Figure 5.Efficiency factor with Omnivore

These figures 4 and 5 show the advantages of using distributed architecture based on Omnivore on the two speedup and efficiency factor.

The speedup factor increases with the number of workers used and the efficiency factor increase with the size of the file to recognize. The efficiency factor is greater than 0.77 with a file of 9000 words which means that the computing power of each worker is used for more than 77 %.

If we use a distributed architecture based on Omnivore with 20 workers then the speedup factor reaches the value 15.39 which amounts to a recognition rate around 828 characters per second which is a very interesting recognition speed compared to the existing products [8] and the results using a dedicated cluster.

V. CONCLUSION AND PERSPECTIVE

In this paper, we proposed the use of Grid Meta-Scheduling and P2P Technologies (Omnivore) for the

design of Arabic distributed OCR system to speedup the recognition process.

Performance evaluation of the proposed approach confirms that Omnivore can provide an effective framework to speedup the recognition process and integrate strong complementary approaches that can lead to the implementation of powerful handwritten OCR systems.

The proposed design approach requires further investigations. In particular, we examining how to distribute the different stages of the OCR system such as pre-processing, segmentation, feature extraction between nodes of Omnivore.

[1] G. Vamvakas, B. Gatos, I. Pratikakis, N. Stamatopoulos, A. Roniotis and S.J. Perantonis, "Hybrid Off-Line OCR for Isolated Handwritten Greek Characters", The Fourth IASTED International Conference on Signal Processing, Pattern Recognition, and Applications (SPPRA 2007), ISBN: 978-0-88986-646-1, pp. 197-202, Innsbruck, Austria, February 2007.

[2] S. Sangsawad and C. Fung Using Content Based Image Retrieval Techniques for the Indexing and Retrieval of Thai Handwritten Documents, IEEE Xplore., vol 1, june 2010.

[3] Ian Foster and Carl Kesselman, editors. The Grid: blueprint for a new computing infrastructure.Morgan Kaufmann, San Francisco, CA, USA, 1999. 82, 84, 87

[4] F. Dabek, B. Zhao, P. Druschel, J. Kubiatowicz, and I. Stoica. Towards a Common API for Structured P2P Overlays. In F. Kaashoek and I. Stoica, editors, Revised Papers from the 2nd International Workshop on P2P Systems (IPTPS' 03), volume 2735 of Lecture Notes in Computer Science, pages 33–44, Berlin, Heidelberg, February 2003. Springer-Verlag.

[5] R. Holley, How Good Can It Get? Analysing and Improving OCR Accuracy in Large Scale Historic Newspaper Digitisation Programs. D- Lib Magazine, March/April 2009, vol. 15 no 3/4

[6] M. Heidt, T. Dörnemann, K. Dörnemann, and B. Freisleben. Omnivore: Integration of Grid Meta-Scheduling and Peerto- Peer Technologies. In Proceedings of 8th International Symposium on Cluster Computing and the Grid (CCGrid 08), pages 316–323, May 2008.

[7] CiyaICR product, http://www.Ciyasoft.com,2004

[8] M.Khemakhem and A. Belghith. Towards A Distributed Arabic OCR Based on the DTW Algorithm: Performance Analysis The International Arab Journal of Information Technology, Vol. 6, No. 2, April 2009.

AUTHORS PROFILE

Hassen Hamdi received in 2008 his Master's Degree in Computer Science from the University of Sfax, Tunisia. He is currently a Ph.D student at the University of Sfax. His research interests include pattern recognition and distributed system.

Maher Khemakhem received his Master of Science, his Ph.D. and Habilitation degrees from the University of Paris 11 (Orsay), France respectively in 1984, 1987 and the Universtity of Sfax, Tunisia in 2008. He is currently Associate Professor in Computer Science at the Higher Institute of Management at the University of Sousse, Tunisia. His research interests include distributed systems, performance analysis, Networks security and pattern recognition.