Towards A Distributed Arabic OCR Based on the DTW Algorithm: Performance Analysis

Maher Khemakhem[1] and Abdelfettah Belghith[2]
[1]Miracl Lab Fsegs, University of Sfax, Tunisia
[2]Rim-Cristal Lab, Ensi University of Manouba, Tunisia

Abstract: *In spite of the diversity of printed Arabic optical character recognition products and proposals, the problem seems to be not yet well solved. The complex morphology and calligraphy of the Arabic writing on one hand and the use of some light approaches on the other hand are behind the poorness of these products. However, some strong proposed approaches didn't find the opportunity to be commercialised because of generally their corresponding complex computing. The dynamic time warping algorithm is considered as one among these strong approaches. In fact, several studies and experiments have shown and confirmed that the printed Arabic optical character recognition based on dynamic time warping algorithm provides a very interesting recognition rate especially for large and huge vocabularies. One of the attractive sides of the dynamic time warping algorithm is its ability to recognize properly connected or cursive characters (words or sub words) without prior segmentation. Furthermore, this algorithm performs the recognition process from within a reference library of isolated characters and owns a very good immunity against noises. Unfortunately, the big amount of its computing during the recognition process makes its execution time very slow and, hence, restricts its utilization. Many researchers attempted to speedup the execution time of this algorithm. Unfortunately, the corresponding proposed solutions require generally specific high cost architectures. Loosely coupled architectures such as grapes or grid computing can provide enough power without additional cost to distribute the complexity of some greedy applications. Consequently, we report in this paper the performance analysis of an analytical and an experimental study of a distributed Arabic optical character recognition based on the dynamic time warping algorithm within loosely coupled architectures. Obtained results confirm that loosely coupled architectures and more specifically grid computing present a very interesting framework to speedup the Arabic optical character recognition based on the dynamic time warping algorithm.*

Keywords: *Arabic OCR, DTW algorithm, loosely coupled architectures, grapes, grid computing, performance analysis.*

Received September 11, 2007; accepted December 22, 2007

1. Introduction

The cursive aspect of Arabic writing is behind the complexity and sometimes the poorness of the majority of the corresponding proposed Optical Character Recognition (OCR) based approaches [5], [15]. Hidden Markov Models (HMM) approaches and the Dynamic Time Warping (DTW) algorithm are considered among the best proposed approaches to the Arabic OCR [5], [2]. Their attractiveness and robustness stem from their inherent mathematical optimization capabilities during the recognition process which are achieved on either isolated or connected (cursive) characters. A comparative study has been conducted between these two approaches to investigate their consistency and robustness and to ascertain their relative advantages [2]. This study confirmed the consistency and robustness of both approaches. More interestingly, it has been shown that the HMM approach is the one recommended whenever the vocabulary size to be recognized is small otherwise the DTW algorithm is rather the one recommended to be used. In this paper, we shall deal with a huge vocabulary such as an Arabic printed documents library, and consequently we shall focus solely on the DTW algorithm.

DTW algorithm is a well known procedure especially in pattern recognition, [12, 25, 13, 24, 9]. The DTW algorithm is the result of the adaptation of Dynamic Programming (DP) to the field of pattern recognition. DP is a very efficient technique introduced by Bellman [7] to solve mathematical problems arising from multistage decision processes. It has since been used to solve wide range of decision problems in computer science. DP is both mathematically sound and computationally efficient in optimal path finding problems since it often drastically reduces the amount of enumeration by avoiding the enumeration of some decision sequences which cannot possibly be optimal.

The adaptation of the DTW algorithm to the printed Arabic OCR provides very interesting recognition and segmentation rates. In our previous works [26, 24, 3], we have conducted several experiments on around 20000 Arabic words randomly chosen from high and medium quality documents. Obtained results show that the recognition rate average is more than 97% and the segmentation rate average is more than 98%. Moreover, we found in particular that the recognition rate (respectively the segmentation rate) increases with the size of the text font used.

Unfortunately, the complex computing of the DTW procedure restricts its utilisation.

Many researchers attempted to solve this problem in proposing several approaches [12, 11, 31, 28, 30, 27, 4, 21, 23]. Nevertheless, such proposals generally require some very specific high cost architectures. In our previous works [23], we found that even loosely coupled architectures such as a grape of computers within a Local Area Network (LAN) can provide without any additional cost an appropriate infrastructure to distribute the complexity of the DTW algorithm.

In this paper, we propose a novel approach to distribute the Arabic OCR based on the DTW algorithm among loosely coupled computers. We develop its analytical and evaluate its performance. Experimental results prove the validity of the derived analytical results. The paper is organized as follows: section 2 describes the details of the Arabic printed cursive OCR based on the DTW algorithm. An overview on loosely coupled architectures is presented in section 3. The details of the proposed approach and the corresponding performance analysis are described and investigated in section 4. Conclusion remarks and future work are presented in section 5.

2. Mechanism of the Arabic OCR Based on the DTW Algorithm

Words in Arabic are inherently written in blocks of connected characters. A prior segmentation of these blocks into separate characters may consequently be needed. Indeed, many proposals considered the segmentation of Arabic words into isolated characters before performing the recognition phase [5, 13, 11]. The viability of using the DTW technique stems from its ability and efficiency to perform the recognition without requiring any prior segmentation [26, 24, 21, 23].

We consider in this paper a reference library of R trained characters forming the Arabic alphabet in some given fonts, and denoted by C_r: r=1, 2, …, R. The technique consists to use the DTW pattern method to match an input character against the reference library. The input character is thus recognized as the reference character that provides the best time alignment, namely character A is recognized to be C_k if the summation distance S_k corresponding to the matching of A to reference character C_k satisfies the following equation [25, 26, 24].

$$S_k = \min_{1 \le r \le R} \{S_r\} \tag{1}$$

Let T constitutes a given connected sequence of Arabic characters to be recognized. T is then composed of a sequence of N feature vectors T_i that are actually representing the concatenation of some sub sequences of feature vectors representing each an unknown character to be recognized. As portrayed on Figure 1 text T lies on the time axis (the X-axis) in such a manner that feature vector T_i is at time i on this axis. The reference library is portrayed on the Y-axis, where reference character C_r is of length l_r, $1 \le r \le R$. Let $S(i, j, r)$ represents the cumulative distance at point (i, j) relative to reference character C_r. The objective here is to detect simultaneously and dynamically the number of characters composing T and recognizing these characters. There surely exists a number k and indices (m_1, m_2, ..., m_k) such that $Cm_1 \oplus Cm_2 \oplus \ldots \oplus Cm_k$ represents the optimal alignment to text T where $\oplus$ denotes the concatenation operation. The path warping from point $(1, 1, m_1)$ to point (N, l_{mk}, k) and representing the optimal alignment is therefore of minimum cumulative distance that is:

$$S(N, l_{mk}, k) = \min_{1 \le r \le R} \{S(N, l_r, r)\} \tag{2}$$

this path, however, is not continuous since it spans many different characters in the distance matrix. We therefore must allow at any time the transition from the end of one reference character to the beginning of another reference character. The end of reference character C_r is first reached whenever the warping function reaches point (i, l_r, r), $i = \left\lceil \frac{lr+1}{2} \right\rceil, \ldots, N$. As we can see from Figure.1, the end of reference characters C_1, C_2, C_3 are first reached at time 3, 4, 3 respectively. The end points of reference characters are shown on Figure1 insides diamonds and points at which transitions occur are within circle. From these times on and up until time N. The warping function always reaches the ends of the reference characters. At each time i, we allow the start of the warping function at the beginning of each reference character along with addition of the smallest cumulative distance of the end points found at time $(i - 1)$. The resulting functional equations are:

$$S(i, j, r) = D(i, j, r) + \min_{\substack{1 \le i \le N \\ 1 \le j \le l_r \\ 1 \le r \le R}} \begin{Bmatrix} S(i-1, j, r), \\ S(i-1, j-1, r), \\ S(i-1, j-2, r) \end{Bmatrix} \tag{3}$$

with the boundary conditions :

$$S(i, 1, r) = D(i, 1, r) + \min_{\substack{1 + \left\lceil \frac{lr+1}{2} \right\rceil \le i \le N \\ 1 \le j \le l_r \\ 1 \le r \le R}} S(i-1, l_k, k) \tag{4}$$

To trace back the warping function and the optimal alignment path, we have to memorize the transition times among reference characters. This can easily be accomplished by the following procedure:

$$b(i,j,r) = \underset{\substack{1\le i\le N\\ 1\le j\le l_r\\ 1\le r\le R}}{\text{trace min}} \left\{ \begin{array}{c} b(i-1, j, r), \\ b(i-1, j-1, r), \\ b(i-1, j-2, r) \end{array} \right\} \tag{5}$$

Where *trace min* is a function that returns the element corresponding to the term that minimizes the functional equations. The functioning of this algorithm is portrayed on Figure 1 by means of the two vectors *VecA* et *VecB*, where *VecB*(i) represents the reference character giving the least cumulative distance at time *i*, and *VecA*(i) provides the link to the start of this reference character in the text *T* . The heavy marked path through the distance matrix represents the optimal alignment of text *T* to the reference library. We observe that the text is recognized as $C_1 \oplus C_3$.

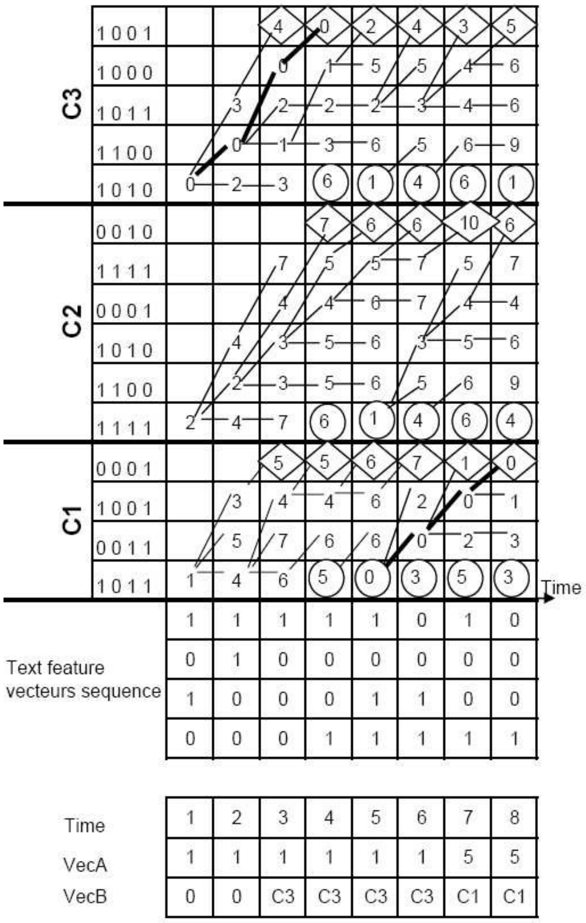

Figure 1. The DTW warping mechanism.

3. Loosely Coupled Architectures

Nowadays, loosely coupled architectures can be regrouped into grapes or clusters and grid computing. What makes attractive these architectures is their ability to provide, without any additional cost, enough computing power that can be used to several purposes especially to speedup some processing greedy algorithms. A grape is a set of computers interconnected together over a high capacity network but with low latency [39]. These interconnected computers can be homogeneous or heterogeneous in terms of architectures and operating systems. A LAN (Local Area Network) is a typical example. A grid computing, however, is a hardware and software infrastructure that can federate (interconnect) over communication networks several heterogeneous resources belonging to different institutions (universities, governmental administrations …).

A grid computing provides a transparent accessibility to the different interconnected resources [18, 6, 19, 10, 32, 17]. The idea behind the grid computing concept or architecture has several motivations. Among these motivation is making easy and transparent the sharing of resources between institutions. Two other important motivations must be reported here; the first one concerns the federation of the computing capacity of the interconnected computers to provide enough power to properly achieve:

- Distributed Supercomputing.
- High-Throughput Computing.
- On-Demand Computing.

The second one concerns the federation of the storage capacity of the interconnected computers to provide enough storage space to achieve properly Data intensive Computing.

There are only two differences between a grape and a grid computing. The first one concerns the localisation of the interconnected computers. In a grape, the interconnected computers and resources belong to the same institution. However, in a grid computing the interconnected computers and resources belong to several institutions. The second one concerns the number of interconnected computers and resources which is generally bigger in grid computing than in grapes.

4. Distributed Arabic OCR

The Arabic OCR based on the DTW procedure described in the preceding section presents many ways on which one could base its parallelization or distribution. The idea of the proposed approach is how to take advantages of the enough power provided by a grape or a grid computing to speedup the execution time of the DTW algorithm? Moreover, it is well known that in most of the cases, organisations computer networks (grapes) are generally underutilized in terms of computing power [10].

Besides, the flow distribution of the algorithm itself induces a big amount of communication and synchronization between the target computers [27]. However, a good data distribution can consequently avoid or/and reduce these induced amounts of communication and synchronization [8]. Conducted experiments show that a substantial reduction in communication costs and synchronization is obtained when each processor or computer participating in the work achieves the recognition process on a separate part of the input Arabic text (or texts) to be recognized. In this case, no synchronisation is needed. We consider in this paper this approach of text distribution among participating processors. The input binary image of the Arabic text to be recognized is first split and assigned to the participating processors. In this case, every computer participating in the work has to achieve the three following tasks. The first one deals with the pre-processing of this raw data and mainly concerns filtering the sub image, framing and positioning and the segmentation of the text into groups of connected (cursive) characters. The second task concerns the description and feature extraction, and hence the determination of the characteristic fragments of every character or group of connected (cursive) characters to be recognized so that a certain combination of characteristic fragments can be assigned with adequate confidence by the decision process to a recognized class. The third one concerns the recognition process of these obtained entities by the DTW algorithm. We will suppose next that:

- We own a big data base of binary images of printed Arabic texts (documents) to be recognized. The size of these binary images which are stored in files is variable;
- the assignment process of these binary files among the target computers will be achieved by a computer named the coordinator (the master) which will be also responsible of the user interface
- The target computers participating in the work and the coordinator belong to the same loosely architecture which can be a grape or a grid computing;
- These computers can be homogenous or heterogeneous in terms of computing capacity;
- The time required by every computer participating in the work to achieve the second and the third tasks previously described are rather negligible;

4.1. The Analytical Study

Significant reductions in the elapsed time, defined as the time elapsing from the start until the completion of the text recognition, can be realized by using a distributed architecture. This effect is known as the speedup factor. This factor is properly defined as the ratio of the elapsed time using sequential mode with just one processor to the elapsed time using the distributed architecture.

The efficiency is another factor commonly used to assess the performance of such distribution. This factor expresses the percentage of the computing yield of the distributed architecture. The efficiency factor is defined here as the ratio of the speedup factor to the number of computers participating in the work.

We denote by P the total number of computers participating in the work. T_p the total number of feature vectors composing the input binary sub image (or images) to be assigned by the coordinator to computer p where $(p=1, 2, ..., P)$. T_p is in fact composed of a set of characters and groups of connected (cursive) characters. X: the total number of feature vectors composing the input binary image of the Arabic text (or texts) to be recognized. We readily have:

$$X = \sum_{p=1}^{P} T_p \tag{6}$$

We suppose next that the size of each assigned T_p is given and decided by an optimal or a pseudo optimal assignment algorithm. The discussion of such an algorithm is beyond the scope of the current paper.

Let b denotes the total number of machine instructions required to execute at time i all the cumulative distances. Consequently, the total number of machine instructions denoted by $Minst$ required to compute the X feature vectors of the entire Arabic text (or texts) to be recognized is readily given by the following equation:

$$Minst = b \times X \tag{7}$$

When the recognition process of the Arabic texts is achieved sequentially, no planning or coordination (coordination includes communication and synchronization) are neither needed nor used. The elapsed time in such a case corresponds just to the execution time of all these input texts. Consequently, the completion time denoted by: $CT\ (1)$ is then expressed as the ratio of $Minst$ to the computing capacity, denoted by σ_0, of the computer used in the sequential mode. Thus, $CT\ (1)$ will be expressed as follows:

$$CT(1) = \frac{Minst}{\sigma_0} \tag{8}$$

In the distributed mode however, we need both planning and coordination among the computers participating in the work. Thus, the completion time, denoted by $CT\ (P)$ depends on the number P of computers used. Let σ_C denotes the computing power of the coordinator. We will suppose next that σ_C is infinitely rapid. Let σ_p denotes the computing power of computer p, where $p = 1, 2... P$. Let γ denotes the size in bytes of any feature vector of T_p. Let $NTPac_p$ denotes the number of packets where $(p=1, 2... P)$ to be transmitted from the coordinator to computer p during the assignment process. $NRPac_p$ is readily expressed as follows:

$$NTPac_p = \frac{T_p \times \gamma}{Pac} \tag{9}$$

where *Pac* denotes the size in bytes of any given packet.

Let C_p denotes the number of recognized characters from within T_p. Let $NRPac_p$ denotes the number of packets to be returned by computer p to the coordinator upon finishing its recognition process. Thus, $NRPac_p$ is readily expressed as follows:

$$NRPac_p = \frac{C_p}{Pac} \qquad (10)$$

We denote by T^* the recognition processing time of the slowest computer among those participating in the work. T^* is consequently expressed as follows:

$$T^* = \max_{1 \le p \le P} \left\{ \frac{b \times T_p}{\sigma_p} \right\} \qquad (11)$$

Let $\alpha_1(P)$, denotes the planning time among the P computers participating in the work. This task is achieved by the coordinator, it corresponds mainly, to the total time required to compute optimally or pseudo optimally all the T_p to be assigned to the different computers; $\alpha_1(P)$ depends directly on the complexity of the algorithm used to achieve this task.

Let $\alpha_2(P)$ denotes the coordination time among the P computers participating in the work. In fact, we may approximate $\alpha_2(P)$ by the following expression:

$$\alpha_2(P) = \sum_{p=1}^{P} (NTPac_p + NRPac_p) \times cap \qquad (12)$$

where *cap* expresses the capacity of the network of the loosely coupled architecture used. The completion time *CT(P)* will be consequently expressed as follows:

$$CT(P) = T^* + \alpha_1(P) + \alpha_2(P) \qquad (13)$$

The expressions of the speedup and efficiency factors are then given by:

$$S(P) = \frac{CT(1)}{CT(P)} \qquad (14)$$

$$E(P) = \frac{S(P)}{P} \qquad (15)$$

Equation 13 shows that $\alpha_1(P)$ and $\alpha_2(P)$ are key parameters of our distribution. We will suppose next that $\alpha_1(P)$ is negligible compared to $\alpha_2(P)$ especially if we use a non complex (light) heuristic to compute in a pseudo optimally manner all the T_p.

4.1.1. Numerical Results

We consider next only the case where the loosely coupled architecture is homogeneous. It means that all the corresponding interconnected computers are homogeneous in terms of computing power and operating system. We ran several experiments on several specific printed Arabic texts that we scanned by an HP scanner with a resolution of 300 dots per inch. Obtained statistics show in particular that:

- The average number of lines per page is 40;
- The average number of characters per line is 70;
- The total size of interlines and spacing characters is approximately equal to 1/5 of the total size of characters composing the studied corpus;
- The average number of feature vectors in one Arabic character is 25 each of which has the dimension 50.

In light of these statistics, we have made the following experiments:

The First Experiment

This experiment aims to study the effect of the distribution of approximately a thousand (*1000*) of printed Arabic text pages over a variable number of target computers when the network capacity is fixed. Consequently, the following data has been considered:

- X = 70000000, the total number of feature vectors composing the input Arabic texts (which are composed of around 1000 pages) to be recognized;
- The reference library *V* to be used is composed of 500 isolated characters. Each Arabic character is approximately represented by 5 different occurrences mutually rpresentig a small difference (including some noises). This value has been found after several conducted experiments.
- b =5000000, the total number of machine instructions required to execute at time *i* all cumulative distances. This is evaluated through extensive experiments.
- σ_p = 1000 Mega Machine Instructions Per Second (MIPS) the computing power of computer *p* where (*p=1, 2... P*) Participating in the work. This value corresponds approximately to the computing power average of our Lab computers.
- P the total number of computers that will participate in the work. This number lies from 1 to 1000.
- σ_0 = 1000 MIPS. It means that the computer used in the sequential mode has the same computing power as any one of those used in the distributed mode.
- cap = 100 Kilo Bytes Per Second (KBS) the capacity of the network of the loosely architecture used. This capacity corresponds approximately to the effective average throughput of our lab LAN.

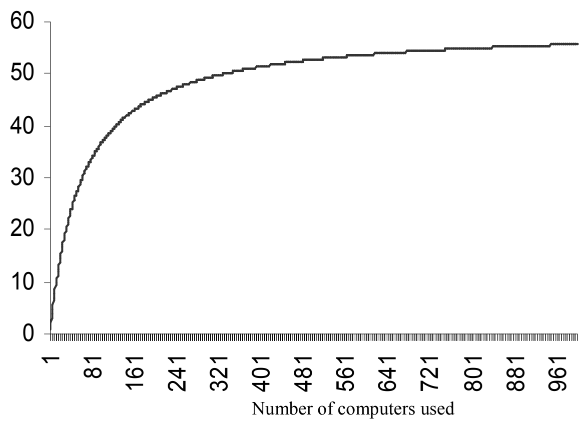

Figure 2. Speedup factor of the first experiment.

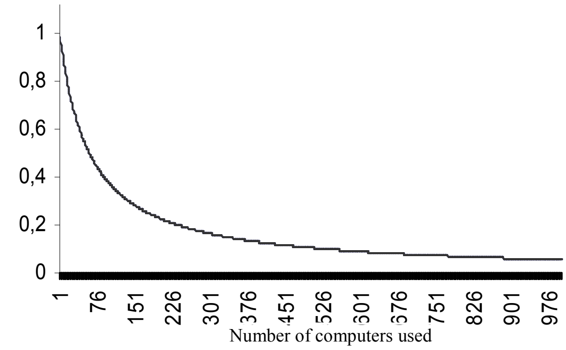

Figure 3. Efficiency factor of the first experiment.

Figures 2 and 3 illustrate the obtained results of this first experiment. These figures show in particular that:

- The speedup factor increases with the number of computers used.
- However, the efficiency factor decreases with the number of computers used.
- If we use 60 computers then the speedup factor reaches the value 30 and the corresponding efficiency factor reaches the value 0.5. This result is very interesting because in this case our proposed OCR system will be able to recognise more than 900 characters per second which is far away from the recognition rates of the existing systems [14]. Besides, the computers participating in the work will use only the half of their computing power. It means that users of this architecture can use the remaining half computing power for other purposes.
- The limiting speedup is around 57 which are obtained when the number of participating computers is above 400. The corresponding efficiency factor is around 0.18. In such a situation set up, the proposed OCR system will be able to recognise more than 1700 characters per second yet consuming only 18% of each participating computer. Using more computers does not provide better speed up; nevertheless it allows using less computing power per participating computer. This result is very interesting since it suggests that OCR systems when built on a loosely coupled architecture allows to have very high recognition rates while at the same time take away a rather small fraction of the processing power of any participating computer.

The Second Experiment

This second experiment aims to study the effect of the network capacity (*cap*) variation when the number of computers participating in the work is fixed and the size of the text to be recognized is fixed too.

We consider here the same values for all parameters as those used in the first experiment except the value of *P* which will be fixed to 60 and the value of cap which will lie between 5 KBS to 10 MBS (Mega Bytes per Second).

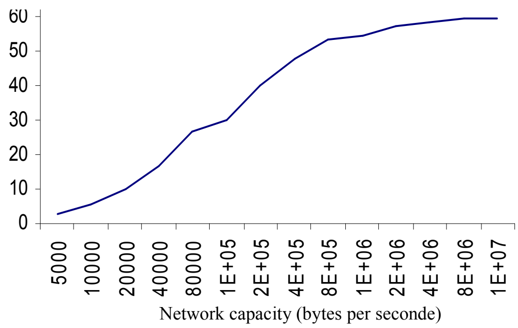

Figure 4. Speedup factor of the second experiment.

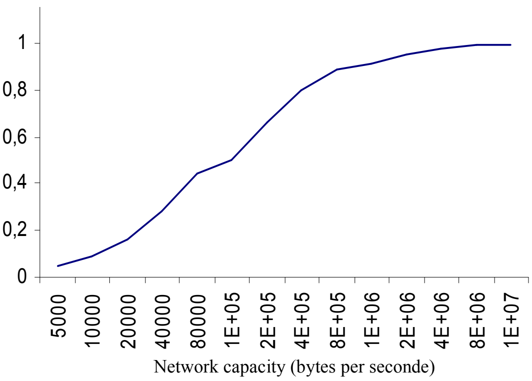

Figure 5. Efficiency factor of the second experiment.

Figures 4 and 5 illustrate the obtained results of this second experiment. These figures portray in particular that:

- The speedup factor increases with the network capacity.
- The efficiency factor increases also with the network capacity.
- For a heavily loaded network corresponding to small values of *cap*, our proposed system still delivers adequate recognition rates. For instance, when the capacity *cap* = 40 KBS the speedup factor reaches the value 17 which amounts to a recognition rate of about 500 characters per second.
- For a lightly loaded network corresponding to large values of *cap*, the speed up is large enough for the system to reach excellent recognition rates. For instance, when *cap* = 1 MBS then the speedup factor reaches the value 54.5 and our OCR system

will be able to recognize more than 1600 characters per second.

Consequently, the proposed approach is very interesting because it can lead easily to the implementation of a very powerful Arabic OCR system.

4.2. The Experimental Study

To validate the results given by our analytical model and ascertain their reliability, we conducted an experimental study on the Scientific Research Tunisian Grid (SRTG) which was recently implemented and tested [16]. Our goal here is mainly to put in practice our approach, yet to demonstrate the obtained benefits. This grid is similar to the XtremWeb-CH [35]. XtremWeb-CH is an improved version of XtremWeb [35]. It interconnects via the Internet several heterogeneous computers of different institutions. The main computer (the middleware) of the SRTG is named the coordinator and the voluntary interconnected computers are named workers. To distribute an application over the SRTG, users have to follow these steps:

- Logon to the GTRS.
- Request the coordinator about the number of interconnected workers and their corresponding computing power and operating systems.
- Make a decision about the number of workers that will participate in the work.
- Prepare an XML file describing the distributed application to be processed [1]. This file includes the physical access paths of the data to be processed (the binary sub images) and the code to be executed by every designated worker. It means here that the optimal or pseudo optimal assignment of the input data is achieved by the user and not the coordinator.
- Upload this file to the coordinator.

After receiving this file, the coordinator assigns to each involved worker the input data and the code to be executed as described in the received XML file. Whenever, the distributed processing is terminated the coordinator must first sort in the right order the returned recognition results from workers and put them in one organized file. Finally, it turns back this file to the user. Our experiment aims to implement the proposed approach and to prove that the speedup factor increases with the number of workers used. We have considered the following conditions:

- The studied application was implemented in the “C sharp” language.
- We have used 20 dedicated homogeneous workers having the exact same configuration: 3GHZ CPU frequency, 512 Mega Octets RAM and running Windows XP-professional.
- We have used a text corpus formed of 7000 Arabic words randomly chosen which were scanned using an HP scanner with a resolution of 300 dpi (dots per inch).
- We have considered also a reference library composed of 103 characters representing approximately the totality of the Arabic alphabet (including the characters shape variation according to their position within words).
- The grid network capacity was around 100KBs.
- The XML file has been generated manually.

Figures 6 and 7 illustrate the obtained results of the described experiment.

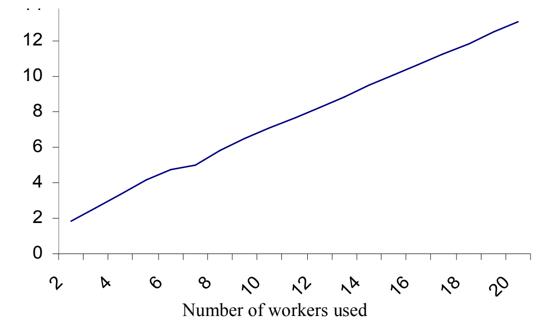

Figure 6. Speedup factor of the experimental study.

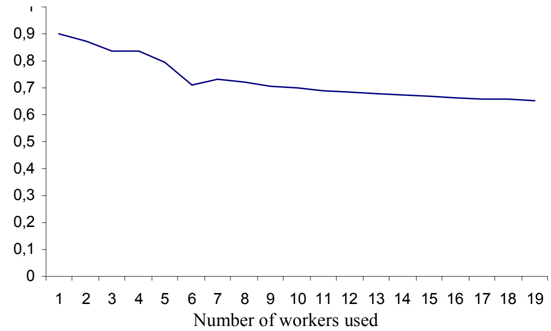

Figure 7. Efficiency factor of the experimental study.

These figures show in particular that:

- The speedup factor increases with the number of workers used.
- The efficiency factor is greater than 0.6 which means that the computing power of each dedicated worker is used for more than 60%. Using more workers will certainly lessen their efficiency and hence we may use non dedicated workers.
- If we use 20 workers then the speedup factor reaches the value 13 which amounts to a recognition rate around 500 characters per second which is a very interesting recognition speed compared to the existing products [14].

5. Conclusion and Perspective

We proposed in this paper a distributed printed Arabic OCR based on the DTW algorithm. What makes attractive the proposed distribution is the target hardware infrastructure that can support it and which can be any loosely coupled architecture such as a grape or a grid computing. We proposed an analytical performance study that allowed us to quantify the speed up and the efficiency factors obtained for different system parameter values. An experimental study is also conducted on an existing grid plate form to validate our analytical results and to confirm the benefits emanating from our proposed approach. The proposed approach allows to reach very substantial recognition rates that could not been attained otherwise. Further, investigations are underway to validate the proposed analytical model through extensive use of the existing GTRS grid plate form.

References

[1] Abdennadher N., "Vers Un Outil Peer-To-Peer Orienté Calcul Intensif," *Computer Journal of Flash Informatique EPFL*, vol. 3897, no. 1, pp. 291-306, 2005.

[2] Abdi N., "Reconnaissance Automatique de L'écriture Imprimée Arabe," *Thesis Dissertation*, Tunisia, 2004.

[3] Abedi N. and Khemakhem M., "Reconnaissance de Caractères Imprimés Cursifs Arabes Par Comparaison Dynamique et Modèle Caché De Markov, " *in Proceedings of Gen Ed Institute*, Tunisia, pp. 56-63, 2004.

[4] Alves C., "Parallel Dynamic Programming for Solving the String Editing Problem on CGM/BSP," *in Proceedings of SPAA*, Canada, pp. 20-29, 2002.

[5] Amin A., "Off-Line Arabic Character Recognition: The State of the Art," *Pattern Recognition Computer Journal,* vol. 31, no. 5, pp. 517-530, 1998.

[6] Apache Xerces, http ://www. globus .org, 2003.

[7] Bellman R., *Dynamic Programming*, New Jersey, UK, 1975.

[8] Ben Amara N., "A Relational Database for Arabic OCR System," *The International Arab Journal of Information Technology (IAJIT)*, vol. 2, no. 4, pp. 206-212, 2005.

[9] Bridle J., "An Algorithm for Connected Word Recognition," *in Proceedings of IEEE International Conference on Acoustics Speech and Signal*, France, pp. 899-902, 1982.

[10] Buyya R., "A Gentle Introduction to Grid Computing and Technologies," *in Proceedings of CSI*, India, pp. 733-738, 2005.

[11] Cheng H., "A VLSI Architecture for Dynamic Time-Wrap Recognition of Handwritten Symbols," *Computer Journal of IEEE Anti Spam SMTP Proxy*, vol. 34, no .3, pp. 1010-1011, 1986.

[12] Cheng H., "VLSI Architecture for Pattern Matching Using Space-Time Domain Expansion Approach," *in Proceedings of IEEE International Conference on Computer Design VLSI and Computing*, New York, pp. 46 - 50, 1985.

[13] Cheung A., "An Arabic Optical Character Recognition System Using Recognition Based Segmentation," *Pattern Recognition Computer Journal*, vol. 34, no. 2, pp. 217-236, 2001.

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

[15] Dehghan M., "Handwritten Farsi (Arabic) Word Recognition: A Holistic Using Discrete HMM," *Pattern Recognition Computer Journal*, vol. 34, no. 5, pp. 1057-1065, 2001.

[16] ESSTT, http://www.esstt.rnu.tn/utic/gtrs/, 2006.

[17] Foster I. and Kesselman C., *The Grid: Bluepoint for a Future Computing Infrastructure*, Morgan Kaufmann, San Francisco, 1999.

[18] Foster I., Kesselman C., and Tuecke S., "The Anatomy of the Grid," *International Journal of Supercomputer Applications*, vol. 16, no. 2, pp. 115-128, 2002.

[19] IBM, *Introduction to Grid Computing with Globus*, IBM Redbook, 2003.

[20] Kanoun S., "Reconnaissance D'images de Textes Arabes Par Approche Affixale," *in Proceedings of MCSEAI*, Tunisia, pp. 25-36, 2004.

[21] Khemakhem M. and Belghith A., "A Multipurpose Multi-Agent System Based on a Loosely Coupled Architecture to Speedup the DTW Algorithm for Arabic Printed Cursive OCR," *in Proceedings of IEEE International Conference on Computer Systems and Applications*, Egypt, pp. 301-311, 2005.

[22] Khemakhem M. and Belghith A., "Agent Based Architecture for Parallel and Distributed Complex Information Processing," *Computer Journal of International Review on Computers and Software*, vol. 2, no. 1, pp. 25-29, 2007.

[23] Khemakhem M. and Belghith A., "The DTW Algorithm for Distributed Printed Cursive OCR within A Multi Agent System," *in Proceedings of ACM*, Egypt, pp. 603- 611, 2007.

[24] Khemakhem M., "Arabic Type Written Character Recognition Using Dynamic Comparison," *in Proceedings of 1st Computer Conference*, Kuwait, pp. 109-118, 1989.

[25] Khemakhem M., "Reconnaissance De Caractères Imprimés Par Comparaison Dynamique," *in Proceedings of AFCET*, France, pp. 147-161, 1987.

[26] Khemakhem M., "Reconnaissance Globale de Caractères Imprimés Arabes et Latins Par Comparaison Dynamique," *in Proceedings Regional Conference on Computer Science and Arabization*, Tunisia, pp. 28-31, 1988.

[27] Khemakhem M., Belghith A., and Ben Ahmed M., "Modélisation Architecturale de la Comparaison Dynamique Distribuée," *in Proceedings in Second International Congress*

on Arabic and Advanced Computer Technology, Morocco, pp. 21-40,1993.

[28] Khemakhem M., Belghith A., and Ben Ahmed M., "Etude et Evaluation de Deux Méthodes de Distribution de L'algorithme de Comparaison Dynamique Pour La Reconnaissance de Caractères Arabes," *in Proceedings of First Maghrebin Symposium on Programming and Systems*, Algeria, pp. 110-117, 1991.

[29] Kumar A., "Model-Based Annotation of Online Handwritten Datasets," *International Institute of Information Technology*, India, pp. 53-60, 2006.

[30] Philip G., "Efficient Parallel Dynamic Programming," *in Preceding the 30th Annual Allerton Conference on Communication Control and Computing*, University of Illinois, pp. 185-194, 1992.

[31] Quénot G., "A Dynamic Programming Processor for Speech Recognition," *IEEE Journal of Solid State Circuits*, vol. 24, no. 9, pp. 338-348, 1989.

[32] Shi Z., "Agent Based Grid Computing," *Applied Mathematical Modelling*, vol. 30, no. 7, pp. 629-640, 2006.

[33] Tapia E., "A Survey on Recognition of on Line Handwritten Mathematical Notation," *Technical Report*, Germany, 2007.

[34] Vuori V., "Experiments with Adaptation Strategies for a Prototype-Based Recognition System for Isolated Handwritten Characters," *Computer Journal of International Journal of Document Analysis and Recognition*, vol.3, no. 3, pp. 150-159, 2001.

[35] Xtremweb-ch, http://www.xtremwebch.net, 2000.

Maher Khemakhem received his master of science and his PhD degrees from the University of Paris 11, France in 1984 and 1987, respectively. He is currently assistant professor in computer science at the Faculty of Economy and Management Sciences at the University of Sfax, Tunisia. His research interests include distributed systems, performance evaluation, and pattern recognition.

Abdelfettah Belghith received his master of science and his PhD degrees from the University of California at Los Angeles in 1982 and 1987, respectively. He is since 1992 a full professor at the National School of Computer Science, University of Mannouba, Tunisia. His research interests include computer networks, wireless networks, multimedia Internet, mobile computing, distributed algorithms, simulation, and performance evaluation.