*International Journal of Computer Sciences and Engineering Systems*
*Vol. 10 No. 1 (June, 2016)*

# The DTW Data Distribution over a Grid Computing Architecture

**Maher KHEMAKHEM[1], Abdelfattah BELGHITH[2] and Mohamed LABIDI[3]**

[1]MIRACL Lab, FSEGS, University of Sfax BP 1088, 3018, Sfax, Tunisia
*E-mail: maher.khemakhem@fsegs.rnu.tn*

[2]CRISTAL Lab, ENSI University of Manouba, Manouba, Tunisia
*E-mail: abdelfattah.belghith@ensi.rnu.tn*

[3]MIRACL Lab, FSEGS, University of SfaxBP 1088, 3018, Sfax, Tunisia
*E-mail: mohamedlabidi@yahoo.fr*

***Abstract:*** *In this paper; we show how grid computing can further speedup Dynamic Time Warping (DTW) algorithm. More specifically, we present experimental results of the data distribution of the Arabic printed cursive OCR based on the Dynamic Time Warping (DTW) algorithm over the Scientiûc Research Tunisian Grid (SRTG). In fact, the Arabic printed cursive OCR based on the DTW algorithm provides very interesting recognition and segmentation rates. Among the advantages of the DTW algorithm, is its ability to achieve prop-erly and simultaneously the recognition and the segmentation of connected or cursive characters from within a reference library of isolated characters. Unfortunately, the big amount of computing to be achieved during the recognition process makes the DTW execution time very slow and hence restricts its utilization. Conducted experiments show and conrm that grid computing presents a very interesting framework to speedup the DTW execution time. In addition, we found that the speedup and the efficiency factors increase with the size of the Arabic text.*

***Keywords:*** The DTW algorithm; Arabic printed cursive OCR; Grid Computing; Performance evaluation.

## 1. INTRODUCTION

Dynamic Time Warp (DTW) algorithm is a well known procedure especially in pattern recognition [31], [1], [13], [2], [3], [4], [12], [14], [32], [15], [8], [30], [29]. In fact, this procedure is the result of the adaptation of dynamic programming to the field of pattern recognition. The purpose of the DTW algorithm is to perform optimal time alignment between a reference pattern and an unknown pattern and evaluate their difference. Arabic printed cursive OCR (Optical Character Recogni-tion) based on the DTW algorithm provides very interesting recog-nition rates. Conducted experiments achieved on high and medium quality documents containing around 20000 Arabic words show that the recognition average rate is more than 98% and the segmentation average rate is more than 99%, [3], [4], [5], [8], [27]. Moreover, we found in particular that the recognition rate (respectively, the segmentation rate) increases when the font size increases. Furthermore, what makes the DTW algorithm very attractive is its ability to recognize and segment simulta-neously and properly Arabic printed cursive characters from within a reference library of isolated characters. Unfortunately, the underlying complex computing of this algorithm makes its execution time very slow and hence restricts its utilization.

Many solutions and approaches have been proposed to speedup the execution time of the DTW algorithm [1], [13], [12], [6], [14], [7], [15], [8], [27]. In this paper, we show how grid computing [26], [16] can further speedup the DTW algorithm. More specifically, we present experimental results of the data distribution of the DTW algorithm. Obtained results conrm that grid computing using SRTG provides indeed a very interesting infrastructure to speedup the execution time of the studied application.

In Section (2), we describe the mechanism of the Arabic printed cursive OCR based on the DTW algorithm. In Section (3), we give a brief overview on grid computing and the SRTG. The details of the distribution of the studied application over the SRTG and the corresponding performance evaluation are described and investigated in Section (4). Concluding remarks and future investigations are presented in section (5).

## 2. ARABIC OCR BASED ON THE DTW ALGORITHM

An OCR system is generally decomposed into four stages. The ûrst one concerns the acquisition of the text scanned image to be provided in the form of pixels or binary data. The second stage deals with the pre-processing of this raw data and mainly concerns ltering the scanned image, framing

and positioning and the segmentation of the text. The pre-processing measurement vectors are however a completely inadequate support for the decision process. This is the task of the third stage which concerns the description and feature extraction, and hence the determination of characteristic fragments of the character or the group of connected (cursive) characters to be recognized so that a certain combination of characteristic fragments can be assigned with adequate condence by the decision process to a recognized class. The nal stage forms the culminating point of the recognition process: the matter of decision on the correct classication of the unknown. What makes DTW an attractive algorithm to use in the recognition process is its ability to eliminate time differences between the characters or shapes to be recognized [5], [8], [29], [27]. Based on the dynamic programming path nding, DTW presents a computationally efcient algorithm to find the optimal time alignment between two occurrences of the same character and more generally between any two given forms. Consider two symbols *A* and *B* represented each by a sequence of feature vectors provided by stage three of the recognition system, namely:

$$A = a_1, a_2, ..., a_i, ... a_m \quad (1)$$

$$B = b_1, b_2, ..., b_j, ... b_n \quad (2)$$

where *m* and *n* are respectively the number of feature vectors composing *A* and *B*. As a measure of the difference between two feature vectors $a_i$ in *A* and $b_j$ in *B*, a distance *D(i, j)* is employed. Consequently, a matrix *D* of distances is formed. The optimal time alignment between the two symbols is represented by the optimal path lying between point *(1, 1)* and point (*m, n*) in *D* and warping through the matrix in such a way that for any point (*i, j*) the cumulative distance *S*(*i, j*) is minimum. A key operation for DTW is the computation of the summation distance *S* representing the cumulative distance *S*(*m, n*) at the end point (*m, n*) and dened as:

$$S = min \sum_{i=1}^{m} D(i, W(i)) \quad (3)$$

where *W*(*i*) represents the warping function (that gov-erns the way to find a significant optimal path in the matrix *D*) with *W*(1) = 1 and *W*(*m*) = *n*. The warping function is in fact a model of time axis (i.e., the axis of the symbol *A* to be recognized) uctuation in a character pattern. By using the continuity conditions [2], [3], [4]: $W(i+1) - W(i) = 0,1,2$ for each $1 \le i \le m$ we obtain the following functional equations on the cumulative distances:

$$S(i, j) = D(i, j) + \min_{\substack{1 \le i \le m \\ 1 \le j \le n}} \begin{Bmatrix} S(i-1, j), \\ S(i-1, j-1), \\ S(i-1, j-2) \end{Bmatrix} \quad (4)$$

With the initial condition:

$$S(1,1) = D(1,1) \quad (5)$$

We then have *S* = *S*(*m, n*). The warping function and the optimal alignment can be found by the backtracking technique.

### A. Isolated Characters Recognition

We consider 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 $C_k$.

$$S_k = \min_{1 \le r \le R} \{S_r\} \quad (6)$$

where $S_k$ is the summation distance corresponding to the matching of *A* with the reference character $C_k$.

### B. Cursive Characters Recognition

Words in many languages and especially Arabic are inherently written in blocks of connected characters. A prior segmentation of these blocks into separate characters may be consequently needed. Indeed many researchers have considered the segmentation of Arabic words into isolated characters before performing the recog-nition phase [10], [11], [9], [33]. We may therefore employ the isolated character recognition algorithm described in the pre-ceding section. The viability of the use of DTW technique, however, is its ability and efciency to perform the recognition without prior segmentation.

Let *T* constitutes a given connected sequence of Arabic characters to be recognized. *T* is then composed of a se-quence 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, the 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 the reference character $C_r$ is of length $l_r$, $1 \le r \le R$.

By abuse or extension of our earlier notation, we use *S(i,j,r)* to represent the cumulative distance at point *(i, j)* relative to the reference character $C_r$. The objective is to detect simultaneously and dynamically the number of characters composing the text *T* and also recognizing these characters.

There surely exists a number *k* and indices ($m_1$, $m_2$, ..., $m_k$) such that

$$Cm_1 \oplus Cm_2 \oplus \ldots \oplus Cm_k$$

repre-sents 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:

$$S(N, l_{mk}, k) = \min_{1 \le r \le R} \{S(N, l_r, r)\} \quad (7)$$

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 a new character. The end of reference character $C_r$ is rst 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 rst reached at time *3*, *4*, and *3* respectively. The end points of the reference characters are shown on Figure 1, inside diamonds and points at which transitions occur are represented within circles. 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:

$$S(i, j, r) = D(i, j, r) + \min_{\substack{1 \le i \le N \\ 1 \le j \le l_r \\ 1 \le r \le R}} \begin{Bmatrix} S(i-1, j, r), \\ S(i-1, j-1, r), \\ S(i-1, j-2, r) \end{Bmatrix} \quad (8)$$

with the boundary conditions:

$$S(i,1,r) = D(i,1,r) + \min_{\substack{1+\left\lceil \frac{lr+1}{2} \right\rceil \le i \le N \\ 1 \le j \le l_r \\ 1 \le r \le R}} S(i-1, l_k, k) \quad (9)$$

To trace back the warping function and retrieve the optimal align-ment path, we have to maintain the transition time (to be memorized) from one reference character to the others. This can easily be accomplished by the following procedure:

$$b(i, j, r) = trace \min_{\substack{1 \le i \le N \\ 1 \le j \le l_r \\ 1 \le r \le R}} \begin{Bmatrix} b(i-1, j, r), \\ b(i-1, j-1, r), \\ b(i-1, j-2, r) \end{Bmatrix} \quad (10)$$

Where the *trace min* function returns the element correspond-ing to the term that minimizes the functional equations. The functioning of this algorithm is portrayed on Figure. 1. through the two vectors *VecA* et *VecB* and 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 exemplifies the computing complexity induced by the DTW algorithm.

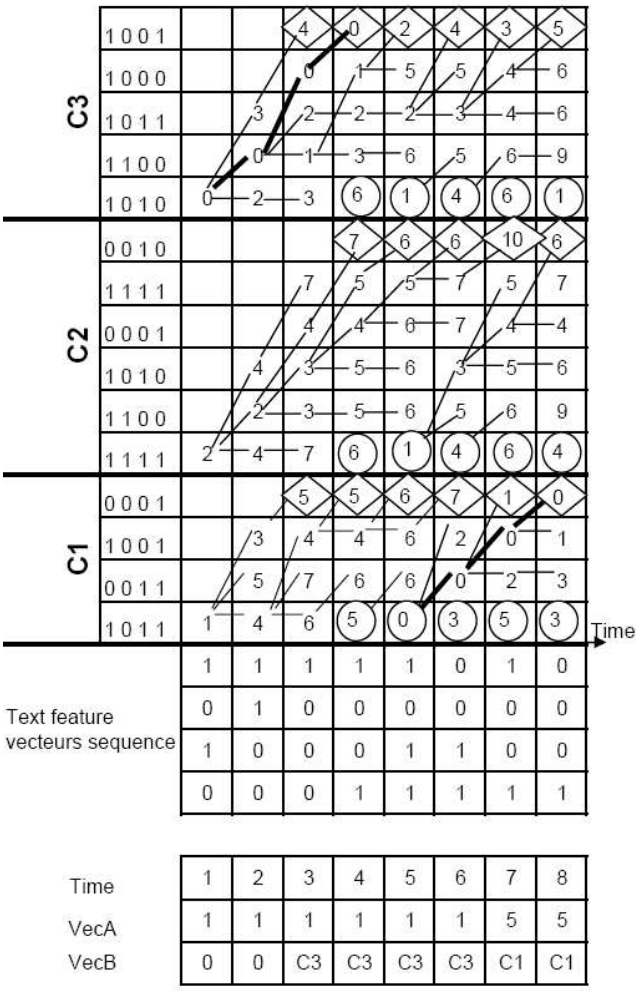


**Figure 1:** The DTW Mechanism.

## 3. GRID COMPUTING

A grid is a collection of machines, sometimes referred to as "nodes," "resources," "members," "donors," "clients," "hosts," "engines," and many other such terms. They all contribute and provide resources to the grid as a whole. Some resources may be used by all users of the grid while others may have specic restrictions [25], [24], [26], [16]. In most organizations, there are large amounts of underutilized computing resources. Most desktop machines are busy less than 5 percent of the time. In some organizations, even the server machines can often be relatively idle. Grid computing provides a framework for exploiting these underutilized resources for a better efciency of resource usage [24].

Often, machines may have enormous unused disk drive capacity. Grid computing, more specically, a "data grid", can be used to aggregate this unused storage into a much larger virtual data store, possibly congured to achieve improved performance and reliability over that of any single machine [24].

Consequently, a grid computing is an infrastructure that allows to many institutions (regardless their geographical

loca-tions) to interconnect a large collection of their heterogeneous computer networks and systems to share together a set of software and/or hardware resources, services, licences ... [23], [24], [26]. This huge ability of sharing resources in various combinations will lead to many advantages such as:

1. increase the efciency of resource usage;
2. facilitate the remote collaboration between institutions and researchers;
3. provide users with a huge computing power;
4. provide users with a huge storage capacity, etc.

### A. The Scientic Research Tunisian Grid (SRTG)

The Scientic Research Tunisian Grid (SRTG) is implemented by the research team UTIC [17]. It is similar to the XtremWeb-CH [18] which is an improved version of XtremWeb [19]. The main goal of the SRTG is to provide to Tunisian researchers an effective experimental framework to achieve their different needs such as the de-ployment of greedy applications and their corresponding performance evaluation.

## 4. THE DTW DATA DISTRIBUTION OVER THE SRTG

The data distribution adopted in this paper has been achieved according to the following steps:

- first of all, the input binary image or (images) of the text(s) to be recognized was (are) split into parts named binary sub images (or sub texts);
- and then, these obtained sub images have been assigned optimally among a set of target computers of the SRTG.

The SRTG is in fact distributed among several institutions using hetero-geneous computers interconnected through the Internet. One of these computers is termed the coordinator and the remaining ones are termed workers. The coordinator is responsible of the management of the OCR process and the coordination among workers. Moreover, it is working as a web server service, consequently, it can for example and instantaneously provide to users the number of available workers. Thus, if we need to launch on the SRTG an Arabic printed cursive OCR then we have rst to log in to the coordinator, ask it about the number, computing capacity and Operating Systems of available workers. Second, we have to fix the number and the specications of the target workers that will participate in the work. Finally, we have to prepare the different XML les describing the distributed application to be processed [21], [22]. These les which include all access paths of the data to be processed (the binary sub images) and the code to be executed by every designated worker, must be sent to the coordinator. After receiving these les, the coordinator assigns them among the target workers. We recall here that the target workers are designated by the coordinator however their number is fixed by the user.

Consequently, every worker participating in the work must achieve the following tasks:

- the pre-processing and segmentation of the binary sub image (they correspond to the second stage of an OCR system);
- the description and feature extraction of the segmented parts which correspond to connected characters (they correspond to the third stages of an OCR system);
- the recognition of these connected characters by the DTW algorithm.

After the processing of these tasks, every worker must turn back its results (recognized sub texts) to the coordinator. In the same manner, the coordinator must rst concatenate in the right order these obtained results into one le and then turns it back to the user (see Figure 2).

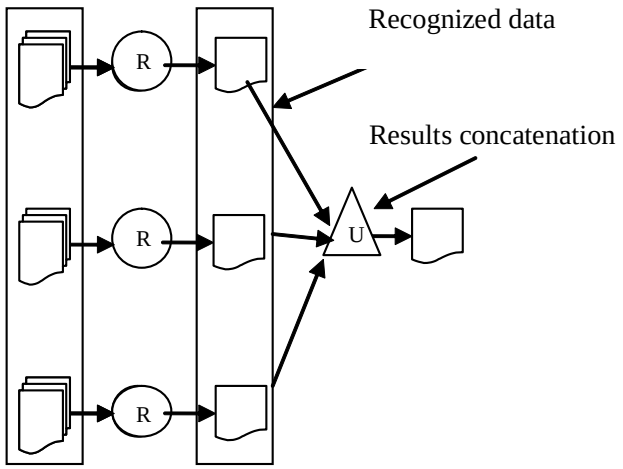


**Figure 2:** Data Distribution Workflow

### A. Experimental Study

We would like to remember that the SRTG was recently implemented and tested by the UTIC research team [17]. Conducted experiments have the following conditions:

- the studied application was implemented in “C sharp” language;
- to make easy the optimal assignment process of the input sub images among the target workers, we have considered in one hand two Arabic cursive texts corpus formed of 3500 and 7000 randomly chosen words which was scanned by a HP scanner with a resolution of 300 dpi (dots per inch). On the other hand, we have considered a set of *9* dedicated homogeneous target workers having the exact same congura-tion which was: *3GHZ* as CPU frequency, *512 Mega* Octets as RAM capacity and Windows XP-professional as operating system;
- the reference library used was composed of *103* charac-ters representing approximately the totality of the Arabic alphabet (including the characters shape variation according to their position in words);
- the grid networking capacity is around *100KBs*;

```
<Ordenancement isRelative="1" >
<Application applicationDescription="
RecARAB " Client=" Mohamed ">
<Module ModuleDescription="M0"
ModuleDirIn="/app/">
<Binary BinaryExecutable="M0"
BinarySubFolder="Windows"
BinaryOsName=" Windows "
BinaryCpuType="ix86" />
</Module>
<Module ModuleDescription="M1"
ModuleDirIn="/app/">
<Binary BinaryExecutable="M1"
BinarySubFolder=" Windows "
BinaryOsName=" Windows "
BinaryCpuType="ix86" />
</Module>
</Application>
<Table>
<Task Description="T0"
Application="RecARAB"
ApplicationModule=" R"
userName=" Mohamed" DirIn="/0/"
FileIn="dir.zip" Final="0">
<Input InputName="ImgText.bmp" />
< Input InputName =" ImgBib.bmp " />
< Input InputName =" fileAscii.txt" />
<Output OutputName="text1.txt" />
</Task>
<Task Description="T1"
Application="RecARAB "
ApplicationModule="R"
userName=" Mohamed " DirIn="/1/"
FileIn="dir.zip" Final="0">
<Input InputName="ImgText.bmp" />
< Input InputName =" ImgBib.bmp " />
< Input InputName =" fileAscii.txt " />
<Output OutputName="text2.txt" />
</Task>
<Task Description="T2"
Application="RecARAB "
ApplicationModule="R"
userName=" Mohamed " DirIn="/2/"
FileIn="dir.zip" Final="0">
<Input InputName="ImgText.bmp" />
< Input InputName =" ImgBib.bmp " />
< Input InputName =" fileAscii.txt " />
<Output OutputName="text3.txt" />
</Task>
<Task Description="T3"
Application="RecARAB "
ApplicationModule="U"
userName=" Mohamed " DirIn="/3/"
FileIn="dir.zip" Final="1">
<Input InputName=" text1.txt " />
<Input InputName=" text2.txt " />
<Input InputName=" text3.txt " />
<Output
OutputName="TexteResultat.txt" />
</Task>
</Table>
</Ordenancement>
```

**Figure 3:** An XML file Sample

- the XML le generation have been achieved manually.

To assess the performances and benets of the distribution of the studied appli-cation, we shall measure the gain in computing time when using the SRTG as opposed to just using one standalone worker. Formally, we define the speed factor as the ratio of the elapsed (execution) time using a sequential mode (a standalone worker) to the elapsed time using the grid computing (SRTG).

The efciency factor constitutes another factor which is commonly used to assess the performance of such deployment. It expresses the percentage of the CPU utilization of all workers participating in the work In our case, this factor is dened as the ratio of the speedup factor to the total number of workers participating in the work.

We recall that the elapsed time using a sequential mode has been computed on one of the 9 homoge-neous target workers of the SRTG previously presented.

Conducted experiments are illustrated by Figure.4 and Figure 5. We show in particular that:

- the speedup factor increases as the number of workers increases;
- the speedup factor increases with the size of the corpus;
- the efficiency factor increases with the size of the corpus too. These results are very interesting because users can govern easily the expected response time, speedup and efciency factors of their application just by playing on the task granularity (the size of the input binary image of the text) which conrms our previous analytical study [28];
- the efficiency factor is always $>0.58$, it means that more than 58% of the computing power of the workers participating in the work is used;
- in the second experiment, if we use 9 workers then the speedup factor reaches the value 6.7 which will lead to the recognition of more than 250 Arabic characters per second. This is a very interesting result given that currently commercialized systems have approximately the same speed but less recognition rate c, f., [20] compared to our approach especially for medium and low quality texts (documents).

Consequently, the SRTG is an interesting infrastructure to speedup the DTW algorithm and certainly other greedy algorithms.

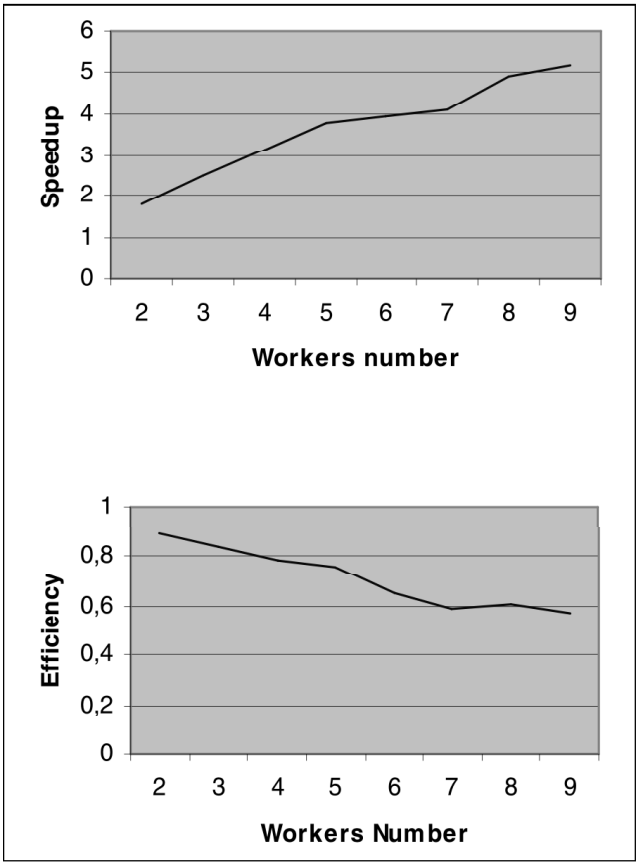


**Figure 4:** Case Where the Text Corpus is Formed of 3500 Words

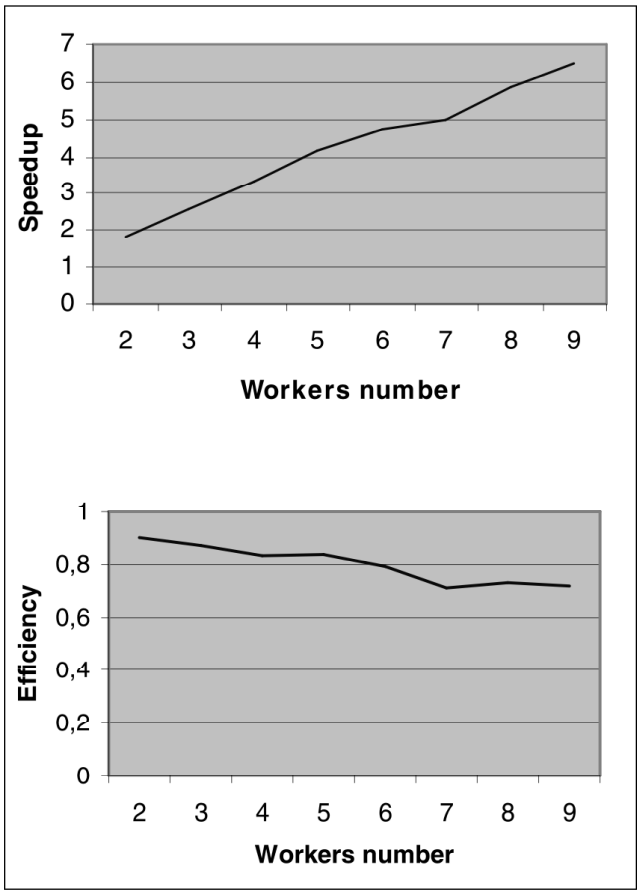


**Figure 5:** Case Where the Text Corpus is Formed of 7000 Words

## 5. CONCLUSION AND PERSPECTIVE

We presented in this paper the experimental results of the DTW data distribution trough the Arabic cursive OCR over a grid computing architecture. The objective of this distribution is to reduce the complex computing of the DTW algorithm. Performance evaluation of this distribution over the Scientic Research Tunisian Grid conrms that this infrastructure can provide an effective framework to speedup the DTW by interesting factor that can lead to a powerful Arabic OCR system. Furthermore, conducted experiments show that the number of workers to be used, the achievable speedup and efciency factors can be easily governed by the users.

Many investigations are under studies, especially the auto-matic optimization of the number of target workers that will participate in the work and the automatic generation of the XML file describing a given greedy application.

## REFERENCES

[1] H. D. Cheng *et al.*, VLSI Architecture for Pattern Matching using Space-time Domain Expansion Approach, *Proc. IEEE International Conference on Computer Design VLSI and Computing*., NY., Oct 7-10, 1985.

[2] M. Khemakhem *et al.* Reconnaissance de Caractères Imprimés Par Comparaison Dynamique, *Proc, AFCET*, Antibes, France, Sept., 1987.

[3] M. Khemakhem., Reconnaissance Globale de caractères imprimés arabes et latins par comparaison dynamique, *Proc, Regional Conference On Computer Science and Arabization*, Tunis, Tunisia, March 1988.

[4] M. Khemakhem *et al.*, Arabic Type Written Character Recognition Using Dynamic Comparison, *Proc. 1st Computer conference* Kuwait, March. 1989.

[5] N. Abedi, M. Khemakhem, Reconnaissance de caractères imprimés cursifs arabes par Comparaison dynamique et modèle caché de Markov *Proc. GEI2004*, Monastir, Tunisia, March 2004.

[6] M. Khemakhem, A. Belghith, M. Ben Ahmed, Etude et Evaluation de deux méthodes de distribution de l'algorithme de comparaison dynamique pour la reconnaissance de caractères arabes *Proc, First Maghrebin Symposium on Programming and Systems*, Algeria, October 1991.

[7] M. Khemakhem, A. Belghith, M. Ben Ahmed, Modélisation archi-tecturale de la Comparaison Dynamique distribuée *Proc, Second In-ternational Congress On Arabic and Advanced Computer Technology*, Casablanca, Morocco, December 1993.

[8] M. Khemakhem and A. Belghith., A Multipurpose Multi-Agent System based on a loosely coupled Architecture to speedup the DTW algorithm for Arabic printed cursive OCR. *Proc. IEEE-AICCSA-2005*, Cairo, Egypt, January 2005.

[9] S. Kanoun et al. Reconnaissance d'images de textes Arabes par approche Afxale, *Proc. MCSEAI'04*, Sousse, Tunisia, May 9-12, 2004.

[10] Amin Off-line Arabic character recognition: The state of the art, *Pattern Recognition*. Vol. 31, No, 5, 1998.

[11] A.Cheung *et al.* An Arabic optical character recognition system using recognition based segmentation, *Pattern Recognition*. Vol. 34, 2001.

[12] G. R. Quénot *et al.*, A Dynamic Programming Processor for Speech Recognition, *IEEE JSSC*, Vol. 24, N(F 9) q. 20, April 1989.

[13] Heng-Da Cheng *et al.*, A VLSI architecture for dynamic time-wrap recognition of handwritten symbols, *IEEE ASSP*, Vol. 34, N. 3, Jun 1986.

[14] Philip G. Bradford, Efficient Parallel Dynamic Programming, *Proc. the 30th Annual Allerton Conference on Communication, Control an Computing*, University of Illinois, 185-194, 1992.

[15] C. E. R. Alves *et al.*, Parallel Dynamic Programming for solving the String Editing Problem on CGM/BSP, *Proc. SPAA'02*, August 10-13, 2002 Winnipeg, Manitoba, Canada.

[16] Zhongzhi Shi *et al*. Agent Based Grid Computing, *Applied Mathematical Modelling* Vol. 30 pp. 629-640 2006 Available on www.sciencedirect.com

[17] Available at: *http://www.esstt.rnu.tn/utic/gtrs/*

[18] Available at: *http://www.xtremwebch.net*

[19] Available at: *http://www.xtremweb.net*

[20] CiyaICR product available on: http://www.ciyasoft.com/

[21] Nabil Abdennadher., XtremWeb-CH: Une plateforme Global Comput-ing pour les applications de haute performance, *Internal Report*, August 2004, HES-SO/ EIG.

[22] Nabil Abdennadher., Vers un outil Peer-To-Peer orienté calcul intensif, *Flash Informatique*, EPFL August 2005.

[23] Ian Foster, Carl Kesselman, and Steven Tuecke., The Anatomy of the Grid, *International Journal of Supercomputer Applications*, 2002.

[24] IBM., Introduction to Grid Computing with Globus *IBM Redbook*, SG24-6895-01 ISBN 0738427969, September 2003.

[25] http://www.globus.org

[26] Rajkumar Buyya *et al*. A Gentle introduction to Grid Computing and Technologies, *Proc. CSI*, India, May 7-19, 2005.

[27] M. Khemakhem and A. Belghith., The DTW Algorithm for Distributed Printed Cursive OCR within A Multi Agent System, *Proc, ACM, ICICIS* Cairo, Egypt, on March 14-18, 2007.

[28] M. Khemakhem and A. Belghith., Agent based architecture for Parallel and Distributed Complex Information Processing. January Issue of the *International Revue on Computers and Software*, Vol. 2 N°1., 2007.

[29] Ernesto Tapia *at al*. A Survey on Recognition of On-Line Handwrit-ten Mathematical Notation *Technical Report* B-07-01 Freie University at Berlin, Institut fur Informatik Takustr. 9, 14195 Berlin, Germany tapia,rojas@inf.fu-berlin.de, January 26, 2007.

[30] Anand Kumar *et al.* Model-Based Annotation of Online Handwritten Datasets, *International Institute of Information Technology*, Gachibowli, Hyderabad-500 032, India, 2006.

[31] J. S. Bridle *et al*. An Algorithm for Connected Word Recognition, *Proc. IEEE, ICASSP*, May 1982, pp. 899-902.

[32] V. Vuori *et al.* Experiments with adaptation strategies for a prototype-based recognition system for isolated handwritten characters, *IJDAR* Vol. 3 pp. 150–159, 2001.

[33] Najoua Ben Amara *et al.* A Relational database for Arabic OCR system, IAJIT Vol. 2, N° 4, Oct. 2005.