

DISTRIBUTED SIGNAL PROCESSING

J. Krupa, A. Pavelka, O. Vyšata, and A. Procházka

Institute of Chemical Technology, Department of Computing and Control Engineering

Abstract

The paper presents possibilities of distributed computing using the MATLAB distributed computing toolbox (DCT) and its application for multi-channel signal processing. The goal of the contribution is in description of parallel task processing through the computer network using cluster of computers and its use for electroencephalogram (EEG) signal analysis. The contribution presents the whole procedure consisting of tasks submission, their processing and collection of results. Comparison of results achieved using distributed processing is summarized in the final part of the paper as well.

1 Introduction

Distributed computing represents a very efficient tool to minimize computational time using either multi-core processing units or clusters of computers. The use of such systems becomes widely used now for extensive problems that can be separated into individual processing tasks.

The first part of the paper presents the general principle of distributed computing and its application in the MATLAB environment. The structure of the whole computational system for different applications is analysed in this section as well.

General steps that are necessary to prepare distributed tasks are presented in the main part of the paper. The whole procedure consisting of tasks submission, their processing and collection of results presented in Fig. 1 is analyzed in this section as well. The complete system forms an efficient computational environment together with specific monitoring tools enabling to follow distribution of tasks between the set of computers, processing time for individual tasks, number of tasks waiting in the queue and estimation of the total processing time.

The whole distributed computing toolbox cluster has been tested for the simple set of tasks to demonstrate its functionality and time consumption. Results achieved were then compared with those obtained for processing of the same problem using a single computer generally with a multi-core processor unit. DCT cluster has been then used for processing of the large set of EEG recordings to reduce the computing time necessary for their frequency analysis.

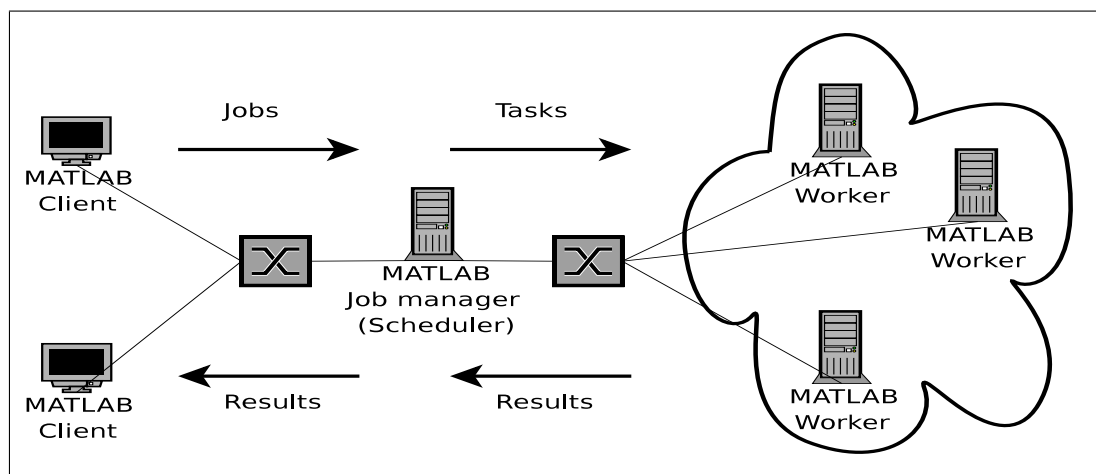


Figure 1: The principles and architecture of the MATLAB Distributed Computing Toolbox and the Distributed Computing Engine

2 Principles of Distributed Computing

Large extensive and time consuming problems can be often divided into smaller parts and processed separately. The MATLAB environment provides possibility to use the Distributed Computing Toolbox (DCT) for this purpose. This system is important for communication between the user's computer and the calculation cluster created by the MATLAB Distributed Computing Engine (DCE). The DCT and the MATLAB DCE enable to coordinate and execute independent MATLAB operations simultaneously on a cluster of computers, speeding up execution of large MATLAB jobs [2, 3].

In the terminology of DCT and DCE the *job* represents a complex problem that can be divided into small elements called *tasks*. Each *task* is evaluated by a *worker* which stand for a calculation session represented by a single computer or by its processor units in case of multi-core systems. All *jobs* and *tasks* are managed by the head node controlling computer called the *job manager*.

The optional *job manager* can run on any machine on the network according to Fig. 1. Individual *jobs* stand in the queue to be processed unless they are promoted, demoted, cancelled, or destroyed. Each *worker* receives a *task* of the running *job* from the *job manager*, executes the *task*, returns the result to the *job manager*, and then receives another *task*. When all *tasks* for a running *job* have been assigned to *workers*, the *job manager* starts running the next *job* with the next available *worker*.

A MATLAB DCE setup usually includes many *workers* that can execute *tasks* simultaneously, speeding up execution of large MATLAB *jobs*. It is generally not important which *worker* executes a specific *task*. Each *worker* evaluates *tasks* once a time, returning results to the *job manager*. The *job manager* then returns results of all *tasks* in the *job* to the client session.

A typical source code for distributed computing contain in general the following steps with an example of their programming in Fig. 2 and 3 for EEG signal analysis described later.

1. The search of the Job Manager and definition of a Job
2. Definition of individual Tasks
3. Submitting of a Job to the Job Queue in Job Manager
4. Retrieving the Jobs Results after the job is finished
5. Destroying of the Job

```
jm = findResource('jobmanager','name','puck_jobmanager');
job = createJob(jm);
set(job,'Name','EEGanalysis')
set(job,'RestartWorker',true)
dctool
```

Figure 2: Definition of job manager

```
for k=1:Tasks
    createTask(job, @EEGtask1, 1, ...
        TimeRange,WindowLength,FreqRange,chan,EEGfilek,bp,c);
end
submit(job); disp('Job submitted!')
waitForState(job)
results = getAllOutputArguments(job);
```

Figure 3: Definition of tasks, job submission and collection of distributed processing results

```

ftp_connection = ftp('hostname','user','password');
binary(ftp_connection);
cd(ftp_connection,'/');
mget(ftp_connection,EEGfile);
close(ftp_connection);

```

Figure 4: FTP file transfer

3 Monitoring of Cluster Calculations

Monitoring of distributed computing is very useful to follow the whole process and to visualize the progress of calculations. After the *job* is submitted to the *job manager* it is possible to close MATLAB session and without any problems it is possible to obtain current job status or results any time later. Command `jm = findResource('jobmanager');` is the most important one. Applying command `getjm` it is possible to receive structure array with all information stored by the *job manager*. For the higher comfort of *job* and *tasks* monitoring it is suitable to use shared files (see <http://www.mathworks.com/matlabcentral/fileexchange/>) in the MATLAB Central File Exchange. Functions available include

1. *dctool.m* - Graphical interface to display the status of a distributed computing network (Fig. 5) providing complete real-time monitoring of *jobs* and *tasks* and enabling to submit or destroy specific *job* or create *JobReport* (Fig. 6)
2. *listJobs.m* - Function used to displays a summary of *jobs* in the *job manager* using the handle of *job manager* as its input parameter
3. *listTasks.m* - Function used to displays the summary of *tasks* belonging to a *job* having the handle of *job* as its input parameter

There is a strong limitation of listed tools that is based on the network speed, number of *workers*, *jobs* and *tasks* in the *job manager*. The increasing number of those elements increase the response time of tools as well.

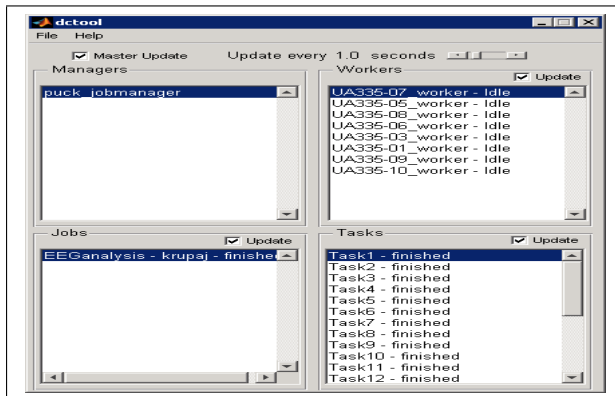


Figure 5: Graphical user interface of the dctool program

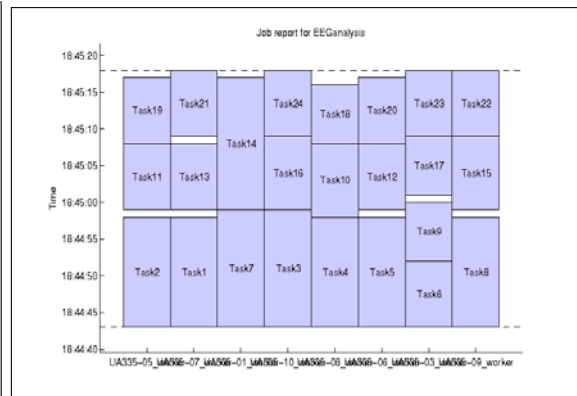


Figure 6: The Job Report figure created by the dctool

Another way how to easily monitor the current status of *job manager* is in the use of the monitoring tool based on the previously mentioned third party tools that would be created for the MATLAB Web server. There is also opportunity to include in such interactive web tools monitoring, starting and stopping of *workers*, in the computation cluster.

The next problem connected with already finished *tasks* is in the storing of *tasks* results in a specific place. This place can be realized by the FTP server or the mailbox. There are possibilities to send mails, download files from web pages and evaluate ftp commands directly from MATLAB environment as well.

4 Distributed Computing in EEG Signal Processing

EEG signal processing [4] can be widely used to provide information about brain activities and various diseases. Owing to the large amount of data values it is necessary to use fast enough computers to obtain results or to apply parallel computational methods.

Distributed computing has been applied for an extensive processing of EEG signals to find frequency similarities between individual channels for very large sets of recordings. Each of them included observations in average 900 s long sampled with the frequency of 128 Hz with an example presented in Fig. 7. To find similarities between 19 channels observed their de-noising has been applied at first [5] to remove frequency of 50 Hz at first with a typical result in Fig. 8. The next step includes the use of the coherence function [5, 1] defined by relation

$$C_{xy}(f) = \frac{|P_{xy}(f)|^2}{P_{xx}(f) P_{yy}(f)} \quad (1)$$

where $P_{xx}(f)$ and $P_{yy}(f)$ stand for power spectral density of individual signals $\{x\}$ and $\{y\}$ and $P_{xy}(f)$ represents their cross spectral density. An example of results of this processing comparing two selected channels is given in Fig. 9.

Each pair of channels have been processed by this function and the average frequency content in the given frequency range evaluated to find channels with the highest correspondence. This processing has been applied for the set of recordings processed individually as separate tasks. Results of best suited channels using this method is given in Fig. 10.

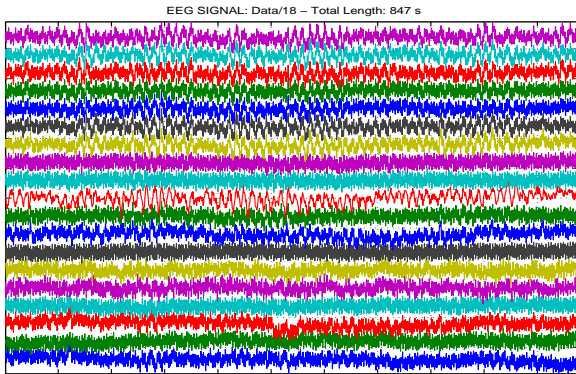


Figure 7: Original EEG signal recording presenting 19 channels with the additive noise of 50 Hz

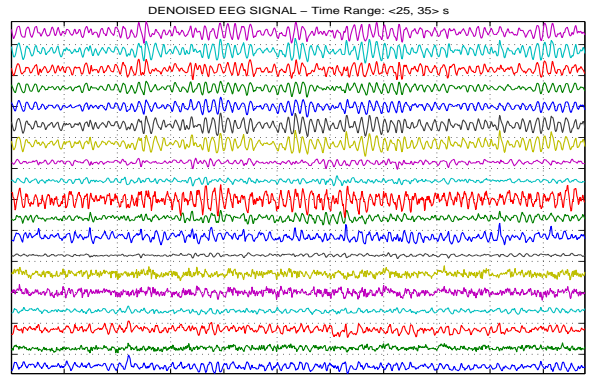


Figure 8: De-noised EEG signal after the removal of the net frequency and remaining frequency components in the range of $\langle 1, 60 \rangle$ Hz

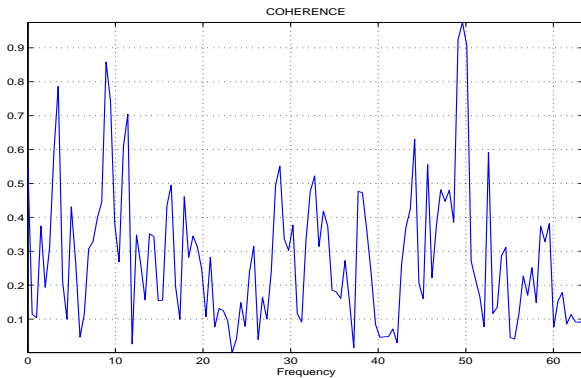


Figure 9: Coherence plot presenting frequency relation between the first and the second EEG channel

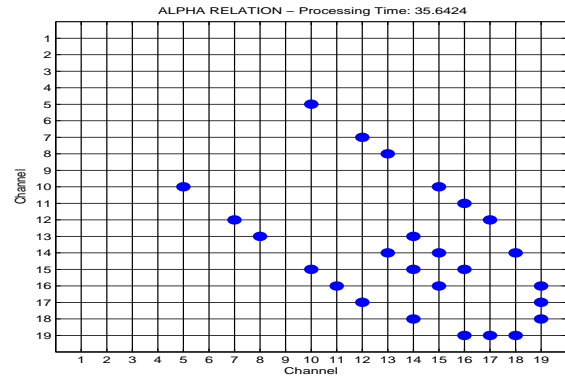


Figure 10: Channels with the most significant coherence in the ALPHA range for the set of 20 EEG recordings

Table 1: COMPARISON OF THE COMPUTATIONAL TIME USING SINGLE-CORE OR DUAL-CORE PROCESSOR UNIT AND CLUSTER OF 8 COMPUTERS FOR EEG SIGNAL PROCESSING

<i>Signal Length [s]</i>	<i>Time [s]</i>		
	<i>Single Processor Unit</i>	<i>Dual-Core Unit</i>	<i>The Cluster of 8 Computers</i>
20	90	88	25
50	122	121	30
100	155	153	34
200	241	241	46
500	463	455	78

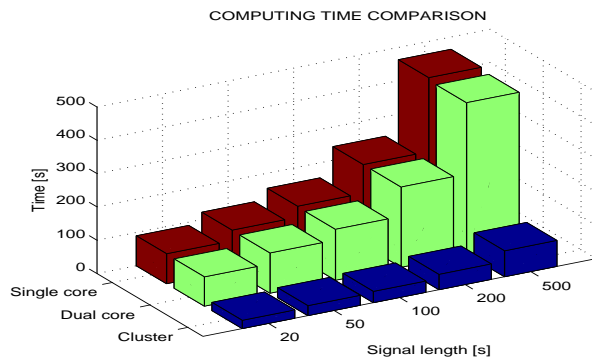


Figure 11: Graphical view of computing time comparison

Results in Fig. 11 demonstrate efficiency of distributed computing using MATLAB Distributed Computing Toolbox. The total time is not eight times better than time spent on one computer. That is because of overhead for data transfer between the cluster nodes. To minimize this overhead it is a good idea to group data to bigger units before they are distributed to workers in the cluster.

Fig. 12 compares results of best corresponding channels according to their frequency components evaluated by the coherence function for different lengths of signals.

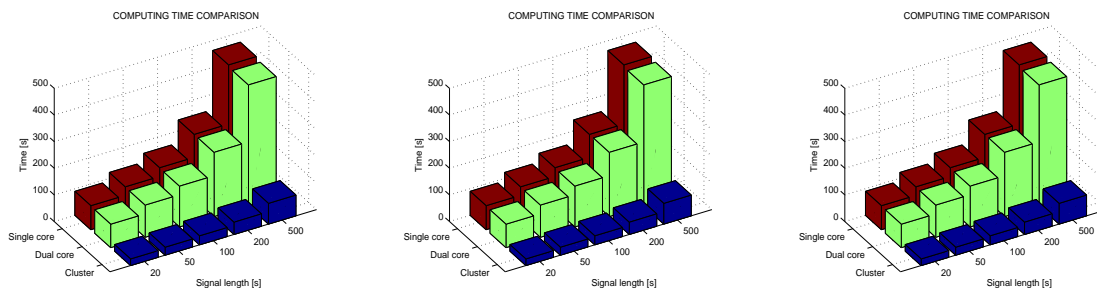


Figure 12: Comparison of the best suited EEG channels for their length (a) 100 s, (b) 200 s, and (c) 500 s

5 Conclusion

The paper presents basic principles of distributed computing and its use for EEG signal analysis to analyze frequency correspondence among its individual channels for the extensive set of data recordings. Resulting table presents the computational time needed for processing using individual computer and the set of processing units proving the efficiency for extensive individual tasks. Further research work will be devoted to the more complex analysis of multi-channel signals.

Acknowledgement

Real data sets have been kindly provided by the Neurocenter Caregroup. The paper has been supported by Research grant No. MSM 6046137306.

References

- [1] D.M. Dobreá, D.C. Dobreá, and M. Costin. An EEG Coherence Based Method Used for Mental Tasks Classification. In *5th IEEE International Conference on Computational Cybernetics (ICCC)*, pages 185–190. IEEE, 2007.
- [2] The MathWorks. *Distributed Computing Toolbox, Users Guide*. The MathWorks, Inc., Natick, MA, version 2 edition, March 2006.
- [3] The MathWorks. *MATLAB Distributed Computing Engine, System Administrators Guide*. The MathWorks, Inc., Natick, MA, version 2 edition, March 2006.
- [4] Saeid Sanei and J. A. Chambers. *EEG Signal Processing*. Wiley - Interscience, 2007.
- [5] S. V. Vaseghi. *Advanced Digital Signal Processing and Noise Reduction*. John Wiley & Sons Ltd, West Sussex, 2006.

Jan Krupa, Ing. Aleš Pavelka, Prof. Aleš Procházka
Institute of Chemical Technology, Prague
Department of Computing and Control Engineering
Technická 1905, 166 28 Prague 6
Phone.: +420 220 444 198, Fax: +420 220 445 053
E-mail: Jan.Krupa@vscht.cz, Ales.Pavelka@gmail.com, A.Prochazka@ieee.org
WWW: <http://dsp.vscht.cz>