DISTRIBUTED COMPUTING SYSTEM FOR EEG SIGNAL PROCESSING USING MATLAB AS ACTIVEX OBJECT IN DCOM MODEL

P. Ježdík [1], R. Čmejla [1], P. Kršek [2], A. Jahodová [2]

[1)] Faculty of Electrical Engineering, Czech Technical University in Prague,
[2)] 2nd Medical School, Charles University in Prague

Abstract

Automated signal processing of electroencephalographic (EEG) data is mostly very sophisticated and time consuming. Also the data amounts of the patients necessary to process are mostly high. Using a parallel signal processing techniques is suitable for saving the time. When the workgroup computing force consist only standard personal computers with standard equipment, a technique easy to implement is necessary. The one suitable technique is the Distributed Component Object Model (DCOM) mechanism that basically allows ActiveX clients to use remote ActiveX objects over a network. This paper describes one possibility of using a number of Matlab instances as ActiveX objects in school network working in synergy when the consequent is decreasing the computing time of EEG signal processing.

1 Introduction

Our workgroup design a signal processing algorithm for EEG records obtained from a set of epilepsy patients. The data records lengths are in orders of tens minutes with sampling frequency 200 and/or 1000Hz. The number of processed channels is mostly between 60 and 128. Using a signal processing algorithm designed by the workgroup, demanded time for process of one EEG channel record equals from ones to tens of minutes. Overall the whole data necessary to process would have taken days when only one computing unit would be used. The channels in the EEG records can be processed separately. It allows use a technique of parallel processing when more computational units are available.

Actually the workgroup equipment is limited to use a number of standard personal computers mostly having single or double core processors. The computers are basically working with WindowsXP system and connected to standard TCP/IP network. Since the Windows NT4.0 the standard equipment of the system is the Distributed Component Object Model (DCOM) which is an object distribution mechanism that allows ActiveX clients to use remote ActiveX objects over a network [1][2]. Also the Matlab allows run its instances as the component object [3].

Once the DCOM setting is done in respect to network and system safety, the client-server structure is ready to use. We use a star structure with one central node. The central node starts the Matlab instances in all connected computers using a `actxserver(....)` Matlab command. See Figure 1.

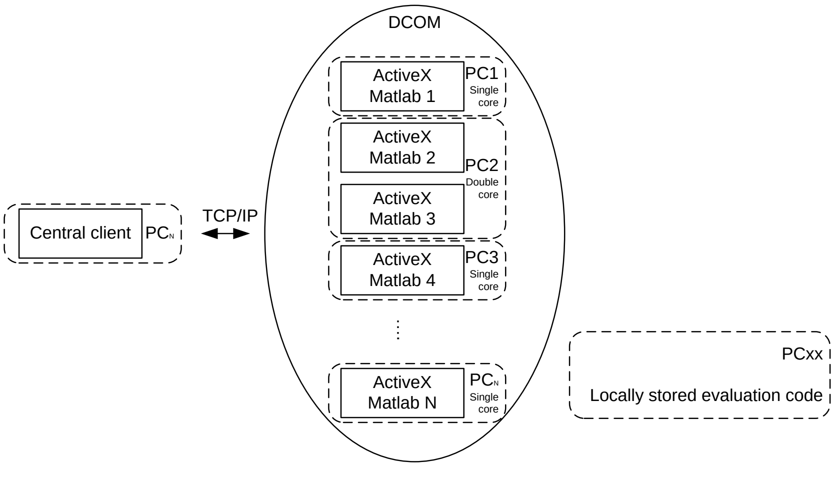

Figure 1: Client server structure of DCOM model of ActiveX Matlab objects

2 Algorithm

The central node starts the same number of instances as the number of cores in each machine. Each machine has a locally stored code for data evaluation and control mechanism. There is used a trick with delay run timer for the evaluation routines. It consequents immediate return of control (program code flow) back to the client. The client starts the delay timer countdown in the remote instance. The script in the remote instance finish immediately and the client can continue in other instances. When a particular counter is down, its evaluation routine starts. The only problem is, how to say the main node, the local results are ready. We used Matlab UDP programming possibilities for making a channel carrying the flag. For illustration see the Figure 2.

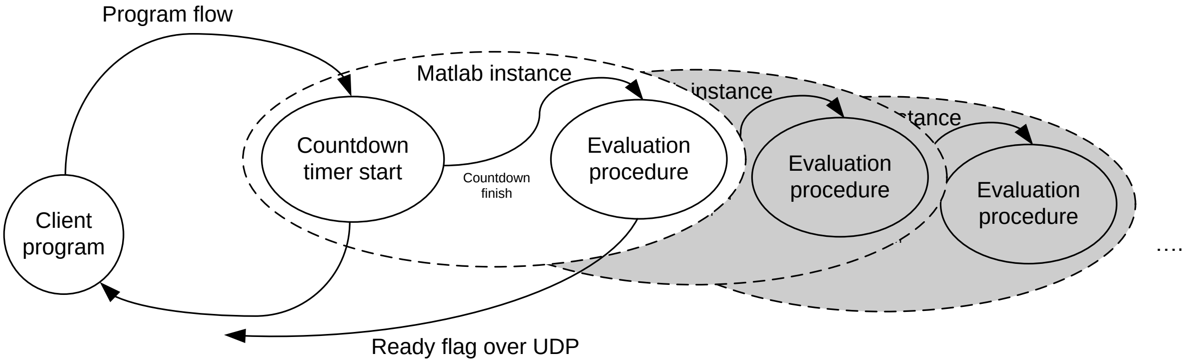

Figure 2: Program flow in the structure

The central node loads the EEG data. Than each channel is distributed to a remote instance until a channel or unused instance left. There is repetitively checked the ready flag from the instances UDP channels. Data are read and instance freed for next channel, if left. Finally all channels are collected back to one results pack.

3 Implementation

Each Matlab ActiveX instance needs the timer, the UDP communication link and timer start function. It can be declared as follows. The code has to be locally stored at each computer supposed to be the ActX Matlab.

```
t1=timer('StartDelay',1,'TimerFcn','RunEval');  % countdown to 1 second
```

The UDP link:

```
ucomlink=udp(ucomlinkRemoteHostAddress,ucomlinkPort);
set (ucomlink,'LocalHost',ucomlinkLocalHost,'LocalPort',ucomlinkPort);
fopen (ucomlink)
```

And the timer start function

```
start (t1)
```

The Evaluation functions stored at local machines are labeled RunEval for this example.

The ActX client e.g. the central node than has to contain following instructions.

```
Remote_instance_handle(ii)=actxserver('matlab.application.single',Address); % runs remote instance
Execute(remote_instance_handle(ii),'servercommunicator') % runs the declaration of the UDP link
Execute(remote_instance_handle(ii),'declaredelayrun'); % runs the declaration of the countdown timer
```

The execution of the evaluation routines at the remote instances is ran by

```
Execute(remote_instance_handle(ii),'startdelaytimer');
```

For data exchange the standard Matlab function PutWorkspaceData and GetWorkspaceData are used.

4 Summary

As an example of computational time savings can be mentioned a comparison of times for computing Directed function [4] for EEG record of one patient. There was evaluated a record of 88 channels, 10 minutes at 200Hz e.g. 120 000 samples per channel. Each channel is recursively processed with each other. One processor unit (about 1.6GHz, 1G RAM) computed the record for more than 20hours. Distributed system consisting 18 units did it in less than hour. There are not exact timing values due to non relevant interest of our research. The exact equation for the system performance describing using the distribution idea could be depend on particular performances of the machines, network throughput etc.

Generally the distribution system saves the records evaluation time. Computing time decreases with indirect proportion of number of computational instances when equal cores are used. It is easy to implement with standard personal computers and a network. Our application works in a student's computing laboratory at the university during nights.

Acknowledgement

This work has been supported by the grants IGA NT11460-4/2010 Intracranial EEG signal processing; epileptogenic zone identification in non-lesional refractory epilepsy patients, SGS 10/272/OHK4/3T/13 Analysis of intracranial EEG recording, and resesearch program MSM6840770012 Transdisciplinary Research in Biomedical Engineering.

References

[1] Grimes, R. (1997), *'DCOM programming'*, WROX, Press Ltd.
[2] Redmond, F. (1997), *'Dcom: Microsoft Distributed Component Object Model '*
[3] The MathWorks Inc (1994-2005), Matlab Help
[4] Sanei, S., Chambers, J. (2007), *EEG Signal Processing*. John Wiley

Ing. Petr Ježdík
e-mail: jezdip1@fel.cvut.cz

Doc. Ing. Roman Čmejla, CSc.
e-mail: cmejla@fel.cvut.cz

Doc. MUDr. Pavel Kršek, PhD.
e-mail: pavel.krsek@post.cz

MUDr. Alena Jahodová
e-mail: a.jagoda@email.cz