

ROS Manual (*Remote Optimization Service
User Manual*)

J. M. García Nieto

7 September 2003

Contents

1	Introduction	2
2	Installing ROS	3
2.1	Windows	3
2.1.1	Client	3
2.1.2	Primary Server	3
2.1.3	Server	4
2.1.4	Worker	4
2.2	Unix/Linux/Solaris	4
2.2.1	Client	4
2.2.2	Primary Server	5
2.2.3	Server	5
2.2.4	Worker	5
3	Using ROS	7
3.1	Local Use	7
3.2	LAN/WAN Use	8
4	Running ROS	10
4.1	Running Primary Server	10
4.2	Running Server	10
4.3	Running Worker	11
4.4	Running Client	12

Chapter 1

Introduction

ROS (Remote Optimization Service) is a system that offers the possibility to us of executing *Optimization Algorithms* remotely, by means of the communication of the necessary information for this objective between different kinds of machines.

Between these different kinds of machines we have:

- *ROS Client*, it provides the client user interface and takes care of the information collection, election of servers and presentation of results.
- *ROS Primary Server*, it provides to client information about the system.
- *ROS Server*, it acts like intermediate server of distribution of the information flow.
- *ROS Worker*, it connects with the algorithms and manipulates them to obtain the result information.

ROS is implemented in *Java* and encapsulates the information by means of standard *XML*. IN addition, provides a visual surroundings to facilitate its use to the user.

The system can be completely installed in a machine to work in Local mode or be distributed in different machines in a *LAN/WAN* network.

We dispose *ROS* versions to different kinds of machines (*Windows, Linux, UNIX, Solaris...*) being able to put in the different kinds of O.S in the system, this is, to install the Client in Windows and the servers in Linux and Solaris, or the Client and a server in Linux and the others in Windows...

In this paper, we offer to you a *help manual* for the *ROS* use. In the following chapter we give instructions for the installation of the system, in chapter 3 we will see as the files of internal information of the system are configured and finished, in chapter 4, with running examples of the system.

Chapter 2

Installing ROS

2.1 Windows

2.1.1 Client

Instructions

- After downloading, unpack `install.zip`.
- Double-click `install.exe`.

Notes

- You may need to install a Java 1.1.8 (or later) virtual machine. You can download one from Sun's Java web site <http://java.sun.com/cgi-bin/java-ports.cgi>.

2.1.2 Primary Server

Instructions

- After downloading, unpack `install_pserver.zip`.
- Double-click `install_pserver.exe`.

Notes

- You may need to install a Java 1.1.8 (or later) virtual machine. You can download one from Sun's Java web site <http://java.sun.com/cgi-bin/java-ports.cgi>.

2.1.3 Server

Instructions

- After downloading, unpack `install_server.zip`.
- Double-click `install_server.exe`.

Notes

- You may need to install a Java 1.1.8 (or later) virtual machine. You can download one from Sun's Java web site <http://java.sun.com/cgi-bin/java-ports.cgi>.

2.1.4 Worker

Instructions

- After downloading, unpack `install_worker.zip`.
- Double-click `install_worker.exe`.

Notes

- You may need to install a Java 1.1.8 (or later) virtual machine. You can download one from Sun's Java web site <http://java.sun.com/cgi-bin/java-ports.cgi>.

2.2 Unix/Linux/Solaris

2.2.1 Client

Instructions

- After downloading, unpack `install.tar.gz` (`% tar xzvf install.tar.gz`).
- Open a shell and, `cd` to the directory where you downloaded the installer.
- At the prompt type: `chmod a+x install.bin`.
- At the prompt type: `sh ./install.bin`.

Notes

- You may need to install a Java 1.1.8 (or later) virtual machine. You can download one from Sun's Java web site <http://java.sun.com/cgi-bin/java-ports.cgi>.

2.2.2 Primary Server

Instructions

- After downloading, unpack `install_pserver.tar.gz` (`% tar xzvf install_pserver.tar.gz`).
- Open a shell and, `cd` to the directory where you downloaded the installer.
- At the prompt type: `chmod a+x install_pserver.bin`.
- At the prompt type: `sh ./install_pserver.bin`.

Notes

- You may need to install a Java 1.1.8 (or later) virtual machine. You can download one from Sun's Java web site <http://java.sun.com/cgi-bin/java-ports.cgi>.

2.2.3 Server

Instructions

- After downloading, unpack `install_server.tar.gz` (`% tar xzvf install_server.tar.gz`).
- Open a shell and, `cd` to the directory where you downloaded the installer.
- At the prompt type: `chmod a+x install_server.bin`.
- At the prompt type: `sh ./install_server.bin`.

Notes

- You may need to install a Java 1.1.8 (or later) virtual machine. You can download one from Sun's Java web site <http://java.sun.com/cgi-bin/java-ports.cgi>.

2.2.4 Worker

Instructions

- After downloading, unpack `install_worker.tar.gz` (`% tar xzvf install_worker.tar.gz`).
- Open a shell and, `cd` to the directory where you downloaded the installer.

- At the prompt type: `chmod a+x install_worker.bin.`
- At the prompt type: `sh ./install_worker.bin.`

Notes

- You may need to install a Java 1.1.8 (or later) virtual machine. You can download one from Sun's Java web site <http://java.sun.com/cgi-bin/java-ports.cgi>.

Chapter 3

Using ROS

3.1 Local Use

We can install all components of **ROS** in a same machine and configure it so that the system works in local way. We enumerated next the steps to follow in each configuration.

1. Using Client: once the Client has been installed, we must write the primary server IP (or `localhost` IP) in the configuration file `ROS/ps_ip.txt`, therefore the client will connect with the local Primary Server.
2. Using Primary Server: in this case the Server IP are introduced . We put in `ROS_PServer/typeip_file.txt` the local IP next to each type of algorithm.
3. Using Server: the configuration file in this case is `ROS_Server/workerconfig.txt`. The local direction next to each type of algorithm are introduced.
4. Using Worker: we install the Worker with it´s actives algorithms.

In the figure above (Figure 3.1), we can see a scheme of configuration files in a local use of **ROS**.

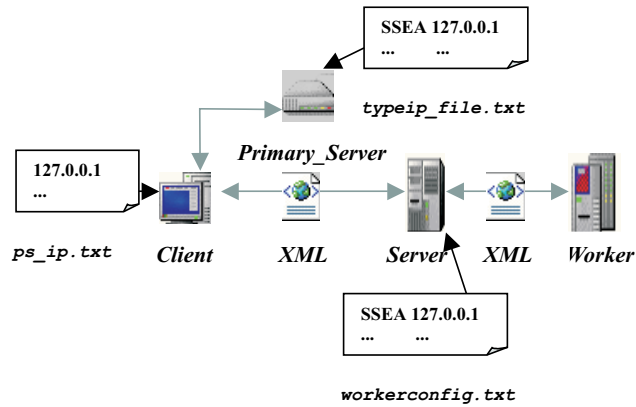


Figure 3.1: Local mode of configuration files in **ROS**.

3.2 LAN/WAN Use

On the other hand, we can distribute the client and servants of ROS in different machines from a LAN/WAN network. In this case we must write the IP addresses of these machines in the configuration files of the system.

1. Using Client: we must write in the configuration file `ROS/ps_ip.txt` the IP addresses of the machines where the Primary Server are lodged.
2. Using Primary Server: we will write in the configuration file `ROS_PServer/typeip_file.txt` the addresses of machines where the installed Server are lodged. In each line it will be written the type of algorithm and address.
3. Using Server: we will write in the configuration file `ROS_Server/workerconfig.txt` the addresses of machines where the installed Server are lodged. In each line it will be written the type of algorithm and address.
4. Using Worker: we install the Worker with it's actives algorithms.

In the figure above (Figure 3.2), we can see a scheme of configuration files in a LAN/WAN use of **ROS**.

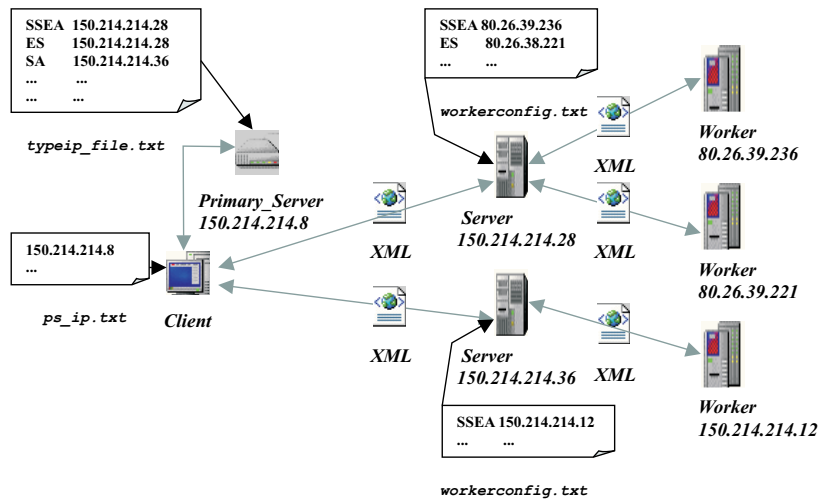


Figure 3.2: LAN/WAN mode of configuration files in **ROS**.

Chapter 4

Running ROS

4.1 Running Primary Server

In order to start up the Primary Server, we must execute the program:

- pressing the executable file `ROS_PServer.exe` (*Windows S.O*).
- writing the *shell* command `% sh ./ROS_PServer.bin` (*Unix/Linux/Solaris*).

It will appear in screen the program represented in the following figure (Figure 4.1).



Figure 4.1: Primary Server service.

This indicates that the service is active. In order to stop the service is enough with pressing the *Stop* button.

4.2 Running Server

To start up the Server, we must execute the program:

- pressing the executable file `ROS_Server.exe` (*Windows S.O*).
- writing the *shell* command `% sh ./ROS_Server.bin` (*Unix/Linux/Solaris*).

It will appear in screen the program represented in the following figure (Figure 4.2).



Figure 4.2: Primary Server service.

This indicates that the service is active. In order to stop the service is enough with pressing the *Stop* button.

4.3 Running Worker

To start up the Worker, is the same one that the previous cases:

- pressing the executable file `ROS_Worker.exe` (*Windows S.O*).
- writing the *shell* command `% sh ./ROS_worker.bin` (*Unix/Linux/Solaris*).

It will appear in screen the program represented in the following figure (Figure 4.3).



Figure 4.3: Primary Server service.

This indicates that the service is active. In order to stop the service is enough with pressing the *Stop* button.

4.4 Running Client

In order to start up the Client, we must execute the program:

- pressing the executable file `ROS_Client.exe` (*Windows S.O.*).
- writing the *shell* command `% sh ./ROS_Client.bin` (*Unix/Linux/Solaris*).

In the first window that appears (Figure 4.4) we will choose the language which we want to work.

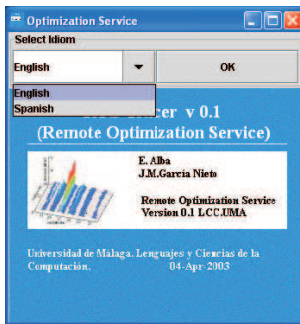


Figure 4.4: Client window to choose idiom.

After choosing a language we pressed *OK* and will appear the main window of the Client program (Figure 4.5).

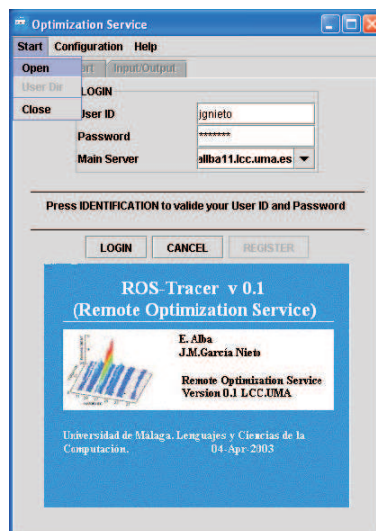


Figure 4.5: Main Client window.

Here we will identify ourselves introducing the user ID and Password, next we chose the address of the Primary Server respect to we want to work and we pressed the *LOGIN* button.

After this, if is a correct identification it will show a message to us to continue. If is no correct, it would show a informative message and we would to the registry pressing *REGISTER*.

When we finished with the identification, we go away to the main menu and we opened the window of type of algorithm and servant choosing (Figure 4.6) by means of the options sequence *Start - > Open* .

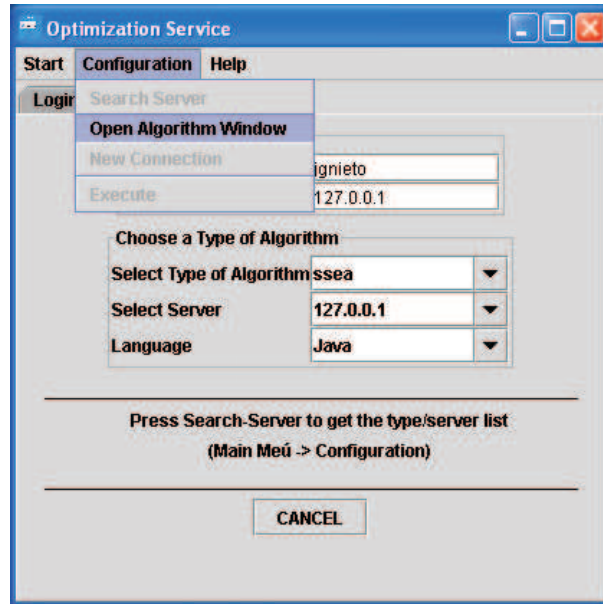


Figure 4.6: Client window for algorithm and server choosing.

In this window we will choose the kind of algorithm, the server machine where to find it and the programming language in which it is implemented of the different options that present to us. We will obtain the list of available kinds of algorithms by means of the option *Configuration - > Search Server* in the main menu.

When completing the election, we will go already to the specifies window of the type of chosen algorithm (Figure 4.7). In order to this, we will go away to the main menu and we will press *Configuration - > Open Algorithm Window*.

Within the algorithm window, we will see the fields of specify parameters (with values by defect) which we must complete.

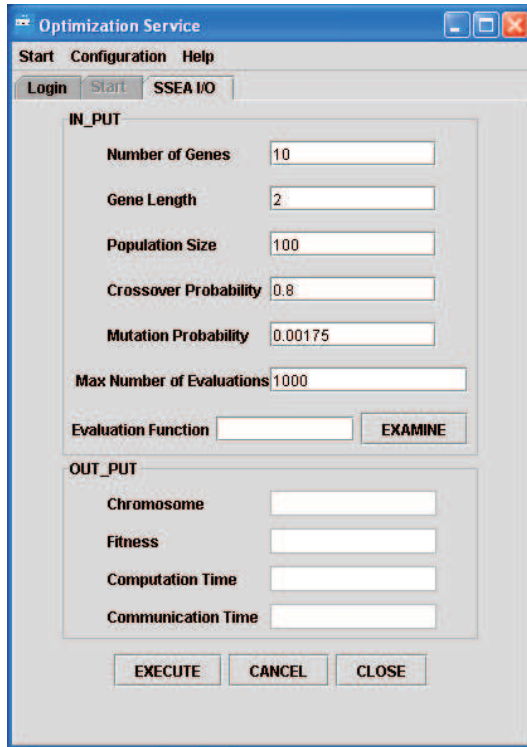


Figure 4.7: Client window of chosen algorithm.

In the case of the *Evaluation Function*, we can load it from an external file by means of the search window (Figure 4.8).

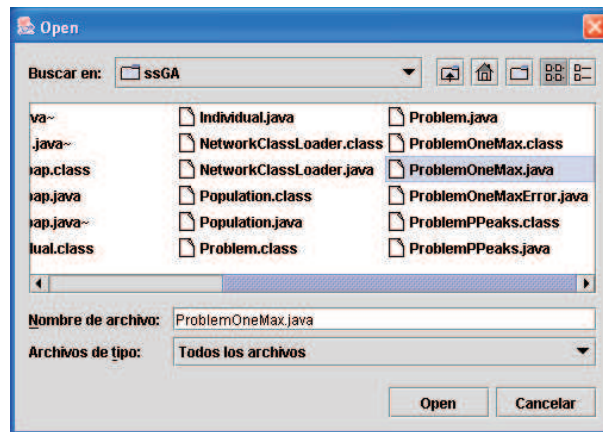


Figure 4.8: Window for Problem chosen.

Once completed all the fields with the corresponding values, we can press the button *EXECUTE*. Now the program will pass to the communication of the data and execution of the algorithm.

Ending the cycle, it will show the results in its corresponding fields (Figure 4.9).

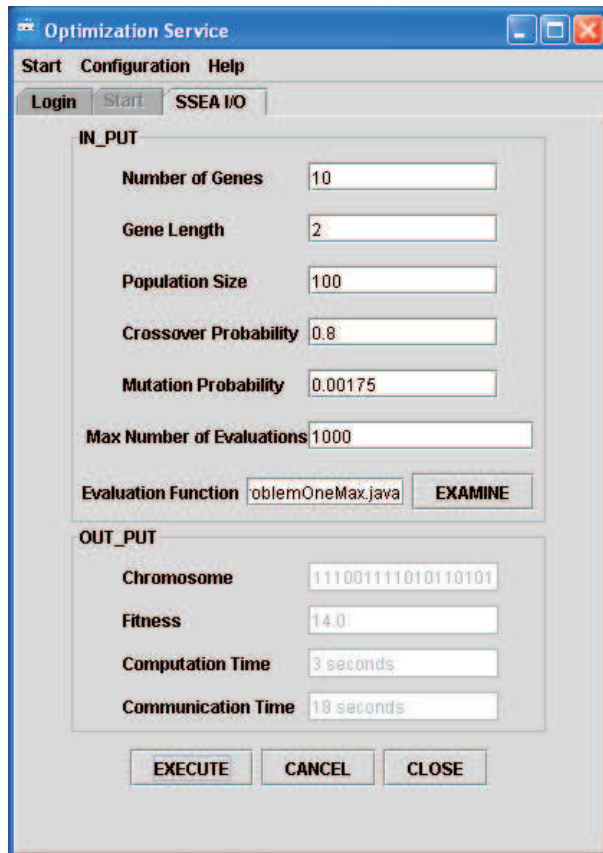


Figure 4.9: Client window with results.

In error case, the system will show messages to us specifying the nature of the same one: communication (Figure 4.10), compilation, execution...

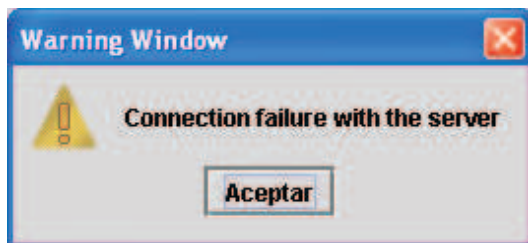


Figure 4.10: Communication error message.