

Yamamoto, Yoshikazu; Nakano, Junji

Working Paper

Distributed computing in a time series analysis system

SFB 373 Discussion Paper, No. 2001,76

Provided in Cooperation with:

Collaborative Research Center 373: Quantification and Simulation of Economic Processes,
Humboldt University Berlin

Suggested Citation: Yamamoto, Yoshikazu; Nakano, Junji (2001) : Distributed computing in a time series analysis system, SFB 373 Discussion Paper, No. 2001,76, Humboldt University of Berlin, Interdisciplinary Research Project 373: Quantification and Simulation of Economic Processes, Berlin, <https://nbn-resolving.de/urn:nbn:de:kobv:11-10050500>

This Version is available at:

<https://hdl.handle.net/10419/62710>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

DISTRIBUTED COMPUTING IN A TIME SERIES ANALYSIS SYSTEM

Yoshikazu Yamamoto * and Junji Nakano †

ABSTRACT

In this paper, we consider simple and easy distributed computing abilities of the statistical system TISAS (Time Series Analysis Supporting system). TISAS was mainly written in the Tcl/Tk language, and was designed for utilizing full merits of a GUI (Graphical User Interface). We add functions for distributed computing without changing the style of the TISAS GUI by using TkPVM, which is an implementation of PVM (Parallel Virtual Machine) libraries through Tcl/Tk.

We adopt a server/client system structure for the new version of TISAS. The server part executes data manipulations and the client part does GUI works. For designing them, we use a Model-View-Controller framework, which makes our implementation relatively simple.

1. Introduction

We designed and implemented an experimental time series analysis system named TISAS (Time Series Analysis Supporting system) for utilizing a modern GUI (Graphical User Interface) effectively (Yamamoto et al. 2000). TISAS works well for small datasets at a first stage of an analysis. However, when we analyze large datasets or continue an analysis deeply even for a small dataset, it becomes rather slow. This is caused by the fact that TISAS is mainly written in the Tcl/Tk language¹ (Ousterhout 1994), and the Tcl/Tk interpreter becomes slow when large datasets are stored in it. It is apparent that we should improve TISAS to have much speed and capacity in some ways for large datasets or complicated analyses.

In addition, the amount of calculation required for statistical works has been increased rapidly, mainly because many new computer intensive methodologies, for examples, simulation experiments and resampling techniques, are developed.

Recently, computer networks are well developed and personal computers become cheap and powerful. Almost all statisticians can use more than one computers connected by networks. Considering these situations, distributed computing technologies look promising and have already been studied in various ways.

We decide to implement new TISAS by using distributed computing technologies for improving the performance. As we do not want to change our GUI style largely and hope

*Faculty of Engineering, Tokushima Bunri University, 1314-1, Shido, Okawa, Kagawa 769-2193, Japan

†The Institute of Statistical Mathematics, 4-6-7, Minami-Azabu, Minato-ku, Tokyo 106-8569, Japan

Key words: Distributed computing, Graphical User Interface, Model-View-Controller, PVM, Statistical system, Tcl/Tk

¹<http://www.scriptics.com/software/tcltk/>

to keep the GUI for the distributed computing ability as simple as possible, we just add a few menu items to the original GUI.

We use TkPVM², which is an implementation of PVM³ (Parallel Virtual Machine) in the Tcl/Tk language. PVM is one of the most famous libraries for distributed computing and enables computers of different types connected by the network to be used like a single parallel computer.

We divide functions of TISAS into a client part and a server part for realizing distributed computing abilities. The client program can communicate with more than one server programs at the same time. The server program handles data and calculates statistics. The client program displays icons of data, statistics and statistical models on servers, and show numerical results by graphs and tables. Operations to icons are sent to remote server programs on which data are stored and calculations are performed. These functions are realized by using the Model-View-Controller framework for the GUI design.

We describe the basic design of TISAS in Section 2. Section 3 explains distributed computing technologies concerning statistical systems, and general purpose libraries such as PVM and MPI. In section 4, a Model-View-Controller framework is explained. Section 5 describes the design of distributed computing abilities of TISAS. At last, Section 6 states some concluding remarks.

2. Original design of TISAS

TISAS was designed to realize an object based GUI for time series analysis, which means that data, derived statistics and fitted statistical models are expressed as objects, and they are represented as icons on the screen (Fig.1). TISAS provides basic and standard linear time series analysis procedures and prediction procedures using AR and ARMA models listed in Table 1. Many of these functions are realized by C++ version of TIMSAC package⁴ (Akaike et al. 1989). TISAS can draw graphs of time series (including residuals and predicted values), autocovariances, spectral densities and has basic data handling procedures such as data transformations. Equations for data transformations are composed by mouse operations on the TISAS GUI. Newly built equation can be added as a new menu item of a popup menu.

If we specify an icon by the mouse, a situation sensitive pop-up menu appears, in which appropriate statistical procedures for the object are listed. We can execute one of them by selecting it. Manipulating icons in this way usually generates new models or statistics, and new icons for expressing them are shown as nodes of a tree corresponding to their historical orders. These icons are distinguished by the color and the shape of them (Table 2).

The dependence of operations is indicated by locations of icons. Calculated statistics icons and fitted model icons are located inside an original statistics or time series icon. An icon of generated time series by data handling and an original time series icon are connected by a line and an equation of the transform is shown above the derived icon. Displaying a history of a data analysis clearly in this way by a tree of icons is useful to understand the process of the analysis and to find a possible future analysis. Handling icons directly helps us to concentrate on a present object in the complicated exploratory data analysis process.

²<http://www.cogsci.kun.nl/tkpvm/welcome.html>

³http://www.epm.ornl.gov/pvm/pvm_home.html

⁴<http://www.ism.ac.jp/software/products-e.html>

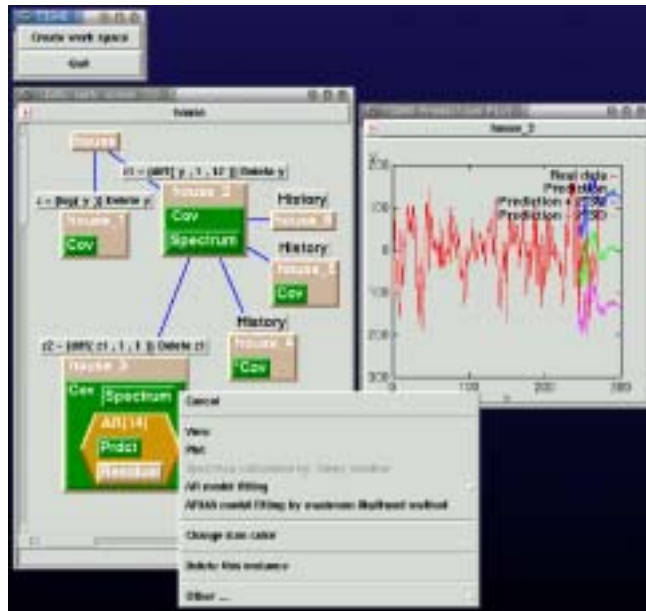


Fig. 1. Displaying the history of a data analysis

3. Some distributed computing technologies

Many software products are available and are studied for utilizing network abilities because of prevalence of Internet and the improvement of network technologies and network programming environments. In this section, we describe some of them for statistical systems and for general uses, separately.

3.1. Distributed computing in statistical systems

In statistical systems, distributed computing technologies were largely used for developing GUIs. The “Composite user interface” (Liu et al. 1995) is an approach to develop GUIs for non-GUI systems, and adds user interface parts to original calculation parts. One

Table 1. Available procedures in TISAS

Sample autocovariance function (Univariate and Multivariate)
Spectral density estimation by sample autocovariance function (Univariate and Multivariate)
AR model fitting by FPE criterion (Univariate and Multivariate)
AR model fitting by AIC criterion (Univariate and Multivariate)
Spectral density estimation by AR model fitting (Univariate and Multivariate)
ARMA model fitting by the maximum likelihood method (Univariate)
Prediction values using AR model or ARMA model (Univariate and Multivariate)
Noise contribution of AR model (Multivariate)

Table 2. A label, color and shape of icons

Label	Color	Shape	
file name	bisque	rectangular	Time series
Cov	green	rectangular	Sample autocovariance function
Spectrum	green	rectangular	Spectral density estimation
AR	red	hexagonal	AR model fitting by FPE criterion
AR	orange	hexagonal	AR model fitting by AIC criterion
ARMA	gray	hexagonal	ARMA model fitting by the maximum likelihood method
Prdct	green	rectangular	Prediction values using AR model or ARMA model
Noise	green	rectangular	Noise contribution of AR model
Residual	bisque	rectangular	Residual of time series and model

of the advantages of this approach is to keep the reliability of calculation results, because a minimum change is required for the calculation program. GUI and calculation parts can run on different computers and communicate mutually. Usually, the calculation program runs on powerful mainframe or workstation computers and a user interface program runs on less powerful personal computers. As a result, we can use the calculation power of large computers from small desktop or laptop personal computers.

This mechanism has been expanded to be used through Internet technologies. Some statistical systems use CGI (Common Gateway Interface) via HTTP (HyperText Transfer Protocol). Users can start a CGI program for calculation on a remote server computer using a WWW (World Wide Web) browser, for example, Microsoft Internet Explorer or Netscape Navigator, as a GUI program. Results of the CGI program are returned in HTML (HyperText Markup Language) and displayed on the WWW browser. As WWW browsers are simple and easy to use, this user interface is thought to be useful and friendly for most users (Nakano et al. 1997). Another advanced example is Rweb⁵, which is a user interface of R⁶ on WWW browsers. It uses JavaScript together with CGI. On Rweb, we can use all commands of R and can see graphs of R. However, we have to note that this approach has difficulty for accessing local files, because WWW browsers have severe limitations for keeping network securities. As it is inconvenient that statisticians cannot analyze data files on their local computers, Nakano (1998) used an original protocol to communicate between a client started from WWW browsers and a server.

Some recent statistical systems realize distributed computing abilities by the Java language⁷, which has advanced network functions such as socket libraries, applets, RMI (Remote Method Invocation) and CORBA (Common Object Request Broker Architecture) (Morrison et al. 1997). West (1995) is an example of an applet based statistical system. The Java GUI of XploRe⁸(Härdle et al. 1999) and Kötter (1997) can be used both as an applet and an application.

3.2. General purpose distributed computing technologies

Most famous and widely used distributed computing technologies are PVM and MPI⁹

⁵<http://www.math.montana.edu/Rweb/>

⁶<http://www.r-project.org/>

⁷<http://java.sun.com/>

⁸<http://www.xploRe-stat.de/>

⁹<http://www.mpi-forum.org/>

(Message Passing Interface), which are low level libraries for adding distributed computing abilities to existing computer languages such as Fortran and C.

PVM (Geist et al. 1994) was developed earlier than MPI. Using PVM, we can easily execute a parallel program on several kinds of operating systems such as Microsoft Windows and UNIX operating systems. As we adopt PVM for developing distributed computing abilities of TISAS, we explain it precisely in the next section.

MPI is a standard library specification for message passing by a committee of vendors, implementers and users, including developers of PVM. MPI was designed for high performance calculations on both massively parallel machines and workstation clusters. MPI is released as both a freely available implementation and vendor-supplied implementations. Now, MPI is a standard low level library for distributed computing.

There exist some projects to develop higher level general distributed computing systems, which are so called global computing systems. Nin^{f10} (Sato et al. 1997) and NetSolve¹¹ (Casanova et al. 1996) are examples. They have client and server programs, for example, NetSolve has agents to look for suitable server computers for a given problem. These systems can use more than one server computers at the same time, and may be useful for data analyses if users can write IDL (Interface Definition Language) programs, which is, however, not easy for statisticians.

4. Distributed computing abilities and technologies for improving TISAS

Statistical systems mentioned above use distributed computing abilities mainly for realizing GUIs for one server computer. If a user has several computers connected by networks, it will be effective to use more than one computer at the same time. However, a statistical system should be easy to use even for executing distributed computing, because users do not want to learn new operations to write complicated programs for using distributed computing abilities. It is desirable that operations for distributed computing are as same as basic ones.

If we use many servers, we have to maintain them to construct a distributed computing environment. Although loads of users increase for it, a statistical system and its GUI should reduce them as much as possible. For example, it is preferable that server programs are automatically started when they are required even if statisticians do not have administrative right of remote server computers.

There are various merits to use several computers simultaneously for distributed computing and parallel computing. It can increase execution speed by dividing large calculation amount into smaller pieces, for example, in simulation or numerical maximization works. It also can contain large datasets including raw data and calculation results separately on each computer. In distributed computing, we should be careful that communications among computers are as little as possible, because the communication overhead is apt to become very large.

For improving TISAS, the second merit of distributed computing seems to be more useful than the first ones. Original TISAS is slow mainly because it stores all calculated results and transformed data in one Tcl/Tk interpreter. Considering the fact that transformed datasets are independent when it is used for calculating basic statistics and fitting models, they can be handled separately on different computers without any trouble.

¹⁰<http://ninf.etl.go.jp/>

¹¹<http://www.cs.utk.edu/netsolve/>

Under these considerations, we design a client part and a server part of the new TISAS. The client can communicate with several server programs simultaneously in order to distribute storage and calculation loads on them without any operations difficulty. We add a few items for distributed computing operations to a menu in the same way as original ones. We let server programs start on demand automatically and can maintain remote server computers easily once we installed server programs. We keep the network traffic as little as possible by not sending same data twice.

For realizing these abilities, we choose PVM through the TkPVM implementation, because TISAS was written in Tcl/Tk. We find that this technology is enough useful for our purposes, although other choices, for example, the Java language or the NetSolve system, may bring richer functions.

As we implement the TISAS GUI as a client program by separating from a server program, we adopt the Model-View Controller (MVC) framework for realizing it. MVC is admitted to be useful this type of GUI construction.

4.1. PVM and TkPVM

PVM is a library for realizing distributed computing on the Fortran and C language. As functions are simple and well designed, users who are good at programming in these languages can implement distributed computing works relatively easily. PVM has a lot of contributed software products, for example, debuggers, job schedulers and automatic load balancers. Moreover, we can use PVM functions from other languages such as Java, Perl¹² and Python¹³. TkPVM is one of such software libraries for using PVM functions from the Tcl/Tk language.

TkPVM is developed for adding PVM abilities to the Tcl/Tk interpreter (Hartung et al. 1996). It is available from the Internet Parallel Computing Archive¹⁴ freely, and works on almost all UNIX operating systems where PVM and the Tcl/Tk language can run.

TkPVM has nineteen Tcl procedures to use PVM abilities such as `spawn` and `send`. The `spawn` procedure starts a program on a specified computer and spawns a process to communicate with the program. The `send` procedure sends all arguments to the process started by the `spawn` procedure. In addition, we can write an event driven program to bind commands to an event, for example, a signal for notifying a process termination on a server computer, by the `bind` procedure. When the event occurs, bound commands will be executed. This ability is the most convenient feature in constructing the GUI for distributed computing, because multiple inputs can be handled simultaneously. For example, users can operate icons or menus even if a server is executing calculations.

4.2. The Model-View-Controller framework

The MVC framework was originally adopted in the Smalltalk language to support its GUI (Pinson et al. 1988). Recently, it is used to build Swing¹⁵ components of the Java language. This is a framework of designing and organizing programs of GUI-intensive systems, and makes the implementation relatively simple. This framework uses three modules for a “model”, a “view” and a “controller”, which are realized by sets of instance objects in the object oriented methodology.

¹²<http://www.perl.com/>

¹³<http://www.python.org/>

¹⁴<http://wotug.ukc.ac.uk/parallel/>

¹⁵<http://java.sun.com/products/jfc/tsc/>

The view and the controller are interface modules between a user and a program. The controller translates a user's operation into a message in the object oriented methodology and sends it to the model. The model is a data structure for an application and internal programs are written independently with an interface program. The model is logically connected to one or more sets of the view and the controller. The view displays a data object of the model and determines the display form. As the view has its own display style, the display form of the model can be changed for each view. This means that it is easy to change the display form of the model by using different views. For example, if we want to change the display form of data from a table to a histogram, we only need to change the link of the model from a table view to a histogram view. Another advantage is that new user interfaces such as new menus are easily added by making a new controller and replacing the old one with it.

The MVC framework is also suitable for statistical software, in which we sometimes see a dataset in different forms such as graphs and tables. We can implement the function that when a user modifies data in a table form, a graph expressing same data is changed automatically and simultaneously.

For these reasons, we design classes of TISAS client and server program using the MVC framework. Although this framework does not have any network abilities originally, we use it for the GUI design on a distributed computing environment by extending the communication between the model and the view or the controller on different computers. Thus, the TISAS client program which consists of the view and the controller and the server program which is the model can run on different computers just like on the same computer. A message of the controller is delivered to the model on a remote server computer and the model sends a message to the view on a client computer.

The server handles data and calculates statistics. The client displays icons of data, statistics and statistical models on servers, and show numerical results by graphs and tables. Objects of data, statistics and statistical models are models of the MVC framework. Objects of icons, graphs and tables are views, and objects of menus are controllers. In TISAS, each model is linked to three sets of the view and the controller: an icon, a graph and a table.

A controller knows a remote server name on which its model resides and sends a message selected by a user's operation to the remote server. Therefore, users do not have to remember the server name where the particular object exists.

5. Implemented distributed computing abilities of TISAS

New TISAS consists of two programs, i.e., a client program named `tisas` and a server program named `tisas-server`. As `tisas-server` is a PVM application, it should be installed in the PVM application directory and can be automatically started by PVM. When we start a TISAS session, we usually type the command `tisas` to connect to the default server program on the same computer. If we specify a server computer name with the client command like `tisas server_name`, the client program starts the server program on the destination computer `server_name` and tries to connect it.

The TISAS client program has two windows. One is for controlling the TISAS GUI process (an upper left window of Fig.2), and another is a work space window in which the history of a data analysis is displayed (a lower left window of Fig.2). One work space window displays a history of the data analysis about one data. If we want analyze other data, we can use new work space window by clicking the `Create work space` button.

When we start a data analysis, we usually specify a data file to be analyzed. If we select the **Load data** item of a pull-down menu of the work space window, a data file name and a remote server name selection window appears (a right side window of Fig.2). The right side of this window is to specify a server computer name and the left side is to select a data file name. Names of available servers are displayed on the lower right of this window. We can select a server computer from this list or by typing a computer name directly. When we click the **Accept** button on this window, TISAS server program is invoked on the specified computer automatically and displays data file names on the server computer on the left side area. When we click the **Select** button on this window after the data file selection, the server program load the file and send a message to show a time series icon to the view.

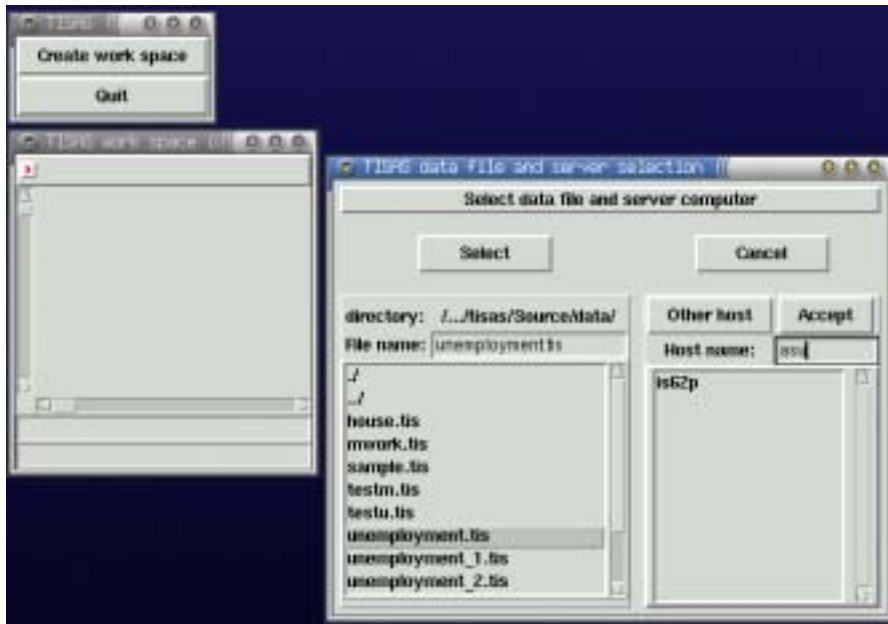


Fig. 2. A data file name and a server computer selection window

The data, which was read by the TISAS server, is displayed in the work space window as an icon. At this time, the server program creates an instance object of time series which is the model and the client program creates the view for an icon and the controller. They begin communicating mutually. Then, an icon of the data file name is displayed in the work space window. In Fig.3, TISAS reads a data file named **unemployment**, and displays an icon labeled **unemployment.tis**. This time series, which is given in Vandaele (1983), is monthly unemployed number of the United States from January 1948 to February 1975 and consists of 326 observations.

If we point an icon of the dataset on the work space window by the mouse, a pop-up menu appears (Fig.3). **List and/or Transform** item of the pop-up menu displays a table and we can transform the data using an upper right window of Fig.3. **Plot** item draws a time series plot on another window like a lower right window of Fig.3. These three forms,

i.e., the icon, the table and the plot, display one dataset on a remote server computer by different views.

The server name on which the data is stored is displayed at the bottom of the work space window when we put the mouse on an icon. In Fig.3, for example, the server computer name `is62p` and a brief explanation `Time series` of the `unemployment` icon are displayed.

The `Move to other host` item moves the time series represented by an icon to a remote server computer. If we select this menu item, we are required to type a remote server name and a file name on the server computer in a window, which is similar to the above mentioned file and server computer selection window. When we generate new time series, for example, by data transformation, we can also specify a remote server name on which new time series are stored. Calculations concerning to the object on the remote server will be performed completely on the same remote server, then new generated objects will also be stored on it, although icons for these objects are displayed in the same work space window. This means that future computational loads for data objects are distributed to the remote computer. We do not need to remember the correspondence between an icon and a remote server computer. Although these functions for distributed computing are simple and limited, they are easy to use and sufficient for improving the efficiency of TISAS, as will be shown in the following example.

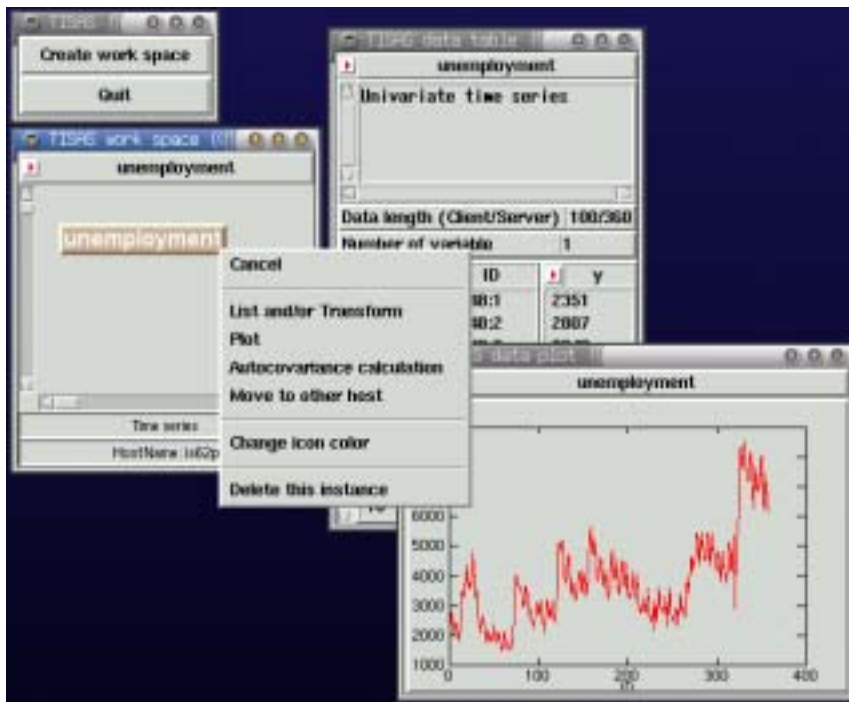


Fig. 3. Showing data by an icon, a table and a graph

At first, we generate three data transformations from `unemployment` data as written in Vandaele (1983). `unemployment_1` is a first order seasonal difference of original data

(unemployment). `unemployment_2` and `unemployment_3` are a first order difference and a second order difference of `unemployment_1`, respectively. We save each new data on different three servers. Fig.4 shows that `unemployment_4` and `unemployment_5` are created by log transformations of `unemployment_2` and `unemployment_1`, and calculate basic statistics and fit AR models to `unemployment_1`, `unemployment_2` and `unemployment_3`. When each calculation on server computers completes, an icon which represent the result appears on the work space window. We can not operate an icon until the result is returned. However, we can perform calculations simultaneously for several data because they are on different computers.

Here, we show an example of the performance improvement in our environment. When all calculations are performed on one computer, calculation to fit an AR model to `unemployment_1` needs about 1.5 seconds, and same calculation for `unemployment_3` needs about 6 seconds. When calculations are performed using distributed computing explained in this section, the first calculation need about 2 seconds, and the second calculations still need about 2 seconds.

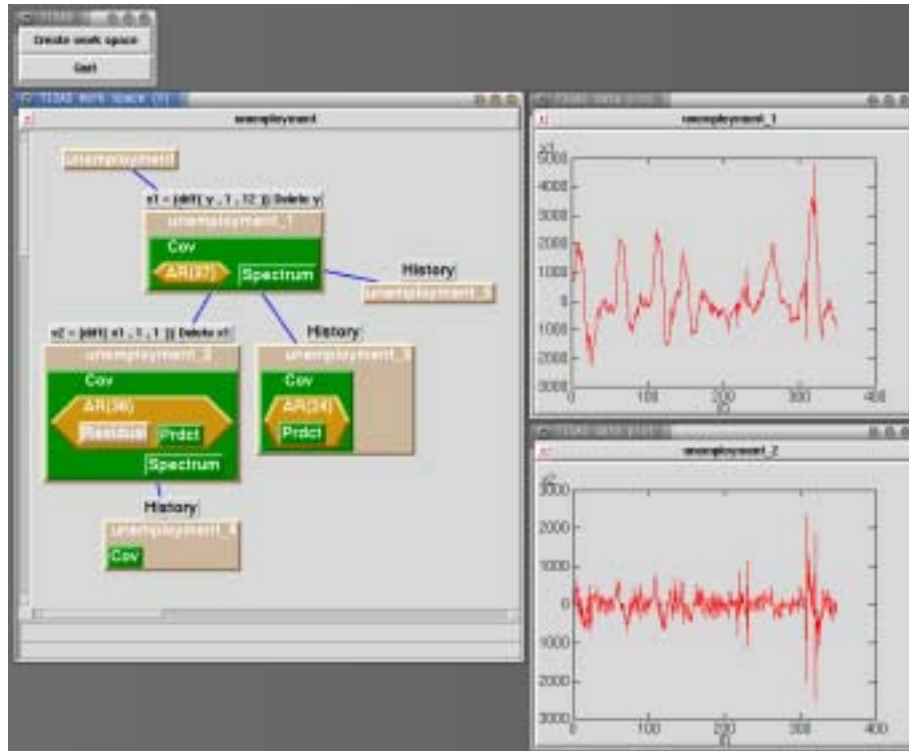


Fig. 4. An example of a TISAS process

We design TISAS in order to keep the network traffic as little as possible. The server program maintains calculation results, and sends them to the client program if necessary. The client program has a cache for a dataset on servers not to send the same dataset twice. Besides, TISAS tries to keep from sending a large dataset. For example, as we do not always

need to see all dataset values, TISAS confirms a user whether it is really necessary to see all dataset values or to see a part of them before executing the `view` command, which requires the data transfer from the server to the client.

TISAS client and server programs can run on almost all UNIX operating systems. The GNU C++ compiler¹⁶, Tcl/Tk, PVM and TkPVM are required to be built from the source code. TISAS can be installed in the home directory in case a user is not a system administrator. New TISAS is not available on Microsoft Windows, because TkPVM does not work on it.

6. Concluding remarks

Original TISAS was designed to realize an experimental new GUI for time series analysis. However, when we analyze large datasets or generate many data, statistics and models, we noticed that it became slow. In this paper, we propose to improve that problem by adopting distributed computing technologies.

Usually, distributed computing in scientific calculations is mainly used for increasing available CPU powers. In statistical calculations, however, increasing available memory and storage by distributed computing is rather important.

This version of TISAS works well on the Intranet use, but lacks firm security abilities for the Internet use. The security of TISAS is based on PVM, whose security mechanism is the same as the R command (Berkeley Remote command) on UNIX, and is not safe enough in the Internet use. We note that new version of PVM can use ssh (Secure Shell) instead of rsh which is one of the R command, and may be safer.

There are another problems for the Internet use of TISAS: server down problem. TISAS, rather TkPVM, can not handle sudden down of the server or network trouble rightly. Possibility of such trouble is higher in Internet than in Intranet.

The installation and the maintenance of TISAS client and server programs are not difficult, for example, `make` command of UNIX is available for the installation. A statistician, who is not a system administrator of computers, can install and use them on their user directory. The server does not need to be started beforehand. Therefore, users can use many computers easily once they installed TISAS as servers.

In case statisticians have several computers, which are connected to a network, and their problems require heavy calculations, the approach used in TISAS might be effective.

REFERENCES

- Akaike, H. and Nakagawa, T. (1989). *Statistical Analysis and Control of Dynamic Systems*, Dordrecht: Kluwer Academic Publishers.
- Casanova, H and Dongarra, J. (1996). *NetSolve: A Network Server for Solving Computational Science Problems*, University of Tennessee Computer Science Technical Report, CS-96-328, May 1996, paper submitted to SC96.
- Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R. and Sunderam, V. (1994). *PVM: Parallel Virtual Machine - A User's Guide and Tutorial for Networked Parallel Computing*, Cambridge: The MIT Press.
- Hartung, K., Muench, S., Schomaker, L., Guiard-Marigny, T., Le Goff, K., MacLavery, R., Nijtmans, J., Camurri, A., Defee I. and Benoit, C. (1996). Development of a System Architecture

¹⁶<http://gcc.gnu.org/>

for the Acquisition, Integration, and Representation of Multimodal Information, *Report of the ESPRIT Project 8579 MIAMI, -WP 3-*

- Härdle, W., Klinke, S. and Müller, M., (1999). *XploRe - Learning Guide*, Berlin: Springer.
- Kötter, T. (1997). Implementation of networked applications for statistical computing. in *Bulletin of the International Statistical Institute, 51th Session Proceedings Book 2*, 47–50.
- Liu, L.-M., Chan, K.-K., Montgomery, A. L. and Muller, M. E. (1995). A system-independent graphical user interface for statistical software, *Computational Statistics & Data Analysis*, **19**(1), 23–44.
- Morrison, M. and Ablan, J. (1997). *Teach Yourself More Java in 21 Days*, New Jersey: Prentice Hall.
- Nakano, J. and White, K. J. (1997). A WWW Interface for the Statistical Package SHAZAM, *Proceedings of the Institute of Statistical Mathematics*, **45**, 41–47.
- Nakano, J. (1998). Graphical User Interface for Statistical Software Using Internet, in *COMP-STAT 1998 Proceedings in Computational Statistics* (ed. Payne, R. and Green, P.), 407–412. Heidelberg: Physica-Verlag.
- Sato, M., Nakada, H., Sekiguchi, S., Matsuoka, S., Nagashima, U. and Takagi, H. (1997). Ninf: Network based Information Library for Global World-Wide Computing Infrastructure, *Proceedings of HPCN '97 (LNCS-1225)*, 491–502.
- Ousterhout, J. K. (1994). *Tcl and the Tk Toolkit*, Reading: Addison-Wesley.
- Pinson, L. J. and Wiener, R. S. (1988). *An Introduction to Object-Oriented Programming and SmallTalk*, Reading: Addison-Wesley.
- Vandaele, W. (1983). *Applied Time Series and Box-Jenkins Models*, Orlando: Academic Press.
- West, R. W. (1995). Statistical applications for the World Wide Web. in *Bulletin of the International Statistical Institute, 51th Session Proceedings Book 2*, 7–10.
- Yamamoto, Y., Nakano, J. and Tamura, Y. (1995). Porting TIMSAC72 for MS-Windows to UNIX, *Proceedings of the Institute of Statistical Mathematics*, **43** (2), 285–292 (in Japanese).
- Yamamoto, Y. and Nakano, J. (2000). A time series analysis system using visual operations. SFB 373 Discussion Paper **32**. (<http://sfb.wiwi.hu-berlin.de/>)