

## PARALLEL APPLICATION LAUNCHING ON COMPUTER CLUSTER

*Bing-Lin Yang*

*State University of Moldova*

### INTRODUCTION

One of the most important usages of computer is scientific calculations. People make use of the high performance of computer to implement the simulation in some scientific research fields, for example, the simulation of biological process. For improving the efficiency of calculating, a new system named computer cluster was built. Computer clusters are usually deployed to improve performance and/or availability over that provided by a single computer, while typically being much more cost-effective than single computers of comparable speed or availability. We developed a Web-Client as a part of the three-level project, named Visual Membrane Petri Nets (VMPN) for practicing launching parallel application on computer cluster. Here, we elaborate Web-Client that offers a graphical user interface for VMPN parallel software tool. The first level of VMPN is a client level. At this level, there are two types of client, one is the normal client (User must install the client software on computer before using it), another is a Web-Client (User can use this system by simply launching browser); the second level of this system is an application level. One application server (Web-service) works on this level, the purpose of this level is collecting users' requisitions, and analyzing, organizing these requisitions in order to send to cluster to compute. The application server also receives the results from cluster and divides them in order to give different clients proper result. The third level is the computation level, since membrane computation needs large computing time, so we use Rocks cluster with 52 processors to perform it. There are several technologies used in the Web-Client such as: Web-Service, Database (MySQL), PHP, SOAP, dynamic web-page technique and so on.

### 1. THEORETICAL PRINCIPLES

Computer is an indispensable tool in the current world. The speed of computation of computer is pivotal, for some application fields that need a high-speed computation such as weather forecast, scientific calculation and engineering calculation, mainframe computers are used.

However, mainframe computer are very expensive and hard to maintain, thus small consumers have no chance to use giant computer or mainframe computer on their work. However, low-cost personal computers are widely used everywhere over the world. As a result, a new technology named computer cluster were created. Computer cluster is a group of linked computers, working together closely so that in many respects they form a single computer. The components of a cluster are commonly, but not always, connected to each other through fast local area networks. Clusters are usually deployed to improve performance and/or availability over that provided by a single computer, while typically being much more cost-effective than single computers of comparable speed or availability. Computer cluster is a system based on parallel computation theory.

Parallel computation is a modern technique used for processing some computations that need a large amount of computation time. Because the limits of computational capability of single processor. We need to pay a lot of time for computing some complex operations, thus people want to divide one large computation into several small computations and calculate them in synchronization. We need to assemble the results when all of the partial computations have been done, so as to we can obtain the result in an acceptant time. In most cases, that can be seen is computer cluster as an implementation of parallel computing.

MPI (Messages Passing Interface) is a good tool for realizing a parallel program. Our parallel computing system uses MPI to implement parallel computing. The Message Passing Interface Standard (MPI) is a message passing library standard. The goal of the Message Passing Interface is to establish a portable, efficient, and flexible standard for message passing that will be widely used for writing message passing programs. As a result, we didn't need to consider too much about how to divide our high computation-time-consumed operation to different nodes when we were developing our program.

### 2. VISUAL MEMBRANE PETRI NETS

There is an instance of parallel computation has been done by us; this implementation is a

medical assistant managing and analysis system which uses parallel computation based on computer cluster to simulate membrane evolution. This application uses three-levels-system theory.

The first level is the client level (Figure 1). At this level, there are two types of client, one is a normal client (User must install the client software on computer before using it), another is a Web-Client (User can use this system by simply launching browser); the second level of this system is an application level, one application server (Web-service) works on this level, the purpose of this level is collecting users requisitions, and analyzing, organizing these requisitions in order to send to cluster to compute.

The application server also receives the results from cluster and divides them so as to give different clients proper result. The third level is the computation level, since membrane computation needs large computing time, therefore, we use computer cluster to perform it. We use a rock cluster with 52 processors as the third level.

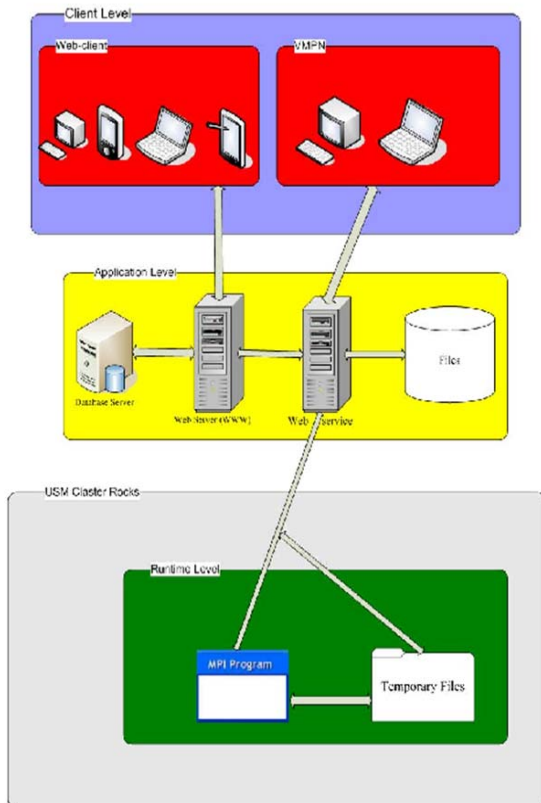


Figure 1.

The Visual Membrane Petri Nets (VMPN) is a parallel software tool for visual representation, formalization and simulation of MPN-models. This tool allows verification and validation of behavioural properties of membrane systems and their visual interactive discrete-continuous simulation. VMPN is a window-based, object-

oriented parallel software tool, in which elements typical of hybrid Petri net models (discrete-continuous places, transitions, arcs, etc.) are manipulated under the assistance of basic syntactical rules that prevent the construction of incorrect models.

Visual simulation is very useful during the early stages of the construction of the model since it represents a powerful tool for its debugging. Moreover, the animation of the correct simulation model may provide important insights on the behaviour of the actual system.

The dynamic graphical facilities are used to visualize the movement of the tokens (fluid), to provide snapshots of the model, and to represent the membrane system evolution. A screensnapshot of a membrane model is represented in Figure 2.

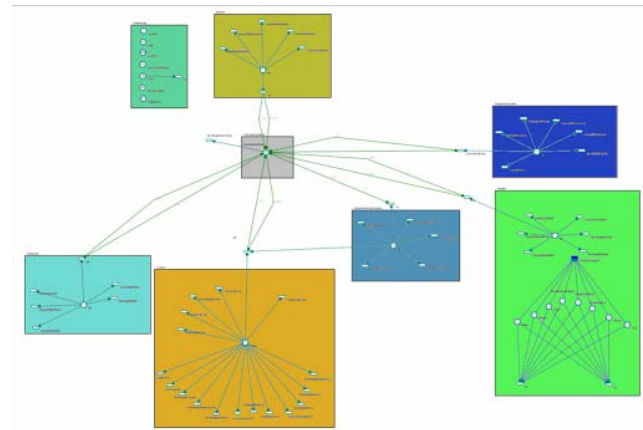


Figure 2.

### 3. WEB-CLIENT OF PETRI NETS

The Web-Client of VMPN is the front-end program that makes user can work on a computer cluster with a friendly and easily understood interface (Figure 2). This Web-Client is designed for clinicians. There are two main parts of Web-Client we developed for composing Web-Client of VMPN, one is the management part, and another is the simulation part. In management part, doctor can input related data after inspecting a patient, Web-Client will store those data in a database (in our project, we use MySQL as database). After the completion of a series of examinations, clinician can use Web-Client to do simulation in order to obtain a preview of the growth of diseases. From a technical point of view, we use PHP programming language to produce HTML dynamically and to get patients data from MySQL database to produce data-message with the same values in XML format in order to Web-service can receive these data correctly. Web-service will return the result of

simulation computing to Web-Client in XML format after computing on rocks cluster. Finally, Web-Client generates 2D analysis charts and 3D analysis charts for clinicians.

The usage and mechanism of Web-Client:

1. Register a new user as a clinician.

We added strategy of checking users input, user only can input letters and numbers as user-name and password as well as the strategy of input two times password avoiding user input a wrong password carelessly. A database is used for storing users' information including user-name and password. Password string is encrypted by MD5 algorithm and stored in the database.

2. Managing patient.

Users (clinicians) must add some data of inspections for patients by using managing part of Web-Client before processing simulation. Users can find a Manage Patient link on the current web page after login. Click Manage Patient super-link to enter managing the patient page (Figure 3).

The first page of Managing part is a patients ID input and searching page. Since we use strategy of ID, so the patients ID is the key index in database and cannot be duplicated. For the new patient, clinicians need to click Add a new patient link at bottom of this page to register a new patient in system.

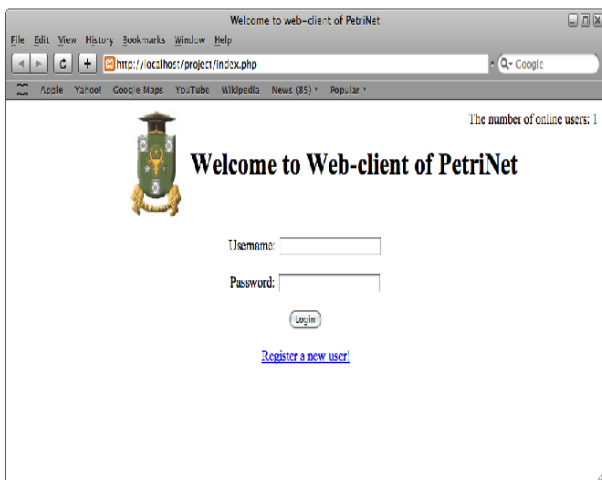


Figure 3.

On the register page, clinicians should do input the basic information (such as first name, last name, gender, age and occupation) for this patient. For an existent patient, clinicians can find his or her ID by using searching ID function on this page, we offer two ways of searching ID, by name and by age, which can help clinicians to find patients ID easily. Input patients ID and click OK button entering the main managing page if clinicians have already known the ID of this patient.

The main managing page is a framework with three pages, the title page, inspection selecting page and content page. The current operating clinicians name and patients name will be displayed on the title page, which can ensure clinicians will not do some mistakes of garbling patients.

Of course, clinicians could change to manage another patient by clicking Select another patient link at bottom right corner of the title page title. The inspection selecting page is placed left of framework, we offer models of "BASIC INFORMATION", "ROUTING INSPECTION", "ROUTING ANALYSIS OF BLOOD", "BIOCHEMICAL ANALYSIS OF BLOOD", "ROUTING ANALYSIS OF URINE", "PRESSURE" and "ALCOHOL CONSUMPTION" for the current version (Figure 4).

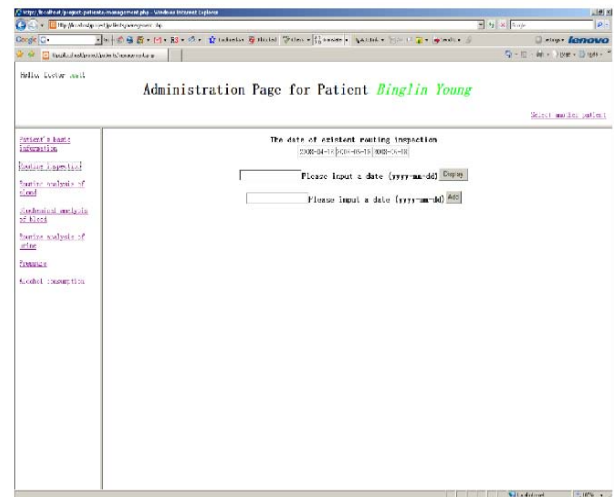


Figure 4

(a) Patients' basic information.

Click the "Basic Information" link on left "Inspection Selecting" area to open the content page of patients' basic information on right area. Clinicians can modify the patients basic information that has been stored after a clinician registered this patient by clicking modify button or delete this patient by clicking delete button. Due to the strategy of dynamic creating tables of different inspections is used in this system, so clinicians will delete (drop) all of tables of this patient in database if a clinician clicks delete button, all of information of this patient will lose and cannot be recovered.

(b) Routing inspection.

System will create tables for each inspection of each patient. In such inspection data recording tables, date of inspection is the key index and cannot be duplicated. The inspection table-names are instituted of patients ID + the name of inspection.

For instance, the name of table of the routing inspection belonged to the patient with ID 1 is “1 routing inspection” in database. For managing these table names, system uses a static table named “table names” to store inspection tables dynamically. The different inspection table-names will be stored in this table when they are created. These tables comply with the standard of BNF-3.

Clicking “Routing Inspection” super-link users could see the first page of content pages of routing inspection. System can check whether data existent or not existent, if existent, the date of routing inspection will be displayed automatically, but if no existent inspection, system will only display two command line on page.

One is for displaying data of inspection; another is for adding a new datum of inspection. Both require clinicians to input a date of inspection in format year-month-day.

If no data were input before, please input a date in second input-box and click “add” button to open “Adding data” page.

There are many items of related inspection will be required to input, after input, click “Add” button to store data into database. If serial data of inspection had been input by clinicians before, clinicians can find the wanted date from the above table easily and input the date into first input-box, then click “Display” button to review the data of inspection.

Clinicians also can, of course, modify or delete these data from “display data page”. For this initial version, there are only six types of inspection allowed to use.

Remain models are similar to this model, because all of them are inspection models. Consequently, other models will not be illustrated here.

### 3. Simulation analysis pages

Clinicians should come back to simulation analysis pages to do simulation and analysis after administrating data of various inspections.

#### (a) Getting model by name

This is a very simple page, clinicians just need to type the name of a model on Web-service in input-box and click “Get” button. Web Client will get the model from Web-Service.

The SOAP technique is used to implement this process here. The key code below:

```
$client=new Soap
Client("http://localhost:1690/ServiceManag
er.asmx?WSDL");
$params=array("modelName"=>"$modelName");
$result=$client->GetModelByName($params);
$resultString=$result->GetModelByNameResult;
```

(b) Modify token-count and/or generate a new model.

Web-Client can analyze the data of model from Web-service and display them in a simple form automatically. Clinicians can modify these data directly, and if a clinician wants to save the new data as a new model, he or she just simply types a new name in New Model Name input-box and click “Simulate button” to continue.

Next time, if a clinician wants to review the process of model with new data, he or she can just input the new model name on “Getting Model by Name” page.

The XML phrasing is used in above functions. Web-Service returns data in XML format, so Web-Client has to analyze the format of XML string and show an easily understood datum to user.

After modifying data, Web-Client has to convert data back to XML format in order to Web-service can understand the string from Web-Client properly. The XML Parser Functions and String Functions of PHP are used for implementing this process. The selecting models page will be displayed in lockstep.

#### (c) Selecting locations page

There are many locations in a model, so web-service analyzed the model and give user a chance to select which locations needed to be simulated. This page is a simplistic web-page; all of locations are listed in form of check boxes. Select which locations needed to be analyzed and go on.

#### (d) Show charts

This is the final step in simulation. Locations that had been selected were executed on cluster in the parallel mode. Moreover, Web service got these results and returned them to Web-Client in XML format. In this case, the XML string only contains two sets of values, the values of token-count and the time of computation of related token-count. Web-Client generates charts according to the data of simulation result and display these chart for users.

Three types of chart could be generated, first of all, the normal 2D coordinate chart of each selected locations will be generated, and then the integrated 2D Chart will be displayed. The integrated 2D chat integrated all of selected locations changing lines in one chart. Finally, a 3D chart can be displayed in order to give user a more exhaustive way to analyze data (Figure5).

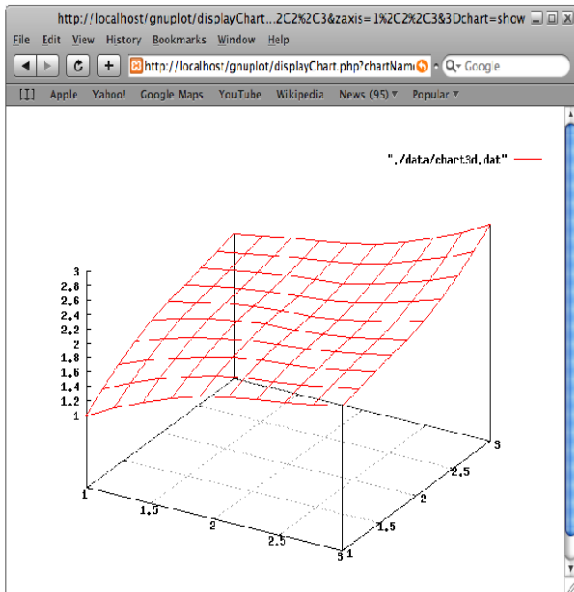


Figure 5.

#### 4. CONCLUSION

While developing the whole system, security mechanism was always put in our consideration. Users' passwords are encrypted by MD5 arithmetic before saving into database, so nobody can know the real password of users. Avoiding some unwelcome connection entering system by typing interruptive URL, we use session technique in our system and added some special Java-Script codes in echo pages. Thus no matter URL of which page (not included homepage) unexpected guests typed in browser, system can navigate browser to the login page of Web-Client. Unauthorized users cannot use Web-Client.

The advantages of Visual Petri-Nets system are that we made the best use of parallel calculating with Rocks cluster in Moldova State University. Clinicians can manage many patients and do simulations synchronously. The advantage of high performance computing is drawn out. The two ways (normal client and Web-Client) of using this system offer the users (clinicians) great convenience. Clinicians even can work on his or her PDA or smart phone in future. The Web Client was built following the standard of W3C, so users can use Petri-Nets no matter which browser is used.

#### Bibliography

1. **John L. Hennessy and David A.** *Computer Architecture: A Quantitative Approach*, 3rd edition, 2002. Morgan Kaufmann, ISBN 1558607242, p. 43.

2. **Blaise Barney.** *Introduction to Parallel Computing*, November 09, 2007.

3. **Profir A, Gutuleac E., Ciurganov D., Opinca C., Danu S.,** *Modeling of molecular genetic mechanisms involved in the development of cardiovascular diseases using Visual Membrane Petri Nets.* "Medical-Surgical Journal" of Iasi, "Gr. T. Popa" University of Medicine and Pharmacy, Iasi, Romania, November 2007, p. 157-160.

4. **Gutuleac E.** *Descriptive Timed Membrane Petri Nets for Modelling of Parallel Computing.* *International Journal of Computers, Communications & Control*, I(3). Agora University Editing House, Oradea, Romania 2006, p. 33-39.

5. **Ciurganov D., Profir A., Gutuleac E., Opinca C., Young B., Lisnic A.** *Modeling of the clinical evolution of type 2 diabetes applying the concept of membrane systems.* 4th ISCB Student Council Symposium, Toronto, Canada, July 2008.

6. *MySQL 5.0 Reference Manual*, May 20, 2008.

7. **Rasmus Lerdorf, Kevin Tatroe.** *Programming PHP*, June 27, 2007.

Recommended for publication: 28.02.2011