Journal of Engineering and Applied Sciences 11 (5): 964-966, 2016
ISSN: 1816-949X

Matrix Multiplication Algorithm for Distributed Computation Network for Cooperation Between Computers Simultaneously

Sadeq Asghari
Department of Mathematics, University of Payame Noor, Tabriz, Iran

Abstract: PVM (Parallel Virtual Machine) is a software box which enables us to make a parallel virtual machine that includes a network of homogenous and non-homogenous computers. The general purpose of PVM system is to create cooperation between network computers for doing the distributed computation in simultaneous or parallel form. In this study we intend to do the matrixes multiplication problem in parallel form, for doing this work, we need a network of computers that due to the ease of work we use of wireless network (ad hoc). Wireless network (ad hoc) is a non-concentrated wireless network that each node of network (network member) sends data for next nodes and the decision for sending the data to the next node is made in dynamic form according to the quality of connection to the network for that node by the node itself. This is the difference point of this kind of network with wired networks in which routers do the operation of routing and sending the data. At first we launch ad hoc network with several computers, then we run the matrix multiplication algorithm with PVM system on this network. At the end of the study, the results obtained from parallel processing and processing in serial form have been studied.

Key words: Matrix multiplication, algorithm, distributed computation network, cooperation, computers simultaneously

INTRODUCTION

PVM (Parallel Virtual Machine) is a system for message exchange which enables us to make a virtual parallel computer by using of a network of computers and it provides this possibility that a homogenous set of computer systems with various architectures on a network like parallel virtual machine to be observed (Al Geist *et al*., 1994). PVM manages the routing of all messages, data conversion and scheduled works on a network with computers with incompatible architectures. The computation model of PVM is simple and common and includes a vast domain of programs structures. Programming the intermediaries is easy. Therefore, the program is permitted to be implemented simply. User writes his/her program in the form of a set of simultaneous works. Works through a governorship library are accessed from routines to PVM sources. These routines permit the works to be started and ended on the network and simultaneity is created between the works. The main exchanges of PVM messages are homogenous-oriented operations and necessitate powerful structures for buffering and transmission. We may assume PVM works like desired controls with dependent structures.

In other words, in any point of execution, the current program of each existing work may be started or ended with other works or computers may eliminate/add from/to virtual machine. Each process may become simultaneous with other works.

How does PVM work?: The most important purposes for PVM are error tolerance, scalability, lack of homogeneity and portability and PVM can make relation between the computers from different kinds in a unique session and it has been also in relation with the minimum amendments according to each one of Unix indexes or operational system with comparable facilitations and the programming interference is simple but complex. Sockets are applied for inter-processing relations and each host in a group of virtual machine can be connected to another host directly and through TCP/UDP and other protocols, the general need of IP relations can be accompanied with elimination of special message routes and using of PVMDs for next messages, some of multi-processor machines don't have special socket about processing nodes but they have work procedure in the form of end to start or back to front.

MATERIALS AND METHODS

Advantages and faults of PVM: The system of parallel virtual machine has advantages and faults that some of its advantages can be mentioned as follows:

- Ability of connection of networks with non-homogenous machines
- Using of C/Fortran and C++ functions
- Accuracy in data conversion and the problems of low level losses
- Ease of installation and doing the regulations

- Portability (independent from hardware)
- High error tolerance
- Flexibility
- Scalability

Some of the faults that can be mentioned for it are as follows:

- Programmer should implement all details of paralleling explicitly
- Difficulty of troubleshooting operations
- Depending on architecture and implementation, PVM may act slower than passage of other messages
- PVM unlike MPI isn't standard

PVM capabilities

User's regulations for clients set: Computation programs run some works on a set of machines that have been selected by user for executing PVM program. Both CPU and multi-processor hardware (including common memory and distributed memory) may be some parts of clients.

Clarity of access to the hardware: Program may see the around hardware as the properties of a set of virtual processing elements or may select facilitations for special machines in the clients set for special computation works.

Computation according to the processes: Paralleling unit in PVM is a work (often and not always, a Unix process) and it is a sequential independent control string which behaves between computation and communication alternately.

The model of explicit message exchange: It is a set of computation works that each one does a part of work of a program by using of data, functions or a combination of divisions by sending and receiving the message explicitly. The messages size is limited only by a part of memory size in access.

Support of non-homogeneities: PVM system supports non-homogeneities of machines, networks and programs. For example, for message exchange, PVM permits the messages which include more than a kind of K-data to be exchanged with machines that have different kind of data demonstration.

Support of multi-processing: PVM uses of facilitations of multi-processing message exchange which has hardware advantages. The structure of PVM system is often formed of two parts as follows:

PVM daemon and PVM library: First part is a daemon which is called PVMD3 and it is sometimes summarized to PVMD, it is placed on all computers for making virtual machine. An example of daemon program is mail program which is run in the back stage and manages all input and output e-mails.

PVMD3 has been considered in this manner that each user who has valid entrance can install this daemon on his machine. When a user intends to run a PVM program, at first makes a virtual machine with PVMD, then PVM program from a Unix message can be run on any client. Several users can regulate overlapping of virtual machines and each user can run the sequential program of PVM simultaneously.

The second part is a library of PVM functions which includes a full set of fundamental functions which are needed for relations between the works of a program. This library includes callable functions of user for exchanging the processes overlapping message, coordination of works and amendment of virtual machine.

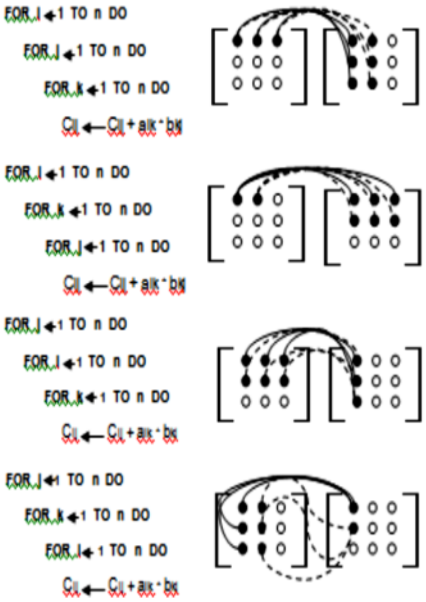
All PVM works are specified by a natural number (TID) and the messages send/receive from/to TID and each TID during the parallel machine is unique that these numbers are determined by local PVMD and aren't available for user. Although, PVM encodes the information for each TID but user expects that TIDs to behave as unclear specification.

For running a program, a user often duplicates a copy of a work usually (master or initiating) manually from a machine to inside the scale set. This process is started in order by other works of PVM and finally the results are put in a set of active works, then for solving the problem, computation is done locally and the messages are exchanged with other works.

RESULTS AND DISCUSSION

Ad hoc networks: Ad hoc networks are said to the instantaneous or temporal networks which are considered for a special purpose. In fact, they are wireless networks that their nodes are movable. Ad hoc wireless network is a non-concentrated wireless network that each node of network (network member) sends data for next nodes. And the decision of sending data for next node is made dynamically according to the quality of connection to the network for that node by the node itself. This is the difference point of this kind of network with wired networks in which routers do the operations of routing and sending the data, in the managed wireless networks that a special node which is recognized as the access point manages the relation of other groups (Fig. 1).

Matrixes multiplication problem: We want to implement the matrixes multiplication problem with parallel method. If the intended matrix is 3×3, therefore for computation of each element of response matrix, we should do three multiplication acts and finally we should add the products.



```
FOR i ← 1 TO n DO
    FOR j ← 1 TO n DO
        FOR k ← 1 TO n DO
            Cij ← Cij + aik * bkj

FOR i ← 1 TO n DO
    FOR k ← 1 TO n DO
        FOR j ← 1 TO n DO
            Cij ← Cij + aik * bkj

FOR j ← 1 TO n DO
    FOR i ← 1 TO n DO
        FOR k ← 1 TO n DO
            Cij ← Cij + aik * bkj

FOR j ← 1 TO n DO
    FOR k ← 1 TO n DO
        FOR i ← 1 TO n DO
            Cij ← Cij + aik * bkj
```

Fig. 1: Different combinations for multiplication of two matrixes, n×n

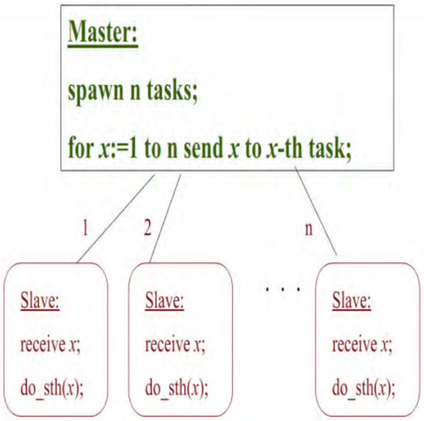


Fig. 2: Schematic design of problem division

For increasing the speed, the computation related to each element can be given over to a processor, that in this example we need 9 processors. Each processor after doing computation sends the result to the master processor, master processor demonstrates the final result. Now if we demonstrate the multiplication of two matrixes in the following form:

$$\begin{bmatrix} a00 & a01 & a02 \\ a10 & a11 & a12 \\ a20 & a21 & a22 \end{bmatrix} \times \begin{bmatrix} b00 & b01 & b02 \\ b10 & b11 & b12 \\ b20 & b21 & b22 \end{bmatrix}$$

The acts done by each processor will be as follows:

$$p1 = a00 \times b00 + a01 \times b10 + a02 \times b20$$
$$p2 = a00 \times b01 + a01 \times b11 + a02 \times b21$$
$$p3 = a00 \times b02 + a01 \times b12 + a02 \times b22$$
$$p4 = a01 \times b00 + a11 \times b10 + a12 \times b20$$
$$p5 = a01 \times b01 + a11 \times b11 + a12 \times b21$$
$$p6 = a01 \times b02 + a11 \times b12 + a12 \times b22$$
$$p7 = a20 \times b00 + a21 \times b10 + a22 \times b20$$
$$p8 = a20 \times b01 + a21 \times b11 + a22 \times b21$$
$$p9 = a20 \times b02 + a21 \times b12 + a22 \times b22$$

The stages of doing the work: At first, we install ad hoc wireless network with 10 lab-top computers. One of the computers is as master and 9 other computers are as slave, then with one of the programming languages that PVM system supports like C or C++ or Fortran, we divide the matrixes multiplication problem into sub-problems and send each part to a computer in the network (Rytter and Pagouritzis, 1994). This design is demonstrated in Fig. 2 schematically.

```
Master
for i = 1 to n send I to i-th task
for i = 1 to n {receive x from i- th task
print result

Slave
Receive i from parent (master)
For i = 1 to n
For j = 1 to n
For k = i to n
Cij = Cij+Bik×Akj
Send Cij to parent
```

CONCLUSION

For multiplication of matrixes with high dimensions by using of a processor (computer), we have the most time of doing the computation that with increase of the processors number, the time of doing the computation is optimized (accelerated). Now if the number of processors becomes more than an extent, due to the existence of time-sharing and overload in relations between computers, the time of computation will be increased.

REFERENCES

Al Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek and V. Sunderam, 1994. PVM: Parallel Virtual Machine-A User's Guide and Tutorial for Networked Parallel Computing. MIT Press, Cambridge.

Rytter, W. and A. Pagouritzis, 1994. Parallel/Distributed computing using Parallel Virtual Machine (PVM). University of Liverpool, Department of Computer Science. http://cgi.csc.liv.ac.uk/~igor/COMP308/ppt/Lect_22pd.