Achieving Organism Clustering Analysis by Using PC Cluster Architecture with MPI Techniques

Williams Tsai
Department of Information Management
National Taitung University
684, Sec. 1, Chung Hua Rd., Taitung, Taiwan 950, R.O.C.

Neng-Mu Shih
Department of Art and Crafts Education
National Taitung University
684, Sec. 1, Chung Hua Rd., Taitung, Taiwan 950, R.O.C

Kun-Lin Hsieh*
Department of Information Management & Research Group in Systemic and Theoretical Sciences
National Taitung University
684, Sec. 1, Chung Hua Rd., Taitung, Taiwan 950, R.O.C.

Abstract: - During the competitive environment, how to sufficiently and efficiently utilize the computational resources under the consideration of cost had became an important action to be done by most practitioners. For the recent years, the PC cluster architecture had been applied into many applications, especial for the computational resources. In this study, we will introduce a case study in Taiwan owing the issue based on PC cluster architecture with MPI techniques. This case is an application of clustering technique to DNA analysis. Not only the hardware/software architecture of PC cluster had been constructed, but the clustering algorithm based on such cluster architecture was also proposed in this study. And, the rationality of the proposed architecture and algorithm can be demonstrated well in this study.

Key-Words: - PC cluster, Self-organizing feature map neural network (SOMNN), computational resource, Message Passing Interface (MPI).

1. Introduction

Personal Computers had been viewed as the basic and necessary equipments at university. The fact that computational resources of CPUs for PCs can keep idle state was frequently met. Even though we execute several application programs on a single PC at the same time, the computational resources of CPUs still can keep a certain degree on the current operating system. Hence, such status will be viewed as resource waste under such viewpoint. Besides, we had known that parallel computing can be performed well on a workstation or a super computer, which are the expensive devices. Generally, the cost or the necessary investment of it will limit the applications for most practitioners, especial for those practitioners with resource limitation. However, there are many applications, e.g. pattern recognition or 3D computation for the current requirements and future development. Hence, how to choose an approach with lower cost will be an important issue for those practitioners and it will be the primary motivation of this study.

In this study, we demonstrated a successful case applying the PC clusters to performing the complicate computation at National University in Taiwan. This case is the clustering analysis for organism based on codon usage in DNA (Ghosh, 2000; Karlin & Mrazek, 1996). And, it is a working research project at the Department of Information Management at Taitung University, Taiwan. The project team lack of the financial support and the cost will be the important limitations of it. And, the project team applied a PC cluster architecture under a teaching laboratory into perform the application. Under such considerations, we will pay our attention to develop a rational and feasible architecture to achieve the analysis of organism clustering based on PC clusters architecture. Herein, the self-organizing feature map neural network (SOMNN) (Kohonen, 1984; Kohonen, 1982; Ye, 2003) will be the clustering technique to be chosen in this working project. The following sections will describe such successful case in detailed. The experimental architecture of PC clusters for hardware and software, and the clustering algorithm based on PC clusters will be also clearly described in this study.

2. Background Introduction

2.1. PC cluster

If we can utilize PCs interconnected by an Ethernet/Fast Ethernet for distributed computing, the networked environment is called PC cluster (Sterling, Salmon, Becker & Savarese, 1999). To implement parallel programming in a PC cluster, the major issue is the distribution of information among PCs. In order to implement distributed computing, it is necessary to choose a programming interface that helps programmer to distribute messages among PCs. For a Windows-based PC cluster, there are several programming interfaces or software packages available. Message Passing Interface (MPI), the standard of message passing programming libraries, provides portable function calls to C, C++, and FORTRAN programmers. Since it is designed to be open sourced and used with homogeneous computer clusters, Windows-based MPI packages can be free downloaded from the Internet. MPICH, a portable implementation of MPI, is available from Mathematics and Computer Science Division (MCS) (2003). This package is designed to work on multiple platforms. MPICH is a modification and extension to MCS’s work. This alternative of MPICH also includes an enhanced version called NT-MPICH that improves the portability and performance of message passing to Windows NT/2000 environment (Bemmerl, 2003). Bemmerl’s NT-MPICH is easy to install and has good performance on message passing. Therefore, in this study, we choose NT-MPICH developed by Bemmerl to carry out our experiments.

2.2. Self-Organizing Feature Map (SOMNN)

The architecture form of the SOMNN network is based on the understanding that the representation of data features might assume the form of a self-organizing feature map that is geometrically organized as a grid or lattice. In the pure form, the SOMNN defines an "elastic net" of points (parameter, reference, or codebook vectors) that are fitted to the input data space to approximate its density function in an ordered way. The algorithm takes thus a set of N-dimensional objects as input and maps them onto nodes of a two-dimensional grid, and it will result in an orderly feature map (Kohonen, 1982; Kohonen, 1990).

The components in SOMNN are the input layer and the topological map, a layer of nodes topologically structured. Every input node is connected to every output node via a variable connection weight. A layer of two-dimensional array of competitive output nodes is used to form the feature map. Figure 1 depicts the architecture of a classic SOMNN. The lattice type of array can be defined to be square, rectangular, hexagonal, or even irregular. The SOMNN belongs to the category of the unsupervised competitive learning networks (Hinton, 1989; Kohonen, 1982; Kohonen, 1990; Hsieh *et al.*, 2006). It is called competitive learning because there is a set of nodes that compete with one another to become active. In the SOMNN, the competitive learning means also that a number of nodes is comparing the same input data with their internal parameters, and the node with the best match (or it can be said as winner) is then tuning itself to that input, in addition the best matching node activates its topographical neighbors in the network to take part in tuning to the same input. More a node is distant from the winning node the learning is weaker. It is also called unsupervised learning because no information concerning the correct clusters is provided to the network during its training. Like any unsupervised clustering method, the SOMNN can be used to find clusters in the input data, and to identify an unknown data vector with one of the clusters. The detailed concept of SOMNN can be referred to Kohonen (1981, 1990).

3. Proposed Approach

In this section, the hardware/software environment of PC clusters will be defined well. And, the proposed SOMNN algorithm based on PC clusters is also expressed in this section. Under the cost consideration, we construct the PC clusters experimental environment with 5 PCs. The architecture diagram will be graphically depicted in Figure 1. As for the specifications of hardware/ software for those 5 PCs, it will be given as follows:

1.CPU: Intel Pentium 4 3.2Ghz
2.RAM: 512MB DDRII RAM
3.Capacity: 80GB HDD
4.OS: Debian Linux Sarge(v3.10)

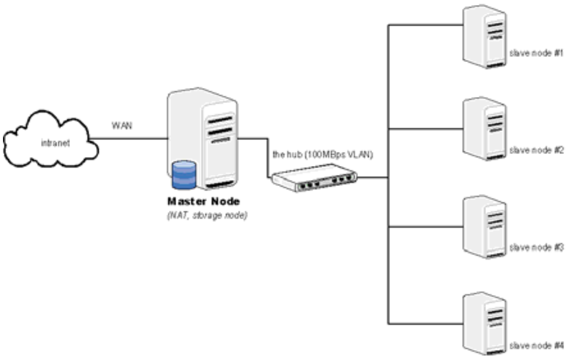

Figure 1. The experimental environment diagram.

Under such experimental environment, the proposed SOMNN algorithm by using MPI techniques based on the PC clusters can be described as follows:

```
int main(){
 MPI_Init();
        MPI_Comm_rank();
        MPI_Comm_size();

        //dispatch data to each node
        if (masterNode){
         seperateData(&inputFileContent, &buffer);
         for(nodeID = 0; nodeID < totalNode; nodeID++)
         MPI_Send(&buffer[nodeID], nodeID);
        }

        //receive data from master PC
        MPI_Recv(&subNodeData, masterNodeID);
        //Start SOM
        subResult = SOMLearn( &subNodeData );
        subGroupCenterPoint =
        SOMgCP( &subNodeData );

        //Return result of each node and the center point
data
        MPI_Send(&subResult, masterNodeID);
        MPI_Send(&subGroupCenterPoint, masterNodeID);
        for(nodeID = 0; nodeID < totalNode; nodeID++){
         MPI_Recv(&collectedData[nodeID], nodeID);
         MPI_Recv(&eachNodeGroupCenterPoint[nodeID],
         nodeID);
        }

        //Perform SOMNN by using the center point of
cluster
        eachGroupBelonging =
SOMLearn(eachNodeGroupCenterPoint);

        //deriving the result depending on the results from
the SOMNN procedure and the result of each node
        summary = makeResult(eachGroupBelonging,
collectedData);
}
```

4. Implantation and Result

The researching group took data from GONOME ALTAS DATABASE (it can be obtained from http:// www.cbs.Dtu.dk/services/GenomeAtlas/) to perform the analysis. At first, six bacteria with the full DNA sequence data will be chosen and the related information will be given as Table 1. A few and definite data will be used to verify the accuracy of the designed algorithm and architecture. The project team made a pre-processing for those DNA sequence to obtain the evaluation index, which is called as codon usage (Ghosh, 2000; Karlin & Mrazek, 1996). All bacteria will include three amino acids (LEU, SER, ARG) data with six codon value, restated, totally have eighteen value to be considered for each bacteria. Basically, those six bacteria can be grouped into three clusters (Hsieh *et al.*, 2007): (A, B) (C, D) and (E, F). That is, we meet a clustering problem with a structure of 6*18. Next, we will apply the proposed SOMNN-PC cluster algorithm to perform the application. The running procedure can be referred to Figure 2. Herein, Figure 2 is a log file to run SOMNN-PC cluster. From the result of Figure 2, we can find out that those bacteria can be clustered into three clusters. The first cluster includes A, B; the second cluster includes C and D; and the third cluster includes E and F. Such clustering result is the same as the actual result. Restate, the rationality, feasibility and accuracy of our proposed SOMNN-PC cluster algorithm can be verified.

Next, we use an easy way to make the performance comparison. The clustering analysis will be performed a single PC via another dataset. This dataset includes 1024 dimensions and the total number of data is 2048. The larger dimensions can make the comparison more meaningful. Then, we recorded the running time for a single PC and PC cluster and it will be listed in Table 2.

From Table 2, we can find out that the time for the proposed SOMNN-PC cluster algorithm is significant less than that on a single PC, i.e. the time for those two cases will be about 1:6. The performance on CPU operating time can be verified.

Table 1. The related data about the bacteria.

No.	Organism	Label	Accession No.	Bases(bps)	Taxo. ID
1	Escherichia coli CFT073	A	AE014075	5231428	199310
2	Escherichia coli K12	B	U00096	4639675	83333
3	Pyrococcus abyssi GE5	C	AL096836	1765118	272844
4	Pyrococcus furiosus DSM 3638	D	AE009950	1908256	186497
5	Bacillus cereus ATCC 10987	E	AE017194	5224283	222523
6	Bacillus cereus E33L	F	CP000001	5300915	288681

Table 2. The comparison result.

Method	Operations	time
Single PC	1024 dims *2048data	326.48sec
PC clusters		52.26sec

```
Starting on pca031 at Mon Nov 27 08:01:09 CST 2006

Nodes used for this job:
------------------------
pca031
pca031
pca030
pca030
------------------------
Working directory is /home/t2c09b00/som, 4
running /home/t2c09b00/som/som on 4 LINUX ch_p4 processors
Created /home/t2c09b00/som/PI25593
This system have 4 nodes.
and we generated 9-nodes SOM networks.
the data count of each node: 2
the data count of all: 8
    Sending data #0 from node 0 to 0(tag:0)...[ OK ]
    Sending data #1 from node 0 to 0(tag:1)...[ OK ]
    Sending data #2 from node 0 to 1(tag:10)...[ OK ]
    Sending data #3 from node 0 to 1(tag:11)...[ OK ]
    Sending data #4 from node 0 to 2(tag:20)...[ OK ]
    Sending data #5 from node 0 to 2(tag:21)...[ OK ]
    Sending data #6 from node 0 to 2(tag:30)...[ OK ]
    Sending data #7 from node 0 to 2(tag:31)...[ OK ]
    node #0 opened socket for receiving data...
    node #0 receiving data (tag:0)...[ OK ]
    node #0 receiving data (tag:1)...[ OK ]
    SOM sub-network established at node #0...[ OK ]
    **********************************
    *                 *
    * node #0      was initialized. *
    *                 *
   **********************************
```

```
    *                          *
    * node #0      was started learning... *
    *                          *
    **********************************
    * node #0       returning data...  *
    *                      *
    *************************************
    node #2 opened socket for receiving data...
    node #2 receiving data (tag:20)...[ OK ]
    node #2 receiving data (tag:21)...[ OK ]
    node #1 opened socket for receiving data...
    node #1 receiving data (tag:10)...[ OK ]
    node #1 receiving data (tag:11)...[ OK ]
    node #1 receiving data (tag:11)...[ OK ]
    SOM sub-network established at node #1...[ OK ]
    **********************************
    *                 *
    * node #1      was initialized. *
    *                 *
     ************************************
    *                          *
    * node #1      was started learning... *
    *                          *
    *************************************
    *                      *
    * node #1       returning data...  *
    *                      *
    *************************************
    SOM sub-network established at node #2...[ OK ]
    node #3 opened socket for receiving data...
    node #3 receiving data (tag:30)...[ OK ]
    node #3 receiving data (tag:31)...[ OK ]
    SOM sub-network established at node #3...[ OK ]
    **********************************
    *                 *
    * node #2      was initialized. *
    *                 *
    **********************************
    *                          *
    * node #2      was started learning... *
    *                          *
    ***********************************
    *                      *
    * node #2       returning data...  *
    *                      *
    ***********************************
    *                 *
    * node #3      was initialized. *
    *                 *
    **********************************
    *                          *
    * node #3      was started learning... *
    *                          *
    ***********************************
        *                          *
    * node #3       returning data...  *
    *                      *
    *************************************
Group #0's members = { 0 0 0 0 0 0 }
Group #1's members = { 1 2 0 0 0 0 }
Group #2's members = { 0 0 0 0 0 0 }
Group #3's members = { 0 0 0 0 0 0 }
Group #4's members = { 0 0 0 0 0 0 }
Group #5's members = { 0 0 0 0 0 0 }
Group #6's members = { 5 6 0 0 0 0 }
Group #7's members = { 0 0 0 0 0 0 }
Group #8's members = { 3 4 0 0 0 0 }
```

Figure 2. The log file for the result of SOMNN-PC cluster.

5. Concluding Remarks

In this study, we introduce a successful case for the issue of computational resource integration based on PC cluster with MPI techniques to achieve the clustering analysis. In this case, a rational and feasible algorithm of SOMNN-PC cluster is to propose at the laboratory environment of university. Except for the efficiency (i.e. processing time) of PC cluster architecture can be enhanced, the function of clustering analysis can also be made. This practice can be viewed as an application reference for the issue of PC cluster architecture in the future. Besides, how to apply the proposed PC cluster architecture into the case with large size, e.g. directly using the DNA sequence code to perform the clustering analysis, will also be a meaningful research in the future. And, the visual and user-friendly interface is also an improvement action to our future research.

Acknowledge

The authors would like to thanks the financial support from the National Science Council at Taiwan, R.O.C. with contract No. NSC 94-2416-H-143-001.

References

[1] Ghosh, T., Studies on codon usage in entamoeba histolytica, *International Journal of Parasitology*, 30, 2000, pp.715-722

[2] Hinton, G. E., Connectionist Learning Procedures", *Artificial Intelligence*, 40, 1989, pp. 185-234.

[3] Jeng, C. -C., Yang, I. -C., Hsieh, K. -L., & Lin, C. –N., Bacteria Classification on Power Spectrums of Complete DNA Sequences by Self-Organizing Map, *Neural Information Processing – Letter and Review*, 2005.

[4] Karlin, S. & Mrazek, J., What drives condon choices in human genes ? *Journal of Molecular Biology*, 262, 1996, pp.459-472

[5] Kohonen, T., Self-organized formation of topologically correct feature maps, *Biological Cybernetics*, 66, 1982, pp 59-69.

[6] Kohonen, T., *Self-organization and associate memory*, Springer-Verlag London, 1984.

[7] Vesanto, J., & Alhoniemi, E., Clustering of the Self-Organizing Map, *IEEE Transactions on Neural Networks*, 11, 3, 2000, pp.306~307

[8] Jang, J.-S. R., Sun, C.-T., & Mizutani, E., *Neuro-Fuzzy and Soft Computing, A Computational Approach to Learning and Machine Intelligence*, Prentice-Hall, 1997.

[9] Sterling, T. L., Salmon, J., Becker, D. J., and Savarese, D. F., *How to Build a Beowulf-A Guide to the Implementation and Application of PC Clusters*, Cambridge, Ma, The MIT Press, 1999.

[10]Alspector, J., & Lippe, D., A study of parallel weight perturbative Gradient Descent, In *Proc. of Advances in Neural Information Processing System (NIPS '96)*, Cambridge, Ma.: The MIT Press, 1996, pp. 803-810.

[11]Sloan, J. D., *High Performance Clusters with OSCAR, Rocks, OpenMosix, and MPI*, CA: O`Reilly, 2004.

[12]Mathematics and Computer Science Division, "MPICH-A Portable Implementation of MPI", Available from: *HTTP://www-unix.mcs.anl.gov/mpi/mpich/*, 2003.

[13]Bermmel, T., MP-MPICH: Multi-Platform MPICH", Available from: *HTTP://www.lfbs.rwth-aachen.de/mpmpich/*, 2003.

[14]Tsai, D.-M., The easy way to construct Cluster, http://linux.vbird.org, 2003.

[15]Hsieh, K. -L., Jeng, C. -C., Yang, I. -C., Chen, Y. -K. & Lin, C. -N., (2007), The Study of Applying a Systematic Procedure Based on SOFM Clustering Technique into Organism Clustering, *Expert Systems with Applications*, Vol. 33, No. 2, pp.330-336.