Solving Facility Location Problem on Cluster Computing

Ei Phyo Wai and Nay Min Tun

***Abstract*—**Computation of facility location problem for every location in the country is not easy simultaneously. Solving the problem is described by using cluster computing. A technique is to design parallel algorithm by using local search with single swap method in order to solve that problem on clusters. Parallel implementation is done by the use of portable parallel programming, Message Passing Interface (MPI), on Microsoft Windows Compute Cluster. In this paper, it presents the algorithm that used local search with single swap method and implementation of the system of a facility to be opened by using MPI on cluster. If large datasets are considered, the process of calculating a reasonable cost for a facility becomes time consuming. The result shows parallel computation of facility location problem on cluster speedups and scales well as problem size increases.

***Keywords*—**cluster, cost, demand, facility location

I. Introduction

THE large amount of money is spent for building a facility in any location in the country. Even though there are many factories in every location in the country, fixed cost for facility location is hard to cost the same amount depending on the size of the facility. Moreover, the fact that populations in each location, also called, customers demand the items, is also considered for running machine in that location. According to this, maximum product count, capital for running machine and customers' demand in that location is computed before making decision for facility location respectively. Consequently, it is needed as a manner to solve facility location problem. Computation of facility location problem for every location in the country is not easy simultaneously. If large datasets are considered, computing the problem becomes time consuming. A practical solution to overcome this problem is to parallelize the parallel algorithm for solving problems in a short period of time. The basic idea of parallelization is built can be efficiently exploited for computational purposes. However, this can only be effectively realized on SIMD (single instruction multiple data) computers.

Computing is also an essential tool in all branches of science and technology. Computing power can be enhanced by using a large number of computers together called clusters [1].Cluster computing is worked by dividing up into computationally independent segments. All machines share resources and must have software such as an MPI implementation installed to allow programs to be run across all nodes [2]. Message Passing Interface can provide a programming library for inter process communication in computer network. MPI is a specification for a library of routines that provide an infrastructure for parallel message passing applications [3].

Although, the majority of today clusters run on Linux operating systems, Microsoft Windows operating systems coupled with Dot Net is also becoming an interesting alternative. The Dot Net Framework is more flexible on any platform and it is more portable with the use of a dot syntax naming scheme other than writing many methods for algorithm.

In this paper, implementation of parallel computing to solve facility location problem is described. Parallel algorithm is implemented by linking Message Passing Interface to C# in Dot Net framework. Section (II) discusses facility location problem description and section (III) describes designing algorithm for solving problem (IV) cluster computing and cluster architecture. Section (V) describes important implementation details with emphasis on parallel computing for finding optimal cost and conclusion are provided in section (VI).

II. Facility Location Problem Description

The capacitated facility location problem involves selecting the minimum cost subset of potential facility sites given a fixed cost and capacity associated with each facility, variable production cost per unit for each facility, transportation costs per unit for goods shipped from the facilities to various customers and demand to be satisfied at each customer location [4].

The problem can be stated as follows.

$$\text{minimize} \quad \sum_i F_i y_i + \sum_i \sum_j c_{ij} d_j x_{ij} \qquad (1)$$

$$\text{s.t} \sum_i x_{ij} = 1 \qquad (2)$$

$$\sum_j d_j x_{ij} \le y_i q_i \quad \forall i \qquad (3)$$

where F_i = fixed cost of facility i

$y_i = \begin{cases} 1 & \text{if facility i is open} \\ 0 & \text{otherwise} \end{cases}$

d_j = demand at customer j

c_{ij} = variable cost of producing one unit at facility i and shipping it to customer j

Ei Phyo Wai is with Mandalay Technological University, Myanmar.
e-mail:maeiphyowai@gmail.com

$$x_{ij} = \begin{cases} 1 & \text{i supplies to j} \\ 0 & \text{otherwise} \end{cases}$$

q_i = capacity at facility i

In (1), F_i is an array of fixed cost for factory i. y_i is a binary array with 0 and 1. If a facility is opened at location i, it will be set to 1 and else set to 0. c_{ij} is a matrix that incorporates with item's product cost and shipping cost from location i to j. x_{ij} is also a matrix and the value is changed depending on customers' demand at facility i that indicates if location j is supplied to a facility at i. The more the locations far away among these, the larger shipping cost distributing items form location i to j is considered. Larger cost for facility location and item distribution can cause not to make the reasonable decision to run machine at that location i and to distribute items to another location j. Otherwise, it is possible to run the machine by finding minimal cost for facility location.

And then, we consider customers' demand and capacity of items at facility consequently. In (3), d_j is an array for demand at customers' location j and q_i is an array for capacity of facility's location i. The gain of running machine at location i should be well when demand at location j is more than quantity of items at location i.

The objective function minimizes the total system cost while the constraints insured that (a) demand is satisfied, (b) capacity is not exceeded, and (c) integrality of the binary variables.

III. The Algorithm For Solving Problem

The algorithm is designed to solve facility location algorithm by finding minimal cost. First, an array Z_j is defined in order to update this array. Depending on the array Z_j , the matrix x_{ij} is made according by row using local search with single swap method. We define Z_j as follows.

$$Z_j = \{z_1, z_2, z_3, ...z_j\}, \; Z_j \in I, I \in N, \; J \in N$$

where Z_j denotes fixed number and to be updated by using local search with single swap method. I is maximum count at facility location i and J is maximum count at customer location j. N is all possible locations. Updating Z_j can make the matrix x_{ij} and provide modification to problem. The matrix x_{ij} is stated as follows.

$$x_{ij} = \begin{bmatrix} x_{11} & x_{12} & .... & x_{1j} \\ x_{21} & x_{22} & .... & x_{2j} \\ ...... & ..... & ..... & ...... \\ x_{i1} & x_{i2} & .... & x_{ij} \end{bmatrix}.$$

$$\text{where} \sum_i x_{ij} = 1 \quad \text{and} \quad \sum_j d_j x_{ij} \le y_i q_i$$

In the problem, F_i and c_{ij} is fixed cost or given.   is a binary array with 0 and 1. If a facility is opened at location i, it will be set to 1 and else set to 0. Demand at customer location j, d_j is considered with location j is supplied by location i, x_{ij}. Modification to the problem, it will be solved by designing minimum cost algorithm. Decision making for a facility to be open if is satisfied all of customer's demand and distribution process facility i to customer j.

Producers of Minimum Cost using Local Search with Single Swap are described as follows.

Algorithm 1: **Procedure** Minimum Cost

```
1. Initialize $Z = \{z_1, z_2, z_3, .....z_j\}$
   $Z^{min} \leftarrow Z$
   Calculate min cost
2. (a) $Z \leftarrow Z^{min}$
   (b) $Z^{temp} \leftarrow Z$
       Random (i, j)
       Update $Z^{temp}$
       $Z^{min} \leftarrow \min\{Z^{temp}, Z^{min}\}$
       Go to 2(b) until count $\le$ N1
3. Return $Z^{min}$
   Go to 2(a) until count$\le$ N2
```

IV. Cluster Computing

There are two common types in parallel computing: supercomputing and cluster computing.

Supercomputing have been in the mainstream of high-performance computing for the past decade. The reasons for this decline are many, but include factors like being expensive to purchase and run, potentially difficult to program, slow to evolve in the face of emerging hardware technologies, and difficult to upgrade without, generally, replacing the whole system.

Cluster computer consists of distributed memory computers that are built from scratch using mass produced PCs and workstations. Although clusters are cheaper and smaller, they are limited by network latency. [5].

A. Cluster Architecture

Microsoft windows compute cluster consists of head node and compute nodes. Head node provides deployment and administration user interfaces (UIs) as well as management services for the compute cluster. The UIs include the compute cluster administrator, the compute cluster job manager, and a command line interface (CLI). Management services include job scheduling as well as job and resource management. Compute node is a computer configured to provide computational resources as part of the compute cluster. Compute nodes allow users to run computational jobs. MPI software that is a key networking component of the compute cluster is used in this work [6].Each of the five network scenarios supported by Compute Cluster System (CCS) offers varying degrees of performance, as illustrated in the following:

1) Two NICs on the head node and one on each of the compute nodes
2) Two NICs on each node
3) Three NICs on the head node and two on each of the cluster nodes
4) Three NICs on each node
5) A single NIC on each node with all network traffic sharing the public network [7].

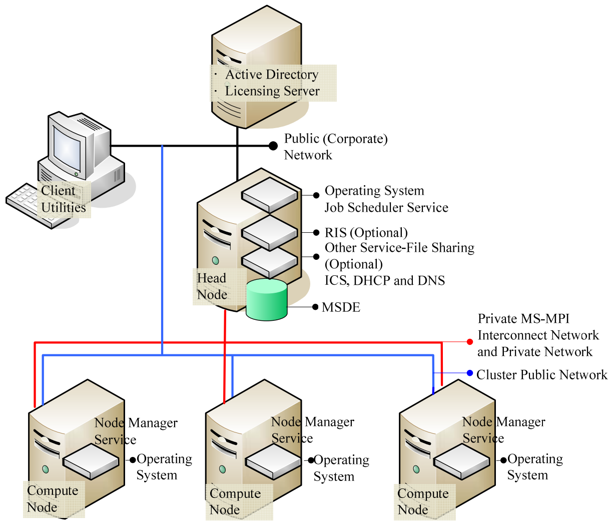

Fig. 1 Typical CCS Network Topology

In our Compute Cluster system, second scenario that is two NICs utilized on each node among all type has to be used, as shown in Fig. 1. This private network is used for cluster management, optional Remote Installation Service (RIS) deployment of node images, and MS MPI traffic. Because each compute node is also directly connected to the public network, debugging applications running on the nodes is easier. It can be connected directly to the nodes when issues arise, making debugging more efficient. Although this scenario is not ideal to run tightly parallel application, it is a convenient topology for the reason that there is no need the more expenditure of money on NICs. Additionally it will significantly improve cluster performance by using separate NIC on each network and there is no traffic. With the added benefit that the head node does not act as a bottleneck, forcing all computational results to pass through it when being reported to operational programs.

V. System Design and Implementation

In this section, we solve facility location problem by finding minimum cost on cluster computing system. . Steps involved in solving facility location problem in parallelism are as follow:

1) Read input data for which finding facility location.
2) Tasks are distributed by the head node to find minimum cost.
3) If the number of nodes in the cluster under consideration is n, divide tasks such that each group has a number of task equal to s = p/n, where p is problem size.
4) For each node, it receives distributed tasks from the head node and compute minimum cost for facility location.

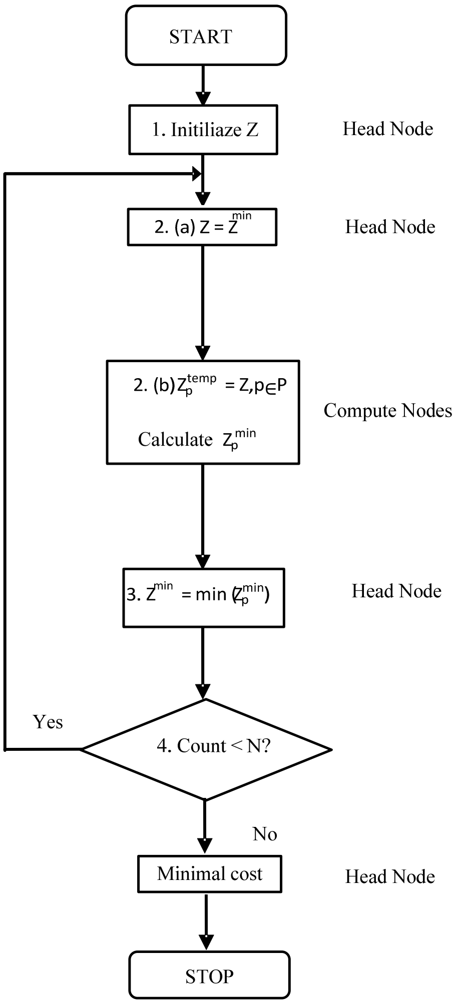

Fig. 2 Flow Chart for a System of a Facility to be Opened

After computing each task at each compute node and then send back the results to the head node.

To begin communication among nodes, initialized MPI and finalized MPI method are used.

MPI.Environment.Initilized < >

MPI.Environment.Finalized < >

For sending data to all nodes, broadcast method is used.

MPI.Intercommunicator.world.Broadcast< >

Result of each node is sent back to the root node with Gather method.

MPI.Intercommunicator.world.Gather < >

In Fig. 2, the task of finding minimum cost is parallelized and the process is repeated in order to find the reasonable cost for facility location. The more the processes are, the more closet cost for location the facility is opened. When more compute nodes are applied on cluster computing and more problem size are computed at each node, the process of finding minimum cost speedups and the reasonable cost is getting within a short period of time.

A. Experimental Results

There are eight compute nodes and one head node. Each node has its own 3.2 GHz Pentium IV processor and 256 MB memory. Windows Compute Cluster Operating System is installed in each node. By measuring the program execution time, the results are as follows.

If problem size is taken as 64 kilobyte, the calculation results are shown in Fig. 3. By seeing the graph, we conclude that if we partition the small amount of data to many nodes, a lot of overhead will be incurred. Therefore, execution time calculated by one node is faster than the time executed by eight nodes.

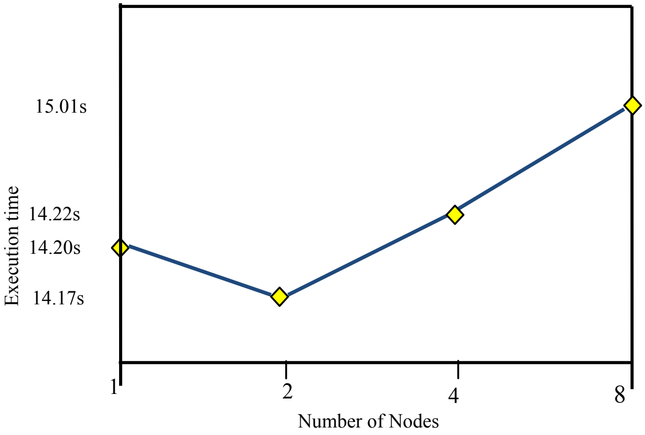

Fig. 3 Calculation Results on 64K Problem Size

TABLE I
EXECUTION TIME ON 64 K PROBLE SIZE

Problem size	*Number of nodes*	*Execution time*
64 K	8 Nodes	16.72s
	4 Nodes	13.59s
	2 Nodes	14.17s
	1 Node	14.20s

TABLE II
EXECUTION TIME ON 1M PROBLEM SIZE

Problem size	*Number of nodes*	*Execution time*
1M	8 Nodes	1min 43.53s
	4 Nodes	1 min 46.81s
	2 Nodes	1 min 48.20s
	1 Node	1 min 50.88s

Otherwise, we take 1M problem size as input data; it will be effective use of cluster computer. In this case, the time executed by one node is slower than the time executed by eight nodes as shown in Fig. 4.

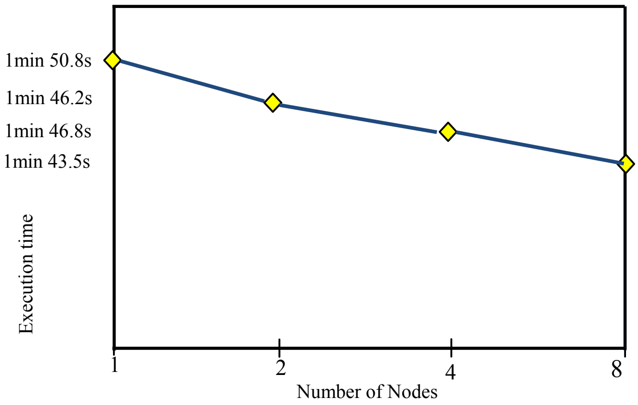

Fig. 4 Calculation Results on 1M Problem Size

VI. CONCLUSION

Facility location is an important issue in the country. Consideration of facility location is a problem because of profits. In this paper, facility location problem is solved on cluster computing system. And then the execution time for 16K problem size and execution time for 1M problem size are described. Due to cluster computing, solving problem speeds up when large problem size is taken as input data set. The implementation of solving facility location problem is done by linking C# and MPI in Dot Net Framework.

ACKNOWLEDGMENT

We would like to thank to Dr Nay Min Tun for his valuable comments and suggestions and the referees for their help in improving the clarity of the presentation.

REFERENCES

[1] Anath Granma, Anshul Gupta, George Karypis and vipin Kumar, “Introduction to Parallel Computing”, 2nd ed.., 2003.

[2] Emir Imamagi, Damir Danijel Žagar, “Cluster Distributions Review,” in Department of Computer Systems,University Computing Centre, Croatia, p. 1.

[3] Fraigniaud, Anne Mignotte, and Yves Robert, editors, Euro-Par '96 Parallel Processing, volume 1 of Lecture Notes in Computer Science, in Luc Bouge, Springer Verlag, p 128–130., 1996.

[4] Candace Arai Yano, “On the Equivalence of an Equipment Replcacement Problem and a Facility Location Problem,” in department of Industrial and Operations Engineering, December, 1984.

[5] Hai Jin, Rajkumar Buyya, Mark Baker, “Cluster Computing Tools, Applications, and Australian Initiatives for Low Cost Supercomputing,” in School of Computer Science, University of Portsmouth, Portsmouth, *Hants*, UK, 2001, p. 2.

[6] E. Lusk. Programming with MPI on clusters. In 3rd IEEE International Conference on Cluster Computing (CLUSTER’01), October 2001.

[7] N. Ruest and D. Ruest, Deploying and Managing Microsoft Windows Compute Cluster Server 2003, p.5-10, November 2005.