Building GraphBased Symmetric Cluster

Felix P. Muga II
Mathematics Department
Ateneo de Manila University
Quezon City, Philippines
fpmuga@admu.edu.ph
Rafael P. Salda˜na
High Performance Computing and Networking Laboratory
Ateneo de Manila University
Quezon City, Philippines
raf@admu.edu.ph
William Emmanuel S. Yu
High Performance Computing and Networking Laboratory
Ateneo de Manila University
Quezon City, Philippines
wyy@admu.edu.ph

ABSTRACT -- In August 2000 the High Performance Computing Research Group (AHPC) of the Ateneo de Manila University built an 8-node Beowulf- class computer designed for computational science applications. As more researchers and students in the University are trained in cluster computing, the need for building a better cluster arises. This year the AHPC proposes to build a large-scale graph-based symmetric cluster. The proposed high performance computing system will be a symmetric cluster with a single-switch latency and f lat networking neighborhood topology. The proposed design also features minimized cost and maximized bandwidth. This presentation will deal with mathematical and computational aspects of graph-based clusters, and design considerations for a large-scale symmetric cluster with a single-switch latency.

KEYWORDS -- symmetric, balanced and flat network neighborhoods, isomorphic graphs, bisection bandwidth, pairwise node bandwidth, regular graphs, switch latency, cluster computing, beowulf, parallel computing

1. INTRODUCTION

Cluster computing is becoming an accepted form of supercomputing. In universities, government institutions and commercial companies, there is a growth in the cluster installation base. In the international scene, there is a race to build the biggest and the fastest clusters.

The popularity of cluster computing is growing among scientific computing and research communities. It is also expanding in the commercial sector, and a large number of very large scale clusters are being deployed. However, according to Amdahl's Law, the speedup of a system is limited by the speedup of a single component in such a system. This is true in the case of cluster computing. It is not simply a case of adding compute nodes to the cluster to make it perform better. Other factors, such as interconnection network, will cause a performance bottleneck.

To improve the performance of a supercomputing cluster, it is important to eliminate bottlenecks. Limitations in network switch sizes, latencies and other network devices do not make this task easier. The use of alternative neighborhood networks can help answer these network limitations.

In August 2000 the High Performance Computing Research Group (AHPC) of the Ateneo de Manila University built an 8-node Beowulf- class computer designed for computational science applications[10,11]. As more researchers and students in the University are trained in cluster computing, the need for building a better cluster arises. This year the AHPC proposes to build a large-scale graph-based symmetric cluster. The proposed high performance computing system will be a symmetric cluster with a single-switch latency and flat networking neighborhood topology. The proposed design also features minimized cost and maximized bandwidth. This presentation will deal with mathematical and computational aspects of graph-based clusters, and design considerations for a large-scale symmetric cluster with a single-switch latency.

2. THE NETWORK PROBLEM

Parallel computation is typically composed of tasks that are parallel and some tasks that are not. Parallel tasks are those tasks that can be accomplished simultaneously with or without active communication. Serial tasks are those tasks that have to be completed one after another in a proper sequence. Serial tasks are usually irreducible and are treated as fixed computational overhead and at times it can expand. Parallel tasks have ideal completion times like $1/N$ where N is the number of parallel tasks undertaken at the same time[5]. Parallel task, however, require a communications overhead between tasks. All of these are made formal in Equation 2 referred to as Amdahl's Law and quantitatively corrected in books in parallel computation[5].

$$\frac{R(P)}{R(1)} = \frac{(T_s + T_p)}{(T_s + (\frac{T_p}{P}))} \tag{1}$$

$$\frac{R(P)}{R(1)} = \frac{(T_s + T_p)}{(T_s + (P * T_{is}) + (\frac{T_p}{P}) + T_{ip})} \tag{2}$$

This law strictly limits the amount of speedup that can be attained from a paralleled program. However, this equation does not consider some other factors such as T_{is} or the average serial time which includes time delays due to Inter-process communications, setup, initialization and other. Another factor is T_{ip} which is the average parallel time spend by each processor performing tasks like initialization, setup and even idle time. With these in mind the more realistic form of Amdahl's Law is shown in Equation 2.2.

It can be seen that the speedup of the entire system is severely limited by this law. The communications overhead while is part of the computation can severely limit the speedup gained. Thus, a large number of cluster nodes cannot be justified if the network will simply reduce its benefits.

3. AGILA NETWORK DESIGN – A GRAPH BASED CLUSTER

We are proposing to enlarge the Athlon Beowulf cluster of the Ateneo High Performance Computing Group known as AGILA from the present set-up of 15 nodes to 256 nodes. Using a 4-way motherboard, our new cluster system will have a total of 1024 processors.

The topology of the cluster system we are proposing is based on 16 copies of a circulant graph of order 16 with jump sizes ± 1, ± 4, ± 7, 8. The vertices of the graph are labelled 0, 1, 2,…, 15 such that vertex u is adjacent to the vertex $u \pm 1$, $u \pm 4$, $u \pm 7$, $u + 8$ (the sum is taken under modulo t).

The next sections discuss the theoretical basis of our proposed topology.

3.1 A GraphBased -Cluster

Let G be a graph with vertices 0, 1, 2, . . . , t - 1 and let G_0 , G_1 , . . . , G_{m-1} be m copies of G. Then the vertices of G_i are labelled as i, $m + i$, $2m + i$, . . . , $tm - m + i$ and the order of G_i is t. (Note that *order* of a graph means the number of vertices of a graph.)

Consider a cluster whose compute nodes are the vertices of the m copies of G. Let us partition the mt compute nodes of the cluster into t subnets such that each subnet C_k consists of the nodes km, $km + 1$, $km + 2$, . . . , $km + m - 1$. We connect the nodes of the cluster to t network switches S_0 , S_1 , . . . , S_{k-1} using the *node-to-switch connection procedure* given below.

Procedure NTS-1

Let u be a compute node belonging to subnet C_k . Then

NTS-1. connect node u to switch S_k .

NTS-2. connect node u to S_{k_1} , S_{k_2} , . . . , $S_{k_{r-1}}$, if k is adjacent to

vertice k_1 , k_2 , . . . , k_{r-1} in G.

Let us denote the cluster based on m copies of a graph G of order t and connected using the node-to-switch connection procedure by C_G (m, R, t) where R is the set $\{r_0 + 1, r_1 + 1, \ldots, r_{t-1} + 1\}$ and each r_k denotes the degree of vertex k in G. If $r_0 = r_1 = \ldots = r_{t-1}$ G is called a regular graph. An example of a regular graph is the circulant graph $G(t; \pm s_0 , \pm s_1 , \ldots , \pm s_r)$ such that the vertices are labelled as 0, 1, . . . , t - 1 and each vertex v is adjacent to vertices $v \pm s_0$, $v \pm s_1$, . . . , $v \pm s_d$ where addition is taken under modulo t. If t is even, then $\frac{t}{2} \equiv -\frac{t}{2} \pmod{t}$. Hence, if $s_d = \frac{t}{2}$ when t is even, the regularity of the circulant graph is odd. Consider the following example.

NTS-1 accounts for 1 network switch for each node, while NTS-2 accounts for r - 1 switches for each node. Hence, each node in the cluster is attached to r network switches. Therefore, there are r NICs installed in each node.

Theorem 1 *Let G be a graph of order t and regularity r - 1. Suppose that the cluster C_G (m, R, t) uses s-port switches. Then $m \leq \left\lfloor \frac{s}{r} \right\rfloor$, i.e ., each subnet can have at most $\left\lfloor \frac{s}{r} \right\rfloor$ compute nodes.*

Proof. The total available number of ports is st. If m copies of the regular graph G are to be used, then the cluster uses mrt ports.

Hence, $mrt \le st \Leftrightarrow mr \le s \Leftrightarrow m \le \frac{s}{r}$.

Since m is a positive integer, $m \le \left\lfloor \frac{s}{r} \right\rfloor$.

Example 1

In Fig. 1, the cluster is based on 6 copies of $G(8; \pm 1, 4)$. This is a 3-regular circulant graph and has 8 vertices. Since NTS-1 connects a node to one switch and NTS-2 connects the node to three other switches, each node in the cluster needs 4 1-port NICs or 1 4-port NICs. If the cluster uses 24-port switches, then it must have $\left\lfloor \frac{24}{4} \right\rfloor = 6$ copies of the base graph. Hence, the cluster has $8 \times 6 = 48$ compute nodes partitioned into 8 components $C_0, C_1, \ldots, C_7$ where each component C_k has the six nodes km, $km + 1$, $km + (m - 1)$. The total number of installed NICs is $4 \times 48 = 192$. The cluster needs 8 network switches labelled as $S_0, S_1, \ldots, S_7$.

A node symmetric cluster or network has no distinguishable node. The "view" of the rest of the network cluster is the same from any node. Rings, fully connected networks, and hypercubes are all node symmetric network. This property is similar to that of the *vertex-transitive graph*. Hence, a cluster C is *node-symmetric* (or *vertex-transitive*) if there exists an *automorphism* ϕ from the cluster's node set $V(C)$ onto itself. If a cluster is node-symmetric we simply call it as a symmetric cluster.

Theorem 2 *If G is a vertex-transitive graph, then the cluster $C_G(m, R, t)$ is symmetric.*

Proof. Let G be a vertex-transitive graph of order t. Define a mapping φ from the vertex set of $C_G(m, R, t)$) onto itself by $\varphi(km + i) = \phi(k)m + i$ where ϕ is an automorphism of of the vertex set of G onto itself. Clearly, φ is an automorphism from the vertex set of $C_G(m, R, t)$ onto itself. Therefore, the cluster $C_G(m, R, t)$ generated by the vertex-transitive graph G is symmetric.

Theorem 3 *If G has diameter 1 or 2, then every pair of nodes in $C_G(m, R, t)$ has a common switch. Therefore, $C_G(m, R, t)$ has the FNN topology.*

Proof. Let $u = k_1m + i_1$ and $v = k_2m + i_2$ be two distinct compute nodes in the $C_G(m, R, t)$. If $k_1 = k_2$, then the two nodes belong to C_k. Hence, they are joined by switch S_k. Suppose that $k_1 \ne k_2$. If k_1 and k_2 are adjacent in G, then u and v are joined by two switches. If k_1 and k_2 are not adjacent (only when diameter is not 1), then they have a common neighbor since G has diameter 2. Hence, u and v are joined by a switch. Therefore, $C_G(m, R, t)$ has the FNN topology.

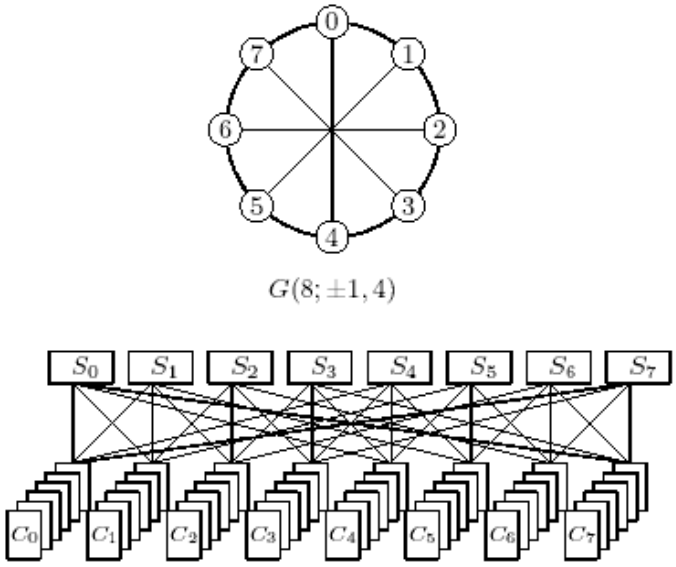


Figure 1. A cluster based on 6 copies of $G(8; \pm 1, 4)$ cluster $C_G(m, R, t)$

4. A PROPOSED GIGANTIC CLUSTER DESIGN

In this section, we shall show a design of a symmetric FNN cluster with about 1000 processors. The base graph is a circulant graph with 16 vertices and jump sizes $\pm 1, \pm 4, \pm 7, 8$. See the graph in Fig. 2.

Since the base graph has $t = 16$ vertices, the cluster needs 16 switches. Also, the base graph is 7 regular. It follows that each compute node needs two 4-port NICs or $r = 8$. On a 128-port 100 Mbps Fast-Ethernet switches, the number of compute nodes per subnet is at most $\left\lfloor \frac{128}{8} \right\rfloor = 16$. Hence, $m = 16$. Each subnet corresponds to a network switch. Thus, the $C_G(16, 8, 16)$ cluster has $16 \times 16 = 256$ compute nodes. Using 4-way motherboards, the $C_G(16, 8, 16)$ cluster can have at most $256 \times 4 = 1{,}024$ processors.

4.1 Bisection Bandwidth of the GraphBased Cluster

A communication link between two nodes in a cluster using the *FNN* topology is the connection from one node to a switch and the connection from the switch to the other node. The number of communication links between two distinct nodes is defined as the *pairwise bandwidth* of the given pair.

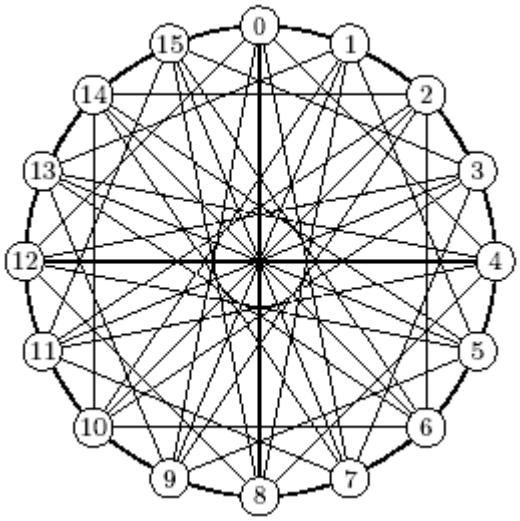


Figure 2. Circulant G (16; ± 1, ± 4, ± 7, 8)

The minimum number of communication links that must be removed to partition a network into two equal halves is defined as the *bisection bandwidth* of *G* is denoted by *BBW* (*G*). This type of partition is known as the *bisection* of the *G*.

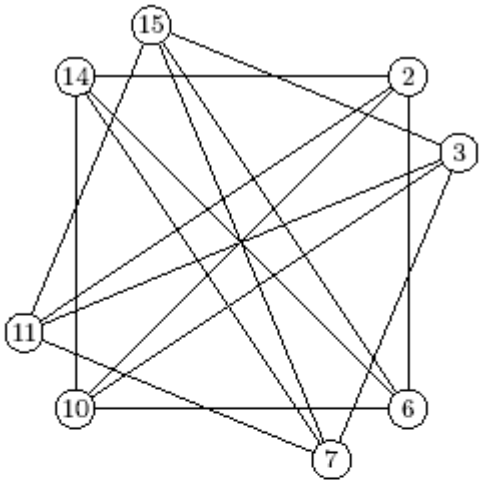
The bisection bandwidth of a *tree* is one, since any partition cuts across only one communication links. The bisection bandwidth of a *hypercube* of order 2^d (number of nodes) is 2^{d-1}, since at least 2^{d-1} communication links must cross any partition of a hypercube into two subcubes. The bisection bandwidth of a complete graph of order p is $p^2 / 4$, if p is even, or $(p^2 - 1) / 4$, if p is odd.

Switches are the *de-facto* standard component for a cluster system. Switches are used to connect different nodes in such a way that each node is given a guaranteed bandwidth. A switch S_k with the nodes in component C_k determines a subnet of nodes of the graph-based cluster. Switch S_k is associated with vertex k in G. If two vertices k_1 and k_2 are adjacent in G, then the subnets determined by S_{k_1} and S_{k_2} are also "adjacent", because they share the same nodes in C_{k_1} and C_{k_2}. Hence, If the edge that is incident to k_1 and k_2 is removed, then the communication links between the nodes in C_{k_1} and S_{k_2} and between the nodes in C_{k_2} and S_{k_1} are removed. There are $2m$ such links. Hence, we have the following theorem.

Theorem 4 *The bisection bandwidth of* C_G (*m, r, t*) *is* $2m \times BBW(G)$.

The bisection width of Circulant *G* (16; ± 1, ± 4, ± 7, 8) requires at least 16 edges to partition the graph into two equal halves. See Fig. 3.

Hence the (bidirectional) bisection bandwidth of C_G (16, 8, 16) is $2 \times 16 \times 16 \times 200$ Mbps = 102.4 Gbps edges.



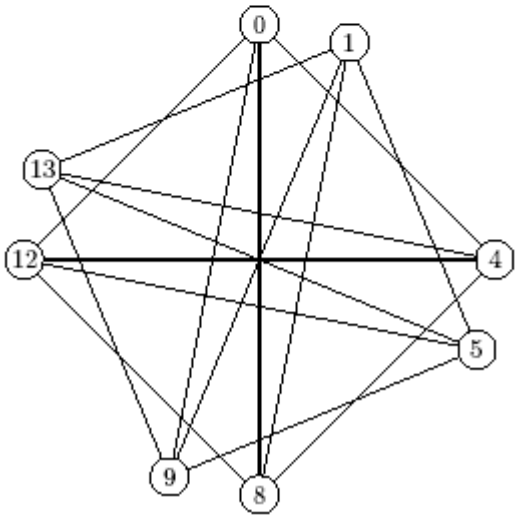


Figure 3. A Bisection of G (16; ± 1, ± 4, ± 7,8)

Theorem 5 *Suppose that* t *is the total number of switches used,* s *be the total number ports used per switch and the cluster has* n *compute nodes. Then the average pairwise links between two compute nodes is*

$$\frac{\binom{s}{2} \times t}{\binom{n}{2}}.$$

Proof. Since the cluster has *t* available switches with *s* available ports per switch, it follows that the total possible links of the cluster is $\binom{s}{2} \times t$. The cluster has n compute nodes. Hence,

the total number of pairs of compute nodes is $\binom{n}{2}$. Therefore, the average number of links between two nodes is

$$\frac{\binom{s}{2} \times t}{\binom{n}{2}}.$$

The average number of links of the C_G (16, 8, 16) cluster is

$$\frac{\binom{128}{2} \times 16}{\binom{256}{2}} = 3.98431373$$ or an average bidirectional pairwise bandwidth of 796.862746 Mbps.

5. CONCLUSION

In this paper we have discussed the theoretical basis of our proposed topology for building graph-based symmetric clusters with single switch latency.

6. ACKNOWLEDGMENT

This project received partial funding from the Philippine Commission on Higher Education Center of Excellence (CHED-COE) Grant of the Ateneo de Manila University.

References

[1] Ahmad, I., “Gigantic Clusters: Where Are They Going?” IEEE Concurrency, 83-85 (April-June 2000).

[2] Becker, D., Sterling, T., Savarese, D., Fryxell, B., and Olson, K., "Communication Overhead for Space Science Applications on the Beowulf Parallel Workstation." High Performance and Distributed Computing, 1995.

[3] Cand R., Dumas, E., Menel, F., Linux Kernel Book. (England, John-Wiley & Sons Ltd), (1997).

[4] Dietz, H. G. and Mattox T. I., "Compiler Techniques For Flat Neighborhood Networks," 13th International Workshop on Languages and Compilers for Parallel Computing(LCPC00). New York, August 11, 2000.

[5] Foster, I., Online Book: Designing Parallel Programs. Viewable online at http://www-unix.mcs.anl.gov/dbpp/.

[6] Katz, D., "Beowulf Applications and User Experiences." Seventh International Symposium on High Performance Distributed Computing. (1998).

[7] Muga, F.P. II, “Maximal Order of 3- and 5-Regular Circulant Graphs”, Matimyas Matematika, (1999).

[8] Muga, F.P. II, “Designing Clusters Using Scalable, Symmetric, Balanced and Flat Neighborhood Networks”, submitted for presentation.

[9] Reschke, C., Sterling, T., Ridge,D., Savarse, D., Beer, D., Merkey P., "A Design Study of Alternative Network Topologies for the Beowulf Parallel Workstation." Proceedings of the 5th IEEE International Symposium on High Performance Distributed Computing 1996.

[10] Salda˜na, R.,J. Garcia, F. Muga II, W. Yu, "Development of a Beowulf-Class High Performance Computing System for Computational Science Applications." Proceedings of the 18th National Physics Congress, Palawan, Philippines2000.

[11] Salda˜na, R.,"Project AGILA: the Ateneo High Performance Computing System." Proceedings of the First Philippine Computing Science Congress, Manila, Philippines 2000.