Entropy, Fault Tolerance, and Multicomputer Networks

Patricia L. Patterson

E-Systems Medical Electronics, Inc.
P. O. Box 660023, Dept. 50300
Dallas, TX 75266-0023

Abstract

An important characteristic of a MIMD parallel processing multicomputer network relating to its topology and algorithmic applications is its fault tolerance or ability to continue effective computation as individual computer nodes fail. This paper defines the new concept of graph entropy as a performance measure relating to that fault tolerance and as an analog to the entropies of information theory and statistical mechanics.

1: Introduction and Definitions

A MIMD (multiple instruction stream, multiple data stream) parallel processing multicomputer network [1] is comprised of a group of computer nodes linked together into a static topology in which each node interacts by message passing only with the nodes to which it is directly linked. No central memory or controller exists. Instead each node has its own local memory, and no memory is shared. And computer operations and communications are asynchronous. For the execution of global algorithms in a *symmetric* MIMD network (where each node is isomorphic to any other node), nodes are often programmed to execute the same local algorithms.

A MIMD multicomputer network may be likened in a loose sense to an assembly of molecules, frozen into a crystalline structure in which each molecule interacts with its near neighbors according to physical laws predominating at the molecular level. A single computer is likened to a molecule; a multicomputer topology is likened to a crystal structure; local computer algorithms are likened to near neighbor physical laws. And as an outsider to a molecular system can observe its macroscopic properties without knowledge of the action of the system at the microscopic level, so can a user of a MIMD system obtain a solution to a global algorithm without knowldege of the action of the system at the local computer level.

It is desirable in a MIMD network, that as computer nodes fail during the execution of a global algorithm, that the execution be minimally affected. The concept of graph entropy to be defined herein relates to the magnitude of that effect. Graph entropy for a multicomputer network will be defined analogously to the concepts from information theory [2] and statistical mechanics [3] in which entropy is a property of a specified experiment, discrete random variable or (molecular) system.

Let x be a random variable with sample space X = $\{x_1, \ldots, x_i, \ldots, x_m\}$ and probability measure $p(x_i) = p_i$. The information theoretic entropy of x in bits is:

$$H(x) = - \Sigma\, p_i \log_2 p_i,$$

where the summation is taken over all i from 1 to m. The Boltzmann-Planck relation of statistical mechanics defines entropy as:

$$S = k \log \Omega,$$

where S is the entropy of an assembly of molecules, k is Boltzmann's constant, and Ω is the *molecular disorder* of the assembly (the number of *microstates* available to the assembly, consistent with any *macroscopic constraints).* (A microstate is a specific way of realizing a given distribution; macroscopic constraints, such as total energy, are observable at the macroscopic level.) Since the probability of occurrence of each microstate is $p_i = 1/\Omega$, this definition is analogous to the information theoretic definition with k = 1.

To analogously define graph entropy, the experiment is to map or embed a graph or its isomorphism onto or into a specified multicomputer topology. Macroscopic (global) constraints will correspond to graph characteristics or relations to the network which are specified by the individual experiment. Local

0-8186-3420-0/93 $03.00 © 1993 IEEE

constraints will be constant for all mapping experiments (as are physical laws for the molecular system) and are as follows: (1) a graph vertex is mapped onto only one network node; (2) not more than one graph vertex is mapped onto the same network node; and (3) adjacent vertices in the graph are mapped into adjacent nodes in the network. Graph entropy therefore is a property of a well defined mapping experiment involving a graph, a specific nodal topology, and specific global (macroscopic) constraints.

The outcome of the experiment is a specific mapping or microstate, with associated random variable, x_i. The likelihood, p_i, of the occurrence of a specific microstate is a function of the mapping algorithm, but it will be assumed here that local algorithms are definable such that each microstate is equally likely ($p(x_i) = 1/x_m$). As an analog to molecular disorder, call the number of possible microstates (mappings) satisfying the macroscopic (global) constraints the *graph disorder*, Ω_g, of the experiment where for counting mappings, identical (i.e., symmetrically similar) nodes are considered distinguishable in the multicomputer network. The *graph entropy* of the experiment is then:

$$H_g = - c \sum 1/\Omega_g \log 1/\Omega_g = c \log \Omega_g,$$

where the summation is taken over all Ω_g mappings.

In general, graph entropy for a MIMD network relates to: (1) the efficiency of the network in passing messages between nodes; (2) the ease and flexibility of mapping global algorithmic graphs onto the network; and (3) the ability of the mapped graphs to reconnect and remap after nodal failure within the network (fault tolerance). Since the magnitude of graph entropy can be used to compare systems according to the above properties but cannot at this time be related to an absolute quantitative performance measure, the constant c in the graph entropy definition will be treated as equal to 1.

2: Examples of Graph Entropy

The graph of a mapping experiment may be either generic (such as *any* tree graph) or fully defined (such as a *specific* tree graph and its isomorphisms). The subscript s used with graph disorder, Ω_s, and graph entropy, H_s, will be lowercase if the graph is specific and uppercase if the graph is generic. Some examples of graph entropy follow for tree graphs, line graphs, and network graphs.

2.1: Tree-Graph Entropy

Consider an n-node network of specified topology and the following macroscopic constraints: (1) the graph forms a spanning tree, i.e., the tree includes all network nodes as vertices; (2) the root node for the tree corresponds to a specified node in the network; and (3) the tree has minimum possible energy relative to the specified tree root, where energy is defined as $E = \sum d_{i,r}$, where the summation is over all nodes i, and where $d_{i,r}$ is the minimal number of links in a network path between node i and the root node.

Simple illustrations of graph entropy for tree graphs are shown in Figure 1 for 9-node square mesh and 2-torus multicomputer networks and in Figure 2 for an 8-node Boolean 3-cube multicomputer network.

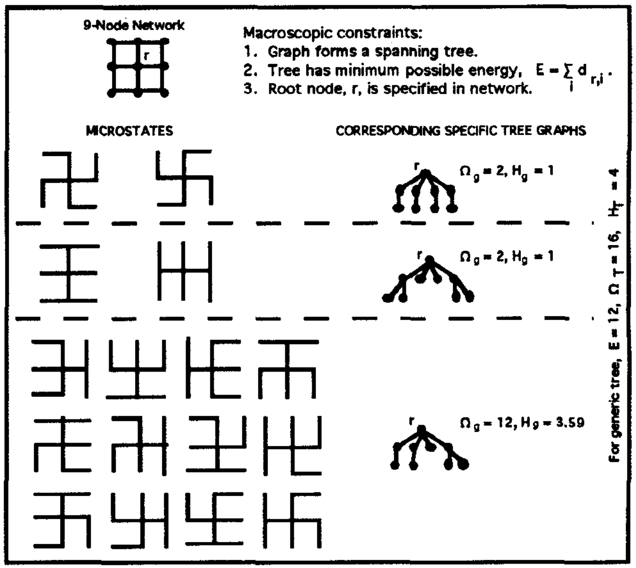
9-Node Network
r
Macroscopic constraints:
1. Graph forms a spanning tree.
2. Tree has minimum possible energy, $E = \sum_i d_{r,i}$.
3. Root node, r, is specified in network.
MICROSTATES
CORRESPONDING SPECIFIC TREE GRAPHS
r
$\Omega_g = 2, H_g = 1$
r
$\Omega_g = 2, H_g = 1$
r
$\Omega_g = 12, H_g = 3.59$
For generic tree, $E = 12$, $\Omega_T = 16$, $H_T = 4$

Figure 1: Tree-Graph Entropy for 9-Node Square Mesh or 2-Torus.

In general, for an unfaulted n-node 2-dimensional square mesh (2-mesh) network (with root at center node) or its corresponding 2-torus,

$$\Omega_T = 2^{(n-2n^{1/2}+1)} \quad \text{(for } n^{1/2} \text{ odd)};$$

and for an unfaulted Boolean N-cube,

$$\Omega_T = \prod d^{(N,d)},$$

where d is nodal shortest path distance from the root node, (N,d) is the binomial

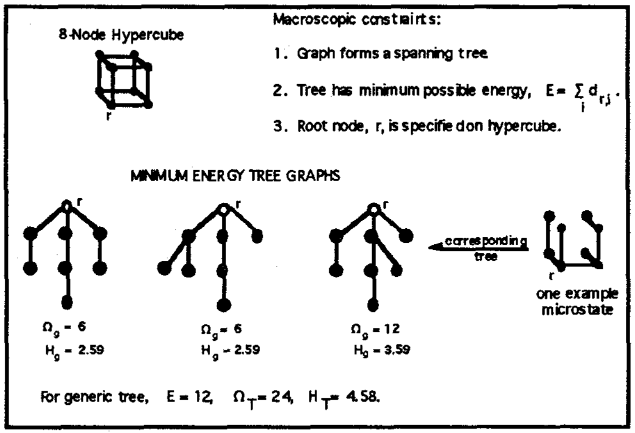
8-Node Hypercube
r
Macroscopic constraints:
1. Graph forms a spanning tree.
2. Tree has minimum possible energy, $E = \sum_i d_{r,i}$.
3. Root node, r, is specifie don hypercube.
MINIMUM ENERGY TREE GRAPHS
r
r
r
corresponding tree
r
one example microstate
$\Omega_g = 6$
$H_g = 2.59$
$\Omega_g = 6$
$H_g = 2.59$
$\Omega_g = 12$
$H_g = 3.59$
For generic tree, $E = 12$, $\Omega_T = 24$, $H_T = 4.58$.

Figure 2: Tree-Graph Entropy for 8-Node Hypercube.

coefficient, and the product is taken from d = 2 to N. These functions are plotted in Figure 3. Observe that for a given network size, n, the Boolean N-cube has greater tree entropy than the 2-mesh or 2-torus.

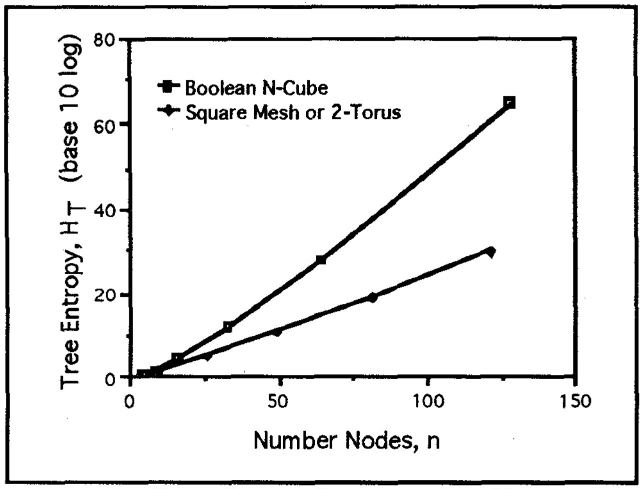

Figure 3: Tree-Graph Entropy for Hypercube, Square Mesh, and 2-Torus Networks.

2.2: Line-Graph Entropy

Consider an n-node network of specified topology and the following macroscopic constraints: (1) the graph is a line graph of length, d; and (2) the two end vertices of the graph are mapped onto two specified network nodes, which are separated by shortest path distance, f ≤ d. The *line graph disorder*, Ω_l, is then the number of different ways the line graph can be mapped between the two specified terminal nodes.

If the length of the line graph is specified as equal to the minimal internodal distance between the two terminal nodes (f = d), the number of different ways the graph can be one-to-one mapped between the nodes is called the *shortest path disorder*, $\Omega_p(d)$, and the *shortest path entropy* is then $H_p = \log_2 \Omega_p$. A symmetric multicomputer network can be characterized by its *shortest path disorder spectrum*, $P[d, \Omega_p(d)]$, where $P[d, \Omega_p(d)]$ is the number of nodes at distance d from a reference node having exactly $\Omega_p(d)$ shortest distance paths to the reference node. In an unfaulted symmetric multicomputer network, the shortest path disorder spectrum is independent of the choice of reference node since all nodes are isomorphic, and the multicomputer network has spectral components at d = 1 to d_m, where d_m is the *network diameter*, the maximum internodal shortest path distance.

As an example shortest path disorder spectrum for a symmetric network, consider an unfaulted Boolean N-cube. This network has *constant* shortest path disorder in the sense that $\Omega_p(d) = d!$ for any two nodes separated by shortest path distance d. The shortest path disorder spectrum for the Boolean N-cube is then non-zero only for single components at d = 1 to N with:

$$P[d, \Omega_p(d)] = P[d, d!] = (N, d),$$

where (N,d) is the binomial coefficient.

2.3: Network-Graph Entropy

Consider an n-node multicomputer network as a graph in which each computer node is a graph vertex and each inter-computer link is a graph edge. The *network disorder*, Ω_n, is then the number of different ways the network graph can be one-to-one mapped onto itself, where for counting mappings, identical (symmetrically similar) nodes are considered distinguishable in the multicomputer network but not in the graph. In graph theoretic terminology, Ω_n is the number of elements in the automorphism group of the network graph [4]. The *network entropy*, H_n, is then:

$$H_n = \log_2 \Omega_n.$$

For an unfaulted N-dimensional "square" mesh network:

$$\Omega_n = 2^N N!,$$

since for a network overlay there are 2^N possible origins (corners) and N! possible axis labelings relative to each origin. For an unfaulted Boolean N-cube:

$$\Omega_n = 2^N N! = n(\log_2 n)!$$

For an n-Node complete connection:

$$\Omega_n = n!$$

since for a network overlay there are n possible origins and (n-1)! possible axis labelings relative to each origin. Note that for an n-node network, Ω_n for the Boolean N-cube exceeds that for a mesh since $2^N = n$ for the N-cube and $2^N < n$ for the mesh. And for an n-node network, a complete connection has greater Ω_n than does any other topology.

3: Maximum Entropy Graph

In general, for a mapping experiment involving a generic graph, many specific graphs will satisfy the constraints of the experiment. The specific graph with the largest graph disorder, Ω_g, and therefore highest graph entropy, H_g, is called the *maximum entropy graph* for that experiment. That graph corresponds to more mappings onto the network than any other graph satisfying the constraints of the experiment. Observe for the mapping experiments of Figures 1 and 2 that the specific tree graphs with the highest graph entropy have $H_g = 3.59$ bits/microstate and are the maximum entropy tree graphs for their respective generic tree experiments.

Relative to lower entropy tree graphs, the maximum entropy tree shows several interesting characteristics, generally independent of network type, including: (1) higher fault tolerance; (2) maximum asymmetry about root node; (3) maximum nodal loading of one tree branch; (4) complementary relation to zero entropy "tree" graph; and (5) amenability to description by simple algorithm for MIMD implementation [5].

Example algorithms for generation of maximum entropy trees with global (macroscopic) constraints as listed in Figures 1 and 2 are as follows:

For an N-Mesh or N-Torus: Define the tree graph by building the tree outward from the root. For each node, among all its possible parents at equal distance from the specified root, connect to the parent node with the greatest number of shortest distance network paths to the root. ([6] described a MIMD implementation of this algorithm.) The graph disorder for the maximum entropy tree in an n-node 2-mesh and its corresponding 2-torus is:

$$\max \Omega_g = 3 \cdot 2^{(2n^{1/2} - 4)}.$$

For a Boolean N-Cube: Unlike the N-mesh or N-torus, in a Boolean N-cube all shortest distance paths of equal length to the root are equivalent. The above algorithm therefore cannot be applied. For an unfaulted Boolean N-cube, the maximum entropy tree graph contains two identical and parallel subtrees, each spanning an (N-1)-dimensional subspace of the N-cube. The maximum entropy tree graph for the N-cube is formed by doubling an (N-1)-cube maximum entropy tree and connecting the two trees by an edge between their roots. The specified root is retained, and the other becomes its daughter. The graph disorder for the maximum entropy tree is then recursively given as:

$$\max \Omega_{g,N} = N\,(\max \Omega_{g,N-1})^2,$$

where $\Omega_{g,N-1}$ is the graph disorder for a Boolean (N-1)-cube. Then:

$$\max H_g = \Sigma\ 2^i \log_2 (N-i),$$

where the summation is taken for i = 0 to N-2.

4: Entropy and Fault Tolerance

As previously mentioned, graph entropy for a multi-computer network relates to: (1) the efficiency of the network in passing messages between nodes; (2) the ease and flexibility of mapping global algorithmic graphs onto the network; and (3) the ability of the mapped graphs to reconnect and remap after nodal failure within the network (fault tolerance). Only the fault-tolerance characteristic will be discussed here. Suffice it to say that to optimize the above performance characteristics, define maximum entropy local algorithms such that p_i = constant for all mappings (microstates) and select multicomputer topologies that give high entropy for the global graph structures relating to the algorithms of interest.

The concept of graph entropy was derived while developing algorithms for the generation and repair of an optimum command/communication tree within a randomly faulting network of symmetrically configured MIMD multicomputer nodes [6]. It was found that tree-graph entropy characterizes the fault tolerance of a tree as related to the continuity of its structure after a multicomputer failure.

The greater the tree entropy, the less the effects of nodal failure on the continuity of existence of the tree graph as related to: (1) the immediate severity of the tree degradation (the number of nodes isolated from the tree root) after the occurrence of a fault; (2) the duration of the *blink time*, the time for a fully connected tree to reform after an isolated fault; (3) the tree energy increase (normalized per node) of the resulting reconnected tree; (4) the time for the tree to reform to a minimum energy configuration relative to the existing root; (5) the complexity of the subsequent tree reoptimization if the tree reconfigures to a minimum energy tree with respect to the network by reselection of a global optimum root node. Complexity of reoptimization is defined by the number of nodes, δ, that must reconfigure (select new parent node) and the total time required for root redefinition and tree reconfiguration into a global minimum energy tree. For example, for an n-node 2-mesh or 2-torus, assuming that a faulted node is identifiable by any unfaulted node physically linked to it: The blink time of a minimum entropy tree is twice that of a maximum entropy tree; the time for conversion to a minimum energy tree relative to the existing root for a minimum entropy tree is $(n^{1/2}/2 - d_r)$ times that for a maximum entropy tree, where d_r is the network path distance from the root to the fault; for tree reconfiguration by root redefinition, for a minimum entropy tree $2n^{1/2}/\delta$ times as many nodes are affected than for a maximum entropy tree and the time for reconfiguration is $n^{1/2}/(2\delta)$ times that for a maximum entropy tree.

In general, for topologies with variable shortest path entropy such as an N-mesh or N-torus, the maximum entropy tree forms in accord with a hierarchy of fault tolerance whereby the more principal a tree branch, the greater its fault tolerance. (The more principal the branch, the greater the number of shortest distance paths between the root and each node on the branch. In a 2-mesh with center root, for example, the square diagonals are most principal.) The maximum entropy tree structure thereby allows for minimal blink time and simple tree reformation after faulting. After an isolated fault, only the node or nodes linked directly to the fault must reconnect for the reformed tree to be minimum energy. Through its hierarchy of fault tolerance the maximum entropy tree moreover minimizes the number of nodes connected along paths most vulnerable to faulting. All nodes are connected as high as possible in the branching hierarchy of the tree so that the number of tree subbranches is maximized, the number of leaf nodes (nodes lying at the end of a branch) is maximized, and the subbranch lengths are minimized. Faulting along any branch then isolates a minimum number of nodes from the tree.

5: Epilogue

We can draw an analogy between the function of a centralized command/communication tree within a MIMD multicomputer network and the increasing complexity of a multi-cellular organism through formation of a central nervous system. These results on fault tolerance then support evolution through entropy increase since the complex structures of maximum entropy will have the greatest intrinsic survivability (fault tolerance). But that's another paper.

References

[1] D. A. Reed and D. C. Grunwald, "The performance of multicomputer interconnection networks," *Computer*, vol. 20, pp. 63-73, June 1987.

[2] F. W. Hamming. *Coding and Information Theory.* Englewood Cliffs, NJ: Prentice-Hall, Inc., 1980.

[3] F. P. Incropera. *Molecular Structure and Thermodynamics.* New York: John Wiley & Sons, 1974.

[4] R. J. Wilson. *Introduction to Graph Theory.* New York: Academic Press, 1972.

[5] P. L. Patterson, "Graph entropy in a multicomputer network," poster presentation at *Fourth SIAM Conference on Parallel Processing for Scientific Computing*, Chicago, Dec. 1989.

[6] P. L. Patterson, "Optimum command tree for faulted mesh of parallel computers," poster presentation at *Fourth SIAM Conference on Parallel Processing for Scientific Computing*, Chicago, Dec. 1989.