LOCALLY VS. GLOBALLY DENSE GRAPHS FOR COMPUTER NETWORKS

by

Leonard Uhr

Computer Sciences Technical Report #412

January 1981

Locally Vs. Globally Dense Graphs for Computer Networks

Leonard Uhr
University of Wisconsin

Abstract

This paper proposes several possible measures for "local density" of sub-regions of a graph, as opposed to measures of global density (e.g., the number of nodes in a regular graph of a given degree and diameter). Global and local density measures are then used to compare graphs. It is shown that, at least when programs are mapped onto sub-regions of a computer network in a way that reflects the flow of messages through the programs' procedures, graphs compounded from locally dense clusters are more appropriate (although they are not as dense globally as are graphs where distant nodes are laced together).

Index Terms: (d,k) graphs, Moore graphs, computer networks, packing density, compounding, graphs of computers, (n,d,k) graphs, size of graphs, network architectures.

We are today in a position where we can begin to build networks with several hundred computers. As VLSI technology matures during the 198Øs and 199Øs, it will become quite feasible to build networks with many thousands of computers. How such large numbers should be linked together is a major unsolved research problem.

A variety of different interconnection topologies have been suggested (e.g., rings, stars, arrays, trees, lenses, x-trees, lattices [1-5]), and several different criteria (e.g. average distance, diameter, maximal girth, symmetry, connectivity [1-3]) with which to evaluate them have been proposed. Almost certainly there is no single overriding criterion, or single provably best network architecture. Just as with the architecture of single-processor computers, or any large and complex systems, many alternative solutions are possible. A final choice depends upon a number of aspects related to the costs and feasibility of existing technologies.

Message-Passing, Diameter, Density, and Graph Structure

This paper examines the issue of packing computers (or processors) as close together as possible, so that the set of computers executing a single program will, when they must pass data and other kinds of messages from one to another, spend as little time as possible. This is important, since message-passing can quickly become the overriding factor in slowing down a network.

This problem has been cast into graph-theoretic terms by considering the problem of finding, for a particular diameter k (the shortest path between the most distant pair of vertices in the graph) and degree d (the number of edges touching each vertex), the regular (n,d,k) graph with as many vertices, n, as possible. Equate each of the computer network's individual computer (or processor) nodes with a vertex; equate the links joining nodes with edges.

Elspas [6] appears to have been the first person to suggest techniques for building graphs with as many nodes as possible. Storwick [7] gave a summarizing table of the densest graphs found so far, up to degree = 1Ø and diameter = 1Ø. Arden and Lee [8] and Toueg and Steiglitz [9] found a number of denser graphs of degree 3 and 4. Very recently, Imase and Itoh [1Ø] proved that de Bruijn networks [11] (shift registers) are asymptotically best found so far, and several compounding techniques have been found at Wisconsin that give densest graphs from roughly 5ØØ to 2Ø,ØØØ nodes [12,13], and a heuristic search program has achieved the densest graphs below 5ØØ nodes [13]. Thus a number of new graphs have been found, using several different techniques, that are promising candidates for computer networks - at least from the point of view of density.

However there are good reasons to think that diameter is not the most appropriate measure. Diameter is concerned with the most distant pairs of nodes. But when a program is mapped onto a network, it seems compelling that those procedures in the program that pass data and in other ways communicate with one another should be assigned to and executed by computers that are as close together as possible. That is, if two computers diameter distance apart must pass messages, the procedures they are executing should be re-assigned, to adjacent computers.

Ideally, the graph of message-passing between the program's procedures should determine the topology of the computer sub-graph that executes that program, so that messages are always passed over only a single link between adjacent nodes. Even better, the program's graph should be decomposed to assign a sub-graph to each network node such that message-passing between nodes is minimized and load of processing at each node is balanced.

To put this another way, diameter assumes that a program might be mapped onto a (possibly disjoint) sub-graph of the graph-network in any random way. Such a random mapping is, indeed, a possible alternative, and is being explored by several researchers. But it is certainly not the only alternative. And it pays the (very likely heavy) price of more frequent and more expensive message-passing, among other problems.

Measures of Local Density

As n increases for a graph of given d and k, we can say we are achieving "globally denser" graphs. There are several possible measures for "locally denser" graphs:

1) A simple measure of Local Density is (n,d,r,v_i), where r is the radial-distance from any single node, v_i, in G.

Then Average Local Density simply averages n over all v_i, for a given degree and radius.

Maximal Local Density is the largest n for any v_i;

Minimal Local Density is the smallest n for any v_i.

2) Next consider the (n,d,r,v_i) sub-graph, S, with n nodes, of a graph, G. Now define Density with respect to the diameter of S, for the (n,k,d,r,v_i) sub-graph.

For all (n,d,r,v_i) sub-graphs,

Average Local Density is the average (n,k,d,r,v_i) sub-graph;

Maximal Local Density is the largest local (n,k,d,r,v_i) sub-graph;

Minimal Local Density is the smallest local (n,k,d,r,v_i) sub-graph.

3) Another alternative decomposes G into (non-overlapping) sub-graphs, as follows: Decompose G into non-overlapping sub-graphs, S_j.

Now the (Average, Maximal, Minimal) Local Density is (average, max, min) of $\max(n,k,d,r,S_j)$ (for each d,k) for the maximal (known) decomposition. Decomposition means that links are broken between sub-graphs; these can now be used to join nodes within the sub-graph.

This measure is appropriate for, and can easily be computed for, compounded graphs [12,13], where local clusters are compounded into larger graphs by raising degree, and introducing new links outside the local cluster sub-graph. This measure, then, gets the (largest) Average Local Density of disjoint sub-graphs (n,d-1,k).

4) Rather than use diameter of sub-graphs, we might use

average distance between nodes, or weighted average distance, or simply a count of nodes 1,2,..r away from each node. These are probably all preferable to diameter, with average distance heavily weighted toward the smallest distance the best. Just as a single program should be mapped into as compact as possible a cluster, so within that cluster the more heavily computers intercommunicate the closer together they should be.

5) We might further vary these measures to allow for a decomposition of G using 2,3,...,s different sub-graphs (all of the same degree and diameter).

Evaluating Networks In Terms of Local Vs. Global Density

It is not clear which of the above measures are best, how closely they may be related, or whether other variants might be preferrable. But for the moment it seems instructive to explore how one of them might be used to evaluate and choose good network structures and to investigate the differences between local and global density (with respect to diameter or average distance).

For simplicity, consider local density using the diameter of the (n,d-1,k) sub-graph that decomposes the graph G. (This implicitly assumes that (n,d-1,k) graphs have been compounded, and that n is a number that will frequently be chosen for the number of computers to assign to one program.)

The following illustrate some of the kinds of interesting comparisons that can be made:

1. The (110,4,5) graph compounded from 11 (10,3,2) Petersen

graphs can obviously be decomposed into those 11 sub-graphs, which are known optimal. The (123,4,5) heuristically-connected tree, whose global density is somewhat better than (11Ø,4,5), if it can be decomposed at all reasonably (which is unknown, and doubtful), will certainly give poorer sub-graph clusters, on the order of (5,3,2).

2. Looking at larger graphs, where the de Bruijn shift registers achieve the best global density, we find a similar situation. Simply "tesselate" some known-dense sub-graph (e.g. (1Ø,3,2)), by starting with one sub-graph, linking each of its nodes to a new sub-graph, linking each of its nodes to a new sub-graph, etc., to give (at least) (1Ø,4,2). In contrast, the de Bruijn network, which, like the heuristically- completed tree, is basically locally a tree, will be on the order of (5,3,2) (that is, with a rather poorer topology from the point of view of local density).

3. When larger local graphs are desired, either good compounds, or good completed trees, can be chosen, and then tesselated, or compounded, or "embossed" (i.e., with nodes of degree d replaced by graphs with d nodes).

4. Emboss a graph with the desired local properties into a graph with the desired global properties whose degree equals the number of nodes in the first graph, by joining each link to the original node to a different node in the graph embossed into it.

5. Emboss complete bi-partite graphs into a tree, by replacing each node with n copies of an n-node graph, replacing each link

in the original tree by n links from each graph (one from each node). This increases local density, and puts many local cluster graphs into close proximity to one another.

Loosening the Coupling Between Clusters, to Increase Local Density

To the extent that message-passing can be contained within each cluster, so that the need to traverse links between clusters becomes rarer and rarer, there might be fewer and fewer inter-cluster links, freeing more and more links for other purposes.

This suggests building graphs using sub-graphs that are as dense as possible, and/or have whatever mix of properties are deemed most desirable, but linking 1,2,...n of each sub-graph's nodes to other sub-graphs. Therefore, assuming local density to be the only criterion, we should search for dense clusters that were regular graphs of degree d, but missing 1,2,...n joins.

Since the basic clusters will be quite small, on the order of tens, or at most a few hundred, nodes, heuristic searches of the sort Leland [13] and Toueg and Stieglitz [9] have used so successfully should work well. Note that such architectures would map especially well onto VLSI chips, if a whole cluster could be fabricated on a single chip. For now the number of links to other chip-contained clusters could be kept within the pin-fanout limits.

Possibly a more realistic, more useful and simpler variant is the following (this can be viewed as simply a different interpretation of the functions of nodes): Assume that a sub-graph cluster is given one extra join to each node (that is, its degree is raised by 1), and that these joins serve to handle the several auxiliary jobs of: a) input-output, b) access to mass stores and other devices, and c) links to other sub-graph clusters. Now these joins will be allocated to each job in proportion to the data-flow traffic of each. We might consider still another alternative interpretation: treat each input-output, mass store, or other device as simply another node in the cluster.

A variant that would allow the network to be adjusted to different traffic loads, under program control, would put a switch on this last link, so that it could be switched to span out to another cluster when that was needed, or to connect to input-output, mass store or other devices when that was needed (usually these will be needed at different times).

Discussion and Conclusions

Local Density seems an interesting measure to explore; it appears to be more appropriate than is global density for large computer networks where each program is mapped, for efficiency, onto a relatively compact sub-network. Probably the best way to achieve high local densities, and to exercise tight control over local density and simply be able to measure it easily, is to construct larger graphs from smaller, locally dense sub-graphs.

This, basically, suggests we should explore compounding and other construction techniques that combine good basic building-blocks, ones that are locally dense. This further allows one to shape these building blocks according to whatever combinations of criteria seem most desirable. As many links as are needed to handle spill-over can be used between clusters. Once these spanning links have been traversed, density (and any other desired characteristics) become good again, since another good sub-graph is now being used.

Measures of local density, and other local characteristics, make compounds of clusters look far more desirable and appropriate than do measures of global density, e.g., diameter. Thus augmented trees and de Bruijn networks, which give many densest graphs from the point of view of diameter, are relatively poor, when compared to compounded clusters, from the point of view of local diameter.

This seems quite reasonable when we remember that globally dense graphs are constructed by linking far-distant nodes, in order to draw regions together and reduce diameter. On the contrary, local density and other local characteristics are achieved by properly shaping each local sub-graph cluster. Note that this is a far easier task, since we now are working with a far smaller graph. And we have decomposed our problem into one where we can first shape the local structures, then use these as building blocks for the next-level structure. This process can be iterated, thus sub-dividing and simplifying our task as much as desired. And we can cycle through the design process, modifying the clusters and/or the compounding/spanning techniques, to best fit one another.

References

[1] Despain, A.M. and Patterson, D.A., X-tree: a tree structured multi-processor computer architecture, *Proc. Fifth Annual Symposium on Computer Architecture*, April, 1978, 144-151.

[2] Finkel, R.A. and Solomon, M.H., Processor interconnection strategies, *Computer Sciences Dept. Tech. Rept. 301*, University of Wisconsin, 1977.

[3] Finkel, R.A. and Solomon, M.H., The Lens interconnection strategy, *Computer Sciences Dept. Tech. Rept. 387*, University of Wisconsin, 1980.

[4] Sullivan, H., T. Bashkov, and D. Klappholz, A large scale, homogenous, fully distributed parallel machine, In: *Proc. Fourth Annual Symposium on Computer Architecture*, 1977, 105-124.

[5] Wittie, L.D., MICRONET: A reconfigurable microcomputer network for distributed systems research, *Simulation*, 1978, *31* 145-153.

[6] Elspas, B., Topological constraints on interconnection-limited logic, *Switching Circuit Theory and Logical Design*, 1964, *S-164*, 133-147.

[7] Storwick, R.M., Improved construction for (*d*,*k*) graphs, *IEEE Trans. Computers* (*Short Notes*), 1970, *19*, 1214-1216.

[8] Arden, B.W. and Lee, H., A multi-tree-structured network, *Proc. Fall Compcon 78*, 1978, 201-210.

[9] Toueg, S. and Steiglitz, K., The design of small-diameter networks by local search, *IEEE Trans. Computers*, 1979, *EC-28*, 537-542.

[10] Imase, M. and Itoh, M., Design to minimize diameter on building-block networks, paper in press, 1980.

[11] de Bruijn, D.G., A combinatorial problem, *Koninklijke Nederlandsche Academie van wetenschappen et Amsterdam*, *Proceedings of the Section of Sciences 49*, 1946, *7*, 758-764.

[12] Uhr, L., Compounding denser (d,k) graph architectures for computer networks, *Computer Sciences Dept. Tech. Rept.*, Univ. of Wisconsin, 1980.

[13] Leland, W., Li, Q., and Uhr, L., Globally dense (d,k) graphs for computer network architectures. *Computer Sciences Dept. Tech. Rept.*, Univ. of Wisconsin, 1980.