THE UNIVERSITY OF MICHIGAN

COMPUTING RESEARCH LABORATORY

A PROBABILISTIC ALGORITHM FOR SCATTERING INFORMATION IN A MULTICOMPUTER SYSTEM

Zvi Drezner and Amnon Barak[†]

CRL-TR-15-84

MARCH 1984

Room 1079, East Engineering Building
Ann Arbor, Michigan 48109
USA
Tel: (313) 763-8000

[†]On leave from The Hebrew University of Jerusalem, Israel.

engn

UMR1033

A PROBABILISTIC ALGORITHM FOR SCATTERING INFORMATION IN A MULTICOMPUTER SYSTEM

Zvi Drezner and Amnon Barak†

March 1984

SUMMARY

In this paper we develop a probabilistic algorithm for scattering information between the nodes of a multicomputer which consists of a set of independent computers that are interconnected by a local area communication network. This algorithm is useful when it is desired to reduce the number of messages and the time delay necessary to transmit information from any node to the other nodes of the multicomputer

The algorithm that we develop is based on messages which are sent by each node, every unit of time to a randomly selected node. We show that for large values of n, it is possible with probability tending to 1, to scatter information to a set of n nodes in $(1 + \ln 2)\log_2 n$ steps.

1. INTRODUCTION

Consider a multicomputer system which consists of n independent nodes that are interconnected by an Ethernet like local area communication network that allows a direct communication link between any pair of nodes as well as broadcast. To utilize such a system efficiently, the operating system of each node must have knowledge about some of the other nodes. For example, information about the availability and location of the resources, length of queues and the current load of some nodes allow other nodes to improve their performance by making better scheduling decisions.

† On leave from The Hebrew University of Jerusalem, Israel.

Suppose that one node possesses an information which has to be dispersed to all the other nodes. A simple algorithm for scattering this information is to have each node send (asynchronously) one message every unit of time. In order to speed-up the propagation of new information, each node includes newly arrived information in its future messages. We note that the scattering algorithm does not know the source and the final destination of the information, therefore we require that all the nodes use the same algorithm. There are two parameters for measuring the effectiveness of the scattering algorithm. First, since each message causes a communication overhead due to a context switch at the receiving nodes, it is necessary to reduce the total number of context switches for a given algorithm. For example, in a broadcast scheme the total number of context switches is n^2, each unit of time. The second parameter is the time delay until all the nodes receive the information. In this paper we develop an algorithm for efficient scattering of information between the nodes of a multicomputer in a manner which reduces the time delay and the total communication overhead.

Let C denote a measure for the complexity of the scattering algorithm. Assume that all the nodes use (asynchronously) the same algorithm and the same unit of time. Then we can define

$$C = S\,t$$

where S is the total number of context switches and t is the number of units of time necessary to scatter a given information to all the nodes. For example, for broadcasts $t = 1$ and $C = n^2$. The lower bound for C is $n \log n$ (where log denotes the base 2 logarithm). This bound can be achieved if one node originates the information and the

algorithm selects in each step a subset of nodes that did not receive this information previously. This bound however can not be achieved if more than one node originate the information.

For a message routing along a ring in which each node have at most two neighbors, it is easy to verify that $C = O(n^2)$. This follows from the fact that the network diameter (maximal distance between any pair of nodes, where the distance is the number of steps in the minimal path connecting the nodes) is half the number of nodes. When using trivalent (cubic) graphs in which each node has at most three neighbors, $S = 3n$. For example, the diameter of the Cube-Connected Cycles network [3] is $5/2 \log n + O(1)$, thus $C \approx 15n/2 \log n$. An improvement of this result can be obtained for the family of dense trivalent graphs that are discussed in [2]. In these cases the diameter of the network is $3/2 \log n + O(1)$, thus $C \approx 9n/2 \log n$. Further special cases of high density graphs for processor interconnection are discussed in [1]. As pointed in [2], the diameter of these graphs is bounded by $1.1 \log n$ therefore the complexity of the corresponding scattering algorithm is $C = 3.3n \log n$. An interesting open problem is to find an interconnection scheme for which C is minimized.

In the above cases the information routing is deterministic because each node sends and receives the information from the same nodes. An alternative approach is to use a nondeterministic scattering by which each node sends its information to a randomly selected node. In this paper we prove that for large values of n, the complexity of this random scattering algorithm is $(1 + \ln 2)n \log n$, which is better than the above deterministic algorithms. Another advantage of random scattering is its adaptability to changes in the network configuration and the availability of the nodes.

2. PROBABILISTIC INFORMATION SCATTERING

Consider a multicomputer system with a set of nodes numbered 1, 2, ...,n. Suppose that during each unit of time, each node selects another node at random and sends to it a message (color), thus $S = n$ for this case. At the beginning one node is colored. We find the time delay t (measured from the time this node sends the first message), necessary for the color to propagate to all the other nodes with a given probability. Note: each node that has the color includes it in its future messages. However, since there is no synchronization between the nodes we assume that each node sends the color if it were available to it at the beginning of the current unit of time. Thus we give a worst case analysis.

Lemma 1: Suppose that we independently draw h random integers in $[1, m]$. Let $Q(m, h)$ denote the probability that each of the m integers is drawn at least once. Then for $h \geq m > 1$:

$$Q(m,h) = \sum_{i=1}^{h-m+1} \binom{h}{i} \left(\frac{1}{m}\right)^i \left(1 - \frac{1}{m}\right)^{h-i} Q(m-1, h-i) . \tag{2.1}$$

Proof: The probability that a specific integer is drawn i times is: $\binom{h}{i} \left(\frac{1}{m}\right)^i \left(1 - \frac{1}{m}\right)^{h-i}$. Under the conditions of the lemma this integer must have been drawn at least once. If this integer was drawn i times, then the remaining $m - 1$ integers are drawn $h - i$ times. The probability that each of these $m - 1$ integers are drawn at least once is $Q(m-1,h-i)$ and the lemma follows.

Note that since $Q(1,h) = 1$, $Q(m,h)$ can recursively be calculated by (2.1) for every

$h \geq m > 1$.

Let $p(j, k)$ be the probability that exactly k nodes are colored after j steps.

Lemma 2: Given that at the beginning of iteration j exactly k nodes are colored. Then

$$p(j+1, k+m) = \binom{n-k}{m} R(m, k) ,$$

where

$$R(m,k) = \sum_{h=m}^{k} \binom{k}{h} \left(\frac{m}{n-1} \right)^{h} \left(\frac{k-1}{n-1} \right)^{k-h} Q(m,h) . \tag{2.2}$$

Proof: Let C_j be the subset of the k colored nodes at the beginning of iteration j. Suppose that at the beginning of iteration $j+1$ additional m out of the remaining $n-k$ nodes are colored. Let $M_j = C_{j+1} - C_j$ be this subset. Note that there are $\binom{n-k}{m}$ possible M_j sets. The probability that a certain node chooses another given node is $1/(n-1)$. During iteration j, there are k colored messages which are sent from C_j to C_{j+1}. Some of the messages are received by C_j and the remaining are received by M_j. The probability that h colored messages are received by M_j is:

$$\binom{k}{h} \left(\frac{m}{n-1} \right)^{h} \left(\frac{k-1}{n-1} \right)^{k-h} .$$

Therefore, the probability $R(m,k)$ that M_j is colored, is given by (2.2) and the probability that exactly $k+m$ nodes are colored after the iteration is:

$$p(j+1, k+m) = \binom{n-k}{m} R(m,k) .$$

The next theorem establishes the recursive relationship between the probabilities of two consecutive iterations.

Theorem 1: Let ν be the largest integer not greater than $i/2$. Then

$$p(j+1,i) = \sum_{m=0}^{\nu} \binom{n-i+m}{m} R(m,i-m)\, p(j,i-m) .$$

Proof: To get i colored nodes at the end of iteration j we must have at least $i/2$ colored nodes at the beginning of the iteration. The result is obtained by adding the probabilities of having $k = i - m$ colored nodes at the beginning of the iteration, each multiplied by the corresponding probability from Lemma 2 of adding m colored nodes.

In Table 1 we give sample values of the probabilities $p(j,n)$ for coloring n-node sets. In the next section we prove that for a given probability, j asymptotically approaches $(1 + \ln 2) \log n$.

j	n=4	n=8	n=16	n=32	n=64	n=128
2	0.2222	0.0	0.0	0.0	0.0	0.0
3	0.7160	0.0061	.	.	.	.
4	0.9099	0.2433	.	.	.	.
5	0.9726	0.6158	0.0243	.	.	.
6	0.9918	0.8443	0.2495	0.0002	.	.
7	0.9976	0.9430	0.5934	0.0385	.	.
8	0.9993	0.9800	0.8249	0.2806	0.0009	.
9	0.9998	0.9931	0.9326	0.6167	0.0613	.
10	0.9999	0.9977	0.9753	0.8355	0.3395	0.0029
11	1.0000	0.9992	0.9911	0.9363	0.6657	0.1020
12	.	0.9997	0.9968	0.9763	0.8600	0.4204
13	.	0.9999	0.9989	0.9913	0.9461	0.7240
14	.	1.0000	0.9996	0.9968	0.9799	0.8875
15	.	.	0.9998	0.9988	0.9926	0.9570
16	.	.	0.9999	0.9996	0.9973	0.9840
17	.	.	1.0000	0.9998	0.9990	0.9940
18	.	.	.	0.9999	0.9996	0.9978
19	.	.	.	1.0000	0.9998	0.9991
20	.	.	.	.	0.9999	0.9996
21	.	.	.	.	1.0000	0.9998
22	.	.	.	.	.	0.9999
23	.	.	.	.	.	1.0000

Table 1: Sample values of $p(j,n)$.

3. ASYMPTOTIC BEHAVIOR OF THE SCATTERING ALGORITHM

We now develop an asymptotic formula for j such that with a given probability, all the nodes are colored. Let $T_n(\alpha)$ be the number of iterations needed to get an expected number of $n - \alpha$ colored nodes, when starting with one colored node.

Lemma 3: $T_n(\alpha)$ is an upper bound for the number of steps needed to color the whole graph with a probability of $1 - \alpha$, i.e.,

$$p\,(T_n(\alpha),\, n) \geq 1 - \alpha\,.$$

Proof: For $j = T_n(\alpha)$,

$$n - \alpha = \sum_{i=1}^{n} i\, p\,(j,i) \leq n\, p\,(j,n) + (n-1)\,(1 - p\,(j,n)) = n - 1 + p\,(j,n).$$

Therefore, $n - \alpha \leq n - 1 + p\,(j,n)$, which yields $p\,(j,n) \geq 1 - \alpha$.

In the following analysis we assume a large value for n.

Lemma 4: Assume that at the beginning of an iteration there are pn colored nodes, $0 \leq p \leq 1$. Then the expected number of colored nodes at the beginning of the next iteration is $\hat{p}n$, where:

$$\hat{p} = 1 - e^{-p}\,(1 - p)\,.$$

Proof: Assume that during the course of the iteration there are xn colored nodes, $p \leq x \leq \hat{p}$. The probability that a node sends a message with the color is p. The probability that during the iteration an uncolored node receives the color is $1 - x$. Therefore, the expected increase in the number of colored nodes after each message is

$p(1-x)$. Thus, after one message the expected ratio of colored nodes is: $x + p(1-x)/n$. Assuming that $1/n$ is infinitesimally small, we define $1/n = \Delta t$. We have

$$\Delta x = p(1-x)\,\Delta t\,. \tag{3.1}$$

We now integrate (3.1) for t between 0 and 1, and x between p to $\hat{p}$:

$$\frac{1}{p}\int_{p}^{\hat{p}} \frac{dx}{1-x} = \int_{0}^{1} dt.$$

This leads to $\ln(1-p) - \ln(1-\hat{p}) = p$, or $1-\hat{p} = e^{-p}(1-p)$, and the lemma follows.

At the beginning the algorithm sets $p = p_0 = 1/n$. After that each iteration yields:

$$p_{j+1} = 1 - e^{-p_j}(1-p_j)\,. \tag{3.2}$$

Lemma 5:

$$T_{2n}(\alpha) \approx T_n(\alpha) + 1 + \ln 2\,.$$

Proof: Let $p_j(n)$ be the sequence defined by (3.2) with $p_0(n) = 1/n$. Since $p_0(2n)$ is small, $p_1(2n) \approx 2p_0(2n) = p_0(n)$. Therefore, $p_{j+1}(2n) \approx p_j(n)$. Let $\mu = T_n(\alpha)$. Then $p_\mu(n) = 1 - \alpha/n$. Thus, $p_{\mu+1}(2n) \approx 1 - \alpha/n$. Since $p_{\mu+1}(2n) \approx 1$, then by (3.2) $1 - p_{\mu+2}(2n) \approx (1 - p_{\mu+1}(2n))/e$. Therefore, $1 - p_{\mu+1+\ln 2}(2n) \approx \alpha/2n$ and the lemma follows.

As a result of Lemma 5:

Theorem 2:

$$T_n(\alpha) \approx (1 + \ln 2) \log n \approx 1.693 \log n.$$

In the previous analysis we assumed that a node sends the color only if the color was available to that node at the beginning of the current unit of time. Starting with one colored node, let $\hat{T}_n(\alpha)$ be the number of iterations needed to get an expected number of $n - \alpha$ colored nodes, when each node sends the color if it had it before sending the message. Note that Lemma 3 is true for $\hat{T}_n(\alpha)$ too.

Theorem 3:

$$\hat{T}_n(\alpha) \approx 2 \ln 2 \log n \approx 1.386 \log n .$$

Proof: The analysis is similar to that of Lemma 4. Equation (3.1) becomes

$$\Delta x = x(1 - x)\,\Delta t ,$$

because the probability that a node sends a colored message is x rather than p. This yields:

$$\frac{\hat{p}}{1 - \hat{p}} = \frac{p}{1 - p} e .$$

Since $p_0 = 1/n$, then $p_0/(1 - p_0) = 1/(n-1)$ and:

$$\frac{p_j}{1 - p_j} = \frac{e^j}{n - 1} .$$

Solving for j when $p_j = (n - \alpha)/n$ yields $e^j = (n - 1)(n - \alpha)/\alpha$, or $j = \hat{T}_n(\alpha) \approx 2 \ln n - \ln \alpha = 2 \ln 2 \log n - \ln \alpha$, and the theorem follows.

References

[1] W.E. Leland, R.A. Finkel, L.. Qiao, M.H. Solomon and L. Uhr, High Density Graphs for Processor Interconnection, Information Processing Letters, 12, (1981), 117-120.

[2] W.E. Leland and M.H. Solomon, Dense Trivalent Graphs for Processor Interconnection, IEEE Trans. Computers, C-31, (1982), 219-222.

[3] F.P. Preparata and J. Vuillemin, The Cube-Connected Cycles: A Versatile Network for Parallel Computation, Comm. ACM, 24, (1981), 300-309.