UM-CRL-TR-3-85

THE UNIVERSITY OF MICHIGAN

COMPUTING RESEARCH LABORATORY

ON THE NUMBER OF ACTIVE NODES IN A MULTICOMPUTER SYSTEM

A. Barak, Z. Dresner, and Y. Gurevich

CRL-TR-3-85

1079 East Engineering Bldg.
Ann Arbor, MI 48109

THE UNIVERSITY OF MICHIGAN

COMPUTING RESEARCH LABORATORY*

ON THE NUMBER OF ACTIVE NODES IN A MULTICOMPUTER SYSTEM

A. Barak, Z. Dresner, and Y. Gurevich

CRL-TR-3-85

April 1985

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

*Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the funding agency.

ON THE NUMBER OF ACTIVE NODES IN A MULTICOMPUTER SYSTEM

Amnon Barak[†], Zvi Drezner and Yuri Gurevich[*]

Department of Electrical Engineering and Computer Science

The University of Michigan, Ann Arbor, MI. 48109

ABSTRACT

In this paper we develop probabilistic algorithms for estimating the number of active nodes in a multicomputer system which consists of independent computers that are interconnected by a communication network. The algorithms are based on routine exchange of messages among the nodes of the multicomputer, using random routing. We show that each active node can find an ϵ-estimate of the fraction λ of active nodes in the system in time that depends only on ϵ and λ. The underlying approach can be used for finding various global properties of distributed systems with decentralized control.

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

[*] Partially supported by NSF grant MCS 83-01022

1. INTRODUCTION

Consider a multicomputer system which consists of N independent computers and workstations that are interconnected by an Ethernet-like communication network, which allows a direct communication link between any pair of nodes. To utilize such a system efficiently, each node must have global information about the system as well as local information about other nodes. For example, information about the (global) average load of the system allows the implementation of an efficient load balancing policy [1]. Similarly, information about the availability and location of resources allows individual nodes to improve their performance by making better scheduling decisions [2].

In this paper we are interested in algorithms for finding information about global properties of such a distributed system. More specifically, we develop algorithms such that each node can find an estimate for the number of active nodes in the system. (An active node is a node which is operational and is participating in the network activities.)

One can easily come up with many different algorithms for finding the number of active nodes. However, we are interested in algorithms which satisfy the following properties. First, we require uniformity; this means that all the nodes use the same algorithm and that there is no central control. We also require a low communication overhead; therefore we do not allow network broadcasts due to the high overhead (which results from the context switches by all the nodes, for every broadcast), and that the length of each message is bounded by $C \log N$ bits, where C is a small constant. (Note that the address part of a message already requires $\log N$ bits.) We further require that all the nodes use the same unit of time although they are not synchronized. Finally, we require that the total number of nodes, N, is known to all the nodes.

We assume that each node maintains a data frame which includes information about its local resources and estimates of global (system wide) information such as the number of active nodes. In order to allow rapid information spreading between the nodes while at the same time reduce the communication overhead, we assume that the information is transmitted by the nodes using (one-way) messages. To reduce the number of these messages, we require that each node sends one message every unit of time. Furthermore, in the algorithms that we develop we require that each node sends its message to a randomly selected node. As proved in [3], this allows an efficient and rapid information exchange among the active nodes. We note that while each node sends exactly one message each unit of time, it may receive several messages or no message at all, during this period; the probability of receiving many messages is extremely low. To Simplify the analysis and the exposition, we assume that every message sent to an active node, is received. The generalization to the case when a bounded fraction of messages, sent to active nodes, is not received, is fairly straightforward.

2. ESTIMATION OF THE NUMBER OF ACTIVE NODES

Let N be the total number of nodes and let n be the current number of active nodes. Let T be the common unit of time of all the nodes. For simplicity, assume $T = 1$. Recall that the nodes are not synchronized. The fundamental transmission algorithm which is used throughout this paper is that every unit of time, each active node sends a one-way message to a randomly selected node (whether active or not). In other words, the sending node does not know if the addressee is active.

Let the nodes be numbered 1, 2, ...,N. We assume that N and n have large values and that the value of n does not fluctuate rapidly. Then the number of messages received by each node, each unit of time, is a random variable whose distribution can be approximated by a Poisson distribution. Let λ be the expected number of messages received by a node in a unit of time. Then $\lambda = n/N$. Our strategy is to find an estimate for n (or λ) by monitoring the number of messages received by an active node over sufficient period of time.

2.1. Let $k(t)$ be the number of messages received by an active node between time t and time $t + 1$. Let $\bar{\lambda}(t)$ be the estimate for λ as measured over the last s units of time. Then,

$$\bar{\lambda}(t) = \sum_{i=1}^{s} k(t - i)/s \ .$$

Now, $k(t)$ has a Poisson distribution with a mean and a variance equal to λ [4]. Therefore, $\bar{\lambda}(t)$ can be approximated by the normal distribution (by the central limit theorem, for large s) with a mean λ and standard deviation $\sqrt{\lambda/s}$. We can now find the value of s, for any required accuracy. If a standard deviation ϵN is required for the estimate of n, then $\sqrt{\lambda/s} = \epsilon$, and therefore,

$$s = \lambda/\epsilon^2 . \tag{1}$$

For example, if $\epsilon = 0.05$ (a standard deviation of 5%), then $\lambda = 0.5$ yields $s = 200$ units of time. Note: we assume here that during the course of the computation, the number of active nodes does not change. Therefore, it is desirable to reduce s as much as possible, since the estimate is used until a new estimate is found after s units of time.

2.2. An improvement of the above scheme can be obtained by continuously updating $\bar{X}(t)$. When a new message is received, the sum $\sum_{i=1}^{s} k(t)$ is updated by adding the new value $k(t)$ and subtracting $k(t-s)$. This requires the storage of s values of $k(t)$ which is a drawback. To overcome this difficulty one can use the estimate $k(t-s) \approx \bar{X}(t)$. This modifies the definition of $\bar{X}$:

$$\bar{X}(t+1) = \frac{s\bar{X}(t) - \bar{X}(t) + k(t)}{s} .$$

Let $\theta = 1/s$, then:

$$\bar{X}(t+1) = (1-\theta)\,\bar{X}(t) + \theta\, k(t) . \tag{2}$$

Observe that this implies that $E(\bar{X}(t)) \rightarrow \lambda$, as t increases. Therefore, we turn our attention to the variance $V(\bar{X}(t))$, assuming a large value of t. It can be shown that $V(\bar{X}(t))$ converges to a value which is denoted by $V(\bar{X})$. Then by (2):

$$V(\bar{X}) = (1-\theta)^2\, V(\bar{X}) + \theta^2\, V(k(t))\,.$$

Since $V(k(t)) = \lambda$,

$$V(\bar{X}) = \frac{\theta}{2-\theta}\,\lambda\,.$$

If as before, $V(\bar{X}) = \epsilon^2$ is required, then (assuming a small value of ϵ and thus a small θ) it is possible to use the approximation $\theta/(2\text{-}\theta) \approx \theta/2$, thus

$$\theta = 2\epsilon^2/\,\lambda. \tag{3}$$

Since $\theta = 1/s$, the value of s in (3) is half its value in (1).

2.3. In the above discussion we assumed that n remains constant in time. Another criterion for measuring the effectiveness of the algorithm is the time delay until $\bar{X}(t)$ approaches a new value when n is changed. Assume that at time t_0, $\bar{X}(t_0) = \lambda_0$, and suppose that λ has changed at time t_0 to λ_1. We find the time t, such that

$$E(\bar{X}(t)) = \lambda_0 + \alpha\,(\lambda_1 - \lambda_0) \tag{4}$$

for a given $0 < \alpha < 1$.

After one unit of time,

$$\Delta(E(\bar{X}(t))) = \theta\,(E(k(t)) - E(\bar{X}(t))).$$

The expected value of $k(t)$ is λ_1. Thus, for the expected value of $\bar{X}(t)$:

$$\frac{d}{dt}E(\bar{X}(t)) = \theta\,(\lambda_1 - E(\bar{X}(t))), \tag{5}$$

with the initial condition $\bar{X}(t_0) = \lambda_0$. The solution of (5) is:

$$E(\overline{\lambda}(t)) = \lambda_1 - (\lambda_1 - \lambda_0)\, e^{-\theta(t-t_0)} \quad . \tag{6}$$

Substituting (4) into (6) and solving for t yields:

$$t = t_0 - \ln(1-\alpha) \,/\, \theta \, . \tag{7}$$

Example: for $\alpha = 0.5$ (50% of the difference accounted for), let $\epsilon = 0.05$ and $\lambda = 0.5$ as before, then $\theta = 0.01$. By (7) $t - t_0 = 69$ time units.

3. IMPROVING THE ESTIMATES

So far we took into account only the message arrival rate. In this section we develop an algorithm which uses additional information to increase the rate of convergence of $\overline{\lambda}(t)$ to λ. Suppose that each node includes in its message the value of its current estimate for λ. We show that this additional information can be used to improve the estimates for λ by individual nodes.

Let $\overline{\lambda}_i(t)$ be an estimate for $\lambda(t)$ by node i. Initially, $\overline{\lambda}_i(0)$ can be set to $1/N$. Assume that $k_i(t)$ messages with estimates $\lambda_1, \lambda_2, \ldots, \lambda_{k_i(t)}$ are received by node i during the time interval from t to $t+1$. Then the following algorithm is executed by each node once every unit of time.

Algorithm 1:

Step 1: *Calculate*

$$\overline{\lambda}_i \leftarrow (1-\theta)\, \frac{\overline{\lambda}_i + \sum_{j=1}^{k_i} \lambda_j}{k_i + 1} + \theta k_i \, ,$$

i.e.,

$$\bar{\lambda}_i(t+1) = (1-\theta)\,\frac{\bar{\lambda}_i(t) + \sum_{j=1}^{k_i(t)} \lambda_j}{k_i(t)+1} + \theta k_i(t).$$

where θ is a given constant.

Step 2: *Choose a random integer m, where $1 \le m \le N$.*

Step 3: *Send a message with $\bar{\lambda}_i$ (i.e., $\bar{\lambda}_i(t+1)$) to node m.*

Step 4: *Wait one unit of time, then return to Step 1.*

Let p_k be the probability that a node receives k messages during one unit of time. Let the mean and the variance of $\bar{\lambda}_i(t)$ be $E(t)$ and $V(t)$ respectively. Assuming that n does not change, we have:

$$E(t+1) = (1-\theta)\,E(t) + \theta\,\lambda\,. \tag{8}$$

It can be shown that $E(t) \rightarrow \lambda$. For a large value of N, it is reasonable to assume that the λ_i's are independent variables. (This was also verified by a simulation). Under these assumptions,

$$V(t+1) = (1-\theta)^2 \sum_{k=0}^{N} p_k\,\frac{V(t)}{k+1} + \theta^2\,\lambda\,. \tag{9}$$

Now, by the Poisson distribution:

$$p_k = \frac{\lambda^k}{k!}\,e^{-\lambda}\,.$$

Therefore,

$$\sum_{k=0}^{N} \frac{p_k}{k+1} \approx \sum_{k=0}^{\infty} \frac{p_k}{k+1} = \frac{1-e^{-\lambda}}{\lambda} .$$

Thus,

$$V(t+1) = (1-\theta)^2 \frac{1-e^{-\lambda}}{\lambda} V(t) + \theta^2 \lambda .$$

From this one can see that $V(t)$ converges to V, where

$$V = \frac{\theta^2 \lambda}{1-(1-\theta)^2(1-e^{-\lambda})/\lambda} .$$

Assume that a standard deviation ϵ is required for the estimate of λ. Then:

$$\epsilon^2 = \frac{\theta^2 \lambda}{1-(1-\theta)^2(1-e^{-\lambda})/\lambda} .$$

Assuming a small ϵ, this yields,

$$\theta \approx \epsilon \sqrt{(\lambda + e^{-\lambda} - 1)/\lambda^2} \approx \epsilon/\sqrt{2} . \tag{10}$$

This last expression is based on the approximation:

$$e^{-\lambda} \approx 1 - \lambda + \lambda^2/2 .$$

Note that the analysis leading to (7) is also valid for (10) for the new value of θ, when $\lambda_1 \leq \lambda_0$ (and no new node become active in $t = t_0$). However, when $\lambda_1 > \lambda_0$ (and for simplicity, no active node become passive in $t = t_0$), the nodes which joined the network have initial estimates for λ close to zero. Therefore, $E(\bar{X}(0)) = \lambda_0^2/\lambda_1$ rather then λ_0. Consequently, in order to find t which satisfies (4), α in (7) should be replaced by

$$\frac{\lambda_0 + \alpha(\lambda_1 - \lambda_0) - \lambda_0^2/\lambda_1}{\lambda_1 - \lambda_0^2/\lambda_1} = \alpha + (1-\alpha)\lambda_0/(\lambda_1 + \lambda_0).$$

The calculated values of $t - t_0$ for various values of λ are given in Table 1, in Section 5. Simulations confirmed these results (which are better than the results obtained in Section 2).

4. THE COUNTER METHOD

An alternative method for estimating the number of active nodes is to monitor the "life span" of the received messages, e.g., to count the number of nodes through which the message has passed. As before, every active node sends one message each unit of time.

Assume that each message has a counter C_i. Each time that a message reaches an active node, its counter is increased by one. When several messages are received by a node, they are merged into a single message and the sum of the counters (each increased by one) forms the value of the new counter. When a new node joins the network or when a node does not receive any message during the last unit of time, it initiates a new message with a counter value zero.

First we find a relationship between the value of the counter and the number of active nodes. Let $C_i(t)$ be the counter generated by node i in time t, and $C(t)$ be the average of all the $C_i(t)$.

Lemma 1: $E(C(t+1))$, the expected counters' average in time $t+1$, is:

$$E(C(t+1)) = \frac{n}{N}(C(t)+1).$$

Proof: The probability that a message is received by an active node is n/N; in this case its counter is increased by one. Note that a merge does not change $C(t)$. Otherwise, a new message with counter value zero is generated. This however does not add to the expected value.

Lemma 2: Let $\hat{n}(t)$ denote the estimate for n by means of the counter method. Then in a steady state (i.e., when $E(C(t+1)) = C(t)$),

$$\hat{n}(t) = N\frac{C(t)}{C(t)+1} \tag{11}$$

Proof: Follows from Lemma 1 by substituting $E(C(t+1)) = C(t)$.

If we get an estimate for the counters' average, then (11) can be used to estimate n. Further improvement can be obtained if each message includes the estimate of $C(t)$ by the sending node. Then an algorithm, similar to Algorithm 1 can be defined.

Algorithm 2:

Assume that $k_i = k_i(t)$ messages are received by a node i between time t and time $t+1$. The j-*th* message contains a counter C_j and an estimate $\hat{C}_j$ for the average counter of the system provided by the sending node.

Step 1: *Assemble a new counter* $C = \sum_{j=1}^{k_i} (C_j + 1)$.

Step 2: *Find the new estimate* $\hat{C} \leftarrow (1-\theta)\frac{\hat{C} + \sum_{j=1}^{k_i} C_j}{k_i + 1} + \theta C$, *where θ is a given constant.*

Step 3: *Find* $\hat{n} = N\frac{\hat{C}}{\hat{C}+1}$.

Step 4: *Choose a random integer m, where $1 \leq m \leq N$.*

Step 5: *Send a message which includes C and $\hat{C}$ to node m.*

Step 6: *Wait one unit of time, then return to step 1.*

Note: When $n \approx N$ the counters may overflow since the messages remain in the system for a long time. Therefore, we have limited in the simulation the value of the counters in step 2 of Algorithm 2, by a constant. For example, requiring a 5% accuracy in the value of the estimate, we limit the value of the counter by 39, which theoretically corresponds to 2.5% deviation, i.e., $n = 0.975\, N$ in (11).

5. RATE OF CONVERGENCE

In order to check the rate of convergence of the algorithms discussed in the last two sections, we have simulated an environment which included $N = 100$, $N = 200$ nodes and $\lambda = n/N = 0.1, 0.2, ..., 1.0$. Initially, we set $\lambda = 0.1$ and executed the algorithms until $E(\bar{X}) \geq \lambda - 0.05$. The number of iterations was recorded. We then executed the algorithm until the variance of $\bar{X}(t)$ stabilized (on a value of about 0.05^2). We then increased λ by 0.1, and repeated the simulation until $\lambda = 1.0$. The second phase of the simulation was done in a reverse order, until $\lambda = 0.1$. The entire simulation was repeated ten times and the resulting average numbers of iterations are given in the tables below.

	n increases			n decreases		
λ	N=100	N=200	$t-t_0$	N=100	N=200	$t-t_0$
0.1	17.0	17.7	19.8	19.4	22.2	19.8
0.2	29.7	29.8	31.4	17.8	23.0	19.8
0.3	33.6	31.2	34.4	19.6	21.1	19.8
0.4	32.5	35.8	35.8	22.6	19.1	19.8
0.5	37.7	37.6	36.6	24.6	20.5	19.8
0.6	38.5	34.8	37.1	20.6	19.3	19.8
0.7	33.9	36.2	37.5	23.2	20.0	19.8
0.8	38.0	37.0	37.8	21.1	23.1	19.8
0.9	35.2	36.6	38.0	19.2	18.7	19.8
1.0	38.4	39.3	38.1	-	-	-
Average	33.45	33.60	34.65	20.90	20.78	19.80

Table 1: Average number of iterations using Algorithm 1.

	n increases		n decreases	
λ	N=100	N=200	N=100	N=200
0.1	14.5	15.4	18.9	18.7
0.2	24.0	25.5	15.3	19.7
0.3	29.8	25.9	18.5	18.0
0.4	28.7	28.2	20.2	19.2
0.5	25.8	29.9	20.5	21.0
0.6	26.0	28.8	21.8	21.3
0.7	24.7	26.7	28.6	25.0
0.8	24.3	24.5	32.6	33.7
0.9	24.7	24.7	28.6	29.0
1.0	35.2	36.3	-	-
Average	26.07	26.59	22.78	22.84

Table 2: Average number of iterations using Algorithm 2.

One observation is that when n increases, Algorithm 2 is better than Algorithm 1 and when n decreases, Algorithm 1 is better than Algorithm 2 for $n \geq 0.6N$. Apparently, Algorithm 2 is better than Algorithm 1 by about 3 iterations. Another observation is that the simulation confirms that the average number of iterations is independent of the value of N.

6. CONCLUSIONS

In this paper we developed a class of algorithms for estimating the number of active nodes in a multicomputer system. We assumed that the topology of the communication network is a complete graph, that the nodes are not synchronized and that the network control is decentralized. These algorithms use messages between the nodes in a random routing.

We show that each node can find an estimate of the number of active nodes in the system. Furthermore, we showed that the accuracy of these estimates is independent of the number of nodes in the system. A simulation that demonstrates the effectiveness of the algorithms was also given.

REFERENCES

[1] Barak A. and Drezner Z., Distributed Algorithms for the Average Load of a Multicomputer, Computing Research Laboratory TR-17-84, The University of Michigan, Ann Arbor, Michigan, March 1984.

[2]. Barak A. and Shiloh A., A distributed load balancing policy for a multicomputer, Software Practice & Experience, to appear.

[3]. Z. Drezner and A. Barak, An Asynchronous Algorithm for Scattering Information between the active Nodes of a Multicomputer System, Department of Elec. Engin. and Computer Science, The University of Michigan, Ann Arbor, MI. 48109, March 1985.

[4]. Feller W., An Introduction to Probability Theory and its Applications, Jon Wiley & Sons, Inc., 1971.

AIIM SCANNER TEST CHART # 2

Spectra

4 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:",./?$0123456789
6 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:",./?$0123456789
8 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:",./?$0123456789
10 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:",./?$0123456789

Times Roman

4 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:",./?$0123456789
6 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:",./?$0123456789
8 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:",./?$0123456789
10 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:",./?$0123456789

Century Schoolbook Bold

4 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:",./?$0123456789
6 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:",./?$0123456789
8 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:",./?$0123456789
10 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:",./?$0123456789

News Gothic Bold Reversed

4 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:",./?$0123456789
6 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:",./?$0123456789
8 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:",./?$0123456789
10 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:",./?$0123456789

Bodoni Italic

4 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:",./?$0123456789
6 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:",./?$0123456789
8 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:",./?$0123456789
10 PT ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz;:",./?$0123456789

Greek and Math Symbols

4 PT ΑΒΓΔΕΞΘΗΙΚΛΜΝΟΠΦΡΣΤΥΩΧΨΖαβγδεξθηικλμνοπφρστυωχψζ≧∓",./≦±=≠°><⊁⊀⩾⩽≡
6 PT ΑΒΓΔΕΞΘΗΙΚΛΜΝΟΠΦΡΣΤΥΩΧΨΖαβγδεξθηικλμνοπφρστυωχψζ≧∓",./≦±=≠°><⊁⊀⩾⩽≡
8 PT ΑΒΓΔΕΞΘΗΙΚΛΜΝΟΠΦΡΣΤΥΩΧΨΖαβγδεξθηικλμνοπφρστυωχψζ≧∓",./≦±=≠°><⊁⊀⩾⩽≡
10 PT ΑΒΓΔΕΞΘΗΙΚΛΜΝΟΠΦΡΣΤΥΩΧΨΖαβγδεξθηικλμνοπφρστυωχψζ≧∓",./≦±=≠°><⊁⊀⩾⩽≡

White

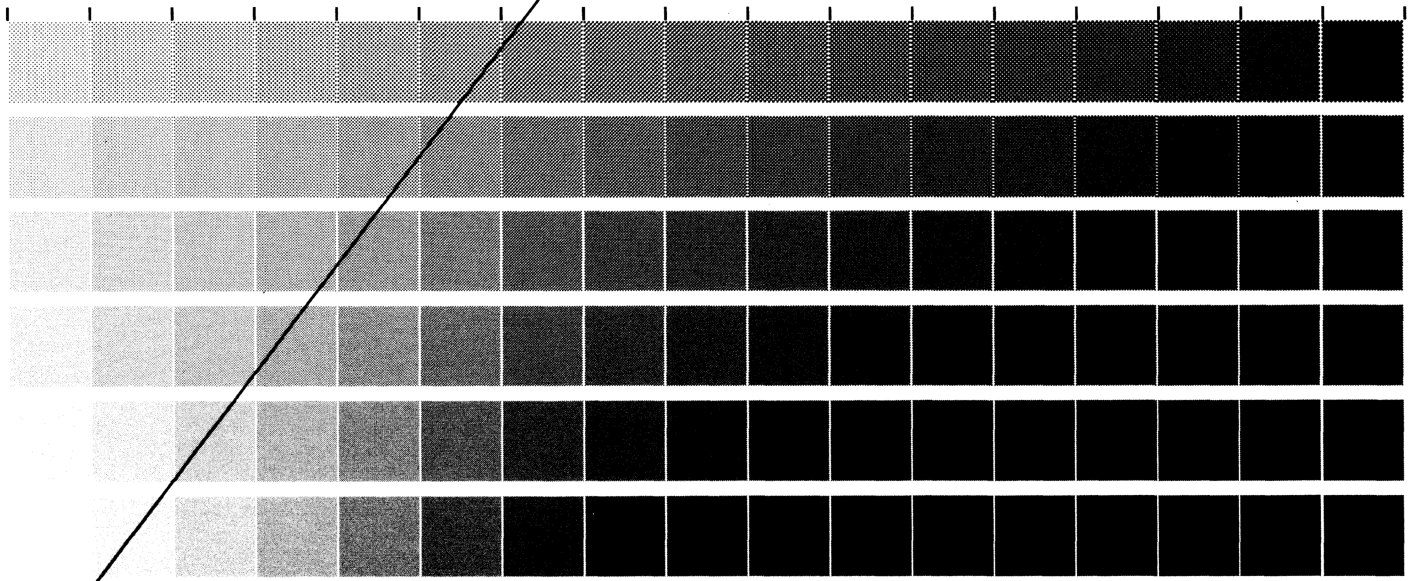
Black

Isolated Characters

e	m	1	2	3	a
4	5	6	7	o	°
8	9	0	h	l	B

A4 Page 6543210

MESH HALFTONE WEDGES



RIT ALPHANUMERIC RESOLUTION TEST OBJECT, RT-1-71

PRODUCED BY GRAPHIC ARTS RESEARCH CENTER

ROCHESTER INSTITUTE OF TECHNOLOGY, ONE LOMB

MEMORIAL DRIVE, ROCHESTER, NEW YORK 14623