Technical Note
Operating Systems

Anita K. Jones
Editor

On an Improved Algorithm for Decentralized Extrema Finding in Circular Configurations of Processors

Wm. Randolph Franklin
Rensselaer Polytechnic Institute

This note presents a more efficient algorithm for finding the largest element in a circular list of processors when messages can be passed in either direction. It passes 2*N*floor(lg *N*) + 3*N* messages in the worst case, compared to Chang and Roberts' *N*(*N* + 1)/2 and Hirschberg and Sinclair's 8*N* + 8*ceiling(*N* lg *N*) messages. The technique is a selective elimination of possible processes, which then merely relay future messages between the remaining contenders.**

CR Categories and Subject Descriptors: C.2.5 [**Computer Communication Networks**]: Local Networks—*rings* D.44 [**Operating Systems**]: Communications Management—*message sending.*

General Term: Algorithms

Additional Key Words and Phrases: decentralized algorithms, distributed systems

Chang and Roberts [1] present an algorithm for finding the largest element stored in a circular set of processes without a central controller, and whose size N is not known in advance. The measure of complexity is the total number of message passes, and their algorithm has an expected complexity of $O(N \lg N)$, but the worst case is $O(N^2)$. This note presents a superior algorithm, which, if the ring is bidirectional, has a worst case complexity of $O(N \lg N)$.

We will define an inactive process as one that knows that it is not the largest; the other processes are active. The two neighbors of an active process are those active processes closest to it in each direction along the ring. In the degenerate case of a ring with only two active processes, each becomes the two neighbors of the other; similarly, if there is only one active process, it becomes both of its neighbors.

The above definitions allow us to define the extrema finding algorithm. Its primary action is the repetition of the following step (we shall see the stopping criterion shortly).

> Each active process sends a message with its value to each of its neighbors and receives such messages from its two active neighbors. If either of the messages it receives is larger than its value, then it makes itself inactive.

The process of sending a message to an active neighbor is apparently complicated by the fact that a given process does not know the exact locations of its active neighbors. This is, in fact, no problem if we pass messages by the convention that inactive processes simply pass on received messages from either direction in the same direction, while active processes do not. Thus, during each step every inactive process receives and forwards two messages, while each active process transmits and receives two messages. The total number of message passes required for each step of the algorithm is $2N$, regardless of the number of currently active processes.

The repetition of steps terminates when in some step a process receives a message from itself; this implies that it is the only active process left and that its value is the largest of the set. As a final action, that process announces that fact to all the other processes in N message passes. We will now analyze the performance of this algorithm.

In each step, at least half of the active processes are eliminated since a process remains active if and only if it is larger than both of its neighbors. Thus there can be at most FLOOR(lg N) steps until there is only one active process. After one more step, that process knows that it is the only one, and sends its final notification. Since each step requires $2N$ message passes, and the final notification another N, the total number of message passes in the worst case is $2N$*FLOOR(lg N) + $3N$. To demonstrate that this complexity can be achieved, we will consider the case that $N = 2^k$ and define a sequence P_i for i from 0 to $N - 1$ that represents the value of the ith process. Specifically, represent i as a k-bit binary number, and let the binary representation of P_i be the left to right reversal of the bit string for i. For example, if $N = 8$, then the sequence P is 0, 4, 2, 6, 1, 5, 3, 7. At each step every second active process is inactivated. In the best case of this algorithm, the processes are arranged in order, and the number of message passes is $5N$. This algorithm also requires a linear number of message passes for the worst case listed in [1].

If we let the unit of execution time be the delay in sending a message from a process to its neighbor, then the worst case execution time is $O(N \lg N)$. We conjecture that the average time is $2.5N$. Sample tests corroborate this but we have not proven it.

This paper was submitted contemporaneously with the manuscript for the published note of Hirschberg and Sinclair [2]. It languished in the editorial process through no fault of the author. The *Communications of the ACM* editors regret tardy publication.

Author's Present Address: Wm. Randolph Franklin, Electrical, Computer, and Systems Engineering Department, Rensselaer Polytechnic Institute, Troy, NY 12181.

Acknowledgments. This material is based upon work prepared for the National Science Foundation under Grant No. ENG-79-08139 and ECS-80-21504. This note has benefitted from extensive rewriting by Jon Bentley, one of the referees.

Postscript. The algorithm presented here was discovered independent of, and simultaneous with [2]. The Hirschberg and Sinclair algorithm requires $8N + 8$ CEILING ($N \lg N$) message passes in the worst case. The algorithm presented here requires $3N + 2N$ FLOOR ($\lg N$) message passes in the worst case.

Received 5/79; accepted 9/81

References

1. Chang, E. and Roberts, R. An improved algorithm for decentralized extrema-finding in circular configurations of processes. *Comm. ACM, 22,* 5 (May 1979), 281–283.
2. Hirschberg, D.S. and Sinclair, J.B. Decentralized extrema-finding in circular configurations of processors. *Comm. ACM, 23,* 11 (Nov, 1980), 627–628.

Applications:
Management Science and Operations Research
Harvey Greenberg
Editor

Estimating and Improving the Quality of Information in a MIS

Richard C. Morey
Duke University

Most discussions of MIS's assume that the information in the records is error-free although it is recognized that errors exist. These errors occur because of delays in processing times, lengthy correction times, and, overly or insufficiently stringent data edits. In order to enable the user to implement data edits and correction procedures tailored to the degree of accuracy needed, this paper presents functional relationships between three common measures of data quality. The MIS addressed is one where records in a MIS are updated as changes occur to the record, e.g., a manpower planning MIS where the changes may relate to a serviceman's rank or skills. Since each of the updating transactions may contain an error, the transactions are subjected to various screens before the stored records are changed. Some of the transactions including some that are correct, are rejected; these are reviewed manually and corrected as necessary. In the meantime, the record is out of date and in error. Some of the transactions that were not rejected also lead to errors. The result is that at any given time the MIS record may contain errors.

For each of several error control mechanisms, we show how to forecast the level of improvement in the accuracy of the MIS record if these options are implemented.

CR Categories and Subject Descriptors: K.6.4 [System Management]—*management audit.*

General Terms: Measurement, Performance, Reliability

Additional Key Words and Phrases: errors, estimation, cost-effectiveness, edits, turnaround times, processing, transactions

The effort was supported by the U.S. Marine Corps under Contract N00014-70-C-0119 with the Office of Naval Research; the author also wishes to acknowledge Thomas C. Varley for drawing the author's attention to this problem, as well as editor's and referees' comments which have substantially improved the exposition.

Author's Present Address: Richard C. Morey, Center for Applied Business Research, Fuqua School of Business Administration, Duke University, Durham, NC 27706