Proceedings of the 1983
Winter Simulation Conference
S. Roberts, J. Banks, B. Schmeiser (eds.)

PERFORMANCE ANALYSIS OF A DISTRIBUTED SIMULATION ALGORITHM BASED ON ACTIVE LOGICAL PROCESSES*

David L. Davidson
Paul F. Reynolds, Jr.
Department of Applied Math and Computer Science
The University of Virginia
Charlottesville, Virginia 22901

RESEARCH SUMMARY

1. INTRODUCTION

The SRADS distributed simulation algorithm, which makes use of "active logical processes" as described elsewhere in this proceedings and in Reynolds (1982), was studied on a network of processors. Results of previous experiments (O'Hallaron 1983) indicated that SRADS could be expected to perform very well under a wide range of conditions. Our implementation was designed to study the algorithm for use in distributed logic simulation, and to provide more detailed information about the algorithm itself. A more detailed discussion can be found in Davidson (1983).

2. OVERVIEW OF THE IMPLEMENTATION

The SRADS algorithm was implemented on a network of three homogenous microprocessors. The communication topology allowed for each processor to communicate directly with each of the other processors. Communications were done via low-to-medium speed lines.

The microprocessor network was an inexpensive way to conduct a feasibility study, with its only drawback being the relative slowness with which work was completed. However, this slowness was compensated for when experiments were designed, so that simulation results could reflect the effects of <u>relative</u> interdependencies between, say, processor speeds and communication speeds, independent of absolute processing speeds.

In addition to the communications software necessary to implement the SRADS algorithm, a combinational logic simulator was written for the network. An attendant preprocessor allowed the specification of partitions of a user-specified logic network so that experiments could be conducted on any number of the available processors.

Special provisions were built into the logic simulator for the case where the simulation was being conducted on one processor. Since the single processor case would represent a benchmark for comparing multi-processor simulations, we wanted to ensure that it executed without the burden of any multi-processor considerations.

3. NATURE OF EXPERIMENTS AND RESULTS

We assume that a process, representing a well defined task, is assigned to a dedicated processor. A process which represents a well defined physical process has been called a "logical process", or LP for short (Chandy and Misra 1979). We use that notation here.

Previous studies (O'Hallaron 1983) indicated that performance of SRADS-based distributed simulations would depend on three primary factors: 1) frequency of communications between LP's relative to the amount of processing required between communications, 2) balance in mean workload among LP's, and 3) variance in the workload within LP's. We studied the effects of these factors using two approaches. First, we ran a small set of simulations of actual logic networks (counters, adders, etc.) Second, we modified the simulator somewhat so that we could define logic networks which had user specified workloads and workload variances.

The effects of frequency of communications were studied as follows. We determined the actual communication costs (in terms of real time) associated with sending a message from one processor to another. This real time value was used as the basic time unit (t) for experiments. Communication frequency was then expressed in terms of mean workload between attempted communications (MWBC). An MWBC of 100t means that 100 units of simulation related work is completed, on the average, between each attempt by the LP to send a message to another LP. By using MWBC we were able to derive implementation-independent results.

*This research was supported in part by IBM Corporation.

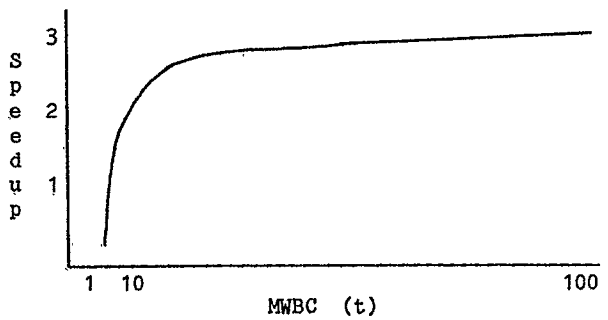

Fig. 1: Effective Speedup as a Function of MWBC (Two LPs, MWBC equal in each)

In Figure 1 we show the effects of MWBC on total finisning time of a simulation in which LP workloads are assumed to be constant. Our observations here are that as long as the MWBC is approximately 10t or greater we derive benefit from using distributed simulation. That is, the amount of processing that must be done by an LP between attempted communications should be approximately an order of magnitude greater than the cost of sending a message from one processor to another. The performance degradation, particularly between 10t and 1t, can be attributed primarily to increased communications-related events list processing within individual LPs.

Setting the MWBCs in two LPs to unequal values greatly reduced the events-list related performance degradation described above. We found that the ratio between differing MWBCs in two LPs had little effect upon finishing times. For example, an experiment in which one LP had a MWBC of 100t and the second a MWBC of 1t was only 12% slower than a second experiment in which each LP had an MWBC of 100t.

Workload balance was studied by defining a total mean volume of work which could then be split between LPs in different proportions. Workload ratios between two LPs were varied from the ideal of 1, in which each LP performed equal amounts of work, to a worst case ratio of 1/3. Our results, using either constant or normally distributed workloads, indicated that performance depends profoundly upon workload ratio. Finishing times at a ratio of 1/3 were from 50% to 100% worse than those at a ratio of 1, depending upon MWBC values within each LP. However, even at a ratio of 0.33 two LPs were able to complete in a time well below that needed for a single LP with an equivalent total workload.

The effect of workload variance was studied by repeating many of the preceeding (normal distribution) workload balance experiments using different variances. With any given workload ratio we found that large changes to workload variance resulted in little performance degradation, typically less than 10% even in instances where variance was increased by a factor of 25. We also observed that as workload ratios became more unbalanced the effects of workload variance became less noticeable.

A series of experiments using negative exponentially distributed workloads yielded results in keeping with those described above; the effects of mean workload imbalances were more profound than those of workload variances.

Experiments conducted using three LPs gave results showing more dependence upon workload variance. Constant workload experiments showed excellent completion times, sometimes greater than four times faster than an equivalent single LP simulation. Normally or exponentially distributed workload experiments showed less marked gains. Network progress is necessarily governed by the speed of the slowest LP, which is that one with the heaviest workload during a given interval. With workload variance possible in each LP, as the number of LPs increases there is a greater probability that at any instant one LP will have a workload greater than the common mean; that LP will slow the entire network.

4. CONCLUSION

The experiments we have conducted indicate that the Active Logical Process method of distributed simulation, as embodied in the SRADS algorithm, is well suited to logic simulation. The most influential performance factor appears to be that of workload balance between the cooperating LPs. Frequency of communications between LPs is a secondary factor, and variance in an individual LP's workload seems of tertiary importance. As larger numbers of LPs are connected, the influence of workload variance will become a more important factor.

Additionally, our experiments indicate that simulation applications are particularly well suited to distributed computation. Not only is the meaningful simulation work distributed, but the total cost of events list maintenance is also reduced as a result of partitioning. Many of our experiments on two LPs showed finishing times less than one-third that of an equivalent sequential simulation; three-LP experiments sometimes were over four times faster. Each LP is responsible for a smaller number of events, and the processing cost associated with each event decreases as well.

REFERENCES

Chandy KM, Misra J (1979), Distributed Simulation: A Case Study in Design and Verification of Distributed Programs, IEEE Transactions on Software Engineering, SE-5, pp. 440-452.

Davidson DL, Reynolds, PF (1983), Implementation and Performance Analysis of SRADS. DAMACS Technical Report # 83-13, Dept. of Applied Math and Computer Science, Univ. of Virginia, Charlottesville, Virginia.

O'Hallaron D (1983), Analysis of a Model for Distributed Simulation, Masters Thesis, Dept. of Applied Math and Computer Science, Univ. of Virginia, Charlottesville, Virginia, January, 72 pp.

Reynolds PF (1982), A Shared Resource Algorithm for Distributed Simulation. In: Proc. of the Ninth Annual Int'l. Computer Architecture Conference, Austin, Texas, April, pp. 259-266.