TRANSACTION TAGGING IN HIGHLY CONGESTED NETWORK SIMULATIONS

Lee W. Schruben, Jack Chung-yu Ho, Joan E. Silverman
and Nelson F. Steele
Cornell University
School of Operations Research and Industrial Engineering
Ithaca, New York 14853

Robert G. Sargent
Department of Industrial Engineering and Operations Research
Syracuse University
Syracuse, New York 13210
(work done while on leave to Cornell University)

ABSTRACT

Process flow network simulation models are often constrained by computer time and memory when there are a large number of transactions active in the model. This paper describes an approach to alleviate the storage problems associated with transaction congestion in large scale network simulation models. Transaction **tagging,** the selective monitoring of entities, is the approach presented. The effect of tagging on primary memory requirements, the information obtained from a simulation, and the cost of running a simulation is investigated. Issues regarding the use of tagging are explored. For certain types of highly congested network simulation models, tagging can be used to reduce storage requirements without significant information loss.

1. Introduction to the Problem

The use of simulation is limited in many applications by the size of the system under study. When the process world view is employed, as in the common network oriented simulation languages, the maximum number of transactions active in the model at any one time can become very large. The results are long processing times and a large main memory requirement. In fact, some models have so many concurrent transactions that the simulation cannot be accommodated within the amount of main memory available to the user. This **transaction congestion** becomes a modeling constraint, limiting the usefulness of simulation.

The method of tagging, which involves an alternative method of data collection, is analyzed as a possible solution to the problem of transaction congestion. Our research investigates the impact of tagging on information loss, memory savings, and simulation cost. It also examines the feasibility of implementing the tagging technique in the SLAM simulation language.

Proceedings of the 1982
Winter Simulation Conference
Highland * Chao * Madrigal, Editors

82CH1844-0/82/0000-0241 $00.75 © 1982 IEEE

2. Proposed Solutions

Several methods of dealing with the congestion problem have previously been suggested. Some are being used, but those studied thus far have identifiable problems. Examples of approaches taken to run large scale simulation models are described below:

- Use a larger and/or faster computer. This alternative may not be available, or may only be available at large expense.

- Break up the simulation model into logical segments which are run simultaneously. Off-line storage is used to pass transactions from one segment to another. This approach is very complex, and would require considerable run time to frequently access off-line storage.

- Consolidate groups of transactions into one transaction which will represent the group as it flows through the model. This approach has limited applicability where transactions cannot reasonably be treated as groups. If the model does not behave the same with grouped transactions as it would with individual transactions, validity will be a problem.

- Break up a large model into smaller segments and run the parts of the model sequentially. The output from one segment is used as the input to the next segment. To maintain validity, the model would have to be broken at points where there is no feedback between the segments. This approach works, however it adds complexity and increases run time.

- Compress a set of attributes into a smaller space through use of a coding scheme. This requires a code for each possible combination of attribute values. The compressed attributes must be decoded and recoded whenever a change is made to an attribute. Also, a coding scheme is specific to a set of attributes. Although this approach saves space, it would most likely increase CPU time and complexity.

- Eliminate attributes from the simulation study. Desired information may be lost, and the scope of the study may be reduced.

- Run a small version of the large system. There are validity problems with this approach in extending results from the smaller system to the larger system, and in scaling the model.

- Use an alternative world view such as the event scheduling or continuous world view. In the event world view, a system is modeled in terms of the events occurring which change the state of the system. The event world view is typically faster and requires less storage than the process world view; however, it is more difficult to use [Law and Kelton, 1982]. In the continuous world view, the discrete entities flowing through the model are viewed as a continuous flow described by differential or difference equations [Pritsker and Pegden, 1979]. This eliminates the need for large amounts of main memory to store transactions, but the entities lose any individual identity.

- Use an unconventional simulation modeling approach. For example, instead of viewing the parts that are processed in a machine shop as entities, one could view the machines themselves as entities and the parts as a flow. This approach only applies to systems which have homogeneous flows such as identical parts [Henriksen, 1981].

- Increase the efficiency of the simulation language. Perhaps fast list processing algorithms can be made faster to reduce run times. This requires modification of the simulation language.

In general, these solution approaches add complexity to running a simulation and introduce questions of model validity.

3. Another Approach: Tagging

The objective is to evaluate a new suggestion [Schruben, et al., 1982]. The technique is called **tagging** and involves an alternate method of data collection. Instead of gathering data on every transaction which passes through the system, only certain "tagged" transactions will be monitored. The memory space which would have been devoted to the storage of data concerning the untagged transactions is saved. If this amount of space is substantial, it may allow certain simulation models previously constrained by memory to be run.

In particular, every ℓ^{th} transaction can be tagged, $\ell = 1,2,3,\ldots$; ℓ is the **lag** between tagged transactions. A traditional simulation study is done with a lag of 1. That is, every transaction is tagged and monitored for data collection. A lag of 2 means data is collected on every other transaction, etc. Tagged transactions will have a full set of attributes, while the untagged transactions will not.

The primary reason to use tagging is for the computer main memory savings it can produce. The amount of memory required to run the simulation will decrease with increasing lag, towards an asymptotic level corresponding to memory which must be devoted to storing the model itself, overhead for each transaction, and other mandatory uses. Given the requirements of the study, a user with a storage constraint can determine the lag which will allow the model to fit into the given amount of space.

The transactions which are untagged undergo the same experiences in the model that the tagged transactions do. That is, the logical flow is unaltered when tagging is in use. Thus the tagging technique does not introduce a model validity problem.

Clearly, if a lag of two were in use, a run of fixed length will produce half the data points it otherwise could have. At first glance it may seem that discarding half the information potentially available is the price of the associated memory savings. This is often not the case. Since simulation output data is typically correlated, the information contained in one data point is also partially contained in the surrounding points. The degree of this redundancy is measured by the correlation. The higher the correlation, the less information is lost when the lag increases and fewer data points are collected. High correlations are often associated with systems that are highly congested and memory savings due to tagging is greatest.

Experiments were performed (with run length held constant) to test information loss in a single server queue model at various lags and levels of traffic intensity, i.e. congestion [Ho, Silverman, Steele, 1982]. The experiments show that information is lost as lag is increased; however for queues run at high congestion tagging with reasonable lags will not result in significant loss of information.

One can recover information loss due to tagging by increasing the run length of the simulation. An experimental procedure is being developed to determine the necessary increase in run length in order for a simulation with a lag of $\ell>1$ to give the same amount of information as the simulation with a lag of one.

The cost of running a simulation which employs the tagging technique is of interest. A cost model of some detail is developed [Ho, et al., 1982]. This cost function is simplified to a function of CPU and storage cost which depends on the simulation run time and model size. When the resulting function is evaluated for various lags at moderate and high correlations, it shows either a strictly increasing cost curve as lag is increased or a curve with a minimum point at a lag value greater than one (see Figures 1 and 2). Higher correlations move the cost minimum further from lag one. Therefore, there may exist a potential cost savings due to tagging depending on the cost structure of a particular computer installation and the characteristics of the simulation model.

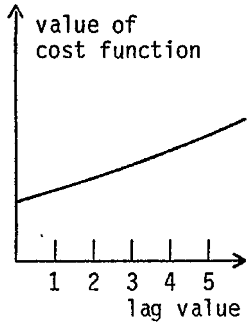

Figure 1

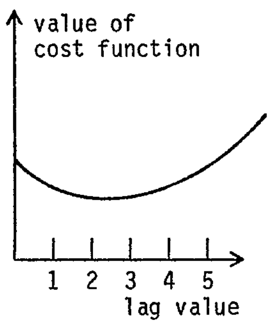

Figure 2

This tagging technique is not currently available in any simulation language. Implementation using standard features of SLAM is discussed in [Ho, et al., 1982].

The tagged transaction technique has potential to alleviate the entity congestion problem associated with large scale network simulation. It may allow modeling of systems previously impossible to simulate, and/or reduce the cost of running the existing models.

4. Future Issues

The work done so far has been to evaluate the basic concepts underlying the tagging approach. An attempt has been made to study whether tagging is a useful idea for dealing with the transaction congestion problem. Using simple systems, tagging's effect on a simulation's storage requirements, cost, and information was studied. Practical issues regarding using tagging were explored. There remains work to be done, however, in several areas.

First, more work needs to be done to identify what types of models are amenable to tagging. In complex systems, where should tagging be done?

Second, there are implementation questions to be answered. If tagging is to be implemented in a language, how should it be done? The possibility of a dynamic tag lag, meaning the program changes the lag as congestion in the system varies, needs to be investigated. Complications can arise if different lags are used in different parts of the model and if there are several attributes for data collection. If a transaction can be tagged, untagged, and retagged, then the instances of tagging must be totally independent of each other and in independent parts of the program. If these requirements are not met, how can tagging methods be implemented in different parts of a system? These concepts require further study.

Third, how can tagging be efficiently used to enable programs now being run in multiple segments (due to a storage constraint) to be run as a whole? Some work has been done in this area to develop a solution technique to find an optimal lag for each segment which when combined will fit into the available space. Implementation of two different lags in different parts of a system was done to get an idea of the implementation issues.

Finally, further research in applying tagging to a large, complex simulation should be done to get a better idea of any applied problems that tagging may have.

References

1. Henriksen, J.O. "GPSS Finding the Appropriate World-View," Proceeding of the Winter Simulation Conference. 1981, pp. 505-516.

2. Ho, J. Silverman, J.E., and Steele, N.F., 1982, Transaction Tagging in Highly Congested Network Simulations, M.Eng. Project Report, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York.

3. Law, Averill M. and Kelton, David W. Simulation Modeling and Analysis, New York: McGraw-Hill, 1982.

4. Pritsker, A.A.B. and Pegden, C.D., Introduction to Simulation and SLAM. New York: Holsted Press (Division of John Wiley and Sons, Inc.), 1979.

5. Schruben, Lee W., Maxwell, William L., and Cogliano, Vincent J. "Transaction Congestion in Large Scale Process Simulation Models." A Proposal for Research, School of Operations Research and Industrial Engineering, Cornell University, 1982.

Acknowledgement: The authors are grateful to the **Bethlehem Steel Corporation** for supporting this project through their university grant-in-aid program.