Performance and Scalability Results for an Aggressive Global Windowing Algorithm

Phillip M. Dickens
Paul F. Reynolds, Jr.
John M. Duva

Computer Science Technical Report No. TR-92-10
April 22, 1992

Performance and Scalability Results for an Aggressive Global Windowing Algorithm

Phillip M. Dickens
Paul F. Reynolds, Jr.

Department of Computer Science
University of Virginia
Charlottesville, Va. 22903
(pmd@koa.cs.virginia.edu)
(pfr@virginia.edu)

J.M. Duva

Department of Applied Mathematics
University of Virginia
Charlottesville, Va. 22903
(jmd@amsun17.apma.virginia.edu)

Abstract Windowing algorithms represent an important class of synchronization protocols for parallel discrete event simulation. In these algorithms, a simulation window is chosen such that all events within the window can be executed concurrently without the possibility of a causality error. Using the terminology of Chandy and Sherman (1989), these are *unconditional* events. Windowing algorithms, as all non-aggressive algorithms, have been criticized for not allowing a computation to proceed because there exists the *possibility* of a causality error. We are interested in the impact of extending the simulation window in order to allow the computation of *conditional* events, that is, those events that may cause an error. In this paper we develop a model to investigate the benefits of extending the simulation window to admit conditional events into the computation stream. Using this model we demonstrate significant performance gains as a result of aggressive processing. Also we prove that our approach is scalable: Performance is not significantly degraded *as the number of LPs approaches infinity*. We validate these results with empirical studies.

Performance and Scalability Results for an Aggressive Global Windowing Algorithm

Phillip M. Dickens
Paul F. Reynolds, Jr.

Department of Computer Science
University of Virgnia
Charlottesville, Va. 22903
(pmd@koa.cs.virginia.edu)
(pfr@virginia.edu)

Abstract Windowing algorithms represent an important class of synchronization protocols for parallel discrete event simulation. In these algorithms, a simulation window is chosen such that all events within the window can be executed concurrently without the possibility of a causality error. Using the terminology of Chandy and Sherman (1989), these are *unconditional* events. Windowing algorithms, as all non-aggressive algorithms, have been criticized for not allowing a computation to proceed because there exists the *possibility* of a causality error. We are interested in the impact of extending the simulation window in order to allow the computation of *conditional* events, that is, those events that may cause an error. In this paper we develop a model to investigate the benefits of extending the simulation window to admit conditional events into the computation stream. Using this model we demonstrate significant performance gains as a result of aggressive processing. Also we prove that our approach is scalable: Performance is not significantly degraded *as the number of LPs approaches infinity.* We validate these results with empirical studies.

1. Introduction

The sheer magnitude of many simulation problems makes the use of conventional sequential techniques impractical. For this reason researchers have turned to the use of multiprocessor architectures to provide the power necessary to simulate these large systems. In a parallel discrete event simulation (PDES) the physical system is partitioned into a set of physical processes (PP) and each PP is modeled by a corresponding logical process (LP). Logical processes communicate through the use of timestamped messages where each message represents a change to the state of the system being simulated. The timestamp of the message represents the time the system state is changed.

Parallel discrete event simulation poses very difficult synchronization problems due to the underlying sense of logical time. Each LP maintains its own logical clock representing the time up to which the corresponding PP has been simulated. The fundamental synchronization problem in PDES is in the determination of when it is permissible for an LP to advance its logical clock. If an LP advances its logical clock too far ahead of any other LP in the system it may receive a message with a timestamp in its logical past. When an LP receives a message with a timestamp in its logical past it is termed a *causality error* or *fault* and may lead to incorrect results.

Most of the protocols developed for parallel discrete event simulation fall into two basic categories. One category (using the terminology developed by Reynolds 1988) is protocols that are *accurate, non-aggressive* and *without risk* (also known as "conservative" protocols, e.g. Chandy and Misra 1979, Lubachevsky 1988, Nicol 1991 and Peacock, Manning and Wong 1978). The second category is protocols that are *accurate, aggressive* and *with risk,* also known as "optimistic", e.g. Time Warp (Jefferson 1985). Protocols that are non-aggressive and without risk do not allow an LP to process a message with timestamp t if it is possible that it will receive another message with a timestamp less than t at some point in the future. Protocols that are aggressive allow an LP to process any event it receives, and any causality errors that result from this aggressive processing are corrected through a *rollback* mechanism.

Non-aggressive protocols are criticized for not allowing a computation to proceed because there exits the *possibility* of a causality error. Thus computations that can possibly result in an error, but generally do not, will not be allowed to proceed. Aggressive protocols are criticized for the high overhead costs associated with state saving and rollback and because of the possibility of cascading rollbacks.

At least two researchers are investigating the benefits of adding aggressiveness to existing non-aggressive protocol (Lubachevsky 1989, Dickens and Reynolds 1990, Dickens and Reynolds 1991). This investigation seeks to maximize the benefits and minimize the costs of both approaches. A very important class of non-aggressive protocols to consider for this new approach is the *synchronous windowing algorithms*. These algorithms are important because they are the only class of protocols for which important scalability properties have been theoretically demonstrated (Lubachevsky 1989a, Nicol 1991).

In a previous paper (Dickens and Reynolds 1991) we describe our basic approach of adding aggressiveness to synchronous windowing algorithms. Also we develop a preliminary analytic model to study its performance. There are two problems with the model as developed. First, the model gives excellent predictions for the behavior of a single LP but provides only a loose upper bound on system performance. Second, we make some restrictive assumptions. In a queueing system we assume each LP has infinite servers. Also, our model cannot handle limited aggressive processing and assumes an LP processes all of its messages as soon as they are received.

The model presented in this paper represents a significant improvement over our earlier model. We have relaxed our restrictive assumptions and now allow queueing and limited aggressive processing. Further, we demonstrate analytically and empirically significant performance gains realized by our approach. Also we prove the scalability of our approach: performance does not degrade *as the number of LPs in the system approaches infinity*.

The rest of the paper is organized as follows. In section 2 we give a brief description of non-aggressive Synchronous Windowing Algorithms and discuss the modifications required to make them aggressive. In section 3 we develop our model to evaluate the performance of our approach. In section four we prove the scalability of our approach and in section 5 we give some empirical results to validate our model. In section 6 we give our conclusions and directions for future research.

2. Aggressive Synchronous Windowing Algorithms

The windowing protocols under discussion generally proceed in three phases. In the first phase, the LPs cooperatively determine the synchronization window. The floor of the window is the minimum timestamp over all unprocessed messages in the system. The ceiling of the window is chosen such that all messages within the window can be executed concurrently without any possibility of a causality error. We term this simulation window defined by the protocol the *Lookahead* window. In the second phase, each LP executes all of its messages with timestamps falling within the Lookahead window. In the third phase, the events generated as a result of execution within the Lookahead window are exchanged. Each phase is separated by a barrier synchronization. The primary difference among the various windowing protocols is the mechanism used to determine the Lookahead window. Note that only those messages with timestamps falling within the Lookahead window (where there is no possibility of a causality error) are considered for execution.

Our modified algorithm extends the simulation window past the Lookahead window established by the non-aggressive protocol. Assume the system is synchronized at logical time T where T is the current window floor. The non-aggressive windowing algorithm defines the Lookahead window from logical time T to logical time T+L, where L is the length of the Lookahead window. Our modified algorithm defines an extended simulation window from the upper bound of the Lookahead window (logical time T+L) to logical time T+L+A. We term this extension to the Lookahead window the *aggressive* window and note that it has a length of A logical time units. In our aggressive algorithm we allow messages with timestamps falling within the aggressive window to be processed as well as those messages with timestamps falling within the Lookahead window. This is shown in Figure 1.

As noted all processing within the Lookahead window is guaranteed to be correct. Processing within the aggressive window (we term this *aggressive* processing) may lead to causality errors. If errors do occur as a result of aggressive processing there must be some mechanism to correct the errors. Aggressive processing that does not result in a causality error is termed *successful aggressive processing*.

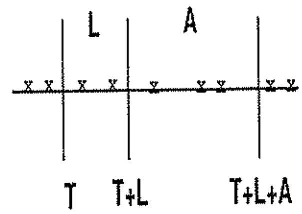


Figure 1.

3. Performance Measurements

We are interested in the improvement in performance as a result of extending the Lookahead window to allow aggressive processing. We define the performance of the non-aggressive protocol as the expected number of messages processed within the Lookahead window. We define the performance of our aggressive protocol as the expected number of messages *successfully* processed in the aggressive window plus the expected number of messages processed in the Lookahead. The expected improvement in performance is the ratio of these two quantities. We are interested in the expected improvement as a function of the aggressive window size.

Another measurement of interest is the probability of a causality error as we begin to process outside of the Lookahead window. We are interested in this error probability as a function of the size of the aggressive window.

4. Model

Our model is closely related to the one developed by Nicol (1991) although he is not investigating processing outside of the Lookahead window. Also his model is more general than the one presented here in that his captures general timestamp increment distributions. Our model is also closely related to the models developed by Akyildiz *et. al.* (1992) and Gupta *et. al.* (1991), and uses the same assumptions. Akyildiz and Gupta however are investigating the behavior of Time Warp which does not limit aggressive processing as we are proposing.

We model our protocol as collection of servers where *activities* occur. There are N LPs, one server per LP and one LP per processor. The assumption of one LP per processor can be modified without difficulty. An activity begins, ends and upon its completion causes other activities. In our model each completion causes exactly one other activity. We assume the completion causes an activity at a server that is picked at random where each server is equally likely to be chosen. Note that this equally likely assumption is a common assumption in the parallel simulation community and is used to make the analysis tractable (Felderman and Kleinrock 1992, Gupta *et. al.* 1991, Nicol 1991, Akyildiz *et. al.* 1991).

The delay in simulation time between when an activity begins and ends is called the *duration* of the activity. We assume each server chooses the duration of an activity from independent, identically distributed exponential distribution with mean $\frac{1}{\lambda}$. Note that our assumption of one server per LP implies that each LP will impose some queueing discipline.

The model presented in this paper assumes a closed queueing system. Also we assume the system is heavily loaded such that the probability of a server being idle is very close to zero. In another paper we relax the assumption of a closed system and model an open system with external Poisson arrival streams.

The width of the Lookahead window (L) at a given iteration of the protocol is a random variable. We state without explanation that in Nicol's protocol L is the minimum of N gamma distributed random variables, where each gamma is the sum of two independent, identically distributed exponential random variables with mean $\frac{1}{\lambda}$. In the analysis to follow we treat L as a constant rather than as a random variable. In particular, we use the expected value of L in our analysis. This expected value can be obtained analytically or through sample simulation runs. Below we demonstrate analytically why using the expected value of L in our equations is very reasonable. Simulation studies validate that using the expected value of L in our model provides excellent results.

As discussed above, there are two primary results in which we are interested. First, we seek the probability of a fault as a function of the aggressive window size. Second, we seek the expected increase in performance due to aggressive processing as a function of the size of the aggressive window.

In order to obtain the results of interest there are several steps that need to be accomplished. First, we need to determine the distribution for the number of messages in the aggressive window at the synchronization point (logical time T). This is the number of messages that can be processed aggressively.

Second, we need to determine the distribution for the number of messages in the Lookahead window at logical time T. This determines the number of messages processed in the non-aggressive version of the algorithm. Third, we need the distribution for the number of messages that subsequently arrive in the aggressive window. These messages are important because they can potentially cause a fault. Fourth, we need the distribution for the timestamps of the messages that are in the aggressive window at logical time T. Finally, we need the distribution for the timestamps of messages that subsequently arrive in the aggressive window. The last two items together determine the probability that an arrival message invalidates a message processed aggressively. In the following sections we determine the distributions for each of these items.

4.1. Number of Messages in Aggressive Window at Logical Time T

At this point we briefly describe one aspect of Nicol's windowing algorithm that is critical to our analysis. In Nicol's algorithm, each LP "pre-sends" its completion messages. That is, the completion time of an activity, and the LP to receive the activity upon its completion, are both calculated at the time the activity begins. The LP to subsequently receive the activity is notified of this reception *at the time the activity begins*. Note that this assumes powerful lookahead capabilities. In particular, it assumes non-preemptive service and that the routing of the activity is unaffected by the load on the network at service completion.

For our analysis, the important aspect of this pre-sending of completion messages is that there will be one "pre-sent" message for every LP that is busy. Further, if the system is synchronized at some time logical time T, each such "pre-sent" message will have a timestamp of at least T. This is because Nicol's algorithm (as all non-aggressive algorithms) guarantees that once the system reaches time logical time T, there are no messages in the system with timestamp less than T.

Our first task in determining the distribution for the number of messages in the aggressive window at logical time T is to determine the probability that a particular LP receives K "pre-sent" completion messages. Recall the assumption that the probability of an idle server is very low. This implies that with high probability each LP will have "pre-sent" a completion message to some LP in the system. This pre-sent completion message represents the completion time of the activity currently receiving service. Further note that the probability of a given LP receiving K of these "pre-sent" completion messages is binomially distributed due to the assumption of equally likely routing.

$$P(K) = (1/N)^K \left(\frac{(N-1)}{N}\right)^{N-K} \frac{N!}{K!(N-K)!} \tag{1}$$

In Equation (1) there is a large number of independent trials (N) and the probability of success (P) at any trial is small (1/N). Let $\lambda_1 = NP = 1$, the probability of success at any trial times the number of trials. As shown by Breiman (1986), the Binomial distribution is approximated very closely by the Poisson distribution (with rate $\lambda_1 = NP$) in the case where P is small and N is large. This is the case in Equation (1) and we conclude that the probability of a given LP receiving K pre-sent completion messages is closely approximated by the Poisson distribution with rate λ=1.

$$P(K) = \frac{e^{-1}}{K!} \tag{2}$$

Now consider the distribution of the timestamp for a "pre-sent" completion message received by a given LP at logical time T. It is known that the service time distribution for such a message is exponentially distributed, and that all such messages will have timestamps greater than T. Due to the memoryless property of the exponential distribution, the residual service time of the activity is also exponentially distributed. Thus the distribution of the timestamp of a "pre-sent" completion message, given that the timestamp is greater than logical time T, is also exponentially distributed. Thus we can view the system as probabilistically restarting at logical time T. For this reason in the equations that follow we define the aggressive window as falling between logical times L and L+A rather than as falling between logical times T+L and T+L+A.

Given the distribution of the timestamps for pre-sent completion messages, and the distribution for the number of such messages received, we can compute the distribution for the number of messages with timestamps falling within the aggressive window (L,L+A). Consider the probability that exactly one of the K "pre-sent" completion messages received by an LP falls within the aggressive window. In order for this to occur, exactly one of the K messages received by the LP must have a timestamp within the aggressive window, and all of the other K-1 messages must have timestamps outside of the aggressive window.

Given that we know the timestamp distribution for "pre-sent" completion messages we can calculate the probability that such a message falls within the aggressive window.

$$P(t \text{ in } AW) = e^{-\lambda L} - e^{-\lambda(L+A)} \tag{3}$$

The probability of a timestamp being outside of the aggressive window is one minus the proability that it falls within the window.

$$P(t \text{ not in } AW) = 1 - (e^{-\lambda L} - e^{-\lambda(L+A)}) \tag{4}$$

The probability that i (independent) messages fall within the aggressive window given that K messages are received is:

$$P(i \mid K) = (e^{-\lambda L} - e^{-\lambda(L+A)})^i \, (1 - e^{-\lambda L} + e^{-\lambda(L+A)})^{K-i} \frac{K!}{i!(K-i)!}. \tag{5}$$

In Equation (5), the $(e^{-\lambda L} - e^{-\lambda(L+A)})^i$ term is the probability of i independent messages falling within the aggressive window. The $(1 - e^{-\lambda L} + e^{-\lambda(L+A)})^{K-i}$ term is the probability that the other $K-i$ messages fall outside of the aggressive window. The $\frac{K!}{i!(K-i)!}$ term is the number of combinations of i messages out of K total messages that can fall within the aggressive window.

Equation (5) gives the probability of i messages falling within the aggressive window given that K messages are received. In order to uncondition, we need to sum over all possible values of K times the probability of K:

$$P(i) = \sum_{K=i}^{N} \frac{e^{-1}}{K!} (e^{-\lambda L} - e^{-\lambda(L+A)})^i \, (1 - e^{-\lambda L} + e^{-\lambda(L+A)})^{K-i} \frac{K!}{i!(K-i)!}. \tag{6}$$

We can rewrite Equation (6) in the following way.

$$P(i) = \frac{e^{-1}}{i!} (e^{-\lambda L} - e^{-\lambda(L+A)})^i \sum_{K=i}^{N} \frac{(1 - e^{-\lambda L} + e^{-\lambda(L+A)})^{K-i}}{(K-i)!} \tag{7}$$

Let $j = K-i$ and rewrite the summation.

$$\sum_{j=0}^{N-i} \frac{(1 - e^{-\lambda L} + e^{-\lambda(L+A)})^j}{(j)!} \tag{8}$$

Recall the following identity.

$$\sum_{j=0}^{\infty} \frac{x^j}{j!} = e^x$$

Thus as N goes to infinity (and $i \ll N$), Equation (8) becomes

$$e^{(1-e^{-\lambda L}+e^{-\lambda(L+A)})}. \tag{9}$$

We use this result as an approximation.

The approximation of Equation (8) given in Equation (9) is very good for reasonable size N. To see this note that $\sum_{j=0}^{\infty} \frac{x^j}{j!}$ converges to e^x very quickly when $x<1$. This is because the numerator gets small and the denominator gets large very quickly as j increases. Thus the contribution to the summation gets small very quickly as j increases. In Equation (8) $1-e^{-\lambda L}+e^{-\lambda(L+A)}$ is less than one as it represents the probability of an exponential random variable being outside of a given range.

Rewrite Equation (7) using the approximation given in Equation(9).

$$P(i) = \frac{(e^{-\lambda L}-e^{-\lambda(L+A)})^i}{i!} \quad e^{-(e^{-\lambda L}-e^{-\lambda(L+A)})} \tag{10}$$

We conclude that a very reasonable approximation of the probability distribution for the number of messages in the aggressive window at time T is the Poisson distribution with parameter $e^{-\lambda L}-e^{-\lambda(L+A)}$.

If we let $L=0$ and $A=L$ in Equation (10) we see that the number of messages in the Lookahead window at time T is also Poisson distributed with parameter $1-e^{-\lambda L}$.

$$P(K) = \frac{(1-e^{-\lambda L})^K}{K!} \quad e^{-(1-e^{-\lambda L})} \tag{11}$$

4.1.1. Completion Message

Before leaving this section we need to say another word about the probability of processing K messages in the aggressive window. In the sections above we derived the probability distribution for the number of "pre-sent" completion messages in the aggressive window at time T. In addition to presending the completion message to the receiving LP, an LP also schedules a completion message on its own event list. The processing of this completion message consists of giving service to the next scheduled activity as well as any statistics gathering required for the simulation. Thus for every server that is busy there is one "pre-sent" completion message and one completion message scheduled on the server's event list. We term this second type of completion message the "complete_service" message.

Consider the "complete_service" message. If this message falls within the aggressive window it affects the number of messages processed aggressively. If it falls within the Lookahead window it affects the number of unconditional messages processed. Due to the relatively small number of messages processed per window it is important to consider the effects of the "complete_service" message.

Recall our assumption that the probability of an idle server is very low. Therefore it is with high probability that each LP will always have a "complete_service" message on its event list. Our analysis assumes this will always be the case. This assumption can be modified to reflect the probability of a "complete_service" message (not equal to one) without difficulty.

The "complete_service" message will have a distance from the synchronization point T that is exponentially distributed. Thus the probability that the "complete_service" message falls within the Lookahead window is $1-e^{-\lambda L}$ and the probability that it falls within the aggressive window is $e^{-\lambda L}-e^{-\lambda(L+A)}$. Given this, we can recompute the probability of K messages within the aggressive window at the synchronization point.

In order to have K=0 messages in the aggressive window the LP must have no "pre-sent" completion messages *and* the "complete_service" message must fall outside of the aggressive window. We give this probability below.

$$P(K=0) = P(M=0,\overline{C}) \quad = e^{-(e^{-\lambda L}-e^{-\lambda(L+A)})}\,(1-e^{-\lambda L}+e^{-\lambda(L+A)}). \tag{12}$$

In Equation (12) the $P(M=0)$ term is the probability of having zero "pre-sent" completion messages in the aggressive window (this probability is given in Equation (10)). The $\overline{C}$ term is the probability of the "complete_service" message falling outside of the aggressive window.

Now consider the probability of $K \geq 1$ message within the aggressive window. One way this can occur is for the LP to have K "pre-sent" completion messages and not have the "complete_service" message fall within the aggressive window. Alternatively, $K-1$ "pre-sent" completion messages *and* the "complete_service" message may fall within the aggressive window. This probability is given below.

$$P(K,K>0) = P(M=K-1,C) + P(M=K,\overline{C}) = \tag{13}$$

$$\frac{(e^{-\lambda L}-e^{-\lambda(L+A)})^{K-1}}{K-1!}\; e^{-(e^{-\lambda L}-e^{-\lambda(L+A)})}\; e^{-\lambda L}-e^{-\lambda(L+A)}$$

$$+\frac{(e^{-\lambda L}-e^{-\lambda(L+A)})^{K}}{K!}\; e^{-(e^{-\lambda L}-e^{-\lambda(L+A)})}(1-e^{-\lambda L}-e^{-\lambda(L+A)}).$$

In Equation (13) M is the number of "pre-sent" completion messages within the aggressive window and C

is the probability that the "complete_service" message falls within the aggressive window.

Similar arguments show that the probability of $K=0$ messages in the Lookahead window is:

$$P(K=0) = e^{-(1-e^{-\lambda L})}\,(1-e^{-\lambda L}). \tag{14}$$

The probability of $K \geq 1$ messages in the Lookahead window is:

$$P(K, K \geq 1) = \frac{(1-e^{-\lambda L})^K}{K!}\, e^{-(1-e^{-\lambda L})}\, e^{-\lambda L} + \frac{(1-e^{-\lambda L})^{K-1}}{K-1!}\, e^{-(1-e^{-\lambda L})}\,(1-e^{-\lambda L}). \tag{15}$$

4.2. Arrival Distribution

Before determining the arrival distribution we briefly review the modified windowing protocol. In the first phase of the protocol the LPs cooperatively determine the simulation window. The floor of the window (the synchronization point) is the minimum unprocessed timestamp in the system (we have been denoting this timestamp logical time T). The ceiling of the simulation window is determined such that all messages within the window can be processed concurrently without possibility of an error. We term this window the Lookahead window and note that it has a width of L logical units. In our modified algorithm we extend the simulation window from logical time T+L to T+L+A. We term the extended simulation window the aggressive window and note that it has a width of A logical units. In the second phase of the protocol all messages with timestamps falling within the Lookahead and the aggressive window are processed. In the third phase, all messages generated as a result of the processing in the second phase are exchanged.

In the non-aggressive algorithm it is guaranteed that no messages with timestamps falling between T and T+L will be received as a result of processing within the Lookahead window. That is, no message processed within the Lookahead window will generate (or cause) a message with a timestamp that also falls within the Lookahead window. It is quite possible however that messages processed within the Lookahead window will generate messages with timestamps that fall within the aggressive window. It is also possible that messages processed within the aggressive window will generate messages with timestamps that again fall within the aggressive window. In either case, messages that arrive in the aggressive window as a result of processing in the second phase of the algorithm are termed *arrival* messages. It is also possible to have arrival message that are generated in subsequent iterations of the algorithm. This is dependent upon the size of aggressive window. We now determine the distribution for the number of arrival messages.

The duration of an activity at a server is exponentially distributed with mean $\frac{1}{\lambda}$. Due to our assumption of a heavily loaded system we know that the output rate of activities at a given server is Poisson distributed with rate λt. Due to the independence of the N servers in the system, the system output will be the merging of N independent Poisson streams. Thus the system output rate will be Poisson distributed with rate $N\lambda t$. As each LP is equally likely to receive an activity upon completion the system output stream forks into N independent Poisson streams. The input rate to any given LP is therefore Poisson with rate $\frac{N\lambda t}{N} = \lambda t$.

We now have enough information to determine the probability distribution for the number of arrival messages. As discussed above, the total input rate into a given LP is Poisson with rate λt. The aggressive window has a width of A logical time units and its total input rate will therefore be Poisson distributed with rate λA. We break this total input rate into two component rates. First, the rate for the number of messages in the aggressive window at the synchronization point (logical time T), and second, the rate of the arrival messages. In Equation (10) we derived the rate for the Poisson distribution governing the number of messages in the aggressive window at the synchronization point $(e^{-\lambda L} - e^{-\lambda(L+A)})$. We term the rate for the Poisson distribution governing the number of arrival messages $\lambda_{arrival}$. Below we show the relationship between these component rates.

$$\lambda A = \lambda_{Arrival} + (e^{-\lambda L} - e^{-\lambda(L+A)}) \tag{16}$$

We now solve for $\lambda_{Arrival}$.

$$\lambda_{Arrival} = \lambda A - [e^{-\lambda L} - e^{-\lambda(L+A)}] \tag{17}$$

We conclude that the distribution for the number of arrival messages is Poisson with rate $\lambda A - (e^{-\lambda L} - e^{-\lambda(L+A)})$.

$$P(K) = \frac{(\lambda A - (e^{-\lambda L} - e^{-\lambda(L+A)}))^K}{K!} \; e^{-(\lambda A - (e^{-\lambda L} - e^{-\lambda(L+A)}))} \tag{18}$$

4.3. Distribution of Timestamps

We seek the distribution for the timestamps of the "pre-sent" completion messages that are within the aggressive window at logical time T. Recall that these messages represent the completion times of activities currently receiving service at some LP. Further recall that all service times are iid exponential random variables with mean $\frac{1}{\lambda}$. Due to the memoryless property of the exponential the residual service time will also be exponential. Thus the timestamps of the "pre-sent" completion messages falling within the aggressive window will be exponentially distributed, but conditioned on the fact that they are within the aggressive window. This distribution is given below.

$$pdf\,(x \mid L \le x \le L+A) = \frac{\lambda e^{-\lambda x}}{e^{-\lambda L} - e^{-\lambda(L+A)}} \tag{19}$$

Equation (19) is obtained from the definition of conditional probability. The numerator is the exponential density function, and the denominator is the probability that an exponentially distributed random variable falls within the aggressive window (given that it is greater than the synchronization point T).

The distribution for timestamps of arrival messages is much more difficult to determine. Recall that arrival messages may be caused by the processing of a message within the Lookahead window or the processing of a message within the aggressive window. An arrival message caused by the processing of a message within the aggressive window is termed *a second generation message*. Clearly there can be higher order generation messages as well. The probability of a second (or higher order) generation is dependent upon the size of the aggressive window. It is very difficult to determine the timestamp distribution of arrival messages since it is not known whether these messages are a result of processing within the Lookahead window or the aggressive window.

We can however constrain the aggressive window size such that the probability of a second or higher order generation message is small. Given a particular aggressive window size A, we can determine the probability that a message received within A is a second generation message. We can then restrict A such that the probability of a second generation message is ε, for some predetermined ε. This is discussed further below.

Given an aggressive window size A such that there is a low probability that an arrival message is a second generation message we can determine the timestamp distribution for arrival messages. Without considering second (or higher) generation messages, the arrival messages will be those caused by processing within the Lookahead window. By the definition of the protocol we know that no message processed within the Lookahead window will generate a message with a timestamp that again falls within the Lookahead window. Thus all such messages will have timestamps greater than or equal to L. Also we know these timestamps are exponentially distributed. Given this we again have the case where these timestamps are exponentially distributed and greater than L. The messages that fall within the aggressive window will therefore also be exponential, conditioned on the fact that they land within the aggressive window. We conclude that the distribution for arrival messages, given that we restrict the the size of the aggressive window to exclude second or higher order generation messages, is the same distribution as given in Equation (19).

Since we cannot strictly guarantee no second generation messages (given an aggressive window size greater than zero), we note that Equation (19) is an approximation for the timestamp distribution of arrival messages. The larger the aggressive window the higher the probability of a second generation message and the worse the approximation becomes. For this reason it seems unreasonable to consider aggressive window sizes that are greater than $\frac{1}{\lambda}$, the mean of the service distribution. For the remainder of this analysis we assume $0 \le A \le \frac{1}{\lambda}$.

4.3.1. Error of Timestamp Distribution Approximation

Given our assumption of no second generation messages we would like to quantify the error of this assumption as a function of the aggressive window size. That is, we would like to determine the probability of a second generation message as a function of the aggressive window size.

We seek the probability that a message generated as a result of the processing of a message within the aggressive window will have a timestamp that again falls within the aggressive window. Recall that the processing of a message consists of adding an exponential increment to the current timestamp of the message. Given a message with timestamp S=s within the aggressive window, we seek the probability that an exponential increment from s is less than A. For a particular S=s this probability is shown below.

$$P(exponential\ increment < A \mid S=s) = 1-e^{-\lambda(A-s)}$$

In order to uncondition we integrate this probability over all possible values of S=s multiplied by the probability of S=s. As shown in Equation (15), the distribution for S is exponential conditioned on being within the aggressive window.

$$P(2nd\ generation) = \int_0^A (1-e^{-\lambda(A-s)}) \frac{\lambda e^{-\lambda s}}{(1-e^{-\lambda A})} ds = \frac{1-(\lambda A e^{-\lambda A} + e^{-\lambda A})}{1-e^{-\lambda A}} \qquad (20)$$

As shown in Equation (20), the probability of a second generation message is relatively small for reasonable size A. For example, with $\lambda = 1$ and $A = .1$ the probability of a second generation message is less than 5%.

Given some probability that a second generation message does occur we need to determine the way in which this affects our analysis. Recall that aggressive messages are exponentially distributed within the aggressive window. This implies that aggressive messages are weighted towards the front of the aggressive window. As shown above, arrival messages caused by processing within the Lookahead window have the same timestamp distribution as the aggressive messages. Thus they will also be weighted towards the front of the aggressive window. Second generation messages will clearly be weighted more towards the back of the aggressive window than first generation messages. For this reason the probability of a second generation message causing a fault will be somewhat less than the probability of a first generation message causing a fault. Since our model does not differentiate between first and second generation arrival messages it will over predict the probability of a fault if there a significant probability of second generation messages.

4.4. Using the Expected Value for L

We have now obtained all of the distributions necessary to determine the probability of a fault and the expected improvement due to aggressive processing. Note that all of our equations have treated L, the size of the Lookahead window, as a constant. Recall that L is the minimum of N gamma distributed random variables, where each gamma is the sum of two exponentially distributed random variables. Thus the value of L is a random variable rather than a constant. In this section we show that it is very reasonable to treat L as a constant, and to use the expected value of L in our equations, when $\lambda L \ll 1$. For now we note that $\lambda L << 1$ for $N \geq 25$. We prove this in the following sections.

Recall the following identity.

$$e^{-\lambda L} = 1 - \lambda L + \frac{\lambda L^2}{2!} - \frac{\lambda L^3}{3!} \ldots$$

When $\lambda L << 1$ a good approximation for $e^{-\lambda L}$ is

$$e^{-\lambda L} \approx 1 - \lambda L.$$

The most common term in our equations involving L is $e^{-\lambda L} - e^{-\lambda(L+A)}$. In this section we show why it is reasonable to use the expected value of L in this particular expression. Similar arguments can be made for other expressions involving L.

Rewrite

$$e^{-\lambda L} - e^{-\lambda(L+A)} = e^{-\lambda L}\ (1 - e^{-\lambda A}).$$

Substituting our approximation for $e^{-\lambda L}$ we have

$$\begin{aligned} e^{-\lambda L} - e^{-\lambda(L+A)} &\approx (1-\lambda L)(1-e^{-\lambda A}) \\ &= (1-e^{-\lambda A}) - \lambda(1-e^{-\lambda A})L. \end{aligned}$$

Note that for fixed A this is a linear function of L.

$$(1-e^{-\lambda A}) - \lambda(1-e^{-\lambda A})L = b + aL$$

Now take the expected value of the expression.

$$\begin{aligned} E[(1-e^{-\lambda A}) - \lambda(1-e^{-\lambda A})L] &= E[b + aL] \\ &= b + a\,E[L] \\ &= (1-e^{-\lambda A}) - \lambda(1-e^{-\lambda A})E[L] \\ &= (1-e^{-\lambda A})(1 - \lambda\,E[L]). \end{aligned}$$

Again using our approximation that $e^{-x} = 1 - x$ for small x we rewrite $1 - \lambda\,E[L]$ as $e^{-\lambda E[L]}$ (again noting that $E[L] \ll 1$).

$$(1-\lambda\,E[L])\ (1-e^{-\lambda A}) \approx e^{-\lambda E[L]}\ (1-e^{-\lambda A}) = e^{-\lambda E[L]} - e^{-\lambda(E[L]+A)}$$

This shows that using the expected value of L in our equations is very reasonable when $\lambda L \ll 1$.

5. Probability of a Fault

In order to fault an LP must process at least one message aggressively and subsequently receive an arrival message with a timestamp in its past. There are many ways this can occur. For example, an LP can process one aggressive message and receive one arrival message with a timestamp in its past. Or an LP can process one aggressive message and receive two arrival messages where one or both messages have timestamps in its past. Theoretically there are an infinite number of ways a fault can occur. In practice however only a few such combinations have any significant associated probability.

Before enumerating the significant terms of the probability of a fault we note that one message in the system needs special consideration. We state without elaboration that as a consequence of processing in the Lookahead window exactly one LP will receive a pre-sent completion message with a timestamp of L. We term this message the *Lookahead message*. Recall that L is the floor of the aggressive window. For this reason the LP that receives the Lookahead message will fault *if it has processed any messages aggressively*.

Due to the assumption of equally likely routing of messages, the probability of a particular LP receiving the Lookahead message is $\frac{1}{N}$. The probability of any LP processing at least one aggressive message is 1 - P(K=0), where P(K=0) is given in Equation (10). Thus the probability of a given LP faulting due to the Lookahead message is $\frac{1}{N}\,(1-P\,(K=0))$.

Below we enumerate the significant terms of the probability of a fault. In this equation P(K) is the probability of processing K aggressive messages and is given in Equation (10). The P(Ar) term is the probability of receiving Ar arrival messages and is given in Equation (18).

$$P\,(F) = 1/N(1-P\,(K=0)) + P\,(K=1,Ar=1)\ .5 + P\,(K=2,Ar=1)\ .75 + \tag{21}$$
$$P\,(K=3,Ar=1)\ (1-.5^3) + P\,(K=1,Ar=2)\ .75 +$$
$$P\,(K=1,Ar=3)\ (1-.5^{3)} + P\,(K=2,Ar=2)\ .25^2 \ldots\ldots\ldots\ldots$$

In Equation (21), the second term is the probability of processing one message aggressively (P(K=1)) times the probability of one arrival message (P(Ar=1)) times the probability that the timestamp of the arrival message is less than the timestamp of the aggressive message (.5). Note that this assumes both messages have the same timestamp distribution as discussed above. The other terms are similarly derived.

6. Number of Messages Successfully Completed

The second primary goal of this research is to capture analytically the improvement in performance made possible by aggressive processing. We measure performance gains by comparing the number of messages processed in the non-aggressive algorithm versus the number of messages processed successfully by adding aggressiveness. Recall that the number of messages processed in the non-aggressive algorithm is the number of messages processed in the Lookahead window. The number of messages processed in the aggressive algorithm is the number of messages processed in the Lookahead window plus the number of messages processed successfully in the aggressive window. By processed successfully we mean processing that is not later found to be invalid. The improvement in performance is the ratio of these two quantities. We begin by computing the expected number of messages processed successfully in the aggressive window.

Note that the improvement in performance calculated as described above is an upper bound on the improvement that can be attained in practice. This is because our model does not account for the correction of any errors that do occur. It may be the case that some valid processing is reprocessed in order to correct errors. Our model gives the expected improvement in performance given that all successful aggressive processing is maintained. It therefore should be viewed as an upper bound on performance improvement. The given correction mechanism will determine how much of this potential improvement is realized.

Given the distributions for the number of messages in the aggressive window at the synchronization point and the number of arrival messages into the aggressive window it is possible to determine the number of aggressive messages processed successfully. Consider the ways in which an LP can process one aggressive message successfully. First, the LP can process one aggressive message and subsequently receive no arrival messages. In this case the aggressive processing cannot be invalidated. Second, the LP can process one aggressive message and receive one arrival message that does not invalidate the processing. Third, the LP can process two aggressive messages and receive one arrival message that invalidates one, but not both, aggressive messages. Clearly there are infinite combinations of events that could be considered. There are only a few such combinations however with any significant associated probability.

Below we give the significant terms for the probability of successfully processing M=1 messages. In this equation P(K) is the probability of processing K aggressive messages. The P(Ar) term is the probability of receiving Ar arrival messages.

$$P(M{=}1) = P(K{=}1,Ar{=}0) + P(K{=}1,Ar{=}1)\ .5 + P(K{=}1,Ar{=}2)\ .25 \tag{22}$$
$$+P(K{=}1,Ar{=}3)\ .5^3 + P(K{=}2,Ar{=}1)\ .5 + P(K{=}2,Ar{=}2)\ .375$$
$$+P(K{=}3,Ar{=}1)\ .375 + P(K{=}3,Ar{=}2)\ .5\ldots\ldots\ldots$$

The first term in Equation (22) is the probability of processing one aggressive message and receiving no arrival messages (P(K=1,Ar=0)). The second term is the probability of processing one aggressive message (K=1), receiving one arrival message (Ar=1), and the arrival message not causing a fault (.5). Note that the probability of the arrival message not causing a fault is the probability of the arrival message having a timestamp greater than the aggressive message. Since the timestamps of both messages have the same distribution this probability is .5.

Equation (23) gives the significant terms for processing two aggressive messages successfully.

$$P(M{=}2) = P(K{=}2,Ar{=}0) + P(K{=}2,Ar{=}1)\ .25 + P(K{=}2,Ar{=}2)\ .25^2 \tag{23}$$
$$+P(K{=}3,Ar{=}1)\ .375 + P(K{=}3,Ar{=}2)\ .04568$$

Finally we present the most important terms for the probability of processing three aggressive messages successfully. We note that the probability of processing four or more aggressive messages successfully is negligible for aggressive window sizes in the range of $0 \le A \le \frac{1}{\lambda}$.

$$P(M{=}3) = P(K{=}3,Ar{=}0) + P(K{=}3,Ar{=}1).5^3 + P(K{=}3,Ar{=}2).0156 \tag{24}$$
$$+P(K{=}4,Ar{=}1).25$$

We now have all of the equations necessary to derive the expected number of aggressive messages successfully processed. In Equation (25) below M is the number of messages processed successfully within the aggressive window.

$$E[M] = 1*P(M=1) + 2*P(M=2) + 3*P(M=3)\ldots\ldots\ldots\ldots \tag{25}$$

Similarly the expected number of messages processed in the Lookahead window is

$$E[LM] = 1*P(LM=1)+2*P(LM=2)+3*P(LM=3)\ldots\ldots.. \tag{26}$$

Recall that the probability of processing K messages in the Lookahead window (P(LM=K)) is given in Equation (11). The expected improvement in performance due to aggressive processing is therefore:

$$E[I] = \frac{E[LM]+E[M]}{E[LM]}. \tag{27}$$

We calculate this expected improvement in performance for various window sizes below.

7. Scalability of Protocol

A very important issue in parallel discrete event simulation is how well a particular protocol scales as the number of LPs increases. As discussed by Lubachevsky (1989) both Time Warp and the Null Message protocol have the potential for explosive overhead costs as the size of the simulation grows. The overhead associated with Time Warp can significantly increase due to large state saving costs and the possibility of cascading rollbacks. The Null Message protocol can have significant overhead costs due to the proliferation of null messages. To date only Nicol (1991) and Lubachevsky (1989) have proven the scalability of their respective protocols.

In order to study the scalability of our approach we examine the probability of a fault as the size of the simulation grows. In this section we show that the upper bound on the probability of a fault at a given LP is approximately 25% *as the number of LPs in the system approaches infinity*. Further, we show that the probability of a fault reaches this maximum value for some particular value of N (the number of LPs in the system). That is, there is some particular value of $N = N^*$ such that once the number of LPs in the system is greater then N^* the more LPs in the system the less the probability of a fault ($N^* < N_1 < N_2 \rightarrow P(F,N_1 > P(F,N_{2)})$. In this section we identify this value of N^*.

Note that our equations do not directly reflect N, the number of LPs in the system. Rather this is reflected in L, the expected value of the Lookahead window. Recall that L is the minimum completion time of the next job to receive service taken over all LPs in the system. Further recall that our model assumes one server per LP, and that the probability of a server being idle is very low. The completion time of the next job to receive service at a given LP will thus be the sum of a) the residual service time of the job currently receiving service and b) the service requirements of the next job to receive service. As all service times are iid exponentials, and because of the memoryless property of the exponential, the completion time of the next job to receive service will the sum of two exponentials. The Lookahead value for the system will therefore be the minimum of N gamma distributed random variables where N is the number of LPs in the system.

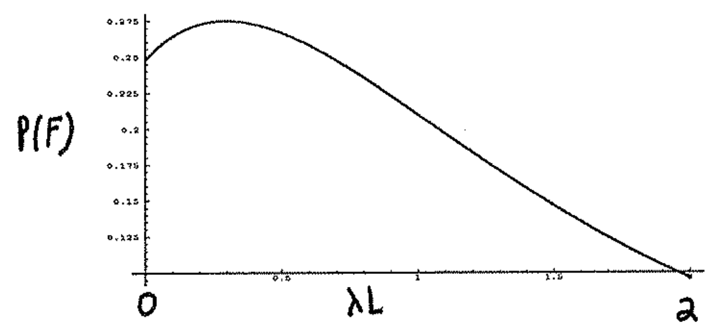
In order to fault an LP must process at least one aggressive message *and* receive at least one arrival message. Note that both of these events are a necessary, but not sufficient condition for a fault. Thus the probability of both events occurring gives an upper bound on the probability of a fault.

The probability of processing at least one aggressive event is $(1-P(K=0))$ where the P(K=0) is given in Equation (10). The probability of receiving at least one arrival event is $(1-P(Ar=0))$ where P(Ar=0) is given in Equation (14). Noting the independence of these events, the probability of both events occurring is given below.

$$P(F=K\geq 1, Ar\geq 1)=(1-[e^{(e^{-\lambda L}-e^{-\lambda(L+A)})}(1-(e^{-\lambda L}-e^{-\lambda(L+A)}))])\quad(1-e^{-(\lambda A-(e^{-\lambda L}-e^{-\lambda(L+A)})})\qquad(28)$$

In Figure 2 we plot Equation (28) as a function of λL. Note that A is set to $\frac{1}{\lambda}$, the maximum aggressive window size considered. Further note that λL can only take on values between 0 and 2. This is because the expected value for a gamma which is the sum of two exponentials is $\frac{2}{\lambda}$. Clearly the maximum value of L, the expected value of the minimum of N such gamma distributed random variables, cannot exceed $\frac{2}{\lambda}$. Also note that as $N\rightarrow\infty$ $\lambda L\rightarrow 0$.

As can be seen from Figure 2 the maximum probability of fault (approximately 27%) is reached when $\lambda L \approx 0.29$. It can also be seen that after this maximum value the probability of a fault decreases slightly as $\lambda L\rightarrow 0$ ($N\rightarrow\infty$). The probability of a fault when $\lambda L = 0$ is approximately 24.7%. Thus we have the very powerful result that the probability of a fault decreases *as the number of LPs in the system approaches infinity.*



It can also be seen from Figure 2 that the probability of a fault rises sharply for $.29 \le \lambda L \le 2$. It would be desirable to either a) reduce this steep rise in the probability of a fault b) show that the number of LPs (N) in the system for $.29 \le \lambda L \le 2$ is sufficiently small such that this steep rise is not critically important. We could reduce the slope of the fault curve by adjusting the size of the aggressive window. The approach we choose is to demonstrate that the number of LPs in the system for $\lambda L = .29$ (N^*) is sufficiently small such that $1 \le N \le N^*$ is not of critical importance.

We begin by noting it is clear that the expected value of the minimum of N gamma distributed random variable is a decreasing function of N. Without going through the derivation we note that the expected value of the minimum of N gamma distributed random variables (each with mean $\frac{2}{\lambda}$) is:

$$L = \int_0^{\infty} N\lambda^2 y^2 (1+\lambda y)^{(N-1)}\, e^{-\lambda N y} dy. \tag{29}$$

Let $\xi = \lambda y$ and $d\xi = \lambda dy$. Rewrite Equation (29) using this variable transformation.

$$L = \frac{1}{\lambda} \int_0^{\infty} N\xi^2 (1+\xi)^{(N-1)} e^{-N\xi} d\xi \tag{30}$$

Note that the integral is a function of N only. Rewrite Equation (30).

$$L = \frac{1}{\lambda} f(N) \tag{31}$$

Because L is a decreasing function of N, f(N) in Equation (31) is also a decreasing function of N.

As shown in Figure 2, $\lambda L \approx .29$ when the maximum probability of a fault is reached. Rewrite this as follows:

$$L \approx \frac{1}{\lambda} .29. \tag{32}$$

Thus we see that f(N) = .29 when the maximum probability of a fault is reached. Since f(N) is a decreasing function of N there is only one value of N, N^*, for which Equation (32) is true.

In order to find N^* we set $\lambda = 1$ and numerically determine the number of gamma distributed random variables (with mean 2) for which the expected value of the minimum of this number is approximately .29. We found this number to be between twenty three and twenty four. Then we ran many simulation studies to validate that the expected value of the minimum of twenty four gamma distributed random variables (each composed of the sum of two exponentials with mean 1) is approximately .29. We conclude that $N^* = 24$. This demonstrates the very important property that if there are more than twenty four LPs in the system that adding more LPs to the system will *decrease the probability of a fault.*

In this section we have investigated the *upper bound* on the probability of a fault given that the size of the aggressive window is set to its maximum value ($\frac{1}{\lambda}$). We used the upper bound of a fault, and the maximum aggressive window size, in order to to demonstrate the scalability of the approach under the worst possible conditions. It is important to note that the results obtained under these conditions are much higher than the results obtained using our explicit estimate of a fault given in Equation (21). In Table 1 we use Equation (21) to derive the maximum probability of a fault, the minimum probability of a fault (obtained when λL=0), and N^* for various aggressive window sizes.

8. Preliminary Empirical Results

We have developed a model to investigate the improvement in performance made possible by adding aggressiveness to an existing non-aggressive protocol. In order to test the validity of the model we ran a series of simulations using a simple FCFS queueing model. The queueing model met the basic

A	P(F) Max	P(F) Min	N*
.1	.002	.0005	5
.3	.02	.009	7
.5	.053	.035	10
.7	.09	.077	15

Table 1.

assumptions of our analysis including an exponential service time distribution, a heavily loaded system and each LP having an equal probability of receiving a given message. We ran a series of simulations with 1500 LPs, a mean service time of $\frac{1}{\lambda} = 1$ and various aggressive window sizes. Then we compared the empirical results with the predictions of our model. Note that the value of L used in our equations was the expected value of the width of the Lookahead window obtained from sample simulation runs.

In Figure 4 we plot the predicted versus observed probability of a fault for various aggressive window sizes. As can be seen, our equations give very accurate predictions. Our equations begin to slightly over predict the probability of a fault when the aggressive window size becomes greater than 40% of the mean service time. Recall that our model over predicts the probability of a fault if there is a significant probability of a second generation message. For aggressive window sizes greater than or equal to 40% of the mean service time the probability of a second generation message becomes large enough such that our model does slightly over predict the probability of a fault.

In Figure 5 we plot the predicted versus observed expected improvement for various aggressive window sizes. Recall that we measure expected improvement as the ratio of the number of messages processed in the aggressive version and the number of messages processed in the non-aggressive version. As can be seen there is no discernable difference between our predicted and the observed upper bound on the increase in performance. It can also be seen that there is significant potential for increased performance using our approach.

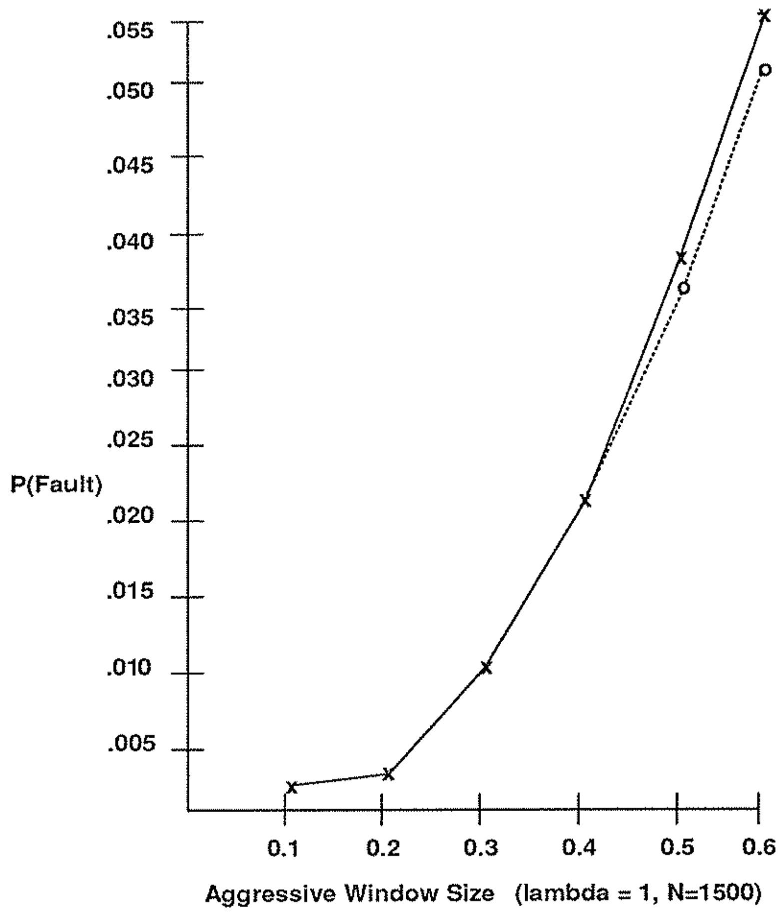

.055
.050
.045
.040
.035
.030
.025
P(Fault)
.020
.015
.010
.005
0.1
0.2
0.3
0.4
0.5
0.6
Aggressive Window Size (lambda = 1, N=1500)

Figure 3.

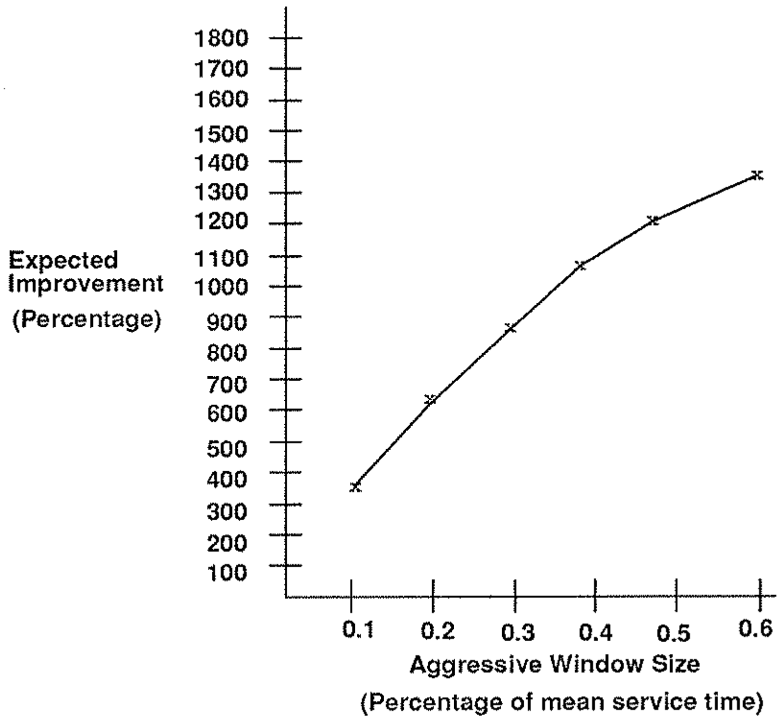


Figure 4.

9. Conclusions

In this paper we developed a model to study the effects of adding aggressiveness to an existing non-aggressive protocol. We have three very significant results from this work. First, we are able to predict the probability of a fault *as a function of the level of aggressiveness.* This is the first time this has been accomplished. Second, we have been able to demonstrate both theoretically and empirically the significant potential for improvement in performance made possible by adding aggressiveness to a non-aggressive approach. Third, we have theoretically demonstrated the very important property that our approach is scalable. We have shown that the probability of a fault does not increase *as the number of LPs approaches infinity.* Also, we have identified the number of LPs, N^*, such that $N^* < N_1 < N_2$ implies that the probability of a fault in a system with N_2 LPs in less than the probability of a fault in a system with N_1 LPs.

We are currently in the process of conducting a sensitivity analysis of our model. In particular, we are interested in the predictive ability of the model given communication patterns other than the equally likely pattern used in our model. Also, we are in the process of developing a protocol to take advantage of the potential increase in performance made possible by this approach.

REFERENCES

Akyildiz, I.F., L. Chen, S.R. Das, R.M. Fujimoto and R.F. Serfozo 1992. "Performance Analysis of "Time Warp" With Limited Memory". *1992 ACM Sigmetrics Conference.*

Breiman, Leo 1986. Probability and Stochastic Processes With a View Toward Applications. *The Scientific Press.* Second Edition.

Chandy, K.M. and J. Misra 1979. "A Case Study in the Design and Verification of Distributed" Programs. *IEEE Transactions on Software Engineering* SE-5,5 May 1979, 440-452.

Chandy, K.M. and R. Sherman 1989. "The Conditional Event Approach to Distributed Simulation". *Proceedings of the 1989 SCS Multiconference on Distributed Simulation*, January, 1989, 93-99.

Dickens, P.M. and P.F. Reynolds 1990. "SRADS with Local Rollback". *Proceedings of the 1990 SCS Multiconference on Distributed Simulation*, January, 1990, 161-164.

Dickens, P.M. and P.F. Reynolds 1991. "A Performance Model for Parallel Discrete Event Simulation". *1991 Winter Simulation Conference.* December, 1991, 618-626.

Felderman, Robert and Leonard Kleinrock 1991. "Bounds and Approximations for Self-Initiating Distributed Simulation Without Lookahead." Submitted (September 1991): *ACM Transactions on Modeling and Computer Simulation.*

Gupta, Anurag, I.F. Akyildiz and R.M. Fujimoto 1991. "Performance Analysis of Time Warp With Multiple Homogeneuos Processors". *IEEE Transactions on Software Engineering.* Volume 17 (No. 10):1013-1027, Oct. 1991.

Jefferson, D.R. 1985. "Virtual Time". *ACM Transactions on Programming Languages and Systems*, 7,3 (1985), 404-425.

Lubachevsky B. 1988. "Bounded Lag Distributed Discrete Event Simulation". *Proceedings of the 1988 SCS Multiconference on Distributed Simulation*, January, 1988,183-191.

Lubachevsky B., A. Shwartz and A. Weiss 1989. "Rollback Sometimes Works... If Filtered". *Proceedings of the 1989 Winter Simulation Conference.* December, 1989, 630-639.

Lubachevsky, B. 1989a. "Scalability of the Bounded Lag Distributed Event Simulation". *Proceedings of the 1989 SCS Multiconference on Distributed Simulation*, January, 1989, 100-105.

Nicol, David M. 1991. "Performance Bounds on Parallel Self-Initiating Discrete Event Simulations". *ACM Transactions on Modelling and Computer Simulation.* Volume 1, No. 1.

Peacock, J.K, E.G. Manning and J.W.Wong 1979. "Distributed Simulation Using a Network of Processors". *Computer Networks* 3 (1979), 44-56, North-Holland Publishing.

Reynolds, P.F. 1988. "A Spectrum of Options for Parallel Simulation". *Proceedings of the 1988 Winter Simulation Conference*, December 12-14, San Diego, California, 325-332.