Reducing Null Message Traffic in Large Parallel and Distributed Systems

Syed S. Rizvi and Khaled M. Elleithy
Computer Science and Engineering Department
University of Bridgeport,
Bridgeport, CT 06601
{srizvi, elleithy}@bridgeport.edu

Aasia Riasat
Department of Computer Science
Institute of Business Management
Karachi, Pakistan 78100
aasia.riasat@iobm.edu.pk

Abstract

Null message algorithm (NMA) is one of the efficient conservative time management algorithms that use null messages to provide synchronization between the logical processes (LPs) in a parallel discrete event simulation (PDES) system. However, the performance of a PDES system could be severely degraded if a large number of null messages need to be generated by LPs to avoid deadlock. In this paper, we present a mathematical model based on the quantitative criteria specified in [12] to optimize the performance of NMA by reducing the null message traffic. Moreover, the proposed mathematical model can be used to approximate the optimal values of some critical parameters such as frequency of transmission, Lookahead (L) values, and the variance of null message elimination. In addition, the performance analysis of the proposed mathematical model incorporates both uniform and non-uniform distribution of L values across multiple output lines of an LP. Our simulation and numerical analysis suggest that an optimal NMA offers better scalability in PDES system if it is used with the proper selection of critical parameters.

Keywords— Conservative distributed simulation, discrete event, null messages, parallel and distributed systems.

1. Introduction

While there has been much research evaluating the performance of conservative NMA in terms of message transmission overhead and processor idle time, there has been comparatively little work devoted to suggesting any potential optimization for the NMA. This paper presents a mathematical model based on the quantitative criteria specified in [12] to optimize the performance of NMA by minimizing the null message transmission across each LP.

In PDES systems, the distributed discrete events need to be tightly synchronized with each other in order to produce the correct results. However, if these discrete events are not properly synchronized, the performance of a PDES environment may degrade significantly [2]. Time management algorithms are, therefore, required to ensure that the execution of a PDES is properly synchronized. Two main classes of time management algorithms are *optimistic* and *conservative*. In optimistic time management algorithm, errors are detected and recovered at run time. However, the performance of optimistic synchronization protocols is mainly dependent on the transmission delay [13]. On the other hand, in conservative PDES, each LP processes events strictly in time stamp order. Since all LPs do not have a consistent view of the state of the entire system, LPs must exchange information to determine when events are safe to process [1, 3].

Although, much research has been done to evaluate the performance of conservative NMA for inefficiencies and overhead [3, 12], none of them suggest any potential optimization for the NMA. Reference [12] proposed a quantitative criterion that incorporates many critical parameters relevant to the performance of NMA. It has been shown that the selection of values for several critical parameters such as the values for Lookahead (L), null message ratio (NMR), and frequency of transmission plays an important role in the generation of null messages [12]. If these values are not properly chosen by a simulation designer, the result will be an excessive number of null messages across each LP. This situation gets more severe when the NMA needs to run to perform a detailed logistics simulation in a distributed environment to simulate a huge amount of data [9].

This paper presents a mathematical model based on the quantitative criteria specified in [12] to optimize the performance of NMA by reducing the null message

978-1-4244-2703-1/08/$25.00

traffic. The reduction in the null message traffic significantly improves the performance of a PDES system by both minimizing the transmission overhead and maintaining a consistent parallelization. Moreover, the proposed mathematical model can be used to approximate the optimal values of some critical parameters such as frequency of transmission, L values, and the variance of null message elimination. These optimal values can be further used to eliminate unnecessary generation of null messages across the LPs. In addition, the performance analysis of the proposed mathematical model incorporates both uniform and non-uniform distribution of L values across multiple output lines of an LP. Our simulation and numerical analysis suggest that an optimal NMA offers better scalability in PDES system if it is used with the proper selection of critical parameters

The rest of the paper is organized as follows. Section 2 provides an overview of the conservative synchronization protocols. Section 3 presents the proposed mathematical model based on the quantitative criteria specified in [12]. Section 4 discusses the potential optimizations in the NMA based on the proposed mathematical model. Section 5 presents a performance analysis for both the proposed mathematical model and the optimizations for NMA. Finally, Section 6 concludes the paper.

2. Related work

Event synchronization is an essential part of parallel simulation. In general, synchronization protocols can be categorized into two different families: conservative and optimistic. Conservative protocols fundamentally maintain causality in event execution by strictly disallowing the processing of events out of timestamp order. The main problems faced in conservative algorithms are overcoming deadlock and guaranteeing the steady progress of simulation time. Examples of conservative mechanisms include Chandy, Misra and Byrant's NMP [6], and Peacock, Manning, and Wong [11] avoided deadlock through null messages. The primary problem associated with null messages is that if their timestamps are chosen inappropriately, the simulation becomes choked with null messages and performance suffers. Some intelligent approaches to null message generation include generation on demand [8], and generation after a time-out [5]. Some earlier research on discrete event simulation has focused on variants of null message protocol (NMP, with the objective of reducing the high null message overhead. For instance, Bain and Scott [4] attempt to simplify the communication topology to resolve the problem of transmitting redundant null messages due to low Lookahead cycles. Other recent developments [10] have focused on incorporating knowledge about the LP into the synchronization algorithms. Cota and Sargent [7] focused on the skew in simulation time between different LPs by exploiting knowledge about the LPs and the topology of the interconnections.

Although earlier work has aimed to optimize the performance of the NMA by proposing the variants of the NMP [3, 4, 8, 10, 12], it has not addressed reducing the exchange of null messages that is caused by improper selection of the parameters.

The principal problem with the NMA is that it uses only the current simulation time of each LP and the L value to predict the minimum time stamp of messages it can generate in the future [12]. These messages with the minimum time stamp are then used to avoid deadlock. As a result, if one of the important parameters such as the L value is chosen poorly, the performance will degrade significantly due to an excessive number of null messages. However, the prediction of minimum time stamps of messages can be improved by understanding the relationship between the time stamp and the L value [12].

3. Mathematical model for NMA

A PDES environment involves synchronization overhead which is added due to the distributed nature of simulation. With NMA, this overhead is mainly associated with the transmission of null messages. Therefore, when comparing the performance of a PDES environment that uses NMA with the performance of sequential execution, the message overhead can make a significant performance difference between the two approaches. Before presenting a proposed mathematical model, it is worth mentioning some of our key assumptions.

- We assume that the value of L may change during the execution of a Lookahead period. However, the values of L can not instantaneously be reduced.
- Initially, a constant event arrival or job intensity rate is assumed for each participating LP in the simulation. However, for the sake of experimental verifications, we also consider the non-uniform distribution of L values across multiple output lines of an LP.
- For the frequency of message transmission, we assume that all messages are equally distributed among the LPs. For the proposed mathematical model, we assume that we have n number of LPs in the simulation where all LPs are connected with each

other by means of highly reliable mesh networks topology. Also, each LP maintains *n-1* input and output links for both input and output neighboring LPs, respectively.

3.1. Definition of system parameters

All model variables, along with their definition, are listed in Table 1. Based on the concept of NMA, we assume that each LP maintains two clock times, one for each of its input and output neighbors as shown in Figure 1. One is the minimum receiving time (*MRT*) for the input neighbor LP whereas the second is the minimum sending time (*MST*) for the output neighbor LP. The *MRT* represents an earliest time when an LP can receive an event message from one of its input neighboring LPs, where as the *MST* represents an earliest time by which an LP can send a message to one of its neighboring LP. The performance (*P*) of a conservative distributed simulation environment mainly depends on the amount of computation required for processing an event per second. In addition, the event arrival rate (ρ) represents the number of events that occur per second (in practice, events occur per simulation second). Unlike performance, the parameter ρ mainly depends on the model. Lookahead (*L*) is measured in seconds. Frequency of transmission (F_T) is the frequency of sending a message from one LP to another. In addition, T_{Null} represents the timestamp of a null message sent from one LP to other LPs.

In order to measure the performance, it is imperative to consider one parameter that can compute simulation time advancement *(STA)*. The *STA* can be defined as a ratio of performance to event arrival rate. This relationship can be expressed as:

$$STA = P/\rho \qquad (1)$$

The value of *MRT* is updated by the time stamp of a null message coming from other neighboring LPs on one of the input links of a receiving LP. Any event scheduled by an LP must have a timestamp at least as large as the LP's simulation time clock [1]. This requirement is also referred as the local causality constraint requirement. To strictly follow this requirement, a large number of null messages can be transmitted by LPs before the non null-messages can be processed. This large message overhead may degrade the performance of a conservative distributed simulation. It is, therefore, worth computing the ratio of null messages to the total messages transmitted among LPs. The NMR can be simply defined as the ratio of total number of null messages to total messages where total messages include both null and event messages. Mathematically, it can be expressed as follows:

$$NMR = \frac{Total\ Number\ of\ Null\ Messages}{Total\ Messages} \qquad (2)$$

Table 1
System parameter definition

Parameter	Definition
P	Computation required for processing an event per second
ρ	Event arrival rate (events per second)
MRT	Minimum receiving time
MST	Minimum sending time
L	Lookahead
STA	Simulation time advancement
F_T	Frequency of transmission
T_{Null}	Timestamp of a null message
T_S	Current simulation of a LP
T_{Total}	Total simulation time in seconds

3.2. The proposed mathematical model

First, we present a mathematical model based on the quantitative criteria specified in [12]. In addition, the proposed mathematical model is also based on the internal architecture of an LP as shown in Fig. 1. The architecture for *m* number of LPs is shown in Fig. 2. Using the quantitative criteria defined in [12], we can approximate the advancement in the simulation time as a ratio of performance to event arrival rate. This leads us to the following mathematical expression of the relative speed for simulation advancement:

$$(P)\left\{T_S / E_{Msg}\right\} \qquad (3)$$

Taking this into account, we can give the following hypothesis for approximating the number of null messages transmitted per LP: "*If we assume that we have an average value of* ***L*** *which associates with one of the output lines of an LP, then* ***P*** *can be approximated as*":

$$P \cong E_{Msg}\,(1/L) \qquad (4)$$

Combining (3) and (4) yields the estimated number of null messages transmitted per LP that has only one output line as shown in (5).

$$Null_{(LP)} \triangleq E_{Msg}\,(1/L)\left(T_S/E_{Msg}\right) \triangleq T_S\,(1/L) \qquad (5)$$

Furthermore, if we assume that we have *O* number of output lines attached with each LP with the uniform distribution of L value on each output line, then (5) can

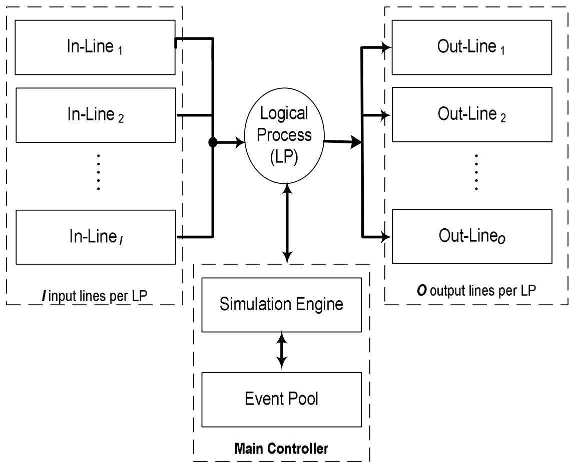

Figure 1. Internal architecture of an LP

be further generalized for O number of output lines per LP as follows:

$$Null_{(LP)} \triangleq E_{Msg}\left(O/L\right)\left(T_S/E_{Msg}\right) \triangleq T_S\left(O/L\right) \qquad (6)$$

It should be noted that (6) represents total number of null messages transmitted per LP via O number of output lines to the neighboring LPs. If we assume that we have m number of total LPs present in a system where each LP has O number of output lines, then this allows us to extend (6) and generalize it for m number of LPs present in a distributed simulation as shown in (7). It can be seen that (7) gives total number of null messages exchange among all LPs.

$$Null_{(m-LP)} \triangleq E_{Msg}\left(O/L\right)m\left(T_S/E_{Msg}\right) \triangleq T_S\left(O/L\right)m \qquad (7)$$

where the term O/L in (7) shows a uniform distribution of L value for O number of output lines.

The assumption of uniform distribution of Lookahead among O output lines of an LP simplifies

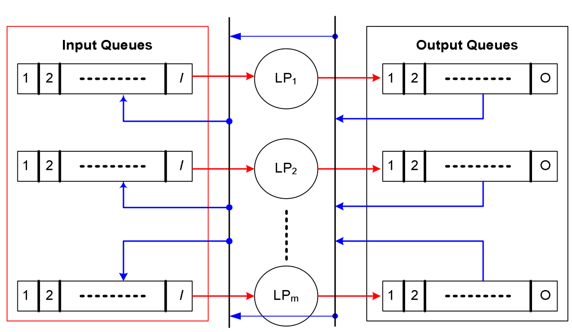

Figure 2. *m* number of LPs with *I* number of input queues and *O* number of output queues

the procedure for computing the number of null messages transmitted per LP to other neighboring LPs. However, the values for L may change during the execution of a Lookahead period that makes the uniform distribution assumption of Lookahead a little unrealistic. This argument leads us to the fact that we should also account a non-uniform distribution method for Lookahead where each output line of an LP can have a different value of L. We can rewrite (6) as:

$$Null_{(LP)} \triangleq \sum_{i=1}^{O}\left(E_{Msg}/L_i\right)\left(T_S/E_{Msg}\right) \triangleq \sum_{i=1}^{O} T_S\left(1/L_i\right) \qquad (8)$$

It should be noted that (8) represents the total number of null messages transmitted per LP to other neighboring LPs.. If we assume that the model is partitioned into m number of total LPs where each LP can have at most O number of output lines, this allows us to extend (8) for m number of LPs.

$$Null_{(m-LP)} \triangleq \sum_{h=1}^{m}\sum_{i=1}^{O}\left(E_{Msg}/L_{hi}\right)\left(T_S/E_{Msg}\right) \triangleq \sum_{h=1}^{m}\sum_{i=1}^{O} T_S\left(1/L_{hi}\right) \qquad (9)$$

It can be evident that (9) gives the total number of null messages exchange among all LPs.

4. Performance optimization of NMA

In this section, we introduce two different ways to optimize the performance of NMA. We first implement the concept of frequency of transmission described in [12] to minimize the exchange of null messages across the LP. Secondly, we present the new concept of variance that works with the frequency of transmission to avoid the unnecessary generation of null messages and consequently minimize the overall synchronization overhead. For both concepts, we derive a closed form mathematical expression that can be used to evaluate the performance of NMA in the presence of deadlock situation.

4.1. Frequency of transmission

Transmission of null-messages on each occurrence of an event results in unnecessary generation of null messages that causes an increase in the synchronization overhead. We believe, instead of sending null message after every one event, it should be transmitted with respect to a certain frequency of transmission. This frequency of transmission (F_T) is a fixed amount of time and it should be measured in simulation second per second. Recall (2) and (4), number of events processed per second per LP can be equated from both equations. This yields the following approximation for F_T in term of L value.

$$E_{Msg} L / F_T \cong 2E_{Msg}(1/L) \Rightarrow L \cong \sqrt{2F_T} \quad (10)$$

Substituting the value of (10) into (5), we get,

$$Null_{(LP)} \triangleq E_{Msg}\left(\frac{1}{\sqrt{2F_T}}\right)\left(\frac{T_S}{E_{Msg}}\right) \triangleq T_S\left(\frac{1}{\sqrt{2F_T}}\right) \quad (11)$$

Equation (11) can be generalized for O number of output lines per LP when the number of null messages is assumed to generate with a certain frequency of transmission as shown in (12).

$$Null_{(LP)} \triangleq E_{Msg}\left(\frac{O}{\sqrt{2F_T}}\right)\left(\frac{T_S}{E_{Msg}}\right) \triangleq T_S\left(\frac{O}{\sqrt{2F_T}}\right) \quad (12)$$

Equation (12) gives an estimated number of null messages transmitted by an LP that has O number of output lines where each line carry an equal percentage of the L value in terms of a fixed frequency of transmission per output line. In addition, if we assume that the system consists of m number of total LPs where each LP has a fixed number of output lines, then (12) can be further extended for m number of LPs.

$$\begin{aligned}&Null_{(m-LP)} \triangleq \left(E_{Msg} \times m\right)\left(O/\sqrt{2F_T}\right)\left(\frac{T_S}{E_{Msg}}\right) \triangleq T_S \times m\left(O/\sqrt{2F_T}\right)\\ &where\ \sqrt{F_T} \xrightarrow{\%} L\end{aligned} \quad (13)$$

where the denominator of (13) (i.e., $O/\sqrt{F_T}$) represents a uniform rate of null message transmission per output line. Based on (13), we can conclude that a non uniformity in null message algorithm results a non linear generation of null messages. An expression can be derived for O number of output lines where each line may carry a different value of F_T

$$\begin{aligned}&Null_{(LP)} \triangleq \sum_{i=1}^{O}\left(\frac{E_{Msg}}{\sqrt{2F_{Ti}}}\right)\left(\frac{T_S}{E_{Msg}}\right) \triangleq \sum_{i=1}^{O} T_S\left(\frac{1}{\sqrt{2F_{Ti}}}\right)\\ &where\ \sqrt{F_{Ti}} \xrightarrow{\%} L_i\end{aligned} \quad (14)$$

Furthermore, (14) can be further extended and generalized for m number of LPs where each LP can have at most O number of output lines.

$$\begin{aligned}&Null_{(m-LP)} \triangleq \sum_{k=1}^{m}\sum_{i=1}^{O}\left(E_{Msg}/\sqrt{2F_{T(ki)}}\right)\left(T_S/E_{Msg}\right)\\ &\triangleq \sum_{k=1}^{m}\sum_{i=1}^{O} T_S\left(\frac{1}{\sqrt{2F_{T(ki)}}}\right) where\ \sqrt{F_{T(ki)}} \xrightarrow{\%} L_{(ki)}\end{aligned} \quad (15)$$

4.2. Variance for null message elimination

Also, in this scenario, it is essential to cancel out the unnecessary generation of null messages. Variance represents the probability of cancellation of unnecessary null messages. The value of variance may exist between 0 and 1. It should also be subtracted from 1, so that we can show that increase in variance causes a decrease in the over all null messages where as a decrease in variance results an increase in null messages. If we consider variance as 0, then it should give us the same results that we could achieve with out using variance. In order to reflect the variance of null message cancellation, we can rewrite (13) for m number of LPs with the uniform distribution of null message transmission per output:

$$\begin{aligned}&Null_{(m-LP)} \triangleq \left(E_{Msg} \times m\right)\left(O/\sqrt{2F_T}\right)\left(\frac{T_S}{E_{Msg}}\right)(1-\sigma)\\ &\triangleq T_S \times m\left(\frac{O}{\sqrt{2F_T}}\right)(1-\sigma)\ where\ \sqrt{F_T} \xrightarrow{\%} L\ and\ 0 \le \sigma < 1\end{aligned} \quad (16)$$

where σ represents probability of null message cancellation.

The same concept of null message cancellation can be implemented with a simulation model where the L values are non-uniformly distributed among O number of output lines. This leads us to the following modification in (16):

$$\begin{aligned}&Null_{(m-LP)} \triangleq \sum_{k=1}^{m}\sum_{i=1}^{O}\left(E_{Msg}/\sqrt{2F_{T(ki)}}\right)\left(\frac{T_S}{E_{Msg}}\right)(1-\sigma)\\ &\triangleq \sum_{k=1}^{m}\sum_{i=1}^{O} T_S\left(1/\sqrt{2F_{T(ki)}}\right)(1-\sigma)\ where\ 0 \le \sigma < 1\end{aligned} \quad (17)$$

5. Performance analysis of NMA

For the sake of performance analysis, we simulate 5 different cases. The system is modeled in C++.

5.1. Multiple output lines per LP

Using (6), Fig.3 shows the null message transmission with the following simulation parameters: simulation time (Ts) = 500 sec, L is uniformly distributed per output line. The number of output line may vary from 0 to 8 for all results as show in Fig.3. Simulation results of Fig. 3 presents a comparison of null message transmission per LP versus multiple output lines.

5.2. Multiple LPs with multiple output lines per LP

We assume that we have multiple LPs with O output lines (fixed per LP). Let the output lines per LP is 4 with the (Ts) of 500 sec. Using (7), Fig.4 shows the null message transmission with the following simulation parameters: Ts = 500 sec, L is uniformly distributed per output lines, the output lines are assumed to be fixed for each LP (O = 4). The numbers of LPs are varied from 1 to 10 as show in Fig.4.

5.3. Multiple output lines per LP with non-uniform distribution of Lookahead

For this simulation, we assume that we have single LP that has O number of output lines where each output line of an LP can have different value of L. Using (8), Fig.5 shows the null message transmission with the following simulation parameters: Ts = 500 sec, L is non-uniformly distributed per output lines. The numbers of output lines may vary from 1 to 10 as show in Fig.5. Also, it should be noted that the value of L is chosen randomly within the range of 0 to 1 and assigned to each output line at run time. This random selection may control the generation of unnecessary null messages as long as the value is chosen appropriately.

5.4. Multiple LPs with multiple fixed output lines

For this simulation, we assume that we have multiple

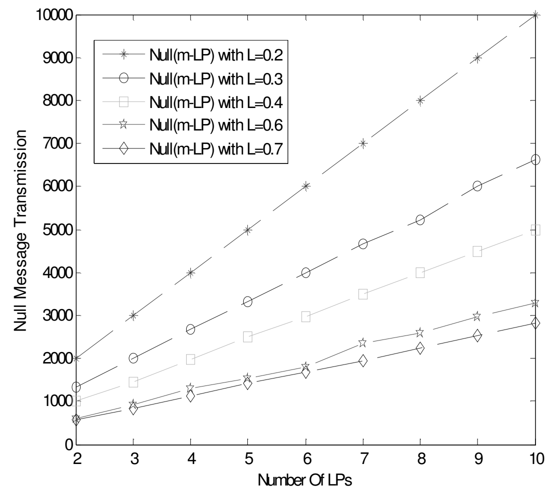

Figure 4. Multiple LPs with fixed output lines per LP versus null message transmission

LPs that can have fixed number of output lines where each line of an LP can have different value of L. Using (9), Fig.6 shows the null message transmission with the following simulation parameters: Ts = 500 sec, L is non-uniformly distributed per output lines. The numbers of LPs are varied from 1 to 20. Also, it should be noted that the value of m and O are both varying quantity for this particular scenario. In harmony with our expectations, the number of null messages increases due to an increase in number of LPs. However, this increase in null messages is limited and controlled due to the random behavior of Lookahead. This can also be considered as irregular networks due to the non uniform distribution.

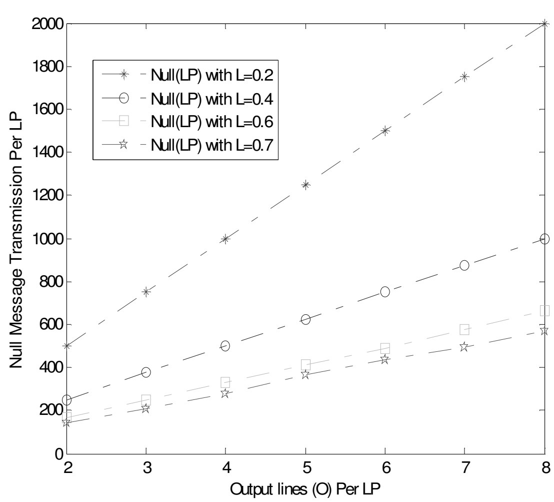

Figure 3. Multiple output lines per LP versus null message transmission per LP

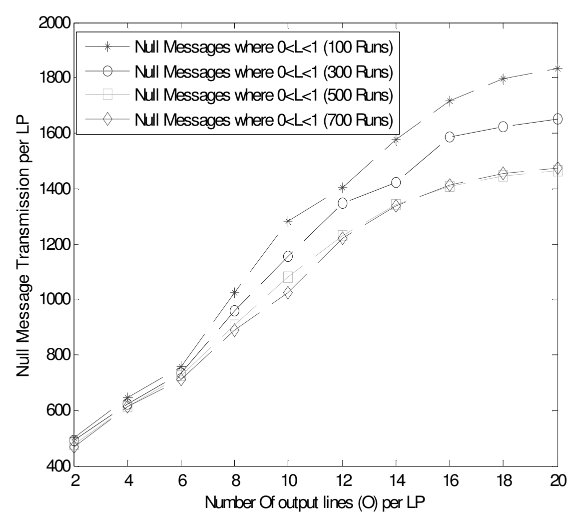

Figure 5. Multiple output lines per LP with non-uniform distribution of L value

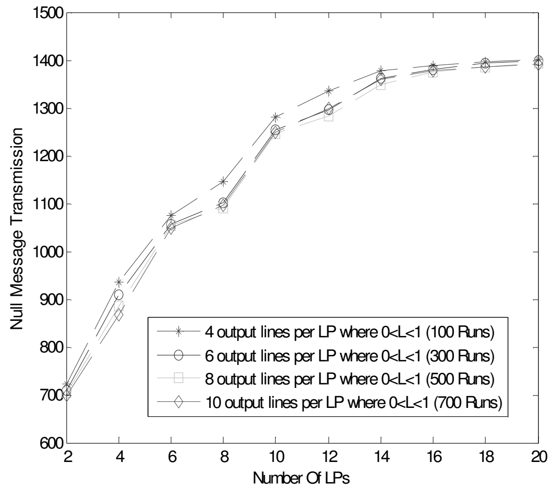

Figure 6. Multiple LPs and multiple fixed output lines with non-uniform distribution of *L* versus null message transmission

6. Conclusion

We have proposed a mathematical model to predict the optimum values of critical parameters that have great impact on the performance of NMA. The derived properties of the proposed mathematical model account for the cases when the NMA would send too many null messages. The proposed mathematical model provides a quick and practical way for simulation designers to predict whether a simulation model has potential to perform well under NMA in a given simulation environment by giving the approximate optimal values of the critical parameters. We have experimentally verified that if critical parameters, specifically the *L* value, are chosen intelligently, we can limit the transmission of null messages among the LPs and consequently improve the performance of NMA in a distributed simulation environment.

REFERENCES

[1] R. M. Fujimoto, "Distributed Simulation system," *Proceeding of the 2003 winter simulation conference*. College of Computing, Georgia Institute of Technology, Atlanta.

[2] Y.M. Teo, Y.K. Ng and B.S.S. Onggo, "Conservative Simulation using Distributed Shared Memory," *Proceedings of the 16th Workshop on Parallel and Distributed Simulation (PADS-02), IEEE Computer Society,* 2002.

[3] B. R. Preiss, W. M. Loucks, J. D. MacIntyre, J. A. Field, "Null Message Cancellation in Conservative Distributed Simulation," *Distributed Simulation 91 Proceedings of the SCS Multiconference on Advances in Parallel and Distributed Simulation,* 1991.

[4] W. L. Bain, and D. S. Scott, "An Algorithm for Time Synchronization in Distributed Discrete Event Simulation", *Proceedings of the SCS Multiconference on Distributed Simulation,* Vol.3, pp. 30-33, February, 1988.

[5] N. J. Davis, D. L. Mannix, W. H. Shaw, and Hartrum, T. C., ''Distributed Discrete-Event Simulation using Null Message Algorithms on Hypercube Architectures,'' *Journal of Parallel and Distributed Computing*, Vol. 8, No. 4, pp. 349-357, April 1990.

[6] K. M. Chandy and J. Misra, "Distributed Simulation: A case study in design and verification of distributed programs", *IEEE Transactions on Software Engineering,* SE-5:5, pp. 440-452, 1979.

[7] B. A. Cota and R. G. Sargent, ''An Algorithm for Parallel Discrete Event Simulation using Common Memory,'' *Proc. 22nd Ann. Simulation Symp.*, pp. 23-31, March 1989.

[8] J. K. Peacock, J. W. Wong, and E. Manning, ''Synchronization of Distributed Simulation using Broadcast Algorithms,'' *Computer Networks*, Vol. 4, pp. 3-10, 1980.

[9] L. A. Belfore, S. Mazumdar, and S. S. Rizvi et al., "Integrating the joint operation feasibility tool with JFAST," *Proceedings of the Fall 2006 Simulation Interoperability Workshop,* Orlando Fl, September 10-15 2006.

[10] D. M. Nicol and P. F. Reynolds, ''Problem Oriented Protocol Design,'' *Proc. 1984 Winter Simulation Conf.*, pp. 471-474, Nov. 1984.

[11] J. K. Peacock, J. W. Wong, and E. Manning, ''A Distributed Approach to Queuing Network Simulation,'' *Proc. 1979 Winter Simulation Conf.*, pp. 39 9-406, Dec. 1979.

[12] Syed S. Rizvi, K. M. Elleithy, Aasia Riasat, "Minimizing the Null Message Exchange in Conservative Distributed Simulation," *International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering, CISSE 2006*, pp. 443-448, Bridgeport CT, December 4-14 2006.

[13] Syed S. Rizvi, Khaled M. Elleithy, and Aasia Riasat, "Trees and Butterflies Barriers in Distributed Simulation System: A Better Approach to Improve Latency and the Processor Idle Time", *IEEE International Conference on Information and Emerging Technologies (ICIET-2007),* pp. 1 – 6, July 06-07, 2007, Karachi, Pakistan.