

# System Buffer Size Requirements: An Application's Perspective\*

Xin Li and Valerie E. Taylor  
Department of Electrical and Computer Engineering  
Northwestern University  
Evanston, IL 60016

## Abstract

This paper entails an in-depth study of the system buffer size requirements for parallel scientific applications that communicate regularly with four or six neighbors. The goal of this study is to address two major issues: (1) for future system designs, which system features significantly impact the system buffer size requirements and (2) for systems where users can set the system buffer size (e.g., IBM SP2 and SGI-Cray T3E), what is a good estimate to use for good performance. We propose an analytical model of buffer size requirements for message passing systems. This model is validated experimentally for the IBM SP2 system. We then use the model to perform sensitivity analysis on buffer size requirements with varying network bandwidth, receive latency, memory copy rate, delay between messages, and different number of messages. The analytical results indicate that the two major factors that affect the buffer size are the delay between messages and the increase in network bandwidth. Further, these two factors are related. The network bandwidth has little impact on buffer size requirements when the delay is small, but has significant impact when the delay is large. The receive startup latency and receive memory copy rate have very little impact on buffer size requirements.

**Keywords** system buffer size, message passing systems, nearest neighbor communication pattern

## 1 Introduction

It is well-known that the overall performance of parallel applications is highly dependent on communication performance. Communication performance is determined by many factors including communication overhead, network bandwidth, message length, number of messages, network contention and message

buffer size. In this paper we focus on the system buffer size, which is very important for application performance. A small buffer size can add idle time to the communication cost. A large buffer size can reduce the amount of memory available to applications, which can cause paging and significantly affect application performance. Hence, it is important to have an appropriately sized system buffer. This paper addresses two major issues related to buffer size requirements: (1) for future system designs, which system features significantly impact the system buffer size requirements and (2) for systems where users can set the system buffer size (e.g., IBM SP2 and SGI-Cray T3E), what is a good estimate to use for good performance.

In distributed memory machines, messages are communicated among processors by one of two possible protocols, eager or rendezvous[4]. The eager protocol, utilizes system buffers at the send and receive nodes. The send buffer allows the program to continue execution after the message has been placed in the buffer; the receive buffer allows the receiver to avoid stalls waiting for the message arrival. The receive buffer holds the messages received from other processors until the corresponding receive has been posted. Hence, the eager protocol reduces the idle time (due to network stalls or non posting of receives) at the cost of copying to the buffers. It is used most often with small to middle size messages.

The rendezvous protocol, bypasses the system buffers to allow copying from the sender's application space to the receiver's application space directly. This protocol requires hand-shaking between the sender and receiver to insure that both nodes are ready for the copy. It saves on the copying to system buffers at the cost of large idle time due to the hand-shaking. This protocol is often used with very large messages to compensate for the large idle time. The eager protocol, however, can result in performance advantages over the rendezvous protocol for most MPMD and SPMD applications for which the load is not perfectly balanced across the nodes. The size of the system buffer

---

\*This work was supported in part by an NSF NGS grant, EIA-9974960, and an NSF ITR grant, 0085952.

is essential for the communication performance in the eager protocol.

This paper entails an in-depth study of the system buffer size requirements for parallel scientific applications. Generally, the default user-level system buffer size is set to a very large value. For example, with the IBM SP2 system, the default user-level system buffer size is 64 MB [6]. As we consider future large scale systems, such as petaflop systems, the number of nodes in the system will be large (in the range of thousands of nodes similar to the ASCII machines). It is important to determine the appropriate size so as not to waste memory. The system buffer can be implemented as virtual memory or resident in memory. The virtual memory implementation allows for a very large buffer size, which is done with the IBM SP2, but can significantly impact performance when a page fault occurs. It is better to keep the buffer resident in memory all the time to avoid page faults; for this case it becomes critical to have a good estimate of the buffer size. The question that results is what size to make this buffer and how the estimate will change under different conditions such as the speed of the network or the number of local messages.

We developed an analytical model of the system buffer for message passing systems and experimentally validated the model on an IBM SP2 system. We used this model to perform sensitivity analysis on the following parameters: (1) the delay between message arrivals, (2) the receive latency, (3) the network bandwidth, (4) the receive memory copy rate, and (5) the number of messages. These factors are analyzed because of the general belief that these factors significantly impact buffer size requirements [14]. The results provide insight into the requirements of a typical system and application.

The analyses indicate that the two major factors that affect the buffer size are the delay between messages and increase in network bandwidth. Further, these two factors are related. The network bandwidth has little impact on the increase in buffer size when the delay is small, but has significant impact when the delay is large. The receive startup latency and receive memory copy rate have very little impact on buffer size requirements. Lastly, as expected, the buffer size varies almost linearly with the number of messages.

The rest of this paper is organized as follows: Section 2 provides background materials for our study. Section 3 proposes our analytical model for system buffer size requirement analysis. Section 4 describes our model validation experiments on an IBM SP2 system. Section 5 presents our sensitivity analysis results

on system buffer size requirements. Section 6 concludes our work.

## 2 Background

We present the background of our study in this section. First we discuss the communication pattern for four and six neighbors. One such application for which this pattern can occur is finite element applications described below. Further, we describe the user-level system buffer management system for two commercially-available machines, Cray T3E and IBM SP2.

### 2.1 Nearest Neighbor Communication Pattern

Finite element applications are widely used scientific applications. The finite element method involves a meshing procedure, which is suitable for parallel and distributed computing. The meshing procedure entails dividing the original problem domain into elements, which are geometrical shapes composed of grid points. Implementing the finite element method in parallel involves partitioning the global domain into  $P$  connected sub-domains that are distributed among  $P$  processors; each processor executes the numerical technique on its assigned sub-domain. Communication occurs between processors that share common grid points [13, 15]. Our work looks at the class of applications that communicate regularly with four neighbors and six neighbors via small fixed-size messages. Our analysis explores the buffer size requirements for four and six messages arriving at a processor, with different delays between message arrivals.

### 2.2 Buffer Size Management

The system buffer management is important for communication performance and it is available on most parallel machines. Users are allowed to fine tune their application performance through certain environment variables or switches. In this section, we present the system buffer management system of the Cray T3E and the IBM SP2.

#### 2.2.1 Cray T3E

On a Cray T3E system, messages can either be sent using the eager protocol or the rendezvous protocol. An environment variable `MPI_BUFFER_MAX` can be used to specify a maximum message size in bytes that

will use the eager protocol for the different MPI send modes, standard, buffer or ready [3]. Users can also decide the size of the system buffer by setting an environment variable `MPLBUFFER_TOTAL`. The default values for both environment variables are unlimited. This implies that messages of any size are buffered by the system (only limited by the total amount of memory available). If `MPLBUFFER_MAX` is set to zero, messages are not buffered. This implies that before sending any data, a standard `MPI_Send` has to block until a matching `MPI_Recv` is posted. The wait time could be arbitrarily long and as a result could degrade the overall application performance.

Michael Resch et. al. [12] have studied the performance of MPI on a Cray T3E machine. Based on a set of microbenchmark measurements (eg. ping-pong), an optimal size of `MPLBUFFER_MAX` was determined to be 4 KB. This size avoids unnecessary idle time for small messages but allows to make use of higher bandwidth for larger messages. They did not conduct any analysis on `MPLBUFFER_TOTAL`, which is the focus of this paper.

### 2.2.2 IBM SP2

On an IBM SP2 system, in addition to some internal buffering in the communication subsystem, an early arrival buffer space is provided at the receiver side [1, 2, 8]. Short messages are sent using the eager protocol, and are buffered by the receiver if the matching receive is not posted. Long messages use the rendezvous protocol. The eager protocol is used up to a threshold value which can be changed via an environment variable `MP_EAGER_LIMIT`. Its default value is 4 KB [6]. Users can also decide the early arrival buffer size by setting an environment variable `MP_BUFFER_MEM`. Its default size is 64 MB [6]. In section 4, we show that similar performance can be achieved with a much smaller buffer size, thereby saving memory.

While the IBM SP2 system allocates a large amount of memory for system buffering, the buffer is implemented as virtual memory. As discussed earlier, this implementation can impact performance when a page fault occurs. Hence, it is important to set the `MP_BUFFER_MEM`, when possible.

## 3 Model Development

In this section we present the methodology for analyzing the system buffer size requirements. We then validate our results on an IBM SP2 system.

### 3.1 Analytical Model

We developed an analytical model for system buffer size requirement analysis. The model is based on five parameters: message incoming rate, message consuming rate, the receiver side startup time, message size and message frequency. The message incoming and consuming rates and the receiver side startup cost can be obtained through one-time experiments for a given machine. The message frequency is application dependent. In the model, we assume that the message sizes are of the same length and the messages in the buffer get serviced in an FIFO fashion.

Figure 1 shows a buffer size requirement analysis procedure for the case of two messages arriving at the buffer. We use this simple case to illustrate our analysis procedure. The X-axis is time in microseconds and the Y-axis is the system buffer size requirement in bytes. In Figure 1, the term  $\lambda$  denotes the message incoming rate and  $\mu$  denotes the message consuming rate. We denote the receiver side's startup cost as  $\alpha$ , corresponding to the startup time prior to copying the message from the system buffer to application space. Message size is denoted as  $L$ . The delay between two messages is  $t1$ , for which the first message arrives at time 0 and the second at time  $t1$ .

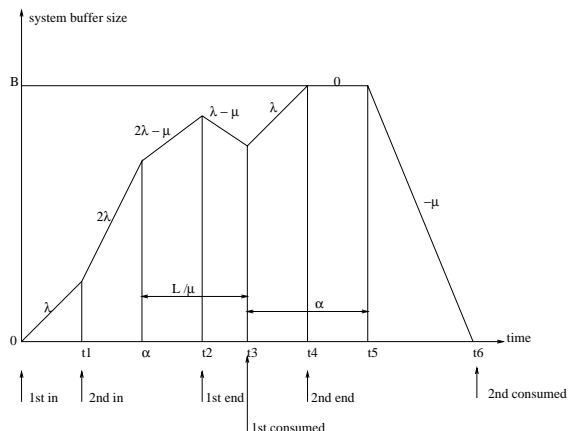


Figure 1: Analytical Model

Given the aforementioned parameters, the analysis entails the following events. The slope between time 0 and  $t1$  is  $\lambda$ , the single message incoming rate. After time  $t1$ , since both messages are arriving each with an incoming rate of  $\lambda$ , the total message incoming rate becomes  $2\lambda$ . After time  $\alpha$ , the receiver starts consuming the first message. The slope now becomes  $2\lambda - \mu$ . At time  $t2$ , the first message has completely arrived. The total message incoming rate becomes  $\lambda - \mu$ . The first message is completely consumed at time  $t3$ . The time

to copy a message from the system buffer into the user space is  $L/\mu$ . While at time  $t_3$ , the second message is still arriving, resulting in the incoming rate equal to  $\lambda$  after time  $t_3$ . After the first message is consumed, the receiver needs to spend another startup time before consuming the second message. In the figure, the second message stops arriving at time  $t_4$ . The receiver is still in its startup process at this time, so the total message incoming rate becomes 0 since we do not have any additional messages arriving. At time  $t_5$ , the receiver starts copying the second message into its user space. The consuming rate is  $\mu$  and the slope goes downwards. After a period of time  $L/\mu$ , the second message gets completely consumed. The system buffer size required throughout this whole process is the value of  $B$ , since the buffer is required to hold all of the incoming messages until they are consumed.

We have developed a program to simulate the analysis procedure. The model requires values for the message incoming rate (or network bandwidth) ( $\lambda$ ), the receiver startup cost ( $\alpha$ ), and the message consuming rate ( $\mu$ ), etc. These parameters are presented in the next section. In the following section, we will use the term network bandwidth for the message incoming rate.

### 3.2 Parameter Specification

In order to validate our analytical model and carry out the buffer size requirement analysis, we obtained communication parameters on an IBM SP2 system and validate the model against the early arrival buffer on the IBM SP2 system. For the experiments, we used the IBM SP2 at Argonne National Laboratory in the Mathematics and Computer Science Division.

We used "ping-pong" benchmark and a variation of the "ping" [9, 10] benchmark to get the network bandwidth ( $\lambda$ ), receiver memory copy rate ( $\mu$ ) and the startup cost ( $\alpha$ ) for point-to-point communications. The code was written in MPI standard mode, for which MPI was compiled to use the native version of the message passing library distributed by IBM. Each benchmark program was executed multiple times, over a range of message sizes, and the minimum value of each parameter was taken.

Table 1 shows the communication parameters we obtained for message size from 4 bytes to 4096 bytes, corresponding to the eager protocol on IBM SP2 system. These parameters were obtained using a linear least square fit to the data. We use these parameters to analyze the system buffer size requirements for the message size in the same range, since by default the system buffer is utilized in this range.

Table 1: Parameters for System Buffer Size Analysis

Message Size (B)	$\alpha$ ( $\mu s$ )	$\lambda$ (MB/s)	$\mu$ (MB/s)
4 - 4096	9	105	91

## 4 Model Validation

Our analytical model estimates the size of the system buffer required by an application. This estimated buffer size should be large enough so as not to stall the communication process of the application, while small enough compared to the default buffer size in order to avoid unnecessary page faults.

In this section we present our experiments for model validation. The experiments entail comparing the performance resulting from using our estimated buffer size versus that resulting from using the default buffer size on the IBM SP2 system. The early arrival buffer space on IBM SP2 systems is used for message handles, data types and messages that arrive before their matching receive is posted. In addition to its use by the application code, the space is used during message passing initialization [1, 2, 7]. A simple empty MPI code was written to measure the buffer size needed for message passing initialization. It was determined that the initialization buffer size is  $10923 * P$  bytes, where  $P$  is the number of processors. This value is added to the results of our analysis to identify the total buffer size requirement. Using the parameters given in Table 1 and a delay value of 5 microseconds ( $\delta = c/8$ , see section 5.1), our analysis indicated a buffer size of 12288 bytes for four message arrivals with message size of 4 KB. Hence, the total buffer size needed for five processors with message size of 4 KB is 75 KB ( $12288 + 10923 * 5$ ).

We developed an MPI program to model four messages sent to one receiver in a staggered fashion and recorded the total communication time. We set up the early arrival buffer space to be 75 KB. We then compared the timing results with that of the default buffer size setting. The program was set up to run 20 times for each message size and the five minimum times were chosen for comparison. Figure 2 shows the comparison of message size in the range from 4 to 4096 bytes. The X-axis is the message size and the Y-axis is the total communication time. The "+" points are the timing results of the default buffer setting of 64 MB and the "o" points are the timing results of our estimated buffer size setting of 75 KB. The dotted line and the solid line in the Figure are the least square fits for the

”o” points and the ”+” points, respectively. The comparison shows that for message sizes in the range of 4 KB the communication times for both buffer sizes are very close. Our estimated buffer size is three orders of magnitude smaller than the default setting (i.e., 75 KB versus 64 MB). It should be noted that when the user sets the buffer size, additional overhead is incurred for each message because of checks conducted by the system. This overhead can be seen with small messages, but gets negligible with messages greater than 1.5 KB.

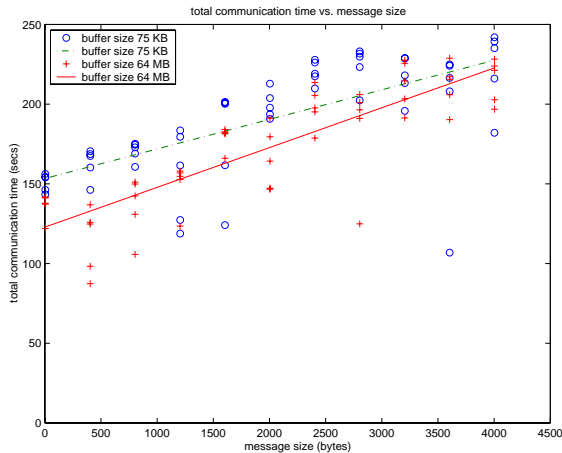


Figure 2: Comparison Between the Default Buffer Setting and the Estimated Buffer Setting

The simulation program that we developed to analyze the buffer size requirements requires only a few seconds to execute. This program is available for others to use to provide an estimate of the buffer requirements for systems that allow user-specified buffer sizes [5].

## 5 System Buffer Size Sensitivity Analysis

In this section, we present our system buffer size sensitivity analysis results using the parameters from the IBM SP2 system. In particular, we used the model to explore the impact on buffer size requirements with varying network bandwidth, receiver startup cost, receive memory copy rate, delays between messages, and different number of messages. For each analysis, we give the exact buffer size requirement for a message size of 4000 bytes (corresponding to the eager protocol) to illustrate the trends.

### 5.1 Delays Between Message Arrivals

In Figure 3, we show how the buffer size requirements change as the delay (denote as  $\delta$ ) between message arrivals changes, where  $c$  equals to  $L/\lambda$  ( $c$  is the time required for the message to be completely placed into the system buffer). This plot is given for four messages. The results indicate that as the delay between messages increases the buffer size decreases significantly. For example, for the message size of 4000 bytes, the buffer size is 12052 bytes for  $\delta = c/8$  but drops to 8971 bytes for  $\delta = c/2$ . We can see that for message size less than 1500 bytes, the buffer size has very little change for delays value between  $c/8$  and  $c/2$ . Further, when the delay is strictly less than  $c/2$ , the buffer size requirements do not change for all message sizes in the range of 0 to 4000 bytes. However, when the delay is greater than or equal to  $c/2$ , there is a significant decrease in the buffer size requirements with increase in message size. For this range of delays, the point at which the split (for which the buffer size decreases significantly) occurs corresponds to the point at which the receiver has enough time to consume a significant portion of the message prior to the arrival of a new message.

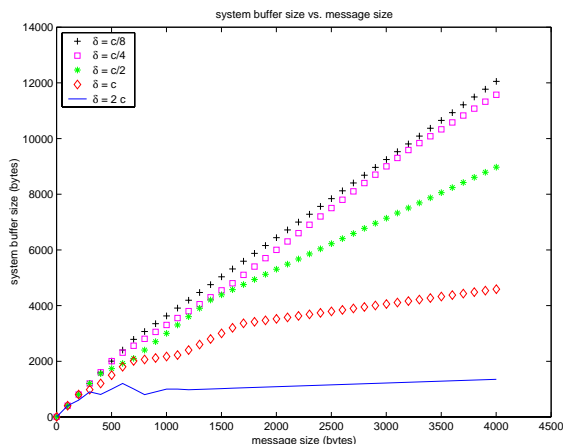


Figure 3: Delays Between Message Arrivals

### 5.2 Varying the Receive Latency

Next we explore the system buffer size requirements when the startup time is varied while other parameters are kept unchanged ( $\delta = c/8$  and the number of messages is four). The startup time is decreased from  $9\mu s$  to  $0\mu s$  first, it is then doubled to  $18\mu s$  to correspond to a receive delay resulting from load imbalance. The results are given in Figure 4. We can see that

the buffer size requirements changes slightly when the receive latency increases or decreases. For example, for the message size of 4000 bytes, the required buffer size is 12052 bytes when  $\alpha = 9\mu s$  and 11233 bytes when  $\alpha = 0\mu s$ . Hence, the startup cost has very little impact on buffer size requirements. As the receive latency becomes very large, the buffer size increases toward a buffer that can hold all of the messages, as expected.

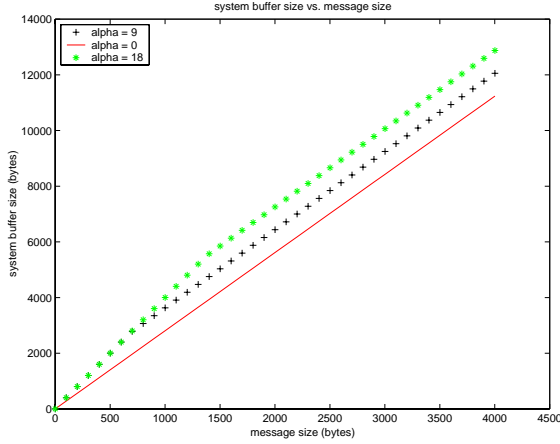


Figure 4: Varying the Receive Latency

### 5.3 Increasing the Network Bandwidth

Figure 5 and 6 show what happens when we increase the network bandwidth ( $\lambda$ ) such that the message incoming rate is twice the value as before. For the message size of 4000 bytes and  $\delta = c/8$ , the buffer size is 12052 bytes for  $\lambda$  and 14436 bytes for  $2\lambda$ . The buffer size is increased by a factor of 1.2 for a 2x increase in bandwidth. For  $\delta = 2c$ , and the message size of 4000 bytes, the buffer size is 1352 bytes for  $\lambda$  and 6324 bytes for  $2\lambda$ . The buffer size is increased by a factor of 4.7 for doubling of the network bandwidth. Hence, the increase in network bandwidth can have a significant impact on buffer size requirements for large delays between messages.

### 5.4 Increasing the Receive Copy Rate

Next we explore the impact on buffer size requirements when the receiver memory copy rate is doubled. In Figure 7, for message size of 4000 bytes,  $\delta = 1/8c$ , the buffer size is 12052 bytes for  $\mu = 91MB/sec$  and 10109 bytes for  $\mu = 182MB/sec$ . There is a small decrease in the buffer size for doubling of the receive copy rate. Further, for messages smaller than 2500 bytes,

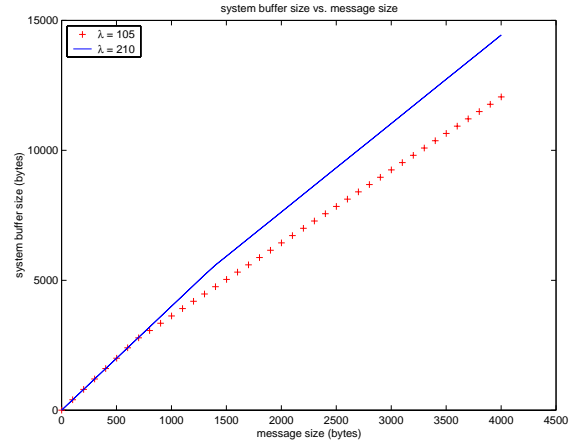


Figure 5: Increasing the Network Bandwidth With  $\delta = c/8$

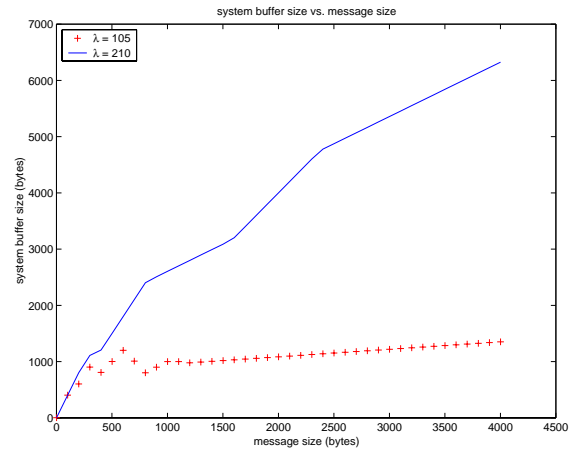


Figure 6: Increasing the Network Bandwidth With  $\delta = 2c$

there is very little decrease in buffer size for doubling the receive copy rate.

### 5.5 Increasing the Number of Messages

In this section, we show what happens when the number of incoming messages is increased to six corresponding to the class of 3-D finite element and finite difference problems. The remaining parameters are unchanged,  $\delta = c/8$ ,  $\lambda = 105MB/sec$ , and  $\mu = 91MB/sec$ . The results are given in Figure 8. For a message size of 4000 bytes, the buffer size is 12052 bytes for 4 messages and 20000 bytes for 6 messages. The buffer size is increased by a factor of 1.65. However, when the delay is  $\delta = 2c$ , which is given in Figure 9, the buffer sizes remain fixed for message

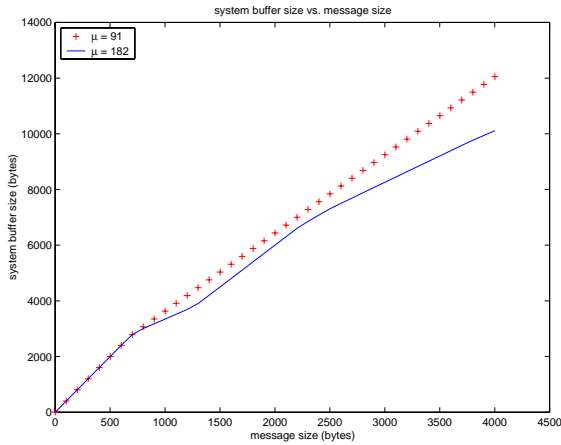


Figure 7: Increasing the Receive Copy Rate

sizes larger than 1200 bytes. Hence, for large delays the buffer size is independent of the number of messages, as expected. For the case of small delays, in the range of  $c/8$  to  $c/4$ , the results show that the buffer size is almost linearly dependent on the number of messages.

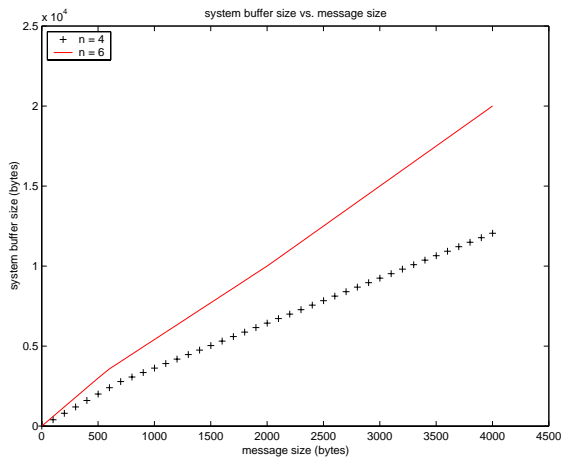


Figure 8: Increasing the Number of Messages When  $\delta = c/8$

## 6 Summary

In this paper we presented an analytical model for analyzing the system buffer size requirements for a typical communication pattern found in widely used scientific parallel applications, such as finite element and finite difference. We focused on message size in the range of 4 KB or smaller, corresponding to the

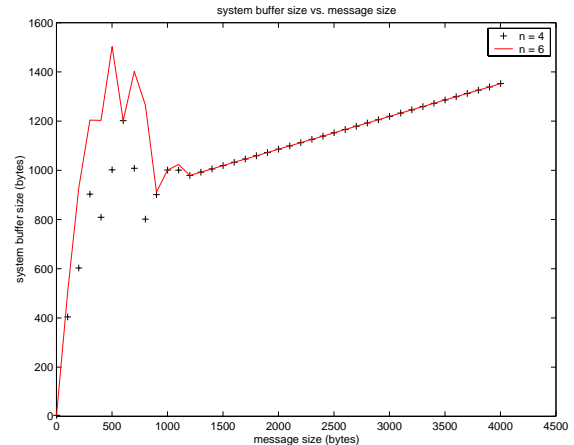


Figure 9: Increasing the Number of Messages When  $\delta = 2c$

eager protocol, which is the focus of this paper. This model was validated for the IBM SP2, for which we used a buffer size that was three orders of magnitude smaller than the default value. Our simulation program is available for others to use to estimate buffer size requirements for systems that allow user-specified buffer sizes.

We used our simulator, based upon our analytical model, to perform a sensitivity analysis for the buffer size requirements when the following features are varied: (1) delay between message arrivals, (2) receive latency, (3) network bandwidth, (4) receive copy rate and (5) number of messages received.

The results indicate the following interesting trends:

- The delay between messages significantly affects the buffer size. When the delay is beyond half of the time required for the message to be completely placed into the system buffer, there is a significant decrease in the buffer size requirement. For example, when  $\delta = c/4$ , the buffer size of four 4000 bytes messages is 11751 bytes, and it is 8971 bytes when  $\delta = c/2$ .
- The receive copy rate has very little impact on buffer size. For example, for  $\delta = c/8$ , when  $\mu = 91MB/sec$ , the buffer size of four 4000 bytes is 12052 bytes. It is 10109 bytes when  $\mu = 182MB/sec$ .
- The increase in network bandwidth can significantly impact the buffer size especially for large delays between message arrivals.
- As expected, however, the buffer size varies close to linearly with the number of messages. Further,

as the receive startup latency increases, the buffer size increases slightly toward a buffer size equals to that required to hold all of the messages. As the value decreases, the buffer decreases slightly.

These results demonstrate that for future system designs, the system buffer size issue should be revisited when the network bandwidth is increased. While the changes in the other two factors (receive startup latency and receive memory copy rate) do not require a new design of the system buffer size.

## References

- [1] H. Franke, E. Wu, P. Pattnaik, and M. Snir. MPI Programming Environment for IBM SP1/SP2, 15th International Conference on Distributed Computing Systems, Vancouver, British Columbia, Canada, 6/95.
- [2] H. Franke, P. Hochschild, P. Pattnaik, J.P. Prost, and M. Snir. MPI-F: An MPI Prototype Implementation on IBM-SP1, 1994 Scalable High Performance Computing Conference, Knoxville, May 1995.
- [3] SGI. *Message Passing Toolkit: MPI Programmer's Manual*. <http://techpubs.sgi.com/library>.
- [4] Message Passing Interface Forum. *MPI: A Message Passing Interface Standard*. <http://www.unix.mcs.anl.gov/mpi/>.
- [5] CELERO Research Group. *CELERO Web Site*. <http://www.ece.nwu.edu/taylor/CELERO>.
- [6] IBM. *IBM Parallel Environment for AIX (PE)*, third edition, October 1998. <http://www.rs6000.ibm.com>.
- [7] William G. Tuel Jr. Message Passing on the IBM POWERparallel SP2. *Proceedings of MPI Developers Conference at Notre Dame*, June 1995.
- [8] M. Snir, P. Hochschild, D.D. Frye, and K.J. Gildea. The Communication Software and Parallel Environment of the IBM SP2. *IBM System Journal*, 34(2):205–221, 1995.
- [9] N. Nupairoj and L.M. Ni. Performance Evaluation of Some MPI Implementations on Workstation Clusters. *Scalable Parallel Library Conference'94*, pages 98–105, Oct 1994.
- [10] N. Nupairoj and L.M. Ni. Benchmarking of Multicast Communication Services. Technical Report MSU-CPS-ACS-003, Michigan State Univ., 1995.
- [11] J. N. Reddy. *An Introduction to The Finite Element Method*. McGraw-Hill, second edition, 1984.
- [12] Michael Resch, Holger Berger, Thomas Boenisch, and Dirk Sihling. *Performance of MPI on a Cray T3E-512*. Third European CRAY-SGI MPP Workshop, PARIS (France), Sept. 11 and 12, 1997
- [13] E. Schwabe, V. Taylor, and M. Hribar. *An In-depth Analysis of the Communication Costs of Parallel Finite Element Applications*. Technical Report, Northwestern University, November 1995.
- [14] Jean Walrand. *Communication Networks*. McGraw-Hill, second edition, 1998.
- [15] G. Yagawa, A. Yoshioka, S Yoshimura, and N. Soneda. Parallel Finite Element Method with a Super Computer Network. *Computers and Structures*, 47(3):407–418, May 1993.