IEEE Symposium on High Performance Computer Architecture
January 1999, Orlando, Florida.

Communication Studies of Single-threaded and Multithreaded Distributed-Memory Machines

(A Short Summary)

Andrew Sohn, Yunheung Paek, Jui-Yuan Ku
Computer Information Science Department
New Jersey Institute of Technology
{sohn,paek,jku}@cis.njit.edu

Yuetsu Kodama, Yoshinori Yamaguchi
Electrotechnical Laboratory
1-1-4 Umezono, Tsukuba-shi, Ibaraki 305, Japan
{kodama,yamaguti}@etl.go.jp

Abstract

This report explicates the communication overlapping capabilities of three distributed-memory machines, SGI/Cray T3E, IBM SP-2 with wide nodes, and the ETL EM-X. Bitonic sorting and Fast Fourier Transform are selected for experiments. Various message sizes are used to determine when, where, how much and why the overlapping takes place. Experimental results with up to 64 processors indicated that the communication performance of EM-X is insensitive to various message sizes while SP-2 is the most sensitive. T3E stayed in between. The EM-X gave the highest communication overlapping capability while T3E did the lowest. The experimental results are compared with the analytical results based on LogP and LogGP communication models.

1 Background, Experimental Settings and Results

Distributed-Memory Multiprocessors (DMMs) have been regarded as a viable architecture of scalable and economical design in building large parallel machines to meet the ever-increasing demand for high performance computing. The major problem which hinders the performance of distributed-memory machines is remote memory latency. DMMs distribute data in a way that there is no overlapping or copying of major data. Typical distributed-memory machines incur a lot of latency, ranging from a few to tens of microseconds for a single remote read operation. The gap between processor cycle and remote memory access time becomes wider, as the processor technology is advanced using rigorous instruction level parallelism.

Numerous approaches have been taken to study the communication issues of DMMs. Distributed-memory machines such as T3E [9], AP1000+, and SP-2 [1] provide some hardware support to overlap computation with communication in a way to take some of the burden from the main processor. Multithreaded machines EM-4 [8], EM-X [7] and Tera [5] aim at tolerating remote memory latency through split-phase read mechanism and context switch [6]. It is the purpose of this report to investigate the overlapping capabilities of DMMs. Specifically, we identify when, where, how much and why overlapping takes place.

Three DMMs are used in our experiments, including IBM SP-2 with wide nodes, SGI/Cray T3E, and the laboratory prototype EM-X. The multithreading capability of EM-X was turned off in this study by using only one thread. Instead, the remote by-passing mechanism has been used to be fair with the others. The multithreading capabilities of EM-X are described in [10]. The two benchmark problems, fine-grain bitonic sorting [3] and FFT, have been implemented on the three machines. The SP-2 and T3E versions are written in C with Message-Passing Interface (MPI). The EM-X version is written in C with a thread library, which can run only on the EM-4 and EM-X multithreaded machines. Details of the overlapping versions of bitonic sorting and FFT are described in [11].

The terms elements and integers are used interchangeably throughout this paper, as are segments and messages. The unit for sorting is *integers* while that for FFT is *points*. An integer is 4 bytes in EM-X and SP-2 and 8 bytes in T3E. A point consists of real and imaginary parts, each of which is 4 bytes in EM-X and SP-2 and 8 bytes in T3E. The following lists the parameters used in this study:

- P = the number of processors, up to 64.
- n = the total number of data elements, up to 8M (the maximum size which can be run on EM-X).
- s = the number of segments per processor
- $m = n/sP$ = message size

Absolute communication times are plotted in Figure 1. We find from the plots that SP2 is very sensitive to message size, favoring message sizes of 512 to 2K elements. On the other hand, EM-X is less sensitive to message size. T3E generally stays in between. In particular, the machine is very efficient for large-sized messages.

Bitonic sorting shows irregular and often higher communication time than FFT. It forms a valley in the performance curve for EM-X, so does FFT for SP2. On the other hand, the T3E communication times continuously decrease as the segment size increases. Recall that FFT has to send twice as many messages because each element has real and imaginary parts. Hence, the FFT communication times should be halved to make a fair comparison with sorting.

In general, increasing the number of processors increases the communication time for these machines when the data size for each processor is fixed. When the number of processors is increased to 64, SP2 and T3E give consistent behavior over the two problems. EM-X, on the other hand, showed there is little change in the communication time when the number of processors is increased to 64.

2 Comparison with Communication Models

The LogP [4] and LogGP models [2] are designed to capture the communication behaviors of distributed-memory machines. The LogP model defines four parameters, L for latency, o for overhead, g for gap between messages, and P for processors for *short* messages. The LogGP model captures the communication behavior of *long* messages by adding G, where G is the gap between bytes of the same message. Using the five terms, L, o, g, G, and P of the LogGP model, the communication time for a single *long* message with k bytes on a single processor can be defined as $t_{1,1} = o + (k-1)G + L + o$. The communication time for sending *two* consecutive messages with message size k_1 and k_2 on a processor will be $t_{1,2} = o + (k_1 - 1)G + g + (k_2 - 1)G + L + o$. Thus, the communication time for sending s consecutive messages on a single processor will be

$$t_{comm} = 2o + L + (s-1)g + \sum_{i=1}^{s} (k_i - 1)G$$

Let b be the number of bytes per word. Since the message sizes are the same for both benchmarks, we have, $k_1 = k_2 = ... = k_s = mb$. The above formulation for sending s consecutive messages of size m words reduces to

$$t_{\mathrm{comm}} = t_{1,s} = 2o + L + (s-1)g + (mb-1)Gs \tag{1}$$

where b is set to 4 for EM-X and SP2 and 8 for T3E. To compare the analytical results with the experimental results shown in Figure 1, let us consider the case with bitonic sorting on SP-2 for P=64 and n=1M, where we have identified that the maximum communication time occurs when the segment (message) size m is 4 words and the number of messages (segments) is $s = n/P*m = 1M/64*4 = 4096$ segments. Using Eq. (1), we find the maximum communication time, t_{max}, for s=4096, and m=4 as

$$t_{max} = 2o + L + (4096 - 1)g + (4*4 - 1)G*4096$$
$$= 2o + L + 4095g + 61440G \quad (2)$$

G is typically tens to hundreds of nano seconds whereas g is several tens of microseconds. Furthermore, o, L and g are usually on the similar order. We assume $G = 0.001g$ [2,4]. Thus, Eq. (2) reduces to

$$t_{max} = 2g + g + 4095g + 61g \approx 4159g \quad (3)$$

Note from Figure 1 that the minimum communication time for bitonic sorting with P=64 and n=1M occurred when the message size m is 4K words and the number of messages (segments) is $s = n/P*m$ = 1M / 64*4K = 4 segments. Using Eq. (1) and the above assumptions on g and G, we approximate

$$t_{min} = 2o + L + (4 - 1)g + (4*4096 - 1)G*4$$
$$= 2o + L + 3g + 65532G = 2g + g + 3g + 66g \approx 72g \quad (4)$$

Taking the ratio of (3) to (4), we find $t_{max}/t_{min} = 4159g/72g =$ **58**. The ratio of t_{max} to t_{min} is determined essentially by the number of segments. Note from Table 1 that t_{max} =8.335 sec and t_{min} =0.132 sec, as highlighted. The empirical ratio is thus **63**. This analytical result closely matches the experimental result for sorting on SP-2.

For the FFT results with m = 4 and 1K, the empirical ratio is 4.370/0.084 = **52** (Table 2). Using the analytical model, we have $t_{max}/t_{min} = 4159g/83g =$ **50** which is also close to the empirical result. From these results, we find that the analytical models can reasonably accurately predict the SP-2 communication times. The empirical results for EM-X and T3E, however, do not agree with the analytical results, as discussed below.

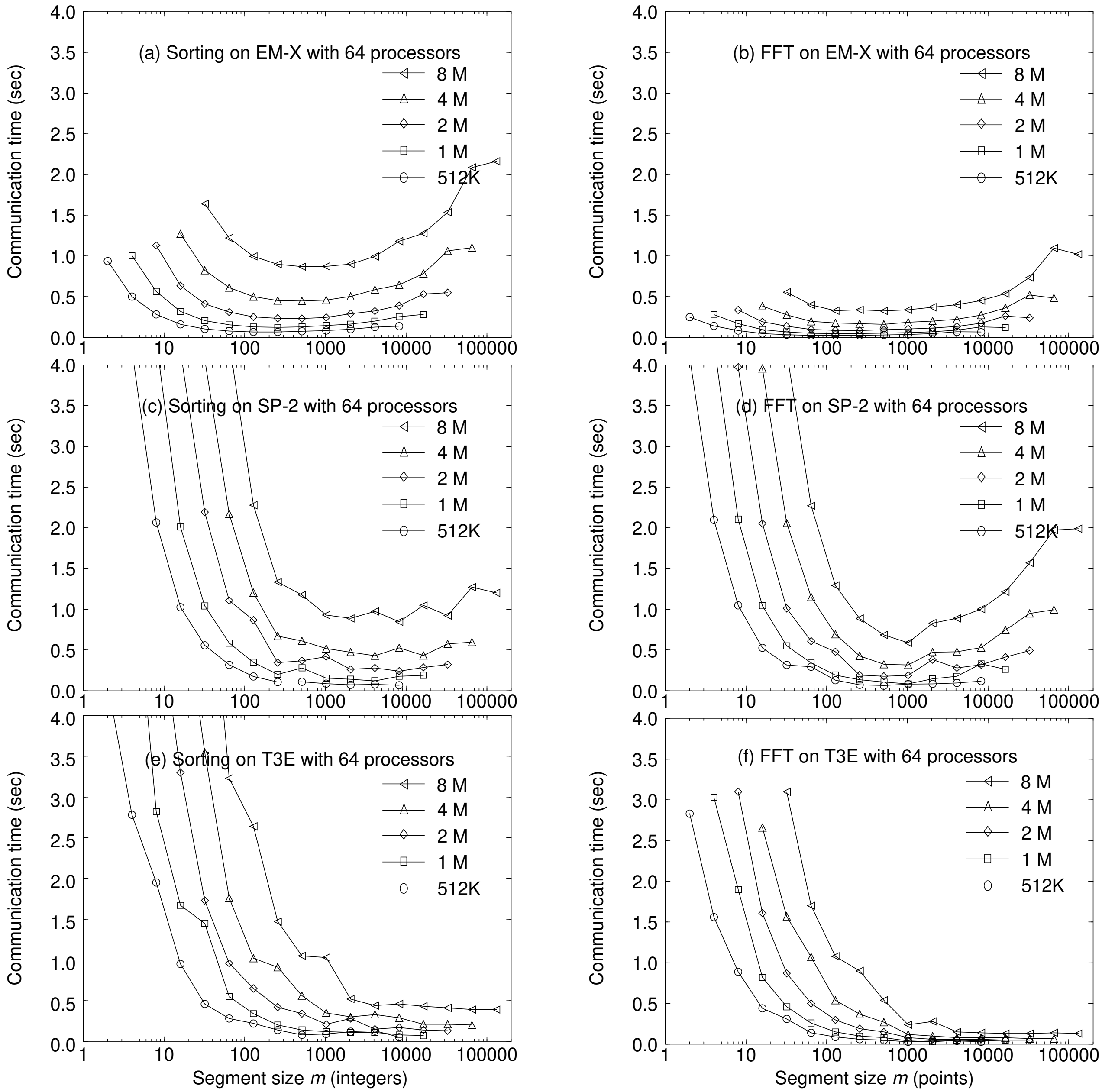

Figure 1: Communication times in seconds on 64 processors.

# of seg s	seg size m	EM-X			SP-2			T3E		
		comp	comm	ratio	comp	comm	ratio	comp	comm	ratio
4096	4	0.441	**1.005**	0.44	0.277	**8.335**	0.03	0.131	**5.134**	0.026
2048	8	0.387	0.565	0.69	0.184	4.299	0.04	0.100	2.824	0.035
1024	16	0.360	0.318	1.13	0.148	2.010	0.07	0.085	1.669	0.051
512	32	0.347	0.206	1.68	0.121	1.041	0.12	0.079	1.451	0.054
256	64	0.340	0.155	2.19	0.112	0.585	0.19	0.075	0.564	0.137
128	128	0.337	0.129	2.61	0.103	0.349	0.30	0.072	0.345	0.209
64	256	0.336	**0.124**	2.71	0.100	0.201	0.50	0.072	0.204	0.353
32	512	0.336	0.130	2.58	0.102	0.281	0.36	0.070	0.145	0.483
16	1K	0.338	0.147	2.30	0.097	0.154	0.63	0.070	0.116	0.603
8	2K	0.344	0.163	2.11	0.096	0.142	0.68	0.070	0.114	0.614
4	4K	0.352	0.200	1.76	0.095	**0.132**	0.72	0.069	0.113	0.611
2	8K	0.374	0.255	1.47	0.095	0.179	0.53	0.070	0.080	0.875
1	16K	0.415	0.282	1.47	0.094	0.190	0.49	0.069	**0.079**	0.873

Table 1: Sample execution times(sec) for bitonic sorting with P=64, n=1M. The highlighted entries represent the t_{min} and t_{max}.

# of seg s	seg size m	EM-X			SP-2			T3E		
		comp	comm	ratio	comp	comm	ratio	comp	comm	ratio
4096	4	1.680	**0.377**	4.46	2.540	**4.370**	0.58	0.271	**3.029**	0.089
2048	8	1.589	0.218	7.29	1.894	2.229	0.85	0.234	1.899	0.123
1024	16	1.553	0.114	13.6	1.058	1.098	0.96	0.218	0.819	0.266
512	32	1.523	0.079	19.3	0.651	0.563	1.16	0.213	0.456	0.467
256	64	1.513	0.058	26.1	0.464	0.311	1.49	0.209	0.262	0.798
128	128	1.508	0.048	31.4	0.390	0.196	1.99	0.207	0.154	1.344
64	256	1.508	**0.047**	32.1	0.323	0.120	2.69	0.207	0.101	2.050
32	512	1.506	0.049	30.1	0.280	0.092	3.04	0.206	0.075	2.747
16	1K	1.504	0.057	26.4	0.265	**0.084**	3.16	0.214	0.041	5.220
8	2K	1.504	0.070	21.5	0.298	0.135	2.21	0.214	**0.040**	5.350
4	4K	1.503	0.092	16.3	0.308	0.134	2.28	0.214	0.048	4.458
2	8K	1.504	0.131	11.5	0.356	0.187	1.90	0.213	0.045	4.733
1	16K	1.513	0.120	12.6	0.393	0.231	1.70	0.213	0.050	4.260

Table 2: Sample execution times(sec) for FFT with P=64, n=1M.

In EM-X, for sorting with P=64 and n=1M, the maximum and minimum communication times for EM-X are 1.005 seconds and 0.124 seconds, resulting in the ratio of **8.1**. For FFT, the ratio is 0.377/0.047 = **8.0**. Unlike SP-2, we can see that the communication behavior of EM-X is highly stable since the ratios are essentially the same across the two different problems. However, these ratios do not match with the analytical ratio of 4222/134 = **32**. The T3E results are also different from the EM-X and SP-2 results. The empirical ratios are **64** and **75**, respectively for sorting and FFT. However, the analytical ratios are **32** and **30**. The T3E and EM-X results do not agree with the analytical results.

There are several reasons for the discrepancy. First, our assumptions on *L, o, g* and *G* were approximations. It is thus likely that these approximations are not accurate for EM-X and T3E. Indeed, EM-X is a fine-grain machine which communicates with fixed-sized packets. Therefore, the assumptions that *o, g* and *L* are on the similar order may not be accurate for EM-X.

Second, the parallel programming paradigms used in the machines are different. EM-X employs *one-sided* communication paradigm based on remote read and write while SP-2 and T3E used *two-sided* communication based on send and receive. The former does not require synchronization while the latter does [11]. Since the overhead associated with one-sided communication is substantially smaller because of no synchronization requirement, the ratios for EM-X were much smaller than those for SP-2 and T3E. The ratios for T3E were larger than the analytical ones and the largest among the three machines. We found that these results are amplified due to the fact that the MPI implementation incurs very high overhead, almost 10 times the SHMEM one [12]. This high overhead offset the gain by the T3E's high-performance network.

Third, the analytical models do not take into consideration the architectural features for c*ommunication-computation overlapping,* such as communication co-processors, remote by-passing mechanisms, and in turn network contention. EM-X uses a remote by-passing mechanism which does not consume the processor cycles. This misprediction of the analytical models on communication behavior leads us to further investigate the overlapping capabilities embedded in these three machines.

3 Communication Efficiency

Figure 2 identifies the efficiency of overlapping for the three machines. Overlapping efficiency is defined as $(T_{\text{comm,1}} - T_{\text{comm,s}}) / T_{\text{comm,1}}$, where $T_{\text{comm,s}}$ is the communication time for s segments. The results indicate that EM-X gave the best communication overlapping capability while T3E did the worst. SP-2 remained in between. To further illustrate these results, we classified the communication times into overhead, barrier synchronization, and latency. *Overhead* refers to the time taken to execute instructions for message preparation. This overhead is fixed and cannot be overlapped with computation since the instructions must be executed to send out packets. *Barrier synchronization* is not part of the original programs but only used at the end of each iteration to synchronize processors and measure various timings. *Latency* is the main target for overlapping. Figure 3 shows the distribution of the communication times for P=64, n=8 M.

The reason EM-X gave the best performance, despite the fact that the EM-X multithreading capability was turned off, is because of the remote by-passing mechanism coupled with fine-grain fixed-sized packet communication. While other machines use the main processor cycles for communication, EM-X does not. Figure 3 shows that the EM-X curves are essentially flat compared to others. The reason is EM-X treats messages the same in the low execution level, regardless of their sizes, except when the message is less than 32 words. When a message of *n* words is sent (remote write), the message is broken into *n* words. Each word is joined by the destination processor number and the target memory address to make an independent packet. Therefore, the difference between a few large messages and many small messages is the number of instructions in the source code level. This is precisely why the overhead curve for EM-X is flat while the overhead for T3E and SP2 forms a valley.

The minimum communication of SP-2 occurs when the message size is 512 to 2K elements. This is obvious because each processor in SP-2 has a 4KB communication buffer. Figure 3 further supports this observation as the sum of the three components are fairly equally weighted in that window. Outside the window, however, SP-2 shows poor communication performance. The reason for the poor performance when the message size is small is because the overhead now becomes the deciding factor. The machine needs to incur very large overhead to generate many small messages. For each small message, some instructions must be executed for message preparation. Overhead is a fixed cost and is not overlapped with computation. When the message size is very large, the overlapping efficiency in SP-2 also drastically reduced because of the small buffer. While there is certainly little overhead, this large-sized messages will occupy the communication channels longer to complete sending/receiving each message, resulting in clogging the bandwidth.

T3E gives essentially no overlapping regardless of the message size. Unlike the other machines, the overall communication time continues to drop even for the largest segment of 1 M words. When the message size is small, the overhead is the dominant factor as in

SP-2. However, for large messages, the latency as well as the overhead continues to drop. The main reason for this little overlapping is its high-throughput network with peak performance of 1.2GB/sec that allows processors to send even fairly large messages with little delay. The plots indicate that T3E gives the lowest latency for large messages among the three because of its high-performance network and large numbers of E-registers dedicated to hide remote memory latency. We believe that the high communication overhead coupled with low latency provides little room to optimize the overall communication time through overlapping.

When the two problems are cross-compared, we find that sorting shows relatively poor overlapping compared to FFT for all three machines. This is due to the *ratio* of computation to communication. The computation time in theory must be at least the same as or larger than the communication time to effectively mask off the communication time. As we have seen in Table 1, the computation-communication ratios for sorting are on the average 1.5 for EM-X, 0.5 for SP-2, and 0.5 for T3E. These small ratios clearly indicate that there was not enough computation to mask off the communication. However, when the ratios for FFT were increased to 15 (EM-X), 2 (SP-2), and 2 (T3E), the amount of overlapping has improved as the plots indicate. SP-2 has now reached nearly 70% while T3E finally shows a sign of improvement. This increase in overlapping is because the machines can now find the time to mask off some of the communication time.

4 Conclusions

Reducing communication times is key to obtaining high performance on distributed-memory machines (DMMs). This report has examined the communication overlapping capabilities of three

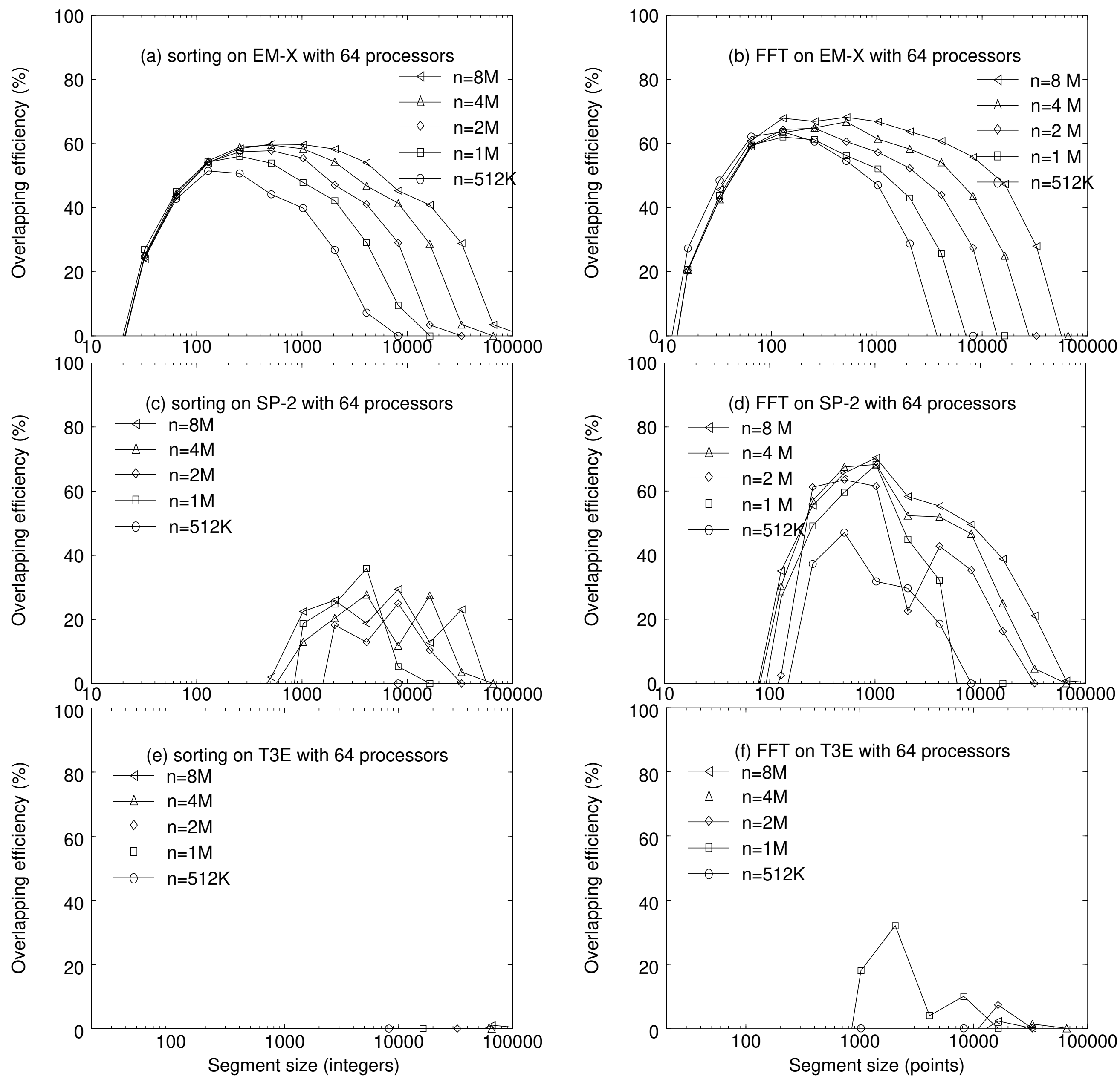

Figure 2: Efficiency of overlapping for bitonic sorting and FFT on three distributed-memory machines.

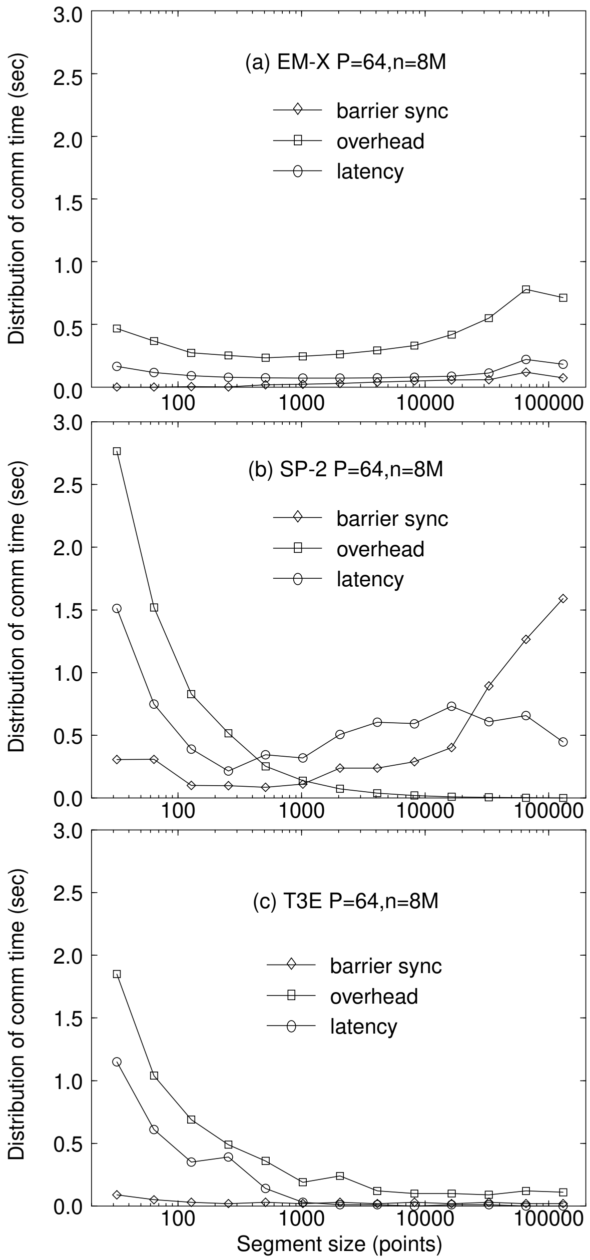

Figure 3: Distribution of communication time for FFT.

DMMs, SGI/Cray T3E, IBM SP-2 and ETL EM-X. Two benchmark problems, bitonic sorting and Fast Fourier Transform, have been selected for experiments. Both problems have been implemented in C with MPI on SP-2 and T3E and with a thread library on EM-X.

Experimental results have indicated that EM-X has showed almost insensitivity to message size because of the remote by-passing mechanism. SP-2 has shown high sensitivity to message size because of the communication buffer of size 4KB. The communication time for T3E has continuously dropped as the message size was increased, showing no size preference, due mainly to the high throughput network and a large of number of external registers.

We have found that the empirical results generally do not agree with the analytical results obtained using the LogP and LogGP models. The reasons include (1) the parallel programming paradigm used in each machine, in particular two-sided versus one-sided communication, (2) the approximation of communication parameters, (3) the lack of incorporating the overlapping capabilities in the analytical models, and (4) the runtime dependent network contention.

Bitonic sorting has been found more difficult to reduce communication time than FFT. The communication times for FFT on SP-2 and EM-X have been reduced to 70% by overlapping. However, communication times for sorting have been reduced to 30% to 60%. The discrepancy is due to the large difference in the ratios of computation to communication. The FFT ratio is almost 10 times larger than the sorting ratio. The performance of T3E, however, has shown very limited, essentially none. This negligible performance has stemmed from the high overhead associated with the MPI implementation. A SHMEM implementation will improve the communication performance at the expense of portability.

Acknowledgments

This work is supported in part by NSF INT-9722545 and NSF INT-9722187. This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Energy Research of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098. Y. Paek is supported in part by Cray Research Inc. and NJIT SBR Program. The T3Es in the Lawrence Berkeley Laboratory and Cray Research Laboratory at Eagan, were used in this work. The SP-2 machines formerly at the NASA Ames and Langley Research Center were used to perform part of the experiments. The authors thank the other EM-X members, Mitsuhisa Sato, Hirofumi Sakane, and Hayato Yamana of the Electrotechincal Laboratory of Japan for various discussions. Andrew Sohn thanks Horst Simon of NERSC, LBL, Joseph Oliger of RIACS at NASA Ames, and Doug Sakal of MRJ Technology Solutions at NASA Ames for their support. The authors thank the referees for their helpful comments.

References

1. T. Agerwala, J. Martin, J. Mirza, D. Sadler, D. Dias, and M. Snir, SP-2 system architecture, *IBM Systems Journal 34*, 1995.
2. A. Alexandrov, M. Ionescu, K. Schauser, and C. Scheiman, LogGP: Incorporating long messages into the LogP model - One step closer towards a realistic model for parallel computation, *ACM SPAA'95*, July 1995, pp. 95-105.
3. K. Batcher, Sorting networks and their applications, in *Proc. the AFIPS Spring Joint Computer Conf. 32*, 1968, pp.307-314.
4. D. Culler, R. Karp, D. Patterson, A. Sahay, K. Schauser, E. Santos, R. Subramonian, and T. von Eicken, LogP: Towards a realistic model of parallel computation, *PPoPP'93*, May 1993.
5. J. Feo and P. Briggs, Tera Programming Workshop, *IEEE/ACM PACT'98*, Paris, France, October 1998.
6. G. Gao, L. Bic and J-L. Gaudiot (Eds.) Advanced topic in data-flow computing and multithreading, IEEE CS Press, 1995.
7. Y. Kodama, H. Sakane, M. Sato, H. Yamana, S. Sakai, and Y. Yamaguchi, The EM-X parallel computer: architecture and basic performance, *ACM ISCA'95*, June 1995, pp.14-23.
8. M. Sato, Y. Kodama, S. Sakai, Y. Yamaguchi, and Y. Koumura, Thread-based programming for the EM-4 hybrid data-flow machine, *ACM ISCA'92*, May 1992, pp.146-155.
9. S. Scott and G. Thorson, The Cray T3E network: adaptive routing in a high performance 3D-Torus, *HOT Interconnects*, 1996.
10. A. Sohn, Y. Kodama, J. Ku, M. Sato, H. Sakane, H. Yamana, S. Sakai, and Y. Yamaguchi, Fine-grain multithreading with the EM-X multiprocessor, *ACM SPAA'97*, pp.189-198.
11. A. Sohn, Y. Paek, J. Ku, Y. Kodama, and Y. Yamaguchi, Communication studies of distributed-memory multiprocessors, NJIT CIS TR 98-3, May 1998. http://www.cs.njit.edu/sohn.
12. T. Welcome, Introduction to the Cray T3E programming environment, http://www.nersc.gov/training/T3E/intro9.html.