Randomized Sorting on Optically Interconnected Parallel Computer

G.Bhattacharya[*], J.Chrostowski[†], F.Dehne[*], P.Palacharla[†]
[*]SchoolofComputerScience, CarletonUniversity
Ottawa, Ontario, Canada K1S 5B6
[†]InstituteforInformationTechnology, NationalResearchCouncil
Ottawa, Ontario, Canada K1A 0R6

December 12, 1995

Abstract

In this paper we present an efficient randomized sorting algorithm for a multiprocessor computer which uses all-to-all broadcast free-space optical interconnects. This algorithm has a better time complexity compared to other sorting algorithms utilizing optical processing that have been proposed in the existing literature. The present algorithm has better performance compared to similar randomized sorting algorithms on electrically interconnected mutiprocessors, due to the higher bandwidths and parallelism of optical interconnections.

1 Introduction

High bandwidth and low latency are the requirements for interconnection networks for high performance computers. As individual processor data rates and complexities grow, electrical interconnects impose a communications bottleneck since the bandwidth of each link and the physical distance these links can cover are limited by power dissipation and electrical crosstalk[11, 12, 5, 8, 9]. The result is a limitation in practical increases in the clock speed and the number of processing nodes in multiprocessor systems. Optical interconnects have the potential to alleviate some of the bottlenecks imposed by electrical interconnects. This is possible by combining the extremely high-bandwidth and parallelism of optics with the logic and buffering capabilities of electronics to produce higher performance interconnection networks[11, 12, 5, 8, 9]. These systems could potentially scale to a large number of processing and memory nodes over relatively large distances.

Interprocessor communication multiprocessor systems consists of short status reports, brief memory requests and large data transfers (e.g. global data). The interconnect operates over a wide range of message lengths, from a single word to a large block of data. This communication can be responsible for a large percentage of the total execution time of an application. A significant portion of communication latency associated with the interconnect network is the routing latency which can be orders of magnitude larger than memory access time. Thus, the speed-up that could be attained if network latency were reduced to a typical memory access would be enormous. In addition, as the applications executed on these systems may require diverse communication patterns between processors, the underlying interconnection network has to be flexible so as to fully utilize the benefits of high-speed processing. Multiprocessor systems using optical interconnects, such as the parallel computer described in this paper, can achieve flexibility and reduce latency.

Free-space and guided-wave optical interconnections have been investigated in parallel computers[11, 12, 5, 8, 9, 18, 14, 7, 19, 13]. Recent efforts at Cray research have incorporated fiber-optic interconnects for clock distribution in the Cray T90 computer system and also in IO subsystem designs to enable very high bandwidth (multi-gigabytes/sec) interconnects between multiple systems over distances up to 200 meters[18]. Similar efforts in using optical interconnects in multiprocessors includes a joint Honeywell-Thinking

Machines project using fiber-optic interconnects in a Connection Machine to lower wire density and enhance performance[14]. Intel used fiber-optic interconnects in a Touchstone Supercomputer to achieve the bandwidth of 1.6 Gbps per mesh interface node for scaling from 512 to 1024 nodes[7]. NTT systems laboratories implemented a 64-processor three dimensional mesh topology using board-to-board free space interconnects[19].

In this paper, we present an efficient randomized sorting algorithm for a free-space optical interconnect based multiprocessor system. Sorting is an important symbolic operation, and implementation of diverse algorithms in many different areas of application incorporate sorting in their intermediate steps. Due to the increasing availability of multiprocessor computers, the theoretical community has devoted considerable effort towards designing efficient parallel sorting algorithms[1]. A typical parallel algorithm proceeds by the divide and conquer approach. Initially each processor contains a portion of the list of n elements to be sorted, which each processor sorts sequentially. These sorted subsets are finally merged. Depending upon the number of steps in which merging takes place, parallel sorting algorithms can be either single-step or multi-step. Batcher's bitonic sorting algorithm[2] is an example of a multi-step algorithm.

The bitonic sorting algorithm has been implemented using optoelectronic smart pixel arrays[13, 6, 21]. A smart pixel is an optoelectronic device that combines optical inputs and/or outputs with electronic processing circuitry, and is capable of being integrated into two-dimensional arrays. The two approaches for optical bitonic sorter presented in the literature use the perfect-shuffle holographic (free-space) interconnect but differ in the smart pixel technology[13, 6, 21]. The first approach is based on the self-electrooptic effect device (SEED) modulators and detectors[6, 21]. The second approach uses the vertical cavity surface emitting lasers (VCSEL) and silicon photodetectors [13]. The CMOS electronic processing elements in both approaches perform the bit-oriented compare-and-exchange operation in parallel. These optoelectronic sorters are special-purpose processors. The time complexity is that of bitonic sorting algorithm, i.e. $O((logN)^2)$ where n is the number of elements/keys to be sorted. The SEED based system is aimed at sorting 1024 8-bit words in about 10us on a 32x32 smart pixel array[21].

In the next section, the architecture of the optically interconnected parallel computer is described. The randomized sorting algorithm and its implementation on the parallel computer is presented in Section 3. The algorithm is analyzed in Section 4. Our conclusions are presented in Section 5.

2 Optically Interconnected Parallel Computer

Multiprocessor systems can perform efficient computations on problems which can be parallelized. The particular parallel architecture will determine how efficiently a particular problem can be computed. In recent years, Multiple Instruction Multiple Data (MIMD) systems are being widely used for parallel computing. Most commercial massively parallel processors are based on multidimensional mesh or hypercube interconnection networks. The multiple-broadcast or all-to-all topology in which all outputs are connected directly to all inputs of all the processing elements/nodes as shown in Figure 1 is the most efficient topology resulting in single-hop systems. Unlike systems based on mesh or hypercube architectures, the single-hop systems allow every processor to directly communicate with one another with no intermediate nodes. In addition, the multiple broadcast networks include one-to-many and many-to-many interconnect schemes.

The parallel computer considered in this paper is a MIMD system, where each of the computing nodes consists of a microprocessor (e.g. DEC Alpha, PowerPC) with local memory and interfaces to access the fully broadcast interconnection network. In such a system, the data stored in the output buffers of each individual computing node, are made available to the input buffers of all computing nodes enabling them to retrieve their specific data. In this way each computing node always accesses and executes its own independent stream of instructions, operating on either local data in the input buffer or on the contents of its own memory. Execution of these instructions allows new data to be produced and broadcast to other processors. An example of a MIMD system using the all-to-all broadcast interconnect topology is the Delft Parallel Processor (DPP9X)[8]. The interconnection network can be implemented using optics as described in the following subsection.

An important performance measure for computer interconnects is latency. In the case of architectures with mesh interconnection, the communication latency increases with the size of the network (number of processing nodes), adversely affecting the performance of multiprocessor systems. On the other hand, the latency in a multi-broadcast system interconnection scales better with the number of processing nodes since it eliminates the routing overhead.

This results in a near-linear relationship between the performance of the multiprocessor system and the number of processing nodes, although at the cost of increased interconnection complexity. In the case of N nodes and a P-bit wide computer word the number of links amounts to $N \times N \times P$. So to builda MIMD parallel computer with large number of processing nodes (e.g. 1024) and with each of them producing a 64-bit wide computer word, almost 64 million interconnections are required, it is evident that only optical technology could offer a solution. Not only does optics offer a superior bandwidth to electronic interconnection for typical inter-computing node distances but is unique in its ability to link millions of space-resolved points in parallel and without crosstalk using simple free-space optics such as lens.

2.1 An Optical Broadcast System - The Kaleidoscope

Solutions based on both free-space and optical fiber interconnections are conceivable and the approach taken here involves a partly fiber-wired, partly free-space parallel data distributor. The free-space distribution system, called the Kaleidoscope$^{\mathrm{TM}}$, is built around a compact and cost-effective arrangement of a commercial camera lens together with a plane facet mirror as shown in Figure 2[8, 9, 10]. The distributor enables the composition, distribution and projections of a complete single-broadcast image of $N \times P$ pixels. The data transport from each processor to the distributor is done through a flat-cable of P fibers. Sampling the information of all the processors simultaneously is achieved via a fiber array which contains N flat-cables of P fibers. Creation of multiple images is accomplished by dividing the beams from the $N \times P$ pixel object into separate sections and refocusing each image by means of separate lenses or a multi-facet mirror or prism. In Figure 2, the facet mirror and lens system allows the input optical signals to be copied for distribution to a number of processing elements.

A prototype of the Kaleidoscope was designed and built at Delft University of Technology[8]. Extensive static and dynamic experiments alongwith simulations of the optical characteristics of the Kaleidoscope were performed[8, 10]. The optical distribution system was tested with reconfigurable 2-D spot patterns generated using acousto-optic modulator[17]. The ultimate degree of data parallelism which can be achieved with the Kaleidoscope is limited by the image quality through optical aberration in the lens system. Furthermore, increasing the number of facets on the mirror to broadcast restricts the optical aperture and lowers the imaging resolution. The Kaleidoscope can be stacked to increase the broadcast density by integrating beam-splitters in the first collimation beam with the limitations imposed by size and the optical power budget[8].

3 Description of the Algorithm

Randomized sorting algorithms runs fastest for large sets of input keys, improving significantly the time complexities of deterministic algorithms. The algorithms use a random number generator, but their running time is independent of the input distribution of keys. Their performance depends only on the output of the random number generator. Here we present a variation of the sample sort algorithm that takes into account the all-to-all broadcast capabilities of the optical interconnection scheme.

Assuming n input keys are to sorted on p processors, the algorithm proceeds in three phases. In the first stage an equal number of n/p keys are distributed among the p processors. Each processor then chooses randomly sk pivots out of the n/p keys, with equal probability pks/n where the parameters k and s are called oversampling ratios. Each processor then locally sorts the sk pivots, and selects every k^{th} pivot as a splitter. So every processor now has s splitters. By using the all-to-all broadcast features of the Kaleidoscope$^{\text{TM}}$ these total of ps splitters from the p processors are now broadcast to all the processors. Each processor now locally sorts these ps splitters, using a sequential (deterministic or randomized) sorting scheme. Each processor now locally selects every s^{th} splitter in the sorted set to obtain the final set of p splitters that partitions the total set of input keys. The keys lying between two successive splitters forms a sub-bucket within each processor. In the second stage, each processor locally places all its keys into the appropriate sub-buckets. We denote by S_{ij} the set of keys within the j^{th} sub-bucket of processor i . Each processor i now broadcasts S_{ij} (for all j) to all the processors using the global communication feature of Kaleidoscope$^{\text{TM}}$. In order to minimize the idle time of each processor the sets S_{ij} are broadcast in every round with a cyclic permutation of the indices. In the final stage each processor k collects all the sub-buckets S_{ik} (for all i), to obtain the bucket S_k. The keys are then sorted locally within each processor and the final sorted set of keys are obtained by performing a merge operation on all the buckets.

The reason for performing the double-sampling with two oversampling ratios k and s is to make the number of keys within each sub-bucket, and within each bucket approximately equal. The randomized time-complexity of this algorithm is $\tilde{O}(\log n)$, where the notation $\tilde{O}$ is defined below. $X = \tilde{O}(f(n))$, if and only if for every $c > c_0 > 1$, $\Pr[X \geq cf(n)] \leq n^{-g(c)}$, where c_0 is a fixed constant and $g(c)$ is a polynomial in c with $g(c) \rightarrow \infty$ for $c \rightarrow \infty$.[15, 16]

4 Analysis of the Algorithm

We will now discuss in detail a probabilistic analysis of the above algorithm. We need the following lemma.

Lemma: Consider a random variable X with binomial distribution. Let r be the number of trials, each of which is successful with probabilty q. The expectation of X is $E(X) = rq$. (a) The Chernoff bound on the probability of X exceeding the expectation value, is given by $Pr[X > \gamma rq] \leq e^{-(1-\gamma)^2 rq/2}$ for $\gamma > 1$. (b) Let $\delta \geq 4$ (not necessarily fixed). If $\delta^2 rq \geq \kappa \ln(n)$ for some constant $\kappa > 0$, then $X = \tilde{O}(\delta rq)$.

The first part of the lemma which stipulates a condition for the random sampling technique to be effective, is used to obtain an estimate on the oversampling ratios as a function of n. In the first stage of the algorithm each processor performs a biased random coin flip for each input key stored in it, such that each is selected with probabilty ksp/n. From the lemma it follows that the average random sample size is $\tilde{O}(pks)$ and that if $sk = \Omega(\ln(n)$, then the number of points selected by each processor is $\tilde{O}(ks)$. Since the random sample is stored at each processor, it is necessary that the maximum random sample size is $\tilde{O}(n/p)$. Thus using the above result for the average random sample size, we have the first requirement,

$$ks \leq \frac{n}{p^2} \tag{1}$$

We now prove some probabilistic results on the bounds on the maximum sizes of the buckets and sub-buckets. These results dictate how efficiently the load is balanced in the final stage of the algorithm, since good load balancing requires the bucket (and sub-bucket) sizes to be approximately equal. From these results we determine the bounds on the oversampling ratios k and s in terms of the input key size, n.

Theorem 1: For any $\alpha \geq 1$, the probability that any bucket contains more than $\alpha n/p$ keys is at most $ne^{-(1-1/\alpha)^2 \alpha s/2}$.

Proof: In order to prove that no bucket receives more than $\alpha n/p$ keys, it suffices to show that the distance from any key to the next splitter in the sorted order, is at most $\alpha n/p$. Considering a single key, its distance l to the next splitter is larger than $\alpha n/p$ only if fewer than s of the next $\alpha n/p$ keys in the sorted order are candidates. Let T denote this set of $\alpha n/p$ keys. The candidates in this set are chosen by selecting each input key in S with probability ps/n. Let Y_I be the number of candidates in T that are chosen from S, using the independent method. It can be seen that

$$\Pr[l \geq \alpha n/p] \leq \Pr[Y_I \leq s] \tag{2}$$

We would thus like to have an upper bound on $\Pr[Y_I \leq s]$. It is known that $\Pr[Y_I = k]$ follows a binomial distribution with parameters r and q, where $r = \alpha n/p$ is the number of independent Bernoulli trials, and $q = ps/n$ is the probability of success in each trial. Thus using the above lemma and substituting $r = \alpha n/p$, $q = ps/n$, and $\gamma = 1/\alpha$, we obtain

$$\Pr[Y_I \leq s] \leq e^{-(1-1/\alpha)^2 \alpha s/2}. \tag{3}$$

The probability that the distance from any of the n keys to the next splitter is more than $\alpha n/p$ is given by multiplying the above probability by n, thus yielding the final result.

Theorem 2: For any $\alpha \geq 1$, the probability that any sub-bucket contains more than $\alpha n/p^2$ keys is at most $\frac{n}{p}e^{-(1-1/\alpha)^2 \alpha k/2}$.

Proof : The details of the proof in this case follows exactly the same logical steps as in the proof of Theorem 1. The distance l' of any key in the sub-bucket, to the next splitter, is larger than $\alpha n/p^2$ only if fewer than ks/p of the next $\alpha n/p^2$ keys in the sorted order are candidates. That is

$$\Pr[l' \geq \alpha n/p^2] \leq \Pr[Y_I \leq sk/p] \tag{4}$$

Using the above lemma, but substituting in this case $r = \alpha n/p^2$, $q = skp/n$, and $\gamma = 1/\alpha$ we obtain the result,

$$\Pr[Y_I \leq sk/p] \leq e^{-(1-1/\alpha)^2 \alpha k/2}. \tag{5}$$

The probability that the distance from any of the n/p keys in the sub-bucket to the next splitter is more than $\alpha n/p^2$ is given by multiplying the above probability by n/p.

We now discuss the efficient choices for k and s using the probability bounds proved above. We wish to put an upper bound on the probability that the maximum bucket size is $\alpha n/p$. We choose this bound to be $n^{-\alpha}$. Thus using Theorem 1, we have

$$ne^{-(1-1/\alpha)^2 \alpha s/2} \leq n^{-\alpha} \qquad (6)$$

Choosing $\alpha = 2$, the above implies that

$$s \geq 12\ln(n). \qquad (7)$$

Thus we obtain a lower bound on the oversampling ratio s. Choosing the same value of $\alpha = 2$ and applying a similar analysis to the result proven in Theorem 2, we obtain

$$k \geq 12\ln(n). \qquad (8)$$

Further since each of the ps splitters must be contained within each sub-bucket, we get $\frac{n}{p} \geq ps$. Substituting the bound on s here, we obtain a strict upper bound on the number of processors in terms of the input key size,

$$p \leq \sqrt{\frac{n}{12\ln n}}. \qquad (9)$$

In the table below we present a set of numbers obeying these bounds, which relate the memory size of the processors to the input key size to be sorted. The required local memory size of the processors L, are approximately $8n/p$, which has been obtained from a consideration of the storage requirements for all the intermediate set of splitters in each processor during execution of this algorithm.

5 Conclusions

We have proposed a randomized algorithm for sorting a set of n input keys on an optically interconnected multiprocessor. The time complexity of this algorithm, $\tilde{O}(\log(n))$, is a significant improvement over that of the bitonic sorting algorithm using optical processors that have been proposed in the existing literature. The randomized algorithm we discuss here is a variant of the samplesort algorithm. The algorithm proposed here is specific to the case of fully interconnected network toplogy obtainable using optical interconnections. In contrast to the simple samplesort algorithm, a double sampling of the set of input keys is necessary for this case, in order to have good load balancing in the intermediate stages of the algorithm. From considerations of load balancing, we derive a some restrictions on the oversampling ratios in terms of the input key size.

References

[1] S.G. Akl, *Parallel Sorting algorithms*, Academic Press, Orlando, Florida (1985).

[2] K. Batcher, Sorting networks and their applications, *Proceedings of the AFIPS Spring Joint Computing Conference*, **Vol. 32**, 307-314 (1968).

[3] F.R. Beyette Jr., P.A. Mitkas, S.A. Feld, and C.W. Wilmsen, Bitonic sorting using an optoelectronic recirculating architecture, *Applied Optics*, **33**(35), 8164-8172 (1994).

[4] G.E. Blelloch, C.E. Leiserson, B.M. Maggs, C.G. Plaxton, S.J. Smith and M. Zagha, A comparison of sorting algorithms for the connection machine CM-2. *Proceedings of the 3rd Annual ACM SPAA,* (1991).

[5] W. T. Cathey, S. Ishihara, S-Y. Lee, J. Chrostowski, Optical information processing systems, *IEICE Trans. Fundamentals* **E75-A**, 28-37, 1992.

[6] M.P.Y. Desmulliez, F.A.P. Tooley, J.A.B. Dines, N.L. Grant, D.J. Goodwill, D. Bailie, B.S. Wherrett, P.W. Foulk, S. Aschroft, and P. Black, Perfect-shuffle interconnected bitonic sorter: optoelectronic design, *Applied Optics*, **34**(23), 5077-5090 (Aug. 1995).

[7] B. E. Floren et. al., Optical interconnects in the Touchstone Supercomputer Program, *Proc. SPIE*, **Vol. 1582**, 46-54, (1991).

[8] E.E.E. Frietman, *Opto-electronic processing and networking: a design study*, Delft University of Technology printing office, Delft, The Netherlands (1995).

[9] E. E. E. Frietman, Optics in multiple-instruction multiple- datastream computers, in *Frontiers of Computing Systems Research*, **Vol. 2**, 131-195, editor S. K. Tewksbury, Plenum Press, (1991).

[10] E. E. E. Frietman, L. Dekker, S. A. Boothroyd, P. Palacharla and J.Chrostowski, Optics for multiple-broadcast massively parallel and independently addressable free-space interconnections, *Proc. of the 3rd IEEE International Workshop on Photonic Networks, Components and Applications*, Atlanta, Georgia, (1993).

[11] J. W. Goodman, F. J. Leonberger, S. Y. Kung and R. A. Athale, Optical interconnections for VLSI systems, *Proc. of IEEE*, **Vol. 72** (7), (1994).

[12] A. Guha, J. Bristow, C. Sullivan and A. Hussain, Optical Interconnections for massively parallel architectures, *Applied Optics*, **29**(8), (1990).

[13] L.J. Irakliotis, S.A. Feld, F.R. Beyette Jr., P.A. Mitkas, and C.W. Wilmsen, Optoelectronic parallel processing with surface-emitting lasers and free-space interconnects, *Journal of Lightwave Technology*, **13**(6), 1074-1084, (1995).

[14] B. O. Kahle, E. C. Parish, T. A. Lane and J. A. Quam, Optical interconnects for interprocessor communications in the Connections Machine, *IEEE Conference on Computer Design*, Cambridge, MA, (1989).

[15] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, (1995).

[16] K. Mulmuley, *Randomized Algorithms: An introduction through computational geometry*, Prentice Hall, New York, NY, (1993).

[17] P. Palacharla, S. A. Boothroyd, W. M. Robertson, J. Chrostowski, Generation of reconfigurable interconnects using a two-dimensional acousto-optic deflector, *Applied Optics*, **33**, 2140-2146, (1994).

[18] B. R. Pecor, Optics for Interconnection: Industry's interests and responsibilities - panel discussion, *Proc. of the Second International conference on massively parallel processing using optical interconnections*, San Antonio, TX, pp. 172-173, (1995).

[19] T. Sakano, T. Matsumoto and K. Noguchi, A three-dimensional mesh multiprocessor system using board-to-board free-space optical interconnects: COSINE III, *International Conference on Computer Design*, 278-283, (1993).

[20] C.W. Stirk and R.A. Athale, Sorting with optical compare-and- exchange modules, *Applied Optics*, **27**, 1721-1726 (1988).

[21] A. C. Walker, et. al., Construction of an optoelectronic bitonic sorter based on CMOS/InGaAs smart pixel technology, *Proc. of the Second International conference on massively parallel processing using optical interconnections*, San Antonio, TX, pp. 180-187, (1995).

p	$n_{\min}/p$	$L_{\min} = 8n_{\min}/p$
4	512	4 K
8	1 K	8 K
16	2 K	16 K
32	8 K	64 K
64	16 K	128 K
128	32 K	256 K
256	64 K	512 K
512	128 K	1 M
1 K	256 K	2 M
2 K	512 K	4 M
4 K	2 M	16 M
8 K	4 M	32 M
16 K	8 M	64 M
32 K	16 M	128 M
64 K	32 M	256 M
128 K	64 M	512 M
256 K	128 M	1 G
512 K	256 M	2 G
1 M	512 M	4 G