Bulk Synchronous Parallel Computing Using a High Bandwidth Optical Interconnect

I. Gourlay, P. M. Dew, K. Djemame
School of Computing, University of Leeds, Leeds, LS2 9JT, UK.

Abstract

The list of applications requiring high performance computing resources is constantly growing. The cost of inter-processor communication is critical in determining the performance of massively parallel computing systems for many of these applications. This paper considers the feasibility of a commodity processor-based system which uses a free-space optical interconnect. A novel architecture, based on this technology, is presented. Analytical and simulation results based on an implementation of BSP (Bulk Synchronous Parallelism) are presented, indicating that a significant performance enhancement, over architectures using conventional interconnect technology, is possible.

1: Introduction

A fundamental aspect of parallel computing is scalability and efficiency. Recent focus has been on computational clusters built from low cost commodity components, e.g. Cox et.al. [2]. These systems offer high performance at low cost and are becoming commonplace within the academic community. For such systems, a critical factor in determining overall performance is the speed with which inter-processor communication occurs. This is particularly true for high-bandwidth applications such as data mining and real-time graphics. However, physical limits on electrical interconnects, as indicated by Miller [7] are likely to limit the communication performance of systems based on such technology [9]. Consequently, this paper considers the feasibility of a massively parallel computational cluster of commodity processors with a high bandwidth Free-Space Optical Interconnect (FSOI). In this system, optically encoded data is guided using a series of mirrors and lenses, rather than (e.g.) optical fibre. The architecture is based around a commodity PC cluster (the term PC is used in the paper to signify a commodity processor), with communication occurring via the FSOI as explained in [11]. An additional smart-pixel based layer is added, with the purpose of interfacing between the PCs and the optical interconnect. This layer must be utilised in such a way as to overcome the problem presented by the bandwidth bottleneck in the links to the PCs. It is envisaged that the smart-pixel layer will have relatively simple computational functionality, e.g. to support combining and re-ordering of messages used in the BSP computational model (see section 2). By ensuring that each PC has a unique link to the smart-pixel layer, it is expected that, since the cost of communication between the smart-pixel and PC layers does not increase with the number of processors, this architecture will be scalable with respect to communication cost. Further details regarding the architecture can be found in [11].

The proposed optoelectronic architecture is characterised by a number of key parameters, shown on figure 1. Definitions of these parameters are given in table 1.

The paper is organised as follows. Section 2 briefly introduces the Bulk Synchronous Parallel (BSP) model (see Skillicorn et.al. [8]) and its implementation on the optoelectronic architecture. Analytic expressions are presented, which allow BSP cost parameters to be estimated. Section 3 presents and compares analytic and simulation results, based on the problem of integer sorting. Finally, conclusions and areas of future work are discussed in section 4.

2: Implementing the Bulk Synchronous Parallel (BSP) model on the optoelectronic architecture

The concept of separating communication from computation and sending large amounts of data at once seems well suited to the optoelectronic architecture, since data packets can be combined into much larger messages

0-7695-1573-8/02/$17.00 (C) 2002 IEEE

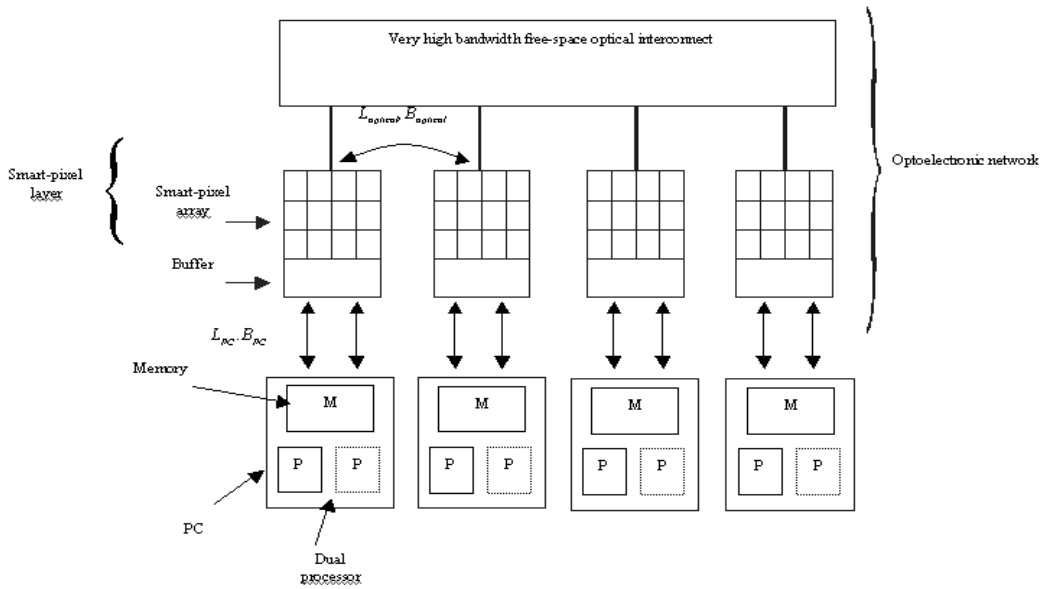

Figure 1: The proposed optoelectronic architecture based on the concept of a computational cluster, with a high speed optical interconnect. The processors in dotted boxes indicate the possibility of using dual processor PCs.

Parameter	Definition
p	Number of processors
sp	Number of flop/s performed by PC
B_{PC}	Number of bits/s that can be communicated between the PC and the smart-pixel layer
L_{PC}	Minimum time required to communicate data between the PC layer and the smart-pixel layer (includes message start-up, time-of-flight, etc.)
$B_{optical}$	Number of bits/s that cam be communicated between smart-pixel arrays (see section 3)
$L_{optical}$	Minimum time required to communicate data between smart-pixel arrays (see section 3)
B_{eff}	Number of bits/s that can be communicated between BSP processors in the optoelectronic implementation of BSP (see section 2)
L_{eff}	Minimum time required to communicated data between BSP processors in the optoelectronic implementation of BSP (see section 2)

Table 1: Definitions of parameters used to characterise the optoelectronic interconnect

before being sent to a processor. Consequently, the BSP computational model is taken as the basis for considering the potential of this architecture.

Section 2.1 introduces the BSP model, followed in section 2.2, by a discussion of the reasoning behind the chosen BSP implementation and the implementation itself. Section 2.3 describes analytic methods for assessing the cost performance of the system architecture, in the context of the model indicated above.

2.1. The BSP model

The BSP model is based on a parallel computer, consisting of a set of processors (each with local memory), a communication network

that delivers messages directly between processors and a mechanism for efficient synchronisation of all or any subset of the processors. A computation on a BSP computer consists of a series of supersteps, each of which involves three phases: Firstly, the processors perform a local computation, i.e. each (or a subset of) the processors perform a computation, using data that is stored in their local memory. This is followed by a communication phase, where each processor sends data to other processors, to be received at the beginning of the next superstep. Finally, a barrier synchronisation takes place; all processors are guaranteed to have received data sent in the previous phase at the end of synchronisation. The structure of BSP computations is appealing, since it fits well with the notion of combining data in to large messages prior to inter-processor communication. The optical bandwidth can only be exploited if large messages are communicated between processors, so that latency does not dominate.

Another significant advantage of the BSP model is the simplicity of the associated cost model, which has been shown to be accurate for a wide range of computations. The cost of a superstep is the sum of three terms, describing local computation (the maximum number of computational steps on a processor, w), barrier synchronisation (l) and communication. The communication term is mgh. Here, g is the time taken to communicate one word under continuous traffic conditions, h is the maximum number of messages sent or received by a processor and m is the maximum number of words in a message. A communication pattern with a given value of h is referred to as an h More details can be found in [6].

Since a significant number of parameters are being used to characterise the system, it would be useful to combine these into smaller, manageable units to simplify the analysis. Consequently the following model is used:

The three-layered optoelectronic system architecture (figure 2) is viewed as equivalent to an architecture A, consisting of a set of processors connected by a simple interconnect, with an effective bandwidth B_{eff} available in communication between a processor pair, under continuous traffic conditions. Similarly, A is characterised by an effective latency L_{eff}, the minimum cost to be paid in any inter-processor communication.

2.2. A data streaming based implementation of the BSP model

In order to utilise the high bandwidth offered by the optoelectronic architecture, it is critical to consider the following two issues. Firstly, due to the multi-layered nature of the architecture, careful consideration of the BSP implementation is required, to ensure that the system does not suffer from prohibitively high latency. Secondly, it is desirable to collect data in the smart-pixel layer and combine it into large messages in order to exploit the optical bandwidth.

Based on the above considerations the following implementation of BSP is proposed. A PC and its corresponding smart-pixel array (and buffer) are viewed as a single BSP processor. Hence, communication between the PC layer and the smart-pixel layer can occur during local computations, while inter-smart-pixel array communication can only occur at the end of a superstep.

It is assumed that there are two channels linking each PC to the smart-pixel layer, so that bi-directional communication can occur simultaneously.

At the beginning of a superstep, data received in the previous superstep resides in the smart-pixel layer. The PC receives only a large enough proportion of the data to begin the local computation, such that it remains busy while more data is arriving. Similarly, data is sent to the smart-pixel layer (where it is stored in the buffers) such that as high a proportion of the data (to be communicated at the end of the superstep) as possible resides in the smart-pixel layer at the end of the local computation.

In order to help to quantify the effectiveness of the data streaming method described above, it is useful to introduce two parameters, r and s:

For a given superstep, r is a lower bound on the fraction of data remaining in the smart-pixel layer associated with any PC after the initial communication between the smart-pixel and PC layers at the beginning of a superstep.

For a given superstep, s is a lower bound on the fraction of data that resides in the smart-pixel layer for any PC, upon completion of the local computation at the end of a superstep.

Note that the buffer storage capacity is assumed to be sufficiently large so that it does not overflow.

It is possible to obtain an analytic expression, allowing the BSP parameter g to be estimated in terms of some of the parameters in table 1.

Recall that g is the cost of communicating a 1-relation under continuous traffic conditions. Notice that this assumes that it costs the same to send h single word messages from a processor as it does to send one message with h words. This is reasonably accurate for large message sizes, but for small messages start-up costs can dominate. This can be incorporated into the model, so that the cost of sending an h-relation of m-sized (i.e. m words) messages is $mg(m)h$. Here (as indicated by Skillicorn et.al. in [8]), the communication throughput is given by,

$$g(m) = \left(\frac{n_{1/2}}{m} + 1 \right) g_\infty \qquad (1)$$

In the above expression, g_∞ is the asymptotic (optimal) communication throughput ratio, reached in the limit of large messages, and $n_{1/2}$ is the message size that would produce half the optimal throughput ratio. The cost, $T_{1-relation}(m)$, of sending a 1-relation of m-sized messages under continuous traffic conditions is given by,

$$T_{1-relation}(m) = L_{eff} + \frac{m}{B_{eff}} \qquad (2)$$

Note that, in Eq. (2) the bandwidth must be expressed in terms of words/second (words/flop when normalised, as required in BSP). The same applies in the equations that follow. From (1) and (2), $n_{1/2} = B_{eff} L_{eff}$ and $g_\infty = 1/B_{eff}$.

The values of B_{eff} and L_{eff} can therefore be estimated by finding the cost of communicating a 1-relation (under continuous traffic conditions). As shown in [11],

$$B_{eff} = \frac{1}{g_\infty} = \frac{1}{\frac{(2-s-r)}{B_{PC}} + \frac{1}{B_{optical}}} \qquad (3)$$

Note that the effective bandwidth depends on s and r, and consequently varies from superstep to superstep in the course of a BSP algorithm. Clearly, the more effective data streaming is, the closer the effective bandwidth is to $B_{optical}$. Consequently, g is algorithm dependent and varies from superstep to superstep within a given algorithm.

$$L_{eff} = \frac{L_{PC}(\sigma(r) + \sigma(s))}{h} + L_{optical} \qquad (4)$$

In Eq. (4), h is included because the cost associated with PC latency is paid only twice, at most, in communicating an h-relation (i.e. the term in $mg(m)h$ associated with PC latency is independent of h).

In addition to g, the barrier synchronisation parameter, l, also involves inter-processor communication. However, barrier synchronisation is not likely to dominate the cost of a well-designed BSP algorithm. Consequently, any performance enhancement as a result of faster barrier synchronisation is ignored.

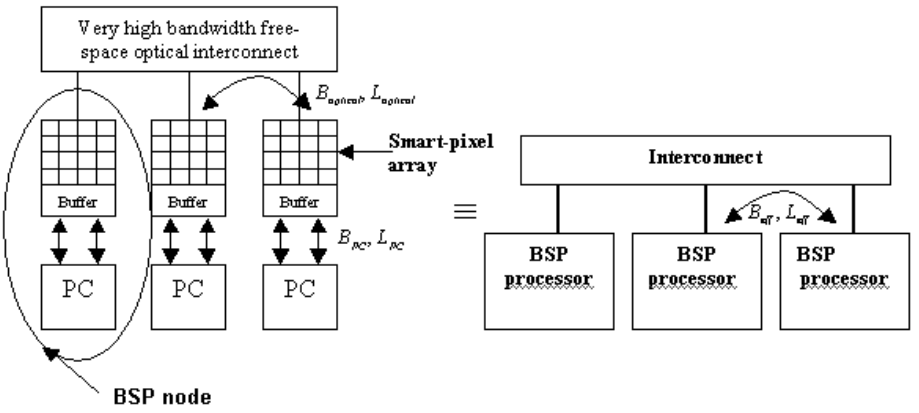

Figure 2: Implementation of the BSP model on the optoelectronic architecture. A BSP processor consists of a PC and its corresponding smart-pixel array and buffer. Equivalence with two-layered system is indicated.

3: Algorithmic case study of the optoelectronic architecture

The practicality of designing algorithms with effective data streaming is assessed in this section, by considering the system performance in a typical application: parallel integer sorting. The algorithm used is parallel sorting by regular sampling (PSRS) [10], chosen because it is asymptotically optimal and is an appropriate algorithm for communicating large messages to utilise the optical bandwidth.

In the implementation of PSRS, data streaming is used both in sending sub-arrays to the smart-pixel layer during local (sequential) sorting and in sending data to the PC layer during merging of primary blocks at the end of the algorithm. As indicated in [11], the cost of the PSRS algorithm is given by,

$$C \approx c\frac{n}{p}\log\left(\frac{n}{p}\right)+\frac{3an}{2p}\log(3p)+g\left(\frac{n}{p^2}\right)\frac{3n}{p}+3l \qquad (5)$$

where the third term on the right-hand side describes the communication cost. Here, *c* and *a* are constants: *a* is twice the number of basic operations required to compare two integers and store in appropriate locations, while *c* is 1.4 (arising from the statistical nature of quicksort) multiplied by the number of operations required to compare and exchange two integers.

A 64-processor machine is considered here. Two parameters are of particular interest: the communication cost and the ratio C_{comp}/C_{comm}. This ratio is critical, since a substantial performance enhancement in communication is of little use if the computation cost dominates.

Results are based on a processor speed of 1Gflop/s and normalised values, $L_{PC}=10^4$ flops and $B_{PC}=1$ bit/flop. In addition, the following expressions were obtained in [11], based on a one-dimensional free space optical interconnect.

$$B_{optical} \le 8192\left[\frac{0.05\times 0.95^X}{1-0.95^X}\right] \quad (Gbit/s) \qquad (6)$$

where, $x = p/2$ and

$$L_{optical} \approx 1.67p + 20 \quad (ns) \qquad (7)$$

In order to assess the validity of the analytic results, simulations of the PSRS algorithm, running on the optoelectronic architecture, were performed. The objectives of the experiments are twofold: 1) an observation of the behaviour of the system in terms of computation and communication phases, and 2) a comparison between the analytical and simulation results. The simulation objectives are achieved by considering a 64-processor machine and varying the size of a word (16, 32 bits) and the number of elements to sort (10^4,...,10^7). The effect which varying these parameters has on computation and communication times are studied and compared to the results obtained analytically. In the examples chosen, data streaming is used to send data from the PC layer to the smart-pixel layer in the first superstep, but is not used in smart-pixel array to PC communication in the final superstep. Further adaptations will be made to the simulator to allow simulations where data streaming is also used in the final superstep.

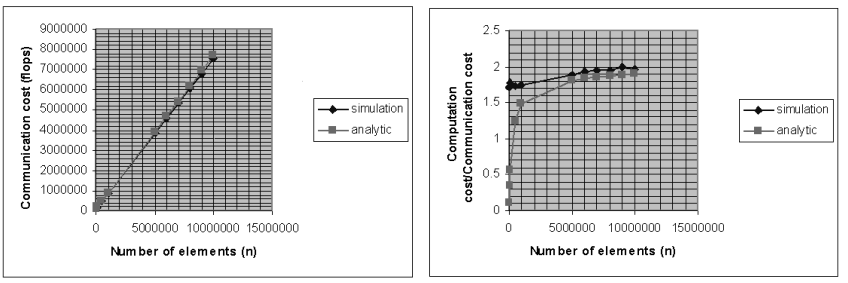

Figure 3: The communication cost and the ratio of computation cost to communication cost as a function of *n*, for word size of 16 bits (*p*=64).

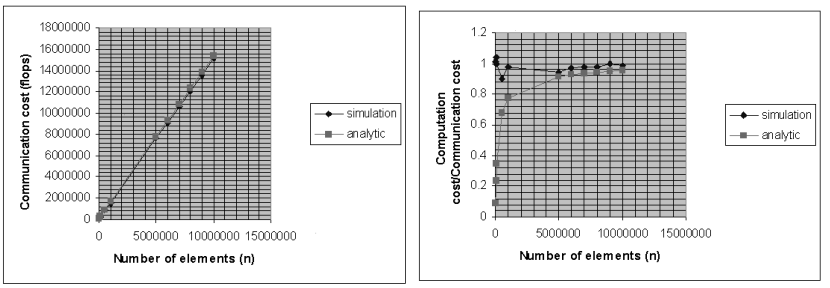

Figure 4: The communication cost and the ratio of computation cost to communication cost as a function of *n*, for word size of 32 bits (*p*=64).

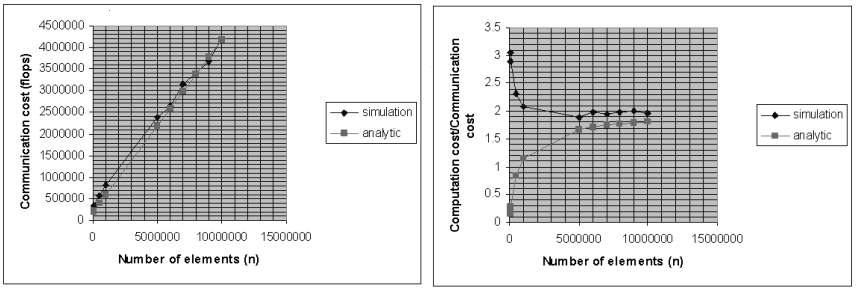

Figure 5: The communication cost and the ratio of computation cost to communication cost as a function of *n*, for word size of 16 bits (*p*=128).

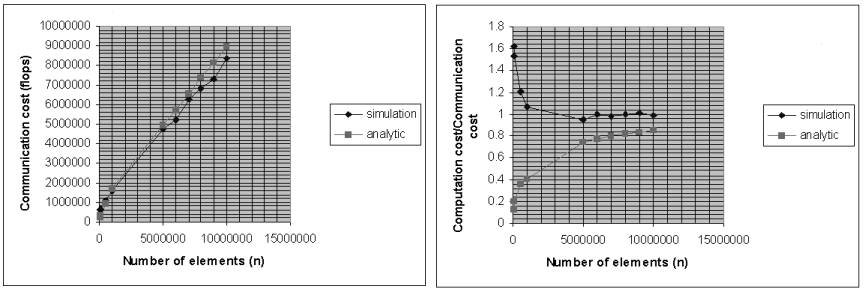

Figure 6: The communication cost and the ratio of computation cost to communication cost as a function of *n*, for word size of 32 bits (*p*=128).

The simulation tool used in obtaining these results, PARSEC (PARallel Simulation Environment for Complex systems) [1], an extension of MAISIE, is a C-based discrete-event simulation language developed at UCLA Parallel Computing Laboratory. This tool adopts the process interaction approach to discrete-event simulation. An object (also referred to as physical process) or set of objects in the physical system is represented by a logical process. Interactions among physical processes (events) are modelled by timestamped messages exchanged among the corresponding logical processes. Although PARSEC programs may be executed using parallel optimistic or conservative protocols as indicated by Fujimoto in [4], the programs developed for simulating the optoelectronic architecture are executed using the traditional sequential simulation protocol (Global Event List). The simulation accounts for event scheduling/execution, taking into consideration both the computation and

communication steps. Events include *sending a message from a PC to smart-pixel array* and *starting a local PC computation*.

The simulator randomly generates the elements to be sorted and performs the three supersteps in the algorithm described above. The simulation has significant storage requirements, with the consequence that results are only presented for up to 10^7 data elements to be sorted. Nevertheless, this is sufficient to make meaningful comparisons with the analytic results in order to validate them.

Figure 3 shows the communication cost and the ratio C_{comp}/C_{comm} for both the analytic and simulation cases, for 16-bit words.

Clearly the agreement improves substantially as *n* increases. Figure 4 shows the same graphs for 32-bit words. Once again it can clearly be seen that the agreement between the simulation and the analytic results improves substantially as *n* increases. In both graphs of C_{comp}/C_{comm} against *n*, the ratio C_{comp}/C_{comm} initially decreases and then begins to increase, according to the simulation results. This is due to the fact that, for small *n*, the computation cost is dominated by the sorting of primary samples in the second superstep. Since this is independent of *n*, this ratio decreases as *n* increases. However, as *n* increases further, the quicksort and merging begin to dominate the computation cost and these increase faster with *n* than the communication cost, so that C_{comp}/C_{comm} begins to increase. Figures 5 and 6 show the same graphs for 128 processors.

Several points are worth making regarding the extent to which the analytical and simulation results are expected to agree. Note that the analytic cost expressions are only approximate. Firstly, the cost of quicksort and merging used in the algorithm (see [11]) are probabilistic, hence the cost presented is only an average cost. Since the run-time is limited by the last processor to finish in each superstep, the average run-time for the quicksort and merging computations may be expected to be a little slower than assumed in the analysis. However, in approximating the merging cost, the worst-case was assumed, Secondly, note that only the terms that are expected to dominate were included in the analytic approach. For example, the cost of sorting primary samples and selecting secondary samples in the second superstep (the cost associated with this is only dependent on *p*, not on *n*), was neglected. This approximation is only valid for sufficiently large *n*. Note also that the cost of barrier synchronisation was neglected, under the assumption that large message sizes would ensure that bandwidth considerations dominate over latency. Naturally, including all of the above terms would improve the accuracy but substantially complicate the analysis. This was considered unnecessary, since the focus is on determining whether the optical bandwidth can be utilised, hence the results of interest are those where *n* is large enough so that bandwidth dominates in the communication cost. The accuracy of this as a function of *n* is made clearer by the results presented below. Consequently, it is expected that the results will agree fairly well for large *n* but not for small *n* and that the general trends, regarding computation and communication costs (and the ratio between them) as *n* increases will be the same in both cases. Both these expectations are satisfied.

The above results indicate that the analytic results are very reasonable for values of *n* exceeding 5 million, for both the 16 and 32-bit (64 processor) cases. Note that all the values of *n* used in the results shown in table 3 exceed this value. Taking these results together, it can be concluded that this approach to implementing the parallel sorting ensures a performance enhancement as long is *n* is large, given that the communication cost constitutes a substantial proportion of the overall cost. For large *n*, the ratio C_{comp}/C_{comm} is approximately 2 for the 16-bit case and 1 for the 32-bit case.

For a conventional cluster, the bandwidth available in communication between a pair of processors under continuous traffic conditions will be substantially less than B_{eff} for the optoelectronic architecture. For example, for the 32-bit word size and $p = 64$, the effective bandwidth for the optoelectronic architecture is 0.99bit/flop, assuming that the PC is operating at 1Gflop/s and the PC bandwidth is 1Gbit/s. For a conventional cluster, the corresponding effective bandwidth is limited by the bandwidth supported by the switch/switches used. For example, noting that a completely connected network (as considered above) requires $p(p-1)/2$ links, a 1Gbit/s switch couldn't support an effective bandwidth of greater than 12Mbit/s when $p = 64$. Clearly this could be improved by using more switches, however even with an effective bandwidth of 100Mbit/s, the communication time for large *n* (i.e. when bandwidth dominates over latency) would be approximately 10 times

greater than the communication time for the optoelectronic architecture.

4: Conclusions and future work

In this paper a parallel system architecture, based on a computational cluster, which makes use of a high bandwidth free-space optical interconnect has been presented and analysed. Both analytic and simulation results have been presented, showing that the optical bandwidth can be exploited to significantly improve inter-processor communication performance. These results were based on the problem of parallel integer sorting. A middle layer, consisting of buffers and smart-pixel arrays with simple computational functionality, is used to manage the bandwidth mismatch between the optical interconnect and the PCs. Large messages are collected in the smart-pixel layer prior to inter-processor communication, using data streaming between the PC and smart-pixel layers, allowing the PC bandwidth bottleneck to be circumvented. By communicating data as a small number of large messages rather than a large number of small messages, the significance of latency is reduced and the optical bandwidth can be utilised. In particular, the PC to smart-pixel layer communication bottleneck can be partially overcome by the data streaming approach discussed in section 2 and the effect of this bottleneck is also reduced by the inherent scalability of this system. This scalability is provided by the fact that the cost of communicating between the smart-pixel and PC layers is independent of the number of processors. Although the optical bandwidth drops off rapidly with p, according to equation (5), this is not a fundamental problem since the value of $B_{optical}$ can be made substantially larger by using a 2D or 3D layout [11].

While the results in this paper indicate the potential of the system architecture presented and show that the optical bandwidth can be exploited, more detailed quantitative results and a thorough analysis of the system architecture is desirable. More detailed simulation results, covering a wider range of cases are desirable. This issue is currently being addressed.

Clearly it is desirable to extend these results to as wide a range of high performance computing applications as possible. Work is underway to consider a number of graphics applications with high bandwidth requirements.

References

1. R. Bagrodia, PARSEC: PARallel Simulation Environment for Complex systems, 1998. http://pcl.cs.ucla.edu/projects/parsec/ .
2. S.J. Cox, D.A. Nicole and K. Takeda, Commodity High Performance Computing at Commodity Prices, Proc. 21st World Ocean and Transputer User Group Technical Meeting, Canterbury, pp.19-26, 1998.
3. J.A.B. Dines, J.F. Snowdon, N. McArdle, Optical Highways for Computing Architectural and Topological Issues, Conference on Lasers and Electro-Optics Europe, 1998. CLEO/Europe. p. 195, 1998
4. R.M. Fujimoto, Parallel and Distributed Simulation Systems, Wiley Interscience, 2000.
5. J. Gourlay, Tsung-Yi Yang, J.A.B. Dines, J.F. Snowdon and A.C. Walker, Development of free-space digital optics in computing, Computer, Volume: 31 Issue: 2 , Feb. 1998, Page(s): 38 –44.
6. J.M.D. Hill, D.B. Skillicorn, Lessons Learned from Implementing BSP, 'High Performance Computing and Networks', Springer Lecture Notes in Computer Science, Vol. 1225, pp. 762-771, 1997.
7. D. A. B. Miller, Physical Reasons for Optical Interconnection, Special Issue on Smart Pixels, Int'l J. Optoelectronics 11 (3), 155-168 (1997).
8. D.B. Skillicorn, J.M.D. Hill, W.D. McColl, Questions and Answers about BSP, Scientific Programming, 6 (3), pp. 249-274, 1997.
9. Technology roadmap, Optoelectronic interconnects for integrated circuits, European Commission, Esprit programme Long Term Research, Microelectronics advanced research initiative MEL-ARI OPTO, June 1998.
10. A. Tiskin, The Design and Analysis of Bulk-Synchronous Parallel Algorithms, PhD thesis, University of Oxford, 1998.
11. I. Gourlay, P.M. Dew, K.Djemame, J.F.Snowdon, G. Russell, T. Lim, B. Layet, 'Supporting highly parallel computing with a high bandwidth interconnect', internal report, 2001(http://www.comp.leeds.ac.uk/research/pubs/publications.shtml).