Algorithms of Basic Communication Operation on the Biswapped Network

Wenhong Wei and Wenjun Xiao

Department of Computer Science, South China University of Technology,
510641 Guangzhou, China
hquwwh@tom.com, wjxiao@scut.edu.cn

Abstract. Biswapped network (BSN) is a new topology for interconnection networks in multiprocessor systems. BSN is built of $2n$ copies of an n-node basic network and total nodes are $2n^2$. Some topological properties of BSN have been investigated, and some algorithms have been developed on the BSN such as sorting and matrix multiplication etc. In this paper, we develop algorithms for some basic communication operations—broadcast, prefix sum and data sum etc.

Keywords: BSN, Broadcast, Prefix sum, Data sum.

1 Introduction

The swapped network is also called as the OTIS-network and has important applications in parallel processing [1,2]. In this network architecture, n^2 processors are divided n groups where there are n processors, and processors in the same group are connected by intra-group link, simultaneously, these groups are connected by inter-group link. But swapped network is not a Cayley graph, and then it is not a symmetrical network architecture, so some algorithms on it are not always convenient. For remedying this limitation about swapped network, [3] proposed biswapped network (BSN), the new network is a class of Cayley graph if the basic network is a Cayley graph and is tight related to the swapped network. BSN is of more regularity than the swapped network. BSN is built of $2n$ copies of an n-node basic network using a simple rule for connectivity that ensures its regularity, modularity, fault tolerance, and algorithmic efficiency. Some topological properties of BSN have been investigated [3], and some algorithms have been developed on the BSN such as sorting and matrix multiplication etc [4].

In most parallel algorithms, processors need to exchange data with other processors, hence it is the most important to develop algorithms of basic communication operation, and algorithms for basic communication operation can be used to arrive at efficient parallel algorithms for numerous applications, from image processing, computational geometry, matrix algebra, graph theory, and so forth [5]. In [6], Wang and Sahni developed algorithms of basic operations on OTIS-Mesh, their basic operation algorithms including broadcast, prefix sum and data sum etc can be only applied to OTIS-Mesh. In this paper, we develop deterministic algorithms of basic communication

M. Bubak et al. (Eds.): ICCS 2008, Part I, LNCS 5101, pp. 347–354, 2008.

operation for parallel computation on the BSN, such as broadcast, prefix sum and data sum etc, and analyze time complexity of these algorithms. According to [4], we know BSN has better topological properties than OTIS, the basic communication operation algorithms on the BSN proposed by us, are more general and better than those on OTIS-Mesh. For example, in a $2n^2$ processors BSN-Mesh, our broadcast algorithm's time complexity is $4\sqrt{n}-2$, but in a n^2 processors OTIS-Mesh, their broadcast algorithm's time complexity is $4\sqrt{n}-3$. As the number of processor in our network is bigger than theirs, we can conclude that our broadcast algorithm is better than theirs. The remainder of this paper is organized as follows. In Section 2, we give the definition of BSN. Section 3 presents the basic data communication algorithms on BSN including broadcast, prefix sum and data sum etc, and analyzes time complexity of these algorithms. Finally, in Section 4, we provide some concluding remarks.

2 Introduction of BSN

Definition 1. Let Ω be a graph with the vertex set $V(\Omega)=\{h_1,h_2,...,h_n\}$ and the arc set $E(\Omega)$. Our biswapped network $\Sigma(\Omega)=\Sigma=(V(\Sigma),E(\Sigma))$ is a graph defined as follows [3]:

$V(\Sigma)=\{\langle g,p,0\rangle, \langle g,p,1\rangle \mid g,p\in V(\Omega)\}$

and

$E(\Sigma)=\{(\langle g,p_1,0\rangle,\langle g,p_2,0\rangle),(\langle g,p_1,1\rangle,\langle g,p_2,1\rangle)\mid (p_1,p_2)\in E(\Omega), g\in V(\Omega)\}\cup\{(\langle g,p,0\rangle,\langle p,g,1\rangle),(\langle g,p,1\rangle,\langle p,g,0\rangle)\mid g,p\in V(\Omega)\}$

Intuitively, if we regard the basis network as group, the definition postulates $2n$ groups, each group being a Ω digraph: n groups, with nodes numbered ⟨group#, processor#, 0⟩, form part 0 of the bipartite graph, and other n groups constitute part 1, with associated node numbers ⟨group#, processor#, 1⟩. Each group p in either part of Σ has the same internal connectivity as Ω (intra-group edges, forming the first set in the definition of $E(\Sigma)$). In addition, node g of group p in part 0/1 is connected to node p in group g of part 1/0 (inter-group or swap edges in the second set in the definition for $E(\Sigma)$). The name "biswapped network" (BSN) arises from two defining properties of the network just introduced: when group are viewed as super-nodes, the resulting graph of super-nodes is a complete $2n$-node bipartite graph, and the inter-group links connect nodes in which the group number and the node number within group are interchanged or swapped.

When $\Omega=C_4$ is ring, an example of the network $\Sigma(C_4)$ is denoted in Fig. 1.

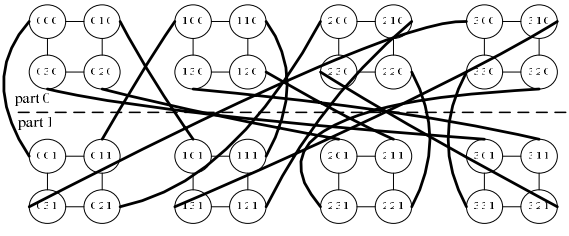


Fig. 1. An example of the BSN with $\Omega=C_4$

Like swapped network (or OTIS), and links between vertices of the same group are regarded as intra-group links. The links between vertices in a group and another group are following a swapping strategy, which are regarded as inter-group links.

3 Basic Communication Operations on the BSN

3.1 Broadcast

Broadcast is, perhaps, the most fundamental operation in the parallel computing. In this operation, data is initially in a single processor, and after broadcasting, it is to be transmitted to all other processors in the same network. For example, if processor <0, 0, 0> has value A in BSN, all $2n^2$ processors of the BSN have value A after broadcasting. Suppose that broadcast is applied in all-ports mode, and it can be accomplished using the following four-step algorithm if we suppose processor u (u=<p, g, 0>) has data x:

Table 1. Algroithm for broadcast

Step 1: processor u transmit its data x to processor v (v=<g, p, 1>) by inter-group connection.
Step 2: processor u and processor v broadcast its data x to all other processors simultaneously by intra-group connection in respective group.
Step 3: each processor in group g and group g' transmits its data x to all other processors by inter-group connection (g' is a group which processor v lie in and g'=p).
Step 4: processor of each group broadcasts the data x to other processors within its group except group g and group g'.

In Step 1, processor u transmit a copy of data x to processor v by an inter-group move, and then processor v has the data x. In Step 2, processor u and processor v broadcast their data x in their respective group simultaneously, so each processor of group g (processor u lie in group g) and group g' (processor v lie in group g') has a copy of data x. Step 3 is similar to Step 1, and after Step 3, the processor of each group which connect to processor u or processor v by inter-group connectivity has a copy of data x. After the last step, each processor of BSN has a copy of data x.

Fig. 2 shows the process of broadcast on the BSN-C_3, data x is stored in processor <0, 0, 0> initially.

In the whole process of this algorithm, we suppose inter-group move and intra-group move are equivalent, so we use one unit to denote inter-group or intra-group move time of one data. If the basic network of $2n^2$ processors BSN is mesh, Step 2 and 4 will take $2(\sqrt{n}-1)$ time each, Step 1 and Step 3 take one unit time each. So the complexity is $4\sqrt{n}-2$. If the basic network of BSN is hypercube or other connectivity graphs, we can analyze the algorithm complexity similarly.

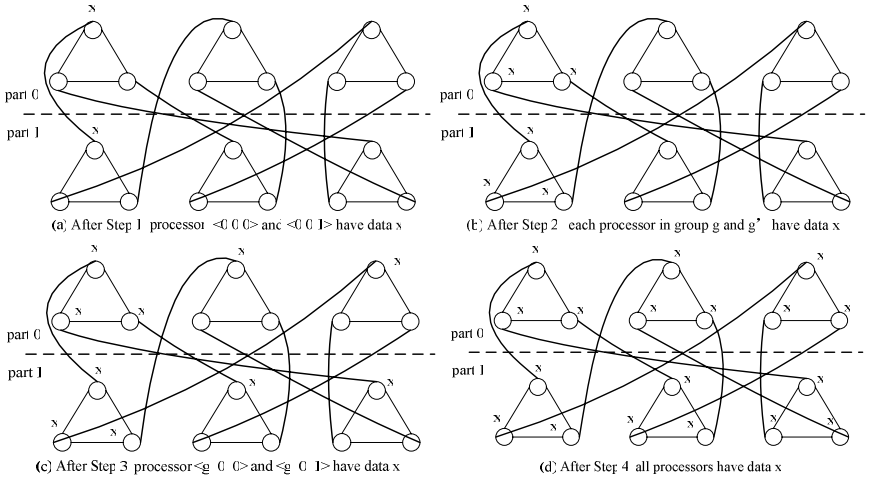


Fig. 2. An example of broadcast on the BSN with $\Omega=C_3$

Theorem 1. The broadcast algorithm on BSN is optimal if the basic network's broadcast algorithm is optimal in all-ports mode.

Proof. If broadcast algorithm is optimal on a network in all-ports mode, the diameter of that network is equal to the move steps of broadcast. Let Ω denote basic network of BSN and D denote its diameter, according to [3], the diameter of BSN is 2D(Ω)+2. In our broadcast algorithm, there are two broadcasts in the basic network, so the move steps are 2D(Ω) if the basic network's broadcast algorithm is optimal. In addition, there are two inter-group moves, and our broadcast algorithm needs 2D(Ω)+2 moves, which equal to the diameter of BSN, so our broadcast algorithm is optimal.

3.2 Prefix Sum

In $2n^2$ processors BSN, if we label each group of part 0 from 0 to n-1 and each group of part 1 from n to $2n$-1, and label each processor of each group from 0 to n-1. Now, let D(p) be the data in processor p, $0 \le p < 2n^2$. In a prefix sum, each processor p computes $PS(p) = \sum_{i=0}^{p} D(i), 0 \le p < 2n^2$. So prefix sum algorithm results from the following equation:

$$PS(p)=SD(p)+LS(p) \tag{1}$$

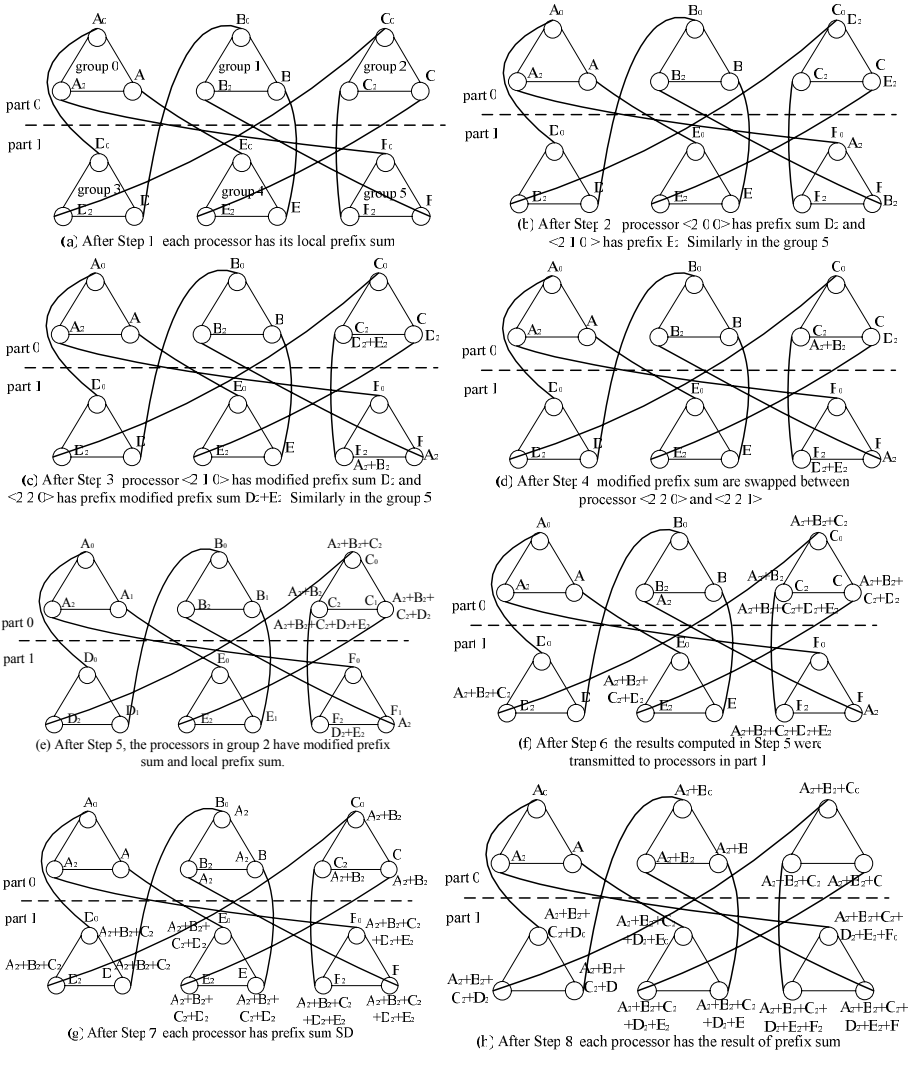
Where SD(p) is the sum of D(i) over all processors i whose group label is smaller than that of p and LS(p) is the local prefix sum within the group of p. The algorithm for prefix sum is shown in Table 2.

Table 2. Algroithm for prefix sum

Step 1: perform a local prefix sum in each group.
Step 2: transmit prefix sums computed in Step 1 for processor n-1 in each group of part 0 to processors in group 2n-1 and for processor n-1 in each group (except group 2n-1) of part 1 to processors in group n-1 by inter-group connection.
Step 3: in group n-1 and group 2n-1, perform a modified prefix sum in data A which is received in Step 2. In this modification, processor P computes $\sum_{i=0}^{P-1} A(i)$ rather than $\sum_{i=0}^{P} A(i)$ ($P \geq 1$).
Step 4: swap between prefix sums computed in Step 3 for processor n-1 in group 2n-1 of part 1 and processor n-1 in group n-1 of part 0 by inter-group connection.
Step 5: after summing result in Step 4 and local prefix sum, processor n-1 in group n-1 of part 0 broadcasts the result to each processor and each processor in group n-1 of part 0 performs sum in the result and data A.
Step 6: transmit prefix sums computed in Step 5 for processor n-1 of group n-1 to each group of part 1 and prefix sums computed in Step 3 for processor n-1 of group 2n-1 to each group of part 0.
Step 7: broadcast the result from Step 6 in each processor.
Step 8: perform sum in local prefix sum and modified prefix sum in each processor.

Following Step 1, each group computed local prefix sum and the result is stored in processor respectively. Step 2 corresponds to inter-group transmission operation, in Step 2, the results of local prefix sum in all groups in part 0 are transmitted to each processor in group 2n-1 of part 1, and similarly, the results of local prefix sum in all groups except for group 2n-1 of part 1 are transmitted to each processor in group n-1 of part 0. In Step 3, processors in group n-1 and group 2n-1 perform a modified prefix sum on the data which is received in previous step, the modified prefix sum of current processor is equal to the prefix sum of preceding processor. For example, if data A_0, A_1, …, A_{n-1} is stored in $processor_0$, $processor_1$, …, $processor_{n-1}$ respectively, the modified prefix sum of n data is equal to 0, A_0, A_0+A_1, …, $A_0+A_1+\ldots+A_{n-2}$. In Step 4, modified prefix sum computed in group n-1 of part 0 and group 2n-1 in part 1 are swapped, and processor n-1 in group n-1 has prefix sum of previous n-1 groups of the part 0 and processor n-1 in group 2n-1 has prefix sum about pervious n-1 groups of the part 1. In Step 5, the data from processor n-1 in group 2n-1 that were added local prefix sum in processor n-1 of group n-1 were broadcasted to other processors in the same group, and then each processor in this group has the prefix sum of previous n groups in part 0, at last, each processor performed sum in it and modified prefix sum. After Step 6, processor n-1 in each group has modified prefix sum. Following Step 7, each processor has modified prefix sum. In last step, each processor computed Equation (1) and each processor in each group has last result.

Fig. 3 shows the process of prefix sum on the BSN-C_3, we denote A_0, A_1, A_2 as the prefix sum of processor <0, 0, 0>, <0, 1, 0>, <0, 2, 0> in group 1 and B_0, B_1, B_2 as the prefix sum of processor <1, 0, 0>, <1, 1, 0>, <1, 2, 0> in group 2, the remainder groups are similar.



(a) After Step 1 each processor has its local prefix sum

(b) After Step 2 processor <2 0 0> has prefix sum D_2 and <2 1 0> has prefix E_2. Similarly in the group 5

(c) After Step 3 processor <2 1 0> has modified prefix sum D_2 and <2 2 0> has prefix modified prefix sum D_2+E_2. Similarly in the group 5

(d) After Step 4 modified prefix sum are swapped between processor <2 2 0> and <2 2 1>

(e) After Step 5, the processors in group 2 have modified prefix sum and local prefix sum.

(f) After Step 6 the results computed in Step 5 were transmitted to processors in part 1

(g) After Step 7 each processor has prefix sum SD

(h) After Step 8 each processor has the result of prefix sum

Fig. 3. An example of prefix sum in the BSN with $\Omega=C_3$

We think about the algorithm complexity in worst occasion now, there are 3 transmitting data in inter-group, 2 broadcasts and 2 prefix sum operations in this algorithm (the time of arithmetic operation is ignored). We know that broadcast prefix sum operation will cost the most time in the array, so if the basic network of BSN is an array, the algorithm complexity is worst. Now we suppose that transmitting and broadcast a data will cost one time unit, the broadcast time is n-1 and prefix sum operation time is also n-1 in n processors array. So the whole algorithm's complexity of $2n^2$ processors BSN is $4n$-1 at worst.

3.3 Data Sum

Data sum is also named as semi-group computing, each processor is to compute the sum of the values of all processors. The algorithm is shown in Table 3.

Table 3. Algroithm for data sum

Step 1: each processor of each group performs data sum in itself group.
Step 2: each processor of each group transmits its data sum to other processor by the inter-group connection.
Step 3: each processor of each group performs data sum operation in received data from other groups.
Step 4: each processor in each group swaps the results that are computed in Step 3 by inter-group connection.
Step 5: the processor of each group performs the sum operation in itself group.

In Step 1, each processor performs data sum in intra-group. After Step 2, each processor in part 0 has the sum of data in part 1, and each processor of the part 1 has the sum of data in part 0 contrarily. In Step 3, each processor sums the data that was received by inter-group connection, and then the processors of part 0 (part 1) have the sum of all processors of the part 1 (part 0). After Step 4, each processor in each group has the sum of all the processors. In Step 5, all the processors perform the sum operation and each processor of each group has final data sum.

In $2n^2$ processors BSN, if the basic network is a complete graph, our algorithm complexity is the best. Suppose that intra-group and inter-group data transmission cost one time unit. One data sum operation is performed in Step 1, Step 3 and Step 5 respectively, which cost 3 time steps in all at best. One data transmission operation is performed in Step 2 and Step 4 which totally cost two time steps. So the whole algorithm complexity is 5 at best.

Table 4. Comparison between our algorithms and [6]

	OTIS-Mesh	BSN-Array	BSN-Mesh	BSN-Complete graph
Broadcast	$4\sqrt[4]{N}-3$	$\sqrt{2N}$	$2\sqrt[4]{8N}-2$	4
Prefix sum	$8\sqrt[4]{N}-6$	$2\sqrt{2N}-1$	$4\sqrt[4]{8N}-5$	7
Data sum	$8\sqrt[4]{N}-7$	$\frac{3}{2}\sqrt{2N}-1$	$3\sqrt[4]{8N}-4$	5

4 Conclusion

In this paper, we have developed the algorithms of basic communication operation on the BSN including broadcast, prefix sum and data sum etc, which are important in parallel computing model, and also analyzed these algorithms' time complexity. We assume that there are N processors in OTIS-Mesh and BSN respectively, comparison

between our basic communication algorithms and [6] in the time complexity shows in Table 4.

From the Table 4, we know that our algorithms including broadcast, prefix sum and data sum are better than the algorithms in [6] when the basic networks are same. In our algorithms, the time complexity is constant where basic network is complete graph, that is, the time complexity is constant at best.

Acknowledgments. This work is supported by the Doctorate Foundation of South China University of Technology and Open Research Foundation of Guangdong Province Key Laboratory of Computer Network.

References

1. Parhami, B.: Swapped Interconnection Networks: Topological, Performance, and Robustness Attributes. Journal of Parallel and Distributed Computing 65, 1443–1452 (2005)
2. Day, K., Al-yyoub, A.: Topological Properties of OTIS-networks. IEEE Transactions on Parallel and Distributed Systems 13(4), 359–366 (2002)
3. Xiao, W.J., Chen, W.D., He, M.X., Wei, W.H., Parhami, B.: Biswapped Network and Their Topological Properties. In: Proceedings Eighth ACIS International Conference on Software Eng., Artific. Intelligence, Networking, and Parallel/Distributed Computing, pp. 193–198 (2007)
4. Wei, W.H., Xiao, W.J.: Matrix Multiplication on the Biswapped-Mesh Network. In: Proceedings Eighth ACIS International Conference on Software Eng., Artific. Intelligence, Networking, and Parallel/Distributed Computing, pp. 211–215 (2007)
5. Sahni, S., Wang, C.F.: BPC Permutations on the OTIS-Mesh Optoelectronic Computer. In: Proc. Fourth International Conference on Massively Parallel Processing Using Optical Interconnections, pp. 130–135 (1997)
6. Wang, C.F., Sahni, S.: Basic Operations on the OTIS-Mesh Optoelectronic Computer. IEEE Trans. Parallel and Distributed Systems 9, 1226–1236 (1998)
7. Coudert, D., Ferreira, A., et al.: Topologies for Optical Interconnection Networks Based on the Optical Transpose Interconnection System. Applied Optical IP 39(17), 2965–2974 (2000)
8. Day, K., Al-yyoub, A.: Topological Properties of OTIS-networks. IEEE Transactions on Parallel and Distributed Systems 13(4), 359–366 (2002)