www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 7 July, 2014 Page No. 7173-7176

Enumeration Sort on OTIS k-Ary n-Cube Architecture

***Abdul Hannan Akhtar*[1]*, Keny T. Lucas*[2]**

[1]Government Polytechnic,
Ranchi, Dist. Ranchi, Jharkhand, PIN – 834001, India
abdulhannan.akhtar@yahoo.com

[2]Xavier Institute of Polytechnic and Technology,
Namkum, Dist. Ranchi, Jharkhand, PIN - 834010, India
kenny.lucas@gmail.com

Abstract: ***Many researchers have been motivated to propose parallel algorithm on Optical Transpose Interconnection System (OTIS) because of its hybrid nature. OTIS exploits both the electronic links as well as free space optical links for connecting processing nodes of interconnection network. In this paper, we have proposed a parallel algorithm for sparse enumeration sort on OTIS k-ary n-cube parallel computer with on a network size of k^{2n}.. In this sorting, the number of keys to be sorted is p^{α}, for some constant $\alpha \leq ½$ and we have assumed $\alpha = ½$ for our proposed algorithm. The algorithm has two variants based on data population techniques. The time complexity of the algorithm has been observed to be 4n(k-1) electronic moves + 3 OTIS moves for the first case. In the second case also it requires the same number of electronic moves but needs only two OTIS moves.***

Keywords: parallel algorithm, interconnection network, OTIS *k*-ary *n*-cube, enumeration sort, time complexity**.**

1. Introduction

An Optical Transpose Interconnection System (OTIS) [1], [2], [3] is a hybrid interconnection Network that uses electronic as well as optical links. The electronic links are used to connect the processors placed within a few millimeters range. These processors in a group can be assumed to be fabricated within a chip. The intergroup processors are connected to each other through free space optical links. The processors within the OTIS models are divided into number of groups. The number of groups in the network and the processors within the group may or may not be the same in number. However, if the number of groups and the number of processors in each group, the bandwidth of the OTIS systems can be maximized and the power consumption can be minimized [4], [5], [6]. The interconnection pattern of each group decides the overall OTIS network architecture. In the recent years, researchers have exploited OTIS architecture to solve many communication and computation intensive problems. A rich literature of works on OTIS model is available such as image processing [7], BPC permutation [8], randomized routing and selection [9], matrix multiplication [10], [11], sorting [12], [13], [14], [15], polynomial interpolation [16], root finding [16], [17], conflict graph [18], gossiping [19], [20], [21].

In this paper, we are proposing a parallel algorithm for sparse enumeration sort on OTIS *k*-ary *n*-cube architecture. The sparse enumeration sort is a class of sorting in which the number of keys to be sorted is much less than the network size. In this sorting, the number of keys is typically assumed to be p^{α} for some constant $\alpha \leq ½$, where *p* is the network size [22]. This assumption has also been followed in [22], [23], [13]. The time complexity of the proposed algorithm is expressed in terms of electronic moves and optical moves. Our proposed algorithm takes $4n(k\text{-}1)$ electronic moves and three OTIS moves for k^n data size on k^{2n} processors for one data initialization technique. For another data initialization approach, it takes the same number of electronic moves but only 2 OTIS moves. The algorithm for the enumeration sort has also been proposed in [13] for OTIS- MOT with a time complexity of $4.5\log N$ and 5 OTIS moves for a network size of *N*.

The paper is organized as follows: the topology of the OTIS *k*-ary *n*-cube is presented in section 2. In section 3, we have presented the proposed algorithm for enumeration sort followed by conclusion in section 4.

2. Topology of OTIS *k*-Ary *n*-Cube

OTIS *k*-ary *n*-cube is composed of k^n disjoint groups of *k*-ary *n*-cube [24], [25]. Each group is *n*-dimensional grid structure and has *k* processing nodes in each dimension. There are k^n groups in the network and each group contains k^n processors within it. Thus the size of the OTIS *k*-ary *n*-cube interconnection network is k^{2n}. The torus (where n=2 or 3) and hypercube (where n=2) are the two most popular architectural variations of *k*-ary *n*-cube. The hypercube architecture has been used in iPSC/2 [26] and iPSC/860 [27] whereas the torus has been used in J-Machine [28], CRAY-3TD [29] and CRAY-3TE [30] parallel computers. In the last few years, many parallel algorithms have been proposed on different variants of OTIS *k*-ary *n*-cube for solving different problems, i.e. matrix multiplication [11], prefix computation [31], gossiping [21]. In any OTIS model, the processors within the group are connected through electronic links whereas the intergroup processors are connected through free space optical links. The processors of one group are connected to the processors of other groups based on optical transpose rule. Let any processor be denoted by processor's number and the group's number within which that processor lies, i.e., $(G_{g_1g_2g_3\cdots g_x}, P_{p_1p_2p_3\cdots p_x})$ then the processor $(G_{g_1g_2g_3\cdots g_x}, P_{p_1p_2p_3\cdots p_x})$ is connected to processor $(G_{p_1p_2p_3\cdots p_x}, P_{g_1g_2g_3\cdots g_x})$ through the free space optical link. As an example, all the processors of first group are connected to the first processor of all the rest of groups in the overall interconnection network through optical links shown as dotted lines of OTIS 3-ary 2-cube as shown in Fig. 1.

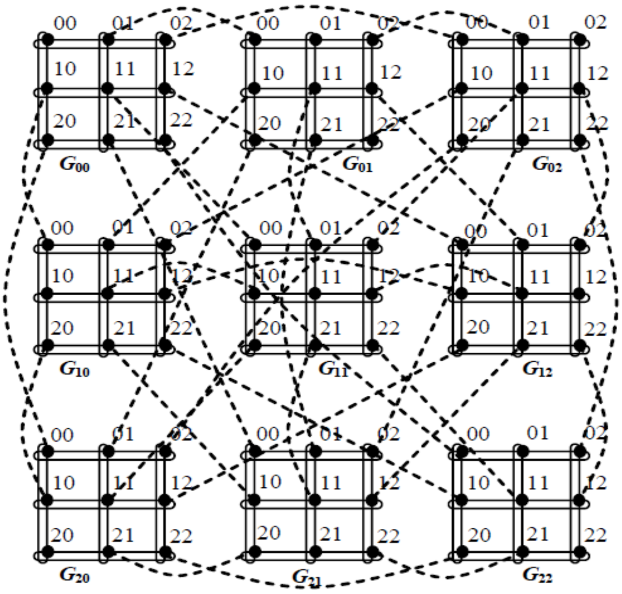

Figure 1: OTIS 3-ary 2-cube

3. Proposed Algorithm for Enumeration Sort

Here, we are proposing parallel algorithm for sparse enumeration sort for which k^n data elements are to be considered for a network size of k^{2n}. In terms of data population, we are considering two cases. In case I, the data elements are populated only in the first group. The data elements are populated only in the first processor of all the groups in case II.

Case I: all the data elements are populated in G_{111}

```
Step 1: Data Initialization
    /* For all processors in $G_{111}$, do in parallel */
        $B_{p_1p_2p_3\cdots p_x} = \beta_{p_1p_2p_3\cdots p_x}$, where $1 \le x \le k$

Step 2: /* For all processors in $G_{111}$, do in parallel */
        $C_{p_1p_2p_3\cdots p_x} = B_{p_1p_2p_3\cdots p_x}$, where $1 \le x \le k$

Step 3: /* For all processors in $G_{111}$, do in parallel */
        Perform OTIS move on $B_{p_1p_2p_3\cdots p_x}$ and $C_{p_1p_2p_3\cdots p_x}$

Step 4. /* For all Groups, do in parallel */
Perform Local Broadcast on $B_{p_1p_2p_3\cdots p_x}$ and
$C_{p_1p_2p_3\cdots p_x}$ where $p_1=p_2=p_3= \ldots =p_x=1$ and $1 \le x \le k$

Step 5: /* For all Groups, do in parallel */
        /* For all processors, do in parallel */
    Perform OTIS move on $C_{p_1p_2p_3\cdots p_x}$, where $1 \le x \le k$

Step 6: /* For all Groups, do in parallel */
        /* For all Processors, do in parallel */
    If $B_{p_1p_2p_3\cdots p_x} \ge C_{p_1p_2p_3\cdots p_x}$, where $1 \le x \le k$
    Then
        $A_{p_1p_2p_3\cdots p_x} = A_{p_1p_2p_3\cdots p_x} + 1$
    Else
        $A_{p_1p_2p_3\cdots p_x} = A_{p_1p_2p_3\cdots p_x} + 0$

Step 7. /* For all Groups, do in parallel */
        /* For all the processors, do in parallel */
Find the sum of $A_{p_1p_2p_3\cdots p_x}$ within the group and broadcast it within the group

Step 8: /* For all Groups, do in parallel */
        /* For all processors, do in parallel */
If $(p_1-1)k^{n-1}+(p_2-1)k^{n-2}+(p_3-1)k^{n-3} \ldots p_x = A_{p_1p_2p_3\cdots p_x}$
Then
    Perform local broadcast on $A_{p_1p_2p_3\cdots p_x}$

Step 9: /* For all Groups, do in parallel */
        /* For all processors, do in parallel */
Perform OTIS move on $A_{p_1p_2p_3\cdots p_x}$ and $B_{p_1p_2p_3\cdots p_x}$
where $A_{p_1p_2p_3\cdots p_x}$ holds the rank of $B_{p_1p_2p_3\cdots p_x}$
```

Time complexity: Steps 1, 2 and 6 take constant unit time each. One OTIS move is required for steps 3, 5 and 9 each. Step 4 and 8, each requires $2n(k-1)$ electronic moves. $2n(k-1)$ electronic steps are required for step 7. Thus the overall time complexity is $4n(k-1)$ electronic moves + 3 OTIS moves.

Case II: the data elements are populated in $P_{p_1p_2p_3\cdots p_x}$ of all the groups where $p_1=p_2=p_3= \ldots =p_x=1$ and $1 \le x \le k$

```
Step 1: Data Initialization
    /* For all groups, do in parallel */
$B_{p_1p_2p_3\cdots p_x} = \beta_{p_1p_2p_3\cdots p_x}$, where $p_1=p_2=p_3= \ldots =p_x=1$ and $1 \le x \le k$

Step 2: /* for all groups, do in parallel */
$C_{p_1p_2p_3\cdots p_x} = B_{p_1p_2p_3\cdots p_x}$, where $p_1=p_2=p_3= \ldots =p_x=1$ and $1 \le x \le k$

Step 3: Perform Local Broadcast on $B_{p_1p_2p_3\cdots p_x}$ and
$C_{p_1p_2p_3\cdots p_x}$ where $p_1=p_2=p_3= \ldots =p_x=1$ and $1 \le x \le k$
```

```
Step 4: /* For all Groups, do in parallel */
            /* For all processors, do in parallel */
        Perform OTIS move on $C_{p_1p_2p_3\cdots p_x}$, where $1\le x \le k$

Step 5: /* For all Groups, do in parallel */
            /* For all Processors, do in parallel */
        If $B_{p_1p_2p_3\cdots p_x} \ge C_{p_1p_2p_3\cdots p_x}$, where $1\le x \le k$
        Then
            $A_{p_1p_2p_3\cdots p_x} = A_{p_1p_2p_3\cdots p_x} +1$
        Else
            $A_{p_1p_2p_3\cdots p_x} = A_{p_1p_2p_3\cdots p_x} +0$

Step 6: /* For all Groups, do in parallel */
            /* For all the processors, do in parallel */
Find the sum of $A_{p_1p_2p_3\cdots p_x}$ within the group and broadcast it within the group

Step 7: /* For all Groups, do in parallel */
            /* For all processors, do in parallel */
If $(p_1-1)k^{n-1}+(p_2-1)k^{n-2}+(p_3-1)k^{n-3} \ldots p_x = A_{p_1p_2p_3\cdots p_x}$
Then
    Perform local broadcast on $A_{p_1p_2p_3\cdots p_x}$

Step 8: * For all Groups, do in parallel */
            /* For all processors, do in parallel */
Perform OTIS move on $A_{p_1p_2p_3\cdots p_x}$ and $B_{p_1p_2p_3\cdots p_x}$
where $A_{p_1p_2p_3\cdots p_x}$ holds the rank of $B_{p_1p_2p_3\cdots p_x}$
```

Time complexity: In case II, steps 1, 2, 5 take constant unit time. One OTIS move is needed in step 4 and 8 each. Steps 3 and 7 take $n(k-1)$ electronic moves each. $2n(k-1)$ electronic moves are needed to compete step 7. Thus the overall time complexity can be expressed as $4n(k-1)$ electronic moves + 2 OTIS moves.

4. Conclusion

In the article we presented parallel algorithm for sparse enumeration sort for a network size of k^{2n} considering two instances of data population techniques. Case I has a time complexity of $4n(k-1)$ electronic moves + 3 OTIS moves. In the second case, it is $4n(k-1)$ electronic moves + 2 OTIS moves. Our proposed algorithm can be compared with [own] where the algorithm for sparse enumeration sort has been presented for OTIS- MOT with a time complexity of $4.5 logN$ and 5 OTIS moves for a network size of N.

References

[1] G. Marsden, P. Marchand, P. Harvey and S. Esener, "Optical Transpose Interconnection System Architecture", Optics Letters, XVIII(13), pp 1083-1085, 1993.

[2] F. Kiamilev, P. Marchand, A. Krishnamoorthy, S. Esener, S. Lee, "Performance Comparison between Optoelectronic and VLSI Multistage Interconnection Networks", Journal of Light Wave Technology, 9(12), pp 1674 – 1692, 1991.

[3] F. Zane, P. Marchand, R. Paturi, S. Esener, "Scalable Network Architecture using the Optical Transpose Interconnection System (OTIS)", In Proceedings of the International Conference on Massively Parallel Processing using Optical Interconnections (MPPOI' 96), San Antonio, Texas, pp 114-121, 1996.

[4] F. Kiamilev, P. Marchand, A. Krishnamoorthy, S. Esener, S. Lee, "Performance Comparison between Optoelectronic and VLSI Multistage Interconnection Networks", Journal of Light Wave Technology, 9(12), pp 1674 – 1692, 1991.

[5] A. Krishnamoorthy, P. Marchand, F. Kiamilev, S. Esener, "Grain-size Consideration for Optoelectronic Multistage Interconnection Networks", Applied optics, XXXI(26), pp 5480-5507, 1992.

[6] M. Feldman, S. Esener, C. Guest, S. Lee, "Comparison between Electrical and Free-space Optical Interconnects based on Power and Speed Consideration", Applied Optics, XXVII(9), pp 1742-1751, 1990.

[7] C. –F. Wang and S. Sahni, "Image Processing on the OTIS-Mesh Optoelectronic Computer", IEEE Transactions on Parallel and Distributed Systems, XI(2), pp 97-109, 2000.

[8] S. Sahni and C. -F. Wang, "BPC Permutations on the OTIS Hypercube Optoelectronic Computer", Informatica, XXII, pp 263-269, 1998.

[9] S. Rajasekaran and S. Sahni, "Randomized Routing, Selection and Sorting on OTIS-Mesh", IEEE Transaction on Parallel and Distributed Systems, IX(9), pp 833-840, 1998.

[10] C. –F. Wang and S. Sahni, 2001, "Matrix Multiplication on the OTIS-Mesh Optoelectronic Computer", IEEE Transactions on Computers, XXXXX(7), pp 635-646, 2001.

[11] Mallick D. K. and Jana P. K., "Matrix Multiplication on OTIS *k*-Ary 2-Cube Network", In Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, pp 224-228, 2008.

[12] A. Osterloh, "Sorting on OTIS-Mesh", In Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS 2000), pp 269-274, 2000.

[13] K. T. Lucas and P. K. Jana, "Sorting and Routing on OTIS-Mesh of Trees", Parallel Processing Letters, XX(2), pp 145-154, 2010.

[14] K. T. Lucas, "Parallel Enumeration Sort on OTIS-Hypercube", In Proceedings of the IC3, pp 21-31.

[15] K. T. Lucas, "Parallel algorithm for Sorting on OTIS-Ring Computer", In Proceedings of the Bangalore Annual Compute Conference (COMPUTE 2009), pp 1-5.

[16] P. K. Jana, "Polynomial Interpolation and Polynomial Root Finding on OTIS-Mesh", Parallel Computing, XXXII(4), pp 301-312, 2006.

[17] K. T. Lucas and P. K. Jana, "Parallel Algorithms for Finding Polynomial Roots on OTIS-Torus", Journal of Supercomputing, XXXXIV(2), pp 139-153, 2010.

[18] K. T. Lucas, D. K. Mallick, P. K. Jana, "Parallel Algorithms for the Conflict Graph on the OTIS-Triangular Array", In Proceedings of the International

Conference on Distributed Computing and Networking, Springer-Verlag Berlin Heidelberg, pp 274-279, 2008.

[19] K. T. Lucas, "The Gossiping on OTIS-Hypercube Optoelectronic Parallel Computer", In Proceedings of the International Conference on Parallel and Distributed Techniques and Applications, Las Vegas, Nevada, USA, pp 185-189, 2007.

[20] T. F. Gonzalez, "An Efficient Algorithm for Gossiping in the Multicasting Communication Environment", IEEE Transactions on Parallel and Distributed Systems, XIV(7), pp 701-708, 2003.

[21] Lucas K. T., "The Gossiping Algorithm for OTIS *k*-Ary *n*-Cube Parallel Computer", In Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, pp 253-257, 2008.

[22] E. Horowitz, S. Sahni and S. Rajasekaran, Fundamentals of Computer Algorithms, Galgotia Publications Pvt. Ltd,, New Delhi, 2002.

[23] S. G Akl, Parallel Sorting Algorithm, Academic Press, Orlando, FI, 1985.

[24] J. D. William, "Perofrmance Analysis of *k*-Ary *n*-Cube Interconnection Networks", IEEE Transaction on Computers, XXXIX, pp 775-785, 1990.

[25] K. Day, "Optical Transpose *k*-ary *n*-cube Networks", Journal of System Architecture, 50, pp 697-705, 2004.

[26] S. F. Nugent, "The iPSC/2 Direct-Connect Communication Technology", In Proceedings of the International Conference on Hypercube Concurrent Computers and Applications, pp 51-60, 1998.

[27] B. Vanvoorst, S. Seidel and E. Barsez, "Workload of an iPSC/860", In Proceeding of the Scalable High-Performance Computing Conference, pp 1-31, 1994.

[28] M. Noakes et. al., "The J-Machine multicomputer: an architectural evaluation", In Proceedings of the International Symposium on Computer Architecture, pp 224-235, 1993.

[29] R. E. Kessler and J. L. Schwarzmeier, "CRAY T3D: A New Dimension for Cray Research," CompCon, Housten, Texas, pp 176-182, 1993.

[30] E. Anderson, J. Brooks, C. Gassl, S. Scott, "Performance of the Cray T3E", In Proceedings of the ACM/IEEE Conference on Supercomputing, San Jose, California, pp 1-7, 1997.

[31] K. T. Lucas, "Parallel Algorithm for Prefix Computation on OTIS *k*-ary *n*-cube Parallel Computer", In Proceedings of the International Journal of Recent Trend in Engineering, I(1), pp 560- 562, 2009.

Author Profile

Abdul Hannan Akhtar received the B.Sc. Engineering in Electronics and Communication engineering from Bihar College of Engineering, Patna (at present National Institute of Technology, Patna) in 1988. He obtained M. E. in Electronics and Communication Engineering from Birla Institute of Technology, Mesra, Ranchi in 1998. He was sponsored by Govt. of Bihar for his M. E. program at BIT, Mesra. Presently, he is working as Lecturer (Selection Grade) in the department of Electronics and Communication Engineering at Government Polytechnic, Ranchi under the Department of Science & Technology, Government of Jharkhand.

Keny Thomas Lucas obtained his B. E. in Electronics and Communication Engineering and M. Tech. in Computer Engineering from Birla Institute of Technology, Mesra, Ranchi. He received his Ph.D in Computer Science and Engineering from Indian School of Mines, Dhanbad. He has rich experience of more than 22 years and has worked in various industries and academic institutions. He worked in Public Sector Undertakings - Projects and Development India Limited (PDIL) and National Hydroelectric Power Corporation (NHPC). He was also associated with Price Waterhouse Coopers and was responsible for enterprise-wide system integration. He has taught at various institutions of national repute - BIT, Mesra, XISS, Ranchi and NIFFT, Hatia. He has contributed many articles in different international journals and conferences and is also a reviewer of international journals and conferences.