RESEARCH PAPER

Dynamic Load Balancing Algorithm on Tor-V-Cube Parallel Interconnection Network for High Performance Computing

Rashmita Padhi
Post Graduate Department of Computer Science and Applications, Utkal University, Bhubaneswar, India, e-mail: rashmita.padhi@utkaluniversity.ac.in

ABSTRACT
The parallel interconnection network is one of the most important aspect considerations in the design of High Performance Computers (HPC) in almost all fields. It serves as the backbone over which the various components of the HPC operate simultaneously. The HPC system will implement efficiently if perfect load balancing is carried out. In this paper the load balancing analysis to crank out the significance of an efficient hybrid parallel network topology that is Tor-V-Cube or TVC. The dynamic load balancing algorithms are developed and analysed for the proposed parallel interconnection network topologies. In the current re-search work, the load balancing technique is chosen to be explored as one of the common platform to measure the performance aspects of the proposed interconnection networks TVC. As a solution to the load balancing problem on TVC network, a new procedure is proposed, analysed, and tested to evaluate the efficiency. This paper provides a brief description of different aspects of load balancing. This is followed by a brief identification of the load balancing problem. The proposed algorithm is fundamentally based on the Dimension Exchange Method (DEM) for load balancing on hypercube

Keywords: Interconnection Topology, High Performance Computing, Message Traffic Density, Cost Effective Factor, Time Cost Effective Factor.

How to cite this article: Rashmita Padhi. Dynamic Load Balancing Algorithm on Tor-V-Cube Parallel Interconnection Network for High Performance Computing. Int J Drug Deliv Technol. 2026;16(77s):1377-1385. DOI: 10.25258/ijddt.16.77s.160

Introduction

Science and Technology are embedded into every aspect of life. With advancement in Internet availability, the demand for faster computing has grown exponentially [47]. The concept of big data has arisen and a huge volume of data is gathered [2, 13, 19 and 68]. It is growing with a pace that is faster than any other measures. Behind every scenario there is a high demand for massive parallel system [3-9, and 25]. The multiprocessor systems are intended to yield faster solutions. But there are certain issues like routing, faults like link failure or node failure, availability, scalability etc which affect the efficiency of the computing system [3 and 8]. The massive computing needs of big data has outpaced the ability of most computers to process it [68]. To cater to the need of faster computing parallel systems are extensively studied [12, 25, 31 and 41]. The hypercube is the robust interconnection behind a parallel computing system [36]. To cater to this need, several variations of hypercube are also suggested [92]. The n-star is also a large-scale interconnection and sufficient research has been done to suggest star variations [8 and 13]. Continuing with this trend few new interconnection networks are proposed in this thesis. They are Tor-varietal cube, Leafy-Tor-cube and Tetrahedral hypercube etc. The growth of these networks is slow in comparison to the star derivatives but they possess enhanced packing density at very low node degree. The current chapter studies the applicability of the new structure in large scale parallel computing as message communication is a crucial aspect of parallel computing [74-76 and 93-94]. The higher rate of computation to communication ratio is always beneficial [61]. In a parallel processing environment, all are concerned with how this ratio changes with increase in either the number of Processing Elements (PE), the size of the problem or both parameters. Extensive experimentation with benchmark problems show that the ratio raises slowly with an increase in dataset size and decreases as PEs get added on [41 and 46]. In real life scenarios the problem size is growing tremendously and the number of processors also need to scale up. The performance of the system need to be meticulously monitored as it may lead to increasing inefficiencies as there are more communications among the processors. Hence the data set size must be scaled proportionately with number of PEs to keep the fraction of time in communication fixed.

An ideal computing system increases its capacity linearly with addition of hardware and /or users leading to a horizontal scalability [3 and 83]. In the scaled environment if there is a fault and the system is not disrupted by the inactive faulty PE, then the system is called redundant. The horizontal scalability and redundancy are achieved through load balancing. The load balancing aims to optimize resources, maximize throughput, minimize response time and evades overload of any single PE [4, 8, 17, 24 and 37]. This results in enhanced reliability and availability [145 and 153]. The Load Balancer (LB) is a critical component of any distributed and parallel system. It helps to spread traffic across a set of PE's to improve responsiveness and availability. The LB's can retain the status of load/resources allocated to each PE while distributing the tasks [86 and 95]. If any PE is not available to take up a new task or is not acknowledging or has initiated an error / fault signal, then the LB will stop forwarding traffic to such a node.

In turn it will search the nearest PE to be assigned. If there is a clustered organization of the PEs, then the load is evenly balanced within the respective clusters and then the surplus loads are balanced through inter cluster load transfers [89 and 64].

This paper is as follows: the preliminaries are discussed in the Section 2. The Section 3 is devoted to propose Load balancing algorithms for the proposed topologies. The Section 4 discusses about the results obtained. The concluding remarks along with scope for future work are placed in the Section 5.

2. Background

In the current re- search work, the load balancing technique is chosen to be explored as one of the common platform to measure the performance aspects of the proposed interconnection networks. As a solution to the load balancing problem on TVC networks, a new procedure is proposed, analysed, and tested to evaluate their efficiency. This section provides a brief description of different aspects of load balancing. This is followed by a brief identification of the load balancing problem. The proposed algorithms are fundamentally based on the Dimension Exchange Method (DEM) for load balancing on hypercube [3].

2.1 Basics of Load Balancing

One of the most important problems to be solved for any interconnection network in a parallel processing system is load balancing. The load balancing problem has been resolved using different approaches on various network topologies and architectures [4 and 8]. The objective of most research works in this area was to reduce the resulting error. An efficient algorithm for perfect load balancing on hypercube multiprocessors was suggested by Jan and Hwang [67] based on the well-known DEM. Perfect load balancing was achieved in [16] by exploiting the notion of regular distributions, which guarantee even distribution of data over the available processors, thuspreventing the error accumulation over dimensions and reducing the error to no morethan one. DEM was also applied as part of load balancing methods on other attractive inter- connection networks constructed from a number of hypercube networks [64]. better performance isexpected to be shown by DEM in terms of execution time, where no global information collection is required.

Task Dispersal in Multiprocessing

The problem of dispersing tasks evenly over multiprocessors is often called token distribution problem [17 and 24] provided the following criteria's hold good:

i) There exists no interrelationship among the all the tasks

ii) Every single task has unit execution time.

Each processing element can be allotted either $\left\lceil\frac{N}{P}\right\rceil$ *or* $\left\lfloor\frac{N}{P}\right\rfloor$ amount of tasks, where N is the total number of tasks to be distributed over the Interconnection network with packing density P.

Perfect Load Balanced

The n-dimensional Hypercube is perfectly load balanced if the tasks assigned to both the (n-1) sub cubes are either equal or differ by 1 [16-17 & 67].

Dimension Exchange Method

The *n*-dimensional hypercube has *n* number of adjacent nodes and for load distribution the tasks are distributed across the *n* links starting from 0 to *(n-1)* [16].

Regular Task Distribution

If there are *N* number of tasks to be distributed over a parallel system with P number of pipelined processors, then the total number of tasks that can be allotted to a processor P_i $(0 \le i \le P)$ is termed as work load denoted as W_i and the same is defined as

$$|W_i| = \begin{cases} \lceil N/P \rceil, & if\, i < (N mod P) \\ \lfloor N/P \rfloor, & otherwise \end{cases}$$

2.2 Load Balancing of Proposed Networks

This section tries to propose load balancing algorithm for the proposed interconnection networks that is TVC network. Here three dynamic dimension exchange load balancing algorithms is proposed for TVC network in this section using the following notations:

Used Notations

V:	The set of vertices of the network
E:	The set of edges of the network
m:	The dimension of the network
R_m:	m^{th} root node of the network
C_n:	n^{th} child node of the network
P_i	i^{th} cube node of the network , where i varies from 0 to 2^m
IP_L	Load of the individual processor of the network
FP_L	Final Load of the network
ID_L	Ideal Load of the network
$L(P_i)$	Load of the i^{th} processor
T_i	i^{th} task assigned to processor
R_i	i^{th} root node of the network where i varies from 0 to $2^m - 1$

L_j	j^{th} leaf node of the network where j varies from 0 to m-1
$L(R_i)$	Load of the i^{th} root node of the network
$L(L_j)$	Load of the j^{th} leaf node of the network
P_l	Partial Load
R_l	Remainder Load
LIF	Load Imbalance Factor

2.3 Load Balancing in TVC

This section tries to propose load balancing algorithms for the Tor-Varietal-Cube (TVC) interconnection network. TVC is the union of two networks that is Tor-cube and Varietal-cube. As the base network of the TVC is varietal cube and two additional links are added to the base structure. The proposed algorithm is the modified version of the dynamic dimension exchange algorithm for load distribution. Additional steps are incorporated to balance the cube nodes while balancing the entire network.

Dynamic Load Migration Algorithm of TVC

The dynamic load migration technique (DLM) for Tor-Varietal-Cube interconnection network is a performance analysis algorithm. It is dynamic as there is no prior knowledge of the assigned load. DLM algorithm takes numerous numbers of tasks to give least load imbalance and, as a result, Load Imbalance Factor (LIF) is reduced. A DLM method for multiprocessor systems is examined. The advantage of the DLM algorithm is that it reduces the execution time and LIF even for numerous tasks in a homogeneous system and uses on TVC interconnection network. TVC is built by linking every node to a node utmost from it. The node count, degree and diameter in TVC network is 2^k, $k + 2$ and $\left\lceil \frac{2k}{3} \right\rceil - 1$ respectively where k is the dimension of the network. The load balancing algorithm for the TVC network for independent task scheduling is as follows:

Algorithm: Dynamic Load Migration in TVC(V, E, m)

Input: *Processor [P_1 ------------ P_{2^m}]*
Task [T_0 ----------------- T_k]

Output: *TVC Network with balanced load processor with minimum execution time for balancing the network*

Procedure

Start

1. *Scheduler assigns random loads or tasks to each processor in the network*
2. *Scheduler calculates total_load and ideal_load*
3. *Check the maximum_overloaded_Processor (MOP) and maximum_ underloaded_Processor (MUP)*
4. *Checks connectivity between MOP and MUP in TVC interconnection network*
5. *Starts execution time.*
6. *Migration of load takes place from MOP to MUP processor among all the cube nodes which are reachable by one hop distance using adjacency look up table using dynamic dimension exchange method*
7. *Scheduler calculates final_total_load*
8. *Repeats steps 3—8 until loads at all nodes are either equal or differ by 1*
9. *Ends execution time.*
10. *Compute Load Imbalance Factor (LIF)*

Stop

Illustration of Load Balancing in TVC

In the following table the load balancing of TVC of dimension 2 is illustrated. The initial allotment of loads and load migration of each phase is shown in the following diagram. The number of migrations is more in each phase of balancing due to the addition of extra torus links in TVC architecture.

Table 1 : Load Migration in TVC_3

Processor	IPL	Adjacency Look Up	Load Migration	FPL
P_{000}	45	{ 001,010,101,111 }	45-5	40
P_{001}	50	{ 000,10,011,110 }	50-11	39
P_{010}	34	{ 000,011,110,101 }	34+5	39

P_{011}	29	{ 001,010,111,100 }	29+11	40
P_{100}	45	{ 001,101,110,011 }	45-5	40
P_{101}	50	{ 000,111,100,010 }	50-11	39
P_{110}	34	{ 100,111,100,001 }	34+5	39
P_{111}	29	{ 101,110,011,000 }	29+11	40
Total	**316**			**316**

Ideal Load $= \frac{Total\ Number\ of\ Loads}{Total\ Number\ of\ Nodes} = \frac{316}{8} = \mathbf{39.5}$

(a)

(b)

s

(c)

Figure 1: (A) Initial Allotment, (B)Phase 1 of Load Migration (C)Final Load Distribution in TVC_3

3. Results and Discussions

This section attempts to justify the suitability of TVC topology as candidates for massive scale parallel system. As they retain the basic structure of Hypercube so its performance is compared with Hypercube [36], Varietal cube [42-43], Tor-cube [122] and the Leafy-cube [98 and 115].

The main objective of this section is to test the efficiency of the proposed dynamic algorithms for load balancing on the new networks of different dimensions. The major two performance metrics like load-imbalance factor (LIF) and loads per processing elements give an indication of the algorithm's efficiency. The dimension exchange algorithm is developed for hypercube, but in case of the proposed networks dynamic load migration method is used. To produce the comprehensive experimental work, simulation is carried out. The experiments of the proposed load balancing methods under study were performed on Processor 12th Gen Intel(R) Core(TM) i7-1255U, 1700 Mhz, 10 Core(s), 12 Logical Processor(s) with 16 GB of RAM and 512GB SSD. The proposed algorithms were developed using Python programming language. The experimental work was conducted under Windows 11 operating system. The load balancing process is implemented using the multithreading approach, which is used to support the parallel execution of the implemented load balancing methodologies. In particular, ProcessPoolExecutor was used to implement multiprocessing. In executor class the max_worker is equal to the maximum number of processors in the network. Depending upon the node connectivity the adjacency list is also prepared which makes the load balancing process effective through task migration. The experiments were done for networks of different sizes by varying the dimension ranging from 4 to 10.

Discussions on Load Balancing Algorithm Time Complexity

The time complexity of the load balancing algorithm for TVC network run for 2^n nodes is same as that of the hyperube as both of them pack same number of computing nodes. Hence the time complexity is (N log P) where $P=2^n$.

Figure 4: Lode per Processing Elements in HC, LC, TC, TVC, LTC and THC Networks

After deciding the time complexity of Load balancing algorithm attempts are made to establish the superiority through comparison with its parent network that is Hypercube. The load per processing element is first calculated and compared for the possible number of tasks depending upon the classification of different size of the system ranging from small scale to very largescale computing as discussed in previous chapter.

The comparison in Fig. 4, shows that the LTC network is quite evenly balanced for the entire range of possible task starting from fifty to fifty thousand. Due to higher number of nodes in LTC network at same dimension than the Hypercube, Tor-Cube and TVC the computing nodes are perfectly balanced. Load per node is quite less and remains consistent with increase in number of tasks over time irrespective of network dimension. The THC topology remains in between both of these. The figure shows three curves as the LC and LTC have same values in terms of number of nodes and therefore coincide. Similarly, the curves of HC, TC and TVC also coincide.

Again in Fig. 5, the comparison of time complexity of proposed load balancing algorithms is done against the network dimension. Here the value is ranging from 2 to 10 as shown in the horizontal axis of the graph in the figure. The LTC network is reasonably load balanced at higher dimensions in comparison to Hypercube, TC and TVC network. Also, it is more or less close to LC network at higher dimensions. However, it is taking less time at lower dimensions. The THC topology serves in the middle range in comparison to HC, TC, TVC and leafy hybrids.

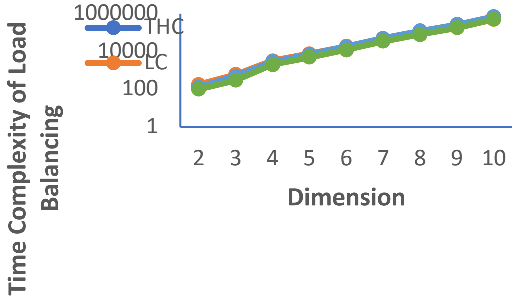

Figure 5: Comparison of Load against Dimension in HC, LC, TC, TVC, LTC and THC Networks

4. Conclusion

The current digital era has witnessed an explosive growth rate in volume of data. The VLSI technology has realized improved trends in processing and memory access speeds as a result the parallel processing is applied to the handling of extremely large data sets [14, 69]. Thus, the parallel interconnection networks are most suitable for implementing the Bigdata systems [2, 13, 19 and 68]. The current work has considered the Tor varietal cube, Leafy tor cube and Tetrahedral Hypercube for load balancing study as these three topologies exactly resemble the Hypercube. The TVC has slower growth rate in comparison to THC and LTC. Leafy Tor-Cube network and THC are suitable for massive scale computing. The dynamic load balancing algorithms are designed for the proposed networks, analysed and experimented. The results so that the algorithms perform well and converged. A detail explanation is also provided on these networks. The efficiency of the proposed network topologies is shown through comparison with parent networks. The parameters considered are lodes per node, execution time complexity and load imbalance factor. It reveals that the LTC topology balances a huge number of processing elements in a short time. Also, the nodes balanced in each phase of load migration is better in the proposed networks due to higher number of node disjoint paths.

References

1. A. Chauhan, A. Punhani, and Nitin, "Comparative analysis of traffic patterns on center connected topologies based on burton normal form," in *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, 2016, vol. 2016.
2. A. Fernandez, S. del Rio, V.Lopez, A.Bawakid, M.J.del Jesus, J.M. Benitez and F.Herrera, "Big data with cloud computing: an insight on the computing environment, MapReduce, and programming frameworks", *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Vol. 4, no. 5, pp. 380-409,2014
3. A. Grama, A. Gupta, G. Karypis and V. Kumar : *Introduction to Parallel Computing*, Pearson Education, (2003).
4. A. Panwar, A. Singh, A. Dixit and G. Parashar, "Cloud Computing and Load Balancing: A Review," *International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES)*, Greater Noida, India, 2022, pp. 334-343, doi: 10.1109/CISES54857.2022.9844367.
5. A. Punhani, P Kumar, "A horizontal fat mesh interconnection network", *Tenth International Conference on Contemporary Computing (IC3)*, pp. 1-5, 2017.
6. A. Punhani, P. Kumar, and N. Nitin, "Routing for Center Concentrated Mesh," *Int. J. Intell. Eng. Syst.*, Vol. 10, no. 1, pp. 86– 94, Feb. 2017.
7. Akash Punhani, Pardeep Kumar and Nitin Nitin, "Three-Dimensional Topology based on Modified Diagonal Mesh Interconnection Network", *Journal of Telecommunication, Electronic and Computer Engineering*, Vol 9, no 3-6, pp. 1-6, 2017.
8. Alireza Sadeghi Milani, Nima Jafari Navimipour, "Load balancing mechanisms and techniques in the cloud environments: Systematic literature review and future trends", *Journal of Network and Computer Applications*, Volume 71, 2016, Pages 86-98, ISSN 1084-8045, 10.1016/j.jnca..06.003, 2016.
9. Ankita Ananya Mohanty and Nibedita Adhikari, "Folded Leafy Cube: A Derivative For High Performance Computing", *Proceedings of Industry Interactive Innovations in Science, Engineering & Technology* (I3SET2K19), Available at SSRN: https://ssrn.com/abstract=3517173 or http://dx.doi.org/10.2139/ssrn.3517173 January 10, 2020.
10. Antoine Bossard, Keiichi Kaneko and S. Peng, "A Node-to-set Disjoint-path Routing Algorithm in Meta-cube", *10th Int. Symposium on Pervasive Systems, Algorithms and Networks*, pp. 23-28, 2010.
11. Behrooz Parhami and Mikhail Rakov, "Perfect Difference Networks and Related Interconnection Structures for Parallel and Distributed Systems", *IEEE Transactions on Parallel and*

Distributed Systems, Vol. 16, no.8, Aug 2005.

12. Behrooz Parhami, "Introduction to Parallel Processing Algorithms and Architectures", *Plenum Series in Computer Science*, Kluwer Academic Publishers, 2002.
13. By Yunquan Zhang, Ting Cao, Shigang Li, Xinhui Tian, Liang Yuan, Haipeng Jia, and Athanasios V. Vasilakos, "Parallel Processing Systems for Big Data: A Survey" 0018-9219 Ó 2016 IEEE.
14. C Vetanovic,"Performance Analysis of the Alpha21364-Based HP GS1280 Multiprocessor," *In Proceedings of 30th Annual International Symposium on Computer Architecture*, pp. 218-228, 2003.
15. C. P. Chang, T. -Y. Sung, and L. -H Hsu, "Edge Congestion and Topological Properties of Crossed Cubes", *IEEE Transactions on Parallel Distributed System*, Vol. 11, no. 1, pp. 64-80,2000.
16. C. P. Chang, T.Y. Sung and L.H. Hsu, "Edge Congestion and Topological Properties of Crossed Cube", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 11, no. 1, pp.64-80, Jan-2000.
17. C. S. Raghavendra, S. Chalasani and R. V. Boppana, "Improved algorithms for load balancing in circuit-switched hypercubes", *Proceedings. The Fifth International Parallel Processing Symposium*, Anaheim, CA, USA, 1991, pp. 537-542, doi: 10.1109/IPPS.1991.153833,1991.
18. C. Yao, K. Li, K. Lin and Y. Shen, "Load Balancing on the Exchanged Hypercube," *2009 Fourth China Grid Annual Conference*, Yangtai, China, pp. 32-35, doi: 10.1109/ChinaGrid.2009.8. 2009.
19. Chienhua Chen, D. P. Agrawal and J. R. Burke, "A class of hierarchical networks for VLSI/WSI based multicomputers", *Proceedings. Fourth CSI/IEEE International Symposium on VLSI Design*, New Delhi, India, , pp. 267-272, doi: 10.1109/ISVD.1991.185128, 1991.
20. Chih-Fong Tsai , Wei-Chao Lin , Shih-Wen Ke "Big Data Mining with Parallel Computing: A Comparison of Distributed and MapReduce Methodologies", *The Journal of Systems & Software*, doi: 10.1016/j.jss.2016.09.007, 2016.
21. Cray Inc., "Cray X1E Supercomputer," http://www.cray.com/downloads/ X1E_datasheet.pdf, 2008.
22. Cray Inc., "Cray XT4 Datasheet," http://www.cray.com/ downloads/Cray_XT4_Datasheet.pdf, 2008.
23. D. B. West, "Introduction to Graph Theory", Second Edition, Prentice Hall, 2001.
24. D. Wang, "Hamiltonian Embedding in Crossed Cubes with Failed Links," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 11, pp. 2117-2124, Nov. 2012, doi: 10.1109/TPDS.2012.30.
25. Dalia Abdulkareem Shafiq, N.Z. Jhanjhi, Azween Abdullah, "Load balancing techniques in cloud computing environment: A review", *Journal of King Saud University - Computer and Information Sciences*, Volume 34, Issue 7, pp. 3910-3933, ISSN 1319-1578, 10.1016/j.jksuci.2021.02.007, 2022.
26. David Padua, "Encyclopedia of Parallel Computing", Springer Science, 2011
27. Ding-Ming Kwai and Behrooz Parhami, "Tight Bounds on the Diameter of Gaussian Cubes", In *The Computer Journal*, Vol 41, No 1, pp. 52-55, 1998.
28. E. Abuelrab, "Embeddings into Crossed Cubes", *Proceedings of the World Congress on Engineering*, WCE2009, vol. I, July 1-3, 2009.
29. E. Szlachcic, "Fault Tolerant Topological Design for Computer Networks", In *Proceedings of the International Conference on Dependability of Computer Systems (DEPCOS-RELCOMEX)*, pp. 150-159,2006.
30. F. Harary, "Conditional Connectivity", *Journal of Networks,* Vol. *13*, pp.347–357, 1983.
31. F. Nasir, A. Samad and J. Siddiqui, "Topological Analysis of Cube Based and Linearly Extensible Networks Using MATLAB", In *Inventive Communication and Computational Technologies. Lecture Notes in Networks and Systems*, Vol 311, pp 107-119, 2022.
32. F. T. Leighton, "Introduction to Parallel Algorithm and Architectures: Arrays, Trees, Hypercubes", Morgan Kaufmann, 1992.
33. F.J.Andujar-Munoz,J.A.Villar-Ortiz,J.L.Sanchez,F.J.Alfaro,and J. Duato, "N-Dimensional Twin Torus Topology," *IEEE Trans.Comput.*, Vol. 64, no. 10, pp. 2847–2861, 2015.

34. Fibonacci Cubes with Applications and Variations, Ömer Eğecioğlu (University of California at Santa Barbara, USA), Sandi Klavžar (University of Ljubljana, Slovenia), and Michel Mollard (CNRS Université Grenoble Alpes, France), 2023.
35. G. Coulouris, J. Dollimore, T. Kindberg and G. Blair, "Distributed systems: concepts and design", 5th Edition. Addison-Wesley.
36. G. H. Chen, J.S. Fuand J. F. Fang, "Hypercomplete : A Pancyclic Recursive Topology for Large Sacle Multicomputer Systems", *Networks*, Vol.35, no.1, pp.56-69,2000.
37. G. Wang, C. -K. Lin, J. Fan, B. Cheng and X. Jia, "A Novel Low Cost Interconnection Architecture Based on the Generalized Hypercube," in *IEEE Transactions on Parallel and Distributed Systems*, Vol. 31, no. 3, pp. 647-662, 1 March 2020, doi: 10.1109/TPDS.2019.2941207.
38. Gene Eu Jan and Yuan Shin Huang, "A Simple Algorithm for Optimal Load Balancing on Hypercube Multiprocessors", in *Proceedings of 2001 International Conference on Parallel and Distributed Processing Techniques and Applications*, At: Las Vegas, Nevada, USA
39. H. Barik, N. Swain, L. Rout, and N. Adhikari, "Double Star: A high performance network for big data", *SSRN eLibrary* CTFC 2019, January 10, 2020.
40. H. -C. Hsu, C. -H. Tsai, G. -J. Wu, S. -H. Liu and P. -L. Lai, "Hierarchical Crossed Cube: A New Interconnection Topology", In *Proceedings of 25th Workshop on Combinatorial Mathematics and Computation Theory*, Taiwan, pp. 70-79, 2008.
41. H. Sim, JC. Oh and HO Lee, "Multiple Reduced Hypercube *MRH*(*n*): A New Interconnection Network Reducing Both Diameter and Edge of Hypercube", In*Grid and Distributed Computing. Communications in Computer and Information Science,* Vol 63. Springer, Berlin, Heidelberg. 10.1007/978-3-642-10549-4_23 GDC 2009.
42. H.J. Siegel, "Interconnection Networks For Large Scale Parallel Processing", 2nd Edition, McGraw Hill, New York 1990.
43. Hao-shun Hung, Jung-Sheng Fu and Gen-Huey Chen, "Fault Free Hamiltonian Cycles in Crossed Cubes with Conditional Link Faults", *Journal of Information Sciences*, Vol. 177, pp.5664-5674, 2007. https://doi.org/10.1016/j.jcss.2017.04.002.
44. Huimei Guo, Eminjan Sabir, Aygul Mamut, "The g-extra connectivity of folded crossed cubes", *Journal of Parallel and Distributed Computing*, Volume 166, , Pages 139-146, ISSN 0743-7315, 2022.
45. J. Bahman, H. Abawajy Jemal and Mohammad K. Akabari, "Performance Modelling and Analysis of Heterogeneous Meta-Computing Systems Interconnection Network Systems", *Journal of Computers and Electrical Engineering*, Vol. 34, no.6, pp. 488-502, 2008.
46. J. Cao , L. Xiao and J -M Xu, "Cycles and paths embedded in varietal hypercubes", *J UnivSci Technol China*, 2014, 44(9): 782–789
47. J. Dean and S. Ghemawat, "Map reduce: a flexible data processing tool", *Communications of the ACM*, Vol. 53, no. 1, pp. 72-77, 2010.
48. J. Fan, "Hamiltonian Connectivity and Cycle-embedding of Mobius Cubes", *Information Processing Letters*, Vol. 82, pp. 113-117, 2002.
49. J. Kim, J. Das, C. R. Das, W. Lin and T.-Y. Feng, "Reliability evaluation of hypercube multicomputers," in *IEEE Transactions on Reliability*, vol. 38, no. 1, pp. 121-129, April 1989, doi: 10.1109/24.24585.
50. J. M Xu, "Combinatorial Theory in Network", Beijing: Science Press, 2013
51. J. M. Kumar and L.M Pattnaik, "Extended-hypercube: A Hierarchical interconnection network of hypercubes", *IEEE Trans: Parallel and Distributed Systems.*, Vol-3, no. 1, pp. 44-57 Jan.(1992).
52. J. M. Xu, "Theory and Application of Graphs", Kluwer Academic Publishers, Dordrecht, London 2003.
53. J. M. Xu, "Theory and Application of Graphs"; Dordrecht/Boston/London: KluwerAcademic Publishers, 2003.
54. J. M. Xu, "Topological Structure and Analysis of Interconnection Networks", Kluwer Academic Publishers, Dordrecht, London 2001.
55. J. M. Xu, M. Ma and Min Lu, "Paths in Mobius Cubes and Crossed cubes", *Information Processing Letters*, Vol. 97, pp. 94-97, 2006.

56. J. Miguel-Alonso, J. Navaridas, and F. J. Ridruejo, “Interconnection Network Simulation Using Traces of MPI Applications”, *International Journal Parallel Programming*, Vol. 37, no. 2, pp. 153-174, 2009.
57. J. Paul and Howard E. Michel, “Computer Systems Performance, Evaluation and Prediction”, Burlington Digital Press, 2003.
58. J. S. Fu and Gen-Huey Chen, “Fault Tolerant Cycle Embedding in Hierarchical Cubic Networks”,*Networks*, vol. 43, no.1, pp.28-33, 2004.
59. J. S. Fu and Gen-Huey Chen, “Hamiltonocity of the Hierarchical Cubic Network”, *Journal of Theory of Computing Systems*, Vol. 35, no.1, pp.59-79, 2002.
60. J. S. Fu, “Fault Tolerant Cycle Embedding in the Hypercube”, *Parallel Computing*, Vol. 29, no.6, pp. 821-832, 2003.
61. J. S. Fu, “Longest Fault-free Paths in Hypercubes with Vertex Faults”, *International Journal of Information Sciences*, Vol. 176, pp. 759-771, 2006.
62. J. S. Fu, G. H. Chen, and D. R. Duh, “Node-disjoint Paths and Related Problems on Hierarchical Cubic Networks”, *Networks*, Vol. 40, no. 3, pp. 142–154, 2002.
63. J. S. Kim, H. O. Lee, M. Kim, and S. W. Kim, “The new Petersentorus networks,” *Journal Supercomputer*, Vol. 71, no. 3, pp. 894–908, 2015.
64. J. W. Jie Wu, "Dimension-Exchange-Based Load Balancing on Injured Hypercubes," *1994 International Conference on Parallel Processing Vol. 3*, North Carolina, USA, pp. 88-91, 1994, doi: 10.1109/ICPP.1994.90.
65. Jianxi Fan, X. Jia and X. Lin, “Complete Path Embeddings in Crossed Cubes”, *Int. Journal of Information Sciences*, Vol.176, pp. 3332-3346, 2006.
66. Jianxi Fan, X. Jia and X. Lin, “Optimal Path Embedding in Crossed Cubes”, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 16, no. 12, pp. 1190-1200, 2005.
67. Jianxi Fan, Xiaola Lin and Xiaohua Jia, “Node Pancyclicity and Edge Pancyclicity of Crossed Cubes”, *Information Processing Letters*, Vol.93, pp.133-138, 2005.
68. Jianxi Fan, Xiaola Lin, Y. Pan and Xiaohua Jia, “Optimal Fault-tolerant Embedding of Paths in Twisted Cubes”, Journal of Parallel and Distributed Computing, vol.67, no.2, pp.205-214, 2007.
69. Jinwoon Woo and S. Sahni, "Load balancing on a hypercube," *[1991] Proceedings. The Fifth International Parallel Processing Symposium*, Anaheim, CA, USA, , pp. 525-530, 1991, doi: 10.1109/IPPS.1991.153831.
70. Jin-Xin Zhou, “On g-extra connectivity of hypercube-like networks”, In *Journal of Computer and System Sciences,* Volume 88, Pages 208-219,ISSN 0022-0000, 2017.
71. Jose Duato, Sudhakar Yalamanchi li and Lionel M. Ni, “Interconnection Networks: An Engineering Approach”, Elsevier Science, USA, 2003.
72. Jywe-Fei Fang, “The Bipanconnectivity and m-Panconnectivity of The Folded Hypercube”, *Theoretical Computer Science*, Vol. 385, no.3, pp.286-300, 2007.
73. K. Bhavani and Sudarson Jena, “Exchanged folded crossed cube: A new interconnection network for parallel computation”, Information Processing Letters, Volume 137, , pp. 40-46, 2018, ISSN 0020-0190.
74. K. Efe, “The Crossed Cube Architecture for Parallel Processing”, *IEEE Transactions on Computers*, Vol. 41, no. 5, pp. 513-524, Sept.1992.
75. K. Zanbouriand N. Jafari Navimipour, “A cloud service composition method using a trust-based clustering algorithm and honeybee mating optimization algorithm”, In *Int J Commun Syst* 33(5):e4259, 2020
76. Kai Hwang and F. A. Briggs, “Computer Architecture and Parallel Processing”, MGH Int. Edition 1985.
77. Kai Hwang, “Advanced Computer Architecture”, TMH Edition 2001.
78. L. D. Wittie, “Communication Structure for A Large Multicomputer System”, *IEEE Transactions on Computers*, Vol. 30, pp. 264-273, 1981.
79. L. J. Fan, “Routing Algorithms on the Bus Based Hypercube Networks”, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 16, no.4, pp.335-348, 2005.
80. L. K. Arora and Rajkumar, “C2Mesh”, *3rd IEEE International Advance Computing Conference (IACC)*, pp. 282–286, 2013.
81. L. K. Li, C. H. Tsai, J. M. Tan and L. H. Hsu, “Bipanconnectivity and Edge-fault-tolerant Bipancyclicity of

Hypercubes", *Information Processing letters*, Vol.87, pp.107-110, 2003.

82. L. N. Bhuyan and D. P. Agarawal, "Generalized Hypercube and Hyperbus Structures for a Computer Network",*IEEE Transactions on Computers*, C-33, no. 4, pp. 323-333, 1984.
83. L. N. Bhuyan and D. P. Agarawal, "Performance of Multiprocessor Interconnection Network",*IEEE Transactions on Computers*, Vol. 2, pp.25-37, 1989.
84. L. N. Bhuyan and D. P. Agrawal, "Design and Performance of Generalized Interconnection Networks",*IEEE Transactions on Computers*, C-32, no. 4, pp.1081-1089, 1983.
85. LiuYouyao, HanJungang and Du Huimin, "A Hypercube-based Scalable Interconnection Network for Massively Parallel Computing", *JOURNAL OF COMPUTERS*, VOL. 3, NO. 10, OCTOBER 2008
86. M -J Ma and J -M Xu. "Algebraic properties and panconnectivity of folded hypercubes". *Ars Combinatoria*, 95: 179–186, 2010.
87. M Jiang, X -Y Hu, Q -L Li, "Fault-tolerant diameter and width diameter of varietal hypercubes"; Appl Math J Chinese Univ Ser A, 2010, 25(3): 372–378 (in Chinese)
88. M. A. Shahid, N. Islam, M. M. Alam, M. M. Su'ud and S. Musa, "A Comprehensive Study of Load Balancing Approaches in the Cloud Computing Environment and a Novel Fault Tolerance Approach," in *IEEE Access*, Vol. 8, pp. 130500-130526, 2020, doi: 10.1109/ACCESS.2020.3009184.
89. M. Bhardwaj, "C2Torus New Interconnection Network Topology Based on 2D Torus", *American Journal of Networks and Communications. Special Issue: Ad Hoc Networks*. Vol. 4, no. 3-1, , pp.1-4, 2015.
90. M. C. Yang, "Binpanconnectivity of Balanced Hypercubes", *Journal of Computers and Mathematics with Applications*, Vol.60, no. 7, pp. 1859-1867, 2010.
91. M. Elwekeil, M.S. Abdalzaher and K. Seddik, "Prolonging smart grid network lifetime through optimising number of sensor nodes and packet length". *IET Commun.*, *13*, 2478–2484,2019.

92.M. M. H. Rahman and S. Horiguchi, "HTN: a New Hierarchical Interconnection Network for Massively Parallel Computers", *IEICE Transactions of Information System*, E86-D, pp. 1479-1486,2003.

93. M. S. Rahman, M. Kaykobad and J. S. Firoz. "New Sufficient Conditions for Hamiltonian Paths", In *The Scientific World Journal*, Vol.2014 , pp. 1-7, 2014.
94. M. V. Rao and N. Chalamaiah, "Hypercube and Its Variant Networks: A Topological Evaluation", *Proceedings of 8th Int. Conference on High- Performance Computing in Asia-Pacific Region*, 2005.