Folded Leafy Cube -A Reliable High Performance Computing System for Big Data

[1]Sadashiba Pati, [2]Nibedita Adhikari, [3]Vinaya Singh

[1]CAPGS, Biju Patnaik University of Technology, Odisha, India

Department of MCA, Utkal University, Vanivihar, Bhubaneswar, India

Department of Computer Science, Usha Martin University, Jharkhand, India

ABSTRACT

In this paper, Folded Leafy cube (FLC) structure is derived from LC network to have better performance parameters. The Folding technique reduces the average node distance. FLC is suitable for very large scale computing system due to comparatively lesser link complexity. Different topological and performance parameters of FLC network are derived. While improving the routing time, the folding technique also enhances the reliability of the FLC topology. The computing load on each node is also balanced in order to claim the superiority of the interconnection structure. In future, attempts will be made to suggest better routing with fault tolerance to make FLC a suitable candidate for high performance and big data systems which have high computation and communication needs.

Keywords: HC, LC, PIN, Reliability, Efficiency

I. INTRODUCTION

Parallel computing has longstanding core strengths in real life applications, dependable and trustworthy systems, computer security and mobile computing as well as communication.The recent electronic devices work with multiple sensors and always stay connected to the Internet. As a result huge amount of data is captured which need to be stored. This large-scale activity and voluminous data has resulted in Big data systems. The high volume and wide variety of data need to be processed accurately at a faster rate. For these reasons the focal point of many researchers is on the parallel, multi-core computing. For many big data applications, the Parallel Interconnection Network (PIN) topology is now receiving much attention. It has become an essential subject as it can establish a critical high performance computing and message passing system. In general, these systems consist of various processing elements, memory units and other resources. These components easily interact through an interconnection network (IN) [1,2 & 3]. An interconnection network can be the backbone of Big data systems, which models the connections among the processing devices for the purpose of inter-device communication [4, 5]. The suitability of an interconnection network topology is evaluated in terms of its degree, diameter, cost, fault tolerance, fault diameter, reliability, load balancing and other such performance attributes. Apart from this, fault tolerance, reliability, cost effectiveness and time cost effectiveness are also taken into consideration while evaluating the performance of any interconnection network topology. Since the IN's provide various mechanisms for data transfer between processing nodes or between processors and memory modules, their topological features vary to a great extent. This variation leads to improvement in Parallel Interconnection Systems. The variations in the parameters are often used by researchers to suggest a new network and its suitability for practical implementation.

Among of all the existing PINs, the hypercube (HC) has been the most favored due to some of its interesting properties such as low node degree, regularity, symmetry, average node distance and routing [6 and 7]. However, in a HC with increase in network dimension the link complexity also increases. It results in depletion of the performance of the network. The network needs to pack more nodes in order to support big data systems. Various techniques have been used by different authors for modification of existing networks in order to make it scalable. To improve the performance a good deal of research is being done on cube based networks and several variations have been proposed with the help of folding, crossed links, extension, hybridization etc. and few derived topologies are Folded hypercube (FHC) [9], Crossed cube (CQ) [8], Folded cross cube (FCQ) [10], Hierarchical Hexagon, Starcrossed cube, Leafy cube (LC) [11,12 and 13]. In FHC network the improvement is achieved by connecting an extra edge called complimentary link among all the farthest nodes of a hypercube.

Leafy cube is also derived from the hypercube structure. This network is recursive in nature. Construction of higher level structure is very easy and with low link complexity it can pack a large number of nodes. The Leafy cube is already observed to have quite superior topological properties in comparison to other networks [11,12]. The leafy structure is an improved topology due to addition of extra nodes at one hop distance. This parameter of LC network makes it suitable for big data system. In leafy cube network, the work load per node and load balancing for different dimension is better as compared to parent hypercube network [6,7].

The current work tries to propose the Folded Leafy Cube by applying the folding technique to Leafy Cube similar to FHC . The Paper is organized as follows: the Section two presents the FLC network topology followed by routing in the proposed network in Section three. Then the Section four presents the performance parameters such as cost effectiveness factor, time effectiveness factor, scalability, dynamic load balancing of FLC network. In Section five Comparisons of topological parameters and dynamic load balancing of FLC with other contemporary networks are discussed. Lastly Concluding remarks are presented in Section six.

II. FLC NETWORK TOPOLOGY

The design of Leafy Cube is obtained by adding extra nodes and edges to each node of the Hypercube. The structure is not regular but it has some better characteristic such as increased packing density at very low cost as compared to the parent network. To improve the performance, the folding technique is applied to the leafy cube and a new topology is designed called the Folded Leafy Cube (FLC) [14]. Some extra complementary links are introduced to improve the network topology while retaining the original structure as it is. Unlike the folded hypercube, here the folding technique is applied only to the leaf nodes to increase the node degree of leaf nodes at each level. The node degree of root nodes or the cube nodes remains the same as the Leafy Cube. The aim is to study the performance parameters of the proposed network similar to HC and LC [1,6,7,11,12]. In the structure additional *n* numbers of links are attached to each cube nodes. Then only the farthest nodes are connected through complementary edges as shown in the figures below. The dashed lines are the complementary edges that connect to the farthest nodes in the network. The

resulting topology is recursive and hierarchical in nature. The new structure of dimension 2 and 3 are shown in Fig. 1 and 2 respectively.All root nodes in FLC are addressed with 1 and 0 binary number system similar to hypercube and leaf nodes are numbered 0 to (n-1). Let s and d be farthest nodes, S(00, 1) or (00, 0) and D (11, 0) or (11, 1). Farthest node is decided basing on Hamming distance by doing XOR operation on root nodes. The farthest node in hypercube is at hamming distance n, where n is the network dimension. In FLC network the leaf nodes of farthest roots are directly connected through complementary edges (dashed lines). To have clarity in picture, few complementary links are drwan in both the figures.

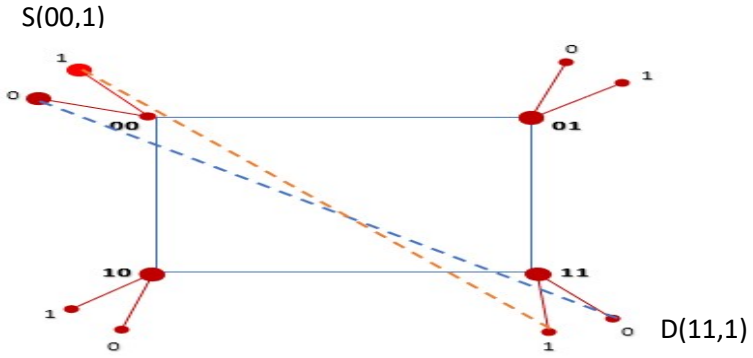

Figure 1: Folded Leafy Cube of degree 2, FLC (2) with Node Addressing

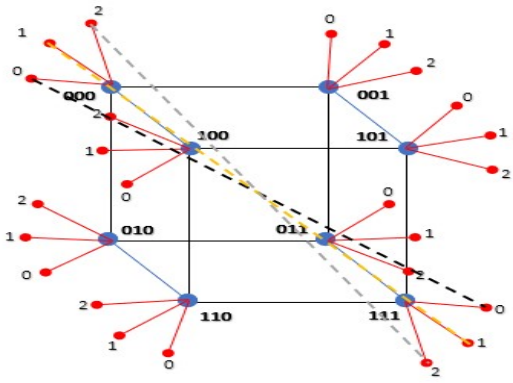

Figure 2: Folded Leafy Cube of degree 3, FLC (3)

2.1 Topological Properties of Folded Leafy Cube

This section briefly describes the various topological properties of the proposed network which are already derived [14]. For convenience the properties are stated without proof.

Degree

Property 1*: The degree of Folded leafy cube of dimension 'n' ie FLC(n) is '2n'.*

Total number of nodes

Property 2: *The total number of nodes in FLC (n) is* $V = 2^n(n+1)$.

No of Edges

Property 3*: The edges of FLC(n) is,* $|E| = n\ 2^{n+1}$

Diameter

This is defined as the minimum distance existing between any two farthest nodes.

Property 4*: The diameter of FLC(n) is d = n*

Cost

In parallel network the cost factor is calculated as the product of the diameter and the degree of the nodes. This factor is widely used in performance evaluation as it indicates the cost of travelling or message passing in the network.

Property 5*: The cost of FLC(n) is* $2n^2$.

Bisection Width

It is defined as the number of edges whose removal will result in two distinct sub networks.

Theorem 1: *The bisection width of the FLC(n)* is $(n+1)2^{n-1}$.

Proof: From the construction of the FLC it is clear that the additional leaf edges do not alter the Hypercube structure. Removal of cube edges will result in two distinct subset along with the leaf nodes when all the complementary edges are removed. So the bisection width of the Folded leafy cube is same given as $(\frac{2^n}{2} + n \times \frac{2^n}{2}) = (n+1)2^{n-1}$.

Average Node Distance

Theorem 2:*The average node distance in FLC(n) is* $\bar{d}_{avg} = \frac{n}{2}$.

Proof: In the proposed network the basic structure is LC but this network has one property that all the additional leaf nodes are travelled in one step. The overall network structure is same as hypercube as well as Folded hyper cube. Thus in the FLC(n) network the final average node distance is $\bar{d}_{avg} = \frac{n}{2}$.

Message Traffic Density

It is the measure of maximum traffic that can be handled in message passing scenario inside a network.

Theorem 3 :*The message traffic density of FLC(n) network is* $\rho = \frac{(n+1)}{4}$.

Proof : The message traffic density is specified as = $\frac{d_{avg}\,N}{E}$, where N is the total number of nodes and E is total number of edges in the network. In this network the basic structure is

LC and the message traffic density is similar to the Leafy cube. From Property 2, N=2^n(n+1) and the Property 3, E = n× 2^{n+1}

So, = $\frac{\bar{d}_{avg}2^n(n+1)}{n(2^{n+1})} = \frac{(n+1)}{4}$.

Link Complexity Growth Rate

The term "Growth rate" is used in connection to the time or space complexity of an algorithms.This paper attempts to introduce a direct measure of the growth of a function, or the growth of the complexity of a network with the size of the problem. Our goal is to give the idea of growth rate a formal definition that can be used both in complexity analysis. The HC is used to derive the FHC network with addition of complimentary links. Then LC network is designed with addition of leaf nodes and then equal number of leaf edges are added. From LC network the FLC topology is derived by applying folding technique to leaf nodes only. But here the leaf nodes are connected to their farthest nodes. Now attempts are made to study the growth rate for link complexity.

From graph theory, it is clear that the number of edges in a graph is directly proportional to the number of nodes

i.e Link∝ Node

$\Rightarrow$ *Link* = C × Node

$\Rightarrow$ C = $\frac{NODE}{LINK}$ (where C - Link complexity factor)

Chelps to decide the growth rate with respect to network dimension or size.

Table 1 illustrates the topological parameters of several cube variations like Hypercube, Crossed Cube, Folded Hypercube, Folded Cross Cube and Leafy Cube with the proposed network Folded Leafy Cube (FLC). Apart from the topological parameters, the performance parameters like fault tolerance, reliability, cost effectiveness and time cost effectiveness are also taken into consideration while evaluating the performance of any network topology, which are discussed next.

Table1: Comparison of the Topological Parameters

Parameters	HC	FHC	CQ	FCC	LC	FLC
Degree	n	n+1	n	n +1	2n	2n
Nodes	2^n	2^n	2^n	2^n	2^n(n+1)	2^n(n+1)
Diameter	n	$\lceil\frac{n}{2}\rceil$	$\lceil\frac{n+1}{2}\rceil$	$\lceil\frac{n}{2}\rceil$	(n+2)	n
Cost	n^2	$(n+1) \times \lceil\frac{n}{2}\rceil$	$\left(n \times \lceil\frac{n+1}{2}\rceil\right)$	(n+1)$\lceil\frac{n}{2}\rceil$	2n(n+2)	$2n^2$

No of edges	$n * 2^{n-1}$	$(n+1) \times 2^{n-1}$	2^{n-1}	$(n+1)2^{2n-2}$	$n2^{n-1} + n2^n$	$n2^{n+1}$
Bisection width	$\frac{2^n}{2}$	$\frac{2^n}{4}$	$\frac{2^n}{2}$	2^n	$\frac{2^n}{2}$	$(n+1)2^{n-1}$
Average Node distance	$\frac{n}{2}$	$\frac{n}{2}$	$\frac{n}{2}$	$\frac{\boldsymbol{n}}{\boldsymbol{2}}$	$\frac{n}{2}$	$\frac{n}{2}$
Mesage Traffic Density	1	$\frac{n}{n+1}$	1	1	$\frac{n+1}{3}$	$\frac{n+1}{4}$

III. ROUTING IN PROPOSED NETWORK

A routing algorithm is a set of step-by-step operations used to direct network traffic efficiently. When a packet of data leaves its source, there are many different paths it can take on its way to the destination. The routing algorithm is used to determine mathematically the shortest path all the time. This measure complements the usual time or space complexity analysis of a network. For routing, the node addressing in FLC is described in earlier section.

The Fig. 1 shows the node addressing of FLC (n) network. It is a FLC(2) network with two leaf nodes at each cube node. Let S and D be the source and destination nodes. Here shortest distance i.e. dist (S,D) =1 due to complementary edges because of folding.

3.1. Routing in FLC Network

This section attempts to propose the routing algorithms for the FLC(n). In the cube based networks, the routing process depends upon the shortest path based on Hamming distance similar to hypercube. There are four cases possible as discussed blow. Before going to algorithm these basics are discussed.

Let 'S' is source and 'D' is destination node for all four cases:

Case I: *S and D both are leaf nodes,*

Case II: *S is Leaf node and D is Cube node,*

Case III: *S is cube node and D is leaf node,*

Case IV: *Both the nodes S and D are cube nodes.*

The Routing algorithms for all the above cases are given below.

Algorithm Case - I
Para do
Step-1:If s-leaf node then
If (root -S xor Root-D) =n, then take complementary edge
else
Move to next cube node (root)
Perform Cube routing

Else if s-cube node and d-leaf node then take one hop
To reach destination node
}

In the above case source and destination nodes both are leaf nodes. Here if the hamming distance between root nodes is *n* then complementary edge is used. Otherwise if source node is leaf node then en-route the message from leaf node to parent cube node, i.e. neighbor node in one step. From source parent (cube node) cube routing will be used to reach destination parent node. Then in one hop the destination node will be reached.

Algorithm Case - II
Step-1: if s-leaf node then
{
move to next cube node (root)
Step-2:Perform Cube routing, travel n steps to reach destination node
}

Here if the source node is leaf node and destination node is cube node then route the message from the leaf node to the root node in one hop and then using simple cube routing move to destination node.

Algorithm Case – III
s is cube node and d is leaf node
Step-1: *if s-cube node then*
{
If (S xor Root_D)=n, then move to leaf node in one hop and use Complementary edge
Else
Perform Cube routing
{
Step-2:Cube node to leaf node one step to reach destination node
}}

Here source node is cube node then check for hamming distance of destination root node is n and take complementary edge. Otherwise perform cube routing to route the message from source node to destination node, or one step to reach destination.

Algorithm Case – IV
s and d both are Cube nodes
Perform cube routing (self-routing in Hypercube)

In the last case both the nodes are cube nodes so directly cube routing will be done or on the other word we can say self-routing will be done. In all the cases the time complexity of routing algorithm is O(n). In most of the cases due to complementary edges, the routing takes less than n steps.

IV. Performance Parameters

Performance analysis determines the failure or success of a project using various parameters.It is very much necessary to do performance analysis of a parallel

interconnection network as it reflects one of the imprtant aspects of a multiprocessor system that is the total cost of a system. To make the parallel interconnection network more attractive, emphasis is also given to fault tolerance and reliability analysis.

4.1. Cost Effectiveness Factor

In cube based networks the number of links is a function of the number of processors. The cost effectiveness factor takes this into account and gives more insight to the performance of the multiprocessor system. Here processor cost and communication link cost are taken into consideration.

Theorem 4: The cost effectiveness factor of the FLC (n) is $\frac{1}{1+\rho(\frac{n+1}{2})}$, where ρ is the ratio of the link to processor cost.

Proof:In general number of links is a function of the total number of nodes. In the proposed network, the no of nodes is

P= 2^n(n+1).

Total no of links is given by E = n2^{n+1} = n.2^n(2)+2^{n+1}-2^{n+1} = 2(n2^n + 2^n)- 2^{n+1}

= 2p - 2^{n+1} = *f(p)*

So g(p) = $\frac{f(p)}{p} = \frac{2(n2^n + 2^n) - 2^{n+1}}{2^n(n+1)} = \frac{2n}{n+1}$

Hence CEF(p)= $\frac{1}{1+\rho g(p)} = \frac{1}{1+\rho(\frac{2n}{n+1})}$

4.2. Time Cost Effectiveness Factor

Time cost effectiveness factor considers time for solution of a problem as a parameter for evaluating the performance. This factor considers the situations where a faster solution is more rewarding than the slower solutions. When speedup of parallel algorithms is known, the above two factors characterize the profitable use of multiprocessor systems.

Theorem 5: The TCEF of LC (n) network is given by network is given by $\frac{1+\sigma}{1+p\left(\frac{n}{2}\right)+\frac{1}{2^n(n+1)}}$.

Proof*:*The TCEF is given by TCEF(p,T_p) = $\frac{1+\sigma T1^{\alpha-1}}{1+\rho g(p)+\frac{T_1{}^{\alpha-1}\sigma}{p}}$

where T_1 is the time required to solve the problem by a single processor using the fastest sequential algorithm, T_p is the time required to solve the problem by a parallel algorithm using a multiprocessor system having *p* processors and ρ is the ratio of the cost of penalty to cost of processors.

As per Theorem above g(p) = n+1 &σ and $\propto$ are assumed to be 1.

TCEF = $\frac{1+\sigma}{1+\rho G(p)+(\frac{\sigma}{p})} = \frac{1+\sigma}{1+\rho(n+1)+(\frac{\sigma}{2^n(n+1)})}$

4.3 Scalability

The Scalability problem is receiving much attention recently. In computation intensive systems the number of computing nodes is increasing exponentially. At this stage the system architects should be aware of the possible complications that will emerge after addition of nodes into a system. The reliable performance of the interconnection network topologies is an important issue in the design of parallel computer systems. The scalability of a parallel system is a measure of its capacity to increase speedup in proportion to the number of Processing Elements (PE). It also reflects a parallel system's ability to utilize increasing processing resources effectively. The Efficiency enables us to determine the fraction of time for which the PE is usefully employed. For a specifiedmagnitude of a problem, if the number of processing elements is enlarged, then the overhead of the system or the idle time of the PE also intensifies. The scalability and the efficiency both depend on the problem size or the input size.

The Isoefficiency is defined as the ratio of speedup to the total number of processors in the system. It determines the ease with which the parallel system can maintain the constant efficiency and hence can achieve scalability. The parallel systems efficiency can be maintained at any value between 0 and 1 depending upon the network size.

The Isoefficiency (f_e) is defined as follows:

Generally the parallel run time (T_P) for 'p' processing elements and input size 'w' is defined as

$$T_p = \frac{w+T_0(w,p)}{p} \quad (1)$$

Where T_0 is the overhead of the system due to idling of the processor

The speedup (S) is defined as

$$S = \frac{W}{T_p} \quad (2)$$

The Isoefficiency $f_e(P)$ is defined as

$$f_e(P) = \frac{S}{P} = \frac{1}{1 + T_0(w,p)/w} \quad (3)$$

If a system can upholdpersistent Isoefficiency for altered problem sizes then it is supposed to be scalable.

4.4 Dynamic Load Balancing

Theload balancing is a method of dividing the total load among the processors in the distributed system to improve the task's response time. This is achieved through different Load Balancing Strategies[12,18]. For HC and LC interconnection networks Dynamic load balancing is used [18]. The Load balancing is performed by a appliance either physical or virtual that identifies in real time which PE in a pool can best meet a given client request, while ensuring heavy network traffic doesn't unduly overwhelm a single processor. In addition to maximizing network capacity and performance, Dynamic load balancing

provides failover. If one server fails, a load balancer immediately redirects its workload to a backup server, thus mitigating the impact on end users.

Load balancing plays a vital role in the operation of distributed and parallel computing. It partitioned the incoming workload into smaller tasks that are assigned to computational resources for concurrent execution. The load may be CPU capacity, memory size, network load, delay, etc. The reason behind load balancing is to handle requests of multiple users without degrading the performance of web server. Load balancer receives requests from user, determines the load on available resources, and sends request to the server which is lightly loaded. The major functions of load balancer are as follows: i) distributes incoming traffic across multiple computational resources, ii) determines resource availability and reliability for task execution, iii) Improves resource utilization, iv) increases client satisfaction, v) provides fault tolerance and flexible framework by adding or subtracting resources as demand occurs

4.4.1 Load Balancing in FLC

The algorithm proposed for Folded leafy cube is Dynamic Load Balancing with Folding (DLBF). As folding technique is applied in LC network it also helps to balance more number of nodes in single iteration. Hence it helps to balance the network sooner than the HC in spite of higher packing density. It is dynamic in the sense that no prior information of the load is assumed. The algorithm takes multiple numbers of tasks to provide least load imbalance. It evaluates the load imbalance factor (LIF) and execution time in the network for FLC network. The benefit of DLBF algorithm is that it minimizes the execution time and LIF even for large number of tasks in homogeneous system and applies on FLC network. The FLC is constructed by linking every child node to a node utmost from it. The number of nodes, degree and diameter in FLC network is 2^n(n+1), 2n and n respectively.

DLBF Algorithm

1. *Initialize Random tasks.*
2. *Calculates the load on each processor, total load and ideal load.*
3. *Finds the maximum overloaded (MOL) and maximum underloaded (MUL) processor.*
4. *Checks connectivity between MOL and MUL in Folded leafy cube network.*
5. *Starts execution.*
6. *Migration of load takes place from MOL to MUL processor.*
7. *Repeats steps 3—6 until processors become moderate.*
8. *Ends Execution Time.*

V. RESULT AND DISSCUSSION

5.1. Comparison of Topological parameters

This section attempts to illustrate the supremacy of the proposed network topology in comparison to the hypercube, crossed cube, folded hypercube, folded cross cube and leafy cube. The topological parameters

Table 2: Comparison of Nodes

N	HC	FHC	LC	FLC
3	8	8	32	32
4	16	16	80	80
5	32	32	192	192
6	64	64	448	448
7	128	128	1024	1024
8	256	256	2304	2304
9	512	512	5120	5120
10	1024	1024	11264	11264

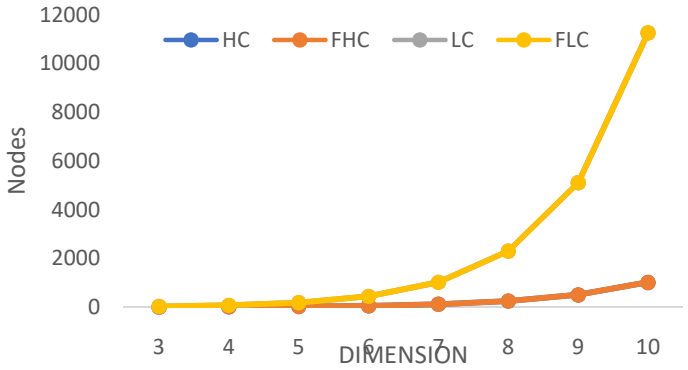

Figure 3: Comparison of Nodes

are listed in Table1 for all topologies. The comparison of nodes and edges are listed in Table 2 and 3 respectively.

The FLC has more number of nodes in comparison to HC and FHC and CQ but its growth is comparatively slow in comparison to star derivatives. Also it is same for comparison of total number of edges versus dimension as shown in Fig 4.The computed values for different networks are listed in Table 3. Due to the addition of complimentary edges, Link complexity is slightly higher.

Table 3: Comparison of Edges

N	HC	FHC	LC	FLC
3	12	16	36	48
4	32	40	96	128
5	80	96	240	320
6	192	224	576	768
7	448	512	1344	1792
8	1024	1152	3072	4096
9	2304	2560	6912	9216
10	5120	5632	15360	20480

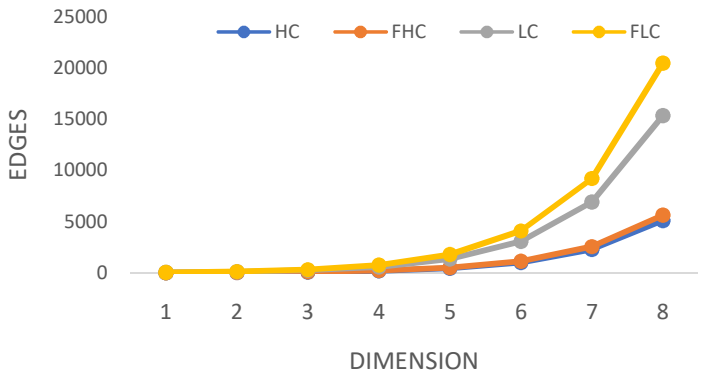

Figure 4: Comparison of Edges

While comparing the diameter, the FLC network is found to have better values in comparison to all other network topologies. The diameter of FLC is exactly same as that of Hypercube while covering more number of nodes at same dimension. It is low in comparison to LC as it is covering same number of nodes in less time due to folding technique. The diameter of FHC has lower value than FLC with low packing density. Hence for massive parallel computing system, FLC is a better candidate, as it covers a large number of nodes with lower value of diameter as shown in the Fig. 5.

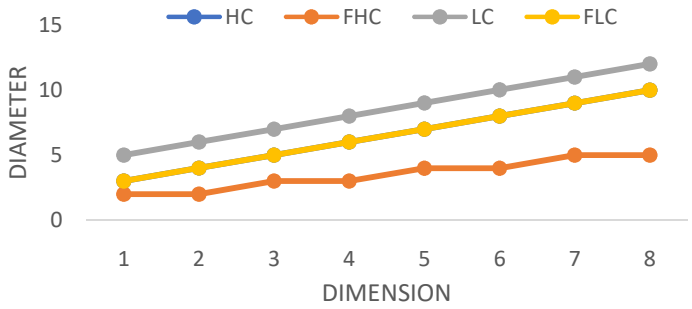

Figure 5: Comparison for Diameter

The Figure 6 depicts the comparison of cost of FLC with the parent network and others. The best thing here is the cost of FLC is not much high as compared to the parent networks, So it is very cost effective. FLC is a hybrid structure but the cost is very efficient as compared to LC. Though the structure used more complimentary edges the cost of travelling in FLC network is not increased in comparison to LC,the parent network.

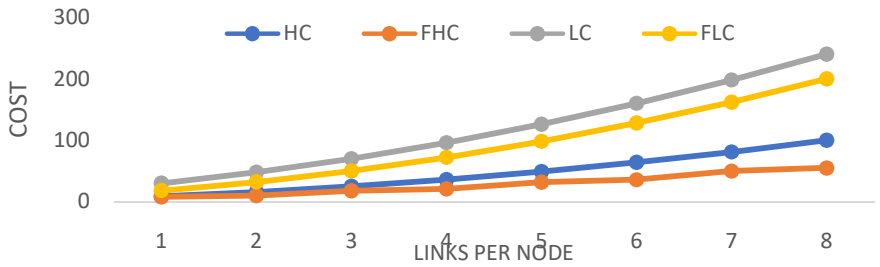

Figure 6: Comparison of Cost

Next the comparison of bisection width of FLC with other topologies is discussed. Due to folding extra edges are added and thus the bisection width value is increased as shown in Figure 7. The Figure 8 shows comparison of message traffic density of all the networks. It is 1 for the hypercube where as the message density of LC and FLC is more than 1due to increased node count. Thus the FLC is a better network as it is close to Hypercube for all values of dimension in comparison to LC.

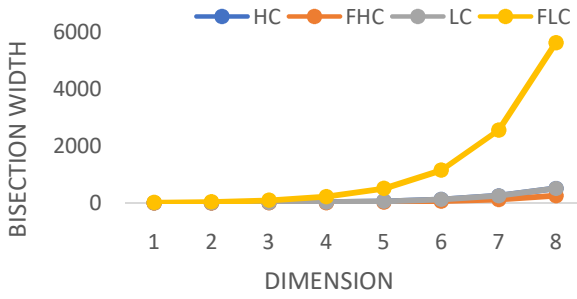

Figure 7: Comparison of Bisection Width

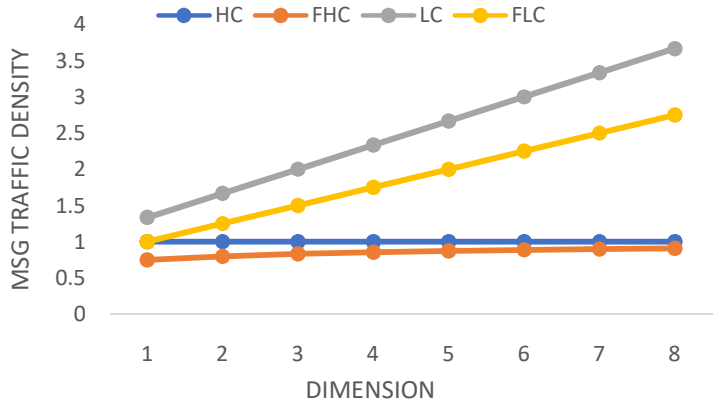

Figure 8: Comparison of Message Traffic Density

Now Table 4 depicts the link complexity growth rate comparison for FLC topology with the parent networks.

Table 4: Growth Rate of Link Complexity

n	HC	FHC	LC	FLC
3	0.666667	0.5	0.888889	0.666667
4	0.5	0.4	0.833333	0.625
5	0.4	0.333333	0.8	0.6
6	0.33333	0.285714	0.777778	0.583333
7	0.285714	0.25	0.761905	0.571429
8	0.25	0.222222	0.75	0.5625
9	0.22	0.2	0.740741	0.555556
10	0.2	0.181818	0.073177	0.054883

Figure 9 show the comparison of growth rate of the respected network with the base network, and show the desirable output that the growth rate of FLC less in comparison to Leafy cube.

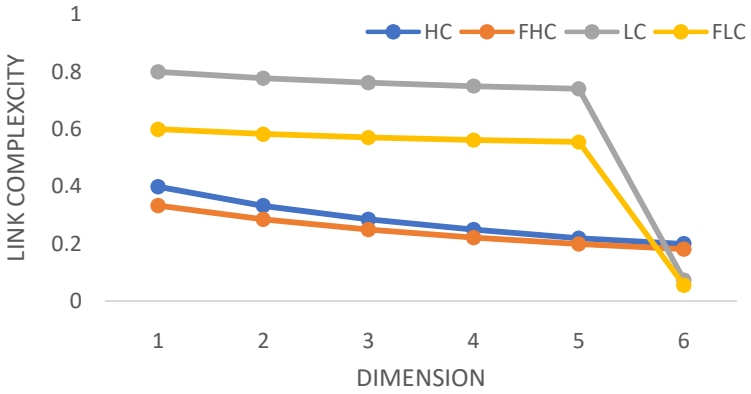

Figure 9: Comparison of Link Complexity

Table 5 and 6 presents the computed values of cost effectiveness and time cost effectiveness of FLC (n). In both the cases FLC exhibits better cost effectiveness and time cost effectiveness in comparison to hypercube and crossed cube as evaluated.

Table 5: Cost Effectiveness Factor of FLC(n)

Dimension	ρ=0.1	ρ=0.2	ρ=0.3	ρ=0.4
3	0.714286	0.555556	0.454545	0.384615
4	0.666667	0.5	0.4	0.333333
5	0.625	0.454545	0.357143	0.294118
6	0.588235	0.416667	0.322581	0.263158
7	0.555556	0.384615	0.294118	0.238095
8	0.526316	0.357143	0.27027	0.217391
9	0.5	0.333333	0.25	0.2
10	0.47619	0.3125	0.232558	0.185185
11	0.454545	0.294118	0.217391	0.172414
12	0.434783	0.277778	0.204082	0.16129

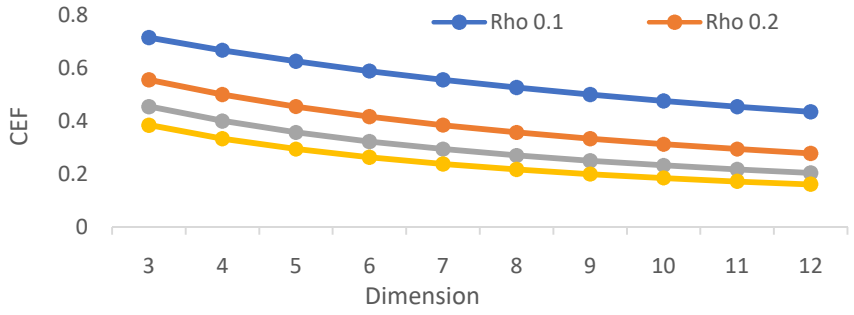

Figure 10: Cost effectiveness factor of FLC (n)

Table 6: Time Cost effectiveness factor of FLC (n) σ = 1, α = 1

Dimension	σ=1	σ =2	σ =3	σ =4
3	1.39738	1.09215	0.896359	0.760095
4	1.322314	0.993789	0.79602	0.6639
5	1.245944	0.906944	0.71296	0.587336
6	1.174928	0.832559	0.644697	0.526007
7	1.110509	0.768942	0.588066	0.47608
8	1.052391	0.714175	0.540477	0.434742
9	0.999902	0.666623	0.499976	0.399984
10	0.952341	0.624983	0.465107	0.370364
12	0.869558	0.555553	0.408162	0.32258

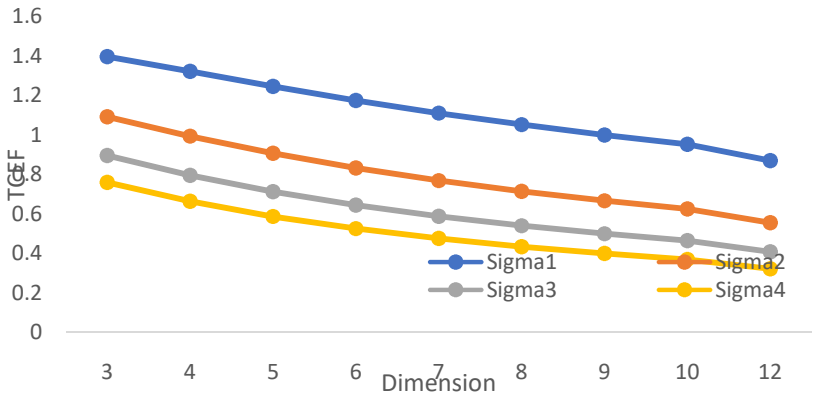

Figure 11: Time Cost Effectiveness Factor of FLC (n)

Figure 12 show computed values for two terminal reliability of the FLC network. The comparison is done with HC,CC, FCC and LC as they are direct derivative of parent structure.The proposed network is more reliable than the parent networks as it has more node disjoint paths with increasing values of n as shown in Figure 12. The Folded Leafy cube exhibits better reliablity and Routing then hypercube, Folded Cross Cube and Leafy cube. The LC contains more number of nodes in FCC is having higher values and they all have equal numbers of nodes and due to folding technique they have more no of edges. However, the FHC contains same no of nodes as FCC, but also they gave better throughput then other networks. Figure 13 depicts the superiority of the proposed network in terms of reliability while keeping n the node degree fixed at 10 but varying t from 1000 to 10,000. The FLC is more consistent.

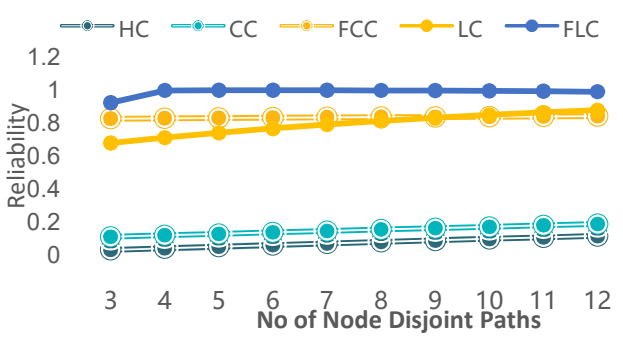

Figure 12: Comparison of Reliability for FLC with Time

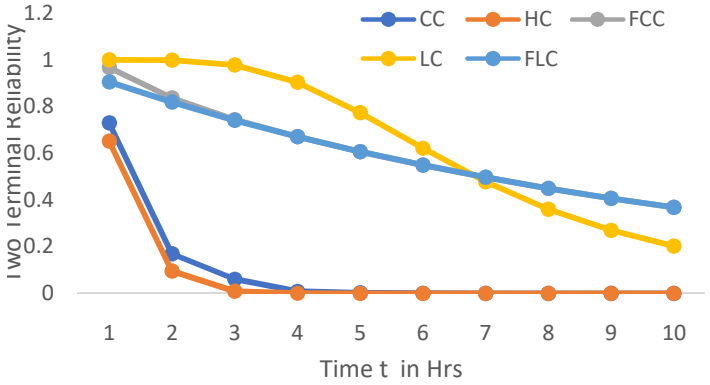

Figure 13: Variation of Reliability

5.2. Discussions on Scalability

Here the different desired problem sizes as discussed in Table 7. The comparison of *Isoeffiency* is described below. In theFig. 14,the horizontal axis represent the packing density and vertical axis represents the computed values for *Isoeffiency*. In earlier works, it shows that the graphs show that the *Isoeffiency* is to be within 0 and1. In the HC, the value is close to 0 for small scale system and for medium scale it goes up high and reaches to 0.9 but never crosses 1. In case of large scale system it falls down and becomes close to 0. In case of FCC the value remains close to 0.8 for small to big scale systems. Beyond that the efficiency falls down and becomes close to 0. In case of LC the values are same with FLC network, the value lies between 0.7 to 0.9. It also never crosses 1. The continuity of the curve established the superiority of the FLC network for small scale to massive large scale computing system.

Table 7: Problem Size Specification

Size(w)	Scale
50-100	Very Small Scale
100-500	Small Scale
500-1000	Medium Scale
1000-5000	Big Scale
5000-10,000	Large Scale
50,000 and more	Very Large Scale

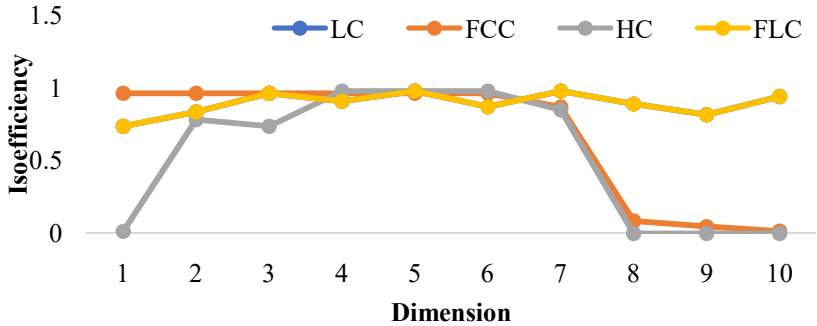

Figure 14: Comparison of Isoefficiency for FLC(n) with Parent Networks

5.3Discussions on Dynamic Load Balancing with Folding

The DLBF algorithm generates multiple types of random load and maps them on the processors and the leaf nodes in FLC networks. Here FLC_2 and FLC_3 are considered. Due to folding technique, the leaf nodes are also connected to other farthest leaf nodes in one hop. The tasks are generated haphazardly for simulation. The total number of tasks is supposed as 400for view. Table 8 represents the input tasks which are randomly generated at the initial stage of algorithm. Total load and ideal load calculated by the central scheduler. Central Scheduler also reports the list of under loaded and overloaded processors. In this case, processors 2 and 3 in case of $FLC_2$are under loaded and processors P0, P1 are overloaded by 11 and 12 respectively as shown in Table. Total load is 120 and there are 12 nodes. Hence the ideal load is 10. Then, it calculates maximum overloaded and maximum under loaded processors i.e. P0and P1 are overloaded by 2 and 4 respectively. Subsequently, it checks the connectivity between maximum overloaded and maximum under loaded processors. The same procedure is applied to leaf nodes also. If it finds the connection, then migration of loads is initiated. Otherwise, it seeks the second maximum under loaded processor for migration and so on. As a result, a set of load-balanced processors is shown in Table 4.9. After each iteration the leaf load (LL) and root load (LR) are calculated and compared. Then migration of load occurs till the ideal load state is achieved. Next Tables 9 (A, B and C) show load balancing in FLC_3 network. Here total load is 400 and total number of PEs are 32. Thus the ideal load is almost 13. the Table 9(A) shows the first and second simulation steps with load migration and intermediate loads at each PE after step 1 and 2. Similarly Tables 9(B) show the steps 3, 4 and 5 and 9(C)

Table 8: Load Balancing in FLC(2) Network

Root	Leaf	TL	LL	LR	Balance		Balance		Final Load
					LL	LR	LL	LR	
0	0	34	11	12	9	11	10	9+1	30
	1		11		9		10		
1	0	38	12	14	10	11	10	11-1	30
	1		12		10		10		
2	0	26	8	10	10	11	10	11-1	30
	1		8		10		10		
3	0	22	7	8	9	11	10	9+1	30
	1		7		9		10		
Total		120						120	120

shows the steps 6, 7 and 8. at each step intermediate loads are calculated to check if ideal load is reached otherwise load migration takes place from MOL to MUL if a connection exists. The matching coloured hilights in the table shows the PEs chosen for load migration for better understanding of the process.

In Dynamic Load Balancing algorithm attempts are made to establish the superiority through comparison with Hypercube, FHC and LC network. The Fig. 15 shows the

comparison of Load imbalance factor. Here the networks considered for comparison are HC, FHC, LC and FLC. But the diagram shows only two graphs as FHC and HC come under same line as they contain exactly same number of nodes. Similar is the case for LC and FLC. But load balancing is better in FLC because root nodes and child nodes get balanced simultaneously. Here the farthest leaf nodes are balanced with load migration due to complementary edge.

Table 9 (A): Load Balancing in FLC(3) (Step 1 &2)

ROOT	LEAF	TL	LL	LR	BAL-1		TL	LL	LR	BAL-2	TL
					LL	LR				LR	
0	0	60	15	15	16	13	61	15	16	15	60
	1		15		16			15			
	2		15		16			15			
1	0	42	10	12	10	13	43	10	13	13	43
	1		10		10			10			
	2		10		10			10			
2	0	54	13	15	13	13	52	13	13	14	53
	1		13		13			13			
	2		13		13			13			
3	0	50	12	14	10	13	43	10	13	13	43
	1		12		10			10			
	2		12		10			10			
4	0	35	8	11	10	13	43	10	13	13	43
	1		8		10			10			
	2		8		10			10			
5	0	49	12	13	12	16	52	13	13	13	52
	1		12		12			13			
	2		12		12			13			
6	0	40	10	10	10	12	42	10	12	14	44
	1		10		10			10			
	2		10		10			10			
7	0	70	17	19	16	16	64	16	16	14	62
	1		17		16			16			
	2		17		16			16			
TOTAL		**400**					**400**				**400**

Table 9 (B): Load Balancing in FLC(3) (Step3,4 and 5)

ROOT	LEAF	TL	LL	LR	BAL-3	TL	LL	LR	BAL-4	TL	LL	LR	BAL-5
					LR				LR				LR
0	0	60	15	15	14	59	14	17	15	57	14	15	13
	1		15				14				14		
	2		15				14				14		
1	0	43	10	13	14	44	10	14	14	44	11	11	13
	1		10				10				11		

	2		10				10				11		
2	0	53	13	14	14	53	13	14	14	53	13	14	14
	1		13				13				13		
	2		13				13				13		
3	0	43	10	13	13	43	10	13	15	45	11	12	13
	1		10				10				11		
	2		10				10				11		
4	0	43	10	13	13	43	10	13	15	45	11	12	13
	1		10				10				11		
	2		10				10				11		
5	0	52	13	13	13	52	13	13	13	52	13	13	13
	1		13				13				13		
	2		13				13				13		
6	0	44	11	11	14	47	11	14	14	47	11	14	13
	1		11				11				11		
	2		11				11				11		
7	0	62	15	17	14	59	14	17	15	57	14	15	14
	1		15				14				14		
	2		15				14				14		
TOTAL		**400**				**400**				**400**			

Table 9 (C): Load Balancing in FLC(3) (Step 6,7 and 8)

ROOT	LEAF	TL	LL	LR	BAL-6 LR	BAL-7 LR	TL	BAL-8 TL	FINAL BALALANCE TL
0	0	55	13	16	15	14	53	50	50
	1		13						
	2		13						
1	0	46	11	13	14	14	47	50	50
	1		11						
	2		11						
2	0	53	13	14	14	14	53	50	50
	1		13						
	2		13						
3	0	46	11	13	13	14	47	50	50
	1		11						
	2		11						
4	0	46	11	13	13	13	46	49	50
	1		11						
	2		11						
5	0	52	13	13	13	13	52	49	50
	1		13						
	2		13						
	0		11						

6	1	46	11	13	13	13	46	51	50
	2		11						
7	0	56	14	14	14	14	56	51	50
	1		14						
	2		14						
TOTAL		**400**					**400**	**400**	**400**

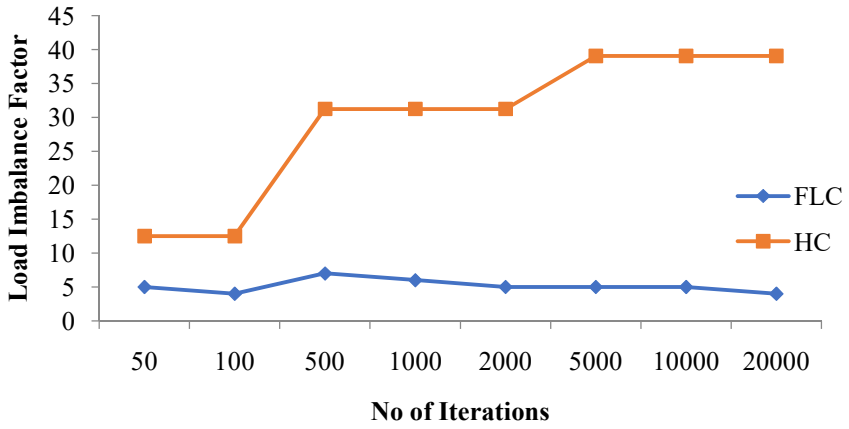

Figure 15: Comparison of Load Imbalance Factor for FLC and HC Network

The load per processing element is calculated and distributed the load according to the task. The load is distributed in an average number that's the throughput result is obtained in Fig.15. The comparison shows that the FLC network is quite evenly balanced for entire range. Due to more no of nodes in FLC network at same dimension as Hypercube, all the computing nodes are perfectly balanced. This algorithm is perfectly suitable for interconnection networks and may be useful for other related distributed computing or big data systems.

VI. CONCLUSION

In Very Large Scale Integration technology has advanced, parallel computers; computers consisting of a number of processing elements dedicated to solving a single problem at a time have become increasingly feasible, spurring a flurry of resource to find efficient parallel network.

The parallel computers based on the organization of the memory system, can have two broad categories: loosely coupled and tightly coupled. The loosely coupled parallel computers also called as multiprocessors. In recent days the use of multiprocessor in various scientific, engineering and general purpose applications has increased to a very great extent. The continuous demand for research in the area of VLSI has enhanced the importance of multiprocessors to a great size. To manage with the voluminous computation, large scale parallel systems needs to be designed to considerably reduce the communication overhead between the processors. So, the design of an efficient interconnection system becomes a vital issue. There are several Interconnection Network are present but it is very difficult to determine the best network.

The current work has proposed a new interconnection topology called Folded Leafy cube. The basic properties of Folded Leafy cube are derived and compared with its parent networks. The routing algorithms are proposed for this new topology with lesser time complexities. The cost of the proposed topology is found to be less. This proposed topology is found superior to the parent networks in terms of reliability, cost effectiveness and time cost effectiveness. The reduced diameter helps to speed up the overall operation of large scale parallel systems. The low link complexity it can pack a large number of links nearly increase but the size and cost is not increased. The bisection width and average node distance remain close to the base structure that is hypercube. Though the count for total nodes is increased, the same for edges is not increased similar to folded hypercube and crossed Cube. Folded Leafy cube improved broadcast time the proposed network is a better candidate for large scale parallel systems which provides better performance and efficient inter processor communication.

To prevent a database from becoming a single point of failure, and to improve scalability, the database is often replicated across multiple machines, and load balancing is used to spread the query load across those replicas. This is one of the main feature of big data system. The FLC being a scalable and load balanced network can be a suitable candidate for big data systems.

REFERENCES

1. L.N. Bhuyan and D. P. Agrawal, “Performance of Multiprocessor Interconnection Network. IEEE Computers, 1989.
2. Jose Duato, SudhakarYalamanchili and M. Ni Lionel,“Interconnection Networks: An Engineering Approach”, Elsevier science, USA; 2003.
3. W. J. Dally and T. Brain, “Principles and Practices of Interconnection Networks”, Elsevier, Morgan Kaufmann Press, San Francisco, 2004.
4. Y. Zhang,"Parallel Processing Systems for Big Data: A Survey," in *Proceedings of the IEEE*, vol. 104, no. 11, pp. 2114-2136, Nov. 2016,
5. C F Tsai, W C Lin and S. W. Ke, “Big Data Mining with Parallel Computing: A comparison of Distributed and Map Reduce Methodology”, Journal of Systems and Software, Volume 122, Pages 83-92, 2016.
6. Y. Saad and M.H. Schultz, “Topological Properties of Hypercubes”, IEEE Transactions on Computers,Vol.37(9), pp. 867-872, 1988.
7. Khaled Day and Anand Tripathy, “A Comparative Study of Topological Properties of Hypercube and Star Graphs”, IEEE Transactions on Parallel and Distributed Systems, Vol.5, No. 1; pp. 31-38Jan 1994.
8. Efe Kemal, “The Crossed Cube Architecture For Parallel Computation.” *IEEE Transaction on Parallel and Distributed Systems,* Sept 1992, 3(5): 513-524.

9. Ahmed EI-Amawy and Shahram Latifi, "Properties and Performance of Folded Hypercubes", *IEEE Transactions on Parallel and Distributed Systems*, Jan-1991, 2(1): 31-42.

10. N Adhikari and C. R. Tripathy "The Folded Crossed Cube: A New Interconnection Network for Parallel Systems"*International Journal ofComputer Applications (*0975 – 8887) Volume 4 – No.3, July 2010.

11. N Adhikari and A. Singh, "Leafy cube: A Novel Hypercube Derivative For parallel systems," *Springer Lecture Notes on Data Engineering and Communication Technology, Article in Press,* ISSN:2367-4512,2019.

12. N K Swain, A Singh, N Adhikari and S Pati, Leafycube: A Scalable and Load Balanced Interconnection Network Topology, International Journal of Research and Analytical Reviews (IJRAR), Vol.6, Issue:2, pp. 873-881, May 2019.

13. N Adhikari and A. Singh, "Reliable, Effective and Fault-Tolerant design of Leafy cube interconnection network topology," *International Journal of Innovative technology and exploring engineering,* Vol.X, Oct 2019.

14. Mohanty, Ankita Ananya and Adhikari, Nibedita, Folded Leafy Cube: A Derivative For High Performance Computing (January 10, 2020). 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

15. N Adhiakri, "On Scalability of Parallel Interconnection Network Topologies" *Proceedings of International Conference on Applications of Computing and Communication Technologies*, 10 March 2018, Book Chapter :Applications of Computing and Communication Technologies, pp.130-138.

16. N. Adhikari and C. R. Tripathy, "On Scalability of Interconnection Network Topologies", Applications of Computing and Communication Technologies, Vol.899, pp.130-138, 2018.

17. A. Gram, A. Gupta and V. Kumar, "Isoefficiency: Measuring the Scalability of Parallel Algorithms and Architectures", IEEE Parallel and Distributed Technology, Vol. 12, No.9, pp.12-21, 1993.

18. M.U.Bokhari and M.Alam, "Performance analysis of dynamic load balancing algorithm for multiprocessor interconnection network" *Elsevier perspectives in science*, 564-566,2016.