ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 7, Issue 7, July 2020

A High-Availability Algorithm for Hypercube Network Architectures Using Parallel Computing

[1]SARANYA M , [2]LATHA.A, [3]KOKILA.S, [4]INDHUMATHI.N
Assistant Professor /CSE, Tagore Institute of Engineering and Technology, Salem, India.[1]
PG Scholar /CSE, Tagore Institute of Engineering and Technology, Salem, India.[2]
Assistant Professor /CSE, Tagore Institute of Engineering and Technology, Salem, India.[3,4]

Abstract: The fast-growing network communication needs to keep the data with more simultaneously connected port switching committed be done the storage problem, which can be over rids the physical systems to avoid the storage allocation. As well here induce the concept of n-port switching. This system can automatically adapt the storage and also deduce the system's ideal time using the algorithm of high availability. Past, they can use the n-port switch to allocate the storage in one by one but the proposed concept might override and execute in parallel. As its predefined storage device with the help of intelligence. If, take one real-time data center which has produced the invoice for their customers and selling their products. In this case, management needs to store lots of information related to products, their price and customer details and also needs to manage the whole resource information. So, they need a lot of storage to store this kind of information as well as the on-premise server. Here, our algorithm shall reduce the overall infrastructure and use the concept of the on-premises server with decision making and zero down time. High Availability algorithm can play a major role in this concept to reduce downtime. Each of the data storage can be configured as the port manner that can be executed simultaneously to check the storage and it can allocate the possible storage to store the data. In most possible cases, we can avoid downtime as well. In existing system can use the algorithm named fault tolerance. It should be working on each element and reduce the faults and prevent the wrong communication and will demonstrate the high availability algorithm introduced.

Keywords: Data center, Interconnection network, Network topology, Hypercube, Incremental scalability, Routing algorithm.

I. INTRODUCTION

Owing to the volatile add to of data volume, data centers form the center of cloud computing more than the Internet. The data center network entail the plan of both the network structures and the linked protocols toward interconnect thousands of or even hundreds of thousands of servers, storage devices and network equipments within a single data center. Meanwhile, the data center can in attendance low equipment cost, high and balanced network capacity, easy expendability, scalable message presentation and robustness of fault tolerance. The data centers usually engage infrastructure services, such as GFS Map-reduce and Dryad while as long as many online applications, such as search, gaming, Web mail and etc. The data center network (DCN) architectures be able to mostly divided into two categories: server-centric and switch-centric. In the server-centric design, servers play the role of both a server and a network forwarder. These design make in the clouds of packet go by on the servers. DCell ,BCube and FiConn drop into this group. A switch-centric network typically consists of multilevel plants of switches to attach the servers. In the switch-centric designs, the interconnection aptitude depends on switches, while servers do not need to be modified for interconnection purpose. Fat-Tree and VL2 networks go to this category.

FiConn, DCell and BCube are all recursively defined network structures for facts centers, representative that the

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 7, Issue 7, July 2020

kth level arrangement is constructed by between several the level structures. In these structures, the number of servers is considerably greater than before with the increase of the levels. For example, if use 16- port switches to make the DCell structure, the number of servers is 272 when k = 1. However, when k is increased from 1to 2, the number of servers can get to 74256. This means that the recursively defined system structures cannot achieve a fine grained growth so that servers can be regularly added to the system in terms of the requirements of applications. HCube is another type of server-centric data center network architecture. The major impulse of HCube is improving the competence of pointed similar data items in Torus-like data centers. In HCube, two data objects are like when the Hamming distance of the identifiers of these two data items is small. The similar data items can be stored on physically nearby servers in HCube. By so doing, a single search request can obtain all the similar data stuff stored on these servers. Nowadays, a new category of data center network architectures, named dual-centric architectures, was proposed . In Dual-Centric data center network architecture, the routing aptitude can be placed on both switches and servers. The dual centric architectures own the advantages of both switch-centric and server-centric architectures, and have a variety of nice properties for practical data centers. In recent years, a group of data centers have been constructed around the world to hold the explosive growth of data. Many companies, such as Microsoft, Google, Face book, Yahoo and Amazon, have invested radically to establish data centers. Huge volumes of hardware, software and database resources in these large-scale data centers can be allocatedenergetically to millions of Internet users at the same time. As the amount of data grow need more storage capacity to keep the data. One typical way to increase storage is to insert more components instead of replace old ones. In practice, instead of adding a huge number of servers at a time, a small number of storage hosts are regularly added from time to time. Expect minimal impact during the add-on on both the system worker and the system itself. A data center architecture with good incremental scalability can be extensive by adding a small number of servers, and its topological property are preserved. It is well known that the interconnection topology significantly affect the presentation of data centers. Attractive properties of an interconnection network include low diameter, scalability, high bisection width and etc. Hypercube can professionally replicate any other network of a like size and is one of the most flexible and efficient networks for parallel computation. The hypercube and its related networks, such as Twisted cube Folded cube Generalized hypercube, Hierarchical hypercube, Exchanged Cube, Crossed Cube ,Exchanged Crossed Cube and ary cube are all good candidate for the architecture of parallel computer systems. HCube is another kind of server-centric data center network architecture. The major motivation of HCube is improving the efficiency of searching similar data items in Torus-like data centers. In HCube, two data items are similar when the Hamming distance of the identifiers of these two data items is small. The similar data items can be stored on physically nearby servers in HCube. By so doing, a single search request can obtain all the similar data items stored on these servers. Nowadays, a new category of data center network architectures, named dual-centric architectures, was proposed. In Dual-Centric data center network architecture, the routing intelligence can be placed on both switches and servers. The dual centric architectures possess the advantages of both switch-centric and server-centric architectures, and have various nice properties for practical data centers. In recent years, a lot of data centers have been constructed around the world to embrace the explosive growth of data. Many companies, such as Microsoft, Google, Face book, Yahoo and Amazon, have invested significantly to establish data centers. Huge volumes of hardware, software and database resources in these large-scale data centers can be allocated dynamically to millions of Internet users simultaneously.

II. PROBLEM DEFINITION

A Highly Scalable Data Center Network

The HSDC architectures are construct by m-port switches and 2-port servers, and can be separated into two categories: complete architecture and incomplete architecture. The incomplete HSDC architectures contain a certain number of idle server ports. A small number of servers can be added into the incomplete HSDC architecture by using these idle server ports. The expansion process can be continued until we get a complete HSDC architecture.

Reduce the downtime

Mainly focused on scalable of cloud data center using the fault tolerant algorithm. Instead of that use High availability algorithm to reduce the downtime for our HSDC Introduce four kinds of data center network Architectures named Fat-Tree, DCell, FiConn and Bcube.

III.RELATED WORK

In this section introduce four kinds of data center network architectures including Fat-Tree, DCell,c FiConn

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 7, Issue 7, July 2020

and BCube. All these architectures are construct by product switches, and have been broadly studied. Among all these four DCN architectures, the Fat-Tree is switch-centric, and the other three architectures go to server-centric.

Fat-Tree: Fat-Tree is a kind of level-based architecture. As shown in Figure 1, the switches in Fat-Tree are divided into three levels including edge level, aggregation level, and core level. A Fat-Tree architecture build with n-port switches has n pods. Each pod contain two levels of n=2 switches. In the edge level, each switch uses n=2 ports to connect n=2 servers, and uses the outstanding n=2 ports to connect the n=2 switches in the aggregation level. There are n2=4 switches in the core level, and each switch uses one port to connect one pod. Therefore, the Fat- Tree architecture consists of 5n2=4 switches and supports n3=4 servers in total. The scalability of Fat-Tree is limited by the ports of switches. As a result, if Fat- tree needs to be expanded and the existing switches are fully utilized, switches must be replaced with more-port ones.

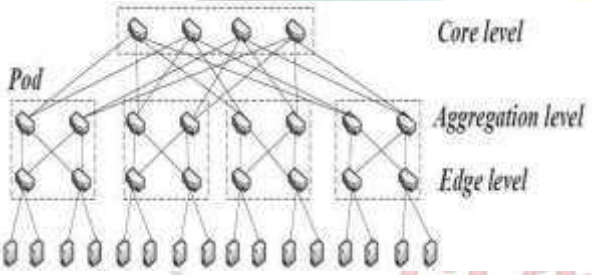

Fig1.1 Fat-Tree structure

DCell: DCell is recursively defined. A DCell0 contains n servers and one commodity n-port switch, while the switch is only used to connect the servers. The t servers in a DCellk connect to the other t DCellk, respectively. The DCell architectures can be expanded to a very big scale. However, the scalability of DCell is limited by the server ports. This means, if some new switches and servers are additional into the DCell, some additional ports have to be added into the servers to start the link connections. Furthermore, the incremental scalability of DCell is incredibly poor. As long as a DCell architecture is skilled, it is very hard to add new servers into the architecture without changing the unique architecture.

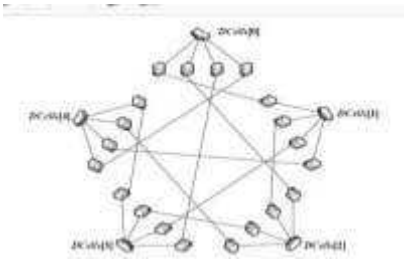

Fig1.2 DCell structure

FiConn: FiConn is constructed by using n-port switch and 2-port servers, where n is an even number. Similar to DCell, FiConn network can be recursively constructed. In FiConn0, every server uses one port to connect the switch. If a FiConn contains b support ports, a FiConnk architecture can be constructed by b=2 + 1 FiConn. In FiConn, each FiConnk uses b=2 servers with available backup ports to connect the other b=2 FiConn. Figure 3 shows FiConn2 with n = 4. FiConn can be expanded without adding extra server ports or switch ports. However, its incremental scalability is not good. In order to solve this problem, incomplete FiConn was proposed. An incomplete FiConnk is constructed by employing a small number of complete FiConn, and the FiConn are fully connected. Unfortunately, the bisection widths of some imperfect FiConns are extremely low.

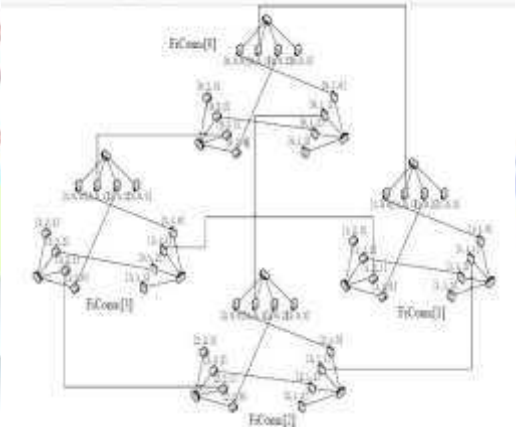

Fig1.3 FiConn structure

BCube: BCube architecture is also defined in a recursive way. BCube0 is construct by n servers and an n-port switch. In general, a BCubek is constructed from nk n-port switches and n BCube units. Every server in a BCubek has k + 1 ports.illustrates BCube1 with n = 4. The BCube architecture is suitable for building shipping-container based data centers which contain up to a few

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 7, Issue 7, July 2020

thousands of servers. However, building the data centers based on BCube incurs very high cost of switches and wirings.

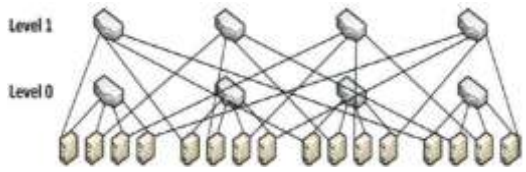

Fig1.4 BCube structure

IV. The HSDC Architectures

The HSDC architectures are construct by m-port switches and 2-port servers, and can be divided into two categories: complete architecture and incomplete architecture. The incomplete HSDC architectures contain a certain number of idle server ports. A small number of servers can be added into the incomplete HSDC architecture by using these inactive server ports. The expansion process can be continued until get a complete HSDC architecture. All the server ports in the complete HSDC architectures are fully busy, which means that the number of servers of the HSDC architecture constructed by m-port switches has reached the upper limit, and thus we cannot add any server into it. In this section, we first give the definition of complete HSDC architecture. After fully analyzing the topological properties of the complete HSDC architecture, propose three kinds of incomplete HSDC architectures. The topological properties of the incomplete architectures are deduced based on the topological properties of the complete architecture.

A. Complete HSDC Architectures

An interconnection network is typically represented by a graph where the vertices place for processors and the edges stand for links reside between processors. A graph Γ = (V; E) is defined by a set V of vertices and a set E of directed edges. The set E is a subset of elements (u; v) of V _ V. E is considered symmetric if (u; v) 2 E, (v; u) 2 E, in which case these two opposite arcs (u; v) and (v; u) are denote as an undirected edge (u; v). A graph is called undirected graph if all edges in it are undirected.

B. Shortest Internode Distance

The path lengths are calculated differently in switch-centric architectures and server-centric architectures. For switch-centric architectures, the length of a path is calculated as the number of links in the path for server-centric architectures, the length is calculated as the number of servers in the path (source and destination excluded) between the two servers. Since the HSDC is a kind of server-centric data center architecture, the shortest distance between the source and destination servers is the path with a minimum number of servers.

C. Diameter

The diameter of graph Γ, denoted by diam(Γ), is defined as the maximum distance for all pairs of distinct vertices in Γ. Diameter is an important topological property of an interconnection network, since it can be a measurement parameter of the communication latency. Low diameter is one of the pleasing properties of an interconnection network.

D. Bisection Width

Bisection width denotes the smallest number of edges to be removed to divider a network into two parts with equal size. A large bisection width imply a high network capacity and a more flexible structure against failures.

E. Routing Algorithm

For any interconnection network, the routing algorithm is a very important communication algorithm. In order to deal with the failure of switches or servers in a real-world simulation, propose a fault-tolerant routing algorithm for HSDCm(m) in this section, as shown in Algorithm 1. This algorithm is fully distributed, which means it can be executed on any vertex and it can detect its neighbor vertices and quickly determine the next unfaulty vertex on the path from the current vertex to the destination vertex.

F. Incomplete HSDC Structures

In practice, instead of constructing complete HSDCm(m) architecture which contains millions of servers directly, use m port switches to construct an incomplete HSDC structure, and the number of servers in it is less than m2m. When need to increase storage can add any number of servers into the incomplete HSDC structures, and their topological properties are preserved. To this end develop three kinds of different incomplete HSDC structures in this section.

G. The First Incomplete Structure HSDCm(n)

HSDCm(n) can be constructed through the method introduced in Theorem 3. According to the proof of propose a routing algorithm In Com Routing1 for HSDCm(n) (shown

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 7, Issue 7, July 2020

as Algorithm 2) which calls the routing algorithm Route.

H. The Second Incomplete Structure HSDCm(n, k)

The HSDCm(n; k) is a sub graph of HSDCm(n+1).When n k,can add another HSDCm(k) into a HSDCm(n; k) to construct a HSDCm(n; k + 1). Furthermore, if add a HSDCm(m=2) into a HSDCm(m=1;m=2) can get a HSDCm(m). As shown in Figure 8, the HSDC4 (3; 2) is constructed by using a HSDC and a HSDC.

I. The Third Incomplete Structure HSDCm(n, k, w)

Add a HSDCm(w) structure into a HSDCm(n; k) to construct another more complex incomplete structure of HSDCm(n; k;w). A HSDCm(n; k; w) contains one HSDCm(n), one HSDCm(k) and one HSDCm(w).the HSDC4(3; 2; 1) is constructed by HSDC4(3), HSDC4(2) and HSDC4(1). The path from (xn+1xn:::x1; y) to (un+1un:::u1;

z) can be constructed in HSDCm(n; k) through the algorithm InCom Routing2.

V. ANALYSIS

A. High availability algorithm

Running server operations using clusters of either physical or virtual computers is all about improving both reliability and performance over and above could expect from a single, high- powered server. Add reliability by avoiding hanging your entire infrastructure on a single point of failure (i.e., a single server). And increase performance through the ability to very quickly add computing power and capacity by scaling up and out.

B. Diameter

The diameter of Fat-Tree is 2log2N which social group to the 2 multiples of the height of the Fat-Tree. The correct diameters of DCell and FiConn are still unknown. The upper limits of these two structures are both 2logmN =1. The diameter of BCube is the smallest among these server-centric architectures, which equals to logmN. As analyzed in the diameter of HSDC is 2m, which is comparatively small but slightly bigger than that of DCell and FiConn. The HSDC can hold applications with real-time necessities.

C. Bisection width

It is apparent that a high bisection width imply a high network aptitude and a more flexible structure. As shown in Table 1, the bisection widths of Fat-Tree and BCube are both equal to N=2. The exact bisection widths of DCell and FiConn are still unknown, and the lower bounds of DCell and FiConn are both N=(4lognN). The bisection width of HSDC equals to N=2m. This implies that there are many potential paths between a pair of servers in the HSDC structure. HSDC is therefore essentially fault-tolerant, and it provides the possibility to plan multi-path routing on top of it.

D. Scalability

Not all the architectures shown in Table 1 fit to construct large scale data center networks. Only FiConn can be expanded without adding server ports or switch ports. The scalabilities of DCell and BCube are limited by server ports. When more servers are extra into these data center networks, the servers need to add more NIC ports. The salabilities of HSDC and Fat-Tree are all incomplete by the number of switch ports. This indicates that cannot add servers into the two architectures when the switch ports are completely occupied. The only way to resolve this problem is to use the switches, which have a large number of ports, to build the data center networks. However, this method is not suitable for Fat- Tree. This is mainly because that the Fat-Tree can only accommodate a small number of servers. As shown in 2, the number of servers in Fat-Tree is much smaller than that of HSDC. Even by using 196-port switches, the Fat-Tree can only contain up to 1882384 servers. In addition, the price of the large-port switch is very high. This will lead to high cost of structure the data center networks. For the HSDC, we can employ switches that have relatively larger number of ports to construct an unfinished architecture. The servers can be added into the incomplete HSDC structure. As more and more servers are added into the architecture, the incomplete HSDC structure will finally become a complete architecture. Table 2 shows the number of servers that can be deployed in HSDCm (m), where m represents the number of ports of a switch. As shown in Table 2, the number of servers that can be deployed in HSDCm(m) significantly increases with the growth of m. This indicates that the size of HSDCm(m) is far more enough to satisfy the requirement of current data centers when using switches which have more than 24 ports.

E. Incremental Scalability

The incremental scalability of data center network architectures indicate that the slow expansion of data centers

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 7, Issue 7, July 2020

should have a smallest collision on the system behavior. For example, adding a few servers or a few thousands servers into the system should not influence the network topologies, such as diameter, bisection width and etc. As shown in Table 1, the incremental scalability of Fat-Tree is good. By continually adding pod into the Fat-Tree architecture, make data center network with random number of servers. Since the number of servers in Fat-Tree is small, the Fat- Tree is not suitable for building large-scale data center networks. Though DCell is a good applicant for scaling out, its incremental scalability is much worse than that of Fat-Tree. Additionally, once the deployment of DCell architecture is talented, it is very difficult to add new servers even a small number of new servers into the architecture without changing the architecture topology. Furthermore, the imbalanced traffic load in DCell will incur significant performance degradation. Like DCell, FiConn is also a recursive network structure. An unfinished FiConn model can be constructed by using a bottomup way. Since the number of high-level links in FiConn is smaller than that of low-level links, shortcut links must be added into the incomplete FiConn structure to increase the bisection width. For example, if there are only two FiConn in an incomplete FiConn the two FiConn will be connected through a single link. The bisection width of this structure is 1. In order to solve this problem, some additional links have to be added into the incomplete FiConn to obtain a higher bisection width. This implies that the connection model of the incomplete FiConn is different from that of the complete one. Then, the routing algorithm for the incomplete FiConn must be changed, leading to a weak incremental scalability. The incremental scalability of HSDC is better than that of DCell and FiConn. Initially, we can use 32-port or 48-port switches to construct a HSDCm(n) structure with m2n servers. When the structure is required to be expanded, 2k severs can be added to the HSDCm(n) structure to construct a HSDCm(n; k). Furthermore, we can add a HSDCm(w) into HSDCm(n; k) to construct a more com plex HSDC structure HSDCm(n; k; w). According to the analysis above, the topological properties of HSDCm(m) and the other three incomplete HSDC structures are similar. This implies that can add arbitrary number of servers (e.g., tens, hundreds, thousands or even tens of thousands) into the incomplete HSDC structures, while all the topological properties are preserved. Compared to Fat-Tree, the HSDC architectures can contain more servers and are more suitable for building large-scale data center network.

F. Throughput

In order to evaluate the throughputs of the unlike network structures, design a flow-level simulator, called mtCloudSi, to estimate the data flows behavior in the real world based on the come near proposed .The mtCloudSim is a kind of flow level network simulator and provides the multi-tenant environment and supports large-scale experiments. The mtCloudSim models the data center network as a network graph. Edges, corresponding to links, are directed and associated according to the given topology. The capacity of each edge can be customized. The simulator determines the delay caused by forwarding, queuing, and transmission and processing by assigning a fixed Round Trip Time (RTT) to each flow. The RTT is decided as 100s in terms of the examination of typical data center networks reported . The simulator can accept a formalized script to generate flows with the given four tuples (source host, destination host, start time, and flow size). When the flow starts, it will be added to the set of active flows. After the flow ends, the simulator will report such event and remove it from the network. Benson et al.conducted an empirical study on the network traffic pattern across ten different data centers belong to three different types of organizations, including universities, enterprises, and cloud data centers. Their definition of cloud data centers includes not only data centers employed by large online service providers offering Internet-facing applications, but also data centers used to host data-intensive (MapReduce style) applications. Some of the data centers have been in operation for over 10 years. They collected and analyzed the datacenter network topologies, flow-level and packet-level statistics.generate a artificial flow workload according to the characteristics summarized. The workload contains 80000 flows with a total size of 4TB. The maximum flow size is 1GB while the minimum size is 1KB. Furthermore, the source and destination hosts for each flow are chosen randomly. Therefore, the workload used in the evaluation can well symbolize the typical datacenter traffic. Five different structures, including Fat-Tree, DCell, BCube, FiConn, and HSDC, are constructed in the simulator to evaluate the throughput1. In the simulation experiments, we use 16-port switches to construct DCell, BCube, FiConn, and HSDC, while each switch used in Fat-Tree has 48 ports. All the structures contain 4096 servers. The data rates of the associations used in the experiments are 1Gbps.

G. Cost and energy consumption

In this section, compare the cost and energy consumption of HSDC with the other data center network

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 7, Issue 7, July 2020

architectures when use them to construct a data center network containing the same number of servers. Since the total cost and energy consumption of the servers are the same only evaluate the cost and energy consumption of the switches and NICs. Table 1 shows that the number of switches in FiConn, DCell and HSDC is the same, which is only one-fifth of that in Fat-Tree. The number of switches in BCube depends on the number of levels, but it is typically bigger than that of FiConn, DCell and HSDC, and is smaller than that of Fat-Tree. In order to analyze the cost and energy consumption intuitively construct data center networks containing 1024, 2048, 4096, 8192 and 16384 servers, respectively. Table 3 shows the price and power consumption of switches and NICs used in the data centers summarize the cost and power consumption incurred by the five different network architectures. As shown in Table 4, the cost and energy consumption of HSDC and FiConn are identical, which are lower than that of Fat-Tree and BCube. This conclusion is also confirmed in Fig. 11. The prices of switch and NIC may vary in the market. However, the different prices do not affect the comparison results in Table 4 and Fig. 11, because the total cost of the data center networks change along with the changes of the prices of the switches and NICs. Since the data center networks must be laid out with wires and cables, we also list the number of links used in these network architectures. As shown in Table 1, the number of links in HSDC and FiConn is the same, which is smaller than that of Fat-Tree, DCell and BCube. Table 4 shows that the number of links in HSDC is less than that of FiConn. This is because that no regular FiConn architecture can exactly contain the specific number of servers should construct an incomplete FiConn architecture. This causes a reduction in the number of links.

According to the above analysis, the cost, energy consumption and number of links of HSDC and FiConn are approximately equal, which are much smaller than that of Fat- Tree, DCell and BCube. As the cost, energy consumption and number of links of HSDC and FiConn maintain a low growth speed with the continuous expansion of the data center network. Since HSDC has better properties of throughput and incremental scalability than FiConn, the HSDC architecture is more suitable for building large-scale data center networks with low cost and energy consumption.

VI. CONCLUSION

In this paper propose a new type of data center network architecture named HSDC based on the hypercube network. The HSDC consists of 2-port servers and low-cost commodity m-port switches. After calculating the shortest internodes distance,develop fault-tolerant routing algorithm for the HSDC. The routing algorithm can be executed on any vertex in the network and constructs a path between any pair of vertices. We also analyze the topological properties of HSDC. The diameter and the bisection width of HSDC are O(log(N=m)) and N=2m, respectively, where N represents the number of servers. In order to achieve incremental scalability, we also propose three kinds of incomplete HSDC structures. When arbitrary number of servers are added into the incomplete HSDC structures, all the topological properties can be preserved. The analysis results indicate that HSDC strikes a good balance among diameter, bisection width, incremental scalability, and other important characteristics in contrast to the state-of-the-art data center network architectures. The cost, energy consumption, and number of links are also evaluated. The experimental results demonstrate that HSDC is the best candidate for building arbitrary-scale data centers. A simulator is also designed to evaluate the throughput of the five network structures use servers which have two NIC ports to construct HSDC in this paper. Due to the fast evolution of computer hardware, servers with more than two embedded NIC ports are emerging. Therefore, in our future work, we will investigate how to leverage those servers to further improve the HSDC architecture.

REFERENCES

[1]. A. El-Amawy and S. Latifi, "Properties and performance of folded hypercube," IEEE Transactions on Parallel and Distributed systems, vol. 2, no. 1, pp. 31–42, 1991.

[2]. M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in ACM SIGCOMM Computer Communication Review, vol. 38, no. 4. ACM, 2008, pp. 63–74.

[3]. K.Bilal, S. U. R. Malik, O. Khalid, A. Hameed, E. Alvarez,V. Wijaysekara, R. Irfan, S. Shrestha, D. Dwivedy, M. Ali et al., "A taxonomy and survey on green data center networks," Future Generation Computer Systems, vol. 36, pp. 189–208, 1984.

[4]. L. N. Bhuyan and D. P. Agrawal, "Generalized hypercube and hyperbus structures for a computer network," IEEE Transactions on computers, vol. 100,

no. 4, pp. 323–333, 2004.

[5]. T. Benson, A. Akella, A. Maltz, and David, "Network traffic characteristics of data centers in the wild," in the 10th ACM SIGCOMM conference on Internet measurement, 2010, pp. 267–280.

[6]. C.-P. Chang, J.-N. Wang, and L.-H. Hsu, "Topological properties of twisted cube," Information Sciences, vol. 113, no. 1, pp. 147–167, 1999.

[7]. W. J. Dally, "Performance analysis of k-ary n-cube interconnection networks," IEEE Transactions on Computers, vol. 39, no. 6, pp. 775 785, 1990.

[8]. C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu, "Bcube: a high performance, server-centric network architecture for modular data centers," ACM SIGCOMM Computer Communication Review, vol. 39, no. 4, pp. 63–74, 2009.

[9]. A.Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "Vl2: a scalable and flexible data center network," in ACM SIGCOMM computer communication review, vol. 39, no. 4. ACM, 2009, pp. 51–62.

[10]. P. K. Loh, W. J. Hsu, and Y. Pan, "The exchanged hypercube," IEEE Transactions on Parallel and Distributed Systems, vol. 16, no. 9, pp. 866–874, 2005

[11]. D. Li, C. Guo, H. Wu, K. Tan, Y. Zhang, S. Lu, and J. Wu, "Scalable and cost-effective interconnection of data-center servers using dual server ports," IEEE/ACM Transactions on Networking, vol. 19, no.1, pp. 102–114, 2011.

[12]. D. Li and J. Wu, "On data center network architectures for interconnecting dual-port servers," IEEE Transactions on Computers, vol. 64, no. 11,pp. 3210–3222, 2015.

[13]. Z. Li and Y. Yang, "Gbc3: A versatile cube-based server- centric network for data centers," IEEE Transactions on Parallel and Distributed Systems, vol. 27, no. 10, pp. 2895– 2910, 2016.

[14]. Y. Liu, J. K. Muppala, M. Veeraraghavan, D. Lin, and M. Hamdi, Data center networks: Topologies, architectures and fault-tolerance characteristics. Springer Science & Business Media, 2013.

[15]. Xie and Y. Deng, "mtcloudsim: A flow-level network simulator for multi-tenant cloud," in Proceedings of the 22th IEEE International Conference on Parallel and Distributed Systems, 2016.

[16]. Dhanapal, R., & Visalakshi, P. (2015). Efficient Clustering protocol based on Ant-Bee agent for Large Scale MANET. International Journal of Applied Engineering Research, 10(52), 2015.

[17]. Mathew, O. Cyril, R. Dhanapal, P. Visalakshi, K. G. Parthiban, and S. Karthik. "Distributed Security Model for Remote Healthcare (DSM-RH) Services in Internet of Things Environment." Journal of Medical Imaging and Health Informatics 10, no. 1 (2020): 185-193.

[18]. Dhanapal, R., & Visalakshi, P. (2016). Real time health care monitoring system for driver community using ADHOC sensor network. Journal of Medical Imaging and Health Informatics, 6(3), 811-815.

[19]. Yuvaraj, N., G. Saravanan, R. Dhanapal, and M. Premkumar. "Towards Efficient Data Transmission using Energy-based Clustering Model (ECM-EDT) in Heterogeneous VANET." (2020).

[20]. Dhanapal, R., T. Akila, Syed Shuja Hussain, and Dinesh Mavaluru. "A Cost-Aware Method for Tasks Allocation on the Internet of Things by Grouping the Submitted Tasks." Journal of Internet Technology 20, no. 7 (2019): 2055-2062.

[21]. Parthiban, K. G., S. Vijayachitra, and R. Dhanapal. "Hybrid Dragonfly Optimization-Based Artificial Neural Network for the Recognition of Epilepsy." International Journal of Computational Intelligence Systems 12, no. 2 (2019): 1261-1269.

[22]. Dhanapal, R., and P. Visalakshi. "A Sector Based Energy Efficient Adaptive Routing Protocol for Large Scale MANET." Research Journal of Applied Sciences, Engineering and Technology 9, no. 7 (2015): 478-484.

[23]. Data, Energy Efficient Privacy-Preserving Big. "MCEEP-BDA: Multilevel Clustering Based-Energy Efficient Privacy-Preserving Big Data Aggregation in Large-Scale WSN."