GIDC: A Cost-Effective and Incremental Scalable Data Center Network Structure

Peng Zhou

Department of Computer Science, Jinan University, Guangzhou 510632, China

Abstract

As the volume of data grows rapidly, servers and network devices are continually added to data centers to store and analyze data. Industry experience shows that instead of adding a large number of servers at once, data center networks are gradually expanded by adding a small number of servers from time to time according to actual needs, i.e., Incremental scalability. To address the many shortcomings of the current dual-centric data center network in terms of incremental scalability, and cost and energy consumption, this paper proposes a new Dual-Centric data center network architecture GIDC based on Hypercube. In order to achieve incremental scalability, this paper further proposes two incomplete GIDC structures, where a small number of servers can be added to the incomplete GIDC structure according to the scaling requirements, while its topological characteristics remain unchanged. The analysis and experimental results show that the throughput of GIDC is comparable to FSquare, 17.45% and 25.5% higher than FCell and FRectangle, respectively. Compared with FCell, FRectangle and FSquare, the cost and energy consumption of GIDC are 10.84%, 22.85% and 29.55% lower, respectively.

Keywords

Data Center Network; Hypercube; Dual-centric; Incremental Scalability.

1. Introduction

The continuous development of the Internet industry has promoted the rapid development of a series of emerging Internet services such as cloud computing, big data, artificial intelligence. To support such services, data centers are essential. As the core infrastructure of big data and cloud computing, data centers have become one of the infrastructures supporting the modern Internet industry.

Currently, data center network structures are mainly divided into three categories, including switch-centric, server-centric and dual-centric. The switch-centric structures consist of multi-layer of switches to connect the servers, and the network connection and routing functions are primarily completed by the switch. Fat-Tree[1], VL2[2], and Jupiter[3] belong to this category. The server-centric structures are constructed recursively, and servers play the role of both a server and a network forwarder. BCube[4], DCell[5], and FiConn[6] belong to this category. In order to avoid the problems of high construction cost of switch-centric structures and low efficiency of data forwarding by the server of server-centric structures. Compared with the switch-centric and server-centric structures, the network connection and routing functions are mainly completed by the switch and server. FCell, FRectangle, and FSquare[7].

As the network scale continues to expand, data centers not only host traditional client/server applications, but also new applications such as GFS and MapReduce [8]. To meet the demands of new computing models and applications, today's data center networks need to meet the following requirements:

(1) Cost- Efficiency: The cost of a data center comes from four main sources: 45% from hardware (e.g., servers, main memory, switches, and storage systems), 25% from infrastructure (e.g., cooling systems and power distribution), 15% from power consumption; and 15% from network resources (e.g., links, devices, and transit) [9]. Therefore, the design of a data center network architecture must strike a good balance between performance and cost (e.g, energy-efficiency).

(2) Incremental scalability: The data center network can add a small number of servers according to the real requirements while all the topological properties are maintained[10]. Faced with the increasing amount of data on the Internet, data centers need to continuously expand to increase their processing capacity. Compared with replacing old servers, adding new servers can be a better way to obtain cost advantages. Currently, the majority of data center network structures only consider the scalability issue, i.e., the data center network can be expanded to add more servers, while the study of incremental scalability is weak. A data center structure with good incremental scalability can be extended by adding a small number of servers while its topological properties are maintained.

This paper proposes a new type of dual-centric data center network structure called GIDC (Greater Incremental Scalability Data Center Network structure) based on Hypercube, which is constructed by using multi-port commercial switches and two-port servers. To achieve incremental scalability, three incomplete GIDC structures are proposed. A small number of servers can be added into the incomplete GIDC structures while their topological properties are maintained. The analysis and experimental show that GIDC sig-nificantly outperform FCell, FRectangle and FSquare in terms of the incremental scalability and robustness. The average throughput of GIDC is approximately equal to that of FSquare and highter than that of FCell and FRectangle by about 17.45% and 25.5%. Compared with the FCell, FRectangle and FSquare, GIDC reduces the cost and energy consumption by about 10.84%, 22.85% and 29.55%, respectively. The GIDC strikes a good balance among incremental scalability, cost and energy consumption in contrast to the state-of-the-art data center network architectures.

The rest of the paper is organized as follows. Section 2 introduces the related work of data center network structures. Section 3 presents the definition of complete GIDC architecture. We compare the GIDC against other data center network structures in Section 4. Finally, Section 5 concludes the paper.

2. Related Work

In this section, we introduce three kinds of representative dual-centric data center network structures including FCell, FRectangle and Fsquare.

2.1. FCell Structure

FCell [7] is a Dual-Centric data center network structure based on a folded Clos topology using two-port servers and n-port switches. FCell(n) is constructed from blocks, each of which has two levels of switches in it. Each block is interconnected to the 2-level $n/2$ switches using the $n/2$ ports of the 1-level n switches, and the remaining ports of the 1-level switches are connected to the $n/2$ servers. The FCell(n) structure consists of $n^2/ 2+1$ blocks, where each block contains $n^2/2$ servers and $3n/2$ switches.The FCell structure is interconnected between the blocks in a similar way to DCell [5]. FCell(4) is shown in Fig. 1.

ISSN: 1813-4890

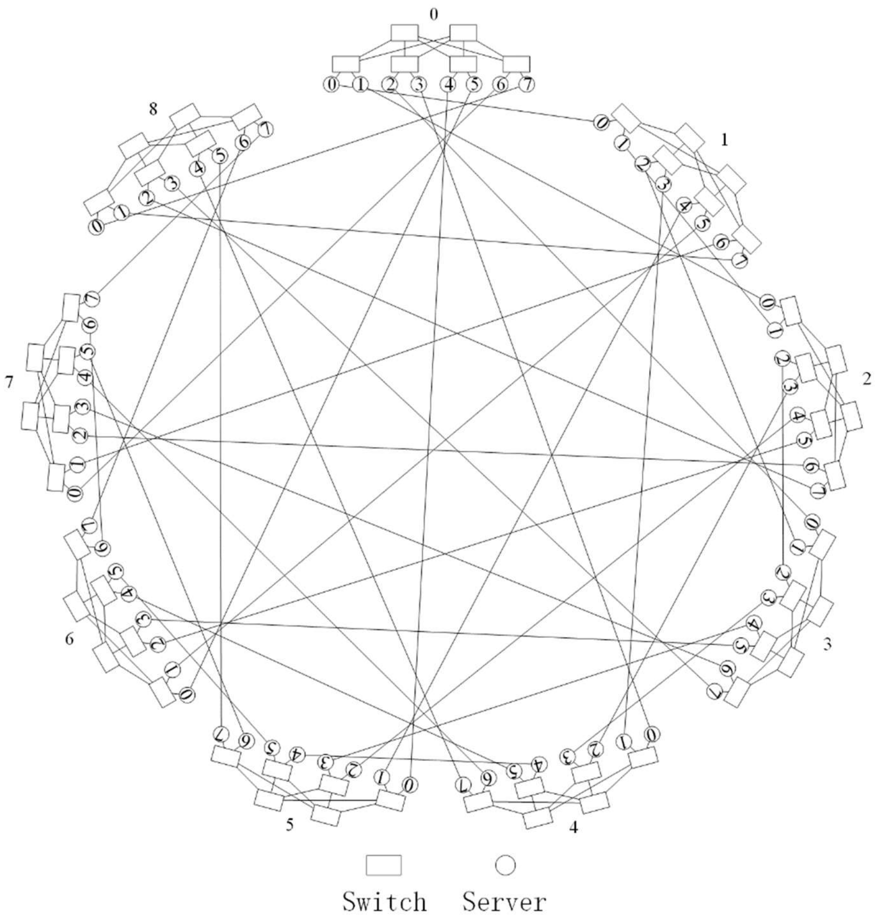

Fig. 1 FCell(4)

2.2. FRectangle Structure

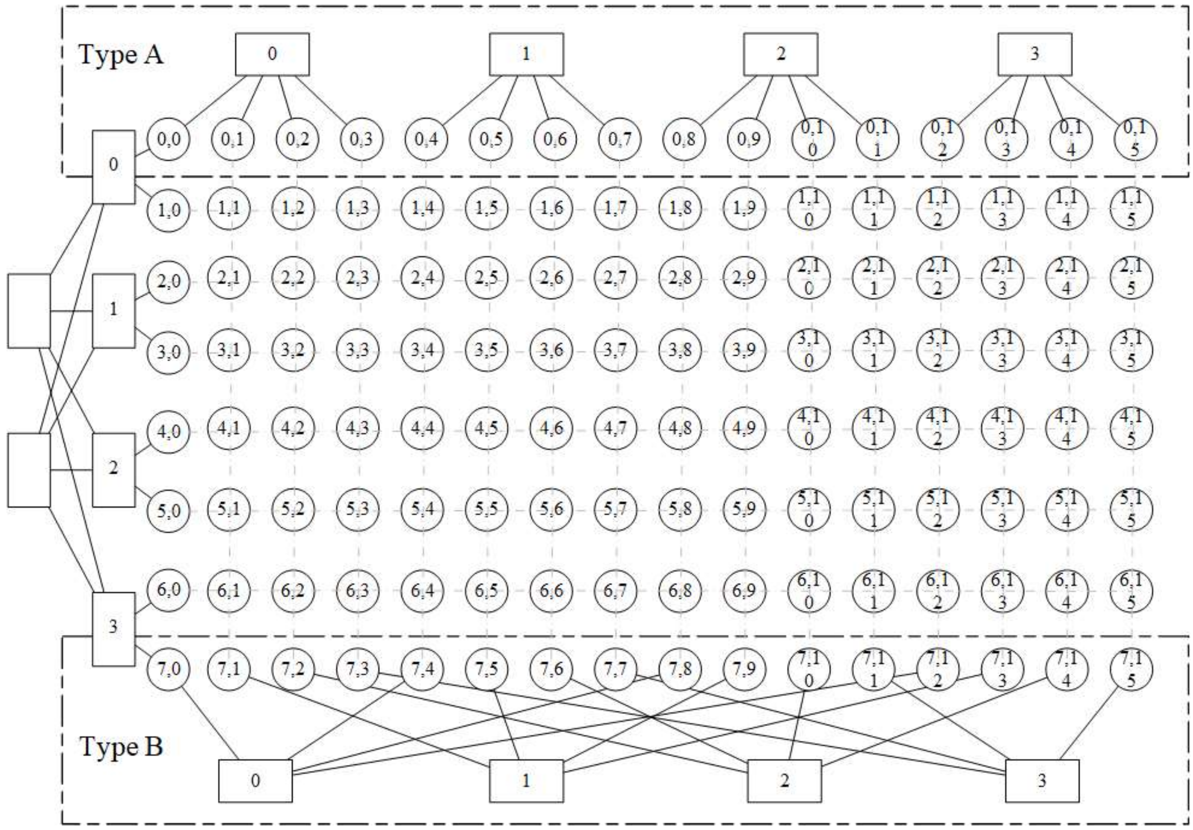

Fig. 2 FRectangle(4)

FRectangle [7] is a 2-dimensional structure consisting of rows and columns, which is constructed using 2-port servers and n-port switches. Each column has the same structure as the block of FCell structure. Each row is structured with n switches connected to n^2 servers. Each row of FRectangle structure has two interconnection types, A and B. Type A interconnection: for server $a(i,j)$ in ith row, $0 \le j \le n^2 - 1$, and if $kn \le j \le kn + n - 1 (0 \le k \le n-1)$, then $a(i,j)$ connects to the kth switch in this row. Type B interconnection: for server $a(i,j)$ in ith row, $0 \le j \le n^2 - 1$, and if $j\%n = k (0 \le k \le n-1)$, then $a(i,j)$ connects to the kth switch in this row. When $i\%2 = 0$, ith row selects type A interconnection, otherwise selects type B. FRectangle(4) is shown in Fig. 2.

2.3. FSquare Structure

Similar to the FRectangle structure, FSquare [7] is also a 2-dimensional structure constructed using 2-port servers and n-port switches. The FSquare structure of each row and column are the same as the block of FCell structure. Fsquare(4) is shown in Fig. 3.

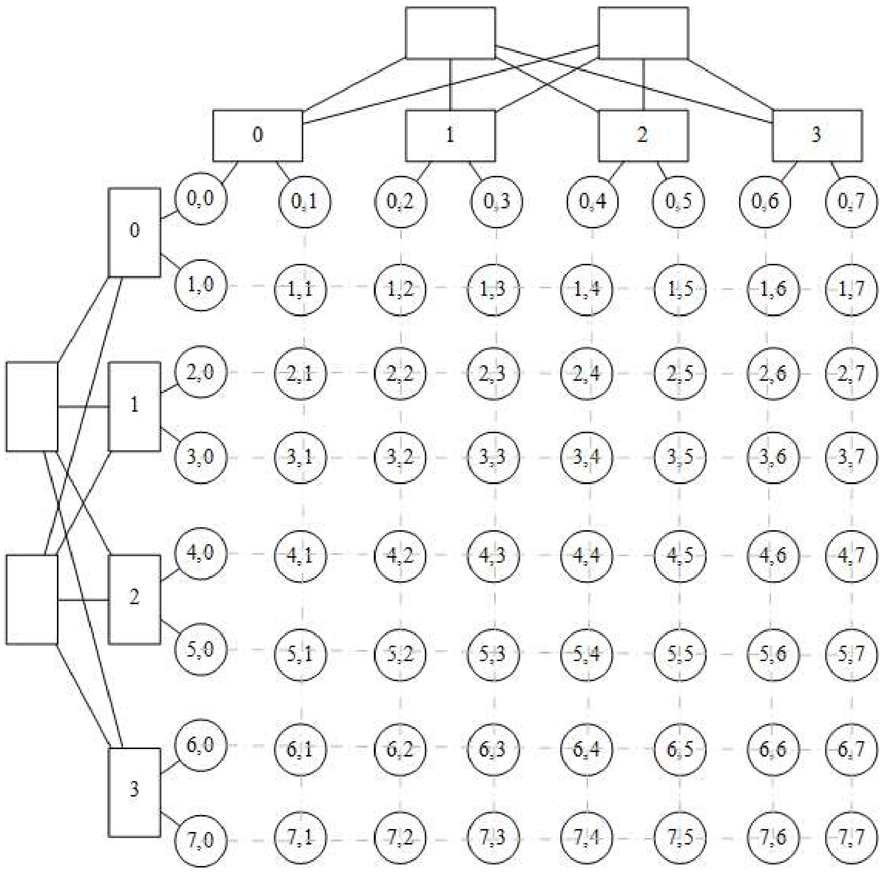

Fig. 3 FSquare(4)

3. GIDC Structure

Since the GIDC structure is constructed based on the Hypercube, we first give the definition of m-dimension Hypercube as follows.

Definition 1. In m-dimension hypercube H_m, the vertices and edges are defined as follows:

The vertices are identified as $(x_{m-1} \cdots x_0)$;

The edges are identified as $\left((x_{m-1} \cdots x_0), (x_{m-1} \cdots \overline{x_y} \cdots x_0)\right)$;

where $x_i \in \{0,1\}$, $0 \le y < m$, and $\overline{x_y}$ is the complement of x_y.

Every vertex in Hypercube network can be viewed as a binary number and vertices that differ in only one digit are connected together. The Hypercube $H_4$is shown in Fig. 4.

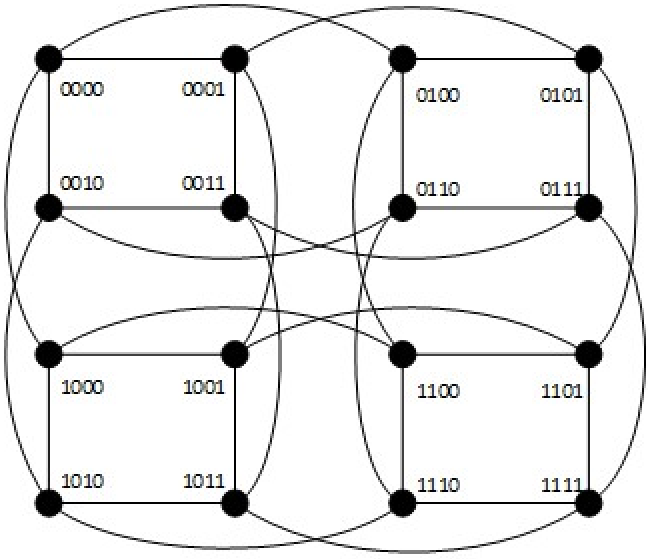

Fig. 4 Hypercube H_4

The GIDC(m) is constructed by using m-port switches and 2-port servers based on m-dimensional Hypercube H_m. The definition of GIDC(m) is as follows.

Definition 2. In GIDC(m), the vertices and edges are defined as follows:

The switches and servers are identified as $(x_{m-1} \cdots x_0; y_1 0)$, $(x_{m-1} \cdots x_0; y_1 y_0)$, respectively;

The edges between switches and servers are identified as:

$$\left((x_{m-1} \cdots x_0; y_1 0), (x_{m-1} \cdots x_0; y_1 y_0)\right)$$

The edges between switches and switches are identified as:

$$\left((x_{m-1} \cdots x_0; y_1 0), \left(x_{m-1} \cdots \overline{x_{y_1}} \cdots x_0; y_1 0\right)\right)$$

The edges between servers and servers are identified as when $y_1 < y_0$,

$$\left((x_{m-1} \cdots x_0; y_1 y_0), (x_{m-1} \cdots x_0; y_0 y_1 + 1)\right)$$

When:

$$y_1 \geq y_0,\ \left((x_{m-1} \cdots x_0; y_1 y_0), (x_{m-1} \cdots x_0; y_0 - 1 y_1)\right)$$

where $x_i \in \{0,1\}$, $m \in [2, +\infty]$, $y_1 \in [0, m-1]$, $y_0 \in [1, m-1]$and $\overline{x_y}$ is the complement of x_y.

GIDC(4) is shown in Fig. 5. The blocks in GIDC(m) consists of m switches and their connected $m(m-1)$ servers, where a GIDC(m) contains 2^m blocks. In each block, each switch is connected to $m-1$ servers, and the servers are interconnected in a similar way to DCell [5].

ISSN: 1813-4890

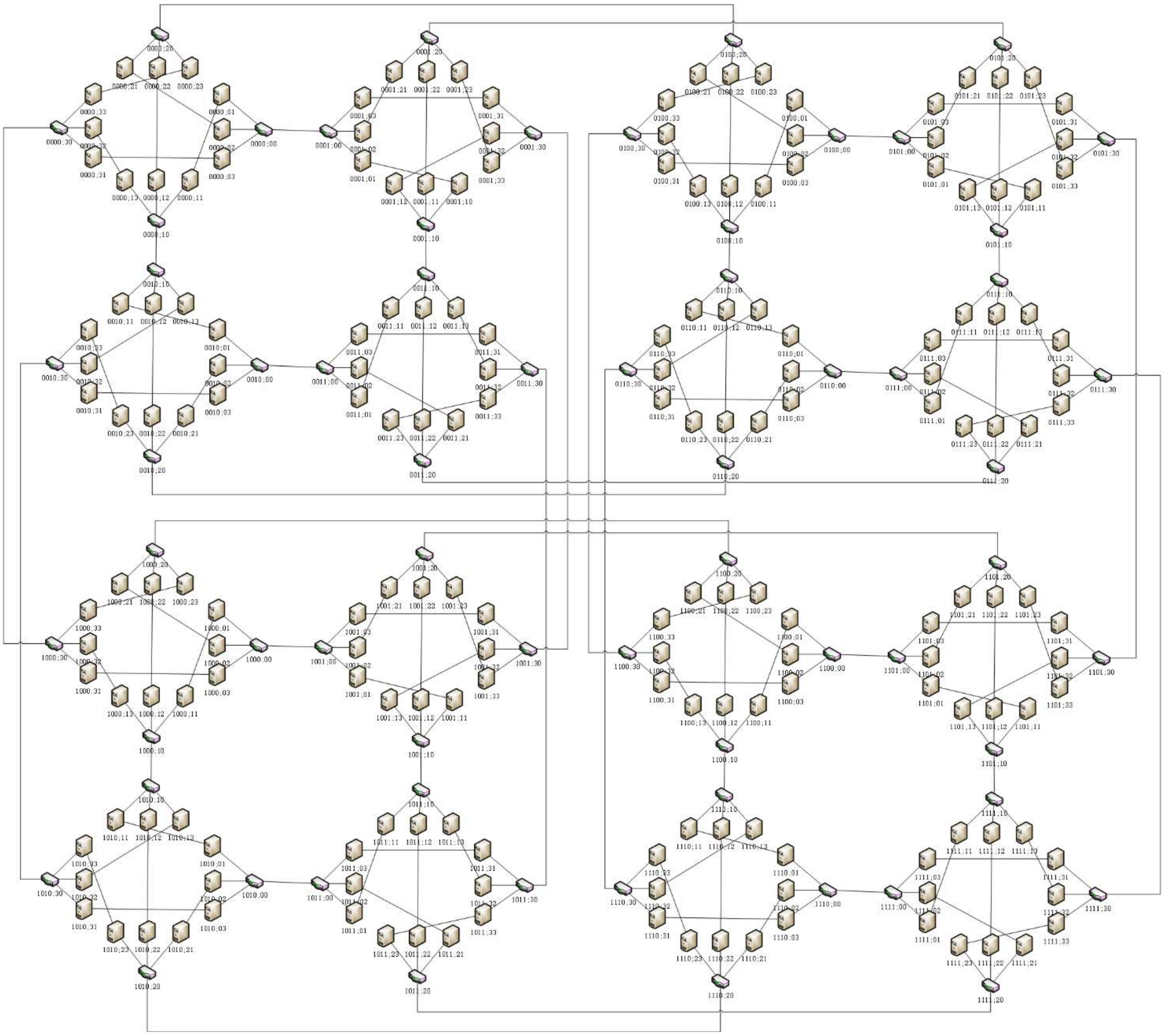

Fig. 5 GIDC(4)

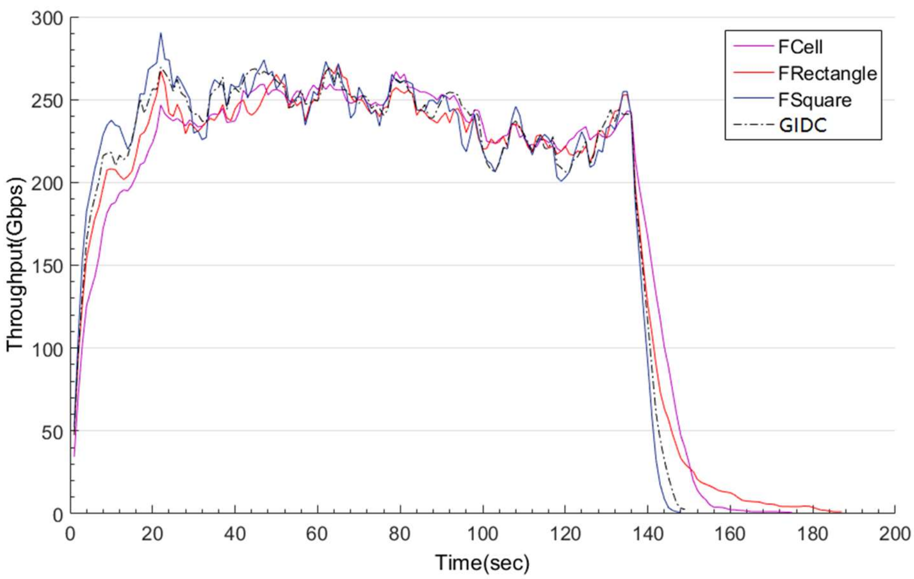

Fig. 6 The throughput of four different data center networks

4. Comparison with Existing Data Center Networks

4.1. Throughput

Based on the method proposed by Al-Fares et al [11], this paper uses the mtCloudSim [12] flow-level simulator for comparative throughput analysis of GIDC, FCell, FRectangle, and FSquare structures. The mtCloudSim treats the data center network as a network graph and customizes the capacity of each edge, and mtCloudSim formalizes flows using four-tuples, including: source host, destination host, start time, and flow size. In this paper, we use the flow workload from [13], which contains 80000 flows with a total size of 4 TB, a maximum flow size of 1 GB, and a minimum flow size of 1 KB. In addition, the source and destination hosts of each flow are randomly selected from 0 to 4096. Therefore, the workloads used in the evaluation are well representative of the data center traffic. The throughput comparison results are shown in Figure 9,the throughput of GIDC(7) is comparable to FSquare(12), which is higher than FCell(12) 17.45% and FRectangle(10) 25.5%, respectively.

4.2. Cost and Energy Consumption

In this section, this paper compares the cost and energy consumption of GIDC and other data center network structures, and we use them to construct a data center network containing the same number of servers. The price and power consumption of the switches and NICs used to construct the data center network are shown in Table 1.

Table 1. Price and Power Consumption of Switch and NICs

	Product	Posts	Price($)	Power(W)
Switch	D-Link DES-1016D	16	150	10
NIC	Intel EXPI9402PT	2	115	7

As shown in Table 2, the cost and energy consumption of GIDC are significantly smaller than those of FCell, FRectangle, and FSquare. Compared to FCell, FRectangle, and FSquare, GIDC reduces cost and energy consumption by 10.84%, 22.85%, and 29.55%, respectively. Since links are the basic equipment for constructing data center networks, the number of links for these structures is listed in this paper. As shown in Table 2, the GIDC also uses far fewer links than the other three structures.

Table 2. Cost, power and link comparison of different data centers with different scale

No. of servers	Data Centers	Cost(k$)	Energy Consumption(kw)	No.of links
1024	FCell	175.4	11	2560
	FRectangle	205.6	13.1	3072
	FSquare	233	14.9	4096
	GIDC	148.5	9.2	2048
2048	FCell	327.7	20.4	5120
	FRectangle	389.1	24.5	6144
	FSquare	419.8	26.6	8192
	GIDC	286.7	17.7	4096
4096	FCell	624.6	38.9	10240
	FRectangle	716.8	45.1	12288
	FSquare	778.2	49.2	16384
	GIDC	573.4	35.5	8192

5. Conclusion

In this paper, we propose a new dual-centric data center network architecture GIDC based on Hypercube. The GIDC(m) is constructed by using m-port switches and 2-port servers. Based on the incomplete Hypercube, we propose two incomplete GIDC architectures to achieve the Incremental scalability of GIDC. The analysis and experimental results show that GIDC has lower cost and energy consumption, good Incremental scalability and high throughput compared with FCell, FRectangle, and FSquare. Therefore, GIDC architecture is more suitable for building large data center network with low cost and energy consumption, and good Incremental scalability.

Future work will focus on research in energy efficiency. Comprehensive energy saving solutions will be proposed based on the characteristics of the traffic generated by the services and the routing algorithms used for the structure. In addition, optical switching technology or wireless transmission technology can be considered for the design and analysis of data center networks.

References

[1] Al-Fares M, Loukissas A, Vahdat A. A scalable, commodity data center network architecture[J]. ACM SIGCOMM computer communication review, 2008, 38(4): 63-74.

[2] Greenberg A, Hamilton J R, Jain N, et al. VL2: A scalable and flexible data center network[C]// Proceedings of the ACM SIGCOMM 2009 conference on Data communication. 2009: 51-62.

[3] Singh A, Ong J, Agarwal A, et al. Jupiter rising: A decade of clos topologies and centralized control in google's datacenter network[J]. ACM SIGCOMM computer communication review, 2015, 45(4): 183-197.

[4] Guo C, Lu G, Li D, et al. BCube: a high performance, server-centric network architecture for modular data centers[C]//Proceedings of the ACM SIGCOMM 2009 conference on Data communication. 2009: 63-74.

[5] Guo C, Wu H, Tan K, et al. Dcell: a scalable and fault-tolerant network structure for data centers[C]// Proceedings of the ACM SIGCOMM 2008 conference on Data communication. 2008: 75-86.

[6] Li D, Guo C, Wu H, et al. Scalable and cost-effective interconnection of data-center servers using dual server ports[J]. IEEE/ACM Transactions on Networking, 2010, 19(1): 102-114.

[7] Li D, Wu J, Liu Z, et al. Towards the tradeoffs in designing data center network architectures[J]. IEEE Transactions on Parallel and Distributed Systems, 2016, 28(1): 260-273.

[8] Ding Z L, Guo D K, Shen J W, et al. Researching data center networking topology for cloud computing[J]. Journal of National Uni-versity of Defense Technology, 2011, 33(6): 1-6.

[9] Greenberg A, Hamilton J, Maltz D A, et al. The cost of a cloud: research problems in data center networks[J]. 2008.

[10] Zhang Z, Deng Y, Min G, et al. HSDC: A highly scalable data center network architecture for greater incremental scalability[J]. IEEE Transactions on Parallel and Distributed Systems, 2018, 30(5): 1105-1119.

[11] Al-Fares M, Radhakrishnan S, Raghavan B, et al. Hedera: dynamic flow scheduling for data center networks[C] //Nsdi. 2010, 10(8): 89-92.

[12] Xie J, Deng Y. mtcloudsim: A flow-level network simulator for multi-tenant cloud[C]//2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS). IEEE, 2016: 332-339.

[13] Benson T, Akella A, Maltz D A. Network traffic characteris-tics of data centers in the wild[C]// Proceedings of the 10th ACM SIGCOMM conference on Internet measurement. 2010: 267-280.