BC3: A LINEARLY EXPANDABLE AND COST-EFFECTIVE SERVER-CENTRIC NETWORK FOR DATA CENTERS

L.Suriya[1], Dr.I.Muthulakshmi[2]
[1]PG scholar, [2]Professor
Department of Computer Science and Engineering VV college of Engineering, Tamil Nadu, India

Abstract: A massive number of server has been important and challenging problem to provide the cost-effective data centers and consistent latency performance. Server-centric data center network topology is suitable for cost efficiency, lack of expandability and imposes a obstacle for data center upgrade. Novel server-centric data center network topology called Bcube connected crossbars (BC3) can provide good network performance, good expandability. When there is a requirement for extension, we can add a new servers and switches into the current BC3 with little adjustment of the current structure. BC3 can include an extensive number of servers while keeping a very small diameter. BC3 is that its diameter increment only linearly to the network order which is better than the majority of the current server-centric networks, whose diameter increment exponentially with network order. In addition, there is a rich arrangement of parallel ways with similar length between any pair of servers in BC3, additionally it will be provide a graceful performance degradation in case of component failure.
Keywords: Data center networks, server-centric, dual-port server, network diameter, topology, expandability.

I. INTRODUCTION

Cloud computing has drawn large in size attentions recently, as it fulfills the desire of using computing resources as a service. To support cloud computing, data centers are fundamental. These days, driven by technology progresses, data centers comprising of tens or even several thousand servers have been assembled by large online suppliers, such as Google , Amazon and Microsoft , in which data center networks (DCNs) play a critical role in the performance of data centers, which can be divided into two main categories: switch-centric networks and server-centric networks. In a switch-centric network [1], [2], [3], [4] switches are assigned for a variety of tasks such as routing and addressing, while servers are only sending and receiving packets in the network. Typical examples include FatTree [5], VL2 [6] and Portland [7]. On the other hand, in a server-centric network, such as [8], [9], [10], the computational intensive tasks like routing are put into the servers, which act not only as end hosts to send and receive packets, but also as relay nodes for each other. DCell , BCube and BC3 belong to this category. A advantage of server-centric networks is that network hardware cost can be reduced extremely, as inexpensive commodity switches are sufficient given that routing tasks have been shifted to servers where computing resources are abundant. Moreover, since servers are much more programmable than switches, server-centric network structures can accelerate the process of network innovation.

In this paper, we propose a novel server-centric network topology for DCNs, called BCube Connected Crossbars, denoted as BC3 for short. BC3 is a generalized cube based network structure. Unlike BC3, which is built using servers with exactly two NIC ports, BC3 can be constructed by servers with any fixed number of NIC ports. Hence it can meet the technique advances in the future. Also, BC3 overcomes the huge expansion cost that Bcube suffers from. Meanwhile, BC3 enjoys a short diameter as well, which increases linearly to the network order. Thus the communication between end hosts in BC3 can enjoy significantly low traffic latency. We present the construction of BC3 and the addressing scheme for servers and switches within it. We also introduce an efficient routing algorithm for one-to-one communication in BC3. In addition, we make a comprehensive comparison and analysis between the proposed BC3 and existing popular server-centric networks such as BCube and DCell. Finally, we conduct simulations to evaluate the performance of BC3. The results show that BC3 achieves the best tradeoffs among many aspects, such as bandwidth provisioning, capital expenditure, average path length and performance against server failures.

II. RELATED WORK

Generalized Hypercube and Hyper bus Structures [11] . There are two types of hypercube structures, generalized hypercube (GHC) and generalized hyper bus (GHB) .The GHC structure has a low cost compared to other hypercube structures. Because of its high connectivity, the fault tolerance is quite good. It also has a low average message distance and a low traffic density in the links. A combined hardware/software architecture, Shunting[12] , that provides a lightweight mechanism for an intrusion prevention system (IPS) to take advantage of the “heavy-tailed” nature of network traffic to offload work from software to hardware. Jellyfish, a high-capacity network interconnect [13] which, by adopting a random graph topology, yields itself naturally to incremental expansion. Jellyfish also allows great flexibility in building networks with different degrees of oversubscription. DCell: A Scalable and Fault-Tolerant Network Structure for Data Centers [14] DCell also provides higher network capacity than the traditional tree based structure for various types of services. BCube: A High Performance, Server-centric Network Architecture for Modular Data Centers[15] BCube exhibits graceful performance degradation as the server and/or switch failure rate increases. Expandable and Cost-Effective Network Structures for Data Centers Using Dual-Port Servers[16] the expandability and the equal server degree. Moreover, HCN

offers a high degree of regularity, scalability, and symmetry that conform to the modular designs of data centers well. It is highly scalable to support hundreds of thousands of servers with the low diameter, low cost, high bisection width, high path diversity for the one-to-one traffic, and good fault-tolerant ability.

III. PROPOSED SYSTEM

In this paper we propose a BC3 structure. It is a recursively defined structure built with switches and dual-port servers. We describe how to build the network recursively and the addressing scheme for both servers and switches. First create a input server after that create a BC3 Creation then perform some Routing operation. They are One-to-One Routing,One-to-Many Routing and One-to-All Routing finally provide the shortest path graph. Figure 1 depicts the block diagram for BC3

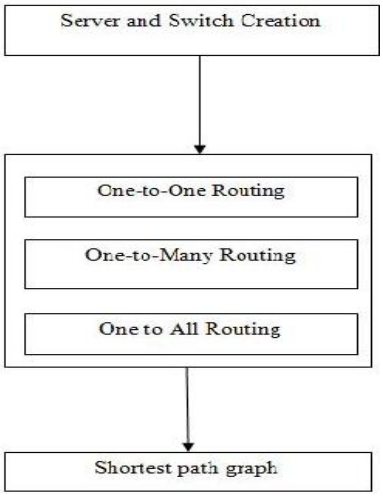

Fig. 1 Block Diagram for BC3

A. BC3 Creation

BC3 is a recursively defined structure built with switches and dual-port servers. First, we call the n servers connecting to a single n-port switch an element. Within an element, each server connects to the switch using its first port, and the second port is left for expansion purpose. We denote BC3 with order k as BC3(n, k), where n is the number of servers connected by each switch in each element. A BC3(n, 0) is simply constructed by one element and n switches, in which each server in the element connects to one of the n switches using its second port. A BC3(n, k) is constructed by n BC3(n, k − 1)s connected with n^k elements.

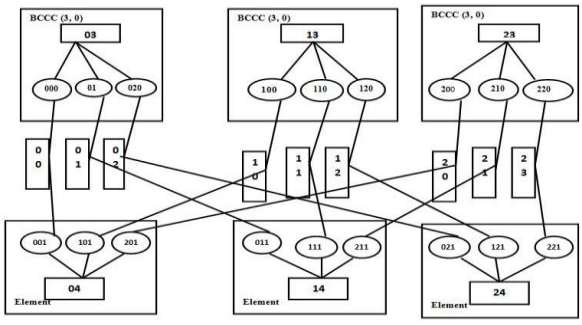

Fig 2. BC3 (3, 1) Architecture

An example of BC3 (3, 1) is shown in Fig. 2. BC3 (3, 1) has 4 BC3 (3, 0) s and 3 elements. Servers 000, 010, and 020 belong to the first BC3 (3, 0), and they are connected to switch 03 via their first ports.

Servers 001, 101and 201 belong to the same element, and they are connected to switch 04 via their first ports. Servers 000 and 001 are connected to switch 00, which is in the first BC3 (3, 0), via their second ports. Server 011 and 211 are neighbours, which are connected by switch 15 via their first ports.

B. One-to-One Routing

In one-to-one routing, a single source is sending packets to a single destination. Suppose two servers A and A_ want to communicate. Let h(A,A') be the hamming distance of A and defined as the number of different digits between their addresses. According to the aforementioned construction procedure, we can see that the maximum hamming distance between any pair of servers in a BC3(n, k) is k + 2. There are some properties are there in One-to One Routing.

Theorem 1: The diameter of a BC3(n, k) is 2(k + 1).Take BC3(8, 2) as an example .Its diameter is 6.The number of hops for this path is also 6.

Theorem 2: There are k+1 node-disjoint paths between any two servers in a BC3(n, k) , there are i different digits between these two servers except for the least significant digit, then the length of each path is at least 2i− 1 and at most 2i+5, that is, the length difference of paths between any pair of those paths is no more than 6.

C. One-to-Many Routing

In one-to-many or multicast communication, a specific source server is sending packets to many destination servers in the network. Existing multicast algorithms and protocols, such as IGMP and PIM designed for Internet, also apply to BCCC, however, they are usually destination driven algorithms or protocols.

D. One-to –All Routing

In one-to-all, or broadcast communication, one specific source server sends packets to all other servers in the network. Broadcast communication is required by many common network protocols, such as ARP, hence, an efficient broadcasting algorithm is necessary. We now design a novel efficient broadcast routing algorithm for the proposed BC3 topology. This routing algorithm takes advantage of the hypercube-like structure of BC3, and recursively broadcasts transmitted packets dimension by dimension.

The routing algorithm operates as follows. First get a permutation, denoted as Π = πk+1πkπk−1 . . .π1, of array [k+1, k, k − 1, . . . , 1]. Π represents the sequence of the dimensions
(or orders) in which the broadcast operation will be performed.

Broadcast Tree

The process can be treated as a broadcast tree, in which the top level, or the root, is the source. The packets are broadcast downward level by level. The servers on the i^{th} level act as intermediate sources to the servers on the $(i - 1)^{th}$ level and

those intermediate sources broadcast to servers in the $(i-1)^{th}$ level in the dimension defined by πi. Thus, if the source is in the πi^{th} dimension, it broadcasts to all its neighbours within the same element. Or it should route to its neighbour first that is in the πi^{th} dimension.

In this way, in each level, divide the broadcast assignment in BCCC (n, i) into n sub-assignments in n BCCC (n, i − 1) s.

BCCC (3, 1) is shown in Fig. 3.2 as an example. Suppose server 000 wants to broadcast some packets. It first broadcasts the packets to its neighbours {010, 020}. For each server in this list including 000 itself, in order to change the dimension they separately send packets to their neighbours, which are 001, 011, and 021, respectively. Then these four servers will independently broadcast packets to their neighbours, which are {101, 201}, {111, 211}, {121, 221}, {131, 231}, respectively. For each server within the four list including 001, 011, and 021, they independently broadcast to all their neighbours connected using their second port. Thus, every server gets the packets from server 000.

E. Shortest path

Theorem 3: The average shortest path length among all pairs of servers in BC3 (n, k) is

$[2(K+1) - K(n-1)/(K+1)n - 2K+1/n - 1/(K+1)n^{K}]N/N - 1$

IV. EXPERIMENTAL RESULTS

The platform used here is MATLAB R2013b and the operating system was windows 7. BC3 is a recursively defined structure built with switches and dual-port servers.

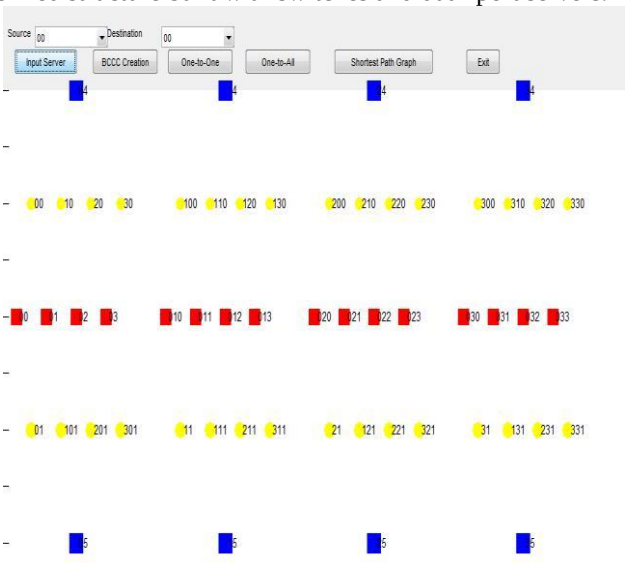

Fig. 3 Input server and switches

Fig 3 shows that there are two type of switches. They are type A switch and type B switch. Blue colour rectangles denote the type A switch. Red colour rectangles denote the type B switch. Type B switch is used for expansion purpose. Yellow colour circles denote the server. The type B switch in a BC3 starts with $0 \leq s0 \leq 3$, and that of a type A switch starts with $4 \leq s0 \leq 5$.

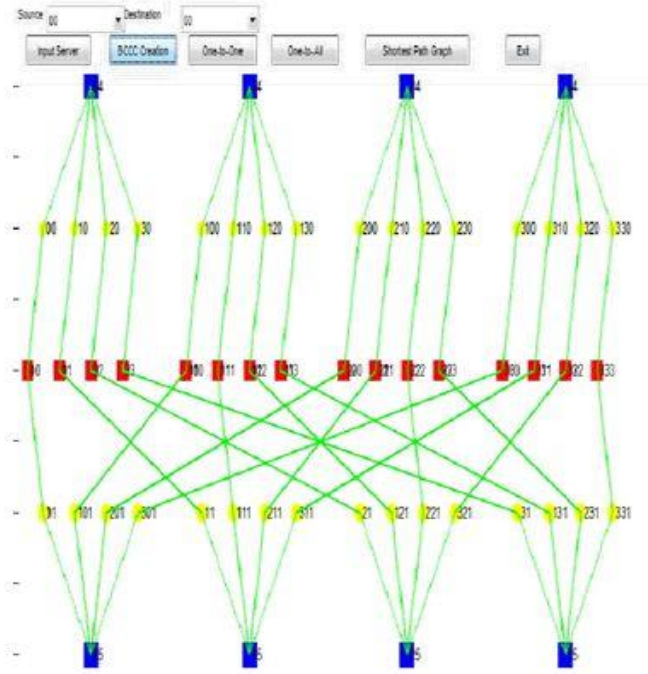

Fig. 4 BC3(4, 1) Creation

Fig 4 denote the n servers connecting to a single n-port switch an element. Within an element, each server connects to the switch using its first port, and the second port is left for expansion purpose. BC3(4, 1) has 4 BC3(4, 0)s and 4 elements. Servers 000, 010, 020 and 030 belong to the first BC3(4, 0), and they are connected to switch 04 via their first ports. Servers 001, 101, 201 and 301 belong to the same element, and they are connected to switch 05 via their first ports. Servers 000 and 001 are connected to switch 00, which is in the first BC3(4, 0), via their second ports. Server 011 and 211 are neighbours, which are connected by switch 15 via their first ports.

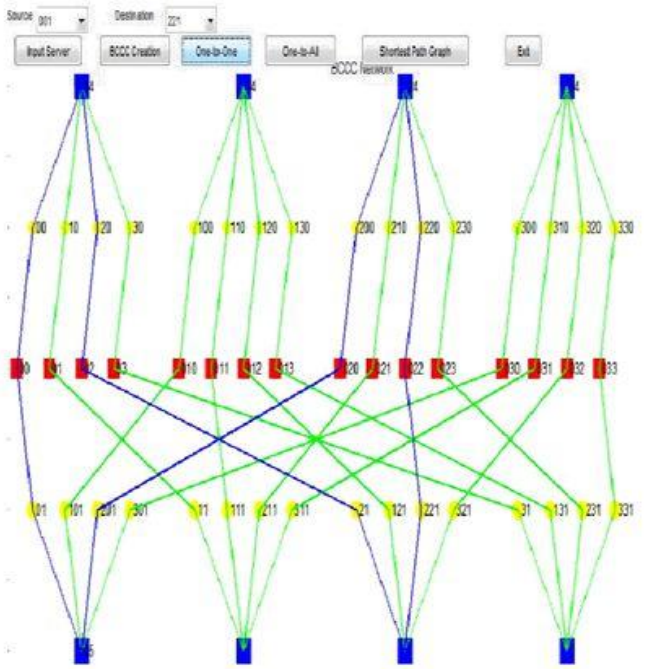

Fig. 5 One-to-One Routing

Fig. 5 as One-to-One Routing. Its diameter is 4. If we want to send a packet from server 001 to server 221,one possible path is 001, 000, 020, 021 and finally to 221. The number of hops needed for going through this path is 4. Another candidate path is from 001, to 201, 200, 220, and finally to 221. The number of hops for this path is also 4.

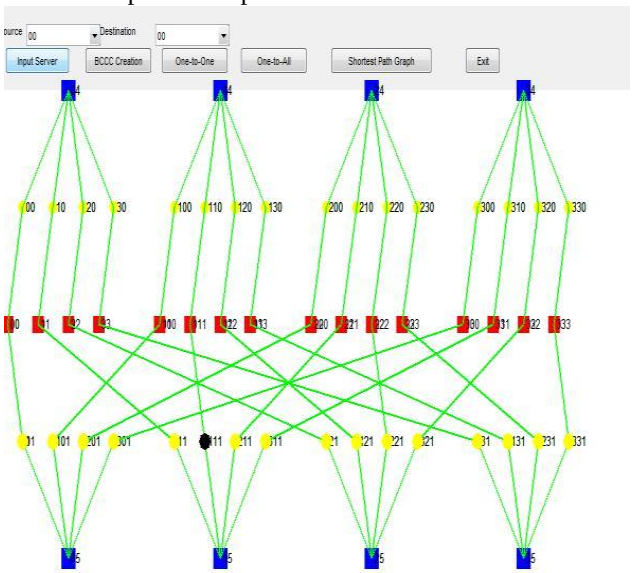

Fig. 6 One-to-All Routing

Fig. 6 shows that one specific server send packet to the all other servers in the network. In this Figure black colour circle indicate the packet flow from one specific source server to all other destination server in the network. server 000 wants to broadcast some packets. It first broadcasts the packets to its neighbors {010, 020, 030}. For each server in this list including 000 itself, in order to change the dimension they separately send packets to their neighbors, which are 001, 011, 021, 031, respectively. Then these four servers will independently broadcast packets to their neighbors, which are {101, 201, 301}, {111, 211, 311}, {121, 221, 321}, {131, 231, 331}, respectively. For each server within the four list including 001, 011, 021 and 030, they independently broadcast to all their neighbors connected using their second port. Thus, every server gets the packets from server 000.

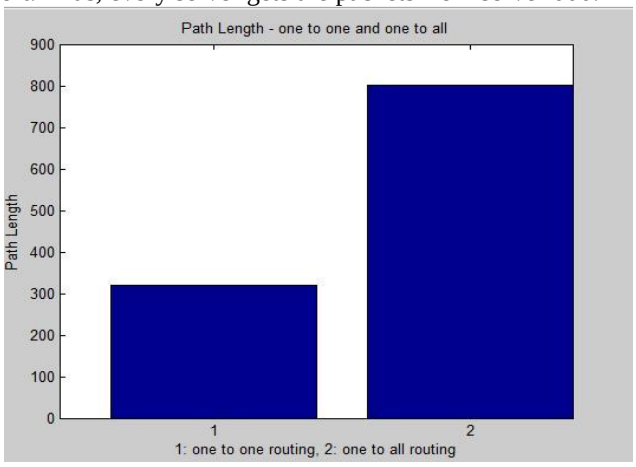

Fig.7 Shortest path graph

Fig. 7 shows the shortest path between One-to-One Routing and One-to-All routing. X-axis denote the Routing operation and Y-axis denote the path length . One-to-One Routing shows that send a packet from server 001 to server 221 . One-to-All Routing the packet flow from one specific source server to all other destination server in the network.

V. CONCLUSIONS

In this paper, we propose a novel server-centric data center network topology, called BC3, which is a recursive network structure. BC3 can be built using only dual-port servers regardless of network size, and its expansion requires little change to the existing network structure. These properties give BC3 a very good expandability. Also, there are a rich set of near-equal-length parallel paths between any pair of servers in BC3, which enables BC3 to provision sufficient transmission bandwidth and have graceful performance degradation upon failure. Finally, the diameter of BC3 only increases linearly to the network order, which means that servers will enjoy low-latency transmission even in a large-size network.

ACKNOWLEDGEMENT

This work was supported in part by Anna University recognized research center lab at VV College of Engineering, Tisaiyanvilai, Tamil Nadu, India.

REFERENCES

[1] Y. Yang and G. M. Masson “Nonblocking Broadcast Switching Networks,” IEEE Trans. Computers, vol. 40, no. 9, pp. 1005-1015, 1991.

[2] Y. Yang and G. M. Masson, “The Necessary Conditions for Clos type Nonblocking Multicast Networks,” IEEE Trans. Computers, vol. 48, no. 11, pp. 1214-1227, 1999.

[3] Y. Yang, “A Class of Interconnection Networks for Multicasting,” IEEE Trans. Computers, vol. 47, no. 8, pp. 899-906, 1998.

[4] Y. Yang and J. Wang, “Wide-sense Nonblocking Clos-Networks under Packing Strategy,” IEEE Trans. Computers, vol. 48, no. 3, pp. 265-284, 1999.

[5] M. Al-Fares, A. Loukissas and A. Vahdat, “A Scalable, Commodity Data Center Network Architecture,” ACM SIGCOMM, Aug.2008.

[6] A. Greenberg, N. Jain, S. Kandula, C. Kim, P. Lahiri, D.A. Maltz and P. Patel, “VL2: A Scalable and Flexible Data Center Network,” ACM SIGCOMM, Aug. 2009

[7] R. Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan,V. Subramanya and A. Vahdat, “PortLand: A Scalable Fault-Tolerant Layer 2 Data Center Network Fabric,” ACM SIGCOMM, Aug. 2009.

[8] C. Guo, et al. “BCube: A High Performance, Server-centric Network Architecture for Modular Data Centers,” ACM SIGCOMM, 2009.

[9] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang and S. Lu. “DCell: A Scalable and Fault-Tolerant Network Structure for Data Centers,” ACM SIGCOMM Aug. 2008.

[10] Z. Li, Z. Guo and Y. Yang, "BCCC: An Expandable Network for Data Centers," IEEE/ACM ANCS, Oct 2014

[11] L. N. Bhuyan and D. P. Agrawal, "Generalized hypercube and hyperbus structures for a computer network," IEEE Trans. Comput., vol. C-33, no. 4, pp. 323–333, Apr. 1984.

[12] J. M. Gonzalez, V. Paxson, and N. Weaver, "Shunting: A hardware/ software architecture for flexible, high-performance network intrusion prevention," in Proc. 14th ACM Conf. Comput. Commun. Secur., 2007, 139–149.

[13] A. Singla, C.-Y. Hong, L. Popa, and P. B. Godfrey, "Jellyfish: Networking data centers randomly," in Proc. 9th USENIX Conf. NSDI, 2012, p. 17.

[14] C. Guo et al., "DCell: A scalable and fault-tolerant network structure for data centers," in Proc. ACM SIGCOMM, Aug. 2008, pp. 75–86.

[15] C. Guo et al., "BCube: A high performance, server-centric network architecture for modular data centers," in Proc. ACM SIGCOMM, Aug. 2009, pp. 63–74.

[16] D. Guo, "Expandable and cost-effective network structures for data centers using dual-port servers," IEEE Trans. Comput., vol. 62, no. 7, 1303–1317, Jul. 2013.

