Parallel Processing Letters

A CLASS OF DATA-CENTER NETWORK MODELS OFFERING SYMMETRY, SCALABILITY, AND RELIABILITY*

WENJUN XIAO
School of Software Engineering
South China University of Technology
Guangzhou 510006, P. R. China

HUOMIN LIANG
School of Computer Science and Engineering
South China University of Technology
Guangzhou 510006, P. R. China

BEHROOZ PARHAMI
Department of Electrical and Computer Engineering
University of California
Santa Barbara, CA 93106-9560, USA

Received July 2012
Revised August 2012
Communicated by S. G. Akl

ABSTRACT

We propose a symmetrical scheme, by drawing results from group theory, and use it to build a new class of data center network models. The results are superior to current network models with respect to a number of performance criteria. Greater symmetry in networks is important, as it leads to simpler structure and more efficient communication algorithms. It also tends to produce better scalability and greater fault tolerance. Our models are general and are expected to find many applications, but they are particularly suitable for large-scale data-center networks.

Keywords: data-center network model, network performance, network cost, routing algorithm, structural symmetry.

1. Introduction and Related Work

Energy costs are increasing rapidly with the widespread deployment of large-scale data centers and their requisite networks. Research for large-scale and energy-efficient data-center networking is in high demand [1]–[4]. Data-center networking entails the design of

*This project was supported by the National Natural Science Foundation of China under grants 60973150 and 61170313.

the network structure and associated protocols to interconnect thousands or even hundreds-of-thousands of servers [1]–[3] at a data center, with low equipment cost, high and balanced network capacity, easy expandability, scalable performance, and extreme reliability, including robustness with respect to link and sever faults.

Proper operation of such data centers is essential to offering numerous online applications, such as search, gaming, and Web mail, as well as infrastructure services, such as GFS [5], Map-reduce [6], and Dryad [7]. It is well understood that the tree-based solution in current practice cannot meet all the requirements [8], [9]. It is thus imperative to look for systematic ways of building network structures for data centers and other applications that overcome the limitations of existing networks.

In this paper, we propose a symmetrical scheme, by drawing results from group theory, and use it to build a new class of data center network models. The results are superior to current network models with respect to a number of performance criteria. For example, the number of ports per switch can be a small constant, the servers need only two ports, the total number of switches can be sublinear in network size (e.g., $N / \log N$), and the diameter can be logarithmic in the number N of servers. Greater symmetry in networks is important, as it leads to simpler structure and more efficient communication algorithms. It also tends to produce better scalability and greater fault tolerance. Our models are general and are expected to find many applications, but they are particularly suitable for large-scale data-center networks.

Following an overview of motivations and related work in Section 1, we propose a group-theory-based symmetrical method in Section 2, where we also discuss the construction and pertinent topological properties of the resulting model. Section 3 is devoted to a routing algorithm for our proposed model. Comparison of the model to other network models appears in Section 4. Section 5 concludes the paper.

The rest of this section is devoted to a brief review of some key interconnection structures that have been proposed for data-center networks, namely the Fat-Tree [8], DCell [9], BCube [10], and FiConn [11].

Fat-Tree has three levels of switches. There are n pods, each containing two levels of $n/2$ switches, i.e., the edge level and the aggregation level. Each n-port switch at the edge level uses $n/2$ ports to connect to $n/2$ servers while using the remaining $n/2$ ports to connect the $n/2$ aggregation level switches in the pod. At the core level, there exist $(n/2)^2$ n-port switches, and each switch has one port connecting to one pod. Therefore, the total number of servers supported by the Fat-Tree network structure is $n^3/4$. Given a typical switch with $n = 48$, a total of 27 648 servers are supported.

DCell is a new, level-based structure. In DCell_0, n servers are connected to an n-port commodity switch. Given t servers in a DCell_k, $(t + 1)$ DCell_k units are used to build a DCell_{k+1}. Each of the t servers in a DCell_k connects to one of the other DCell_k units. In this way, DCell achieves high scalability and wide bisection.

BCube is also a server-centric interconnection topology, but is targeted for large, shipping-container-sized data centers, typically composed of 1K-2K servers. It is also a

level-based structure. A $BCube_0$ simply consists of n servers connected to an n-port switch. A $BCube_1$ is built from n $BCube_0$ units and n-port switches. More generally, a $BCube_k$ is constructed from n $BCube_{k-1}$ units and n^k n-port switches. Each server in a $BCube_k$ has $k + 1$ ports.

FiConn network proposed in [11] uses a recursive construction scheme similar to DCell. However, each server in FiConn can have only two interfaces. FiConn suffers from the problem of unevenly loaded links.

2. A New Class of Data-Center Network Models

2.1. *Introducing the Cayley Network Model*

We use the terminology and notational conventions of algebraic graph theory [12], [13]. Let N be the number of servers and M the number of switches. Let $T = N + M$. Assume that the network contains a cycle of T nodes. Combine the M switch nodes into a regular graph. The other nodes are 2-port server nodes.

We begin by assuming that N and M are integers of the following special forms (these restrictions can be readily relaxed): $N = k^h n^l$, $M = k^a n^b$, where $a \le h$ and $b \le l$. Then:

$$T = k^h n^l + k^a n^b = k^a n^b (k^{h-a} n^{l-b} + 1)$$

We demonstrate the construction of the network model by means of some examples. Before doing so, however, we need to introduce the definitions of Cayley graph and coset graph from algebraic graph theory [12], [13].

Definition 1. *Let G be a finite group with the identity element e and the generating set S ($e \notin S$). Then, $\Gamma = Cay(G, S)$ is the Cayley digraph on G with connection set S if $V(\Gamma) = G$ and $E(\Gamma) = \{(g, gs) \mid g \in G, s \in S\}$.*

Definition 2. *Let G be a finite group with the identity element e and the generating set S. Assume that $e \notin S$ and $g^{-1} \in S$ iff $g \in S$. Then, $\Gamma = Cay(G, S)$ is the Cayley graph on G with connection set S if $V(\Gamma) = G$ and $E(\Gamma) = \{(g, gs) \mid g \in G, s \in S\}$.*

Definition 3. *Let K be a subgroup of G (denoted as $K \le G$). The coset graph of S and K is $\Delta = Cos(G, K, S)$, whose node set is the right coset G/K. For $g, g' \in G$, Kg and Kg' have an edge iff for some $k, k' \in K$, there is $s \in S$, such that $kgs = k'g'$.*

Hypercube, cube-connected cycles, and butterfly networks are well-known examples of Cayley graphs, while de Bruijn and shuffle-exchange networks are instances of coset graphs [14]. The survey paper [15] contains an extensive list of references on Cayley graphs and their varied applications.

Let us consider an example. The de Bruijn graph is the following directed coset graph, with the undirected version being similar:

$$G = Z_n^{\,k} Z_k,\ K = Z_k,\ S = \{(0, 0, \ldots, 0, 0; 1), (0, 0, \ldots, 0, 1; 1)\},\ \Delta = Cos(G, K, S)$$

The cube-connected-cycles network is the following Cayley digraph, with its undirected version also being similar:

$$G = Z_n^{\,k} Z_k,\ S = \{(0, 0, \ldots, 0, 0; 1), (0, 0, \ldots, 0, 1; 0)\},\ \Gamma = Cay(G, S)$$

Biswapped networks form a new class of interconnection structures that have been shown to have important advantages over the popular swapped or OTIS networks [16]. Accordingly, we place some emphasis on the following example. Let Ω be any digraph with the vertex set $V(\Omega) = \{g_1, g_2, \ldots, g_n\}$ and the arc set $E(\Omega)$. The biswapped interconnection network $Bsw(\Omega) = \Sigma = (V(\Sigma), E(\Sigma))$ is a digraph with its vertex and edge sets specified as:

$$V(\Sigma) = \{\langle 0, p, g\rangle, \langle 1, p, g\rangle \mid p, g \in V(\Omega)\}$$

$$E(\Sigma) = \{(\langle 0, p, g_1\rangle, \langle 0, p, g_2\rangle), (\langle 1, p, g_1\rangle, \langle 1, p, g_2\rangle) \mid p \in V(\Omega), (g_1, g_2) \in E(\Omega)\}$$
$$\cup \{(\langle 0, p, g\rangle, \langle 1, g, p\rangle), (\langle 1, p, g\rangle, \langle 0, g, p\rangle) \mid p, g \in V(\Omega)\}$$

The definition postulates $2n$ clusters, each being an Ω digraph: n clusters, with nodes indexed $\langle 0, \text{cluster\#}, \text{node\#}\rangle$, form part 0 of the bipartite graph, and n clusters constitute part 1, with associated node indices $\langle 1, \text{cluster\#}, \text{node\#}\rangle$. Each cluster p in either part of Σ has the same internal connectivity as Ω (intracluster edges, forming the first set in the definition of $E(\Sigma)$). In addition, node g of cluster p in part 0/1 is connected to node p in cluster g of part 1/0 (intercluster or swap edges of the second set in the definition. The name "biswapped network" (BSN) arises from two defining properties of the network just introduced: when clusters are viewed as supernodes, the resulting graph of supernodes is the complete $2n$-node bipartite graph $K_{n,n}$, and the intercluster links connect nodes in which the cluster number and the node number within the cluster are interchanged or swapped.

We could continue our presentation with directed networks, deriving results for undirected networks as special cases. However, because data-center networks are usually undirected, we focus on undirected graphs in the rest of this paper. Note that the definition of $E(\Sigma)$, provided at the beginning of this section, ensures a symmetric directed network (that is, an undirected graph) Σ, when the basis network Ω is symmetric. Hence, combining the directed edges $(\langle 0, p, g\rangle, \langle 1, g, p\rangle)$ and $(\langle 1, g, p\rangle, \langle 0, p, g\rangle)$ leads to undirected versions of our biswapped networks.

According to [16], we have the following.

Theorem 1. (1) *If Ω is a Cayley graph, then so is Σ;*
(2) $|\Sigma| = 2|\Omega|^2$, *where $|\Theta|$ denotes the order or size of any graph Θ;*
(3) $D(\Sigma) = 2D(\Omega) + 2$, *where $D(\Theta)$ denotes the diameter of any graph Θ;*
(4) $deg(\Sigma) = deg(\Omega) + 1$, *where $deg(\Theta)$ denotes the degree of any graph Θ.*

According to [16], we also have the following result. Let H be a finite group and S a generator set of H, with $\Omega = Cay(H, S)$ and $H \times H$ the direct product of the group H and itself. Let $G = (H \times H)\langle t\rangle = \langle t\rangle(H \times H)$ be a semidirect product of the group $H \times H$ by the cyclic group $\langle t\rangle$, where t is an element of order 2, and $t(p, g)t = (g, p)$ for any $p, g \in H$. Let $S' = \{(e, s) \mid s \in S\} \subseteq H \times H$ and $T = S' \cup \{t\}$. Then, $Cay(G, T) = \Sigma$.

Now let $\Gamma_0 = \Omega$ and $\Gamma_1 = \Sigma$. Proceeding recursively for k steps, we can construct the graph Γ_k. It is easy to see that, by Theorem 1, we have the following.

Proposition 1. (1) $|\Gamma_k| = 2^{2^k-1}|\Omega|^{2^k}$;
(2) $D(\Gamma_k) = 2^k D(\Omega) + 2^k$;
(3) $deg(\Gamma_k) = deg(\Omega) + k$.

Also according to [16], we have the following desirable properties for Γ_k when Ω is connected.

Proposition 2. (1) If Ω has a Hamiltonian cycle, then so does Γ_k;
(2) If Ω has a shortest-path routing algorithm, then a shortest-path routing algorithm can be devised for Γ_k;
(3) Γ_k is maximally fault-tolerant (this strong result is surprising, given that Ω is only required to be connected);
(4) If $D(\Omega)$ is "small," then so is $D(\Gamma_k)$;
(5) If $deg(\Omega)$ is "small," then so is $deg(\Gamma_k)$.

Next, we propose some examples of *Cayley* networks. Intuitively, we first construct a regular graph of M nodes, and identify a Hamiltonian cycle H in it. This is possible for nearly all regular graphs. Form a cycle of length T on H by inserting the same number of nodes along each edge of H. The graph *Cayley*, formed in this way, is highly symmetric. We proceed with our first example.

Let $N = n^{k+1}$ and $M = n^k$. Then, $T = n^k(n + 1)$ and $M = N/n$. The M nodes constitute the de Bruijn graph of dimension k. The network diameter is of order $\log N$ when n is small, and the number of ports per switch may be a small constant. Our construction for $n = 2$ and $k = 3$ is depicted in Fig. 1.

Let $N = k^2n^k$ and $M = kn^k$. Then, $T = kn^k(k + 1)$ and $M = N/k$. The network contains a cycle of T nodes, and M nodes become the cube-connected-cycles graph of dimension k. The network diameter is of order $\log N$ when n is small, and the number of ports per switch may be a small constant. The latter construction for $n = 2$ and $k = 3$ is depicted in Fig. 2. The server nodes are partially drawn in order to avoid clutter.

Let $N = 2^{2^k-1}n^{2^{k+1}}$ and $M = 2^{2^k-1}n^{2^k}$. Then, $T = 2^{2^k-1}n^{2^k}(n + 1)$ and $M = N/n$. The network contains a cycle of T nodes, and M nodes become the graph Γ_k formed with k recursive steps. The network diameter is of order $\log N$ when n is small, and the number of ports per switch may be a small constant. In Fig. 3, the server nodes are not drawn in order to avoid clutter.

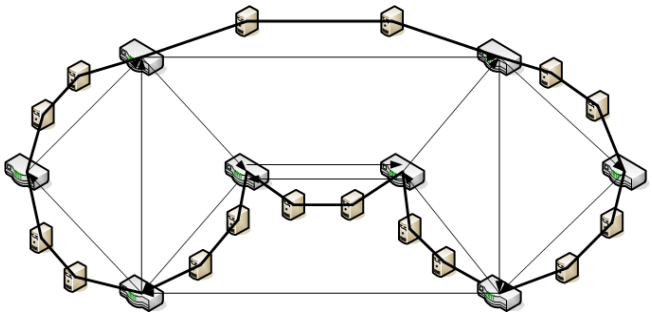

Fig. 1. *Cayley* model based on the de Bruijn graph, with $n = 2$ and $k = 3$.

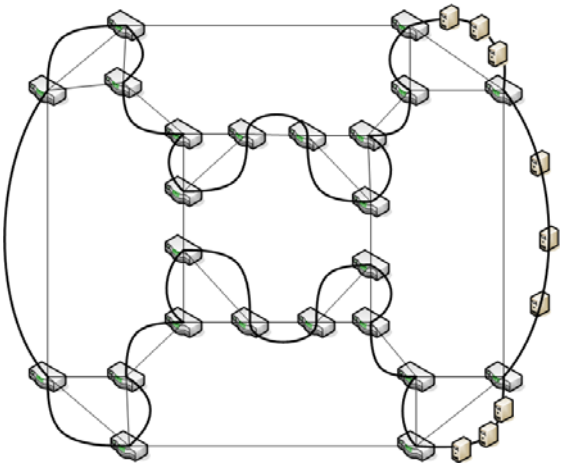

Fig. 2. *Cayley* model based on cube-connected cycles network, with $n = 2$ and $k = 3$.

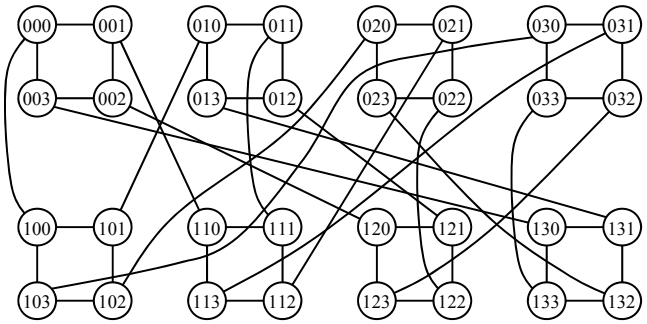

Fig. 3. An example 32-node biswapped network Σ formed from the basis graph $\Omega = C_4$. Each line represents two directed edges in opposite directions. To avoid clutter, the switch node index $\langle i, p, g \rangle$ is shown as *ipg* (i, p, g are part, cluster, and node indices, respectively) and server nodes are not drawn.

As is evident from the examples above, our data center network model, which uses symmetrical construction, is superior to current network models in many aspects of performance. For example, the number of ports per switch may be a small constant, the servers need only 2 ports, the number of switches may be of order $N / \log N$, and the diameter of network may be of order $\log N$, where N is the number of servers. Greater symmetry leads to simpler structure and communication algorithms. It also leads to greater scalability and fault-tolerance [12]. In the next subsection, we present some results on topological properties for our model.

2.2. *Topological Properties of Cayley*

It is well known that a Cayley graph is node-transitive, thus forming an apt model for the study of symmetric networks. The coset graph is regular and also possesses a number of symmetries. Data center networks have nodes of two kinds: servers and switches. Servers must be connected into cycles, if we want them to have only two ports. For symmetry, we may want that switches have similar structures. Hence they can be denoted as coset graphs (Cayley graphs, in particular). They can be chosen to be large graphs of small degree and diameter, leading to large data-center network models, because coset graphs form a very general class of graphs. The regular graph of switch nodes is of small diameter when n is small. Under these conditions, we have the following properties.

Property 1. The diameter of *Cayley* network may be of order $\log N$ for suitably small n.

Property 2. The number of ports per server in *Cayley* is 2.

Property 3. The number of ports on switches in *Cayley* may be a small constant.

3. Shortest-Path Routing Algorithm for *Cayley*

We use the cube-connected-cycles version of our network as an example. Node addressing is as follows (refer to Fig. 4). The switch node addressing is the same as in the cube-connected-cycles network, but includes an extra dimension that is assigned the value 0. For example, (110,1) in the cube-connected cycles becomes (110,1,0) in our model. The server addressing is determined within the cycle of nodes. We give each cycle a direction, as shown by the heavy black line in Fig. 4:

$$(000,0,0) \rightarrow (001,1,0) \rightarrow \ldots \rightarrow (000,1,0) \rightarrow (000,0,0)$$

Then, the server addressing between two switches is determined by the address of the switch node P that precedes it in the cycle. The aforementioned two dimensions of address for a server are the same as those of P. The final dimension begins from 1 and increases by 1 for each additional server.

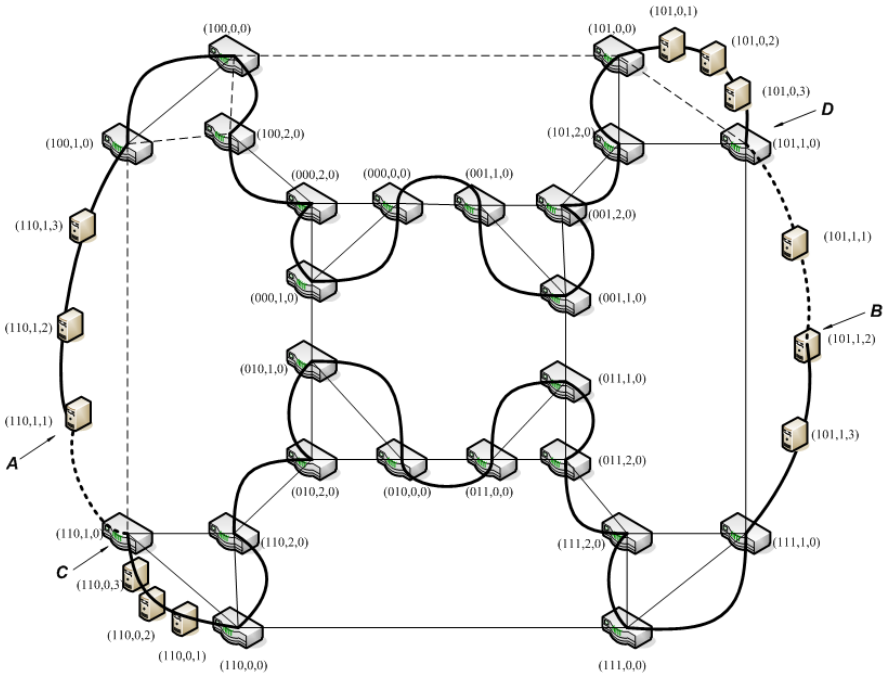

Fig. 4. Addressing and routing of *Cayley* based on the cube-connected-cycles network.

Algorithm 1. Routing algorithm for *Cayley* based on the cube-connected-cycles network.

Input: source node ($x_1x_2 \ldots x_q, y, z$) and destination node ($r_1r_2 \ldots r_q, s, t$)
Output: identifier of the next node after source on a shortest path to destination
if ($x_1x_2 \ldots x_q == r_1r_2 \ldots r_q$ && $y == s$) {
    if ($z > t$)
        return ($x_1x_2 \ldots x_q, y, z - 1$);
    else
        return ($x_1x_2 \ldots x_q, y, z + 1$);
}
if (z != 0) {
    if ($z \geq q/2$)
        return ($x_1x_2 \ldots x_q, y, z + 1$);
    else
        return ($x_1x_2 \ldots x_q, y, z - 1$);
}
if ($x_1x_2 \ldots x_q == r_1r_2 \ldots r_q$)
    return ($x_1x_2 \ldots x_q, y + 1, z$);
else {
    if (x_{q-y} != r_{q-y})
        return ($x_1x_2 \ldots x_{q-y-1}r_{q-y}x_{q-y+1} \ldots x_q, y, z$);
    else
        return ($x_1x_2 \ldots x_{q-y-1}r_{q-y}x_{q-y+1} \ldots x_q, y + 1, z$);
}

Intuitively, routing is done as follows: assume that source node is server *A* and destination node is server *B* in Fig. 4. Source *A* transmits to the nearby switch node *C*, the message finds its way to node *D* on the switch subnetwork, and *D* forwards to the nearby destination server *B* along the cycle. Routing from *C* to *D* is simple and of high performance because the switch subnetwork has high symmetry. We can add a server on each link of the switch subnetwork, when strengthening the routing function is necessary.

Routing is performed by Algorithm 1. In Fig. 4, routing from (110,1,1) to (101,1,2) is denoted by a dashed line, encompassing the steps: (110,1,1) → (110,1,0) → (100,1,0) → (100,2,0) → (100,0,0) → (101,0,0) → (101,1,0) → (101,1,1) → (101,1,2).

4. Comparing *Cayley* with Other Models

From Table1, we see that the scalability of *Cayley* is not limited by the number of server ports. The fat-tree and FiConn have the same advantage, but DCell and BCube are limited in scalability by server ports. The scalability of *Cayley* is not limited by the number of switch ports either, since the number of switch ports is a small constant *c*. DCell, BCube, and FiConn share the same advantage, but switch ports do limit the scalability of fat-tree. Building *Cayley* requires fewer switches than the other models. To summarize, the structure of *Cayley* is simpler and, thus, its communication performance is higher. Scalability and fault tolerance are also better, making it suitable for building large-scale data-center networks.

5. Conclusion

In this paper, we have proposed *Cayley* as a new data center network model based on group theory. *Cayley* has greater symmetry than other proposed models, thus leading to simpler structure and communication algorithms. Scalability and fault tolerance are also better. *Cayley* includes large graphs of small node degree and diameter, because coset graphs form a very general class of graph. Hence *Cayley* is a reasonably general model and can be tailored to applications for large-scale date-center networks.

Because of the generality and flexibility of our constructions, we believe that they will have further applications to all interconnection networks, providing an interesting area for further research. In particular, simulating the design of *Cayley*, to experimentally verify the desirable properties of fault-tolerant routing, balanced communication traffic, and resilience to node and link failures, as predicted by theory, can be postulated.

Table 1. Comparison of Fat-tree, DCell, BCube, FiConn, and *Cayley*.

	Fat-tree	DCell	BCube	FiConn	*Cayley*
Scalability limited by server ports (# ports)	No (≤ 2)	Yes ($k+1$)	Yes ($k+1$)	No (≤ 2)	No (≤ 2)
Scalability limited by switch ports (# ports)	Yes (n)	No (n)	No (n)	No (n)	No ($\leq c$)
Number of switches	$5N/n$	N/n	$(k+1)N/n$	N/n	$N/\log N$
Maximum one-to-one throughput	1	k	$k+1$	2	2

References

[1] T. Hoff, Google architecture, Nov. 2008, online document accessed on June 28, 2012: http://highscalability.com/google-architecture

[2] L. Rabbe, Powering the Yahoo! network, Nov. 2006, online document accessed on June 28, 2012: http://yodel.yahoo.com/2006/11/27/powering-the-yahoo-network

[3] Y. Zhang and N. Ansari, On architecture design, congestion notification, TCP incast and power consumption in data centers, *IEEE Commun. Surveys & Tutorials*, 2012, to appear.

[4] J. Snyder, Microsoft: datacenter growth defies Moore's law, Apr. 2007, online document accessed on June 28, 2012: http://www.pcworld.com/article/id,130921/article.html

[5] S. Ghemawat, H. Gobio, and S. Leungm, The Google file system, in *Proc. ACM Symposium on Operating Systems Principles*, 2003, 29–43.

[6] J. Dean and S. Ghemawat, MapReduce: simplified data processing on large clusters, in *Proc. Symposium on Operating Systems Design & Implementation*, 2004, 137–150.

[7] M. Isard, M. Budiu, Y. Yu, et al., Dryad: distributed data-parallel programs from sequential building blocks, in *Proc. ACM European Conference on Computer Systems*, 2007, 59–72.

[8] M. Al-Fares, A. Loukissas, and A. Vahdat, A scalable, commodity data center network architecture, in *Proc. ACM Conference on Data Communication*, 2008, 63–74.

[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, in *Proc. ACM Conference on Data Communication*, 2008, 75–86.

[10] 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, in *Proc. ACM Conference on Data Communication*, 2009, 63–74.

[11] D. Li, C. Guo, H. Wu, K. Tan, Y. Zhang, and S. Lu, FiConn: using backup port for server interconnection in data centers, in *Proc. INFOCOM*, 2009, 2276–2285.

[12] N. Biggs, *Algebraic Graph Theory* (Cambridge Univ. Press, New York, 2nd ed. 1994).

[13] B. Alspach, Cayley graphs, in *Topics in Algebraic Graph Theory*, eds. L. Beineke and R. Wilson (Cambridge Univ. Press, New York, 2004), pp. 156–178.

[14] W. Xaio and B. Parhami, Some mathematical properties of Cayley digraphs with applications to interconnection network design, *Int'l J. Comput. Math.* **82** (2005) 521–528.

[15] A. Kelarev, J. Ryan, and J. Yearwood, Cayley graphs as classifiers for data mining: the influence of asymmetries, *Discrete Math.* **309** (2009) 5360–5369.

[16] W. Xiao, B. Parhami, W. Chen, M. He, and W. Wei, Biswapped networks: a family of interconnection architectures with advantages over swapped or OTIS networks, *Int'l J. Comput. Math.* **88** (2011) 2669–2684.