Data Center Network Architectures

Juha Salo
Aalto University School of Science and Technology
juha.salo@tkk.fi

Abstract

Data centers have become increasingly essential part of Internet services and networking, thus setting major demands for a modern data center network architecture. Demands, such as support for cloud computing, efficiency, scalability and being economical, results in interesting challenges from a network architecture point of view. In this paper we explain a typical data center network architecture in the industry, the challenges modern data center networks encounter today and introduce proposed solutions by a recent research.

KEYWORDS: data center network, cloud computing, optimization

1 Introduction

Data centers run by Mirosoft, Yahoo, Google and Amazon host tens of thousands servers to provide services across the Internet. The only component that has not changed during the vast development in data centers is the networking [5].

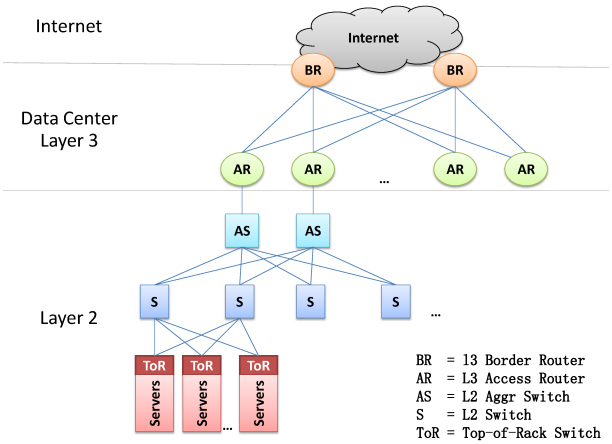

Figure 1: A typical data center network architecture [8, 7].

Typical data center network architecture usually consists of switches and routers in two- or three-level hierarchy [1]. Figure 1 is an instance of three-level design. Our figure is adapted from several papers [8, 7] and contains hierarcy of top layer core routers on the top connecting to the servers in racks at the bottom. The hierarchy consists of data center layer 3, layer 2 and Top of Rack switches connecting servers. Data center layer 3 includes requests arriving from the Internet as IP. Layer 3 access and border routers manage traffic into and out of the data center. Aggregation switches (AS) aggregate connections to layer 3 access routers (AR) and provide redundancy. Layer 2 domain contains usually several thousands servers that are isolated to server groups by virtual LAN (VLAN) partitioning. Top of Rack (ToR) switch usually connects 20 to 40 servers in the rack by 1 Gbps link. In addition, all links use Ethernet as protocol for Physical Layer.

1.1 Enterprise and Cloud Service Data Centers

Major companies, such as Google, Microsoft and Yahoo have made vast scale investments for data centers supporting cloud services. According to Greenberg et al. [6] data centers supporting cloud services differ from typical enterprise data centers. First, cloud service data centers require automation, in contrast to an enterprise data center where automation might be just partial. Second, cloud service data centers must support large economies of scale. For instance, 100 000 servers might be reality. Last, cloud service data centers should scale out, not to scale-up. Scaling out distributes workload to low cost hardware, in contrast to updating high cost hardware. Instances of an enterprise data center designs are Ethane [3] and SEATTLE [10].

According to [8], the enterprise networking architectures were originally developed for much smaller data centers, in contrast to the ones existing today. The limitations of the traditional architecture have resulted in several workarounds and patches for the protocols to keep up with the new demands on data centers.

Kandula et al. [7] explains cloud services relation to data centers and the problem with agility. Cloud services have been a major factor for the innovation of data centers and introduces agile thinking in the data centers. Cloud services provide bulk deployments and flexible reallocation of servers to different services. However, these improvements have an economic impact. Shared data center of 100 000 servers costs $12 million per month. The current data center designs prevent them being highly profitable and economical, because the designs does not provide mandatory agility. Agility in data center means responding to different demands more rapidly, instead of trying to avoid any change by preallocating resources for unkown demands. Agile design in data centers promises cost savings and improved risk management.

1.2 Cost structure

In [6] authors propose interesting data about the cost structure in a data center. Networking in data centers consumes 15% of total costs, as shown in table 1. However, networking has a more widespread impact on the whole system. Innovating the networking is the key to reduce the total costs.

Amortized Cost	Component	Sub-Components
45%	Servers	CPU, memory, storage
25%	Infrastructure	Power distribution, cooling
15%	Power draw	Electrical utility costs
15%	Network	Links, transit, equipment

Table 1: Typical data center costs [6].

The greatest portion of total costs belong to the servers. To allow efficient use of the hardware, a high level of utilization, the data centers should provide a method to dynamically grow the number of servers and allow focusing resources on optimal locations. Now the fragmentation of resources prevents the server utilization.

Reducing infrastructure costs might depend on allowing scale out model of low cost servers. Scaling out in a data center might mean shifting the responsibility of expensive qualities of servers, such as failure rate from a single server to the whole system. By allowing the network architecture to scale out, the low failure rate is ensured by having multiple cheap servers, rather than a few expensive ones.

Power related costs are similar to the network's. IT devices consume 59% of each watt delivered, 8% to distribution losses and 33% for cooling. Cooling related costs could be reduced by allowing the data centers to run hotter, thus maybe requiring the network to be more resilient and mesh-like.

Significant fraction of network related costs goes to networking equipment. Other portions of the total costs of the network relate to wide area networking, including traffic to end users, traffic between data centers and regional facilities. Reducing the network costs focuses on optimizing the traffic and data center placement.

In this paper we focus only on the problems and solutions of the data center network architectures. Section 2 of this paper covers the main problems of network architectures. Next, section 3 evaluates different proposed solutions. Last, in conclusion we summarize the the main topics.

2 Problems with network architectures

2.1 Scalability and physical constraints

The data center should scale out, in other words adding components to increase the capacity instead of replacing existing hardware [8]. According to [9], a more attractive performance-to-price ratio can be achieved by using commodity hardware, because the per-port cost is cheaper for the commodity hardware than with the more technically advanced ones. In addition to high costs of scaling up, a typical network architecture that depends on a few high-end hardware is not fault tolerant, specially on the higher level of the hierarchy [7]. Scaling out distributes the responsibility and risk to several devices, rather than reducing the risk by more advanced technology on a fewer devices.

Cabling complexity increases when the amount of network elements requiring wiring grows higher. Also, the wiring complexity might affect the setup and restrict the scalability. For instance in modular data center (MDC) design the long cables between the containers might become a practical barrier for the scalability [13].

Physical constraints [12], such as high density in racks might lead to heating and power issues. These important issues affect data center factors, including cost, reliability and uptime. A good design should take heating, placement and power consumption amont others in concern.

2.2 Resource oversubscription and fragmentation

According to several studies [8, 7] oversubscription ratio increases rapidly when moving up in the typical network architecture hierarchy, as in figure 1. Oversubscription ratio means the ratio of subscriptions to what is available. For instance, 1:20 oversubscription ratio could be 20 different 1 Gbps servers subscribed to one 1Gbps router. Ratio of 1:1 means that the subscriber can communicate with full bandwidth. In a typical network architecture the oversubscription ratio can be 1:240 for the paths crossing the top layer. Limited server-to-server capacity limits the data center performance and fragments the server pool, because unused resources can not be assigned where they are needed. To avoid this problem all applications should be placed carefully and taking the impact of the traffic in concern. However, in practice this a real challenge.

Limited server-to-server capacity [8, 7] leads to designers clustering the servers near each others in the hierarchy, because the distance in the hierarchy affects the performance and cost of the communication. In addition, access routers assign IPs topologically for the layer 2, thus placing services outside layer 2-domain requires additional configuration. In todays data centers the additional configuration is avoided by reserving resources, thus wasting resources. Even reservation can not predict if a service needs more than there is reserved, resulting in allocating resources from other services. As a consequence of the dependencies resources are fragmented and isolated.

2.3 Reliability, utilization and fault tolerance

Data centers suffer from poor reliability and utilization [7]. If some component of the data center fails, there must be some method to keep the data center functioning. Usually counterpart elements exists, so when an access router fails for instance the counterpart handles the load. However, this leads to elements use only 50% of maximum capacity. Multiple paths are not effectively used in current data center network

architectures. Two paths at most is the limit in conventional network architectures.

Techniques, such as Equal Cost Multipath Routing (ECMP) can be used to utilize multiple paths. According to Al-Fares et al. [1], ECMP is currently supported by switches, however several challenges are yet to be resolved, such as routing tables grows multiplicatively to number of paths used, thus presumably increasing lookup latency and cost.

According to [2], links in the core of data centers compared to the average are more utilized and links on the edge are affected by higher losses on average. This research is based on the SNMP data of 19 different data centers.

Fault tolerance is essential for data centers. In case of a hardware failures the system performance should degrade gracefully [9]. In optimal situation, the system should not have major performance impacts, changes should occur smoothly and recovery time should be minimal. Graceful performance degradation is difficult in a typical network architecture, for instance a failure on the upper layers in the hierarchy can lead to performance issues with millions of users for several hours [7].

2.4 Cost

Cost is a major factor that affects the data center network architecture related decisions [1]. One method to reduce costs is to oversubscribe data center network elements. However, oversubscription leads to problems as stated earlier. Next, we are introducing the results of a study [1] how maintaining 1:1 subscriptioin ratio relates to cost by having different type of network design. Table 2 represent the maximum cluster size supported by the most advanced 10 GigE and commodity GigE switches during a specific year. The table is divided in two different topologies:

- **Hierarchical design** contains advanced 10 GigE switches on layer 3 and as aggregation switches on layer 2. Commodity GigE switches are used on the edge in the hierarchical design. Until recently, the port density of advanced switches has limited the maximum cluster size. Also, aggregation switches did not have 10 GigE uplinks until recent new products. The price difference compared to Fat-tree design is significant.
- **Fat-tree** is a topology that supports building a large-scale commodity network from commodity switches, in contrast to building a traditional hierarchical network using expensive advanced switches. In this table, fat-tree is just an example of such commodity networks. Fat-tree includes commodity GigE switches on all layers in the network architecture. It is worth noting the cost difference between hierarchical and fat-tree design. The total costs of Fat-tree design during the years has reduced rapidly, because of the decreasing price trend of the commodity hardware.

2.5 Incast

Chen et al. [4] researched TCP Throughput Collapse, also known as Incast, which causes under-utilization of link capacity. A vast majority of data centers use TCP for communication between the nodes and Incast might occur in this type of many-to-one environment, which is different from the original assumptions TCP based its design. In other words, TCP does not suit for a special data center environment with low latencies and high bandwidths, thus limits the full use of all capacity.

	Hierarchical design			Fat-tree		
Year	**10 GigE**	**Hosts**	**Cost/ GigE**	**GigE**	**Hosts**	**Cost/ GigE**
2002	28-port	4,480	$25.3K	28-port	5,488	$4.5K
2004	32-port	7,680	$4.4K	48-port	27,648	$1.6K
2006	64-port	10,240	$2.1K	48-port	27,648	$1.2K
2008	128-port	20,480	$1.8K	48-port	27,648	$0.3K

Table 2: The largest cluster sizes supported by switched with an oversubscription ratio 1:1 during 2002 - 2008 [1].

In Incast a receiver requests data from multiple senders. Upon receiving the request the senders start transmitting data to the original receiver concurrently with the other senders. However, in the middle of the connection from sender to receiver is a bottleneck link resulting a collapse in the throughput the receiver receives the data. The resulted network congestion affects all the senders using the same bottleneck link.

Upgrading and increasing the buffer sizes of swithes and routers delays congestion, but in high latency and banwidth data center environment the buffers can still fill up in a short period of time. In addition, large buffer switches and routers are expensive.

3 Proposed network architectures

3.1 Fat-tree

Al-Fares et al. [1] introduces Fat-tree, as seen in figure 2, that enables the use of inexpensive commodity network elements for the architecture. All switching elements in the network are indentical. Also, there are always some paths to the end hosts that will use the full bandwidth. Further, the cost of Fat-tree network is less than traditional one as seen in table 2.

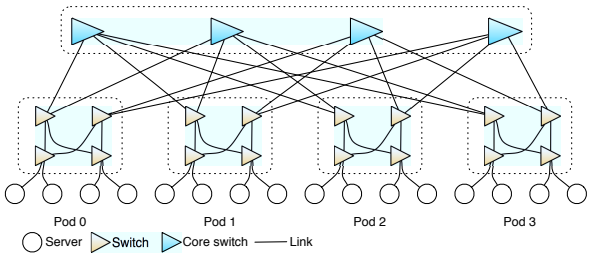

Figure 2: Fat-tree design [1].

The nature of IP/Ethernet is to establish connection between source and destination using single routing path. Single routing path leads to major performance issues in Fat-tree design. To prevent the performance issues this design pro-

poses two-level routing tables, which can be implemented in hardware using Content-Addressable Memory (CAM).

Further, the size of Fat-tree depends on the switch properties. Switch with 48 ports can support a network with 27,648 hosts and scaling out to support networks with over 100,000 hosts requires improved switches. In addition, wiring can be very serious challenge with Fat-tree design. In [1] authors propose packaging solutions in their paper.

To fully validate Fat-tree design, further work is required. However, the lack of support for performance isolation, agility and the requirement for non-existing features in commodity switches might be a major drawback concerning Fat-tree [7].

3.2 Monsoon

In [8] the authors propose a blueprint called Monsoon, a mesh-like architecture for "cloud"-services that uses commodity switches to reduce the cost and allows powerful scaling over to 100,000 servers. Monsoon improves performance by the use of Valiant Load Balancing (VLB). Figure 3 illustrates an overview of the Monsoon architecture. The architecure is divided in to Ethernet layer 2 and IP layer 3, however Monsoon focuses on the layer 2. The benefits of layer 2 include cost savings, elimination of the server fragmentation (we want to have one huge flat address space) and avoiding disturbance of the IP-layer functionality. There is also other research [11] on comparing the benefits of using Ethernet over other technologies.

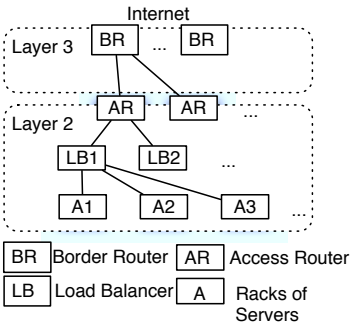

Figure 3: Monsoon design [8].

Monsoon requires layer 2 switches to have programmable control plane software, MAC-in-MAC tunneling and 16K MAC entries. Also, top-of-rack switch should handle 20 server's 1-Gbps link onto 2 10-Gbps uplinks. The upper layer switches should have 144 ports with 10-Gbps. This architecture allows over 100,000 servers with no oversubscribed links in layer 2. The load balancers (LB) can be built from commodity servers, instead of specialized and expensive hardware. IP layer 3 is responsible for dividing requests from Internet equally to access routers (AR) by Equal Cost MultiPath (ECMP).

Networking stack of a server requires replacing ARP with a user-mode process called Monsoon Agent and encapsulator, which is a new virtual Mac interface that encapsulate Ethernet frames. The Monsoon networking stack needs path information from a Directory Service. There are several ways to implement the Directory Service. Another service needed for the Monsoon design is Ingress Server, which works with Access Routers (AR). Ingress Server is required for Monsoon load spreading and encapsulation for the VLB.

3.3 BCube, MDCube

BCube [9] is a shipping-container based on modular data center (MDC) design. MDCs are formed by a frew thousands of servers that are interconnected via switches that is then packed into a 20- or 40-feet shipping-container. MDC offers short deployment time, lower cooling and manufacturing cost, and higher system and power density. Shipping container based products are already offered by major companies in the field, such as HP, Microsoft and Sun.

MDCube [13] is a structure to construct mega-data centers based on containers. Containers in MDCube follow the BCube design, which connects thousands of servers inside the container. In other words, MDCube is a design to achieve a mega-data center using BCube-based containers as building blocks.

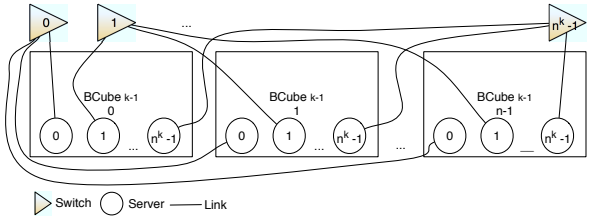

Figure 4: BCube design [9].

Figure 4 illustrates BCube [9] server centric design, which uses only commercial-off-the-shelf (COTS) switches and commodity servers. Each server has small number of network ports, that connect to mini-switches. The routing intelligence is left for the server. The authors claim Clos topology based solutions, such as Monsoon, VL2 and Fat-tree do not support one-to-x (one-to-one, one-to-several and one-to-all) well, in contrast to BCube. In addition, results show that BCube offers more graceful performance degradation than typical network architectures.

In [13] authors propose MDCube that provides good fault-tolerance, high network capacity for mega-data centers and manageable cabling complexity. BCube containers in MDCube are interconnected by using high-speed interfaces of switches in BCube. BCube containers acts as a virtual node in MDCube with the MDCube switches being virtual interfaces to these virtual nodes. MDCube is a server-centric design, thus leaving the logic to the servers. MDCube seems to require networking stack modifications for load balancing and fault tolerant routing as in server-centric manner. Routing to external networks is provided by reserving switches and servers in BCubes for the external connections.

Compared to other single structure designs, MDCube inter-container cables number is reduced almost magnitude of two orders. MDCube supports millions of servers in a data center and provides container based deployment solution.

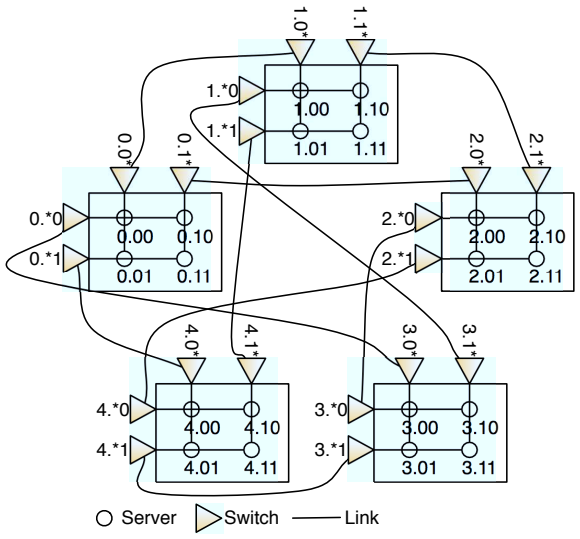

Figure 5: Example MDCube design [13].

3.4 VL2

Greenberg et al. [7] introduces VL2, a network architecture that uses Valiant Load Balancing (VLB) for traffic spreading, address resolution supporting large server pools and flat addressing to avoid fragmentation of resources. The actual topology provides path diversity. Overall VL2 is promised to solve many current problems by offering agility, since it creates an illusion of a single whole data center wide layer-2 switch by creating a virtual layer. Also, VL2 eliminates the need for overscubscriping links in the network by the network design.

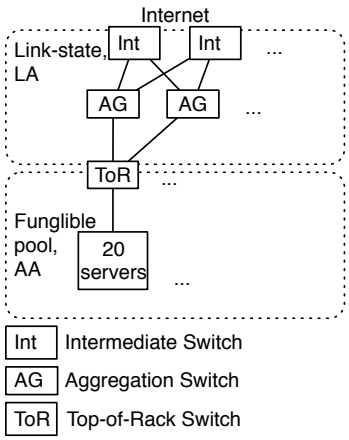

Figure 6: VL2 design [7].

Like Monsoon, VL2 requires a directory service and server agent for VL2 addressing and routing. It seems VL2 requires changes to servers' network stacks to enable VL2 adddressing and routing design. Key concepts in VL2 addressing and routing are application-specific addresses (AAs) and location-specific addresses (LAs) that are used to separate server name from locations, thus providing agility. LAs are assigned for all switches and interfaces, while AAs are only used in applications.

One VL2 design principle is to allow ipmlementation on existing hardware, so that VL2 could be taken in use even today. The authors evaluated VL2 performance by a working prototype. The results indicate that VL2 is efficient and achieves high load balancing fairness. In addition, rough cost estimates also indicate that a typical network without oversubscribed links costs 14 times more than equivalent VL2 network.

However, the authors of MDCube [13] claim that VL2 design is still expensive, since they use rather high end switches in the layer 2. For instance building a 1 million server network would require 448 448-port 10Gbps intermediate and aggregate switches.

3.5 Fixing Incast

Chen et al. research for TCP Incast solutions [4] focused on TCP-based methods, preffering existing technology over creating a new one. It might also be more cost efficient and more attractive for the data center operators. An instance of non TCP-based method is a global scheduler on the application level. Global scheduler would require modifications to all the applications, but as said earlier this type of solutions are unattractive because of the complexity and effort.

When congestion occurs the sender shifts to retransmission timeout (RTO) state. RTO is the time delay between retransmissions. In linux for example, the default RTO value is set to $200ms$. The results so far indicate that lowering the $200ms$ RTO to $200\mu s$ might have important part solving the Incast problem. However, using very low RTO values often requires high-resolution timers in the systems. Despite the requirement for high-resolution timers, the optimization of TCP is under study and results, including RTO modifications, have been promising so far.

4 Conclusion

In this paper we first introduced the current trend in the data center industry among some information about the cost structure. Next, we explained the problems with todays data center network architectures, including scalability, physical constraints, resource oversubscription and fragmentation, reliability, utilization, fault tolerance, cost and Incast. Last, we introduced some recently proposed solutions for the problems. We covered Monsoon, VL2, Fat-tree and MDCube. Each of the proposed solutions had their strengths and weaknesses, however we estimate that the ones more favorable for the industry are the ones that are deployable even today and require minimal effort for the existing hardware.

One common trend with all of the proposed solutions was to use commodity hardware, instead of expensive more advanced devices. The commodity hardware can be used to develope a network that scales out, which is one of the most important requirements new data centers demand from network architectures. Due to the complexity of the requirements, the proposed solutions had changes on various layers of the OSI model. These changes focused on switches on Link Layer, routers on Networking Layer and TCP on Transport Layer.

References

[1] M. Al-Fares, A. Loukissas, and A. Vahdat. A scalable, commodity data center network architecture. *SIGCOMM Comput. Commun. Rev.*, 38(4):63–74, 2008.

[2] T. Benson, A. Anand, A. Akella, and M. Zhang. Understanding data center traffic characteristics. In *WREN '09: Proceedings of the 1st ACM workshop on Research on enterprise networking*, pages 65–72, New York, NY, USA, 2009. ACM.

[3] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker. Ethane: taking control of the enterprise. *SIGCOMM Comput. Commun. Rev.*, 37(4):1–12, 2007.

[4] Y. Chen, R. Griffith, J. Liu, R. H. Katz, and A. D. Joseph. Understanding tcp incast throughput collapse in datacenter networks. In *WREN '09: Proceedings of the 1st ACM workshop on Research on enterprise networking*, pages 73–82, New York, NY, USA, 2009. ACM.

[5] P. Costa, T. Zahn, A. Rowstron, G. O'Shea, and S. Schubert. Why should we integrate services, servers, and networking in a data center? In *WREN '09: Proceedings of the 1st ACM workshop on Research on enterprise networking*, pages 111–118, New York, NY, USA, 2009. ACM.

[6] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel. The cost of a cloud: research problems in data center networks. *SIGCOMM Comput. Commun. Rev.*, 39(1):68–73, 2009.

[7] 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 *SIGCOMM '09: Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, pages 51–62, New York, NY, USA, 2009. ACM.

[8] A. Greenberg, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta. Towards a next generation data center architecture: scalability and commoditization. In *PRESTO '08: Proceedings of the ACM workshop on Programmable routers for extensible services of tomorrow*, pages 57–62, New York, NY, USA, 2008. ACM.

[9] 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 *SIGCOMM '09: Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, pages 63–74, New York, NY, USA, 2009. ACM.

[10] C. Kim, M. Caesar, and J. Rexford. Floodless in seattle: a scalable ethernet architecture for large enterprises. *SIGCOMM Comput. Commun. Rev.*, 38(4):3–14, 2008.

[11] A. Myers, T. E. Ng, and H. Zhang. Rethinking the service model: Scaling ethernet to a million nodes. In *ACM SIGCOMM Workshop on Hot Topics in Networking*, 2004.

[12] R. Sharma, C. Bash, C. Patel, and M. Beitelmal. Experimental investigation of design and performance of data centers. In *Thermal and Thermomechanical Phenomena in Electronic Systems, 2004. ITHERM '04. The Ninth Intersociety Conference on*, pages 579–585 Vol.1, June 2004.

[13] H. Wu, G. Lu, D. Li, C. Guo, and Y. Zhang. Mdcube: a high performance network structure for modular data center interconnection. In *CoNEXT '09: Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pages 25–36, New York, NY, USA, 2009. ACM.