A Scalable, Commodity Data Center Network Architecture

BY
MOHAMMAD AL-FARES
ALEXANDER LOUKISSAS
AMIN VAHDAT

PRESENTED BY NANXI CHEN
MAY. 5, 2009

Overview

- Background
- Fat tree topology
- Routing
- Miscellaneous
- Conclusion

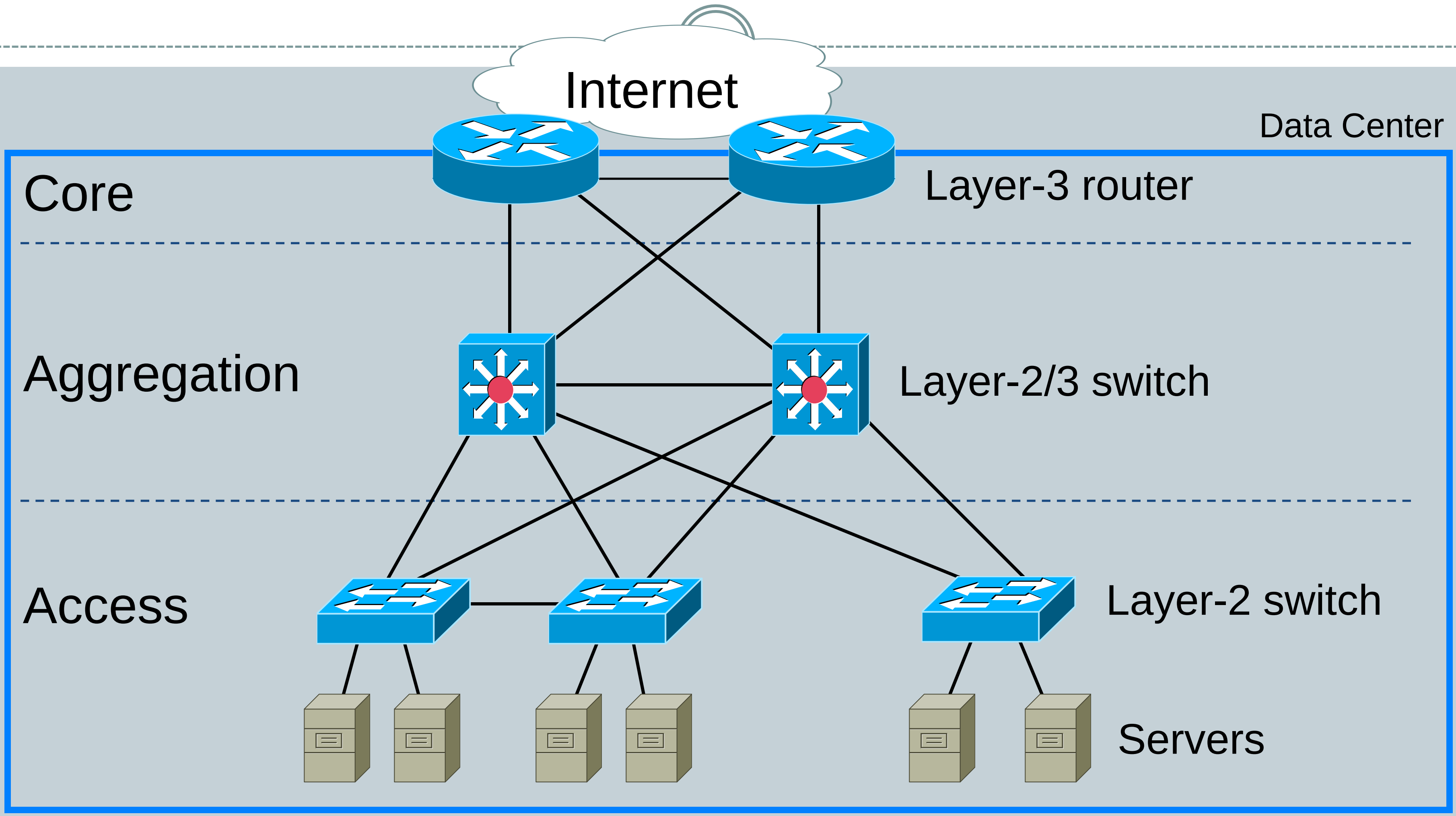
What is Data Center

- Wiki Definition: A **data center** or **datacenter** is a facility used to house computer systems and associated components.
- Nowadays, data center consisting of tens of thousands of PCs are increasingly common in universities, research labs, and companies.
- Data center can be used to do scientific computing, financial analysis, data analysis and warehousing, and providing large-scale network services.

Size of DC Network

- The size of Data centers increases exponentially
- Microsoft
  - 110,000 servers for online services, 75000 new servers to be added in 2008
  - The number of servers is expected to double every 14 months
- Google
  - Currently has 450,000 servers
  - The increasing rate is expected almost the same as Microsoft
- Yahoo!
  - hundreds of thousands

Common data center topology



Problems With common DC topology

- Single point of failure
- Core routers are bottleneck
  - Require high-end routers
  - High-end routers are very expensive
- Switching hardware cost to interconnect 20,000 hosts with full bandwidth
  - $7,000 for each 48-port GigE switch at the edge
  - $700,000 for 128-port 10 GigE switches in the aggregation and core layers.
  - approximately $37M.

Prohibitive

Properties of solutions

- Backwards compatible with existing infrastructure
  - No changes in application
  - Support of layer 2 (Ethernet) and IP
- Cost effective
  - Low power consumption & heat emission
  - Cheap infrastructure
- Scalable interconnection bandwidth
  - an arbitrary host can communicate with any other host at the full bandwidth of its local network interface.

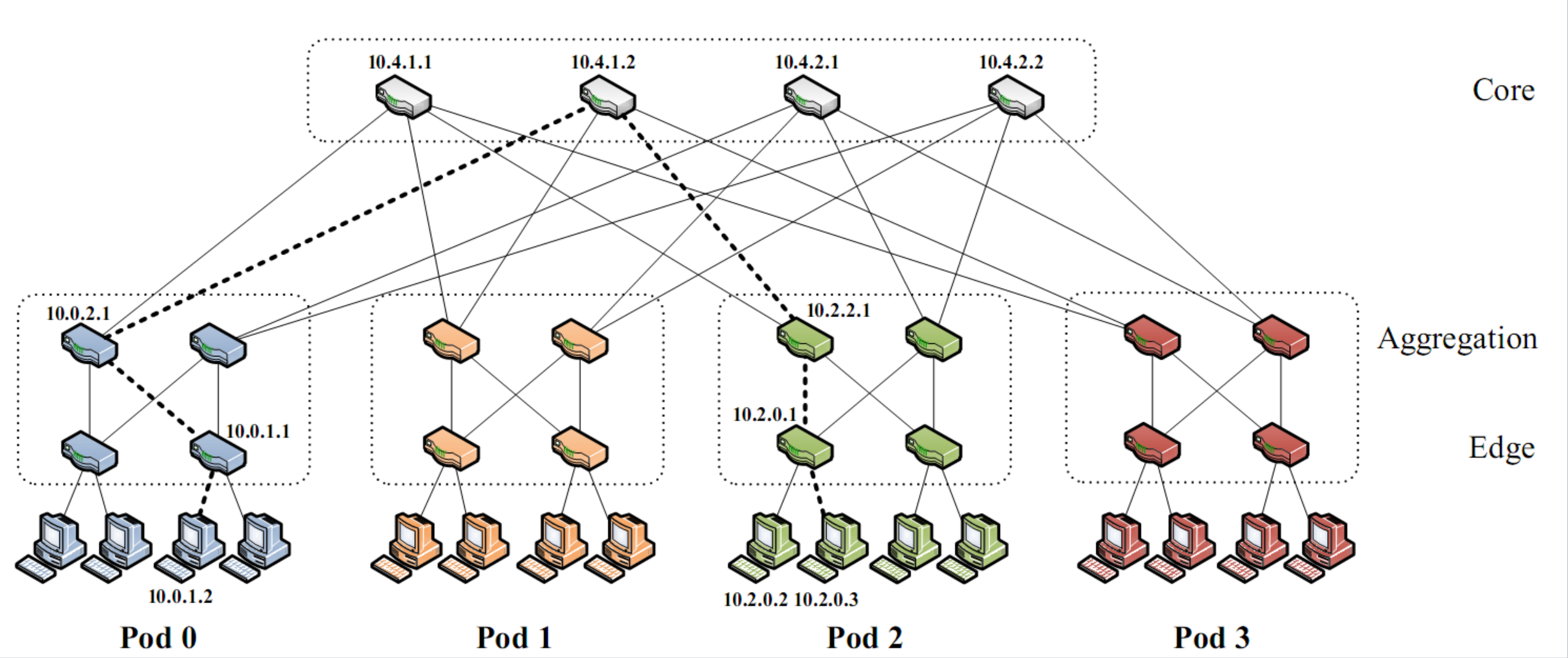
Where comes the idea?

- Today, the price differential between commodity and non-commodity switches provides a strong incentive to build large-scale communication networks from many small commodity switches rather than fewer larger and more expensive ones.
- More than fifty years ago, similar trends happened in telephone area. Charles Clos design a network topology that delivers high levels of bandwidth for many end devices by appropriately interconnecting smaller commodity switches

FAT Tree based Solution

- Connect end-host together using a fat tree topology
- *k pods*
- *Each* containing two layers of *k/2 switches*. *Each k-port switch in the* lower layer is directly connected to *k/2 hosts*. *Each of the remaining k/2 ports is connected to k/2 of the k ports in the aggregation* layer of the hierarchy.
- (*k/2)^2 k-port core switches*. *Each core switch has one* port connected to each of *k pods*. *The i th port of any core switches* connected to pod *i such that consecutive ports in the aggregation* layer of each pod switch are connected to core switches on (*k/2)* strides.
- In general, a fat-tree built with *k-port switches supports k^3 /4 hosts*.

Fat-Tree Topology



Routing

- There are multiple paths.
- But if we use routing algorithm such as OSPF, it is possible for a small subset of core switches, perhaps only one, to be chosen as the intermediate links between pods. That is, layer 3 will only use one of the existing equal cost paths.
- Packet re-ordering occurs if layer 3 blindly takes advantage of path diversity.

2 Level look-ups

- pod switches: 10.*pod.switch.1*
- core switches: 10.*k.j.i* (i, j from [1, (k/2)])
- hosts: 10.*pod.switch.ID* (ID from [2, k/2+1])
- First level is prefix lookup
  - Used to route down the topology to endhost
- Second level is a suffix lookup
  - Used to route up towards core
  - Diffuses and spreads out traffic
  - Maintains packet ordering by using the same ports for the same endhost

FAT-tree Modified

- Enforce special addressing scheme in DC
  - Allows host attached to same switch to route only through switch
  - Allows inter-pod traffic to stay within pod
- Use two level look-ups to distribute traffic and maintain packet ordering.

Prefix	Output port
10.2.0.0/24	0
10.2.1.0/24	1
0.0.0.0/0	

→

Suffix	Output port
0.0.0.2/8	2
0.0.0.3/8	3

Diffusion Optimizations

- Flow classification
  - Eliminates local congestion
  - Assign to traffic to ports on a per-flow basis instead of a per-host basis
- Flow scheduling
  - Eliminates global congestion
  - Prevent long lived flows from sharing the same links
  - Assign long lived flows to different links

Experiment Setup

- 20 switches and 16 end hosts
- Multiplex these them onto ten physical machines, interconnected by a 48-port ProCurve 2900 switch with 1 Gigabit Ethernet links.
- Each pod of switches is hosted on one machine; each pod's hosts are hosted on one machine; and the two remaining machines run two core switches each.
- Both the switches and the hosts are Click configurations, running in user level. All virtual links between the Click elements in the network are bandwidth-limited to 96Mbit/s to ensure that the configuration is not CPU limited.
- Each host generates a constant 96Mbit/s of outgoing traffic.

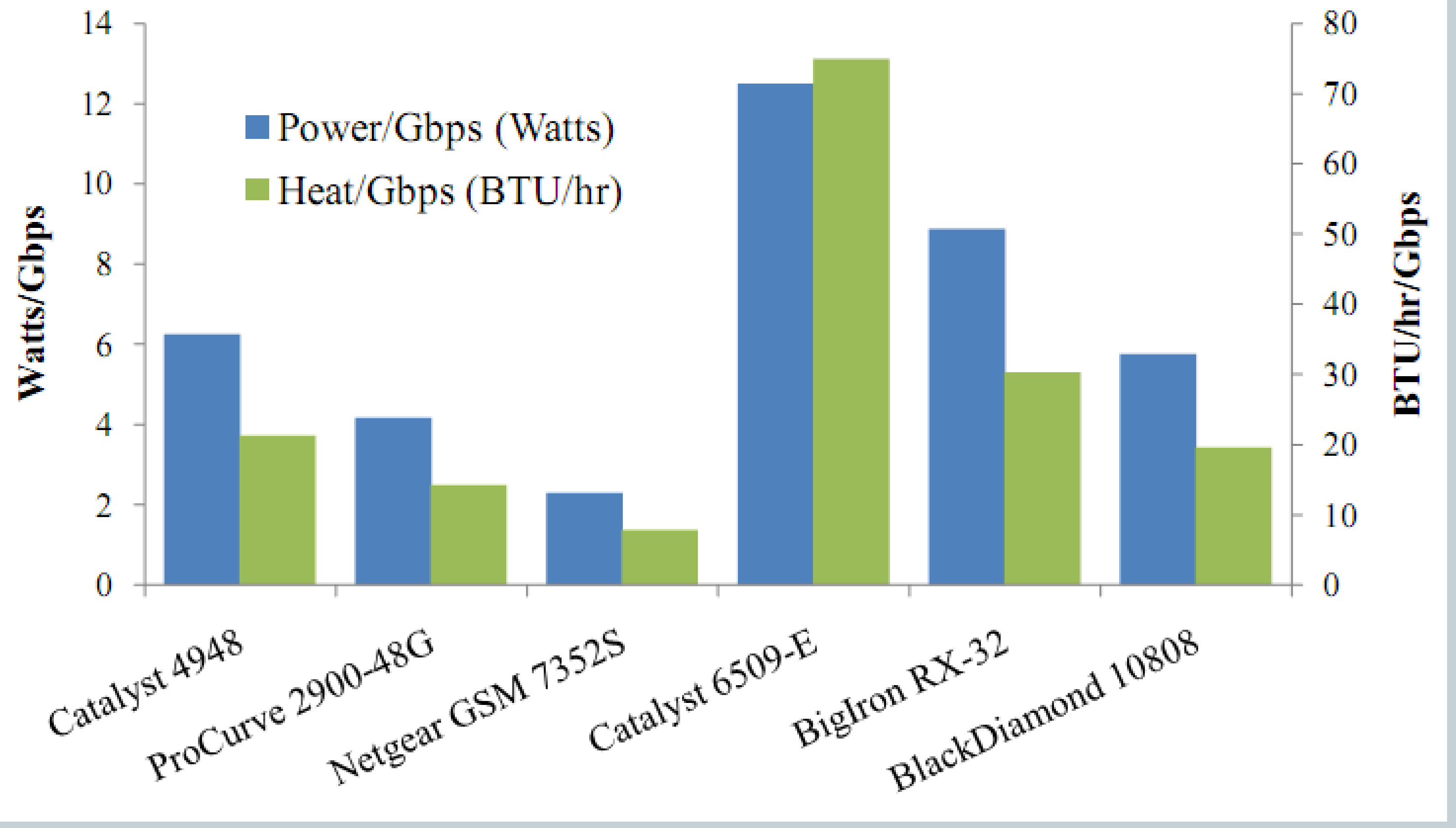
Results: Network Utilization

Test	Tree	Two-Level Table	Flow Classification	Flow Scheduling
Random	53.4%	75.0%	76.3%	93.5%
Stride (1)	100.0%	100.0%	100.0%	100.0%
Stride (2)	78.1%	100.0%	100.0%	99.5%
Stride (4)	27.9%	100.0%	100.0%	100.0%
Stride (8)	28.0%	100.0%	100.0%	99.9%
Staggered Prob (1.0, 0.0)	100.0%	100.0%	100.0%	100.0%
Staggered Prob (0.5, 0.3)	83.6%	82.0%	86.2%	93.4%
Staggered Prob (0.2, 0.3)	64.9%	75.6%	80.2%	88.5%
Worst cases:				
Inter-pod Incoming	28.0%	50.6%	75.1%	99.9%
Same-ID Outgoing	27.8%	38.5%	75.4%	87.4%

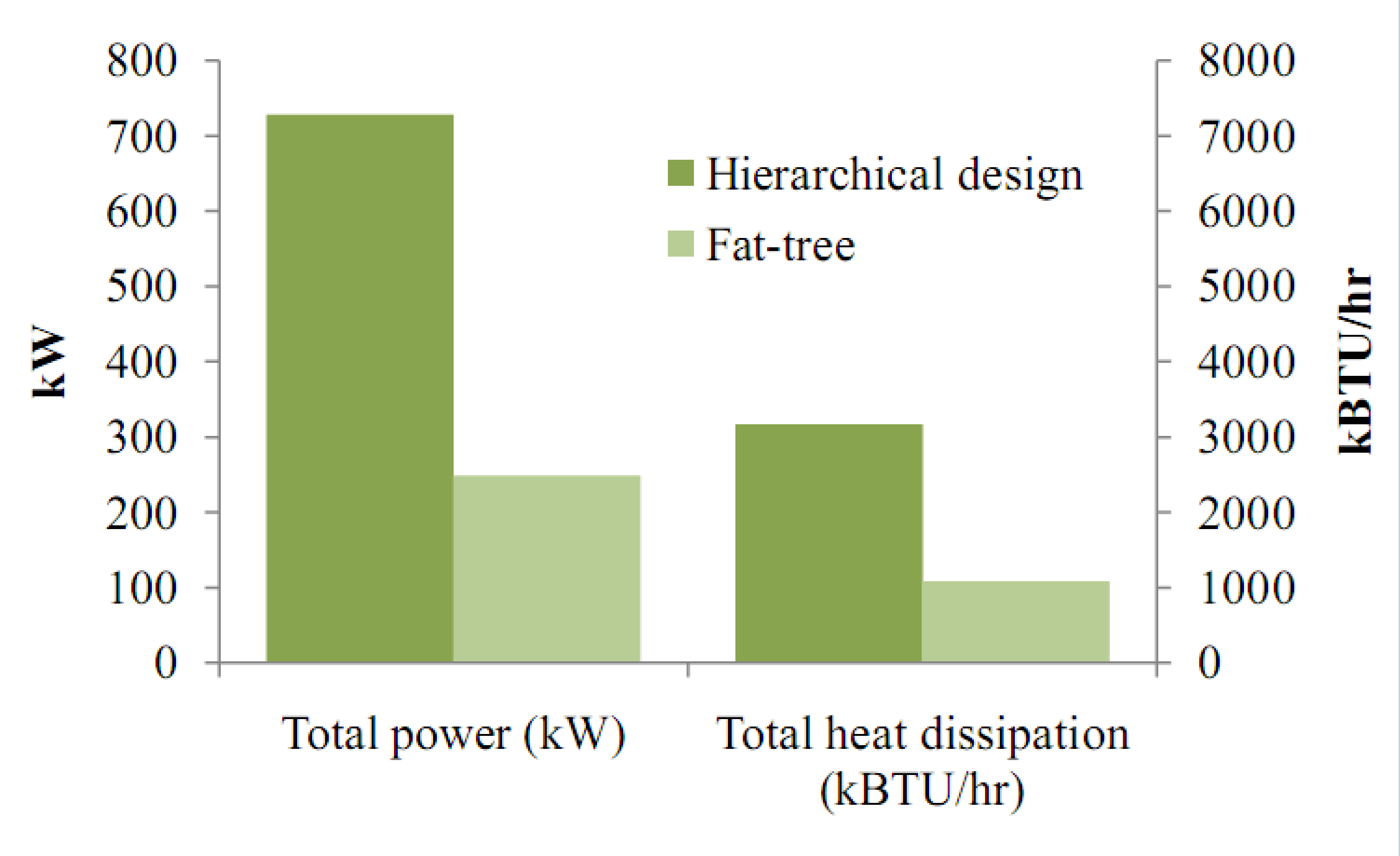
Memory and latency for Flow Scheduling

k	Hosts	Avg Time/ Req (μs)	Link-state Memory	Flow-state Memory
4	16	50.9	64 B	4 KB
16	1,024	55.3	4 KB	205 KB
24	3,456	116.8	14 KB	691 KB
32	8,192	237.6	33 KB	1.64 MB
48	27,648	754.43	111 KB	5.53 MB

Cost of maintaining switches



Results: Heat & Power Consumption



They also talk about

- Fault-Tolerance (the impact of different link failure)
- Implementation
- Packaging

Conclusion

- Develop a data center communication architecture that leverages commodity Ethernet switches to deliver scalable bandwidth for large-scale clusters.
- Base our topology around fat-tree and develop techniques to perform scalable routing while remaining backward compatible with Ethernet, IP, and TCP.
- Overall, fat-tree architecture is able to deliver scalable bandwidth at significantly lower cost than existing techniques.

Key to the SUCCESS

- Choosing a good topic is very important. Active Net, p2p, DC… Last year is Data Center year. Prof S. Lu said this year Sigcomm will accept a number of DC paper.

Key to the SUCCESS

- Practical work. First of all, the topic of data center itself is practical. Second, their design concerns many industry rules: cost, power, heat dissipation, packaging and backward compatibility. Of course, they also implement their design.

Key to the SUCCESS

- Deep, complete, detailed and explainable. General idea-fat tree topology, innovative? No. But they dig out nearly all the problems may be encountered.
- The paper is well organized. It convinces readers that their work is much better than the hierarchy topology because there are so many advantages: power/cost/heat dissipation/bandwidth. However, the real advantage is only one, they do not use high-end switches.
- Define goals and point out potential problems to avoid being attacked.