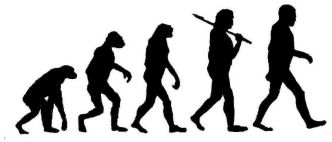
Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network

Singh, A. et al.
Proc. of ACM SIGCOMM '15, 45(4):183-197, Oct. 2015



Chun-Yu and Zaina

Outlines

- Introduction
- Network Evolution
- External Connectivity
- Software Control
- Experience
- Conclusion & Discussion

Introduction

Source: https://www.google.com/about/datacenters/inside/locations/index.html

Introduction (Cont'd)

- Bandwidth demands are doubling every 12 - 15 months

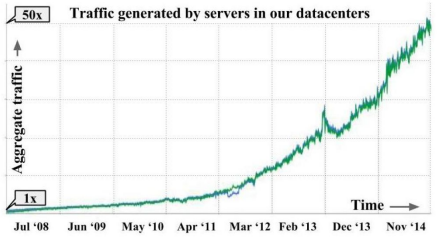


Figure 1: Aggregate server traffic in our datacenter fleet.

Introduction (Cont'd)

- The old architecture suffered from performance and costs
  - Limited bandwidth (average ~ 100Mbps per host)

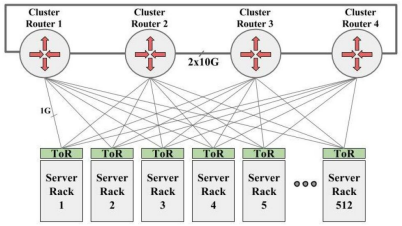


Figure 2: A traditional 2Tbps four-post cluster (2004). Top of Rack (ToR) switches serving 40 1G-connected servers were connected via 1G links to four 512 1G port Cluster Routers (CRs) connected with 10G sidelinks.

Introduction (Cont'd)

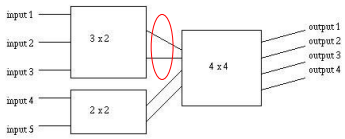
- The evolution of Google's data centers => Motives and Solution Concepts
  - Clos Network
  - Merchant Silicon
  - Centralized Control Protocol

Datacenter Generation	First Deployed	Merchant Silicon	ToR Config	Aggregation Block Config	Spine Block Config	Fabric Speed	Host Speed	Bisection BW
Four-Post CRs	2004	vendor	48x1G	-	-	10G	1G	2T
Firehose 1.0	2005	8x10G 4x10G (ToR)	2x10G up 24x1G down	2x32x10G (B)	32x10G (NB)	10G	1G	10T
Firehose 1.1	2006	8x10G	4x10G up 48x1G down	64x10G (B)	32x10G (NB)	10G	1G	10T
Watchtower	2008	16x10G	4x10G up 48x1G down	4x128x10G (NB)	128x10G (NB)	10G	nx1G	82T
Saturn	2009	24x10G	24x10G	4x288x10G (NB)	288x10G (NB)	10G	nx10G	207T
Jupiter	2012	16x40G	16x40G	8x128x40G (B)	128x40G (NB)	10/40G	nx10G/ nx40G	1.3P

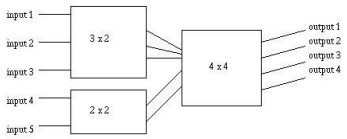
Table 2: Multiple generations of datacenter networks. (B) indicates blocking, (NB) indicates Nonblocking.

Blocking and Non-Blocking

- Consider in a circuit switching network
  - Non-blocking refers to that regardless of the settings in the switch, if an input and output are not already busy, then the switch can connect them.



A Simple Blocking Network

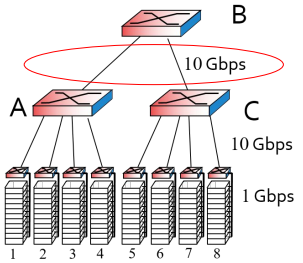


A Simple Nonblocking Network

Source: http://people.seas.harvard.edu/~jones/cscie129/nu_lectures/lecture11/switching/clos_network/clos_network.html

Traditional Data Center Network

- Traditional Tree Topology Suffers from
  - Low bandwidth utilization
    - Different bisection bandwidth induces bottleneck
  - Single node failure



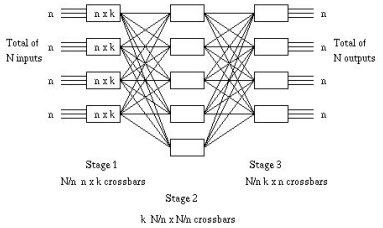
Tree Topology

Source: EECS 489 Slides

Clos Network

- Connecting a large number of ports by using only small-sized switches

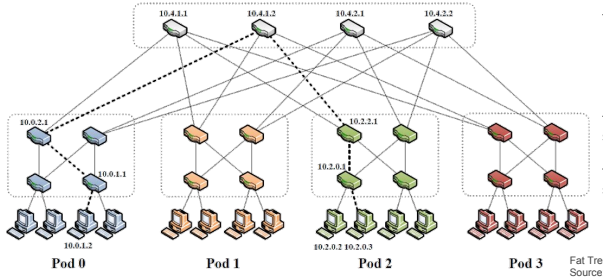
- Ex. Suppose we have N inputs and N outputs and we divide the inputs and outputs into N/n n x k groups
  - If k >= 2n +1,
    the network is non-blocking



Source: http://people.seas.harvard.edu/~jones/cscie129/nu_lectures/lecture11/switching/clos_network/clos_network.html

Clos Network Example (Fat Tree Topology)

- Benefits
  - Tolerance of node failure
  - Increase throughput by Equal Cost Multi-Path (ECMP) routing



Fat Tree Topology
Source: Al-Fares, Loukissas, and Vahdat, "A Scalable, Commodity Data Center Network Architecture," SIGCOMM '08.

Network Evolution

Datacenter Generation	First Deployed	Merchant Silicon	ToR Config	Aggregation Block Config	Spine Block Config	Fabric Speed	Host Speed	Bisection BW
Four-Post CRs	2004	vendor	48x1G	-	-	10G	1G	2T
Firehose 1.0	2005	8x10G 4x10G (ToR)	2x10G up 24x1G down	2x32x10G (B)	32x10G (NB)	10G	1G	10T
Firehose 1.1	2006	8x10G	4x10G up 48x1G down	64x10G (B)	32x10G (NB)	10G	1G	10T
Watchtower	2008	16x10G	4x10G up 48x1G down	4x128x10G (NB)	128x10G (NB)	10G	nx1G	82T
Saturn	2009	24x10G	24x10G	4x288x10G (NB)	288x10G (NB)	10G	nx10G	207T
Jupiter	2012	16x40G	16x40G	8x128x40G (B)	128x40G (NB)	10/40G	nx10G/ nx40G	1.3P

Table 2: Multiple generations of datacenter networks. (B) indicates blocking, (NB) indicates Nonblocking.

Firehose 1.0

Goal: Deliver 10Gbps nonblocking bisection b/w to each of 10K servers

- Fabric switch: 4x10G ports facing up & 4x10G ports facing down - except top most stage
- ToR switch delivered 2x10GE ports to the Fabric & 24x1GE south facing ports (20x1GE ports connected to server)
- Agg. block hosts 16 ToRs (320 m/c) & exposed 32x10G ports towards 32 spine blocks
- Spine block had 32x10G toward 32 agg. Blocks => fabric that scales to 10K m/c at 1G average b/w to any m/c in the fabric

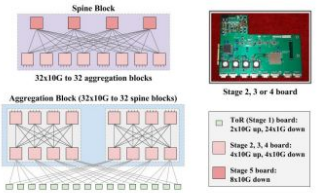


Figure 5: Firehose 1.0 topology. Top right shows a sample 8x10G port fabric board in Firehose 1.0, which formed Stages 2, 3 or 4 of the topology.

Drawbacks:

- ToR switch had low radix: causing issues when links failed
- On integrating switching fabric into servers via a PCI board, -> low server uptime => servers crashed & upgraded more frequently with long reboot times
- Worse for servers housing ToRs that connect multiple servers to topology
- Wiring complexity, Electric reliability, Availability

Never went into production!!!

Firehose 1.1 / FH1.1

First Production Clos!!

Figure 6: Firehose 1.1 packaging and topology. The top left picture shows a linecard version of the board from Figure 5. The top right picture shows a Firehose 1.1 chassis housing 6 such linecards. The bottom figure shows the aggregation block in Firehose 1.1, which was different from Firehose 1.0.

- Custom enclosures standardized around the compact PCI chassis with 6 independent linecards & SBC to control them via PCI
- No backplane in the fabric chassis to interconnect the switch chips
- All ports connected to external copper cables were wired on the d/c flow
- Separate CPN to configure & manage the SBCs of the fabric

- Spine block was the same as FH1.0
- Edge aggregation block wasn't.
  - ToR had 4x10G uplinks & 48x1G links to servers
  - ToRs were developed as separate 1RU switches with their own CPU controller
  - Buddy two ToR switches together : 2 were connected to the fabric & 2 were connected sideways to the paired ToR
- Stage 2 & 3 switches were cabled in a single vlock
- Each ToR had 40 m/cs and the FH1.1 agg. Block could scale to 640 m/cs at 2:1 subscription (2x & more robust)
- EOE cables were developed for the interconnect b/w ToRs and the next stage of FH switching infra

Figure 7: Two Firehose racks (left), each with 3 chassis with bulky CX4 cables from remote racks. The top right figure shows an aisle of cabled racks.

EOE cable capable of spanning 100m compared to 14m CX4

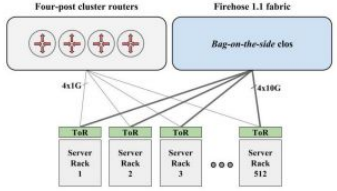


Figure 8: Firehose 1.1 deployed as a *bag-on-the-side Clos* fabric.

Concerns:

- Deploying unproven new n/w technology
- ToR forwards default traffic to the 4 post cluster
- More specific intra cluster traffic uses uplinks to FH1.1

Watchtower: Global Deployment

3G Cluster Fabric - using next gen merchant silicon chips, 16x10G to build a traditional switch chassis w/ a backplane

- 8 line cards, each w/ 3 switch chips
  - 2 chips have half their ports externally facing = 16x10GE SFP+ ports
  - All 3 chips connect to a backplane for port to port connectivity
- Easier deployment, Larger fabrics w/ more b/w to individual servers
- Less cabling complexity

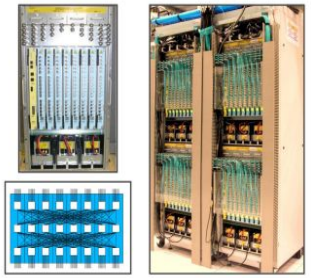


Figure 9: A 128x10G port Watchtower chassis (top left). The internal non-blocking topology over eight linecards (bottom left). Four chassis housed in two racks cabled with fiber (right).

Advantages

- Bundling reduces complexity
- Manufacturing fiber in bundles is more cost effective : reduced cost & expedited bringup

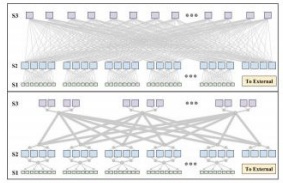


Figure 10: Reducing deployment complexity by bundling cables. Stages 1, 2 and 3 in the fabric are labeled S1, S2 and S3, respectively.

# Individual cables	15872
# S2-S3 bundles (16-way)	512
Normalized cost of fiber/m in 16-way bundle	55%
# S2-ToR bundles (8-way)	960
Normalized cost of fiber/m in 8-way bundle	60%
# Total cable bundles	1472
Normalized cost of fiber/m with bundling (capex + opex)	57%

Table 3: Benefits of cable bundling in Watchtower.

Due to variation in b/w demand of individual clusters

- Enabled watchtower fabrics to support depopulated deployment (initially deployed only 50% of max bisection b/w)
  - If demand grew, populate it to 100%

A	B
50% capacity, depopulates half of S2 switches	50% capacity, depopulates half of S3 switches
2x more depop. elements	½ depopulated elements
Higher initial cost	Gradual upfront cost
ToR has 1/2 burst b/w	ToR has twice burst b/w
Watchtower, Saturn	Jupiter

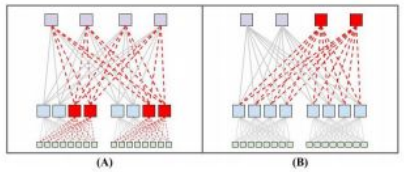


Figure 11: Two ways to depopulate the fabric for 50% capacity.

Saturn: Fabric Scaling and 10G Servers

- Increase server b/w demands & increase max cluster scale
- 24x10G merchant silicon building blocks
- 12-linecards to provide a 288 port non-blocking switch
- Chassis are coupled with new Pluto single-chip ToR switches
- Servers can burst at 10Gbps across the fabric for the first time

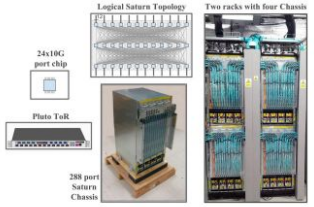


Figure 12: Components of a Saturn fabric. A 24x10G Pluto ToR Switch and a 12-linecard 288x10G Saturn chassis (including logical topology) built from the same switch chip. Four Saturn chassis housed in two racks cabled with fiber (right).

Jupiter: A 40G Datacenter-scale Fabric

- Upgrading networks by forklifting existing clusters stranded hosts already in production
- Fabric supports heterogeneous hardware & speeds

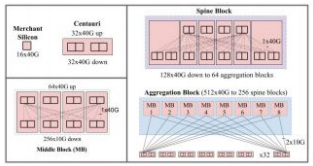


Figure 13: Building blocks used in the Jupiter topology.

- A Centauri Chassis, a 4 RU chassis with 2 linecards, each with 2 switch chips with 16x40G ports controlled by a separate CPU linecard
- Configurable to be 4x10G or 40G (burst b/w)
- 4 Centauri Chassi = MB => 2 stage blocking network with 256x10G links for ToR & 64x40G for rest of the fabric
- Ach ToR connects to 8 such MBs with dual redundancy
- Full pop / depop (agg. blocks)
- 1.3 Pbps bisection b/w among servers

Figure 14: Jupiter Middle blocks housed in racks.

External Connectivity

WCC: Decommissioning Cluster Routers

- First few Watchtower deployments, all cluster fabrics = bag-on-the-side n/w coexisiting with legacy n/w
- Limiting ToR burst b/w was a downside

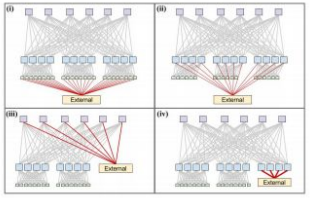


Figure 15: Four options to connect to the external network layer.

- WCC : Connect the fabric directly to inter-cluster n/w area with CBRs
  - Reserve some links from each ToR
  - Reserve ports in each agg. Block
  - Reserve ports in each spine block
  - **Build a separate agg. Block for external connectivity**
- Parallel Links between each CBR switch in these blocks & external switch
- Standard eBGP between the CBRs & inter cluster n/w switches
- CBRs learnt the default route via BGP & redistributed the route via Firepath

WCC enabled:

- Cluster fabric to be truly standalone
- Unlocked high throughput bulk data transfer between clusters
- Modular h/w & s/w of the CBR switch came into play in several use cases of networking hierarchy

Inter-Cluster Networking

- CBRs for WCC enabled clusters introduced 2.56Tbps - 5.76Tbps in fabrics
- External layers were still based on expensive port-constrained vendor gear
- Hence REPLACE!

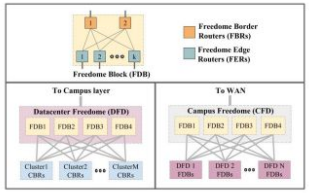


Figure 16: Two-stage fabrics used for inter-cluster and intra-campus connectivity.

- Using BGP (coming ahead), build 2 stage fabrics to use BGP at Inter cluster & intra campus connectivity layers
- Freedome blocks: 8x more cluster facing that next hierarchy level ports
- Each Block
  - Freedome Edge routers OR Freedome Border routers
  - eBGP to connect to both north & south facing peers
  - iBGP internal to each Block

Datacenter Freedome:

- 4 independent blocks connecting multiple clusters in the same datacenter
- Inter cluster traffic local to the same building travels from the source cluster's CBR layer to the Datacenter Freedome (local to the Edge Router layer)
- Freedome Border Router ports - connect to campus connectivity layer on the north
- 8x more b/w for traffic within a building
- ALSO has 4 Independent Freedome blocks to connect to multiple Datacenter freedoms in a campus on the south & WAN connectivity layer on the north

- Independent Blocks = Crucial for maintaining performance on Freedomes

Firepath (Custom Interior Gateway Protocol)

- New routing mechanism for
  - Firehose
  - Watchover
  - Saturn
  - (Jupiter???)

- Why building a new one? (Reasons a decade ago)
  - Existing routing protocols have poor support for multipath, equal-cost forwarding
  - No high quality open source routing stacks
  - Overhead of running broadcast-based routing protocols is high (scalability issue)
  - Network manageability (configuration)

Neighbor Discovery (ND)

- An online liveness and peer correctness protocol

- Peers exchange their local port IDs with each other to
  - Compare with their expected peer port IDs to verify correctness
  - Serve as keep alive messages

Firepath (Cont'd)

- Firepath Client (Fabric switches)
  - Load static topology
  - Collect local interface states
  - Transmit state info to Firepath Master
- Firepath Master (Spine switches)
  - Construct Link State Database (LSD)
  - Distribute LSD to clients

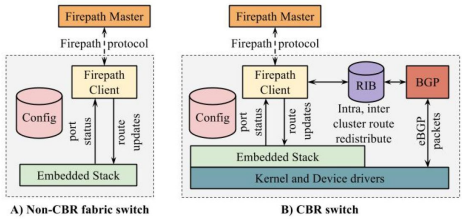


Figure 17: Firepath component interactions.

Firepath Master Redundancy Protocol

- Redundant master instances are on pre-selected spine switches.
- This info is stored in their static configuration.
- Backup masters is in sync (LSD) with active master
- The master with latest LSD becomes the next active master when the original active master is down

Cluster Border Router

- Integration of BGP and Firepath in CBRs
- Firepath manages only one single path for all outbound traffic (CBR prefix)

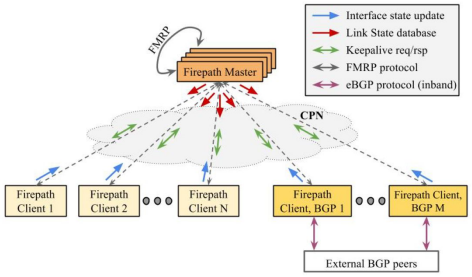


Figure 18: Protocol messages between Firepath client and Firepath master, between Firepath masters and between CBR and external BGP speakers.

Fabric Management

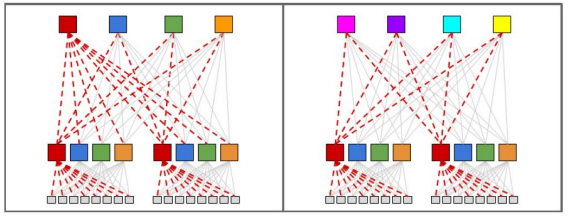


Figure 20: Multi-color fabric chassis upgrade.

Fabric Congestion

- Causes
  - Inherent burstiness of flows
  - Limited buffering
  - Oversubscription
  - Imperfect flow hashing (load balancing)
- Solution (For 25% average utilization from 1% to < 0.01%)
  - Dropping packets based on QoS
  - Tuning hosts' TCP congestion window
  - Link-level paused (only in early fabrics)
  - Using Explicit Congestion Notification (?)
  - Monitoring application BW requirements to deploy links or repopulate links
  - Tuning buffer sharing in merchant silicon
  - Tuning flow hashing functions

Outages Do Happen

- Control software
  - ND and route computing contend for CPU resources
  - Snowball effect: fabric reboot => routing churn => ND not responding fast enough
- Aging hardware expose unhandled failure mode
  - Redundant standby links are not monitored
- Misconfiguration
  - Read/write BGP configuration at the same time

Conclusion and Discussion

- Some take away points
  - Centralized control vs. distributed control
  - Clos network
  - The concept of updating while running
- Q&A?
  - How to design system parameters?
  - How to do some simulation before deployment