Cloud Computing

ECPE 276

Google

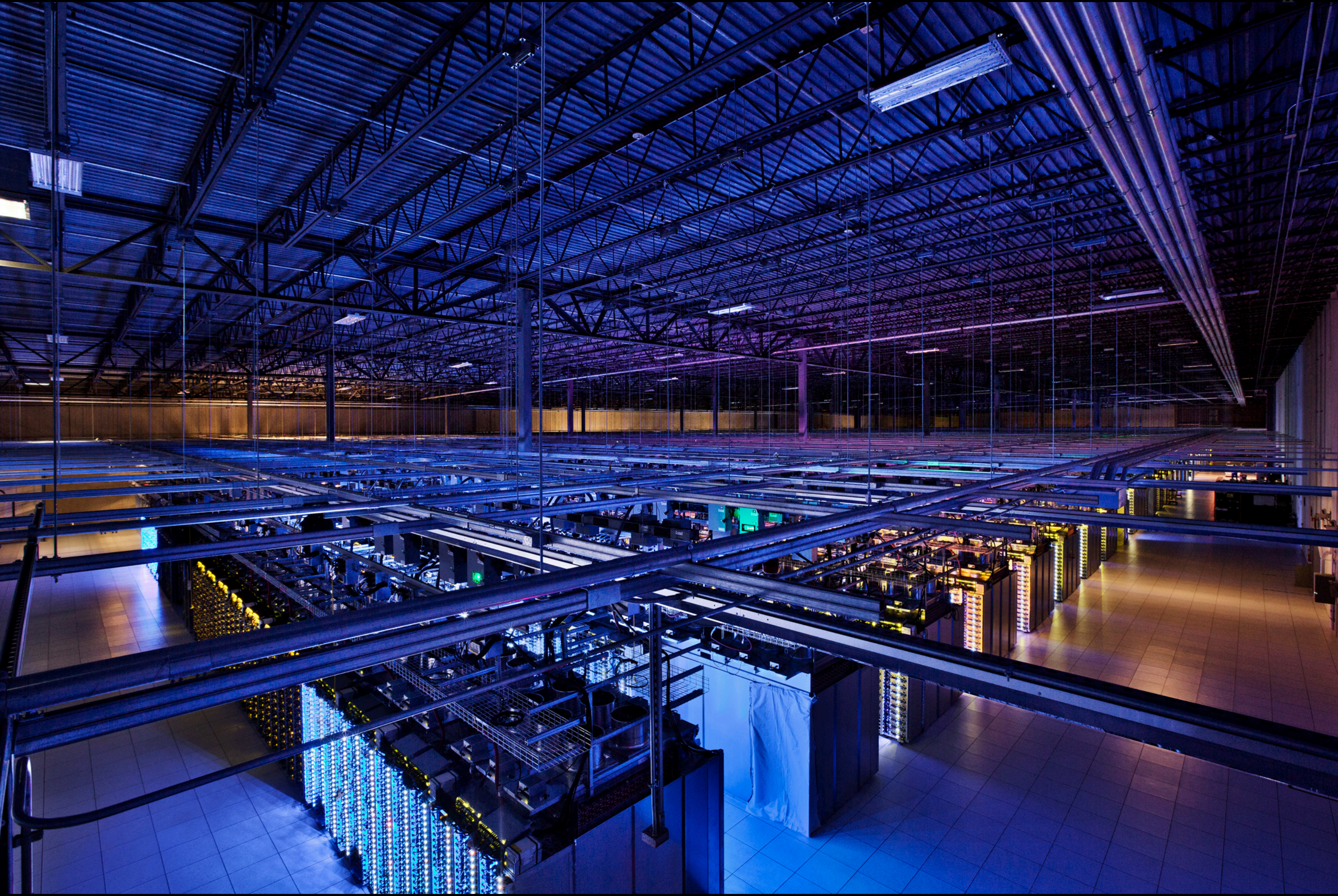
Datacenter Network

Arjun Singh et. al, "**Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network**", in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication* (SIGCOMM'2015), 2015

Inside a Datacenter

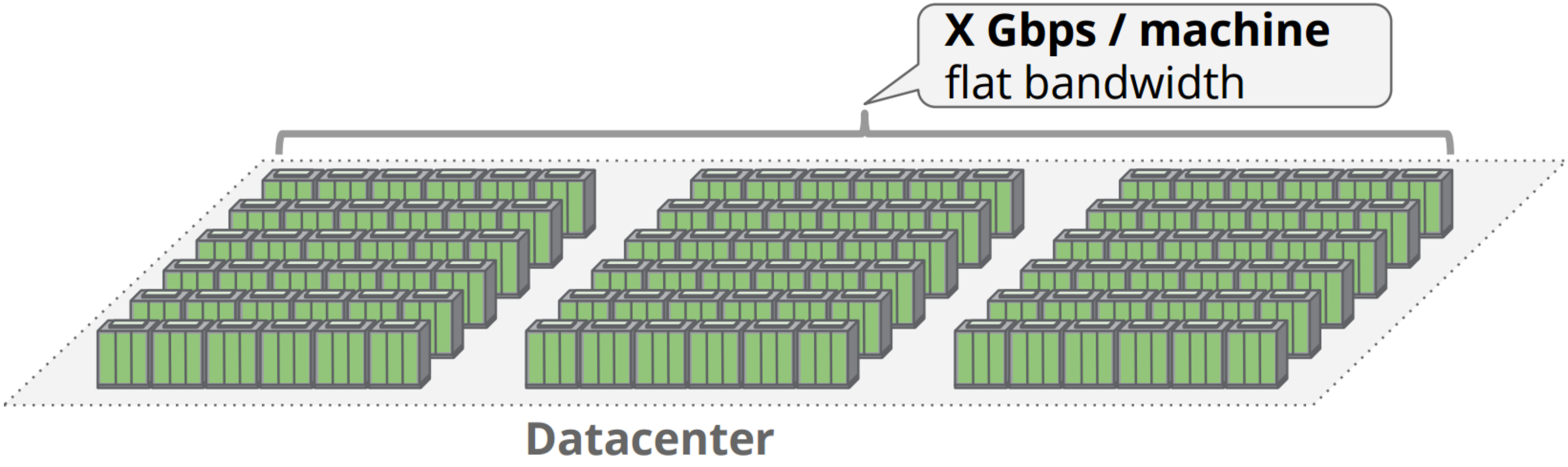
- 10s or 100s of thousands of servers
- Petabytes of data storage
- Single "applications" spread across many thousands of servers (e.g., Amazon.com)
  - Application components such as caches, web servers, databases, distributed file servers, …
  - Each component is "scaled" to meet needs of millions of users

George Porter, UC San Diego
SIGCOMM 2015 Preview Session
http://yuba.stanford.edu/~huangty/sigcomm15_preview/Sigcomm15_DC_Preview.pdf



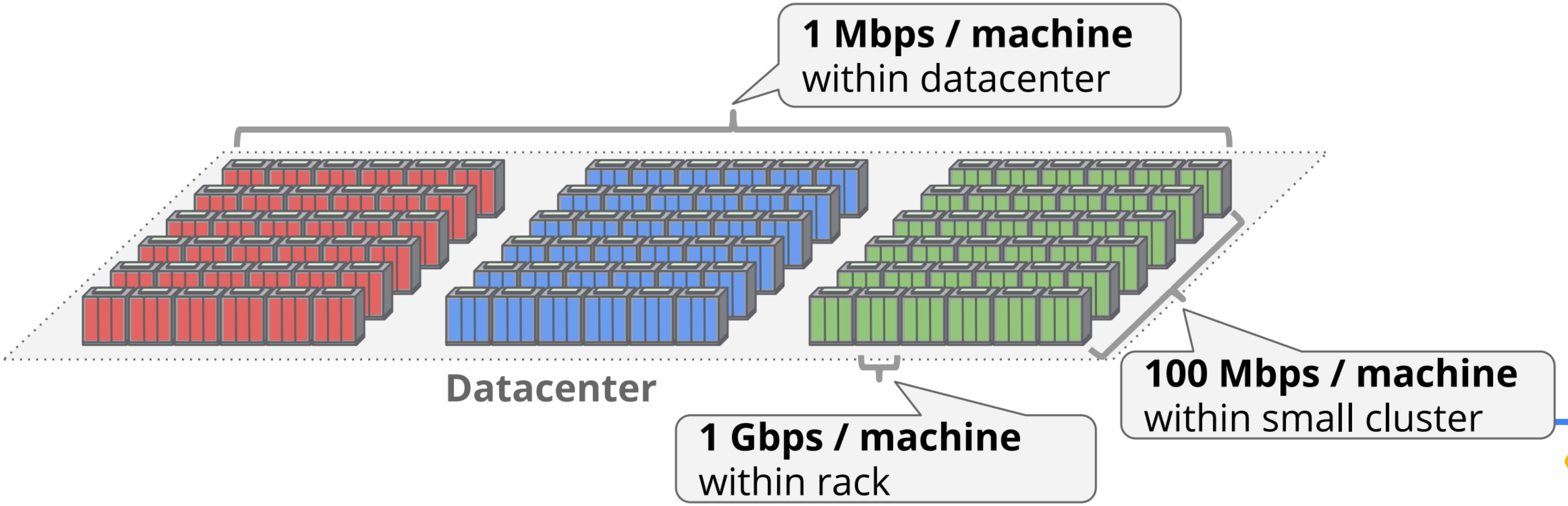
The Dream

- Equivalent ("flat") bandwidth between any two servers in the **building-scale** network
  - Simplifies scheduling (no locality worries!)
  - No *resource stranding* in different clusters
  - Allows application scaling



Reality (10 years ago)

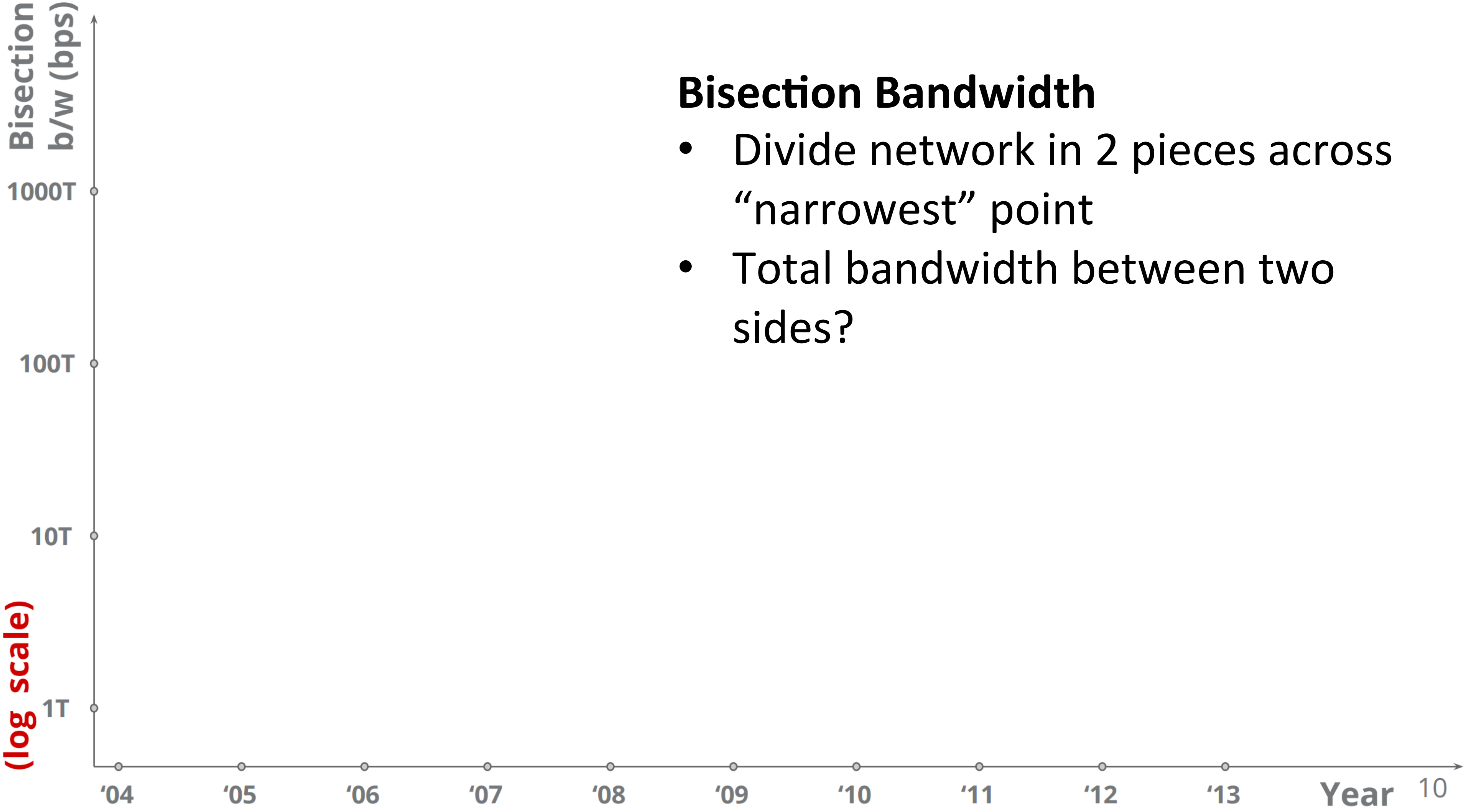
- ***Islands of bandwidth***
  - Tradeoffs in balancing placing data close together (for high bandwidth) vs correlated network failures



1 Mbps / machine
within datacenter

100 Mbps / machine
within small cluster

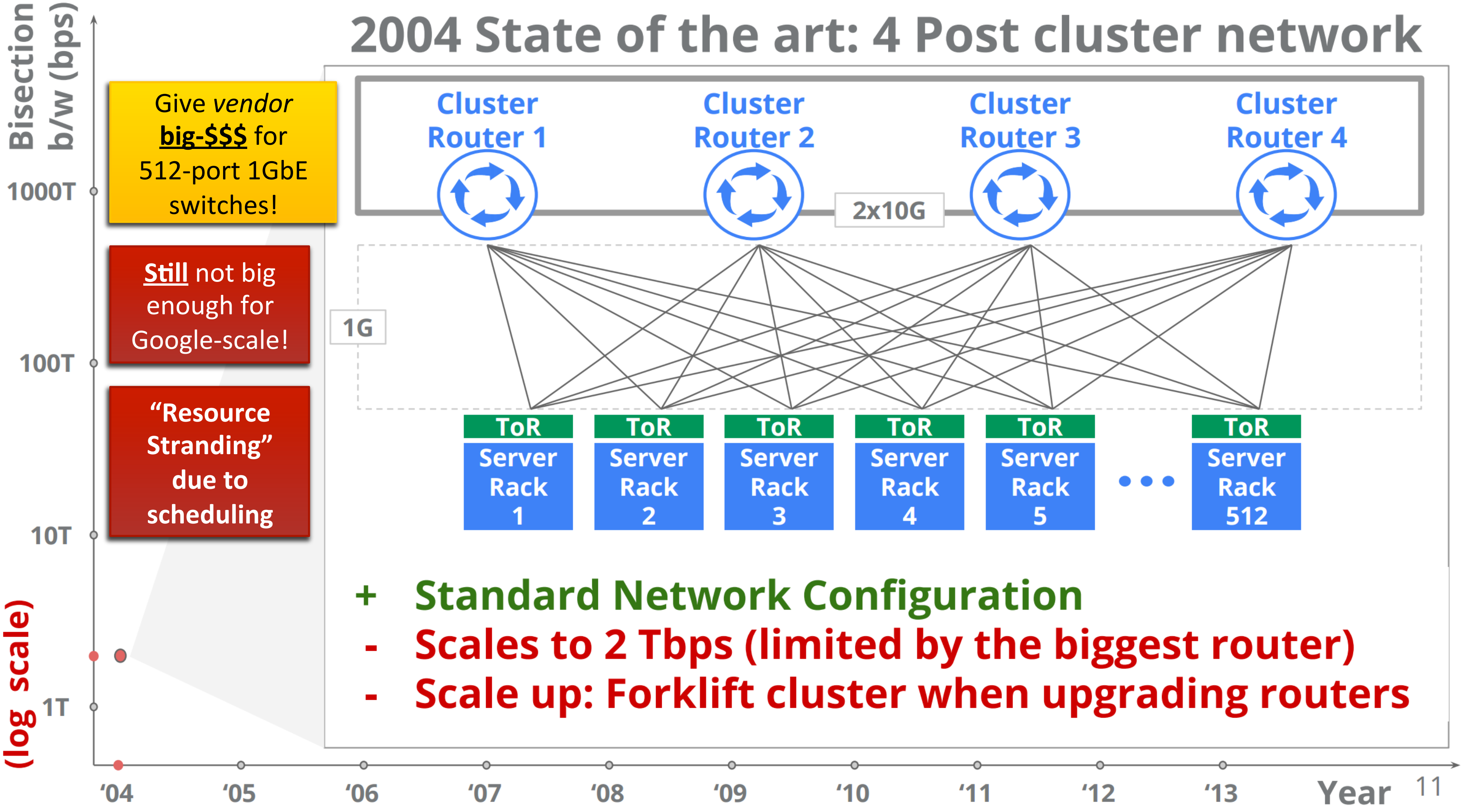
1 Gbps / machine
within rack



Bisection Bandwidth

- Divide network in 2 pieces across "narrowest" point
- Total bandwidth between two sides?

2004 State of the art: 4 Post cluster network



Give *vendor* <u>**big-$$$**</u> for 512-port 1GbE switches!

<u>**Still**</u> not big enough for Google-scale!

"Resource Stranding" due to scheduling

- **+ Standard Network Configuration**
- **- Scales to 2 Tbps (limited by the biggest router)**
- **- Scale up: Forklift cluster when upgrading routers**

Opportunity!

- Datacenter network (DCN) is **not** like the public Internet

The Internet	Data Center Network

George Porter, UC San Diego
SIGCOMM 2015 Preview Session
http://yuba.stanford.edu/~huangty/sigcomm15_preview/Sigcomm15_DC_Preview.pdf

Opportunity

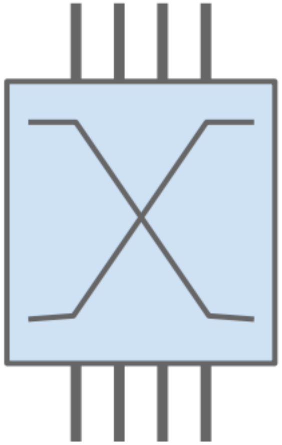
- How would you design a network to support 1M endpoints?
- If you could…
  - Control all the endpoints and the network?
  - Violate layering, end-to-end principle?
  - Build custom hardware?
  - Assume common OS, dataplane functions?

Top-to-bottom rethinking of the network

George Porter, UC San Diego
SIGCOMM 2015 Preview Session
http://yuba.stanford.edu/~huangty/sigcomm15_preview/Sigcomm15_DC_Preview.pdf

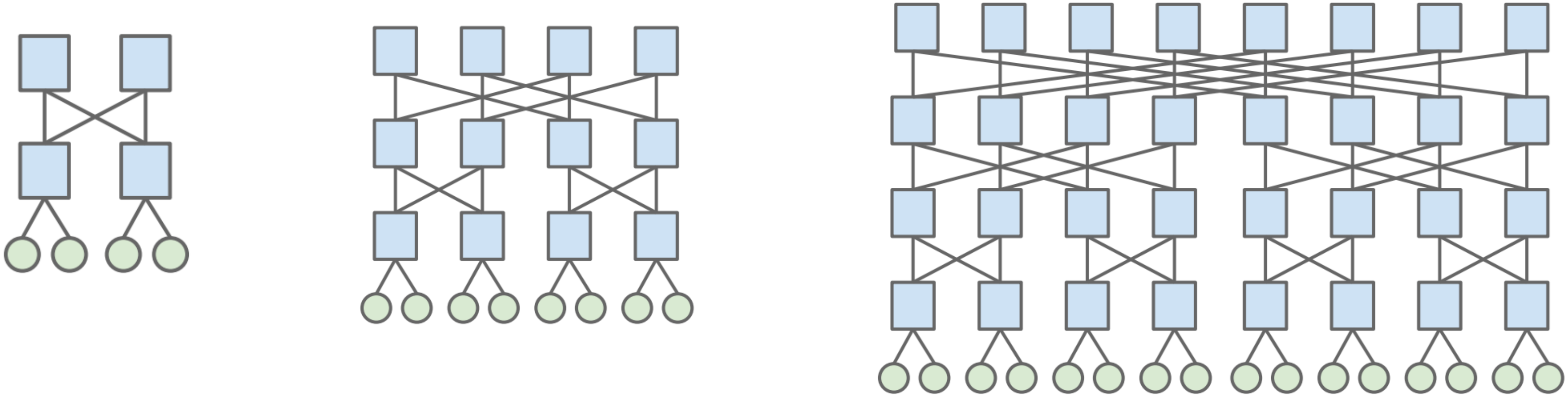
Solutions

- Commodity silicon
  - Off-the-shelf, cheap, switching devices
  - Upgrade frequently



Solutions

- Multi-Stage *Clos* topologies
  - Assemble many low-radix switches to arbitrary scale
  - Non-blocking
  - "**Scale-Out**" w/ cheap components *(cloud approach)* vs "Scale-Up" w/pricy components



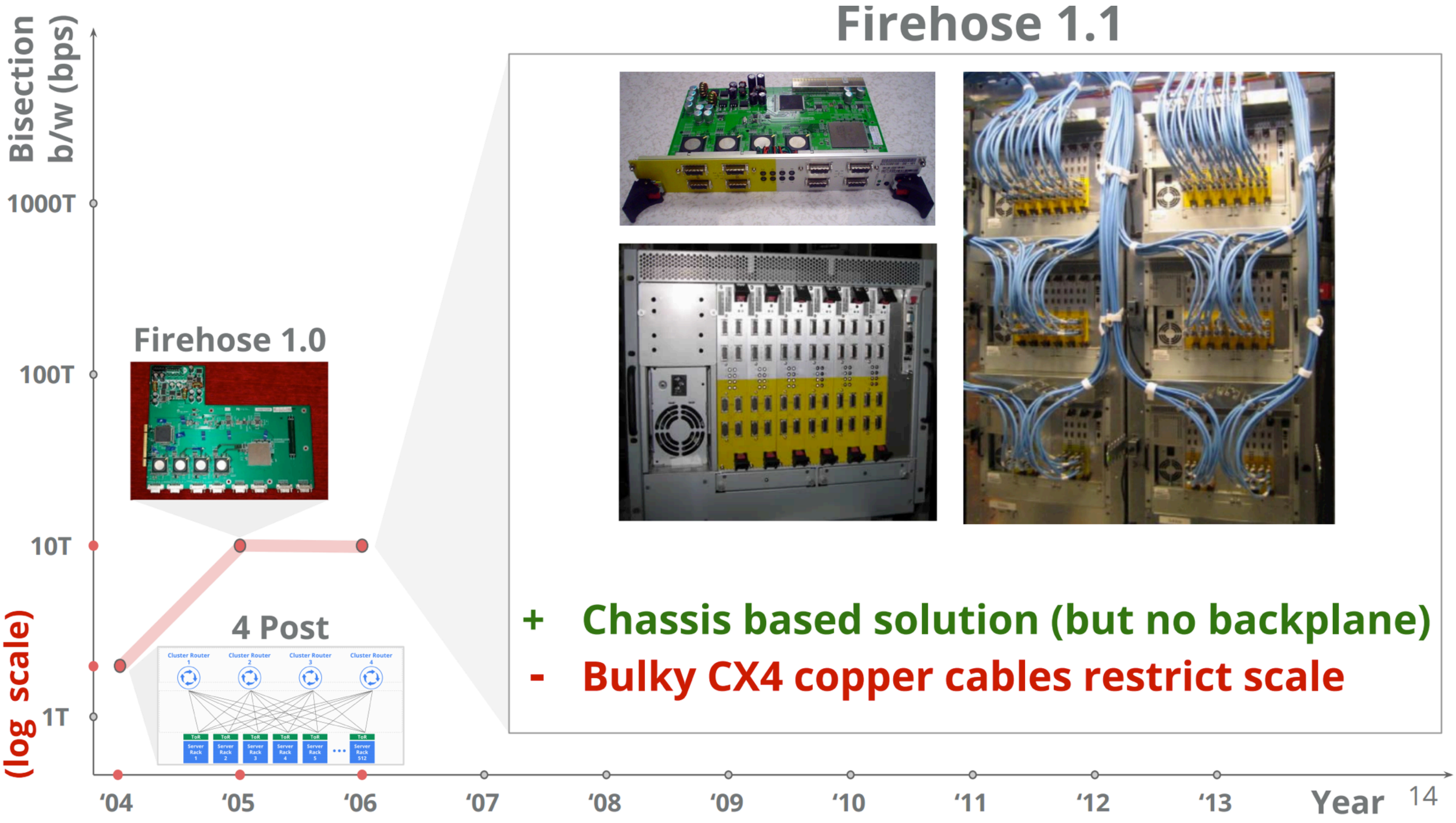
Solutions

- Centralized network control
- Observation – Physical network topology is fixed (aside from link/switch failures)
  - No need to "discover" new links!
  - Administrator will notify controller about (rare) new links
- Collect and distribute link-state information from *one* (dynamically determined) node in network
  - Individual switches calculate their own forwarding tables based on these *exceptions* to the underlying (normal) network topology

Multiple Generations of DCNs

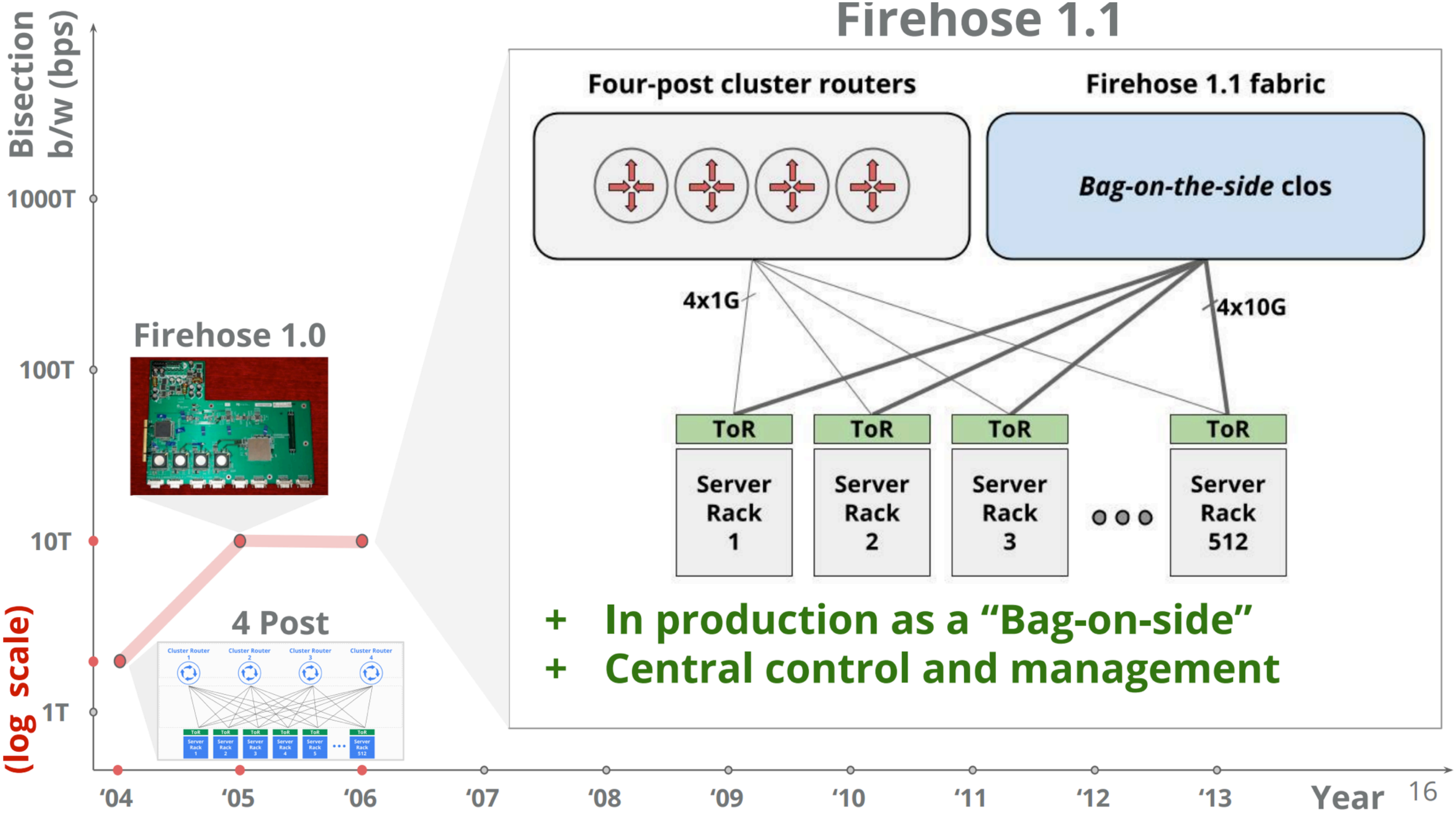
- **Firehose 1.0** (2004)
  - Server-based hardware (PCI boards)
  - Experimental – never deployed
- **Firehose 1.1** (2005)
  - New hardware platform
  - Small-scale deployment
- **Watchtower** (2008)
  - Global deployment
- **Saturn** (2009)
- **Jupiter** (2012)

Firehose 1.1

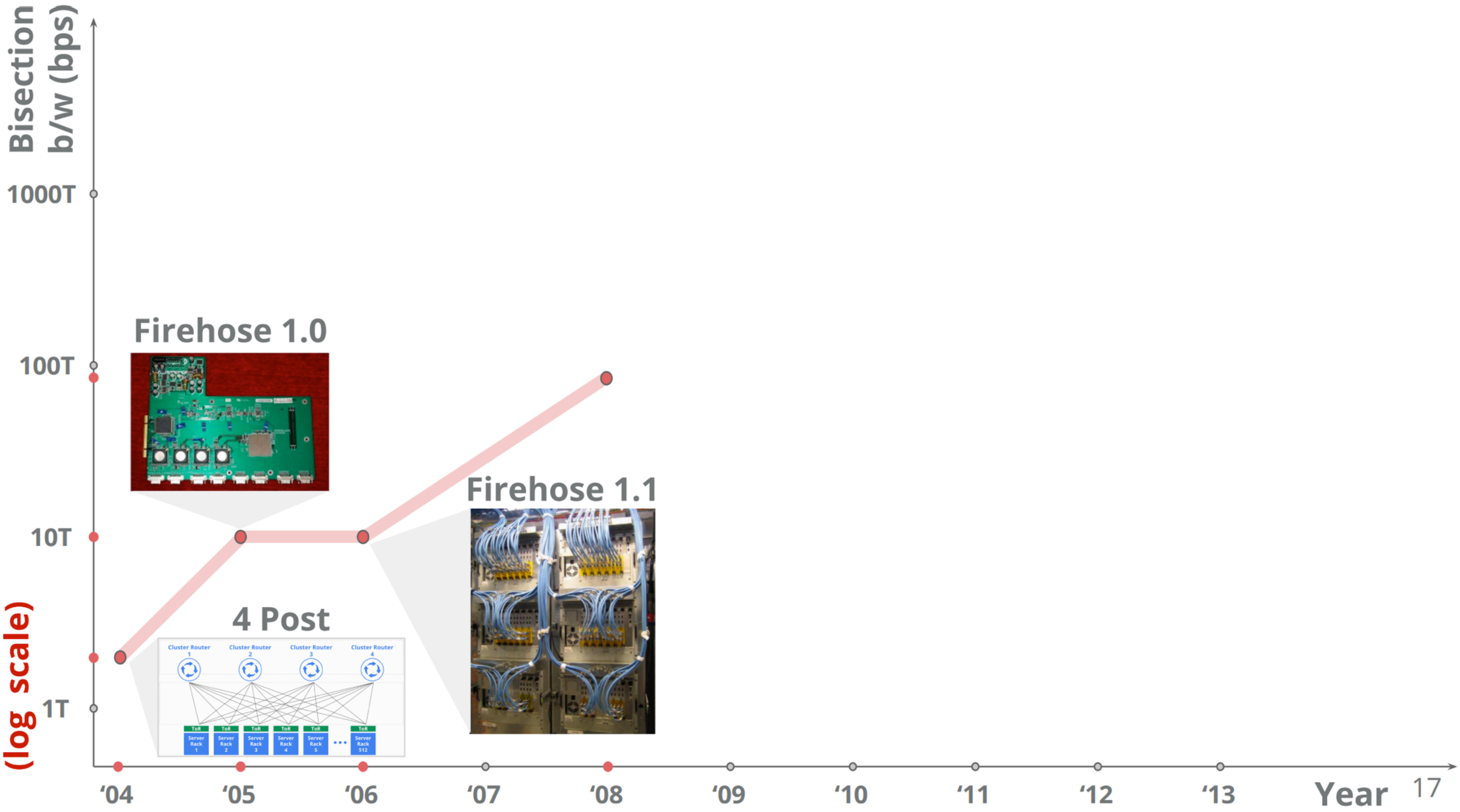


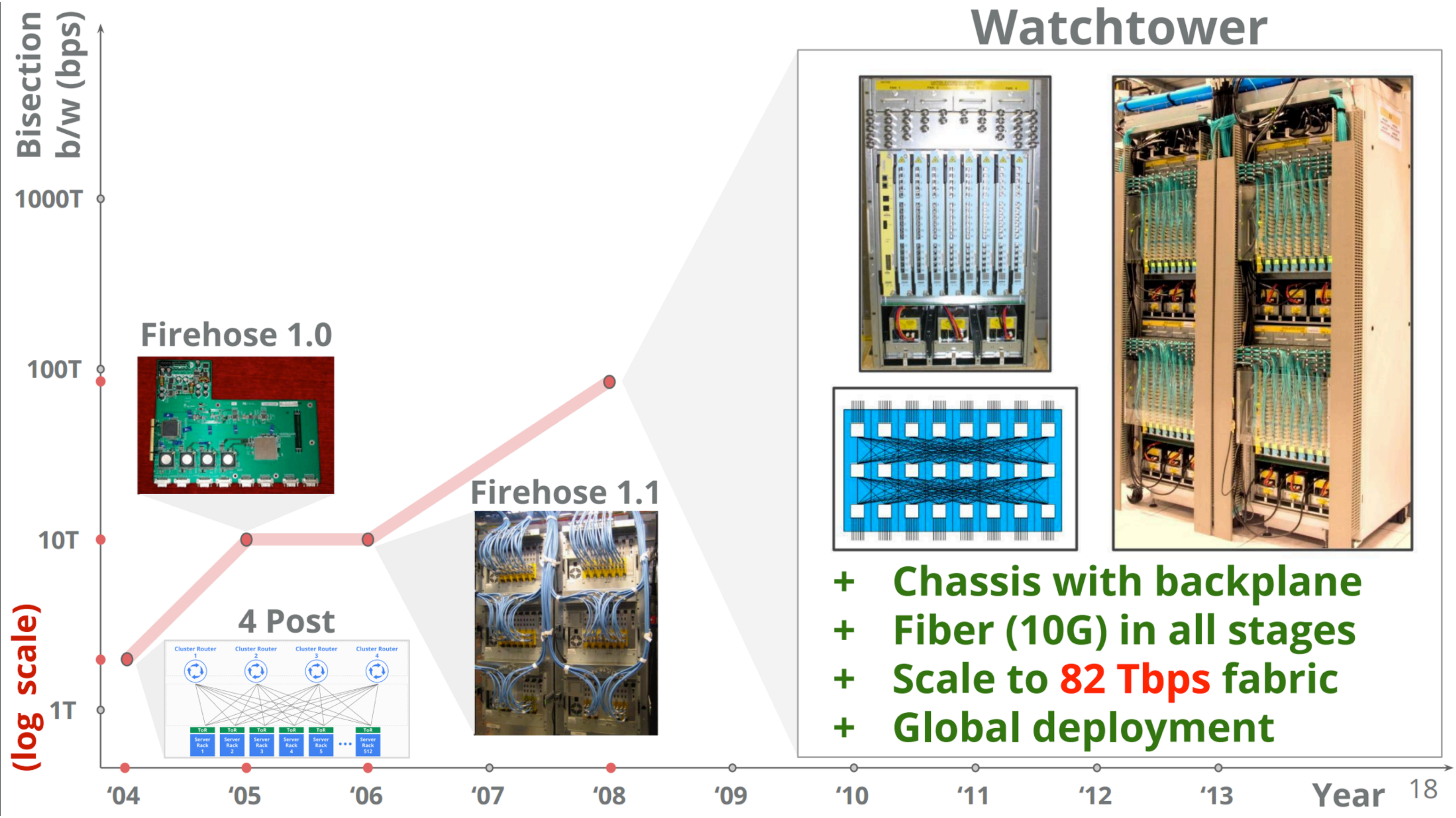
- + Chassis based solution (but no backplane)
- - Bulky CX4 copper cables restrict scale

Firehose 1.1

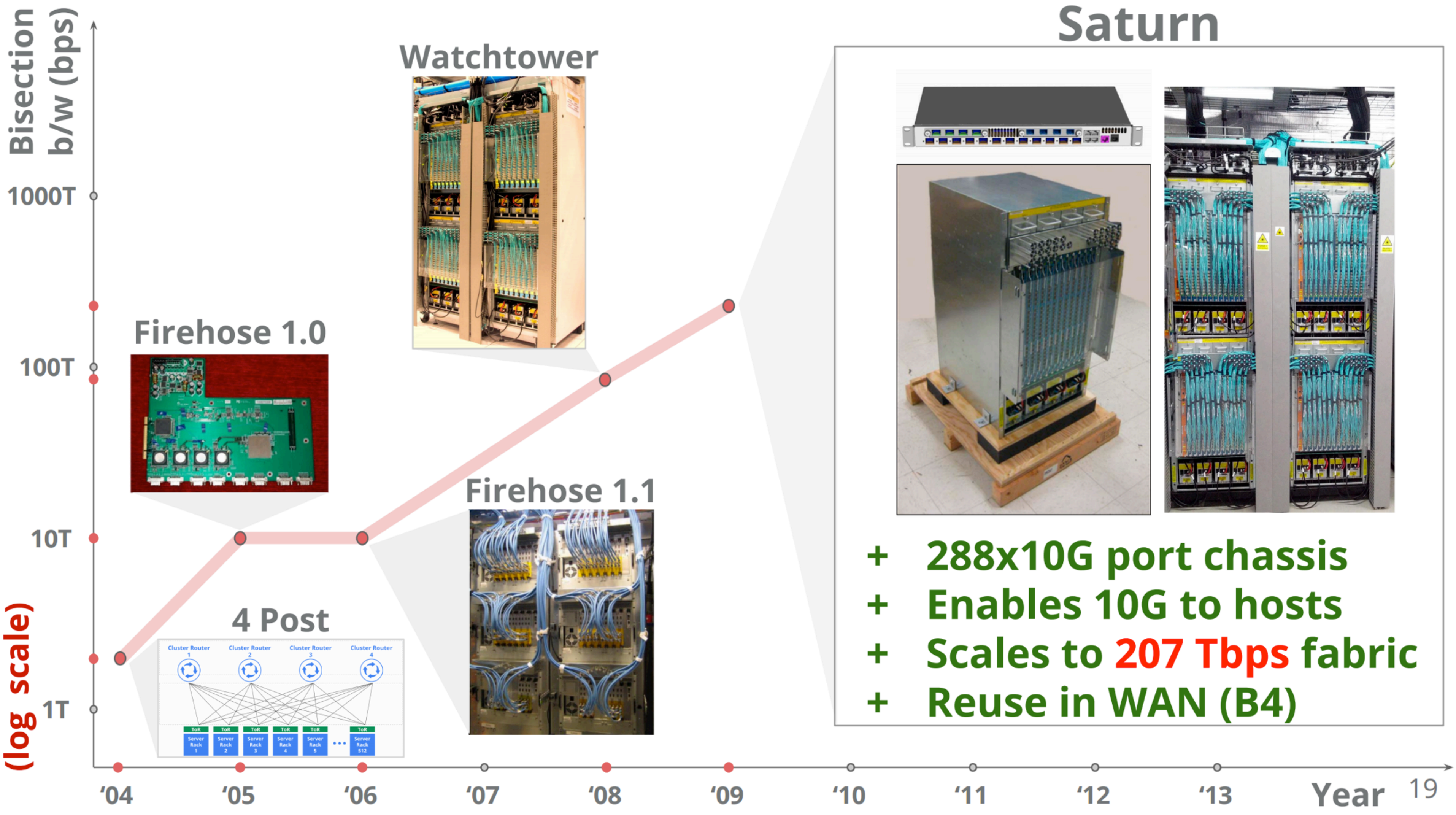


- + In production as a "Bag-on-side"
- + Central control and management

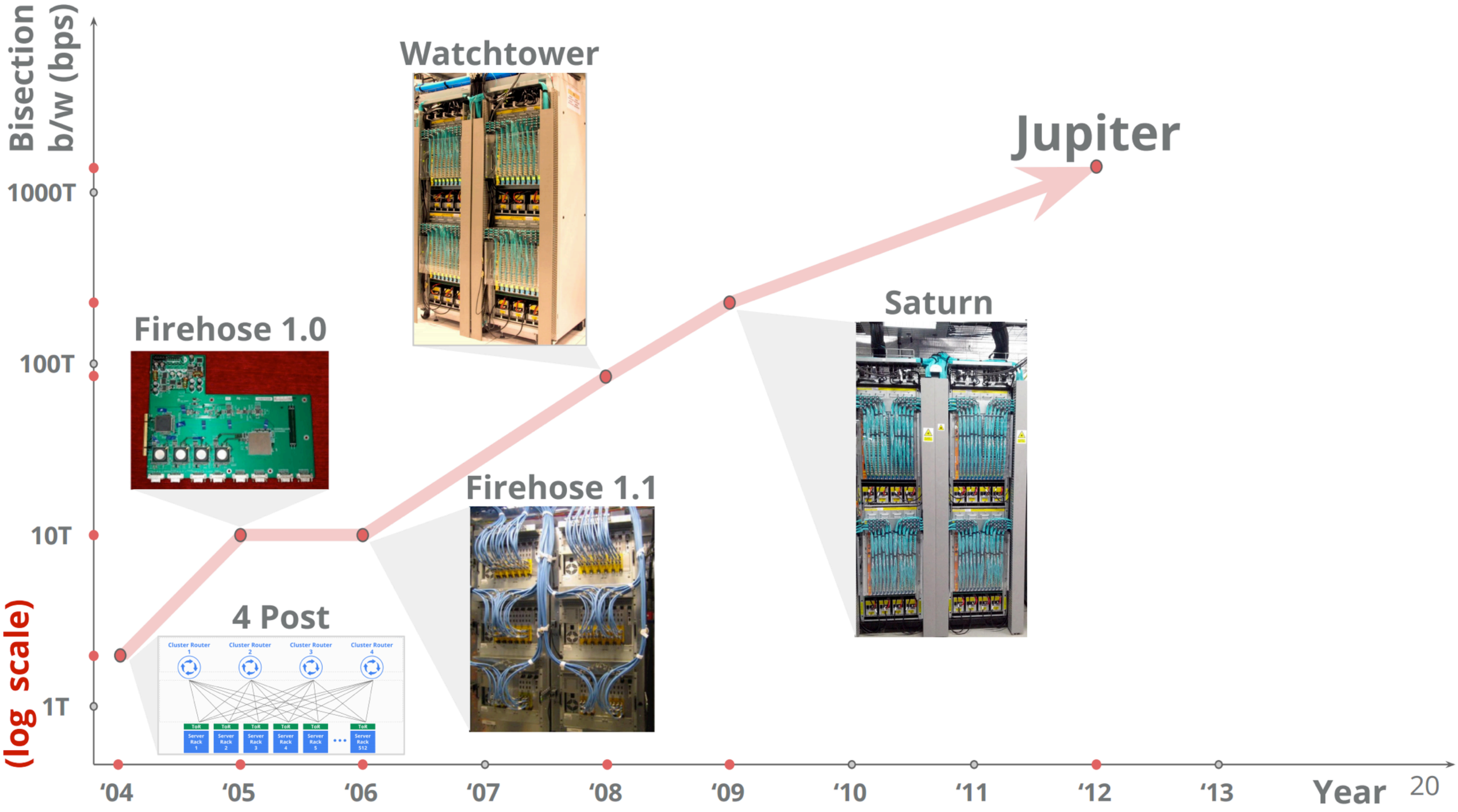

Bisection b/w (bps)
(log scale)
1000T
100T
10T
1T
Firehose 1.0
Firehose 1.1
4 Post
Cluster Router 1
Cluster Router 2
Cluster Router 3
Cluster Router 4
ToR
Server Rack 1
Server Rack 2
Server Rack 3
Server Rack 4
Server Rack 5
...
Server Rack 512
'04
'05
'06
'07
'08
'09
'10
'11
'12
'13
Year

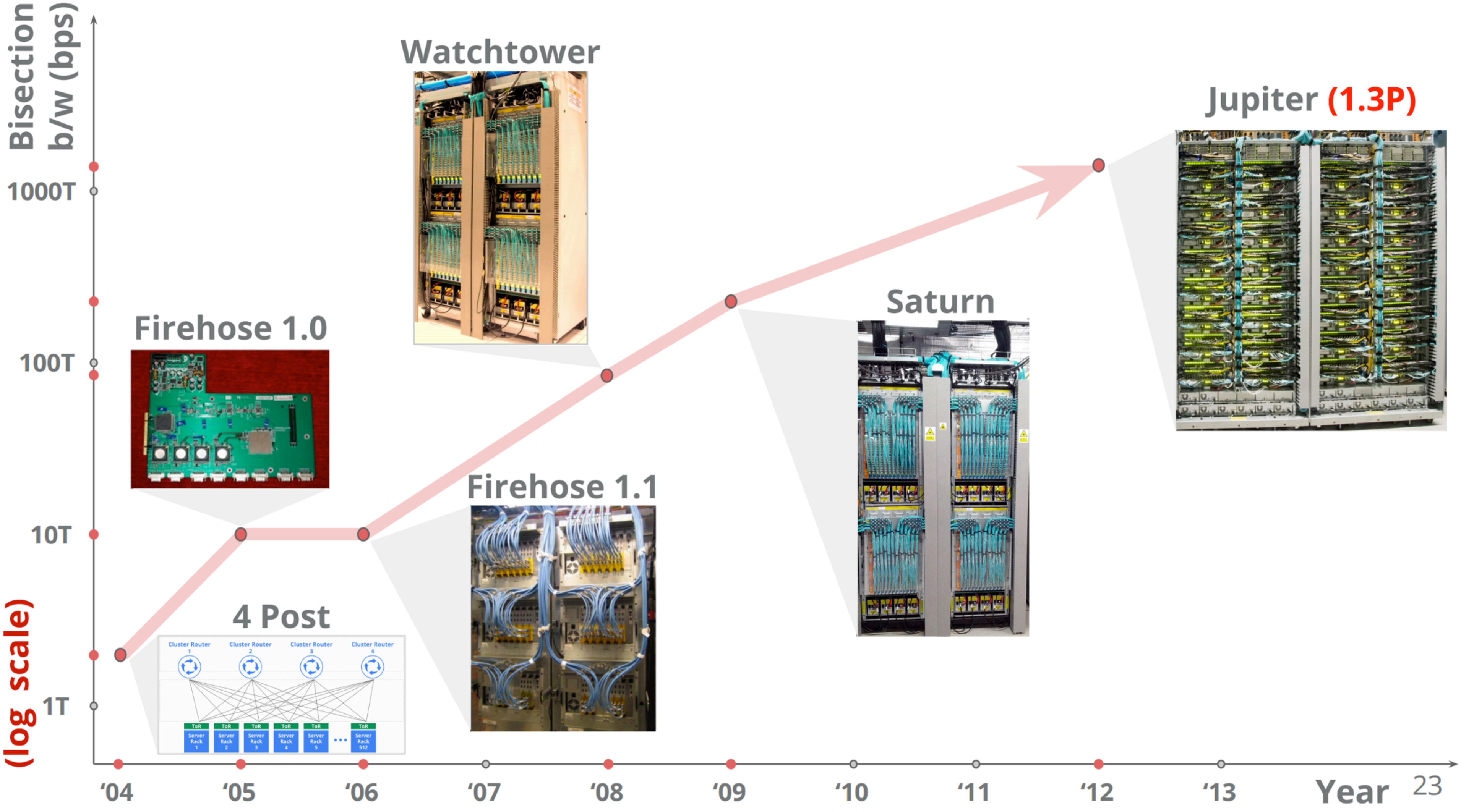


- + Chassis with backplane
- + Fiber (10G) in all stages
- + Scale to 82 Tbps fabric
- + Global deployment



- + 288x10G port chassis
- + Enables 10G to hosts
- + Scales to 207 Tbps fabric
- + Reuse in WAN (B4)


Bisection b/w (bps)
(log scale)
1000T
100T
10T
1T
Watchtower
Jupiter
Firehose 1.0
Saturn
Firehose 1.1
4 Post
Cluster Router 1
Cluster Router 2
Cluster Router 3
Cluster Router 4
ToR
Server Rack 1
Server Rack 2
Server Rack 3
Server Rack 4
Server Rack 5
Server Rack 512
'04
'05
'06
'07
'08
'09
'10
'11
'12
'13
Year


Bisection b/w (bps)
(log scale)
1000T
100T
10T
1T
'04
'05
'06
'07
'08
'09
'10
'11
'12
'13
Year
23
4 Post
Cluster Router 1
Cluster Router 2
Cluster Router 3
Cluster Router 4
ToR
Server Rack 1
Server Rack 2
Server Rack 3
Server Rack 4
Server Rack 5
...
Server Rack 512
Firehose 1.0
Firehose 1.1
Watchtower
Saturn
Jupiter (1.3P)

Jupiter racks

- + Enables 40G to hosts
- + External control servers
- + OpenFlow

Google

Jupiter Results

- Network bisection bandwidth grows 3 orders of magnitude (Tbps to Pbps) in 10 years
- **100,000** servers can communicate with one another in an arbitrary pattern at **10Gb**/s

Software-Defined Networking (SDN)

- Existing control protocols (10+ years ago)
  - OSPF, ISIS, BGP, etc;
  - Box-centric configuration/management
- DCN required new central control/management system
  - Limited support for multipath forwarding *(at the time)*
  - No robust open source stacks *(at the time)*
  - Broadcast protocol scalability a concern *at scale*
  - Network manageability painful with individual switch configuration
    - Goal: Same configuration for all switches

Observations

- Must be able to incrementally upgrade network
  - Datacenter-scale facilities are too expense to sit idle during "scorched earth" updates
- Logically centralized control plane with peer-to-peer data plane beats full decentralization
  - Significantly simplifies system design
- Scale out >> scale up
- Small on-chip buffers in commodity hardware can be alleviated in software
  - ECN – Explicit congestion notification (switches)
  - DCTCP – Linux network stack that reacts to ECN
  - *Only works because entire system (HW+SW) is controlled!*

Discussion Questions

- What worked and what didn’t work for Google?
- Who else can use this technology besides Google?
- Where are we going in 5 years?
- Strengths and weaknesses of the paper?

References

- ONS 2015 Keynote w/ Amin Vahdat https://www.youtube.com/watch?v=FaAZAII2x0w
- http://googleresearch.blogspot.com/2015/08/pulling-back-curtain-on-googles-network.html
- http://conferences.sigcomm.org/sigcomm/2015/pdf/papers/p183.pdf