Chapter 1: Introduction

Much in our life is now on the web

Google

amazon.com



NETFLIX

Microsoft® Office 365

Spotify

Google docs

bing

salesforce



facebook

The web is driven by data



+ amazon.com = Product Recommendations



+ Spotify® = Custom Stations



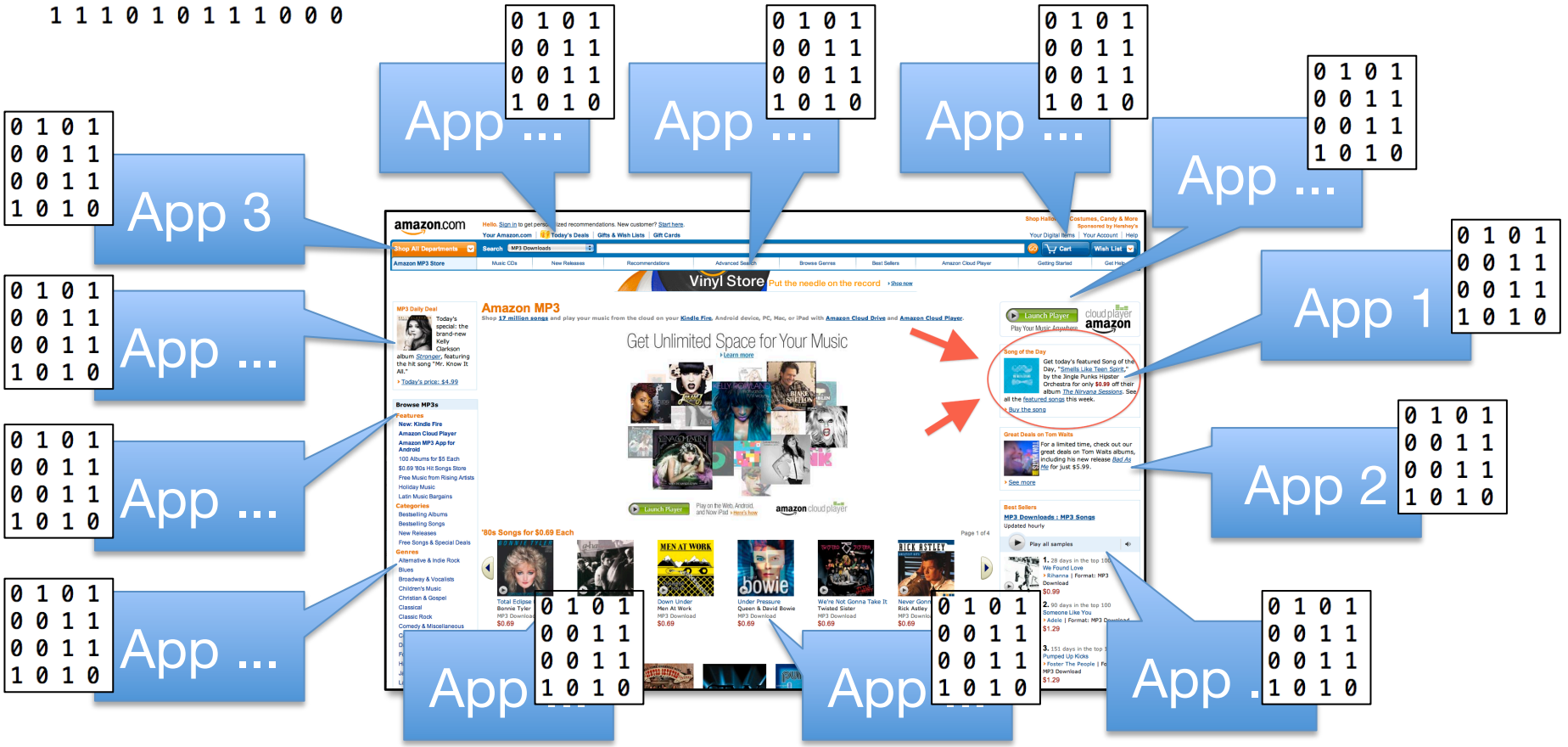
+ Google = Personalized Search

Data-driven, per-user customization



+ amazon.com =

Product Recommendations



Cloud Computing



- Elastic resources
  - Expand and contract resources
  - Pay-per-use
  - Infrastructure on demand
- Multi-tenancy
  - Multiple independent users
  - Security and resource isolation
  - Amortize the cost of the (shared) infrastructure
- Flexible service management
  - Resiliency: isolate failure of servers and storage
  - Workload movement: move work to other locations

Cloud Service Models

- Software as a Service (Saas)
  - Provider licenses applications to users as a service
  - e.g., customer relationship management, email, …
  - Avoid costs of installation, maintenance, patches, …
- Platform as a Service (Paas)
  - Provider offers software platform for building applications
  - e.g., Google's App-Engine
  - Avoid worrying about scalability of platform
- Infrastructure as a Service (Iaas)
  - Provider offers raw computing, storage, and network
  - e.g., Amazon's Elastic Computing Cloud (EC2)
  - Avoid buying servers and estimating resource needs

Data centers with 100,000+ servers

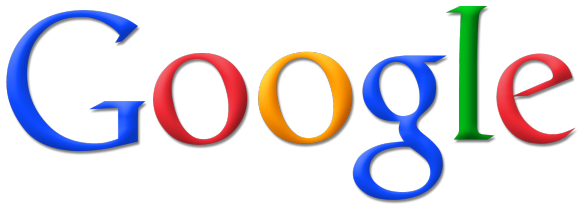








These things are *really* big



100 billion searches per month



1.15 billion users

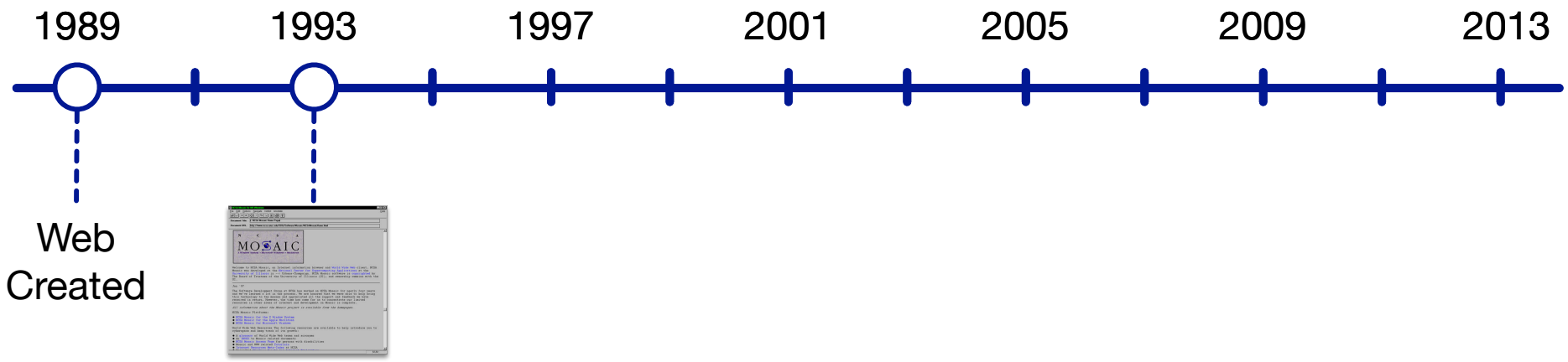


120+ million users

The need for rapid growth



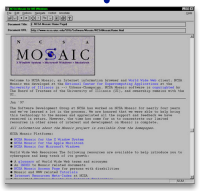
The need for rapid growth



The need for rapid growth

1989 1993 1997 2001 2005 2009 2013

Web Created



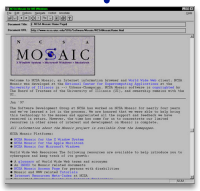


Google's 1st cluster (15 years)

The need for rapid growth

1989 1993 1997 2001 2005 2009 2013

Web Created



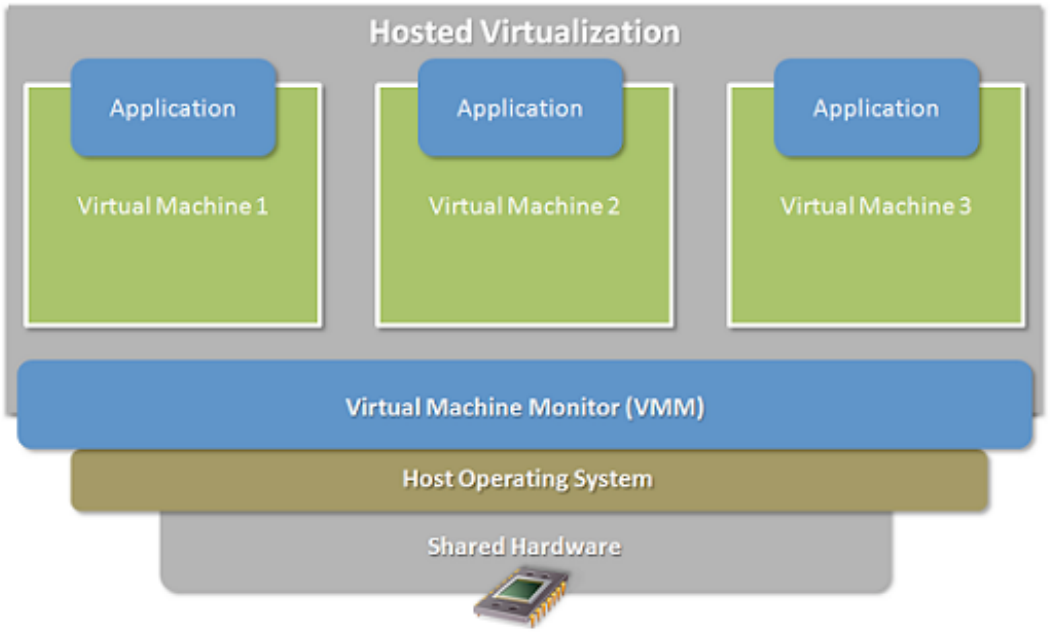


Google's 1st cluster (15 years)

facebook

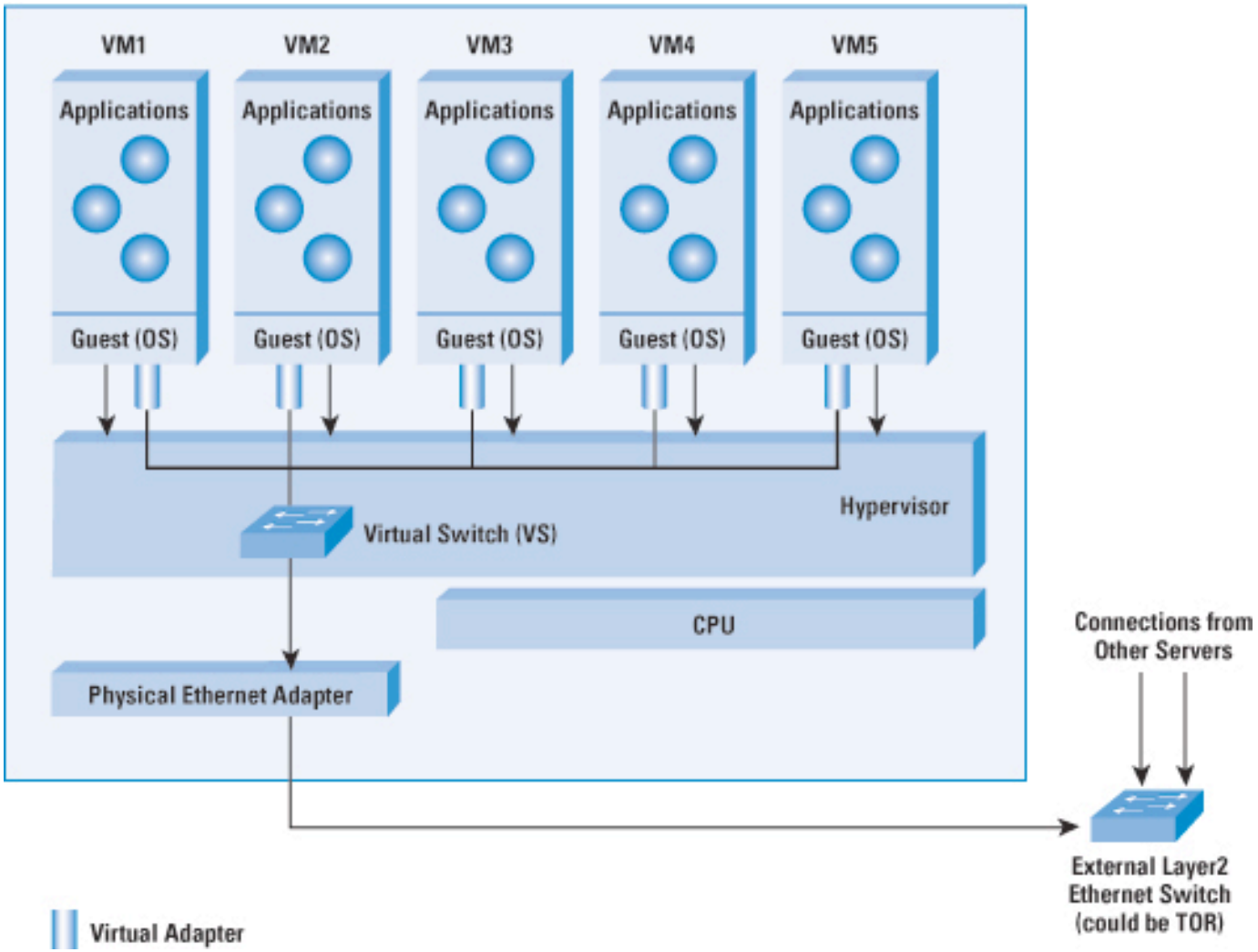
(10 years)

Host Virtualization

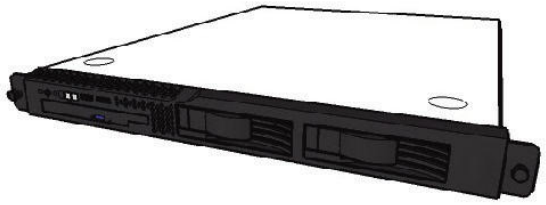


- Multiple virtual machines on one physical machine
- Applications run unmodified as on real machine
- VM can migrate from one computer to another

VMM Virtual Switches



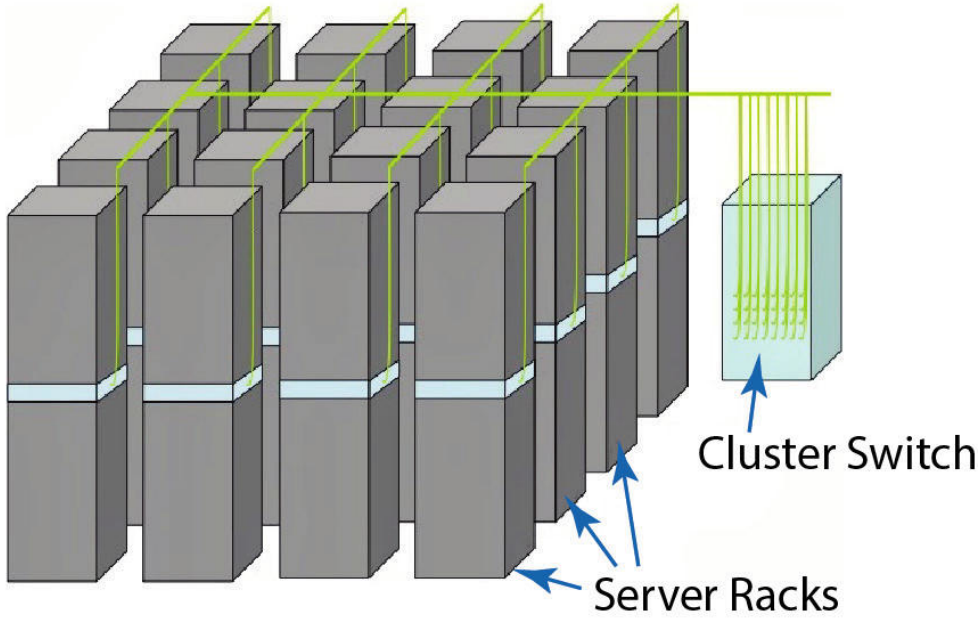
Building blocks of modern data centers



Network switch

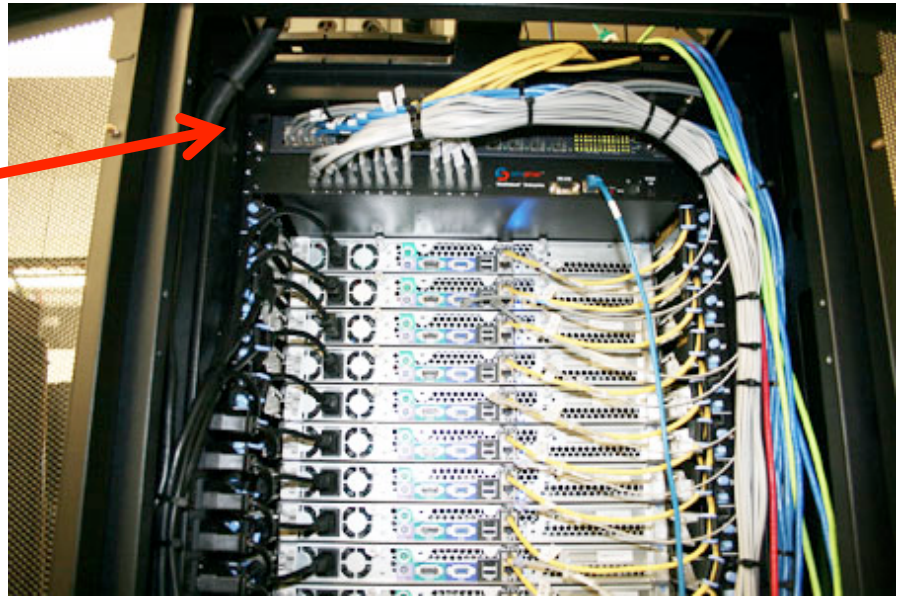


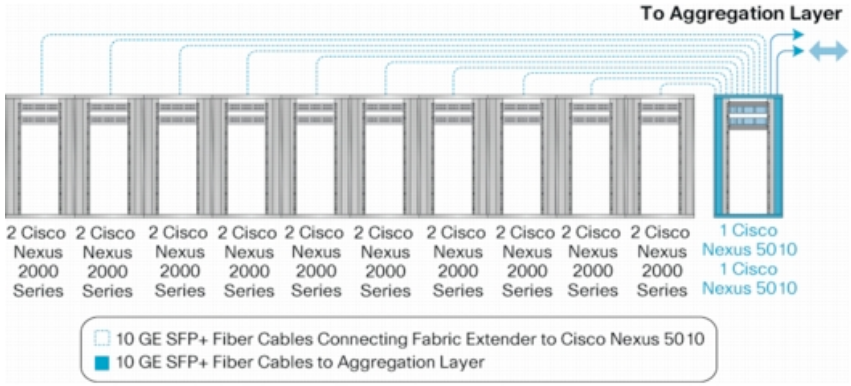
Rack



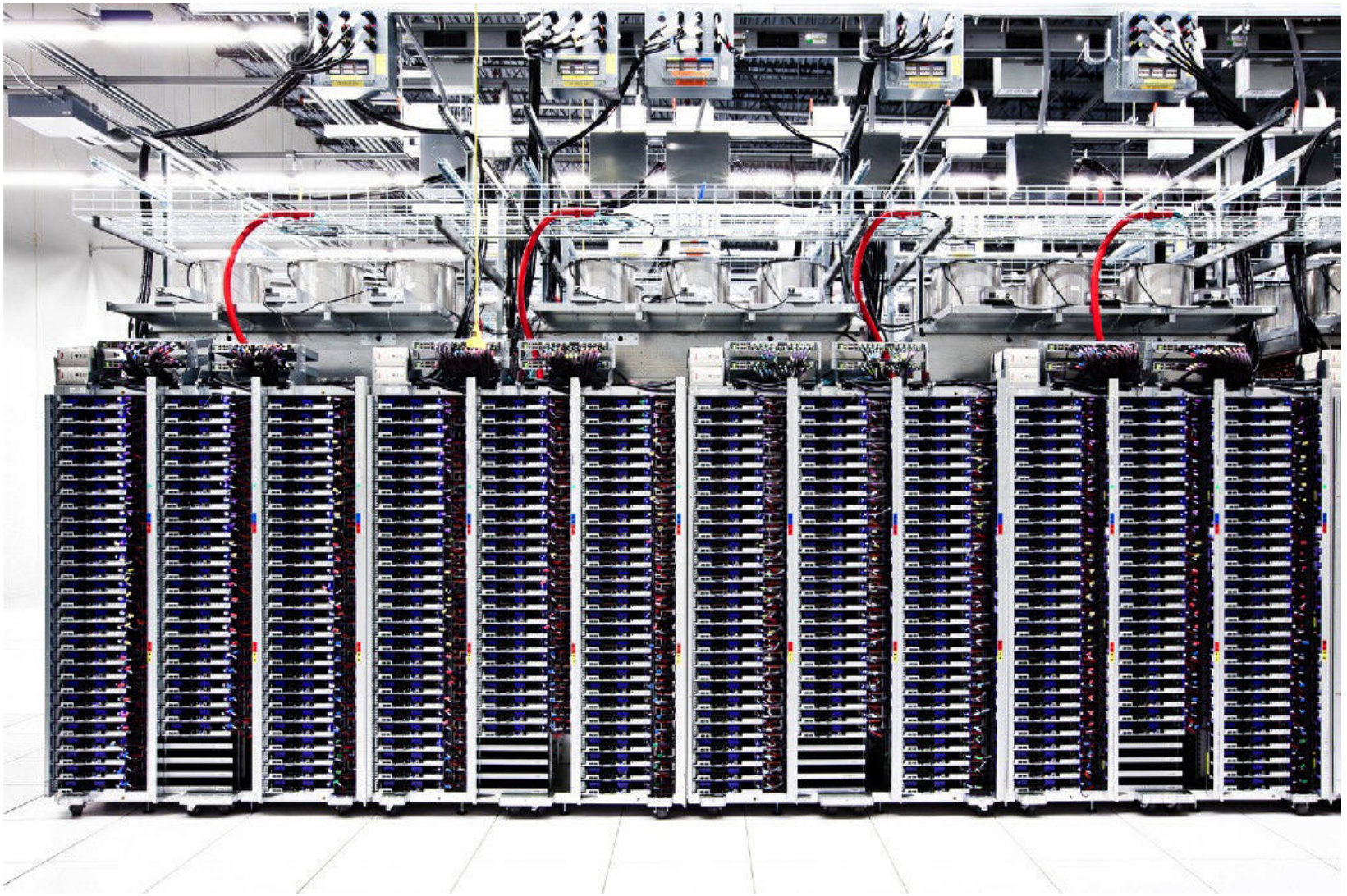
Top-of-Rack Architecture

- Rack of servers
  - Commodity servers
  - And top-of-rack switch
- Modular design
  - Preconfigured racks
  - Power, network, and storage cabling
- Aggregate to the next level





Racks of servers (Google)



Facebook



Google



Extreme Modularity

- Containers



- Many containers



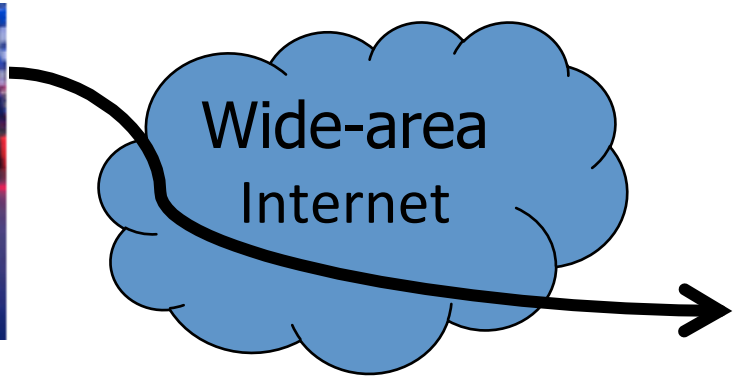
Not just a collection of servers

- A data center isn't just a "small internet"
- Why?
  - Administered as a single domain
  - Trusted administrators
  - No need to be compatible with the "outside world"
    - Except for traffic to/from users
  - No need for international standards bodies
    - Though why do standards help?

“Front-End” datacenter traffic



Data center

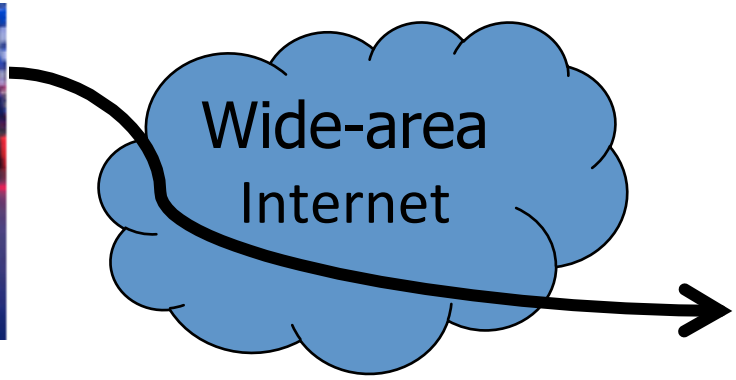




Internet
Users

“Front-End” datacenter traffic



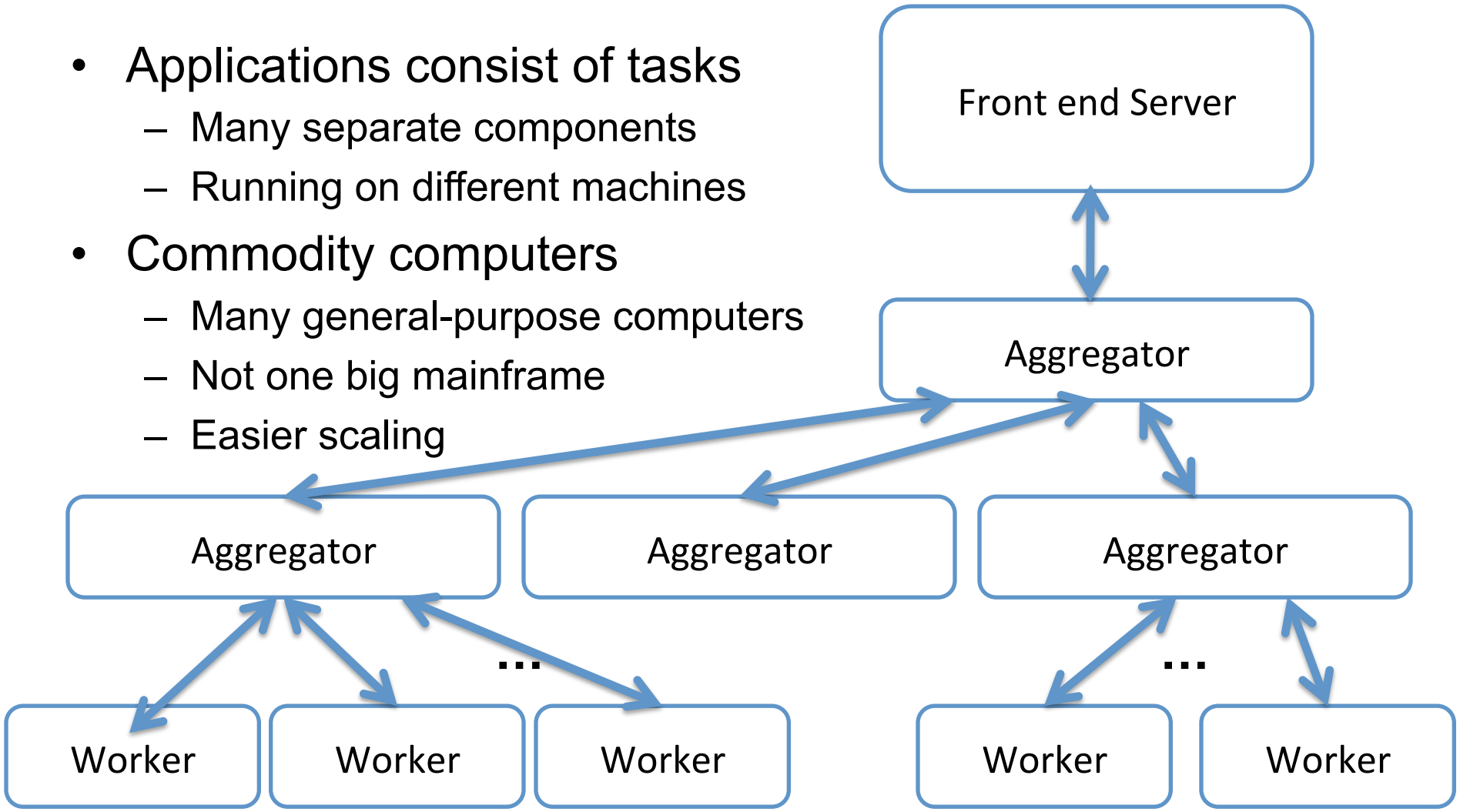




- Data sizes driven by the content that users actually consume
  - Growth largely due to higher bitrate content (IP TV/movies, iPhone Facetime)
- Mobile Internet source of new users
- Often constrained by the “last mile”

Multi-Tier Applications

- Applications consist of tasks
  - Many separate components
  - Running on different machines
- Commodity computers
  - Many general-purpose computers
  - Not one big mainframe
  - Easier scaling



“Back-end” datacenter traffic

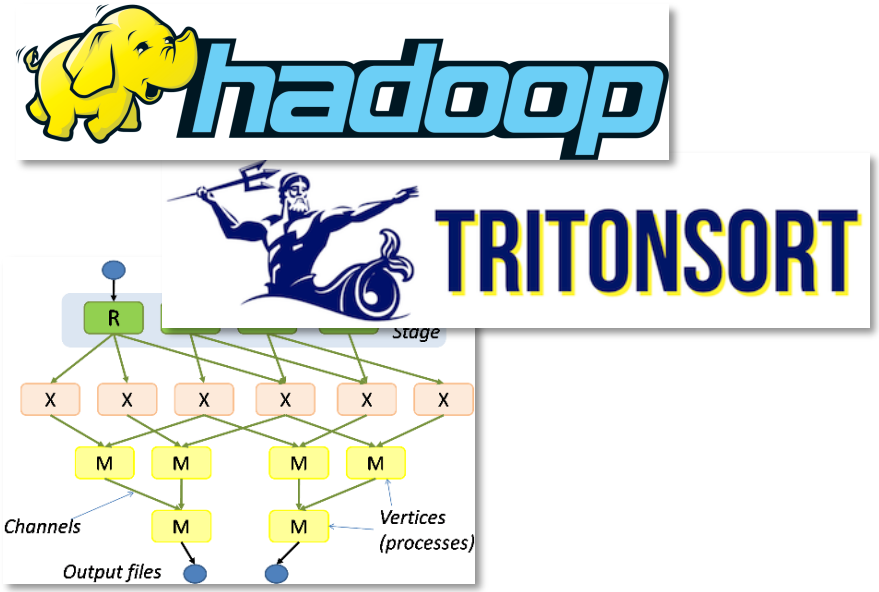
- Back-end analytics:
  - Connections between information
  - “Users who bought X also bought Y”
- Key differentiator determining success
  - Facebook vs Friendster
  - Amazon vs Buy.com
- Large-scale “join” computations spanning thousands of nodes
  - Need bandwidth as well as all-to-all connectivity



- Sorting / Searching
- Collaborative Filtering
- Map/Reduce
- Distributed Key/Value stores
- Video storage, post-production, and transmission

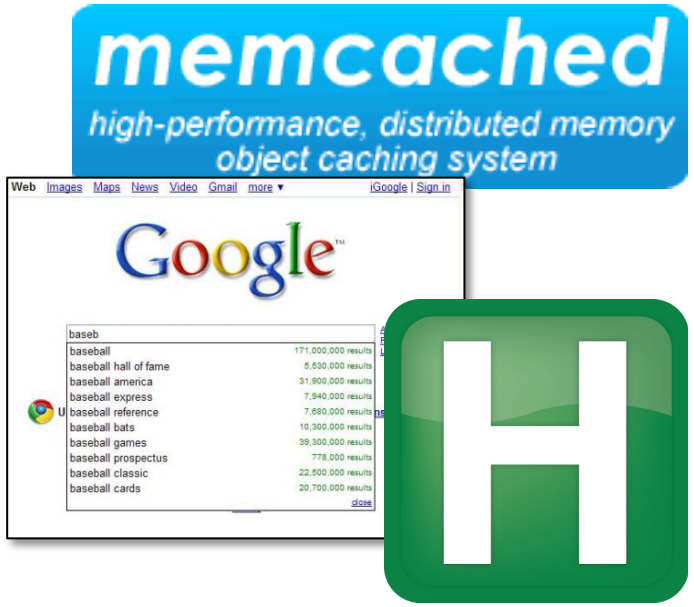
Data-intensive application requirements

All-to-all



- Performance gated on latency of shuffle phases
  - Need high bisection bandwidth

Gather/Scatter



- Performance gated on speed of slowest RPC/ parallel operation
  - Need low variance

Increasingly stringent network requirements

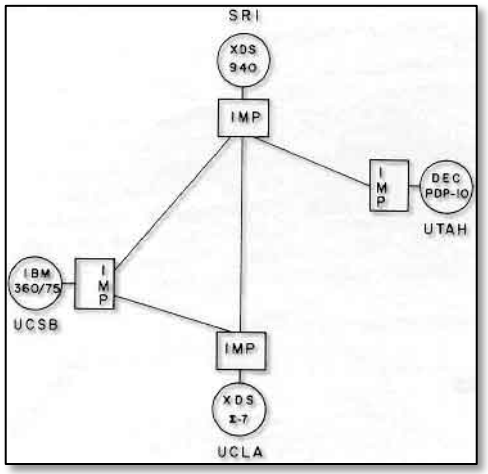




- Low one-way latency (10s of microseconds)
- 10 Gbps at TOR (and soon endhosts)
- Congestion-free operation/low queuing
- Dynamic traffic…
- …and an increasingly dynamic topology

From networks to backplanes

- Before:
  - Network connects servers to users (FTP, telnet, …)
  - Massive computing = tightly coupled supercomputer
    - Proprietary interconnects
    - Working sets of data



- Today:
  - Servers connected to each other
  - Data-intensive, web-scale
  - Massive computing = Datacenters
    - Commodity
    - Datacenter network *becomes the computing backplane*



Data Center Challenges

- Traffic load balancing
- Support for VM migration
- Achieving bisection bandwidth
- Power savings / Cooling
- Network management (provisioning)
- Security (dealing with multiple tenants)

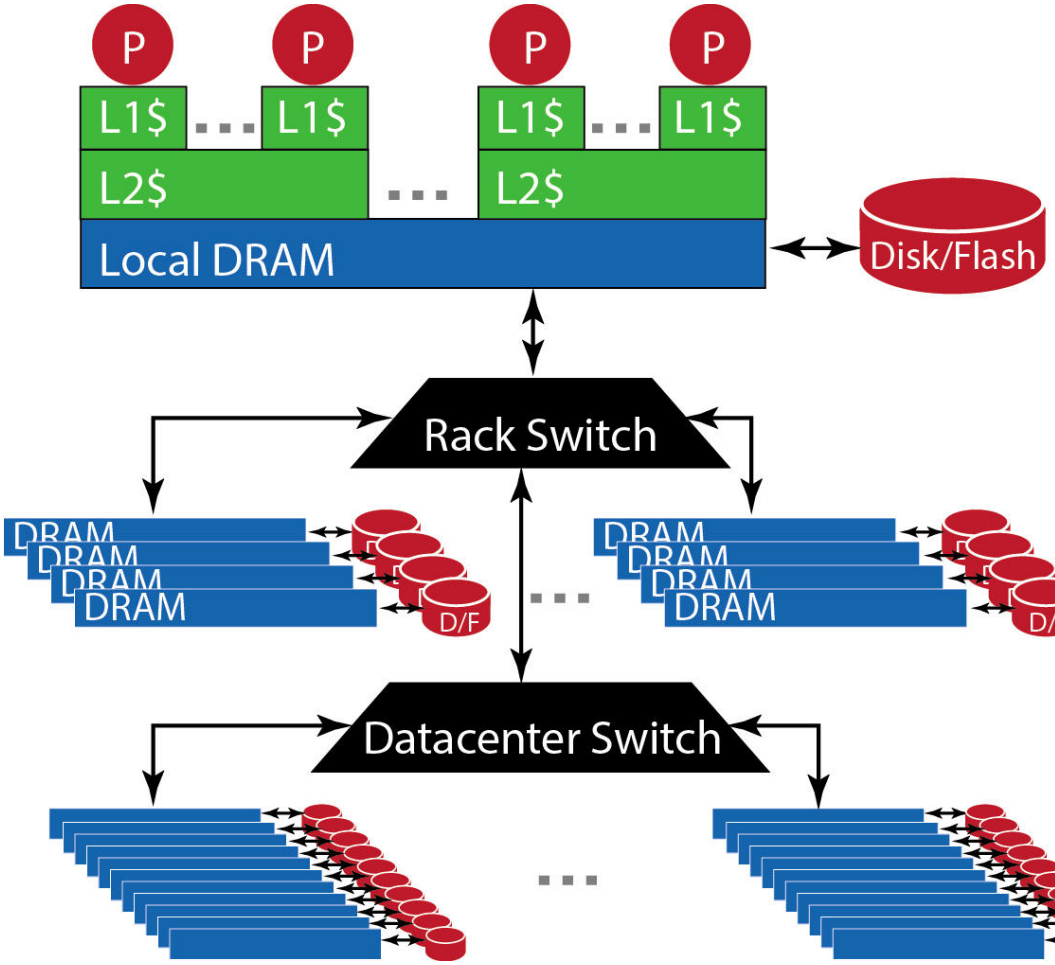
Data Center Storage Example

Photos @ Facebook

	April 2009	Current
Total	15 billion photos 60 billion images 1.5 petabytes	65 billion photos 260 billion images 20 petabytes
Upload Rate	220 million photos / week 25 terabytes	1 billion photos / week 60 terabytes
Serving Rate	550,000 images / sec	1 million images / sec

Finding a Needle in Haystack: Facebook's Photo Storage, OSDI'10

The storage hierarchy

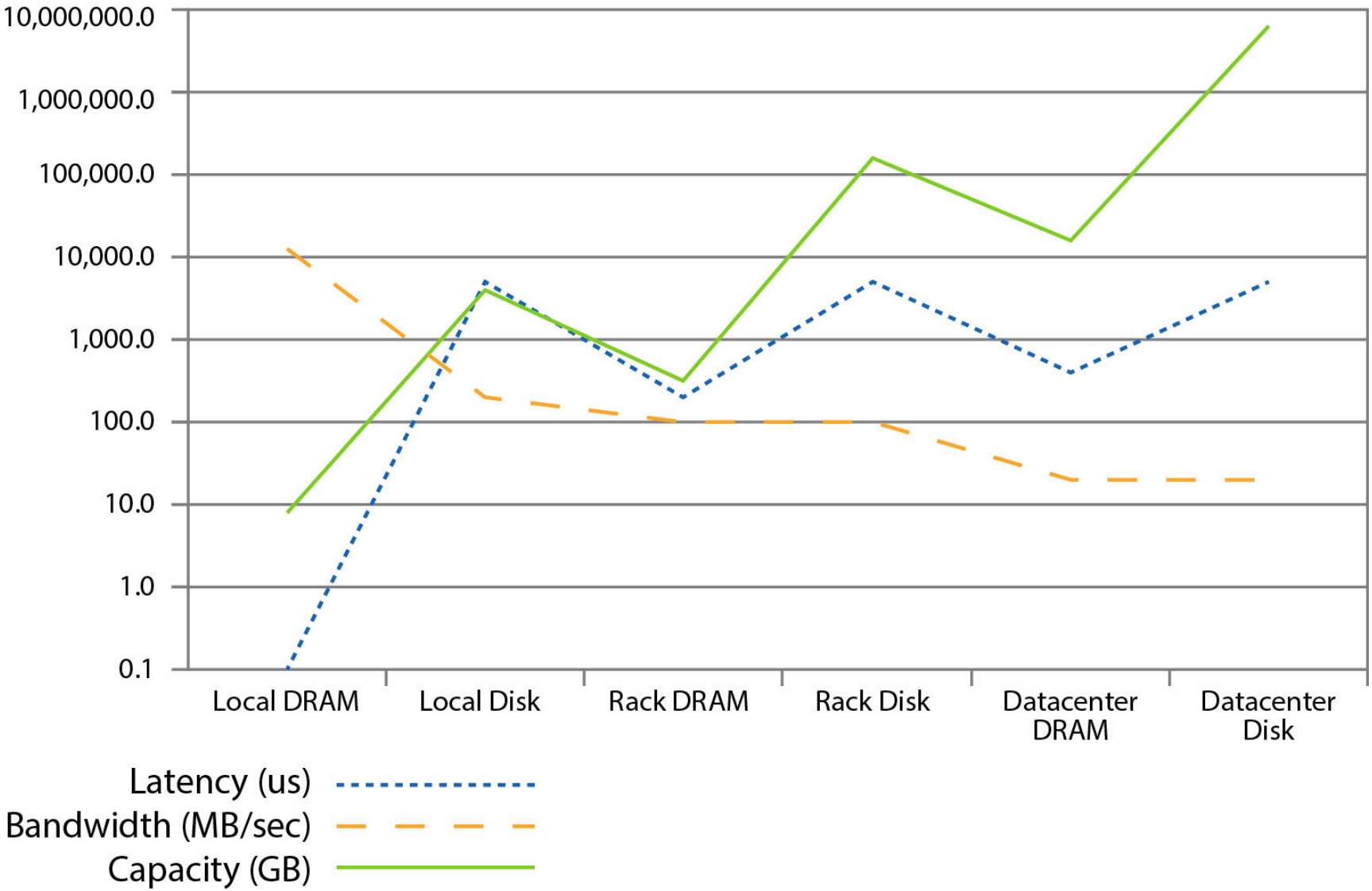


One Server
DRAM: 16 GB, 100 ns, 20 GB/s
Disk: 2T B, 10 ms, 200 MB/s
Flash: 128 GB, 100 us, 1 GB/s

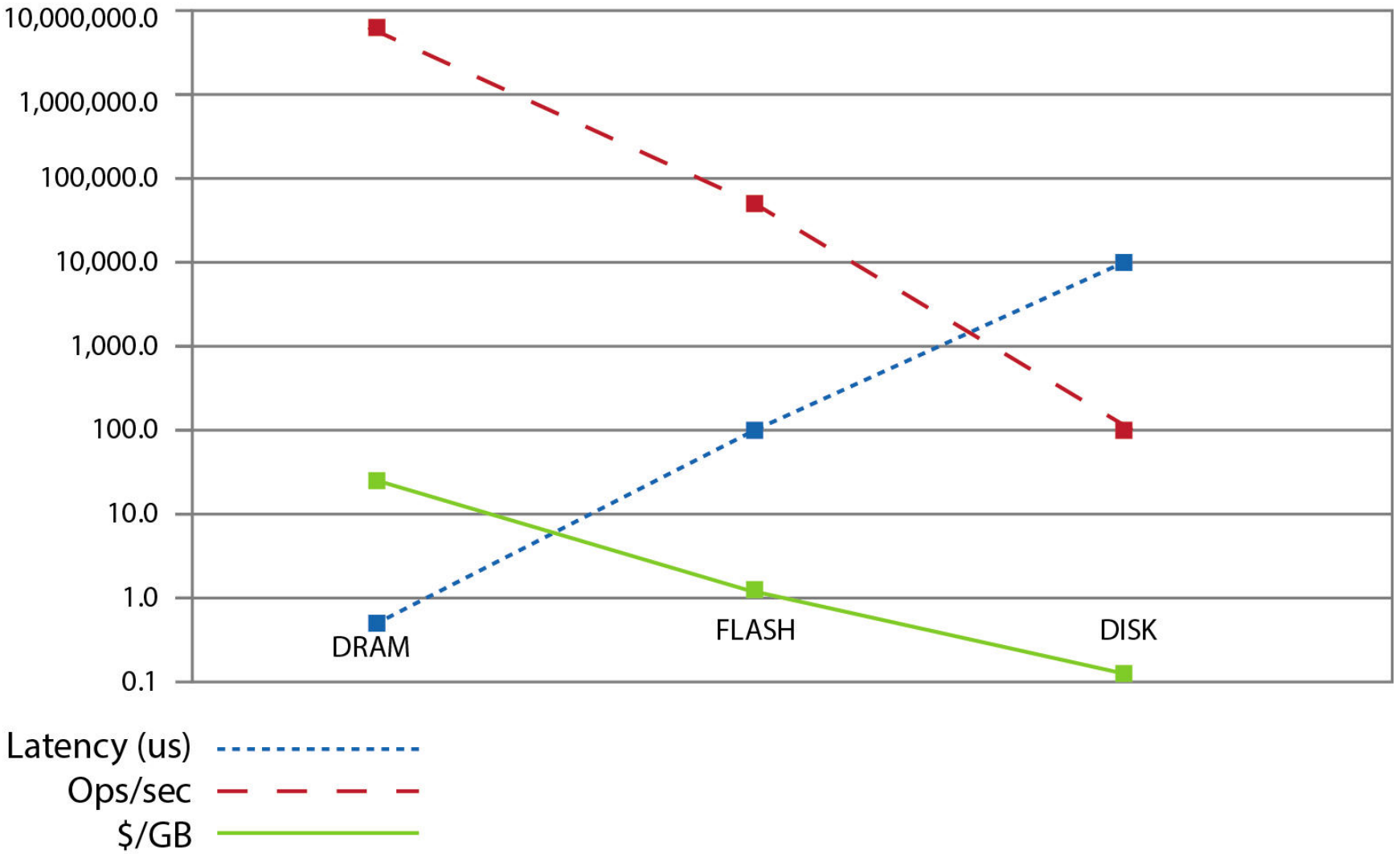
Local Rack (80 servers)
DRAM: 1 TB, 300 us, 100 MB/s
Disk: 160 TB, 11 ms, 100 MB/s
Flash: 20 TB, 400 us, 100 MB/s

Cluster (30 racks)
DRAM: 30 TB, 500 us, 10 MB/s
Disk: 4.80 PB, 12 ms, 10 MB/s
Flash: 600 TB, 600 us, 10 MB/s

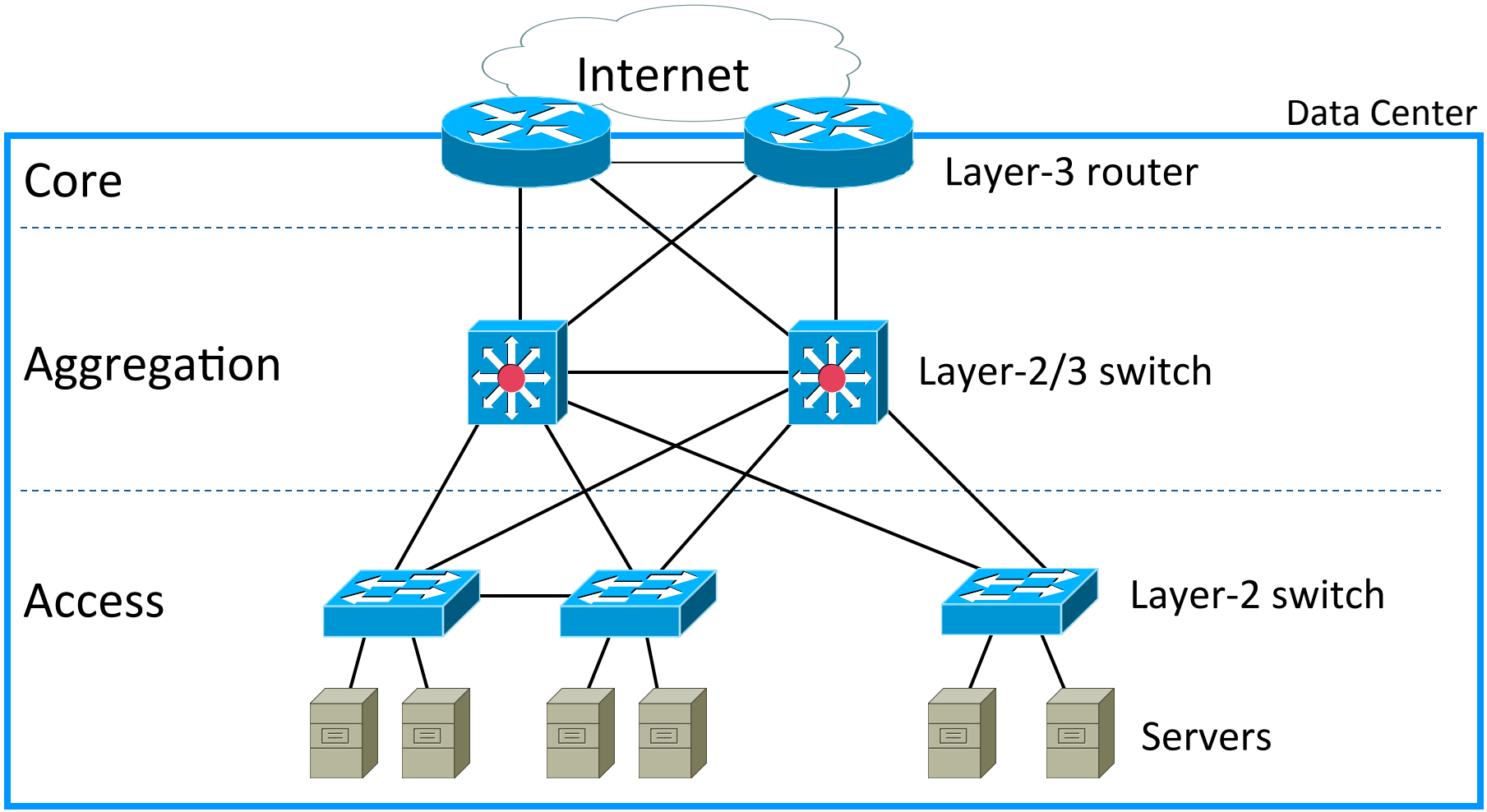
Latency, bandwidth, and capacity



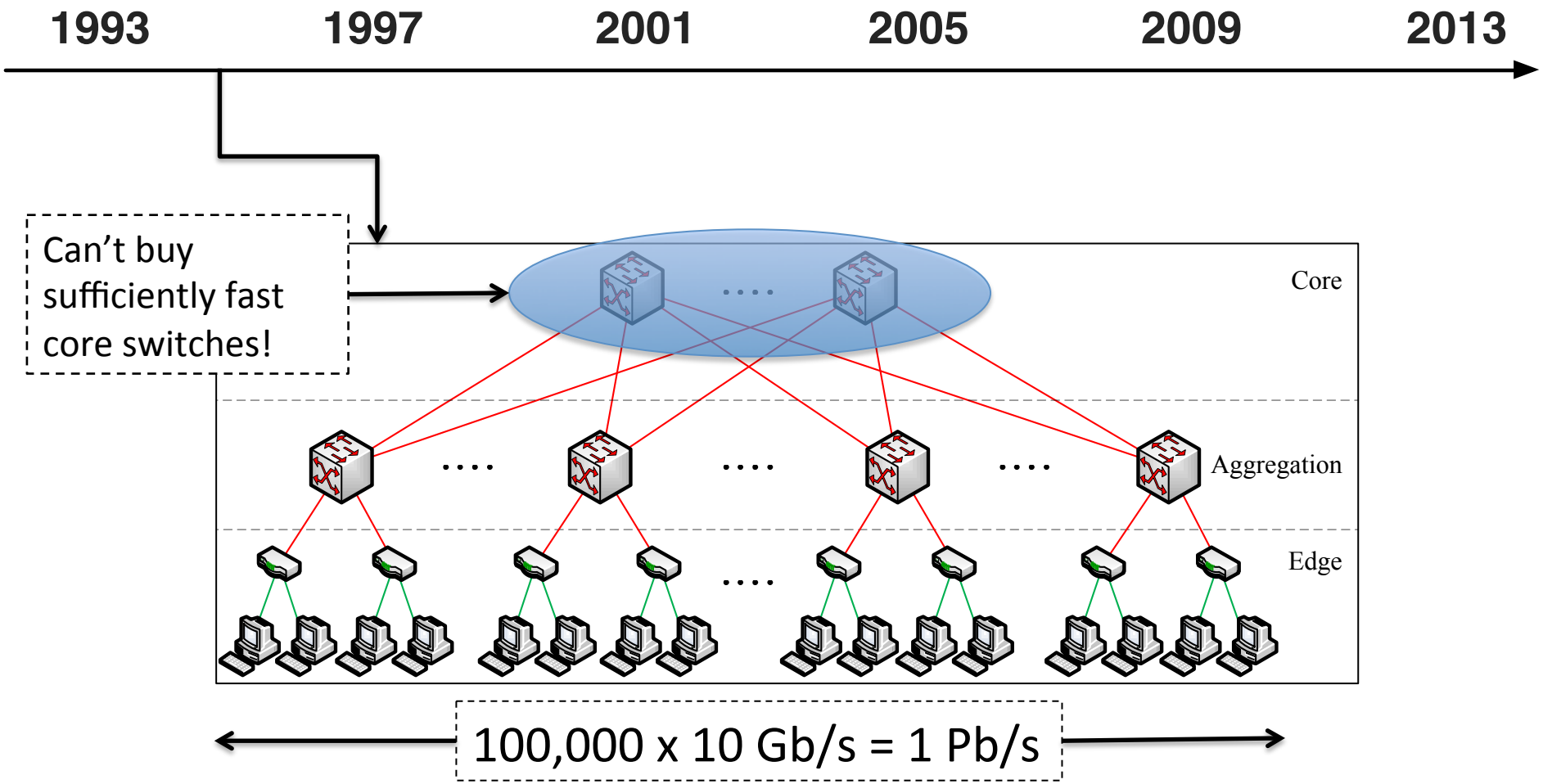
Performance of flash



Traditional DC Topology



Tree-based network topologies



DC Network Requirements

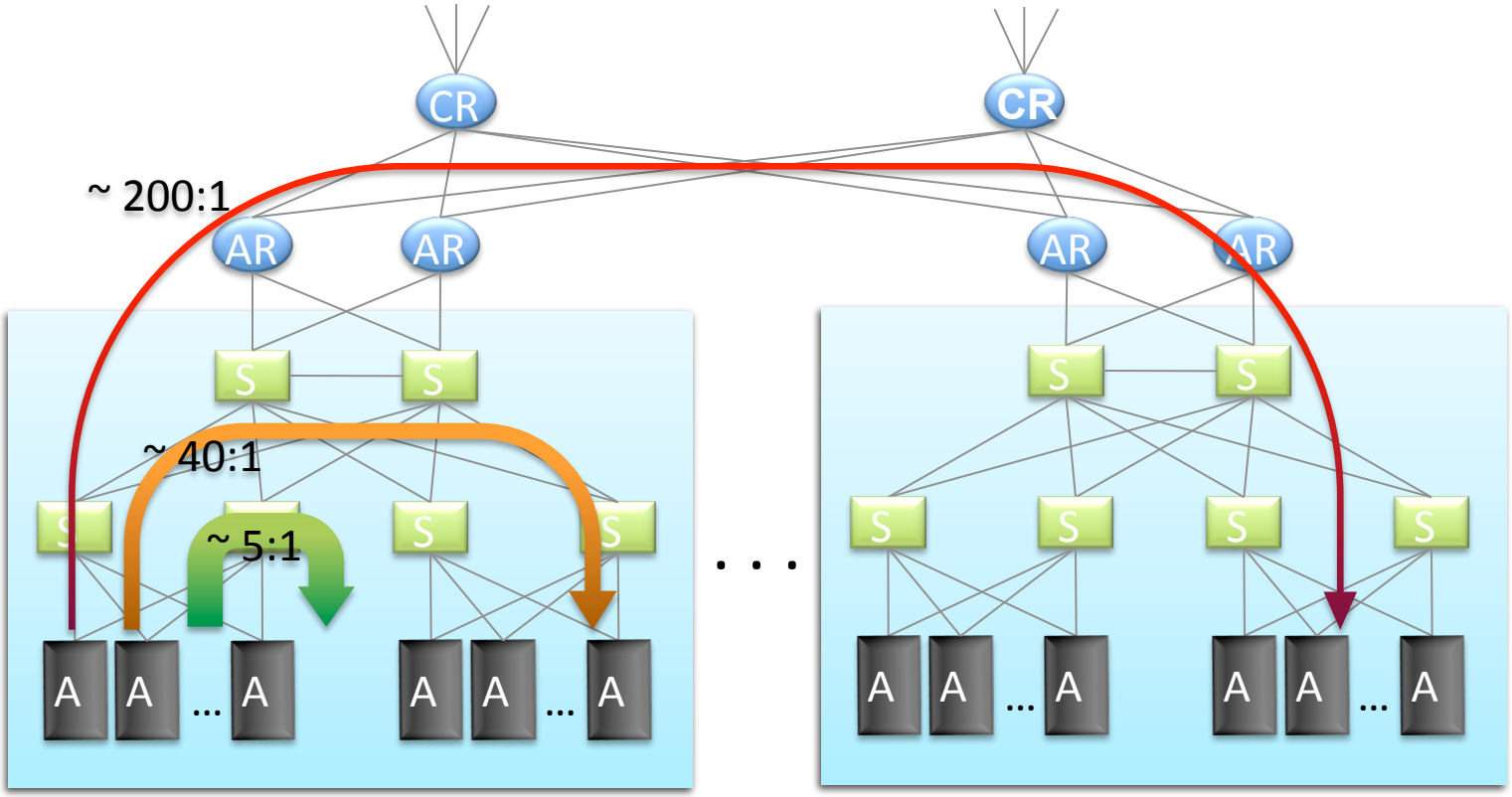
- Scalability
  - Incremental build out?
- Reliability
  - Loop free forwarding
- VM migration

- Reasonable management burden
  - Humans in the loop?

Traditional Topologies

- Over subscription of links higher up in the topology
- Tradeoff between cost and provisioning
- Single point of failure

Capacity Bottlenecks



Management: L2 vs. L3

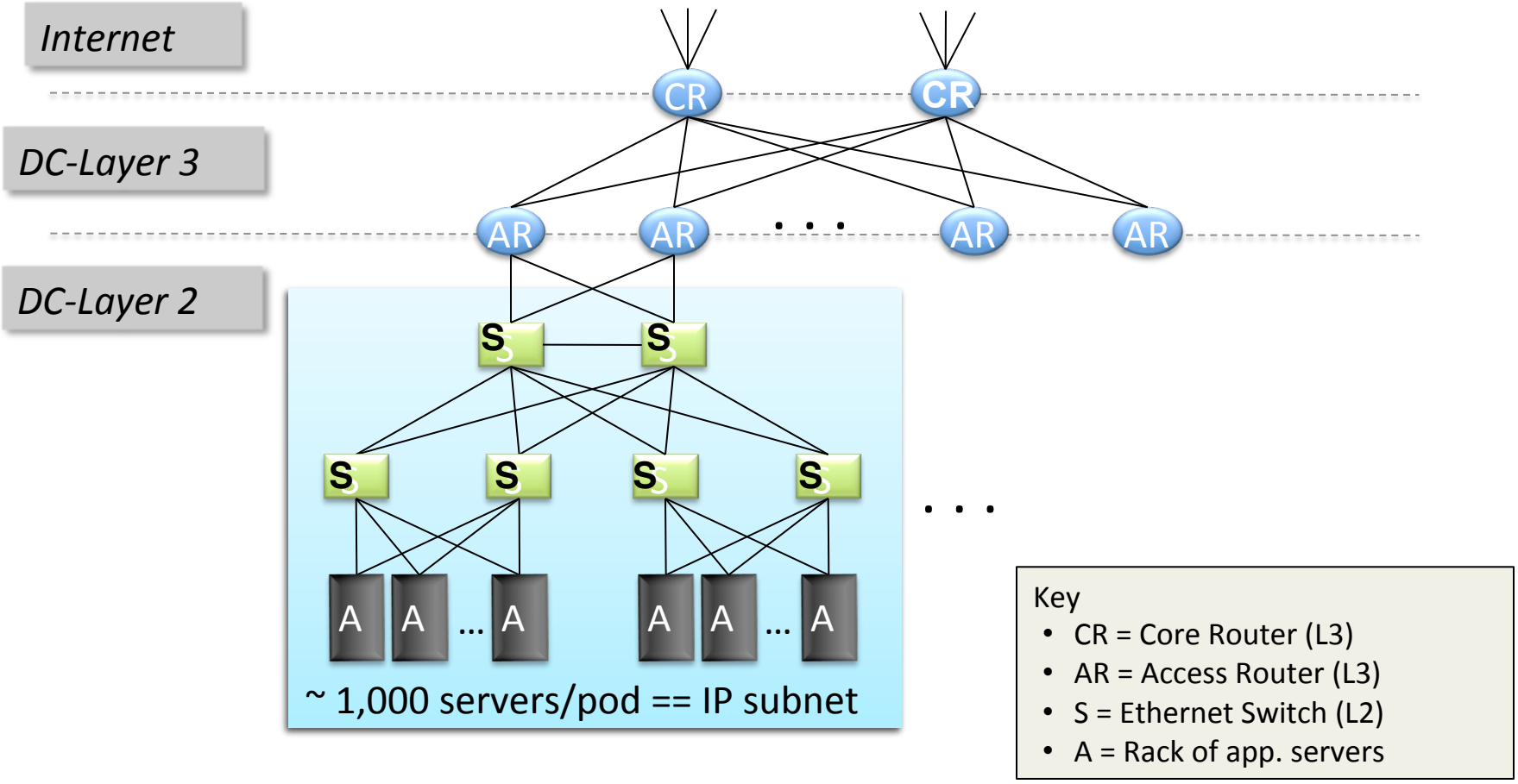
- Ethernet switching (layer 2)
  - Cheaper switch equipment
  - Fixed addresses and auto-configuration
  - Seamless mobility, migration, and failover
- IP routing (layer 3)
  - Scalability through hierarchical addressing
  - Efficiency through shortest-path routing
  - Multipath routing through equal-cost multipath

- Data centers often connect layer-2 islands by IP routers

Advantages of Layer 2

- Certain monitoring apps require server with same role to be on the same VLAN
- Using same IP on dual homed servers
- Allows organic growth of server farms
- VM migration is easier

Layer 2 Pods w/L3 Backbone

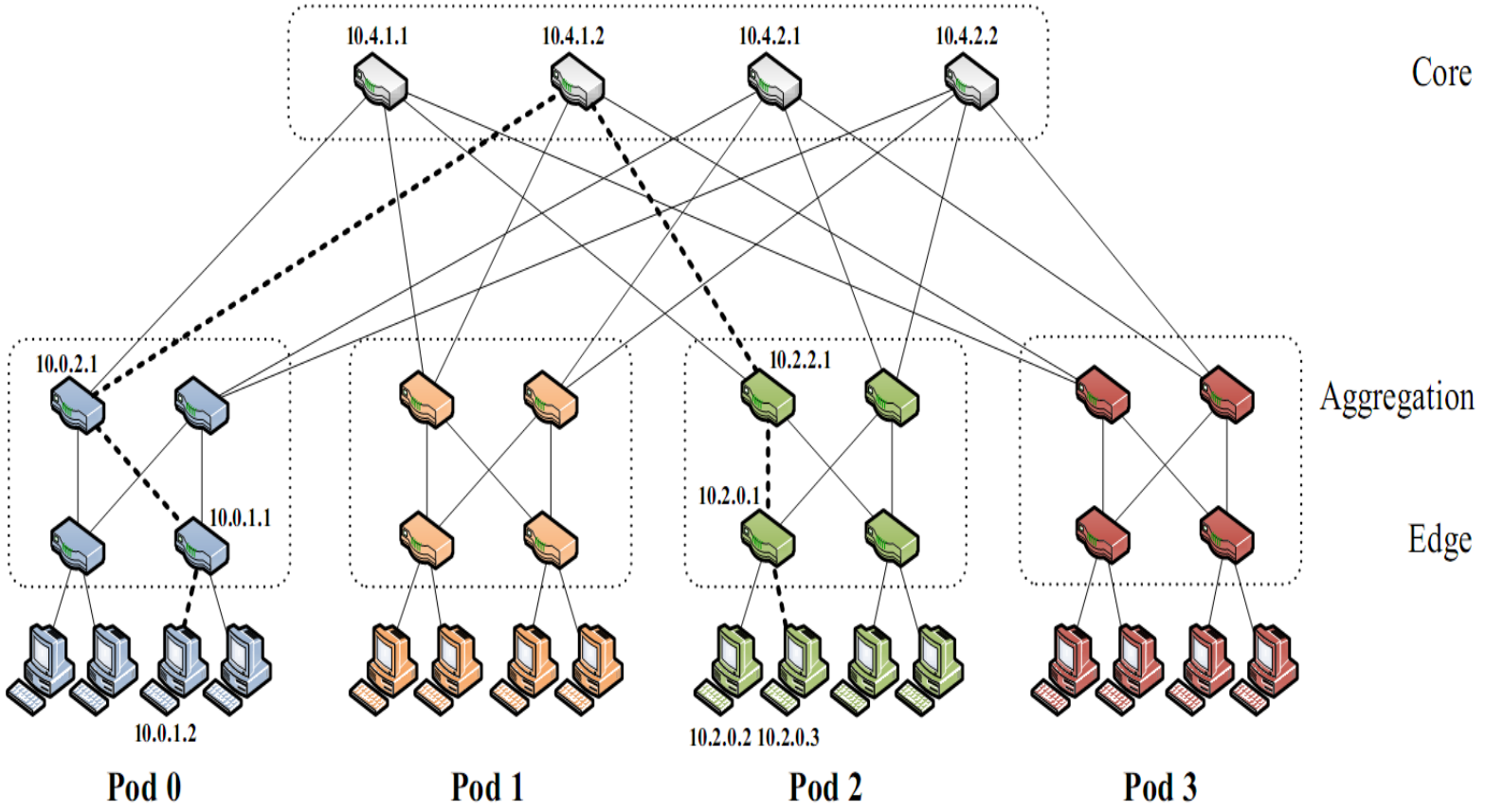


FAT Tree-Based Solution

- An all Layer-3 solution

- Connect end-host together using a “fat-tree” topology
  - Infrastructure consist of cheap devices
    - Each port supports same speed as endhost
  - All devices can transmit at line speed if packets are distributed along existing paths
  - A k-port fat tree can support $k^3/4$ hosts

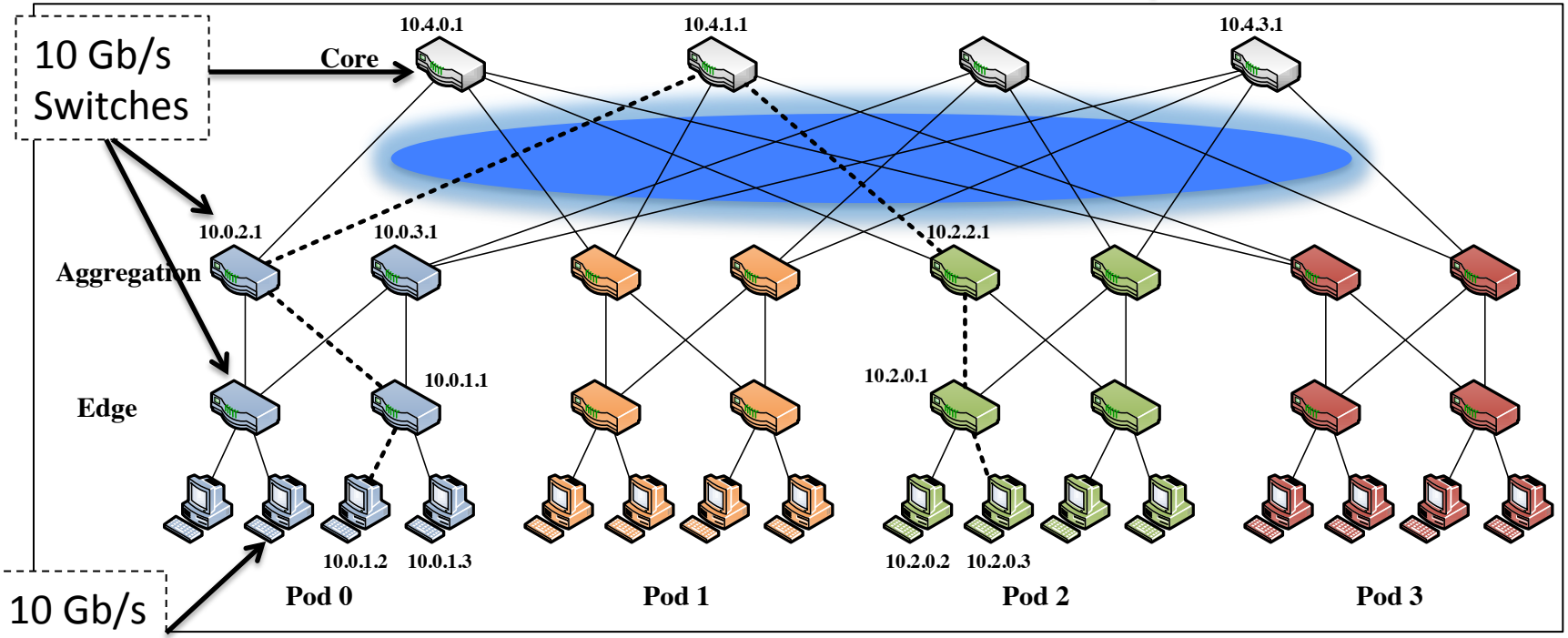
“Fat-Tree” Topology



Folded-Clos multi-rooted Trees

1993 1997 2001 2005 2009 2013

Al Fares, et al., Sigcomm'08



Major data center principle: parallelism

- Unlike smaller networks, data center networks exhibit massive parallelism
  - FatTrees have many many paths from source to destination
  - Computation spread across many processors located in many servers
  - A single storage service spread across 1000s of individual storage servers
- How to program such a beast?
  - Next lecture!