

Systems Research: a Graduate Student Perspective

Nodir Kodirov

PhD student at Computer Science



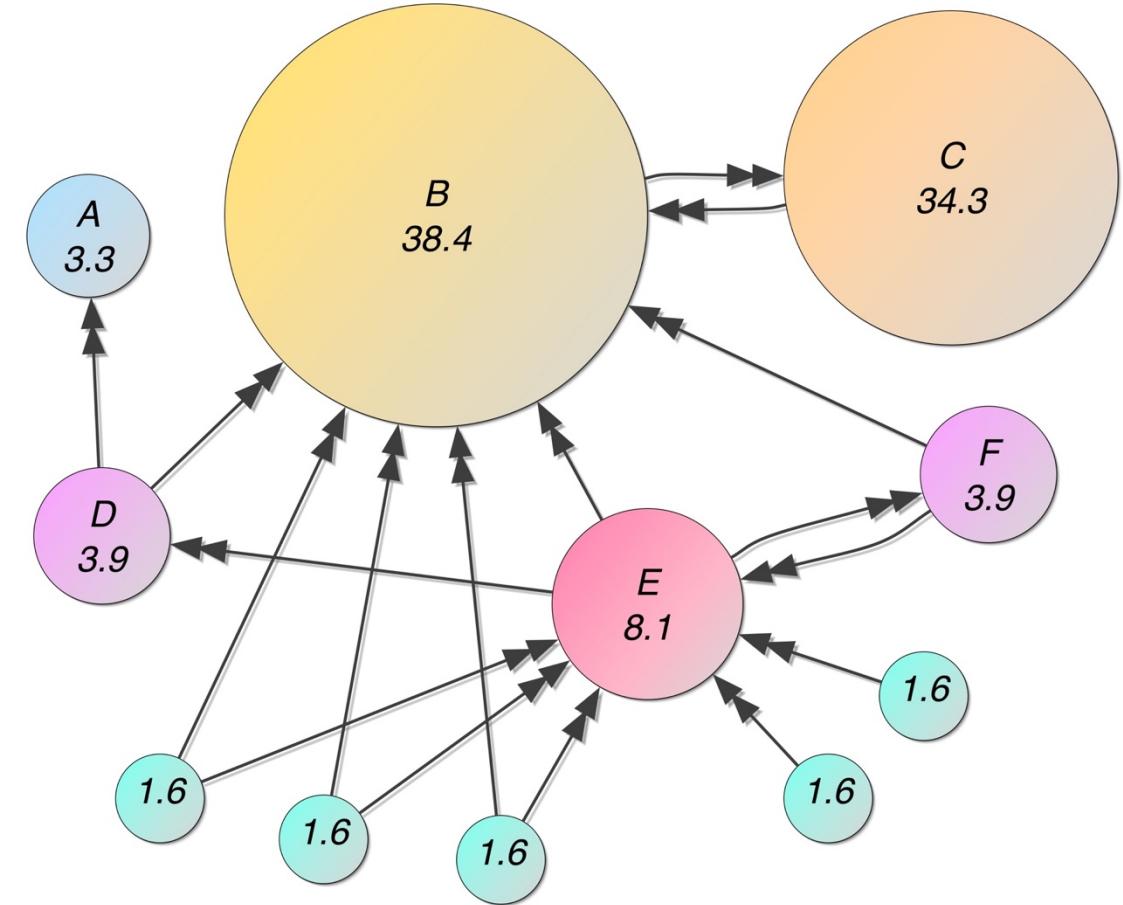
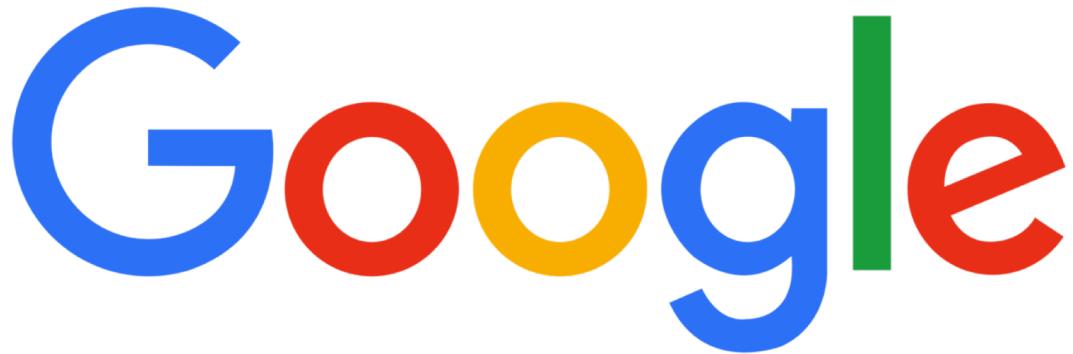
THE UNIVERSITY
OF BRITISH COLUMBIA

<https://www.cs.ubc.ca/~knodir/>

Research: Pushes Science Forward

The PageRank Citation Ranking:
Bringing Order to the Web

January 29, 1998

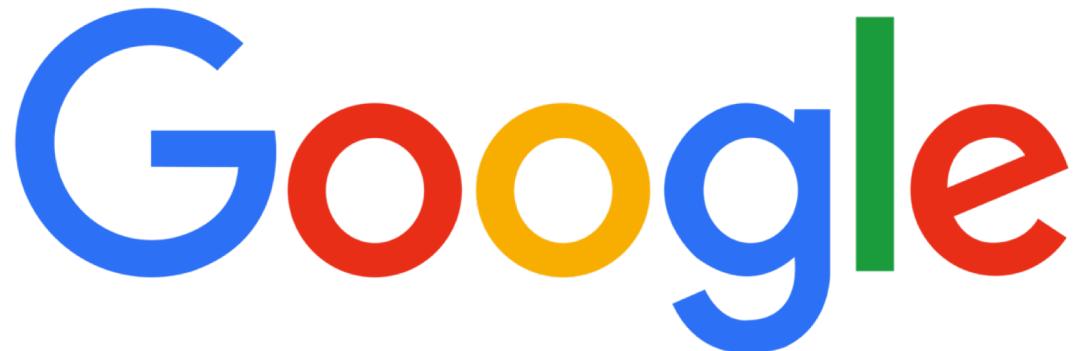


L. Page, S. Brin, R. Motwani, and T. Winograd., The PageRank Citation Ranking: Bringing Order to the Web
Proceedings of the 7th International World Wide Web Conference, 1998

Research: Pushes Science Forward

The PageRank Citation Ranking:
Bringing Order to the Web

January 29, 1998

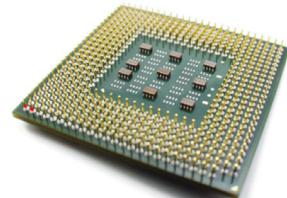


Systems Research

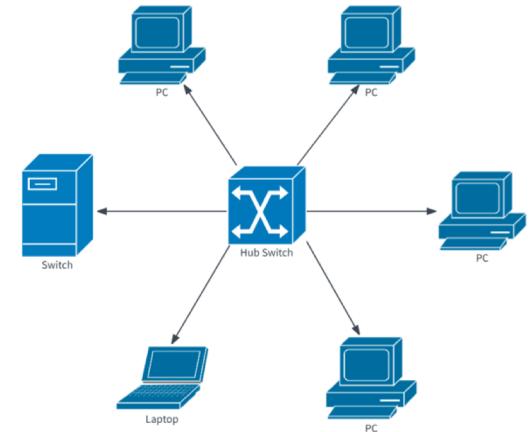
Storage



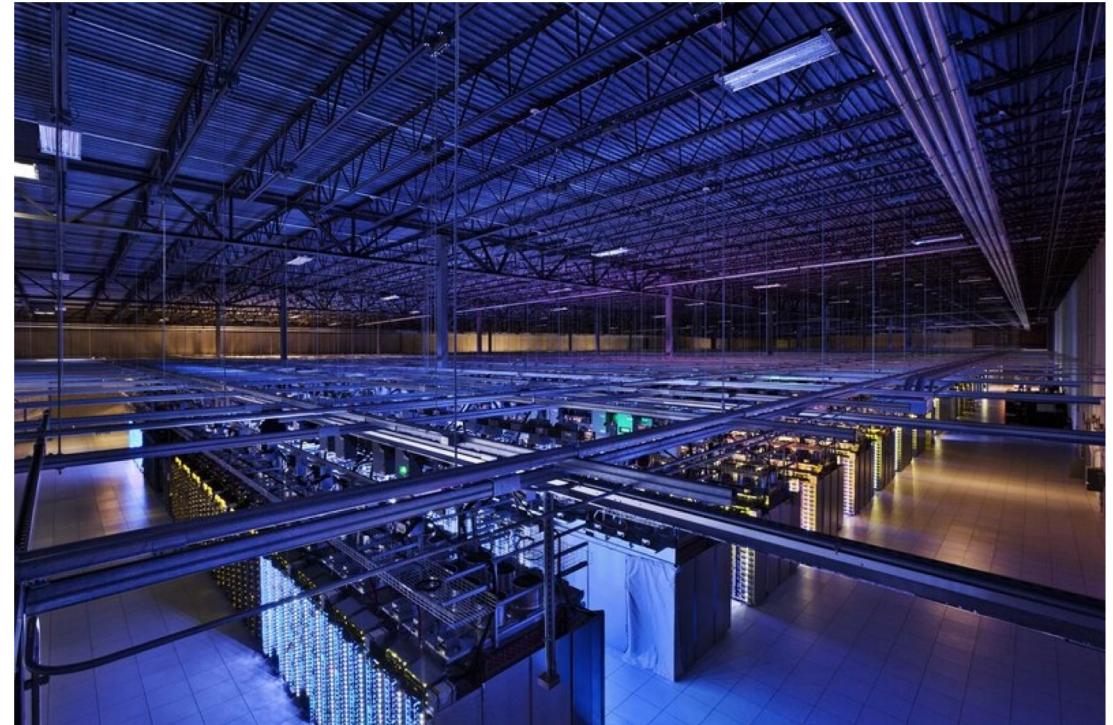
Compute



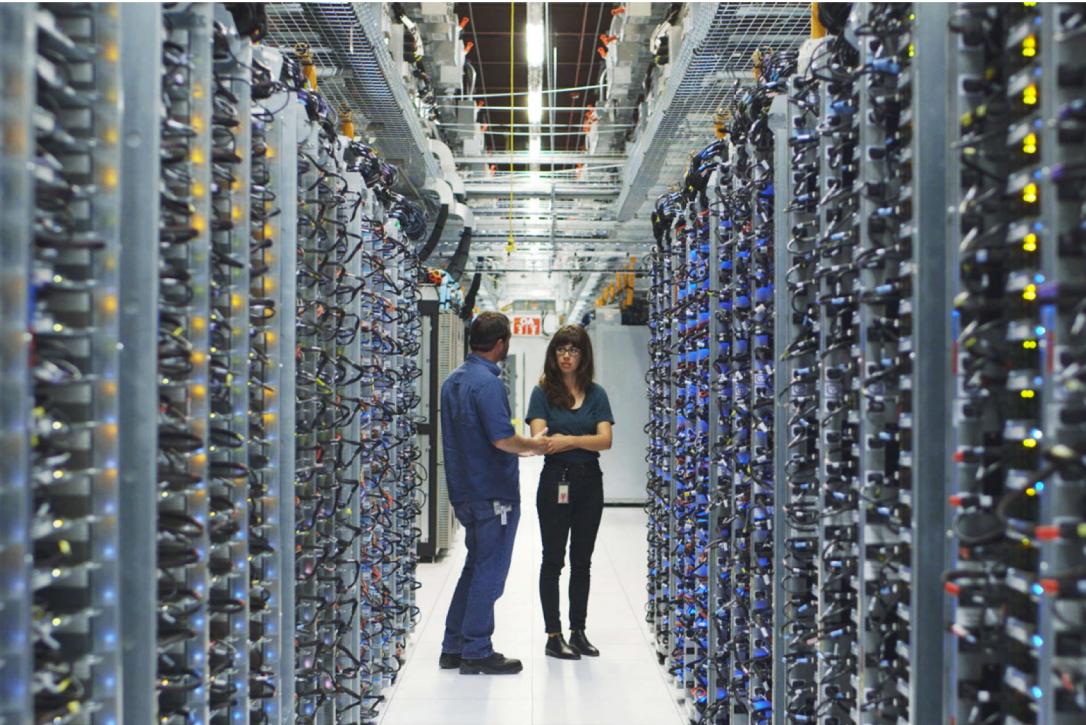
Networking



Systems Research: for data centers



Systems Research: Storage



- Problem (around year 2000)
 - Large data (100s of TB, 1000s of disks)
 - Concurrent requests (100s of clients)
 - Continuous processing (24/7)
 - Disk failures (more likely at scale)
- Solution
 - Distributed file system
 - Google File System (GFS)

Systems Research: Storage

The Google File System

Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung
Google*

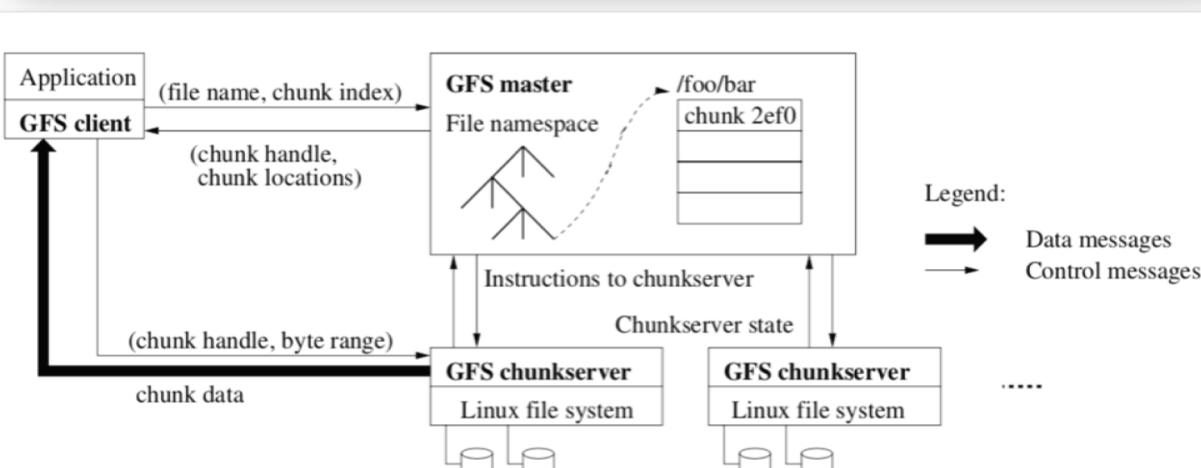


Figure 1: GFS Architecture

- Problem (around year 2000)
 - Large data (100s of TB, 1000s of disks)
 - Concurrent requests (100s of clients)
 - Continuous processing (24/7)
 - Disk failures (more likely at scale)
- Solution
 - Distributed file system
 - Google File System (GFS)

Systems Research: Storage

GFS inspired HDFS
(Hadoop Distributed File System)



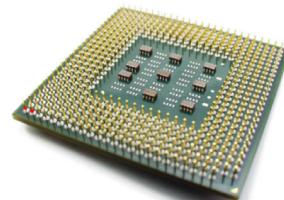
- Problem (around year 2000)
 - Large data (100s of TB, 1000s of disks)
 - Concurrent requests (100s of clients)
 - Continuous processing (24/7)
 - Disk failures (more likely at scale)
- Solution
 - Distributed file system
 - Google File System (GFS)

Systems Research

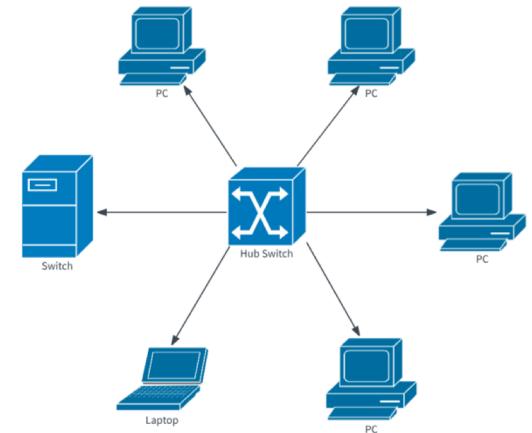
Storage



Compute



Networking

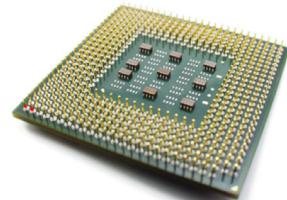


Systems Research

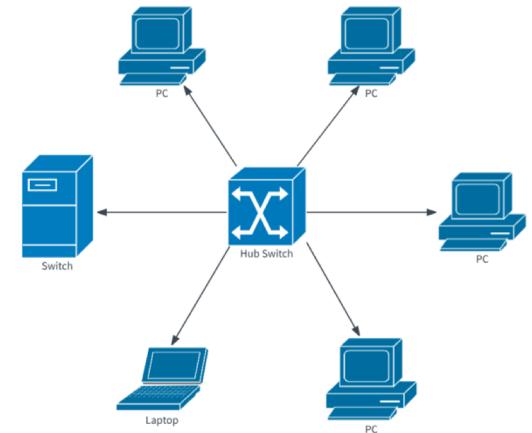
Storage



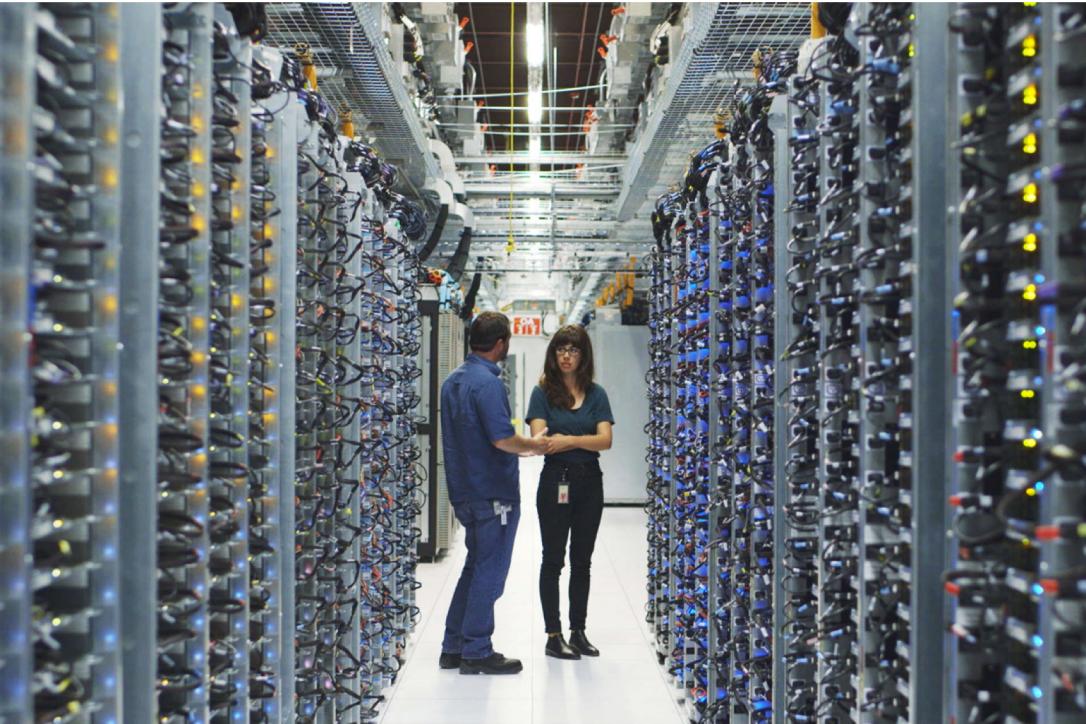
Compute



Networking



Systems Research: Compute



- Problem (around year 2000)
 - Lots of servers (1000s)
 - Low utilization per server ($\lesssim 12\%$)
 - Isolation between apps (security, performance)
- Solution
 - Server virtualization: run multiple Virtual Machines (VM) on a single server

Systems Research: Compute

Xen and the Art of Virtualization

Paul Barham*, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris,
Alex Ho, Rolf Neugebauer†, Ian Pratt, Andrew Warfield

University of Cambridge Computer Laboratory
15 JJ Thomson Avenue, Cambridge, UK, CB3 0FD

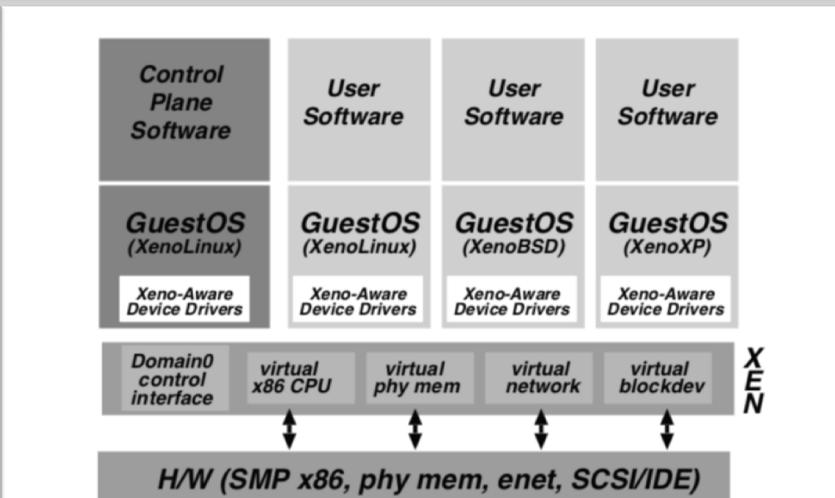


Figure 1: The structure of a machine running the Xen hypervisor, hosting a number of different guest operating systems, including *Domain0* running control software in a XenoLinux environment.

- Problem (around year 2000)
 - Lots of servers (1000s)
 - Low utilization per server ($\lesssim 12\%$)
 - Isolation between apps (security, performance)
- Solution
 - Server virtualization: run multiple Virtual Machines (VM) on a single server

Systems Research: Compute

Amazon Web Services (AWS) adopted Xen and offered EC2 (Elastic Compute Cloud)



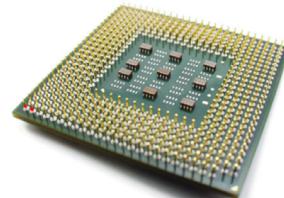
- Problem (around year 2000)
 - Lots of servers (1000s)
 - Low utilization per server ($\lesssim 12\%$)
 - Isolation between apps (security, performance)
- Solution
 - Server virtualization: run multiple Virtual Machines (VM) on a single server

Systems Research

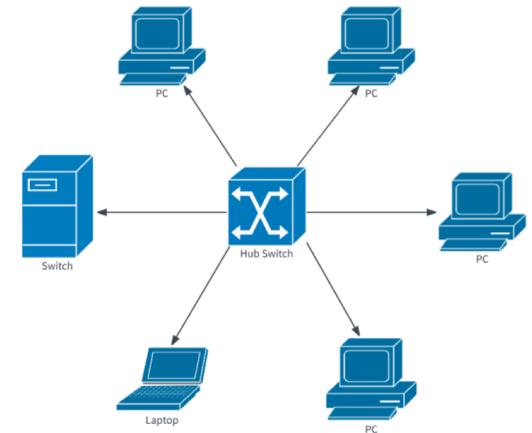
Storage



Compute



Networking

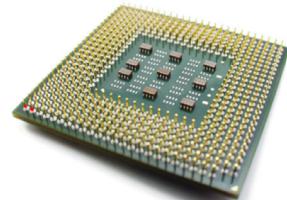


Systems Research

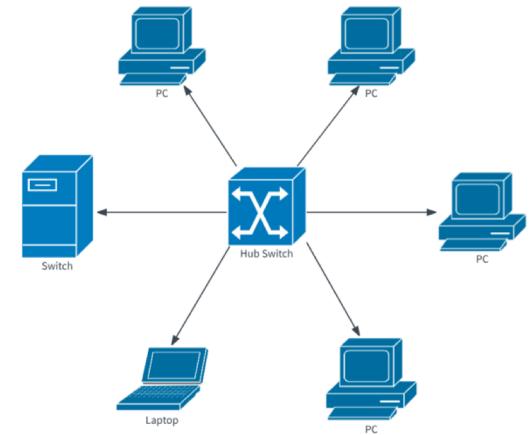
Storage



Compute



Networking



Systems Research: Networking



- Problem (around year 2005)
 - Networks evolve **slow**
 - Difficult to **test** new protocols
 - **Expensive** to build and operate
- Solution
 - Software-defined networking (**SDN**)
 - Incentivize network equipment vendors (e.g., Cisco, Juniper) to support **open protocols** (e.g., OpenFlow)

Systems Research: Networking

OpenFlow: Enabling Innovation in Campus Networks

Nick McKeown
Stanford University

Tom Anderson
University of Washington

Hari Balakrishnan
MIT

Guru Parulkar
Stanford University

Larry Peterson
Princeton University

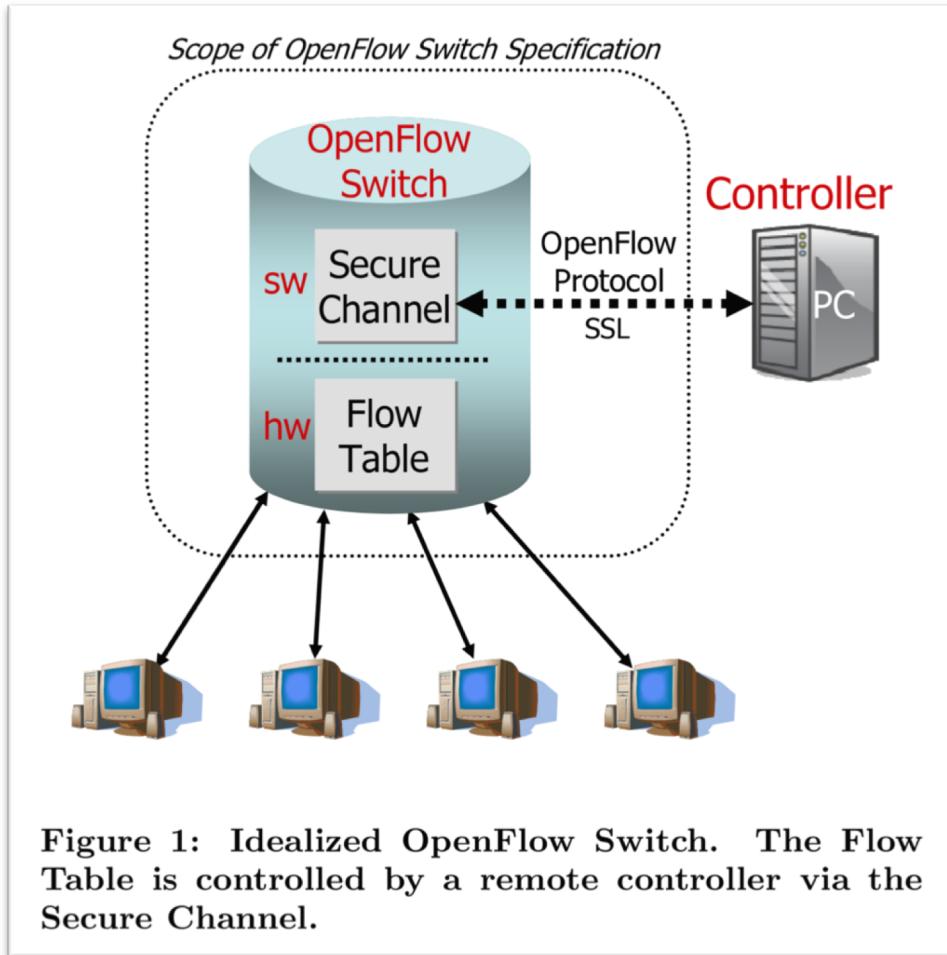
Jennifer Rexford
Princeton University

Scott Shenker
University of California,
Berkeley

Jonathan Turner
Washington University in
St. Louis

- Problem (around year 2005)
 - Networks evolve **slow**
 - Difficult to **test** new protocols
 - **Expensive** to build and operate
- Solution
 - Software-defined networking (**SDN**)
 - Incentivize network equipment vendors (e.g., Cisco, Juniper) to support **open protocols** (e.g., OpenFlow)

Systems Research: Networking



- Problem (around year 2005)
 - Networks evolve **slow**
 - Difficult to **test** new protocols
 - **Expensive** to build and operate
- Solution
 - Software-defined networking (**SDN**)
 - Incentivize network equipment vendors (e.g., Cisco, Juniper) to support **open protocols** (e.g., OpenFlow)

Systems Research: Networking



ARISTA



BAREFOOT metaswitch
NETWORKS



JUNIPER
NETWORKS

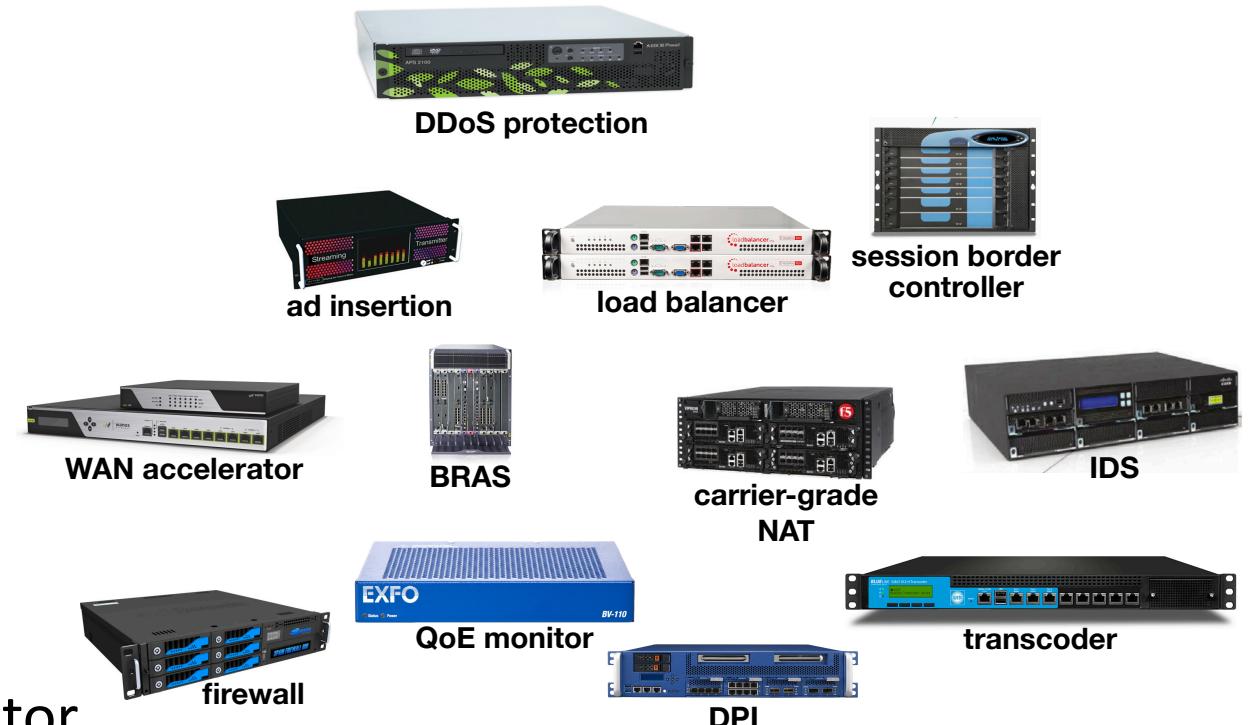


- Problem (around year 2005)
 - Networks evolve **slow**
 - Difficult to **test** new protocols
 - **Expensive** to build and operate
- Solution
 - Software-defined networking (**SDN**)
 - Incentivize network equipment vendors (e.g., Cisco, Juniper) to support **open protocols** (e.g., OpenFlow)

My research: Network Functions

My research: Network Functions

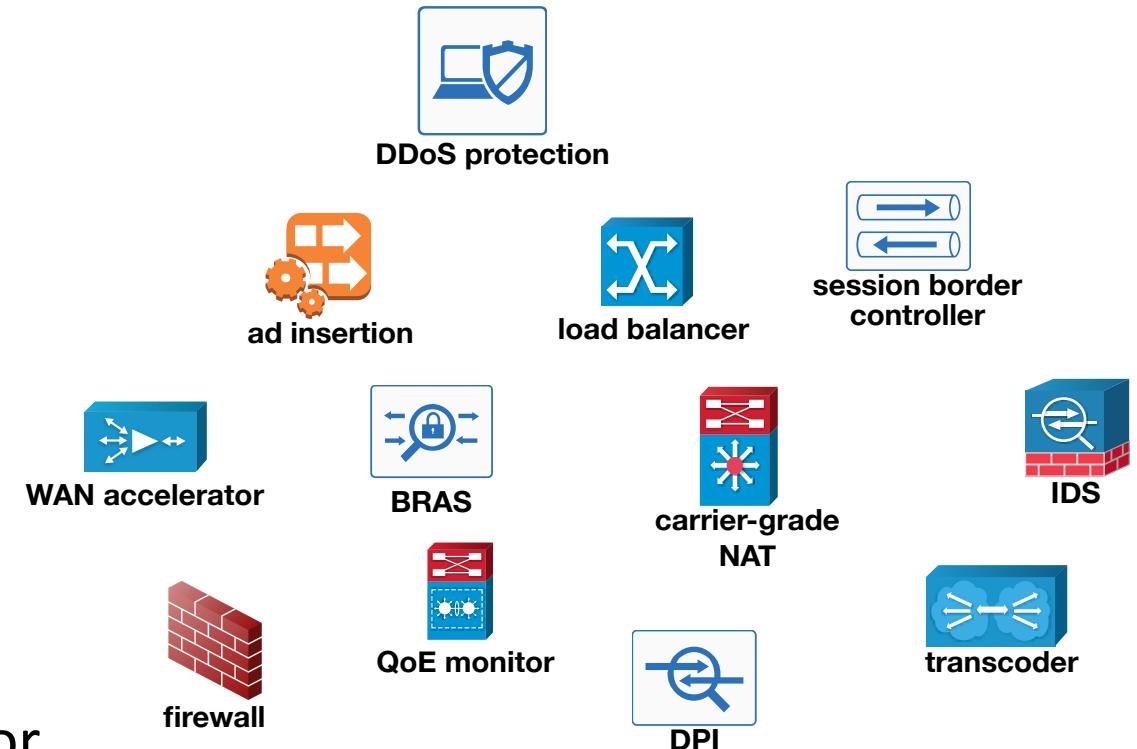
- Security
 - Firewall, DDoS protection, DPI
- Monitoring
 - QoE monitor, Network Stats
- Services
 - Ad insertion, Transcoder
- Network optimization
 - NAT, Load-balancer, WAN accelerator



Sherry et al. find # of middleboxes are \approx to # of L2/L3 devices in enterprise

My research: Network Functions

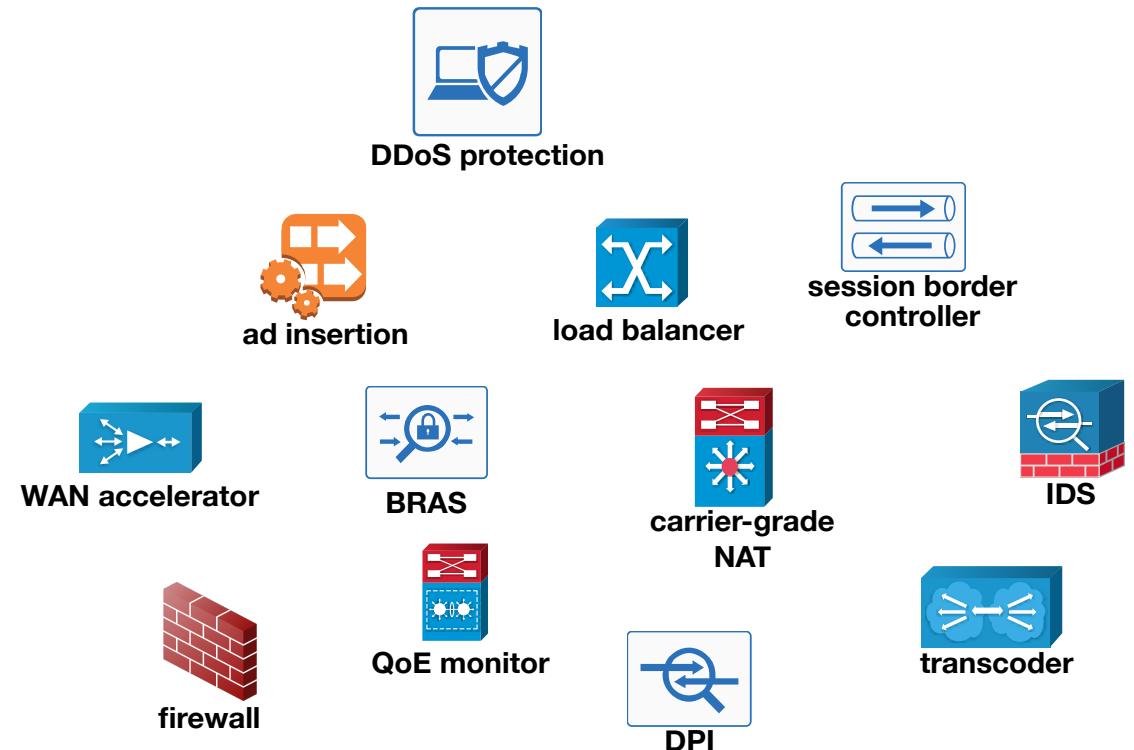
- Security
 - Firewall, DDoS protection, DPI
- Monitoring
 - QoE monitor, Network Stats
- Services
 - Ad insertion, Transcoder
- Network optimization
 - NAT, Load-balancer, WAN accelerator



Sherry et al. find # of middleboxes are \approx to # of L2/L3 devices in enterprise

Benefits of Virtualized Network Functions (VNF)

- **Elasticity**
 - Quick scale up and down NFs
- **Fast upgrades**
 - No need to wait for new hardware
- **Quick configuration, recovery**
 - Failover to the backup NF instance
- **Outsourcing**

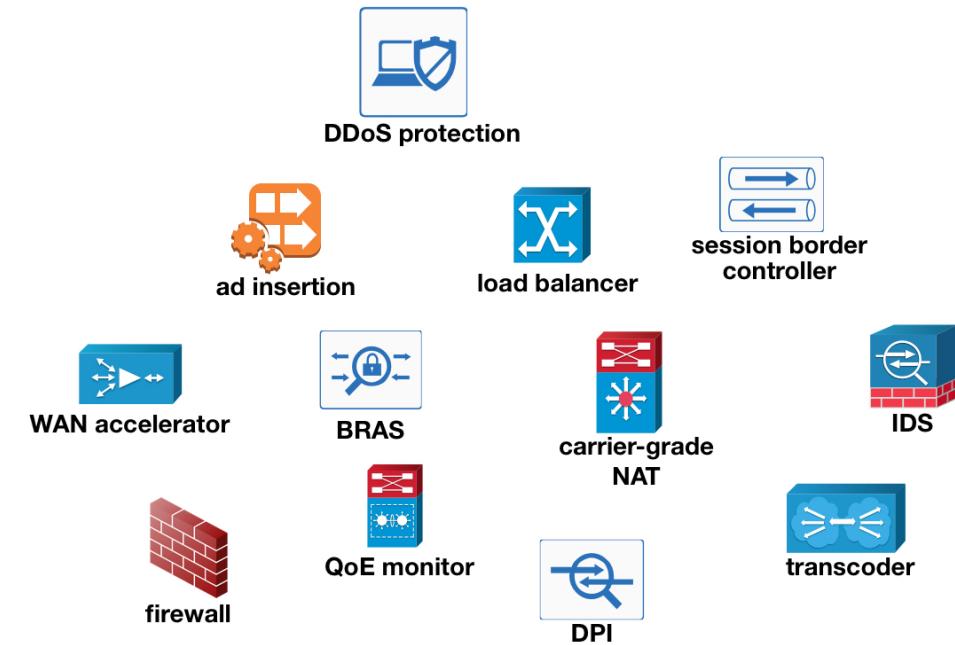
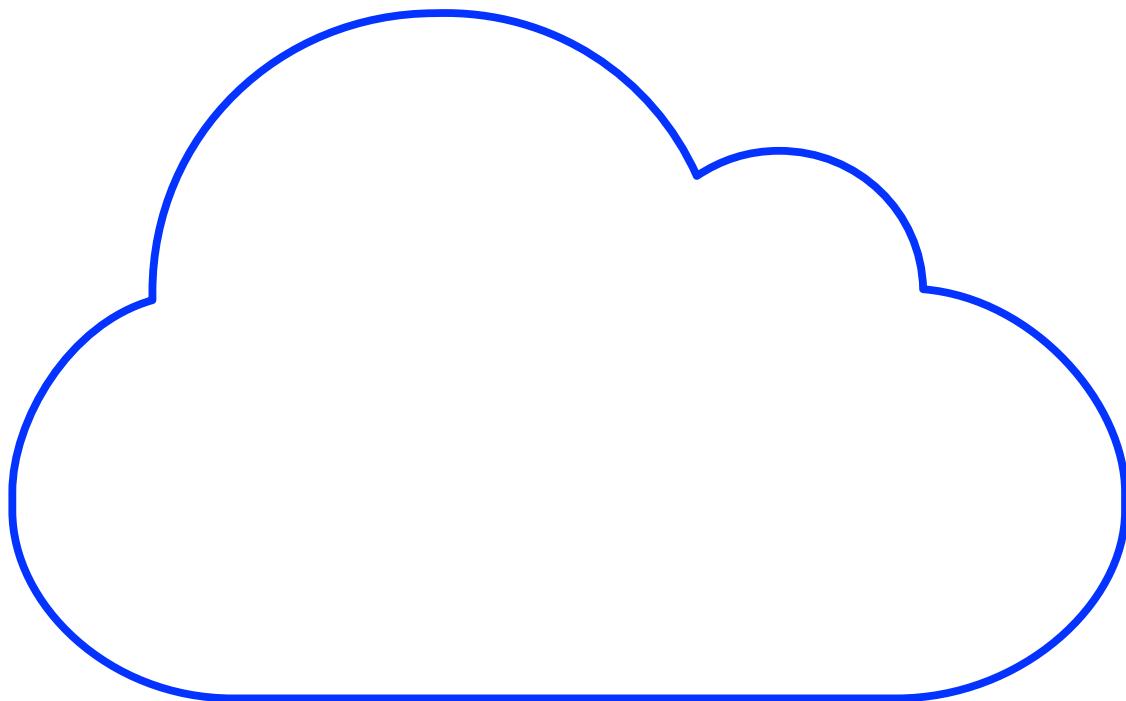


Sherry et al. Making Middleboxes Someone Else's Problem: Network Processing as a Cloud Service, SIGCOMM'12

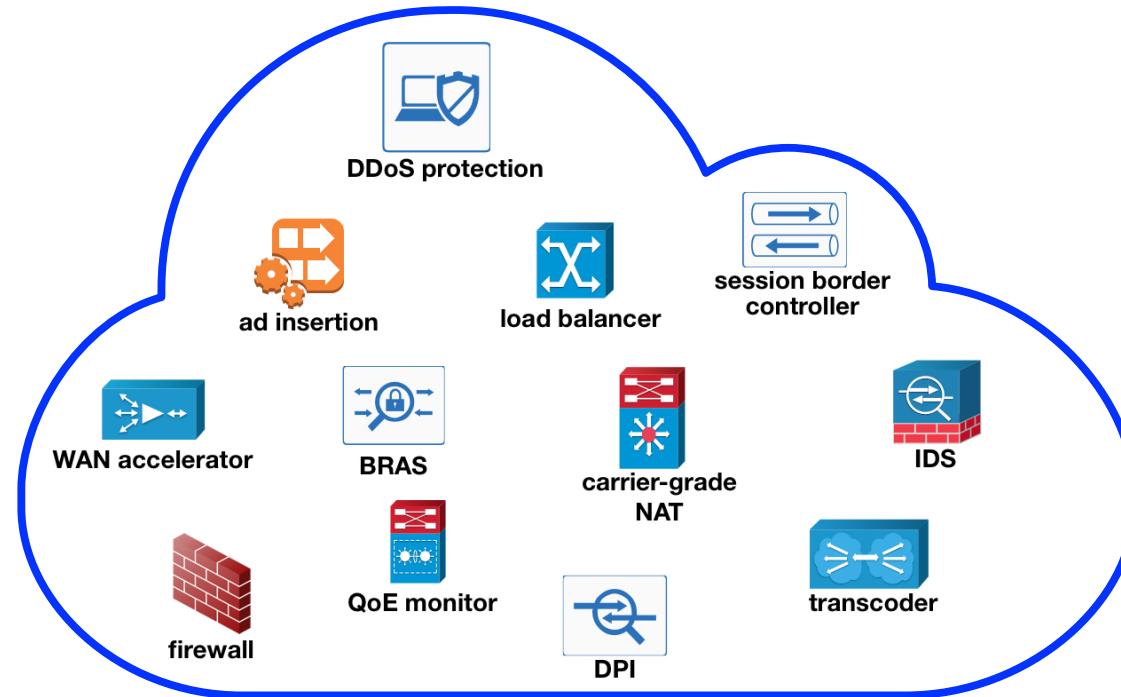
Rajagopalan et al., Split/Merge: System Support for Elastic Execution in Virtual Middleboxes, NSDI'13

Martins et al., ClickOS and the Art of Network Function Virtualization, NSDI'14

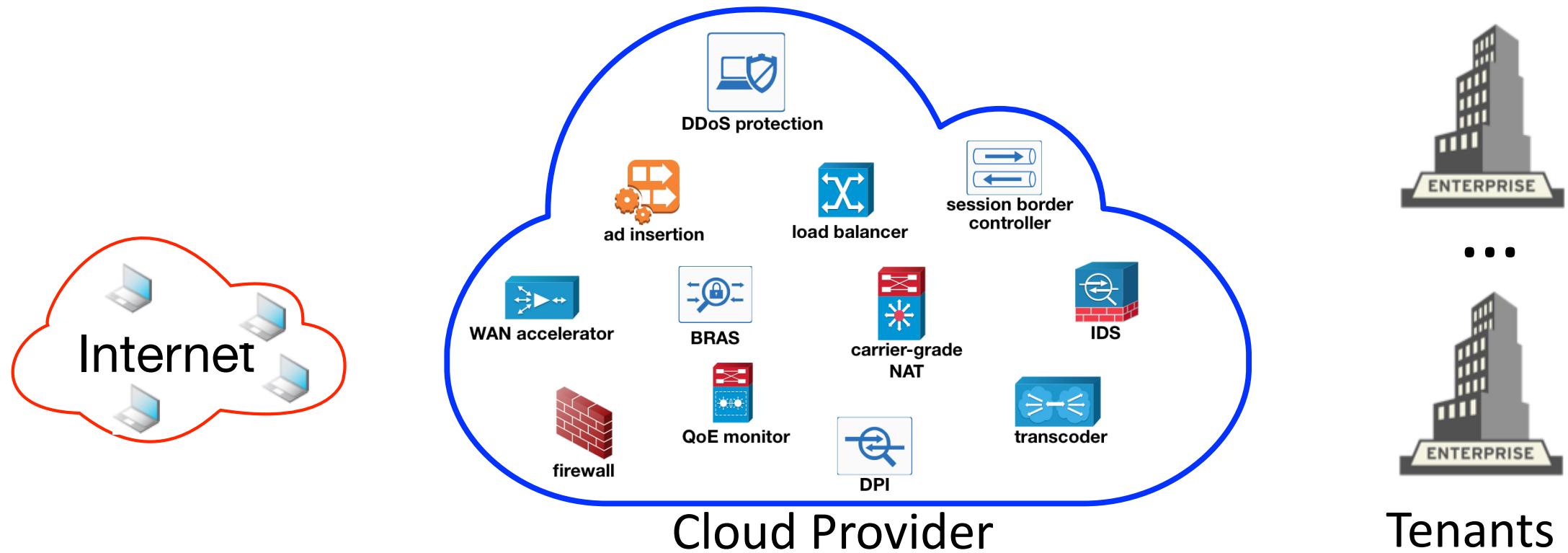
Outsourcing VNFs to the Cloud



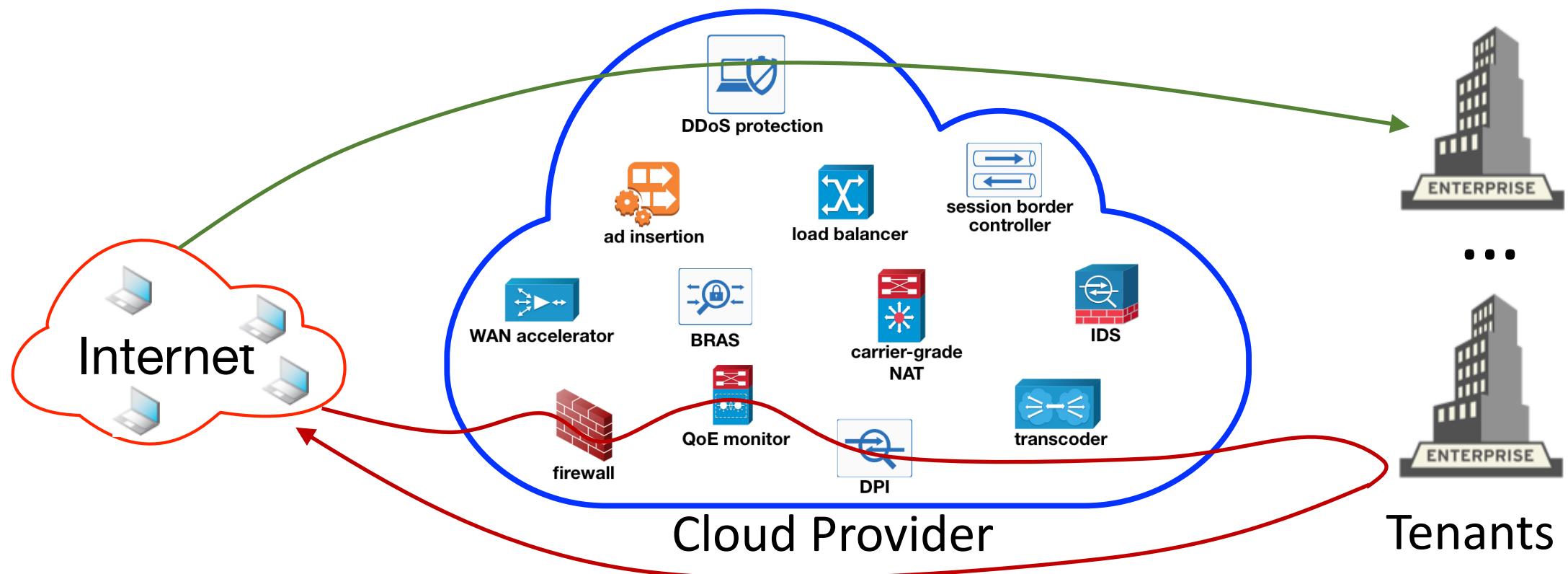
Outsourcing VNFs to the Cloud



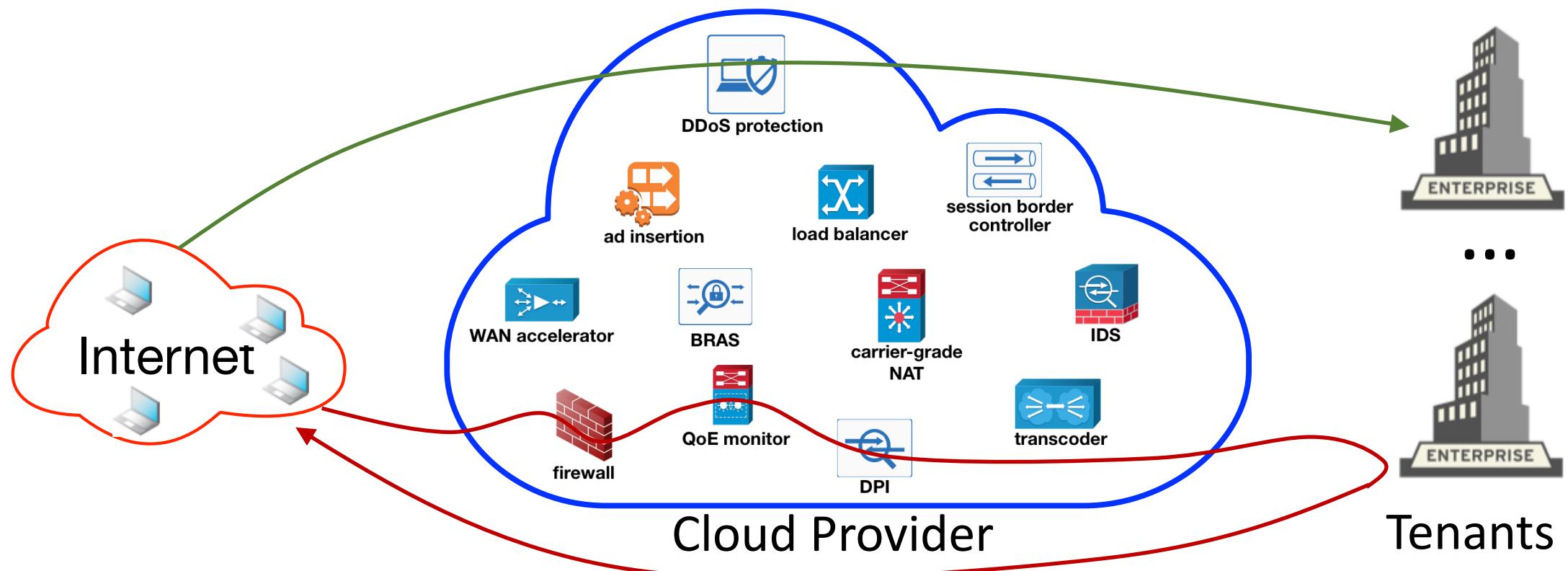
Outsourcing VNFs to the Cloud



Outsourcing VNFs to the Cloud



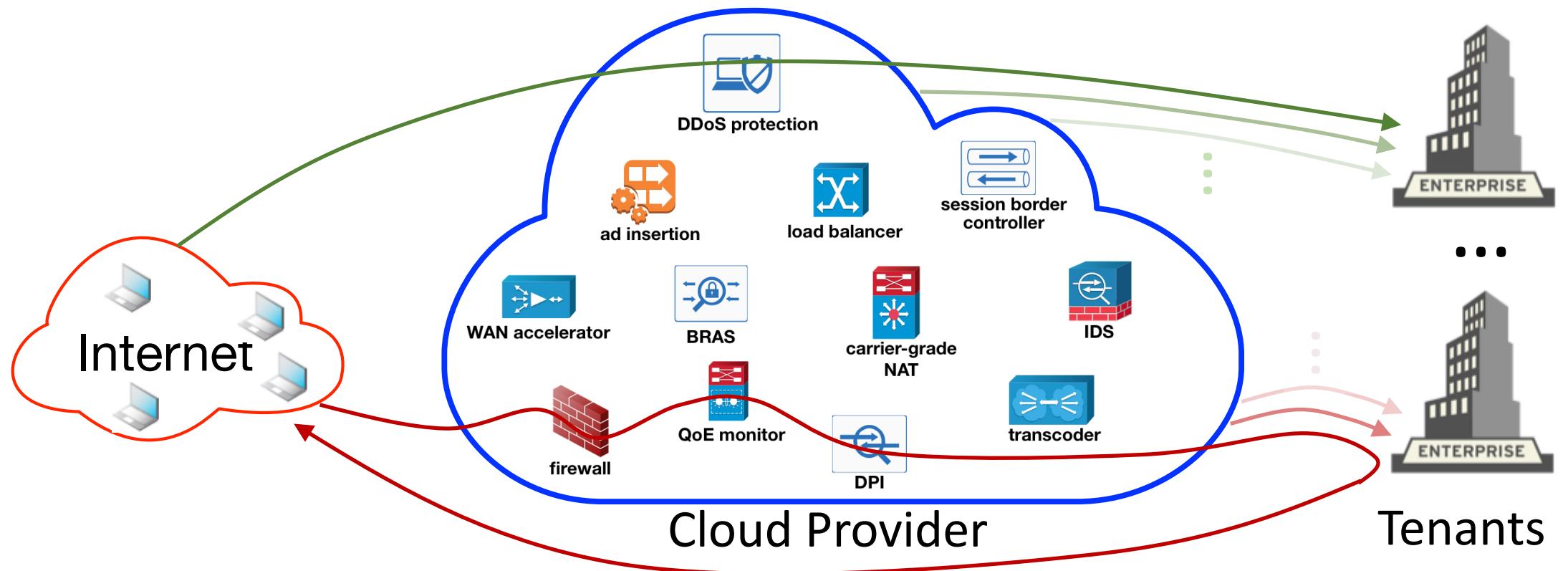
Outsourcing VNFs to the Cloud



Challenges of outsourcing VNFs

How can cloud providers achieve high **data center utilization**?

How can tenants **allocate and manage** their VNFs?



Our contributions: API and algorithm

How can cloud providers achieve
high **data center utilization**?

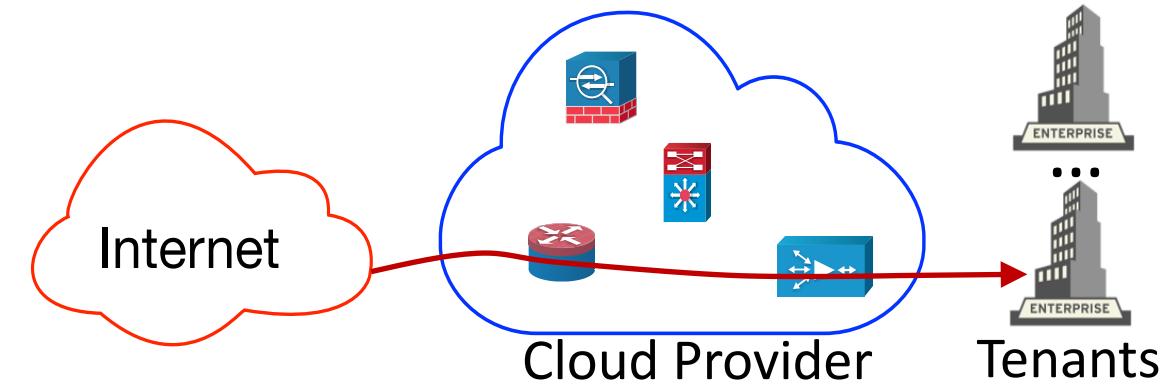
How can tenants **allocate and
manage** their VNFs?

Our contributions: API and algorithm

How can cloud providers achieve high **data center utilization**?

How can tenants **allocate and manage** their VNFs?

- API to allocate and manage VNFs
- Three **algorithms**
 - implement the API, and
 - achieve high data center utilization
- **Evaluation**
 - simulate: in data center scale with **1000+ servers**
 - **Daisy**: emulate chain management at rack-scale

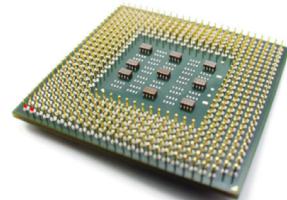


Systems Research

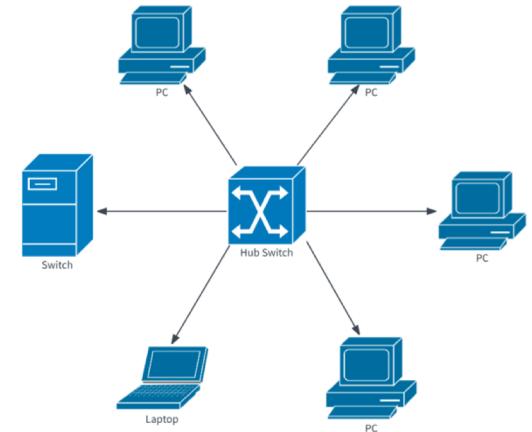
Storage



Compute



Networking

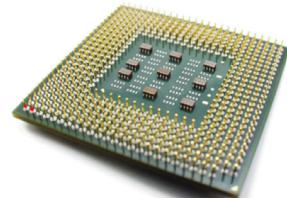


Systems Research

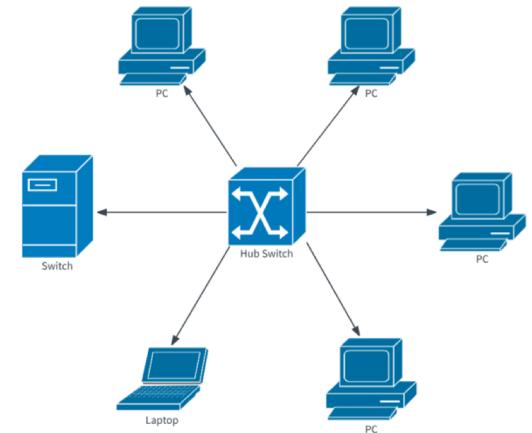
Storage



Compute



Networking



Systems Research: open problems

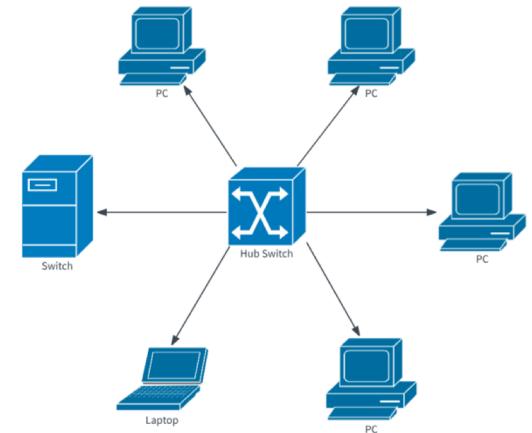
Storage



Compute

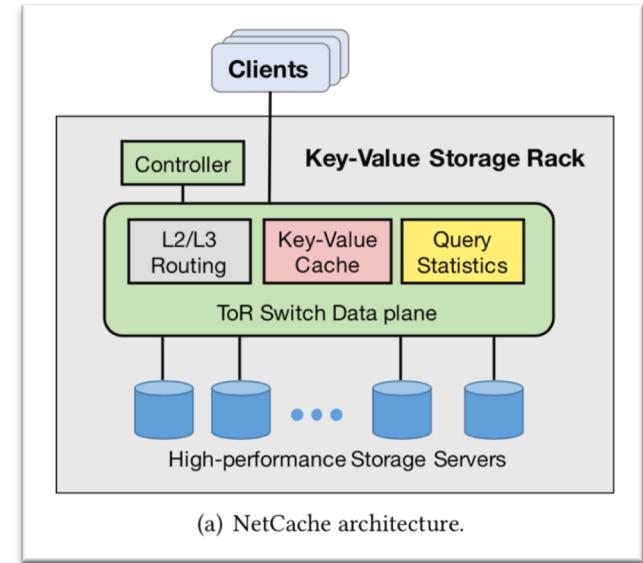


Networking



Open Problems: Networking

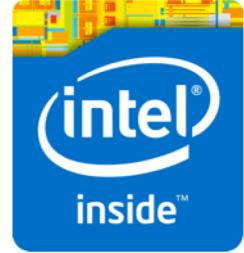
- In-network processing
 - How to **co-design network and application** for better app performance?
 - Jin et al., NetCache: Balancing Key-Value Stores with Fast In-Network Caching, SOSP'17
- How to **program** large networks?
 - Bosshart et al., P4: Programming Protocol-Independent Packet Processors, SIGCOMM CCR 2014
- How to make networking device more **robust**?
 - Zaostrovnykh et al., A Formally Verified NAT, SIGCOMM'17



(a) NetCache architecture.



Open Problems: Compute



- Isolation vs. performance
 - How to provide better **isolation** (security) without sacrificing a performance?
 - Lipp et al., Meltdown: Reading Kernel Memory from User Space, USENIX Security 2018
- Virtualization for **disaggregated** architecture
 - Shan et al, LegoOS: A Disseminated, Distributed OS for Hardware Resource Disaggregation, OSDI'18



Meltdown



Spectre

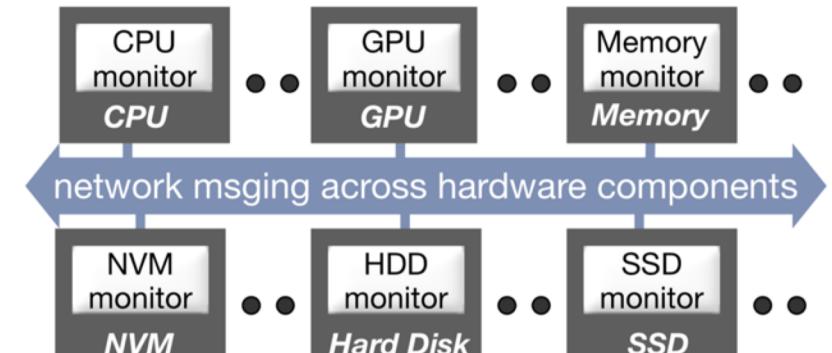
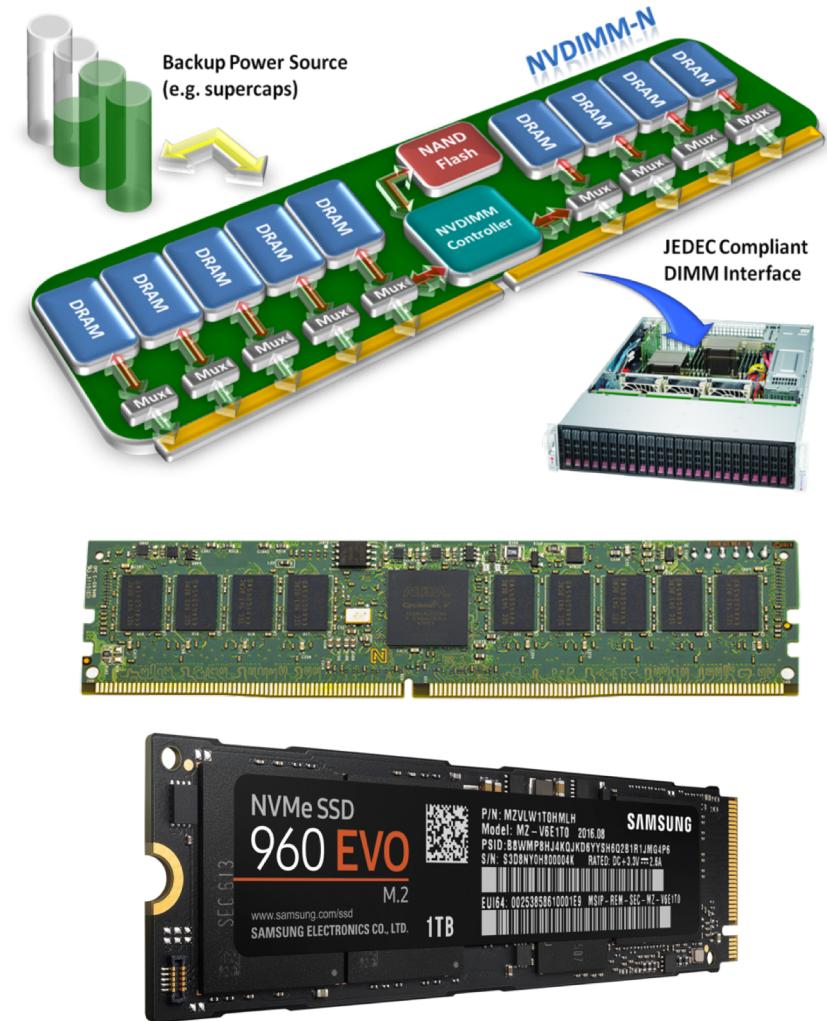


Figure 3: Splitkernel Architecture.

Open Problems: Storage

- New device on IO stack
 - How to leverage NVDIMM (a.k.a. non-volatile RAM) in data center applications?
 - $\text{RAM} < \text{NVDIMM}_{\text{latency, capacity}} < \text{SSD} (\approx 10x)$
 - $\text{RAM} < \text{NVDIMM}_{\text{IOPS}} < \text{SSD} (\approx 100x)$
 - Son et al, High-Performance Transaction Processing in Journaling File Systems, FAST'18
- CAP theorem (in distributed file systems)
 - Consistency, Availability, Partition tolerance
 - Alagappan et al., Protocol-Aware Recovery for Consensus-Based Storage, FAST'18



Andy Rudoff, Persistent Memory: The Value to HPC and the Challenges, Intel Corporation, 2017,
<https://passlab.github.io/mchpc/mchpc2017Proceedings/MCHPC17-AndyRudoff-presentation.pdf>
CAP theorem, https://en.wikipedia.org/wiki/CAP_theorem

Disclaimer: Research is Broad (even within CS)

Systems
(Compute, Storage, Networking)

Disclaimer: Research is Broad (even within CS)

Algorithms

Systems

(Compute, Storage, Networking)

DB

Security

Disclaimer: Research is Broad (even within CS)

Game theory

Algorithms

Programming Languages

Bioinformatics

Systems

Information Theory

(Compute, Storage, Networking)

DB

Security

Machine Learning

Human-Computer Interaction

Robotics

...

Takeaways and conclusion

- Systems research
 - Storage: distributed file systems (GFS)
 - Compute: virtualization (Xen hypervisor)
 - Networking: Software-defined networking (OpenFlow)
 - My research: Network Functions at data center scale
- Open problems
 - Storage: NVDIMM and CAP theorem
 - Compute: Isolation vs. performance, virtualization for disaggregated arch
 - Networking: in-network processing, programmability, robustness
- Exciting times ahead!

Thank you!

Backup Slides

Top Systems conferences

- Compute
 - [SOSP](#), [OSDI](#), [HotOS](#), [EuroSys](#), USENIX [ATC](#)
- Storage
 - [FAST](#), [EuroSys](#), USENIX [ATC](#)
- Networking
 - [SIGCOMM](#), [NSDI](#), [CoNEXT](#), [HotNets](#), [EuroSys](#), USENIX [ATC](#)
- Security
 - USENIX [Security](#), IEEE [Symposium on Security and Privacy](#)

Why at graduate school?

- Freedom to explore
- A collaborative environment
- Publication is encouraged



Should I apply to grad school?

- A Survival Guide to a PhD, by Andrej Karpathy (Director of AI at Tesla),
<http://karpathy.github.io/2016/09/07/phd/>
- The Ph.D. Grind: A Ph.D. Student Memoir, by Philip J. Guo (Prof. at UCSD), <http://pgbovine.net/PhD-memoir/pguo-PhD-grind.pdf>
- Good SoP sample <http://cwfletcher.net/Pages/SoP.php>

How to apply to grad school?

- See requirements from individual departments
 - University websites have **prospective students** or **admissions** section
 - UBC CS admissions <https://www.cs.ubc.ca/students/grad/admissions>
- **Maximize** acceptance probability by
 - **Independent** research: side projects, extra curriculum activities
 - **Publications**: conference papers, posters, reports, etc.
 - **Better marks**: language certificate, GPA, awards
 - **Strong** Statement of Purpose (SoP)
 - Highly **supportive** recommendation letters

How to decide which school?

- Many **factors**
 - **Academic**: research topic, supervisor match, university ranking, etc.
 - **Non-academic**: budget, family, country, visa friendliness, immigration, etc.
- Unclear which factors have more weight
 - No one-size-fits-all, highly **subjective**
- Rule of thumb: maximize ***happiness***
 - Happiness is also subjective: make ***your*** choice
- **Personal experience**: research topic and supervisor weight the most

How to find scholarships/funding?

- Rule of thumb: **Google** is your best friend
 - Look at university websites, government scholarship, commercial sponsors
 - Generally, universities have *awards* section
 - <https://students.ubc.ca/enrolment/finances/awards-scholarships-bursaries>
 - Ask from your home university
 - Awards department, international collaboration dept., study abroad dept., etc.
- Canadian scholarships
 - <http://www.scholarships-bourses.gc.ca/scholarships-bourses/>
 - <https://www.canada.ca/en/services/finance/educationfunding/scholarships.html>
- Korean scholarships
 - <http://www.niiied.go.kr/eng/main.do>
 - http://www.studyinkorea.go.kr/en/sub/gks/allnew_invite.do