# Towards Predictable Datacenter Networks

Hitesh Ballani, Paolo Costa,
Thomas Karagiannis and Ant Rowstron

Microsoft Research, Cambridge

# This talk is about …

## Guaranteeing network performance for tenants in multi-tenant datacenters

### Multi-tenant datacenters

- Datacenters with multiple (possibly competing) ***tenants***
- Private datacenters
  - Run by organizations like Facebook, Intel, etc.
  - **Tenants**: Product groups and applications
- Cloud datacenters
  - Amazon EC2, Microsoft Azure, Rackspace, etc.
  - **Tenants**: Users renting virtual machines

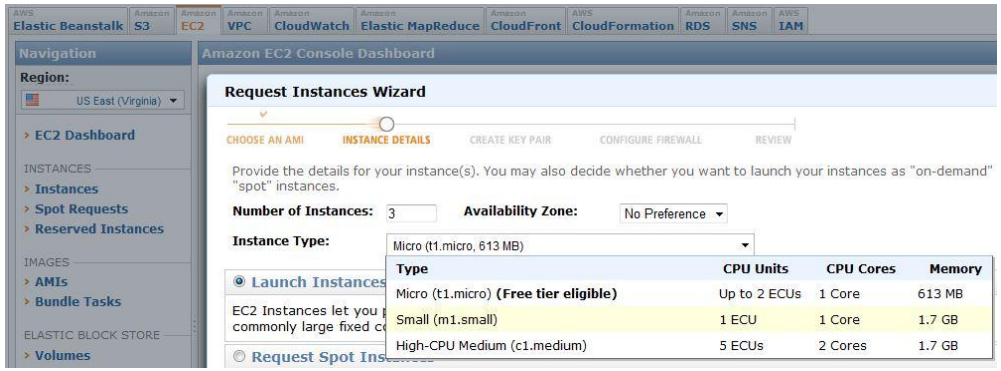
# Cloud datacenters 101

## Simple interface: Tenants ask for a set of VMs

Tenant

***Request VMs***

Amazon EC2 Interface



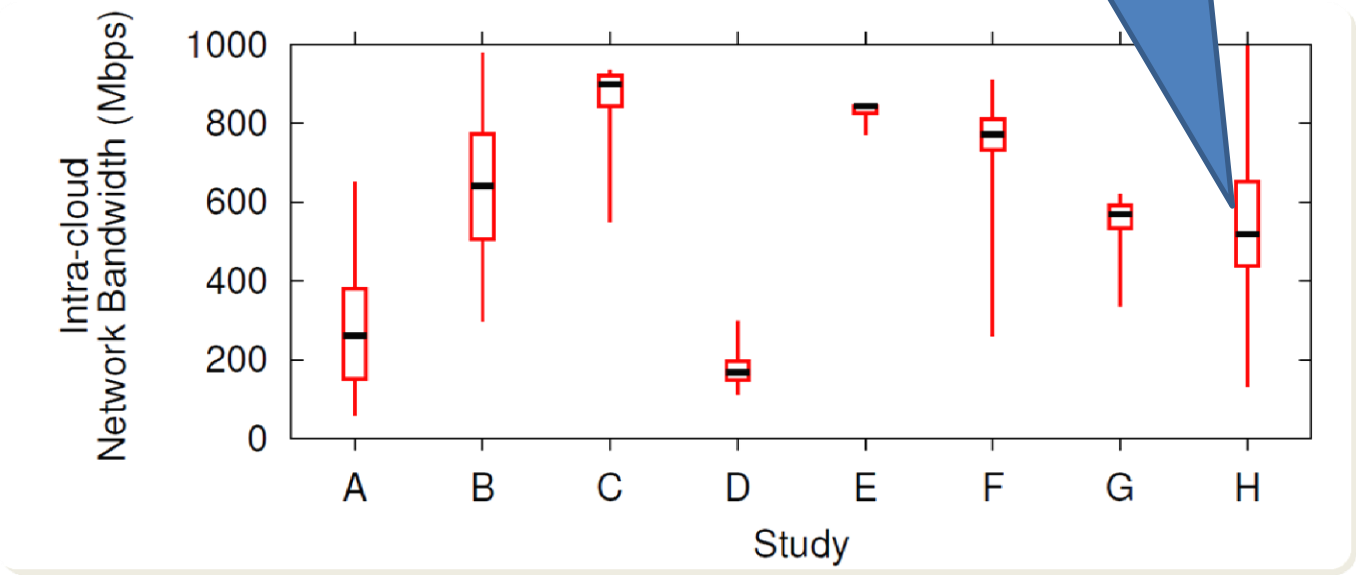

- Charging is per-VM, per-hour
  - Amazon EC2 small instances: $0.085/hour
  - No (intra-cloud) network cost

**Network performance is not guaranteed**

Bandwidth between a tenant's VMs depends on their placement, network load, protocols used, etc.

# Performance variability i

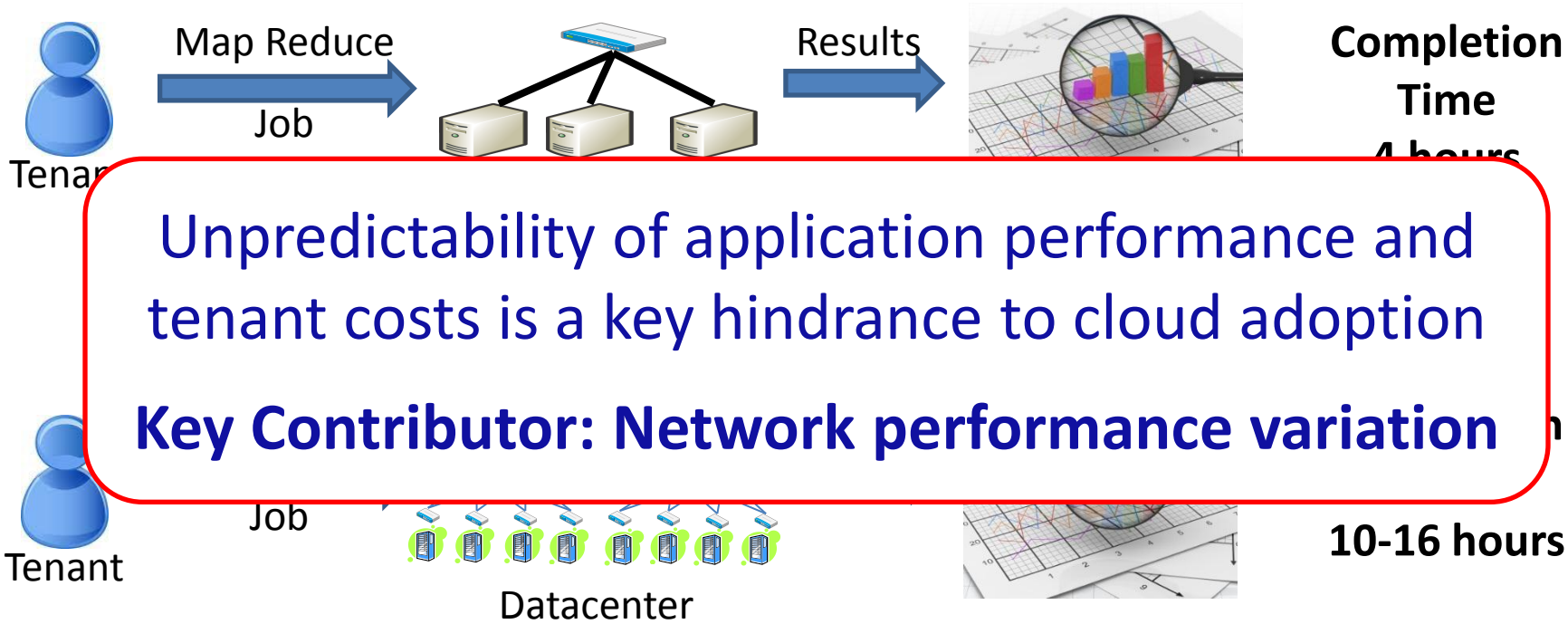
Up to 5x variability




| Study | Study | Provider | Duration |
|---|---|---|---|
| A | [Giurgui'10] | Amazon EC2 | n/a |
| B | [Schad'10] | Amazon EC2 | 31 days |
| C/D/E | [Li'10] | (Azure, EC2, Rackspace) | 1 day |
| F/G | [Yu'10] | Amazon EC2 | 1 day |
| H | [Mangot'09] | Amazon EC2 | 1 day |

# Network performance can vary ... so what?

Data analytics on an isolated cluster




Unpredictability of application performance and tenant costs is a key hindrance to cloud adoption

**Key Contributor: Network performance variation**

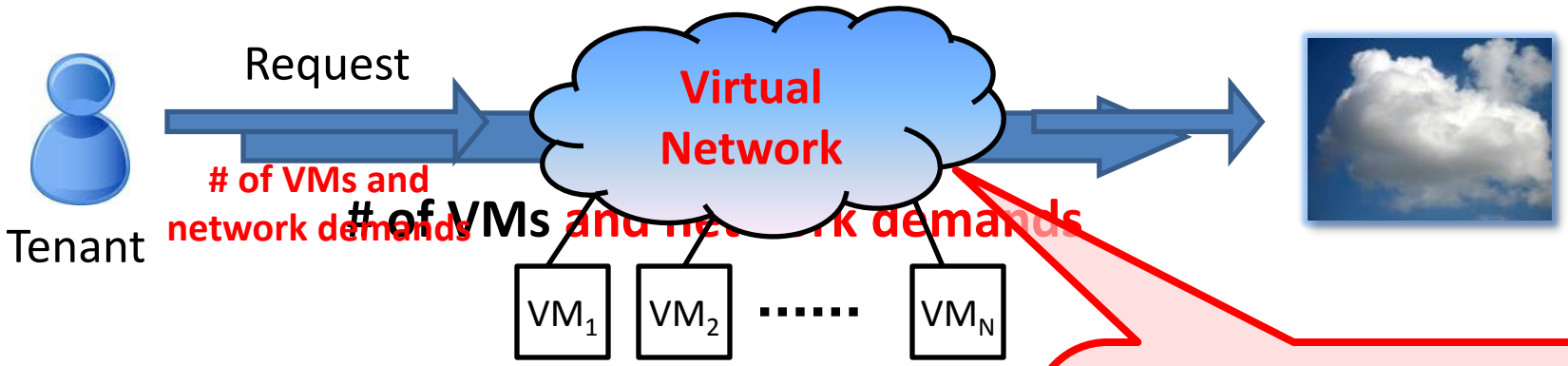
**Variable tenant costs**

Expected cost (based on 4 hour completion time) = $100

**Actual cost = $250-400**

# Predictable datacenter networks

Extend the tenant-provider interface to account for the network




**Key Idea**: Tenants are offered a virtual network with bandwidth guarantees

This decouples tenant performance from provider infrastructure

***Contributions-***

**Virtual network** abstractions

- To capture tenant network demands

**Oktopus:** Proof of concept system

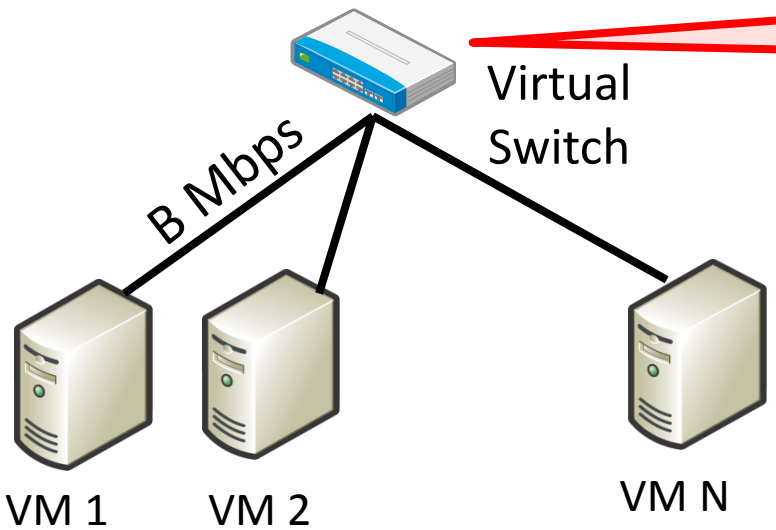
- Implements virtual networks in multi-tenant datacenters
- Can be incrementally deployed **today!**

# Talk Outline

- Introduction
- Virtual network abstractions
- Oktopus
  - Allocating virtual networks
  - Enforcing virtual networks
- Evaluation

# Abstraction 1: Virtual Cluster (*VC*)

**Motivation**: In enterprises, tenants run applications on dedicated Ethernet clusters




Total bandwidth
= N * B

**Request <N, B>**

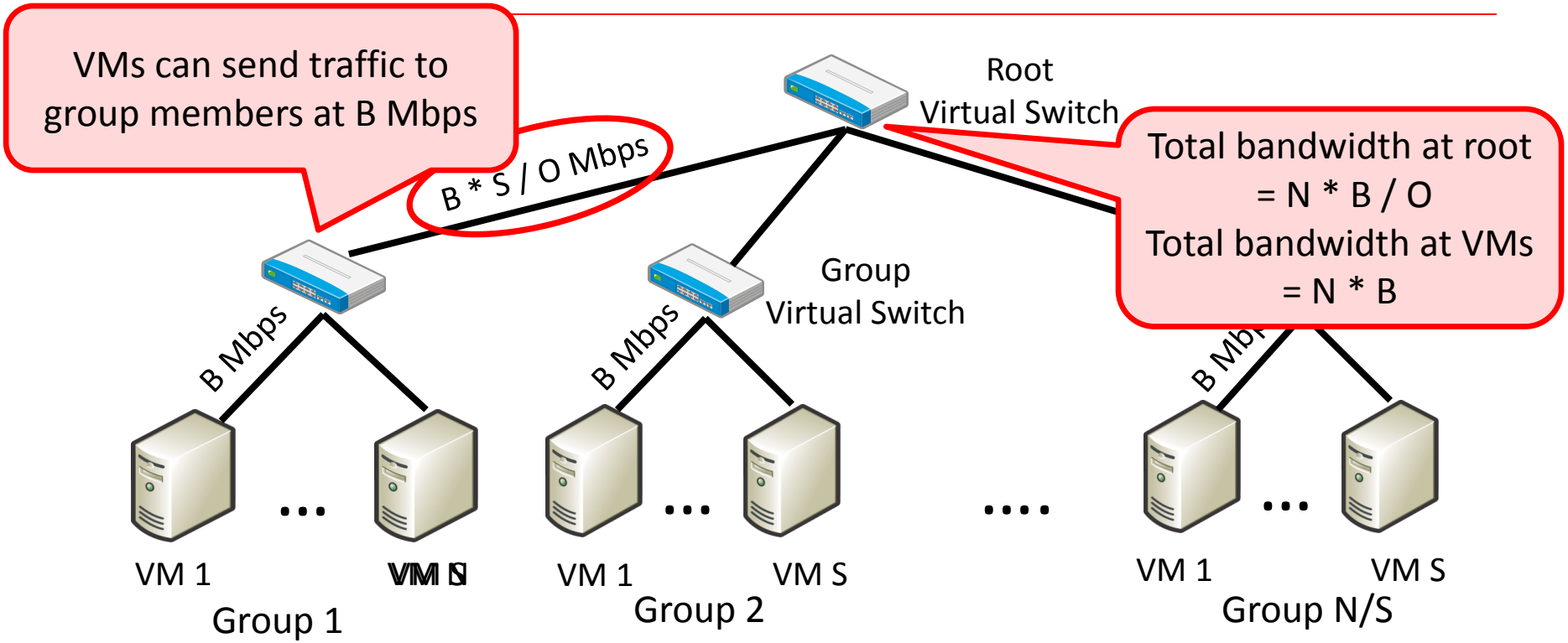
*N VMs. Each VM can send and receive at B Mbps*

**Tenants get a network with no oversubscription**

✓ Suitable for data-intensive apps. (MapReduce, BLAST)

✗ Moderate provider flexibility

# Abstraction 2: Virtual Oversubscribed Cluster (*VOC*)




**Request <N, B, S, O>**

**Motivation**: Many applications moving to the cloud have

**VOC capitalizes on tenant communication patterns**

✓ Suitable for typical applications (*though not all*)

✓ Improved provider flexibility

# Talk Outline

- Introduction
- Virtual network abstractions
- Oktopus
  - Allocating virtual networks
  - Enforcing virtual networks
- Evaluation

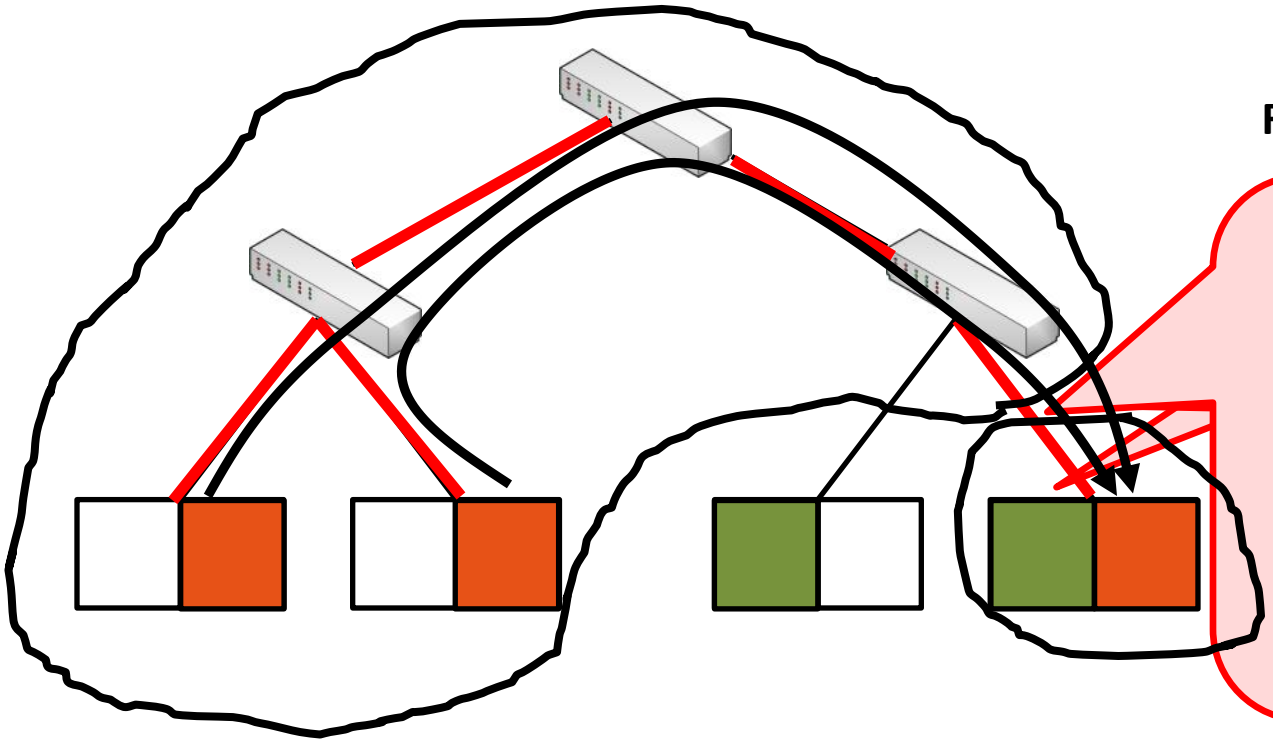
# Oktopus

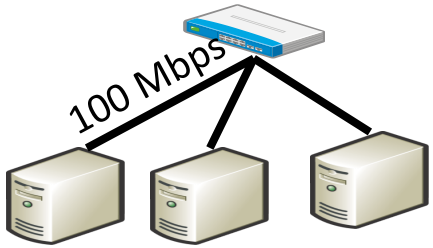
## Offers virtual networks to tenants in datacenters

### Two main components

- Management plane: ***Allocation of tenant requests***
  - Allocates tenant requests to physical infrastructure
  - Accounts for tenant network bandwidth requirements
- Data plane: ***Enforcement of virtual networks***
  - Enforces tenant bandwidth requirements
  - Achieved through rate limiting at end hosts

# Allocating Virtual Clusters






**Request : <3 VMs, 100 Mbps>**

Max Sending Rate = 2*100 = 200

Max Receive Rate = 1*100 = 100

**B/W needed on link** = Min (200, 100) = **100Mbps**

For a virtual cluster <***N,B***>, bandwidth needed on a link that connects ***m*** VMs on one side and ***N-m*** on the other side is ***Min(m, N-m) * B***

**How to find a valid allocation?**

An allocation must ensure that the VMs assigned to a link divide the virtual cluster into two parts.

Allocate VMs to physical machines

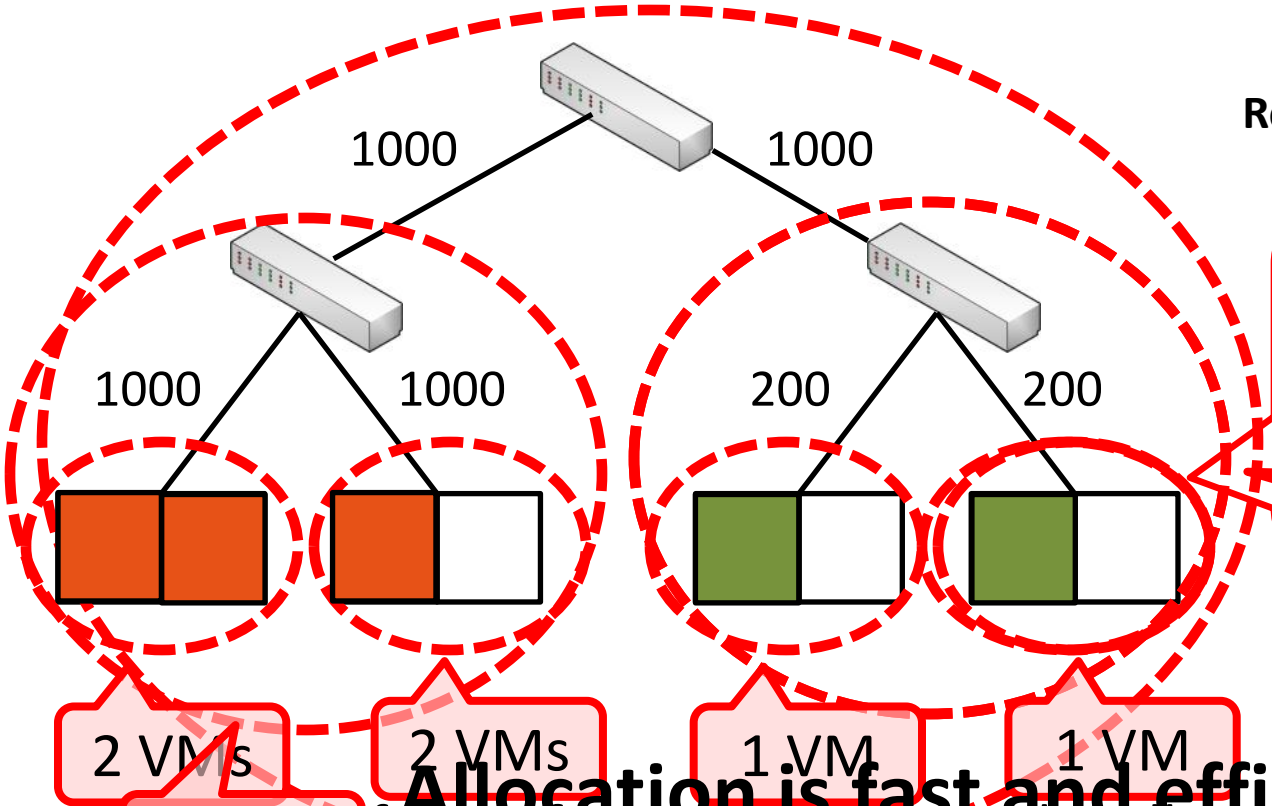
Allocate bandwidth

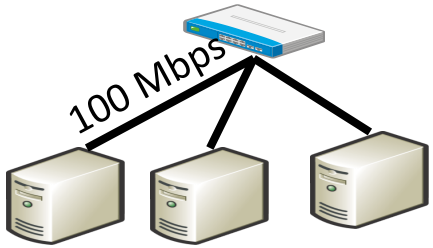
Allocate a tenant's VMs to physical machines

For a ***valid allocation***, all traffic from the left to right part of the cluster can traverse the link.

Bandwidth needed <= Link's Residual Bandwidth

# Allocation Algorithm




**Request : <3 VMs, 100 Mbps>**

**Solution**

At most 1 VM for this tenant can be allocated here

2 VMs

3 VMs

**Allocation is fast and efficient**

Constraints for *# of VMs (Key intuition)* allocated to the machine-

1. VMs are allocated to the machine-
2. 3 VMs are requested
3. Enough b/w on outbound link ⇒ min *(m, 3-m)**100 <= 200

Packing VMs together motivated by the fact that datacenter networks are typically oversubscribed

Traverse up the hierarchy and determine the lowest level at which all 3 VMs can be allocated

Allocation can be extended for cases like *fault resiliency*, etc.

**Greedy allocation algorithm**

Validity conditions can be used to determine the number of VMs that can be allocated to any level of the datacenter, machines, racks and so on

# Talk Outline

- Introduction
- Virtual network abstractions
- Oktopus
  - Allocating virtual networks
  - Enforcing virtual networks
- Evaluation

# Enforcement in Oktopus: Key highlights

## Oktopus enforces virtual networks at end hosts

- Use egress rate limiters at end hosts
  - Implement on hypervisor/VMM

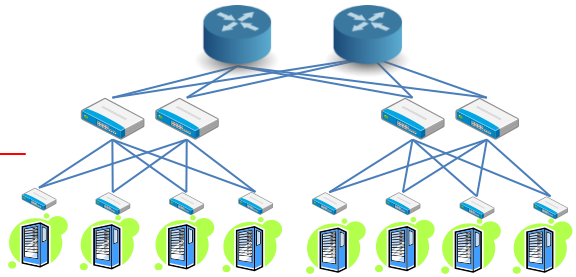
## Oktopus can be deployed *today*

- No changes to tenant applications
- No network support
- Tenants without virtual networks can be supported
  - Good for incremental roll out

# Talk Outline

- Introduction
- Virtual network abstractions
- Oktopus
  - Allocating virtual networks
  - Enforcing virtual networks
- Evaluation

# Datacenter Simulator



## Flow-based simulator

- 16,000 servers and 4 VMs/server ⇨ 64,000 VMs
- Three-tier network topology (10:1 oversubscription)

## Tenants submit requests for VMs and execute jobs

- Job: VMs process and shuffle data between each other

## **Baseline**: representative of today's setup

- Tenants simply ask for VMs
- VMs are allocated in a locality-aware fashion

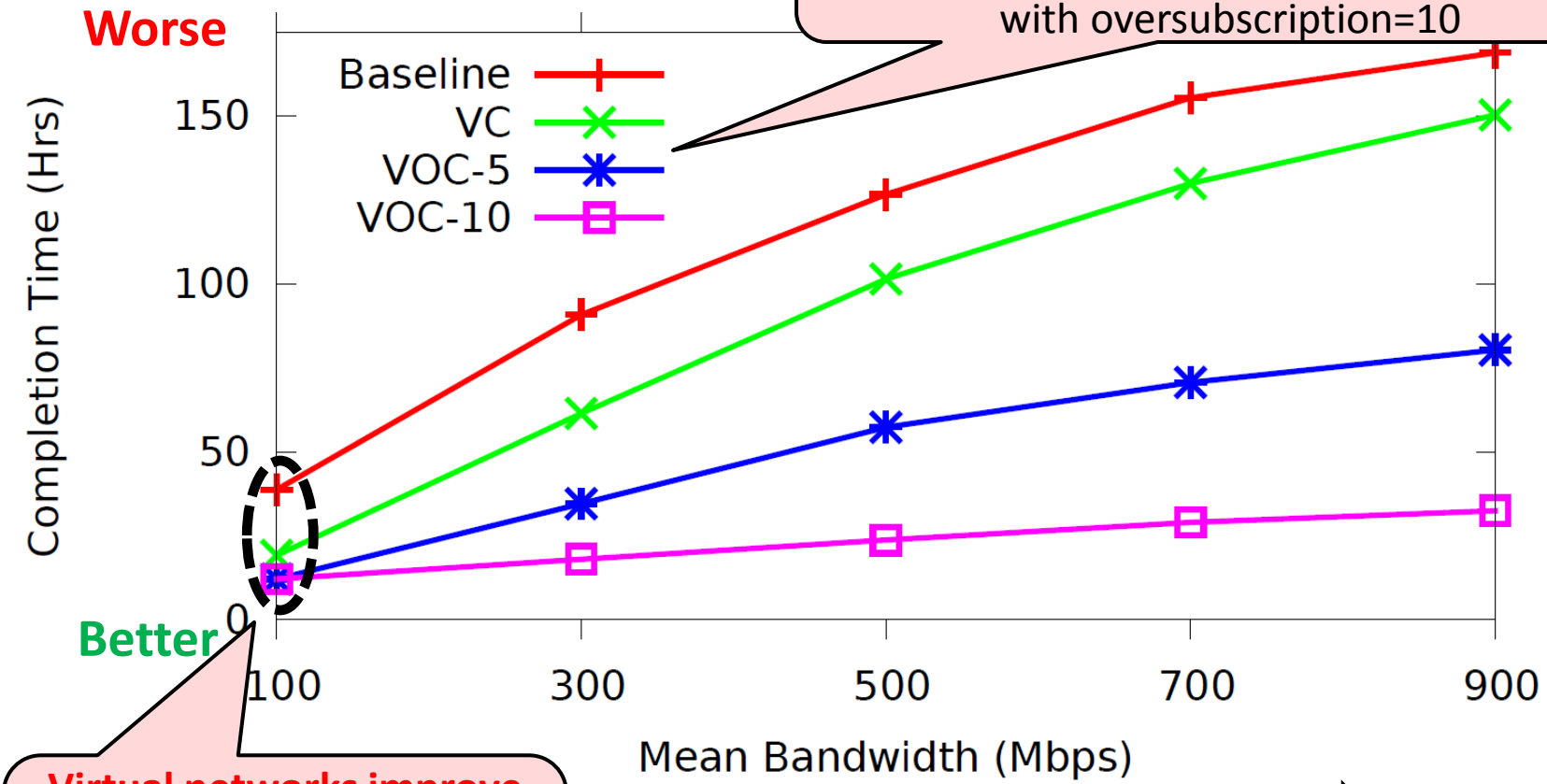
## Virtual network request

- Tenants ask for Virtual Cluster (VC) or Virtual Oversubscribed Cluster (VOC)

# Private datacenters

**VC** is Virtual Cluster

**VOC-10** is Virtual Oversubscribed Cluster with oversubscription=10



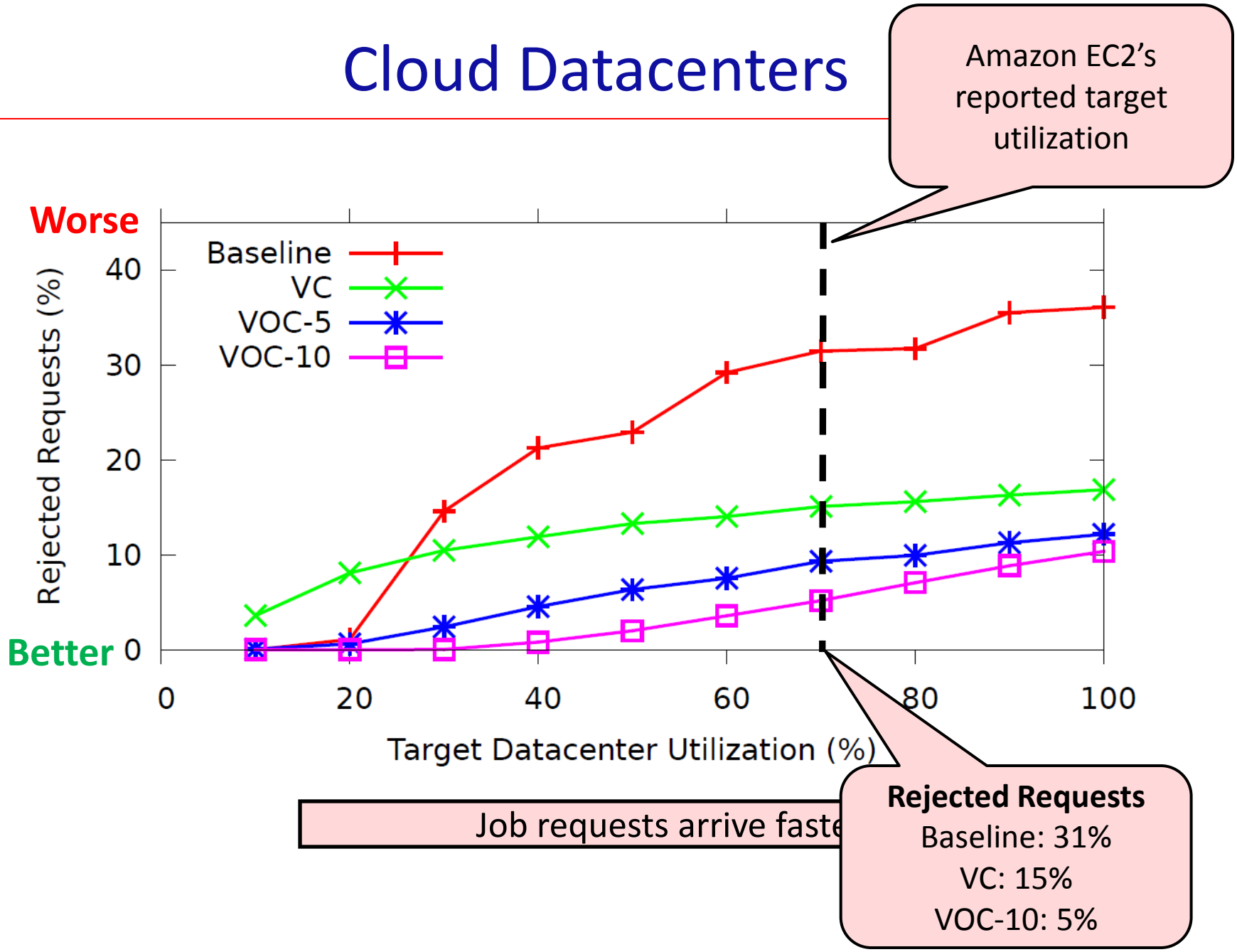

**Virtual networks improve completion time**

VC: 50% of Baseline

VOC-10: 31% of Baseline

bs become more network intensive

# Cloud Datacenters



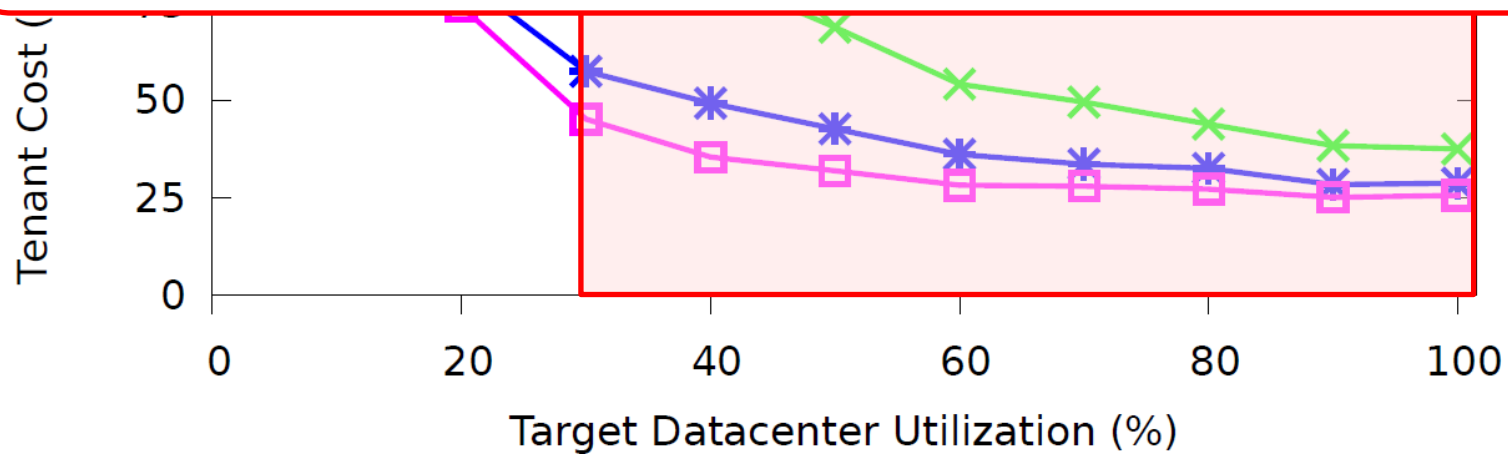
# Tenant Costs

What should tenants pay to ensure *provider revenue neutrality*, i.e. provider revenue remains the same with all approaches

Based on today's EC2 prices, i.e. $0.085/hour for each VM

**Provider revenue increases while tenants pay less**

At 70% target utilization, provider revenue increases by 20% and median tenant cost reduces by 42%



# Oktopus Deployment

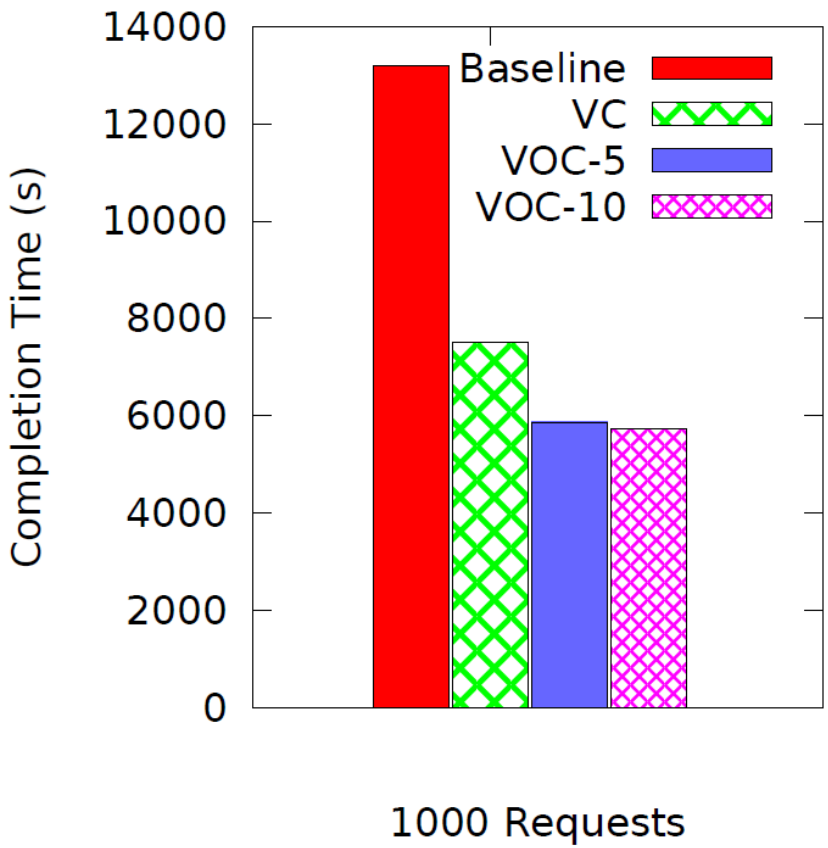
## Implementation scales well and imposes low overhead

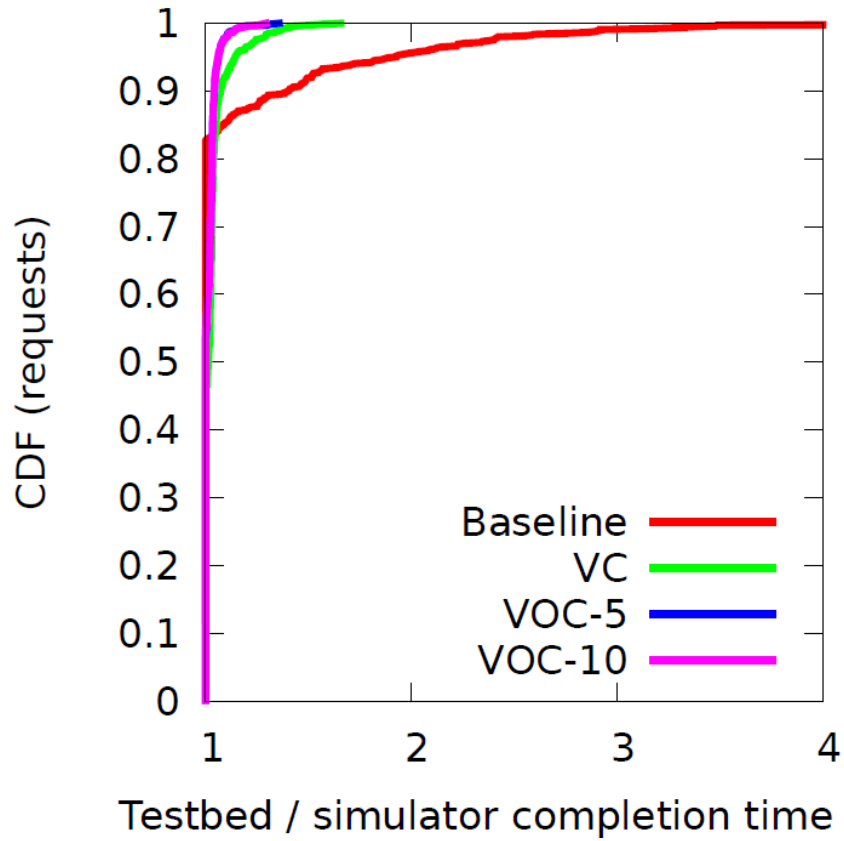
- Allocation of virtual networks is fast
  - In a datacenter with $10^5$ machines, median allocation time is 0.35ms
- Enforcement of virtual networks is cheap
  - Use Traffic Control API to enforce rate limits at end hosts

## Deployment on testbed with 25 end hosts

- End hosts arranged in five racks

# Oktopus Deployment




**Cross-validation of simulation results**

Completion time for jobs in the simulator matches that on the testbed

# Summary

## Proposal: Offer virtual networks to tenants

- Virtual network abstractions
  - Resemble physical networks in enterprises
  - Make transition easier for tenants

## Proof of concept: Oktopus

- Tenants get guaranteed network performance
- Sufficient multiplexing for providers
- Win-win: *tenants pay less, providers earn more!*

## How to determine tenant network demands?

Ongoing work: Map high-level goals (like desired completion time) to Oktopus abstractions

Thank you