End-to-end Performance Isolation through Virtual Datacenters

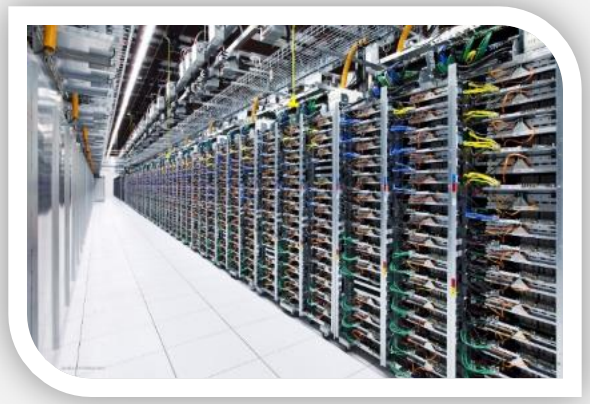
Sebastian Angel
UT Austin and NYU

Hitesh Ballani, Thomas Karagiannis, Greg O'Shea, Eno Thereska
Microsoft Research

Microsoft Research

Enterprise datacenter
(single tenant)

Enterprise datacenter
(single tenant)



Cloud datacenters
(multiple tenants)

Cloud datacenters offer services implemented by appliances

- Persistent storage

Key-value store

Blockstore

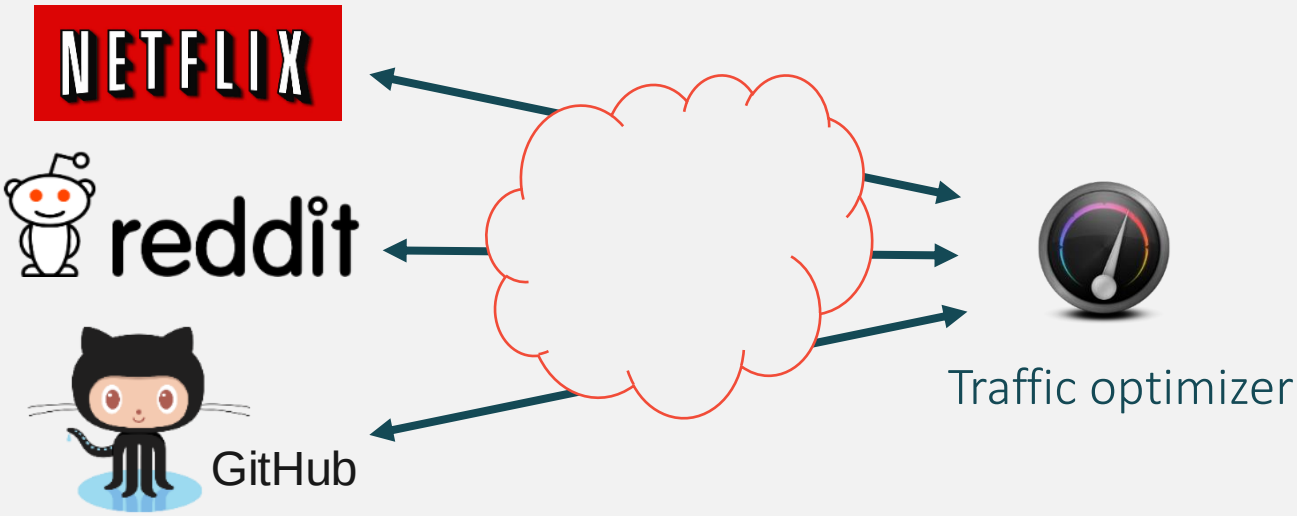
- Middleboxes

Load balancer

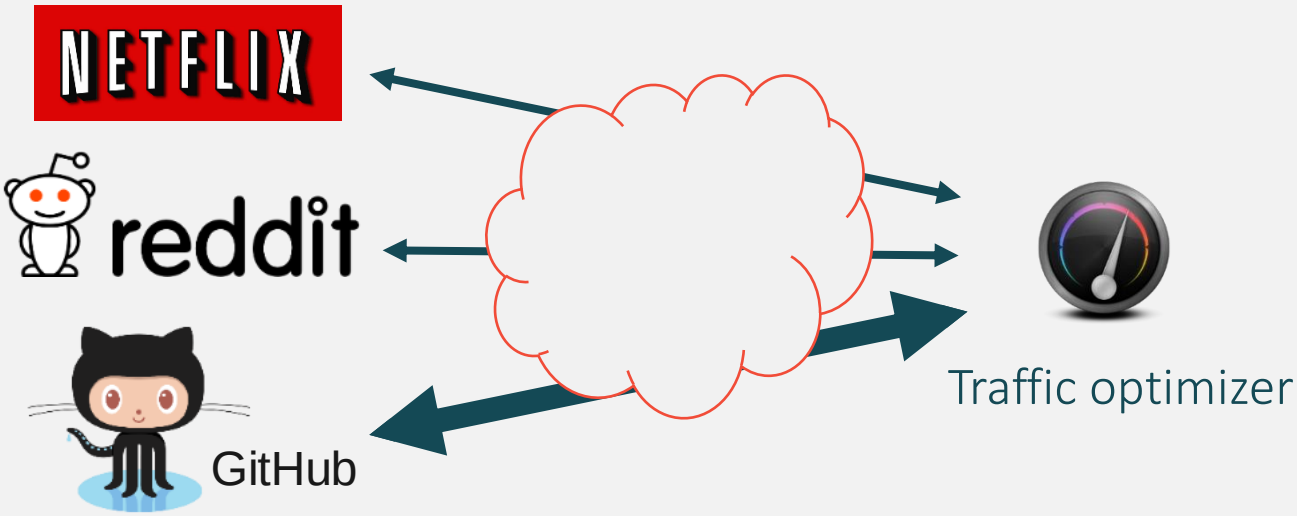
Encryption device

Traffic optimizer

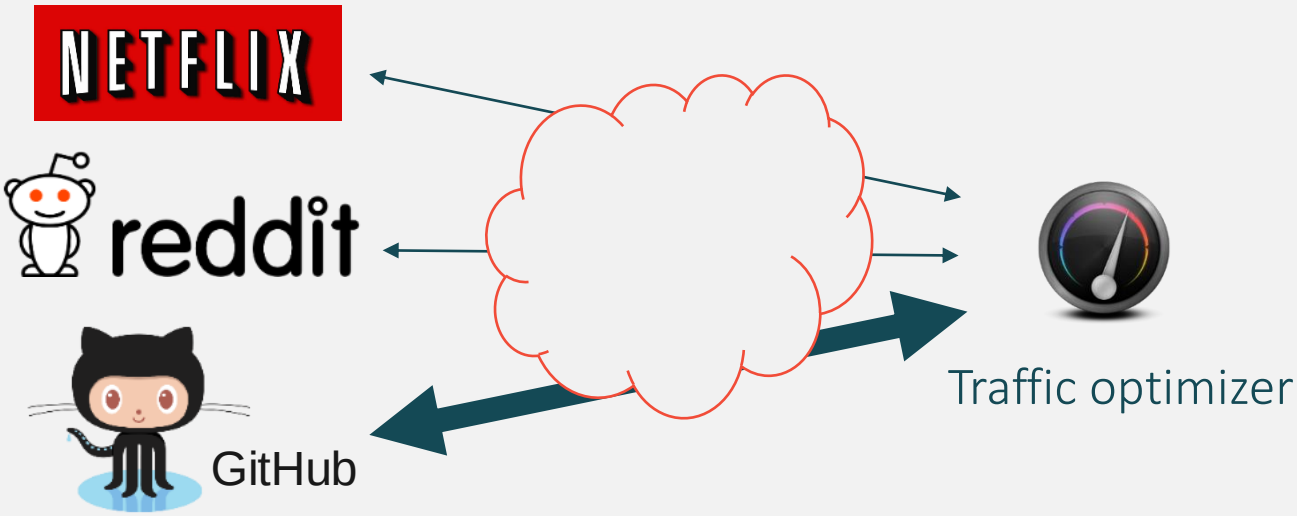
Appliances (and the network) are shared among tenants



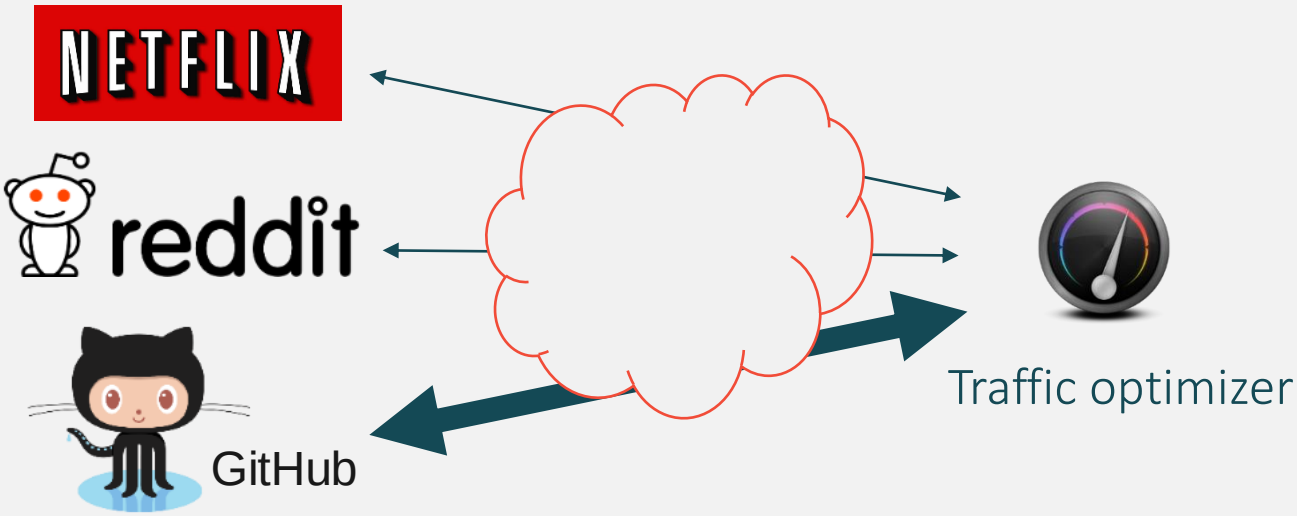
Appliances (and the network) are shared among tenants



Appliances (and the network) are shared among tenants

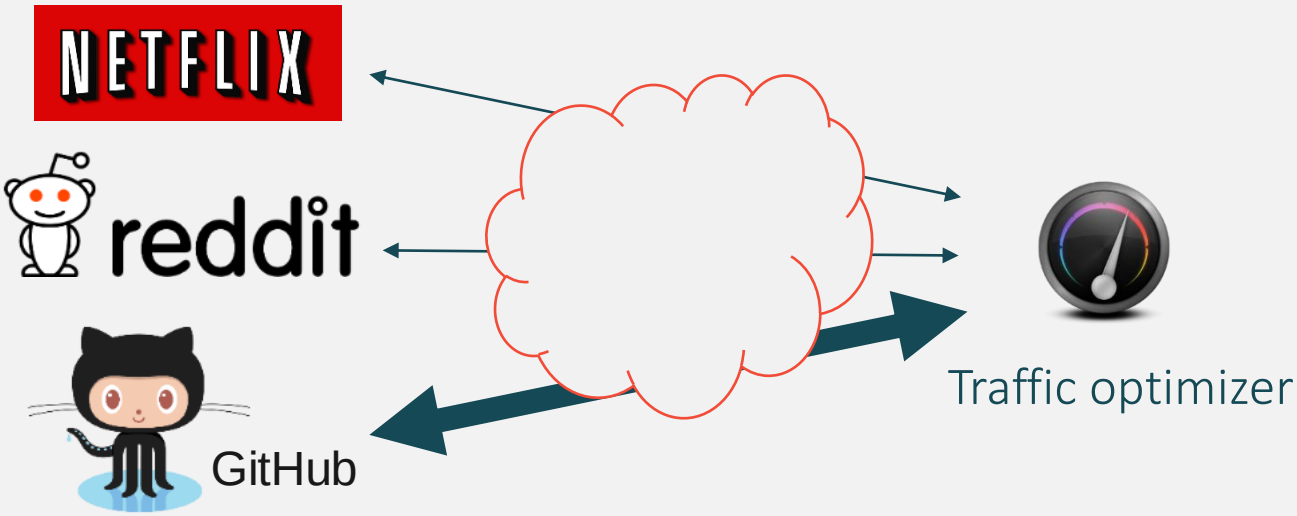


Appliances (and the network) are shared among tenants



Network and appliance throughput varies by up to 5X in datacenters due to contention at shared resources [SIGCOMM'11]

Appliances (and the network) are shared among tenants



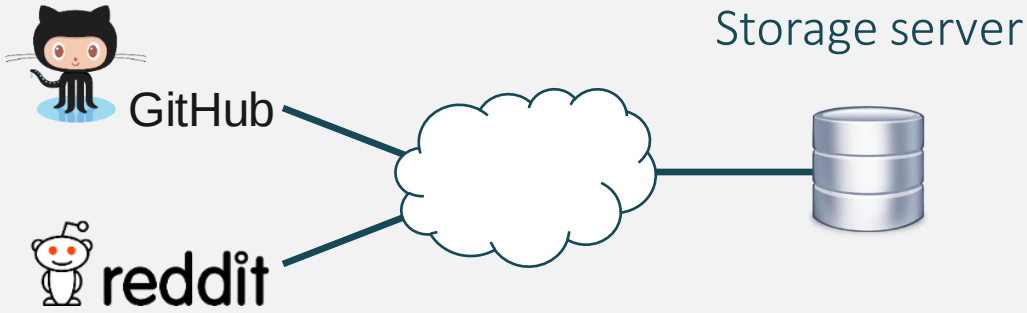
Network and appliance throughput varies by up to 5X in datacenters due to contention at shared resources [SIGCOMM'11]

Applications experience degraded and unpredictable performance

Tenants should get end-to-end guarantees

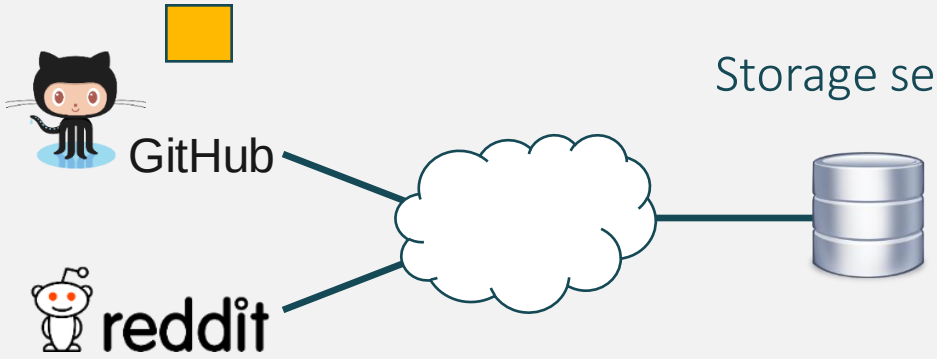
- Isolate tenants across the network
  - Oktopus [SIGCOMM'11], ElasticSwitch [SIGCOMM'13]
- Isolate tenants at appliances
  - DRFQ [SIGCOMM'12], Pisces [OSDI'12], IOFlow [SOSP'13]

Enforcing per-resource guarantees is not sufficient



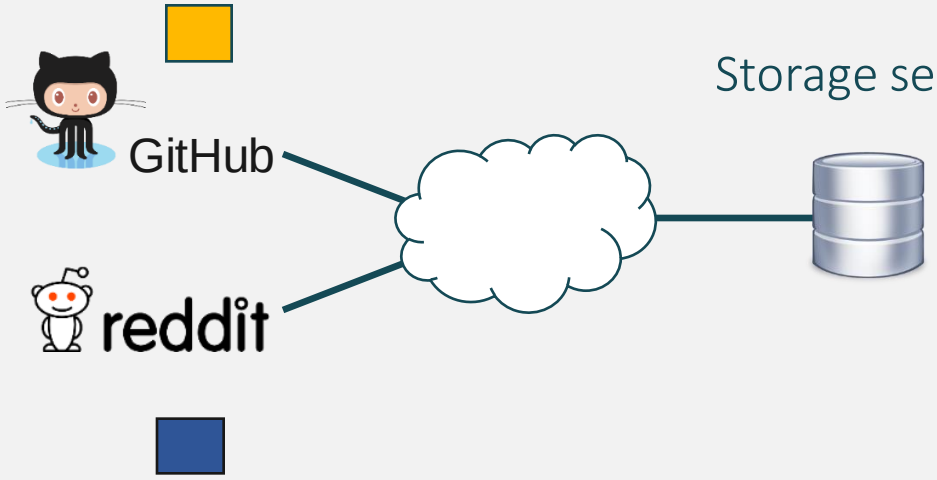
Enforcing per-resource guarantees is not sufficient

Large Read request (request itself is a tiny header)



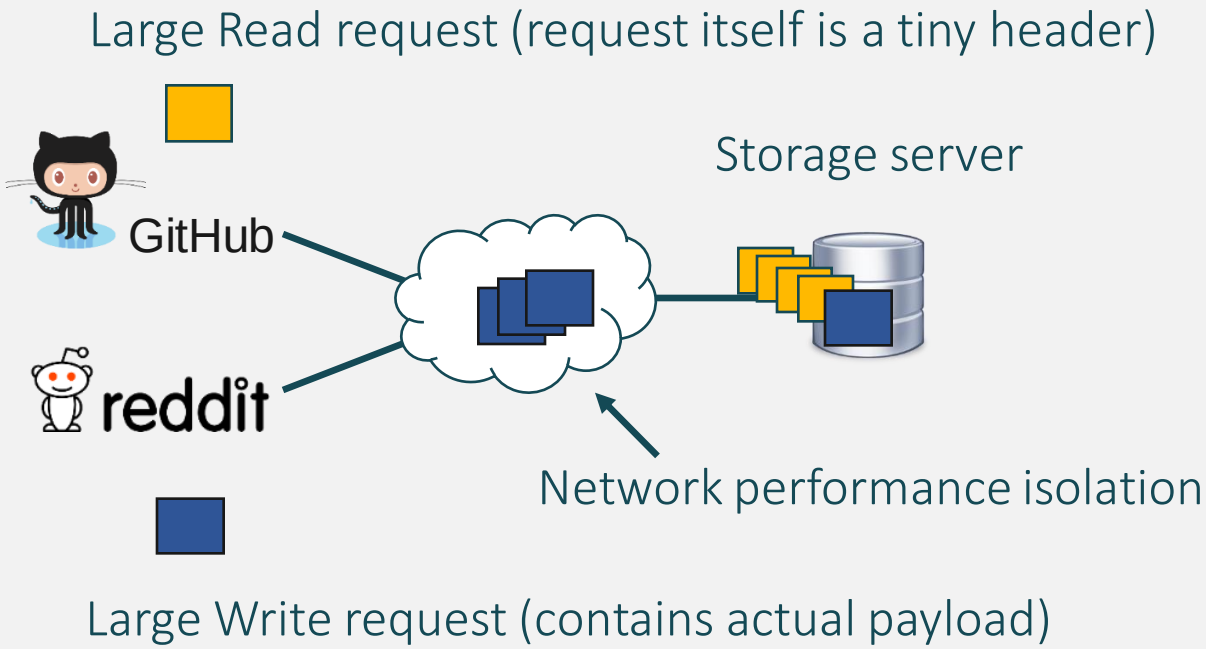
Enforcing per-resource guarantees is not sufficient

Large Read request (request itself is a tiny header)

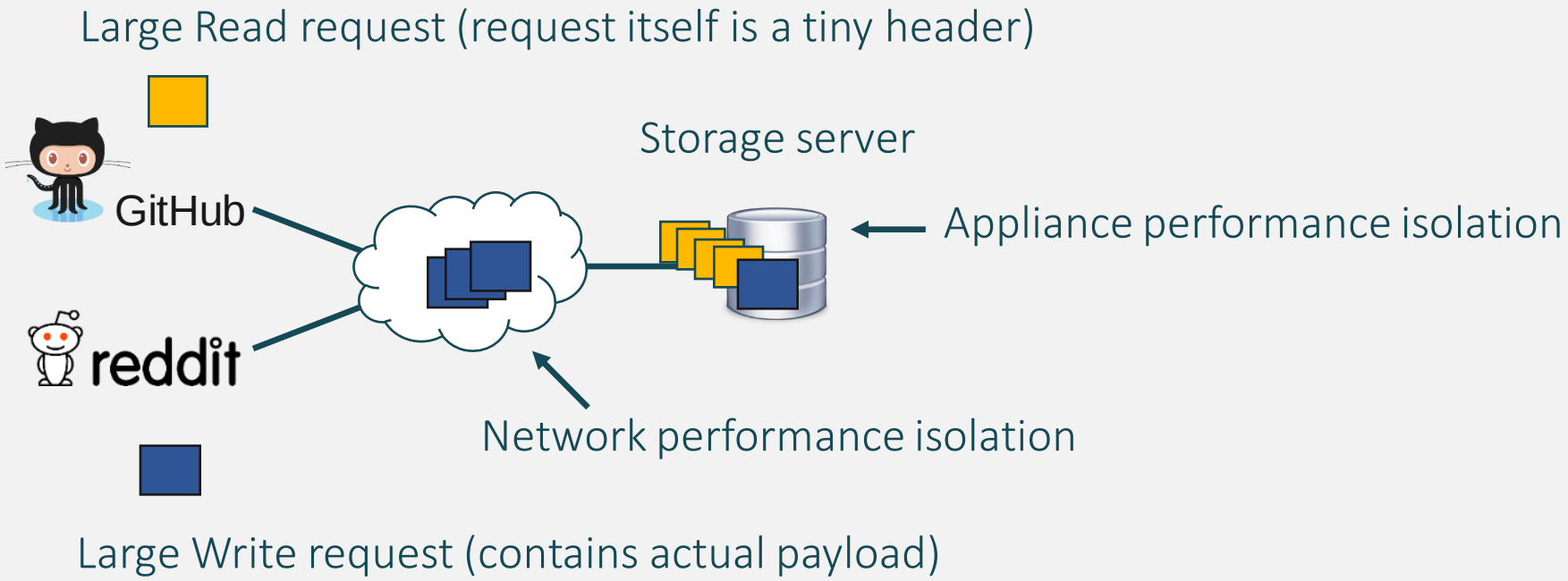


Large Write request (contains actual payload)

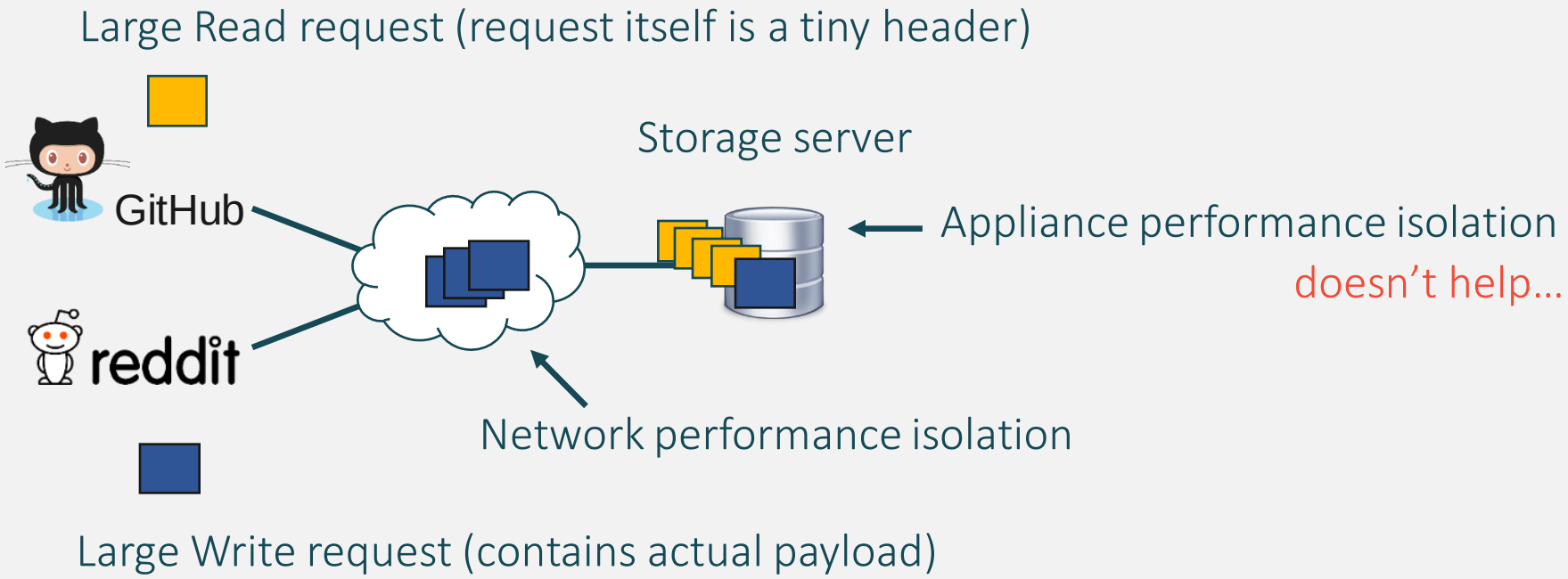
Enforcing per-resource guarantees is not sufficient



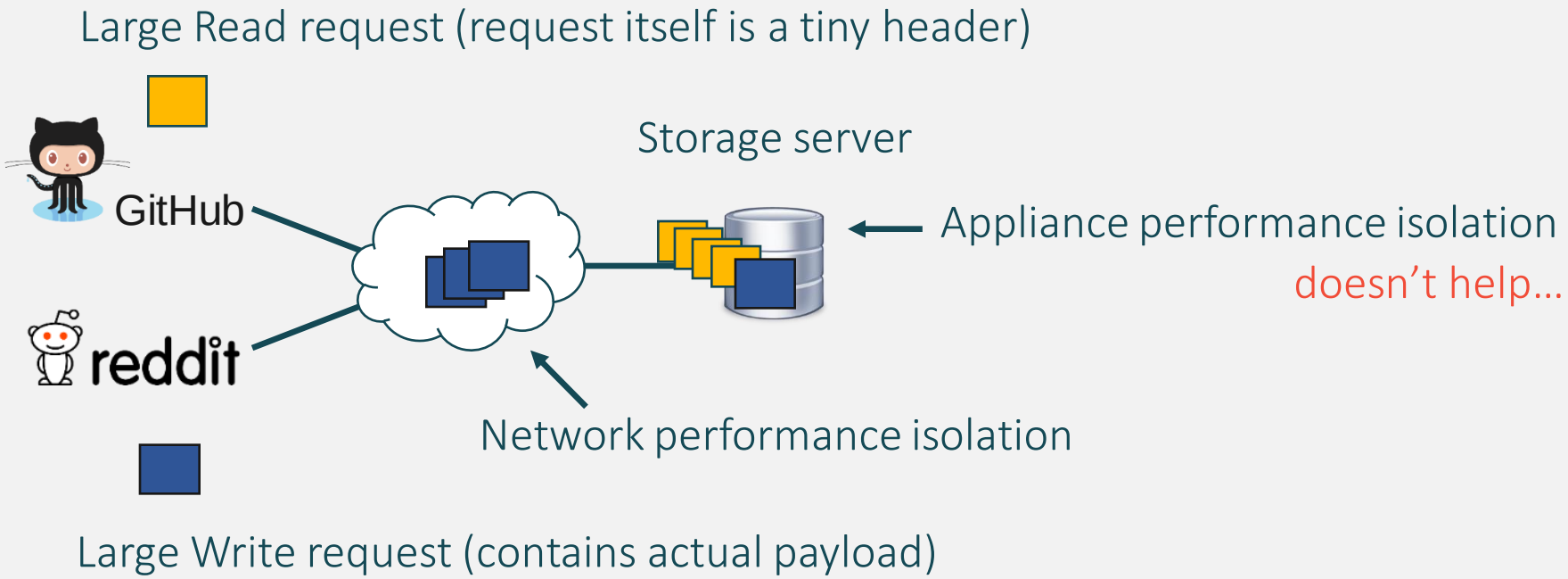
Enforcing per-resource guarantees is not sufficient



Enforcing per-resource guarantees is not sufficient



Enforcing per-resource guarantees is not sufficient

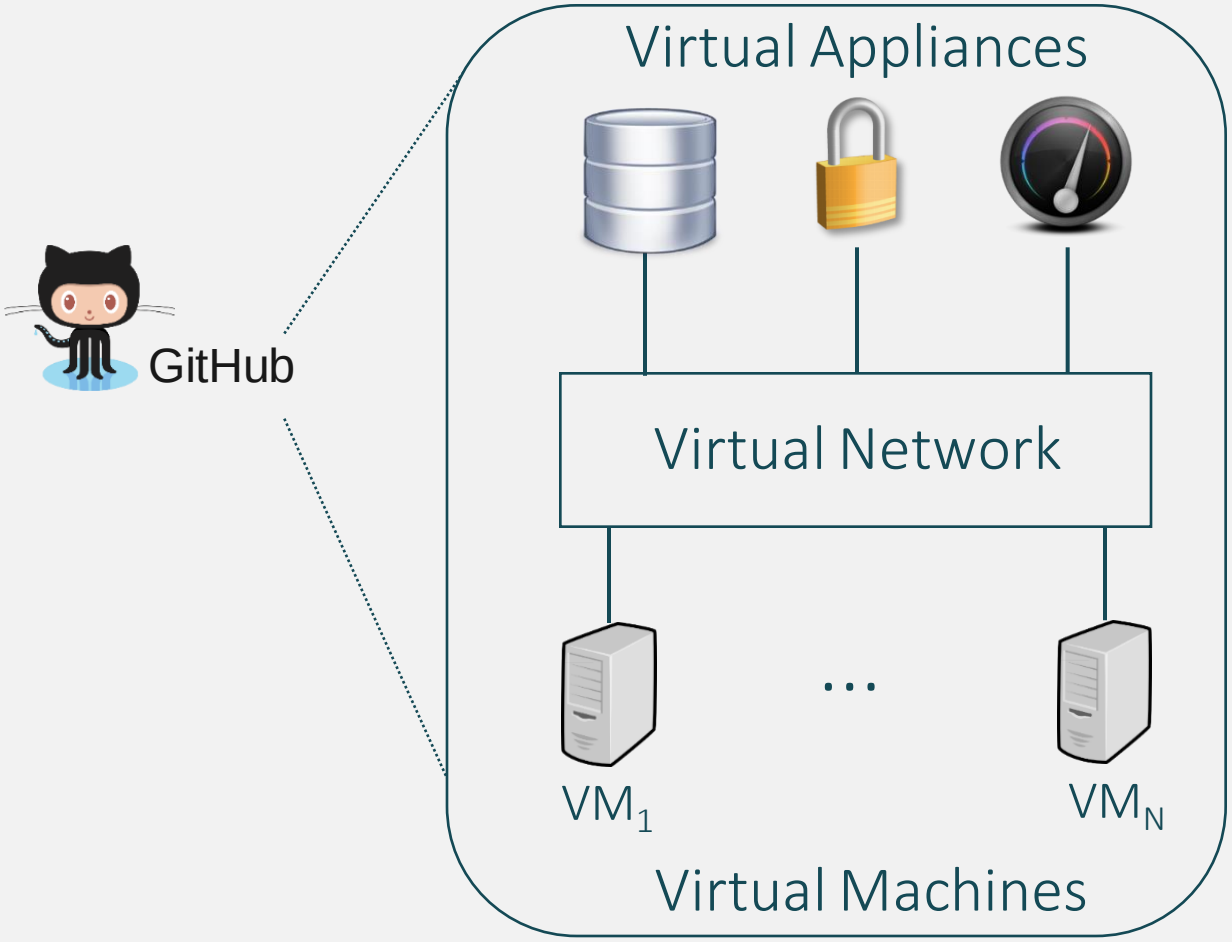


We need a new abstraction that encapsulates the semantics of end-to-end guarantees

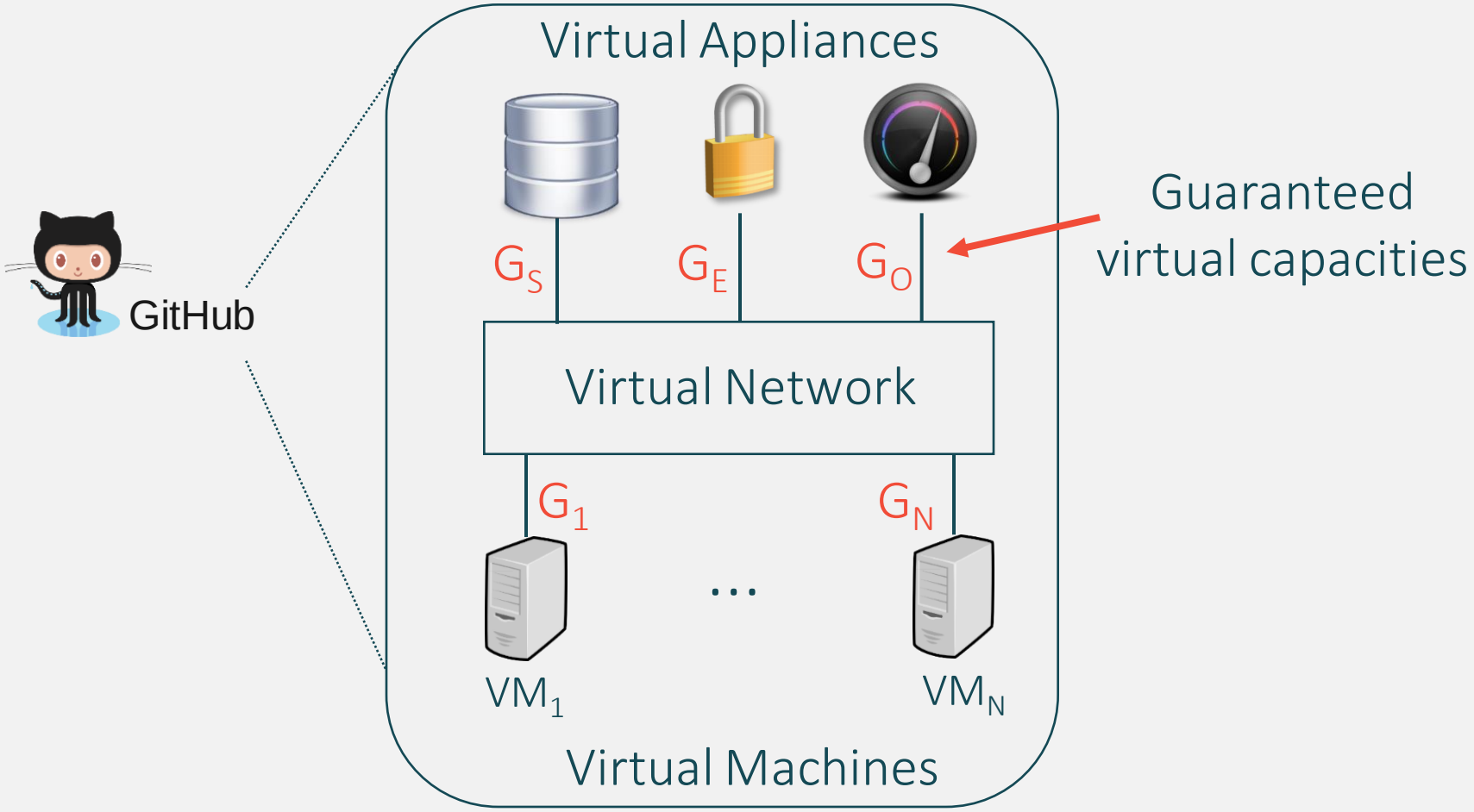
Outline

- ✓Introduction
- Virtual Datacenter (VDC) abstraction
- Throughput metric
- Architecture of Pulsar
- Experimental evaluation

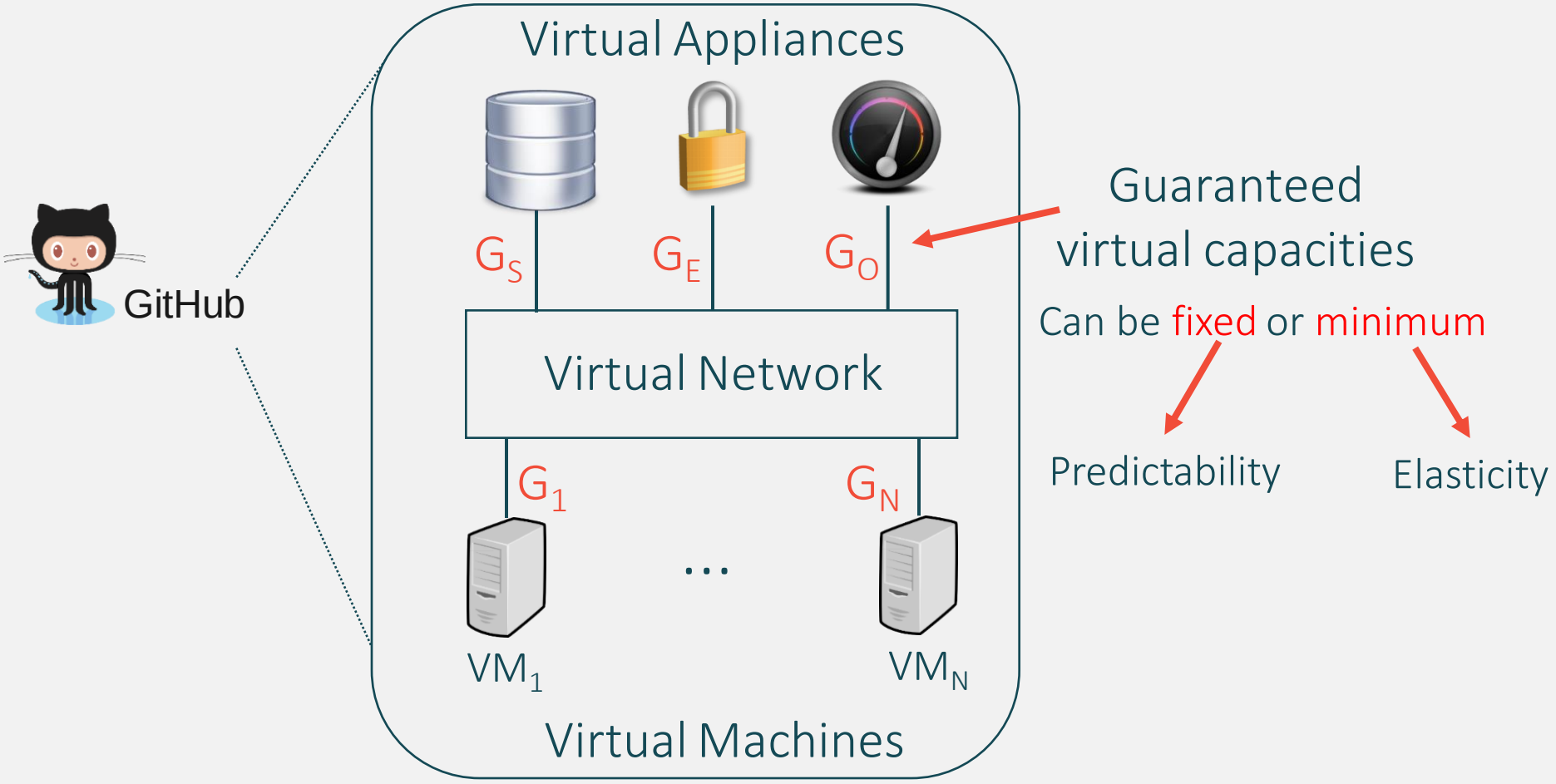
Virtual Datacenters (VDC) encapsulate end-to-end guarantees



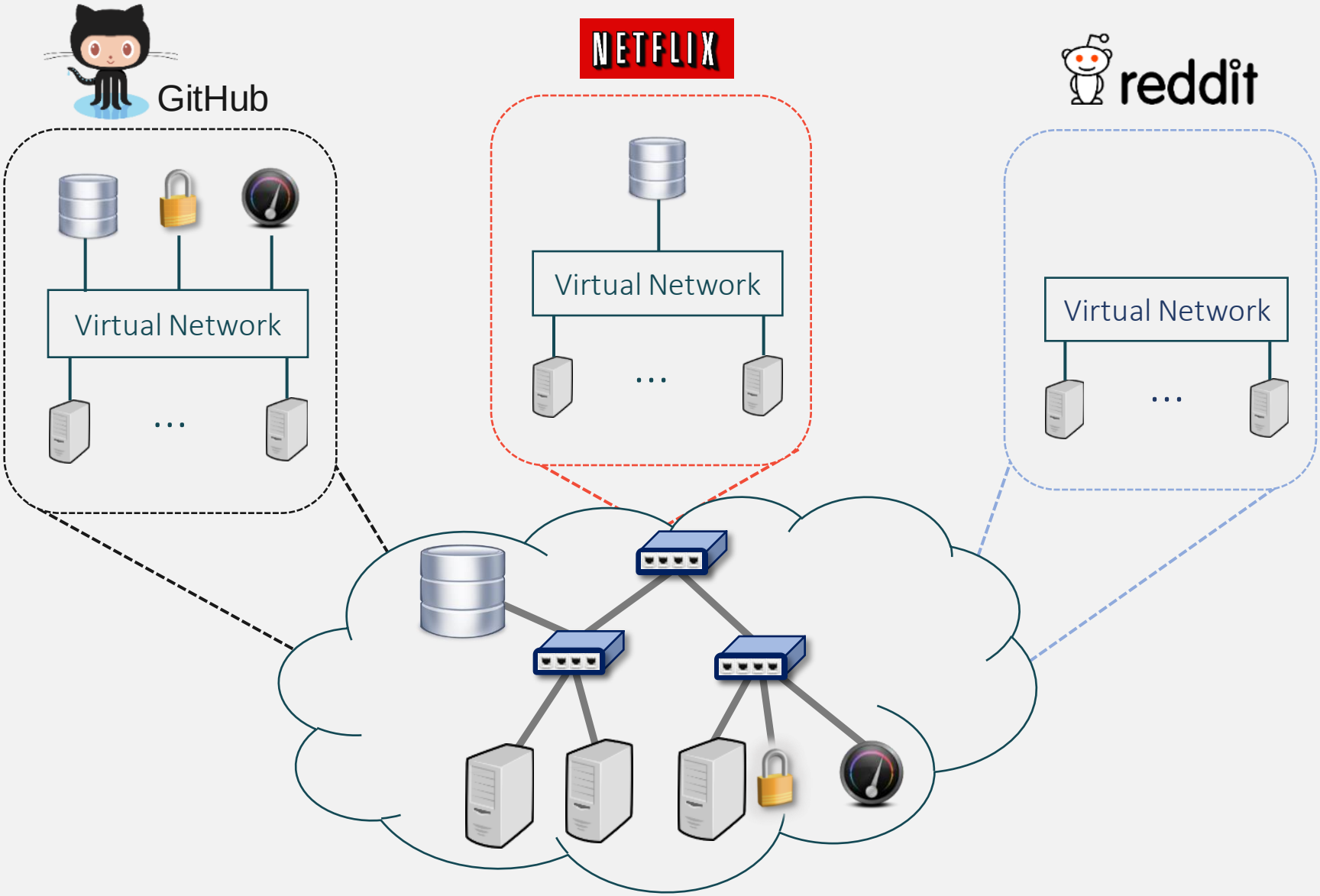
Virtual Datacenters (VDC) encapsulate end-to-end guarantees



Virtual Datacenters (VDC) encapsulate end-to-end guarantees



Admission control is performed when placing tenants' VDCs



Outline

- ✓Introduction
- ✓Virtual Datacenter (VDC) abstraction
- Throughput metric
- Architecture of Pulsar
- Experimental evaluation

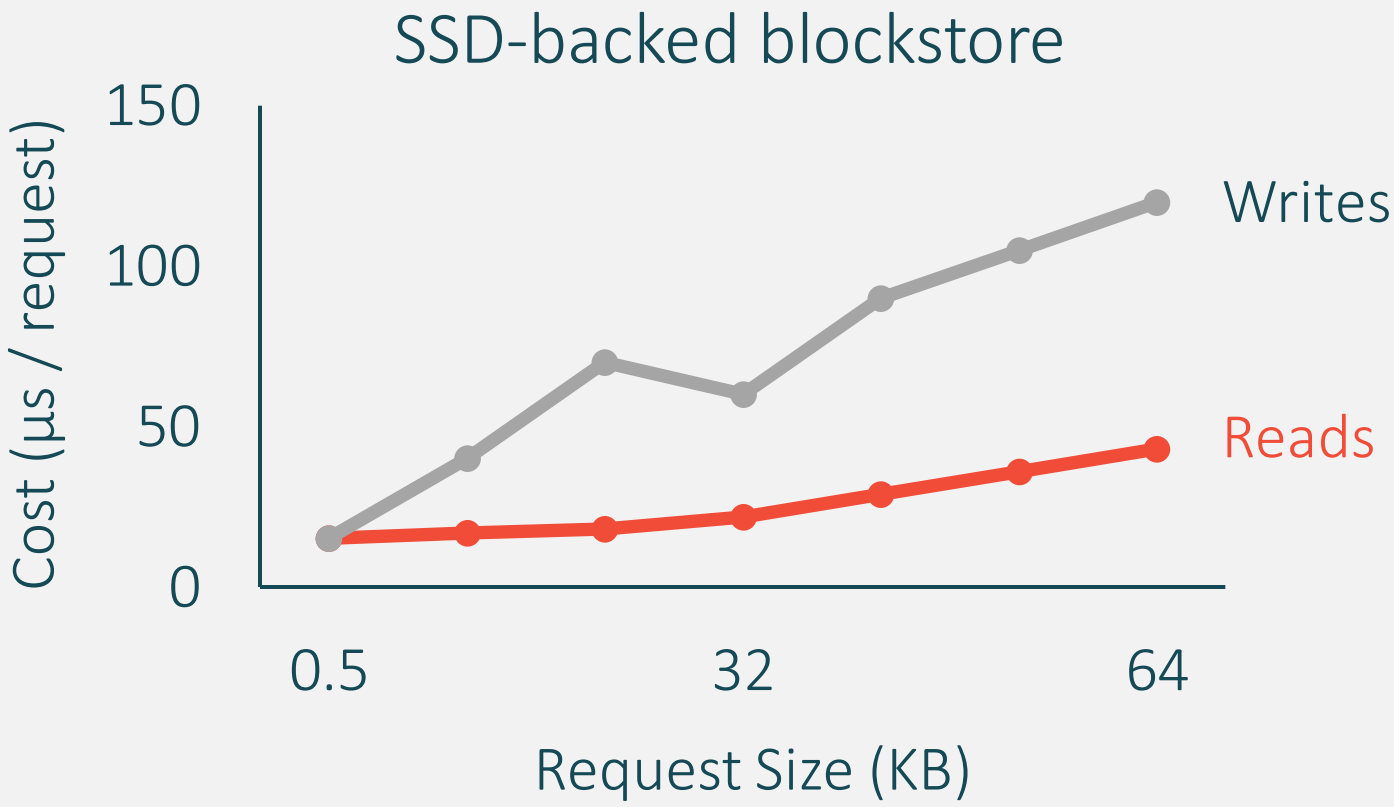
Standard throughput metrics have major tradeoffs

- Relative metrics: % share of the appliance
  - ✓ Provider: easy provisioning
  - ✗ Tenants: performance variability still present

Standard throughput metrics have major tradeoffs

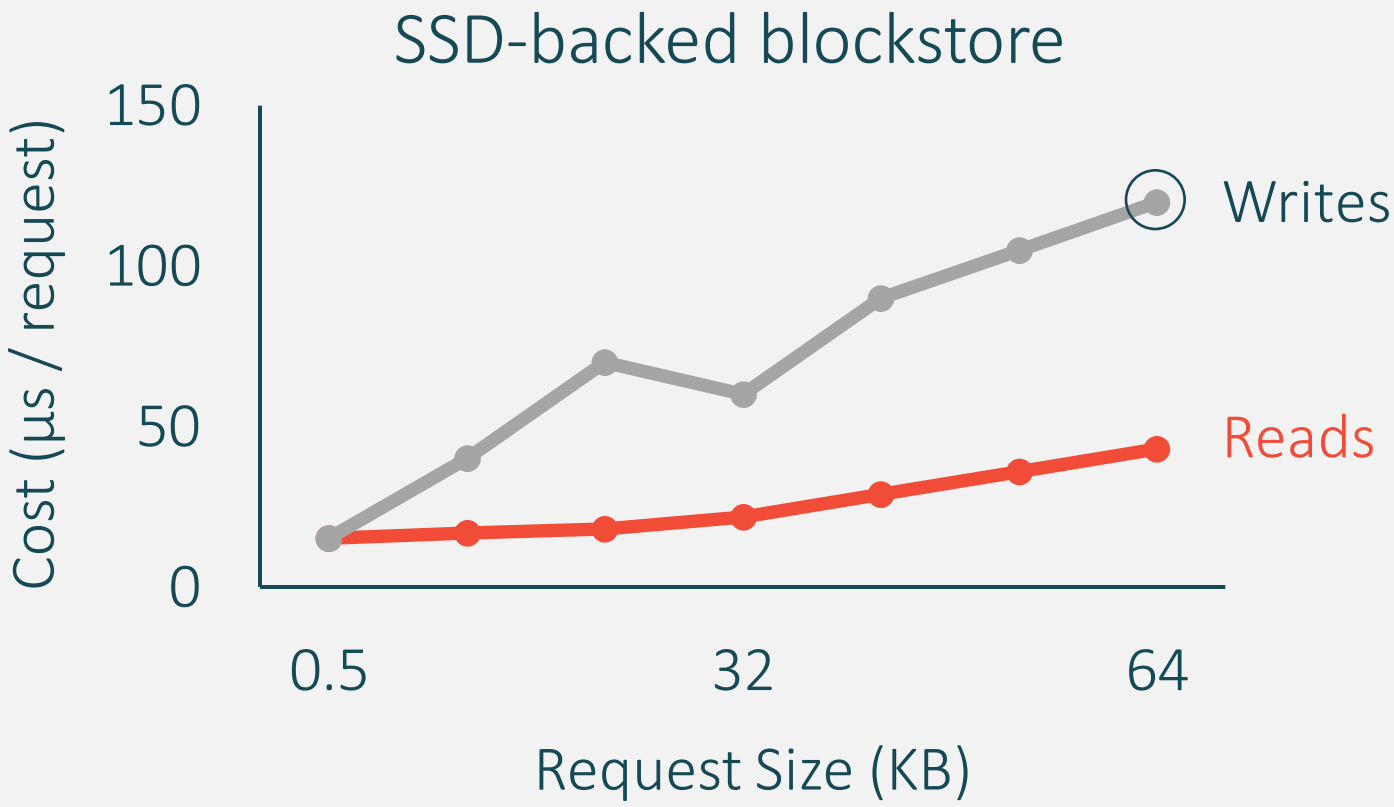
- Relative metrics: % share of the appliance
  - ✓ Provider: easy provisioning
  - ✗ Tenants: performance variability still present
- Absolute metrics: requests/second or bytes/second
  - ✗ Provider: provisioning based on costliest request
  - ✓ Tenants: absolute throughput guarantees

Request cost varies with request characteristics



Request cost varies with request characteristics

Guaranteeing requests/second requires conservative provisioning



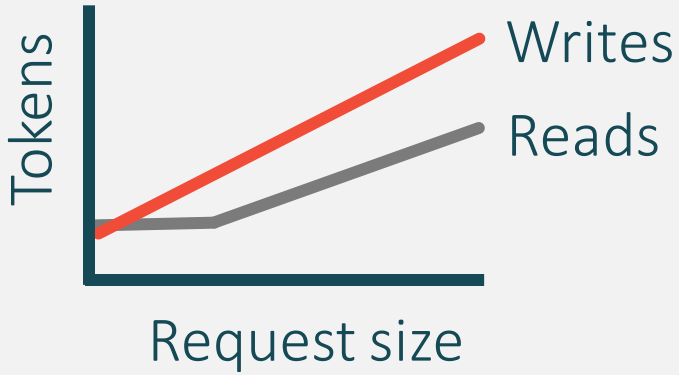
Virtual request cost (in tokens)

- Provider selects a (fixed) virtual cost function for each resource

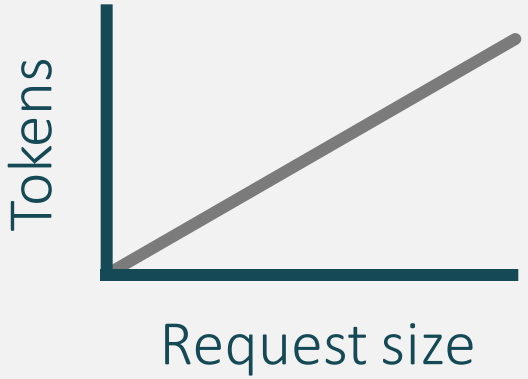
Virtual request cost (in tokens)

- Provider selects a (fixed) virtual cost function for each resource

SSD-backed blockstore



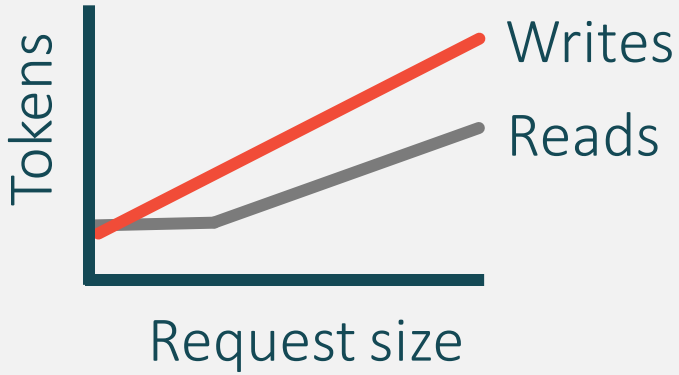
Network Links / NICs



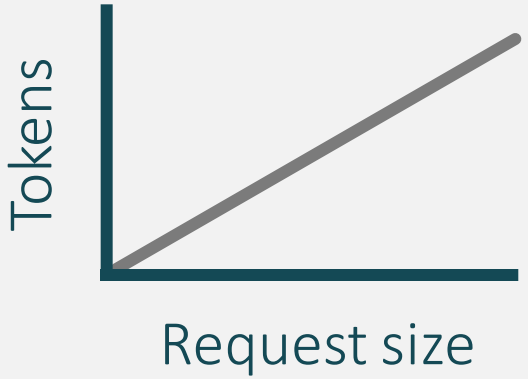
Virtual request cost (in tokens)

- Provider selects a (fixed) virtual cost function for each resource

SSD-backed blockstore



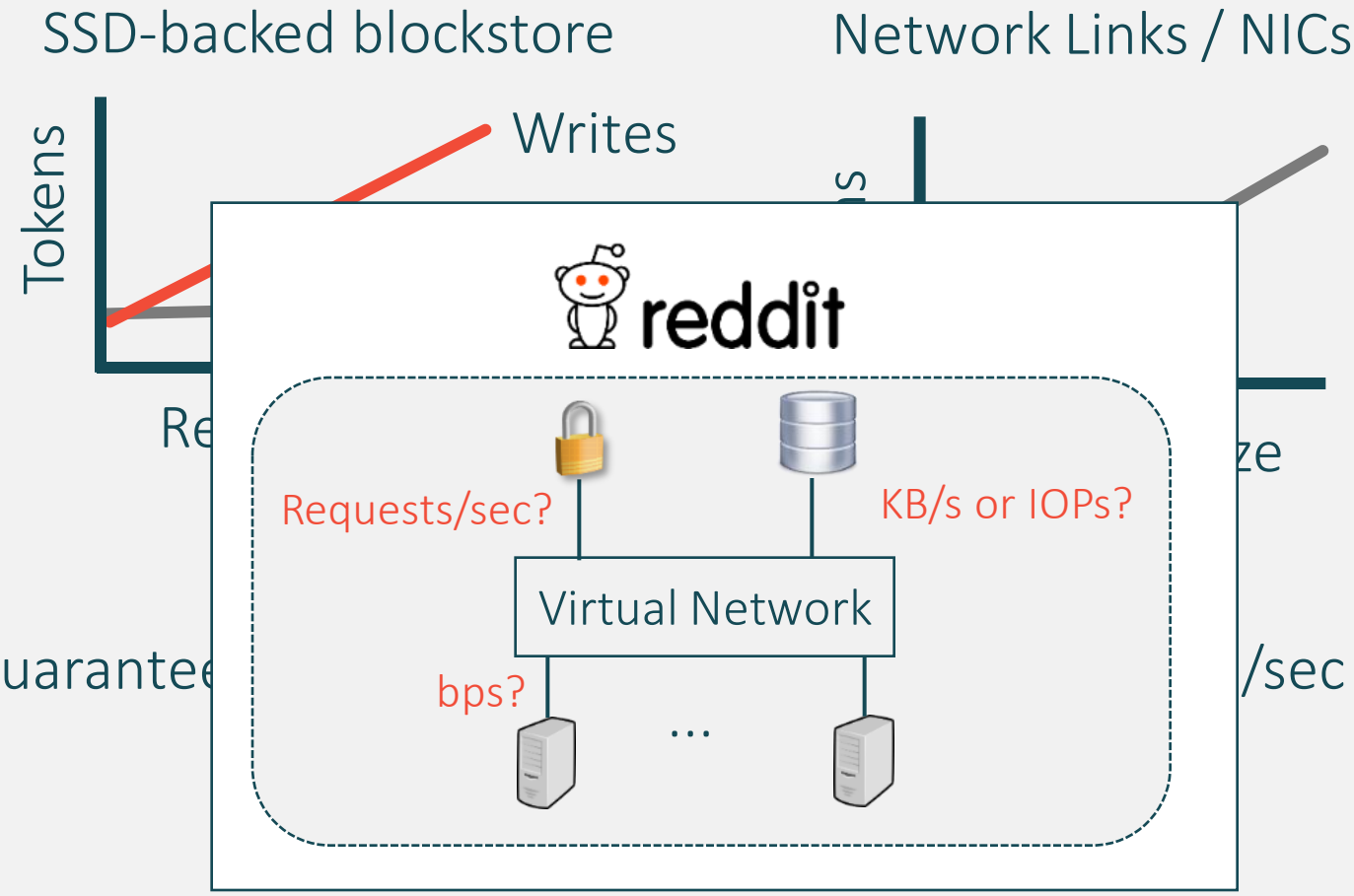
Network Links / NICs



- Guaranteed virtual capacities are defined in tokens/sec

Virtual request cost (in tokens)

- Provider selects a (fixed) virtual cost function for each resource



- Guarantee

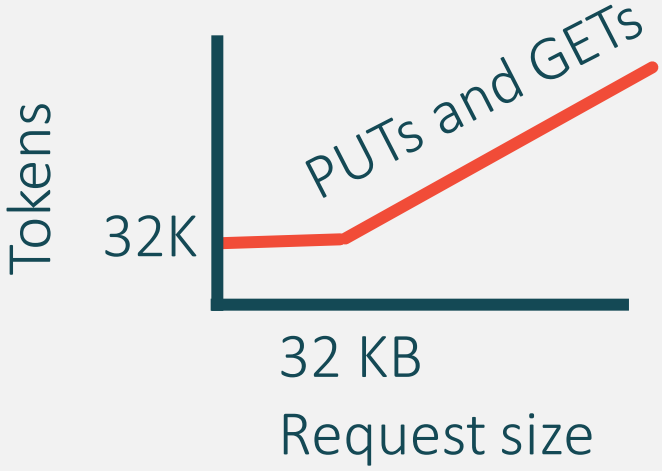
Virtual request cost strikes a good compromise between tenants and the provider

- Tenants can translate their guarantees to other metrics for their workloads

- Provider has more flexibility when provisioning the datacenter

Virtual request cost strikes a good compromise between tenants and the provider

- Tenants can translate their guarantees to other metrics for their workloads

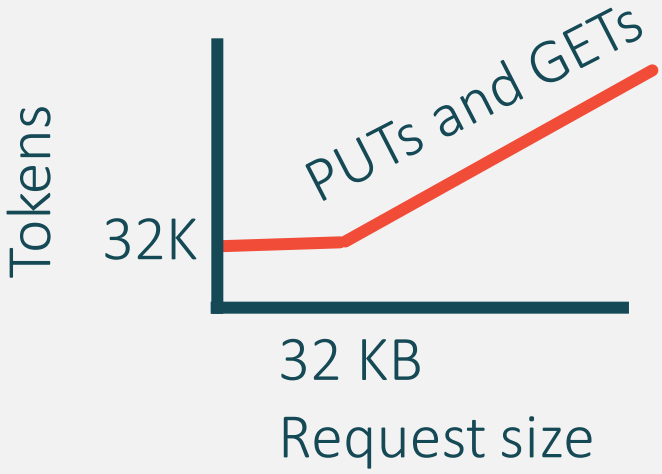


Guarantee: 32 Ktokens/sec

- Provider has more flexibility when provisioning the datacenter

Virtual request cost strikes a good compromise between tenants and the provider

- Tenants can translate their guarantees to other metrics for their workloads



Guarantee: 32 Ktokens/sec

Workload:

- 8 KB PUTs --> 2 PUT/sec
- 32 KB GETs --> 1 GET/sec

- Provider has more flexibility when provisioning the datacenter

Outline

- ✓Introduction
- ✓Virtual Datacenter (VDC) abstraction
- ✓ Throughput metric
- Architecture of Pulsar
- Experimental evaluation

Design goals and decisions

- No modification to appliances, switches, guest OSes, applications
- Preserve work-conservation
- Enable rich policies that are easy to change

Design goals and decisions

- No modification to appliances, switches, guest OSes, applications
  - Perform enforcement entirely at end-hosts via a rate enforcer
- Preserve work-conservation
- Enable rich policies that are easy to change

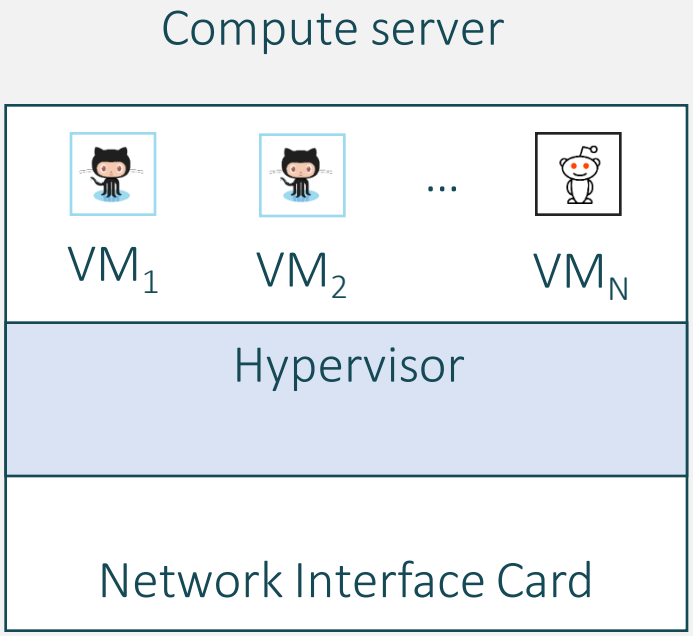
Design goals and decisions

- No modification to appliances, switches, guest OSes, applications
  - Perform enforcement entirely at end-hosts via a rate enforcer
- Preserve work-conservation
  - Allocate unused resources to tenants and VMs that can use them
  - Requires coordination between rate enforcers
- Enable rich policies that are easy to change

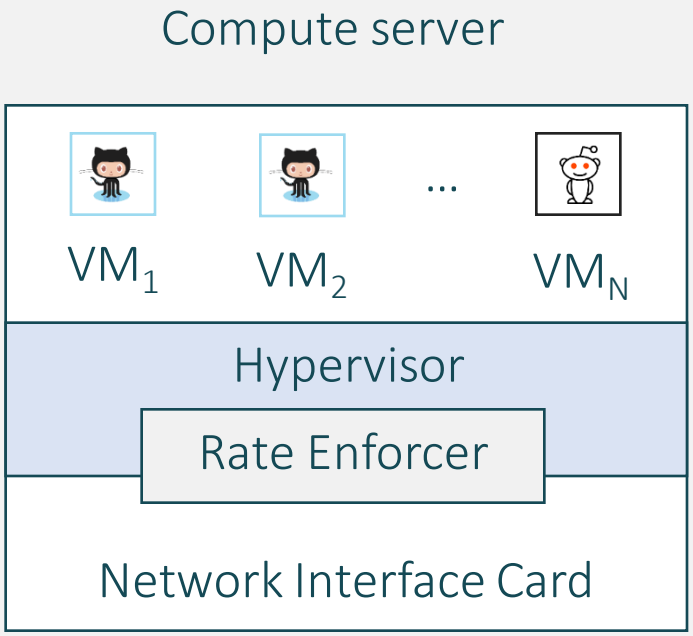
Design goals and decisions

- No modification to appliances, switches, guest OSes, applications
  - Perform enforcement entirely at end-hosts via a rate enforcer
- Preserve work-conservation
  - Allocate unused resources to tenants and VMs that can use them
  - Requires coordination between rate enforcers
- Enable rich policies that are easy to change
  - Simplicity is key
  - Perform coordination through a centralized control plane

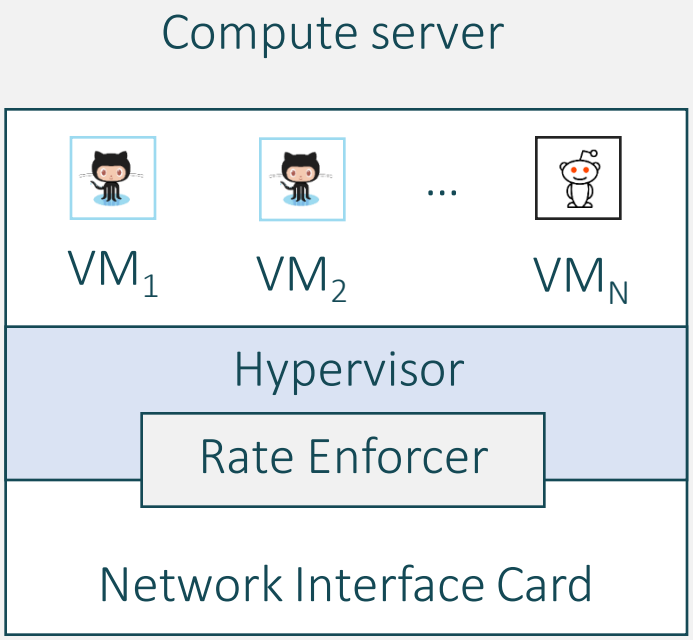
Pulsar’s architecture



Pulsar's architecture



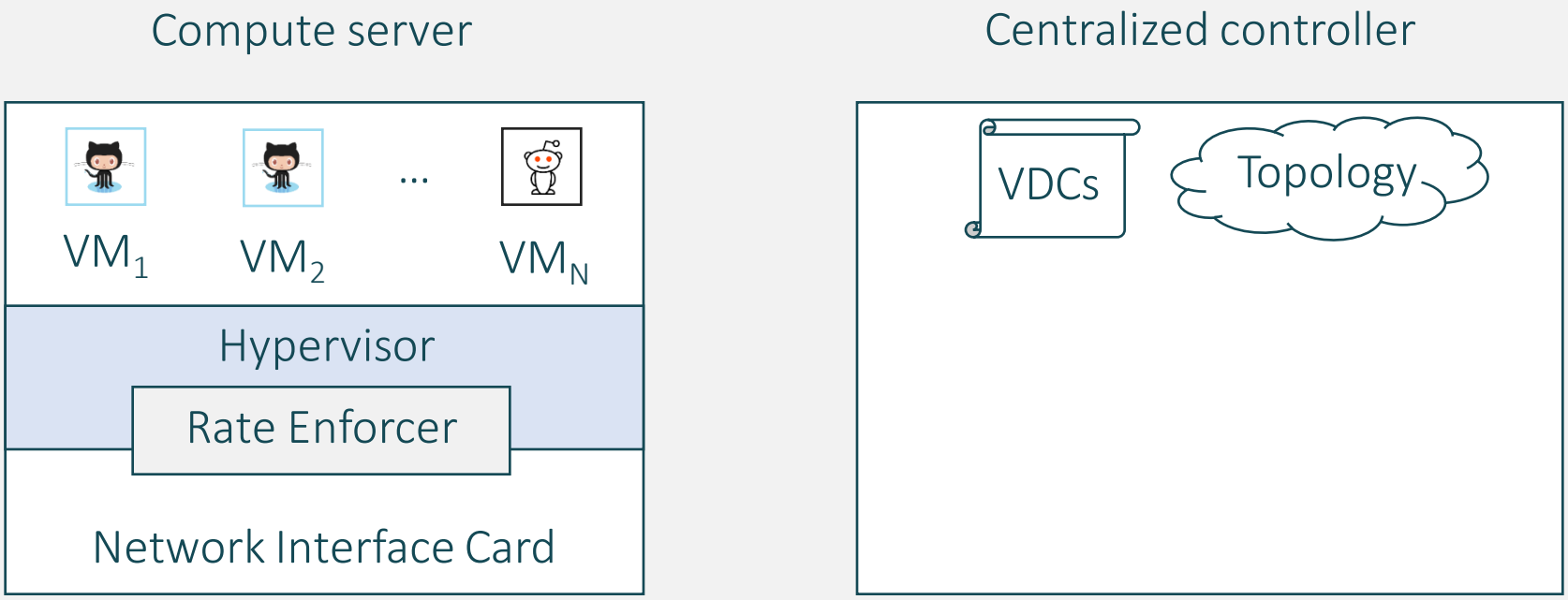
Pulsar's architecture



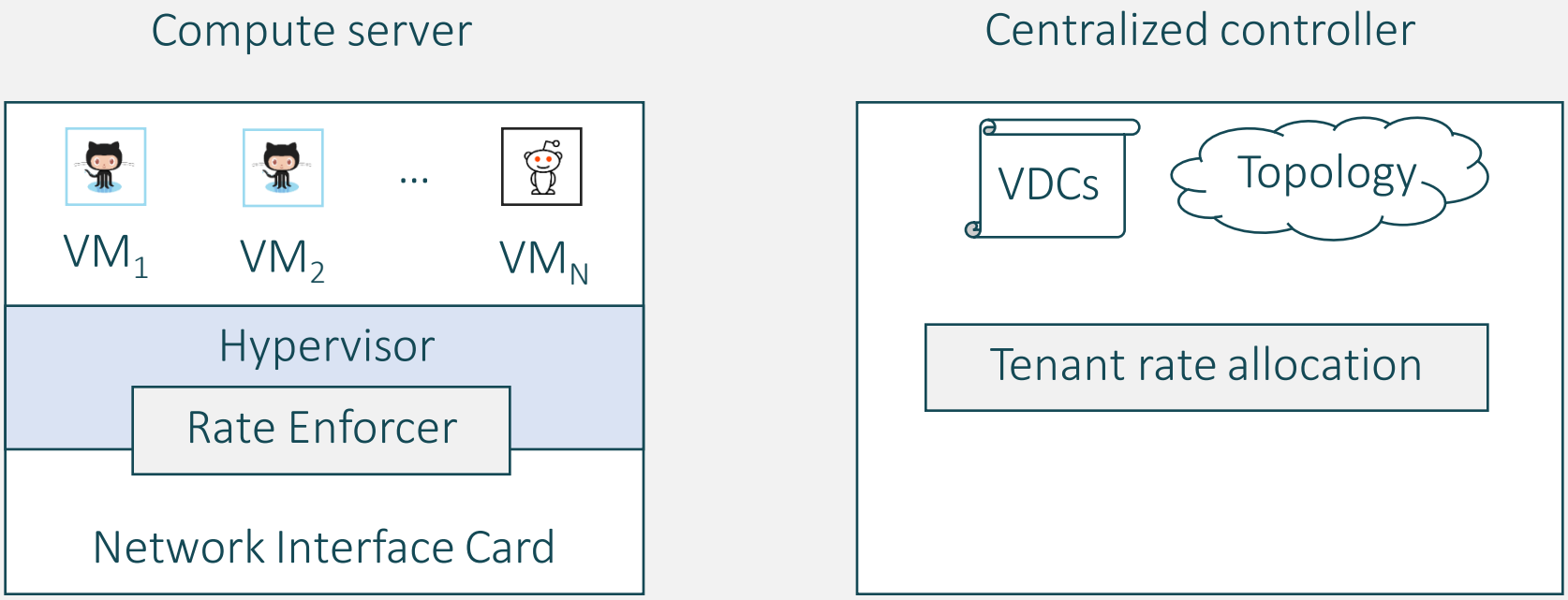
Rate enforcer

- Estimates the demand of every VM
- Emulates request cost at different resources by applying cost functions
- Enforces allocation provided by controller

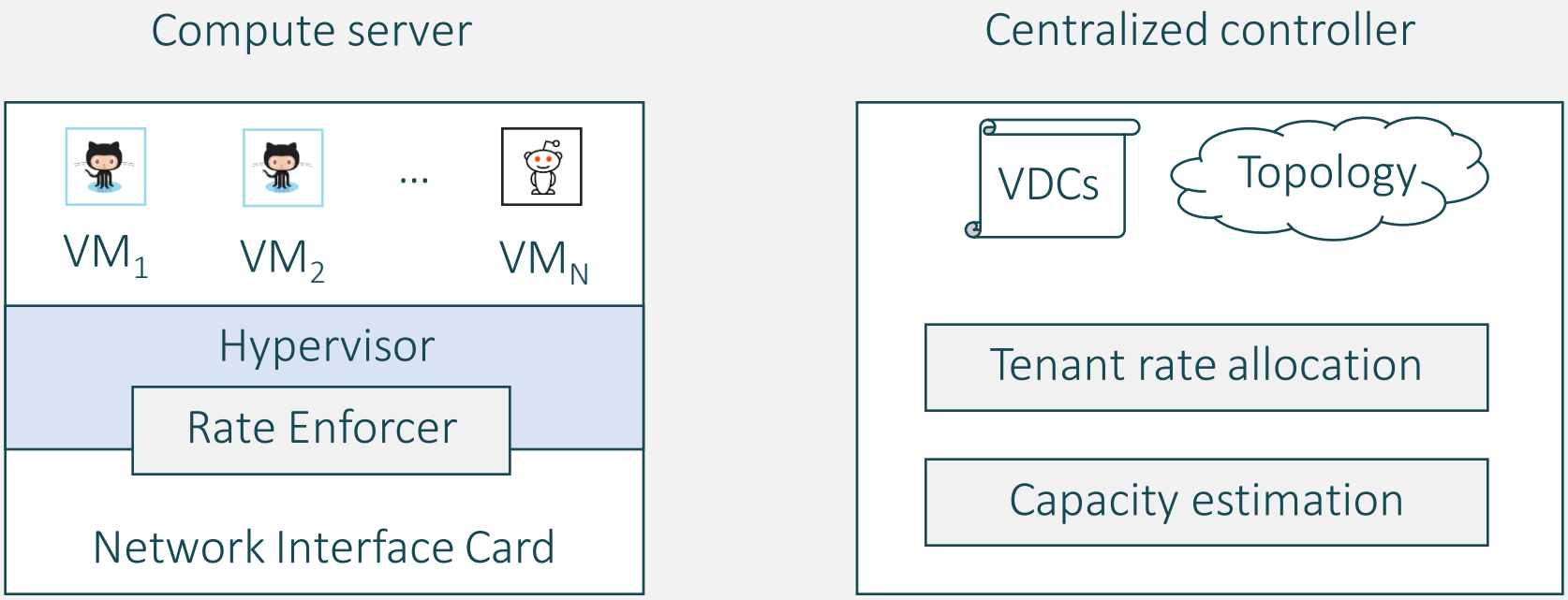
Pulsar's architecture



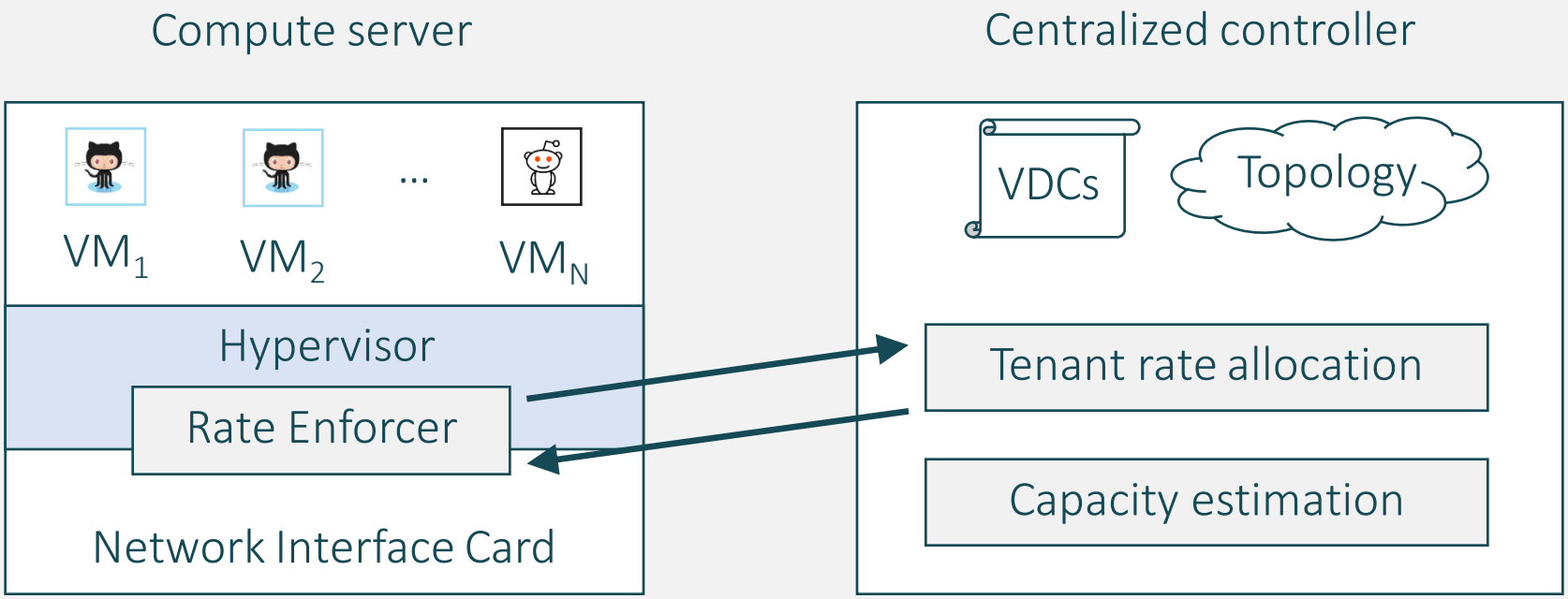
Pulsar's architecture



Pulsar's architecture

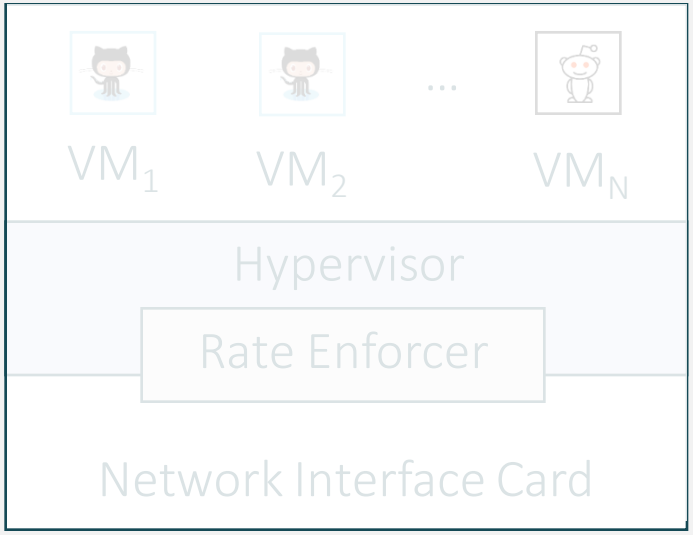


Pulsar's architecture

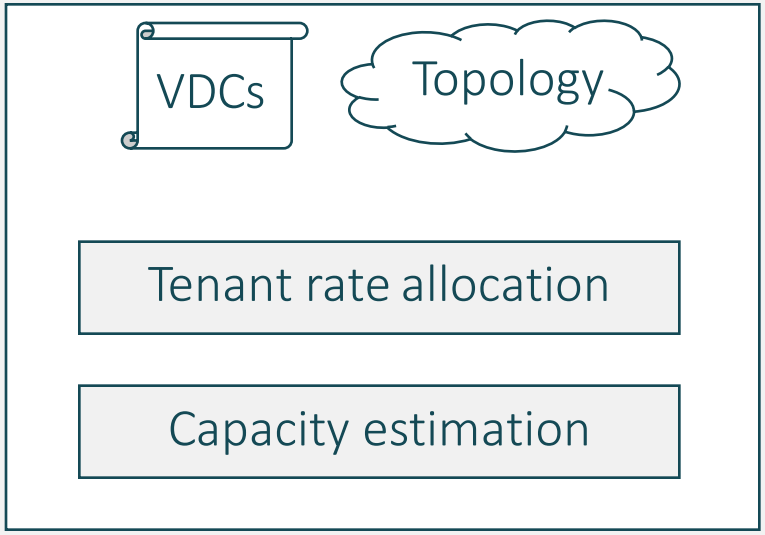


Pulsar's architecture

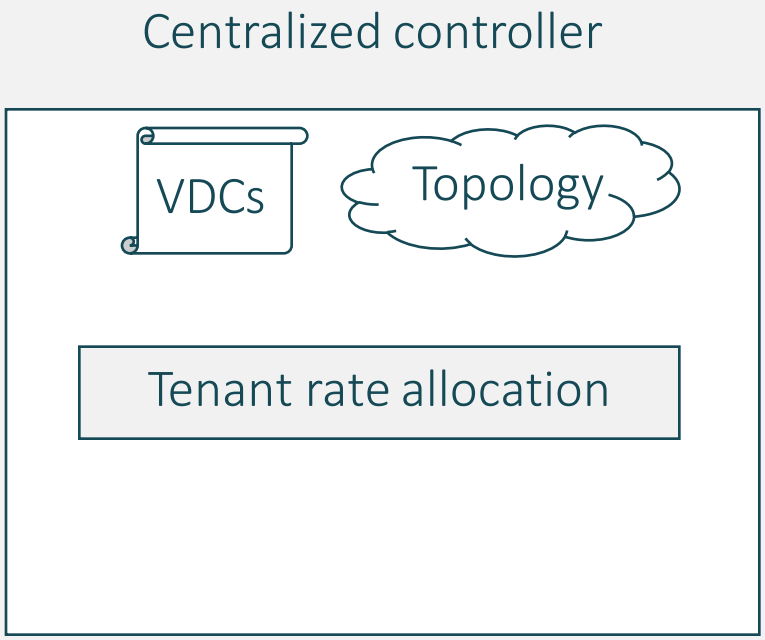
Compute server



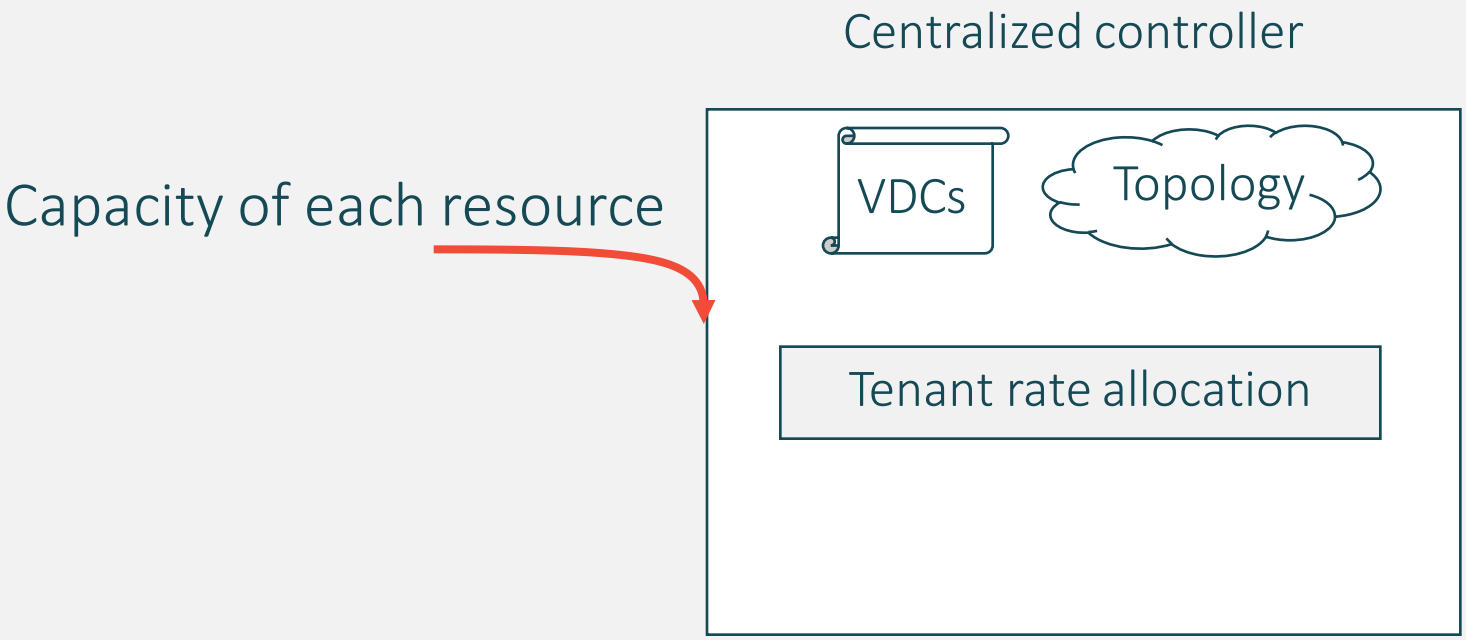
Centralized controller



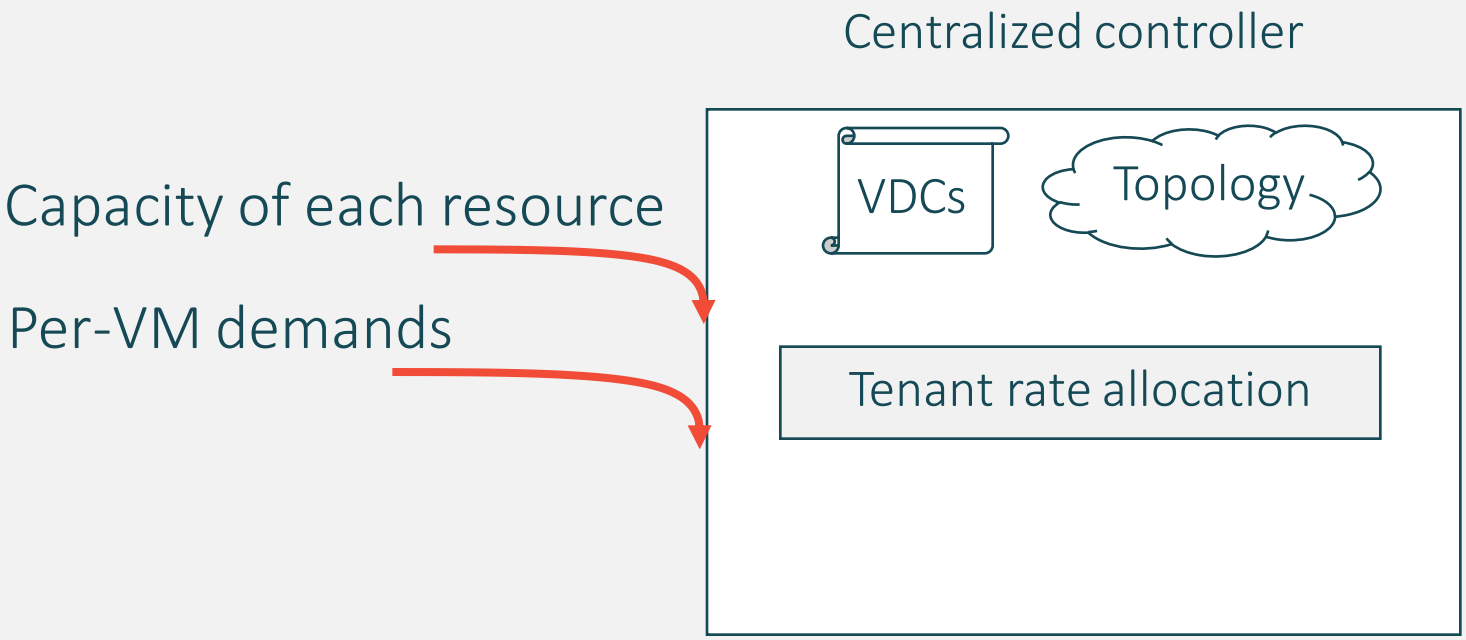
Pulsar allocates resources to each VM-VM pair



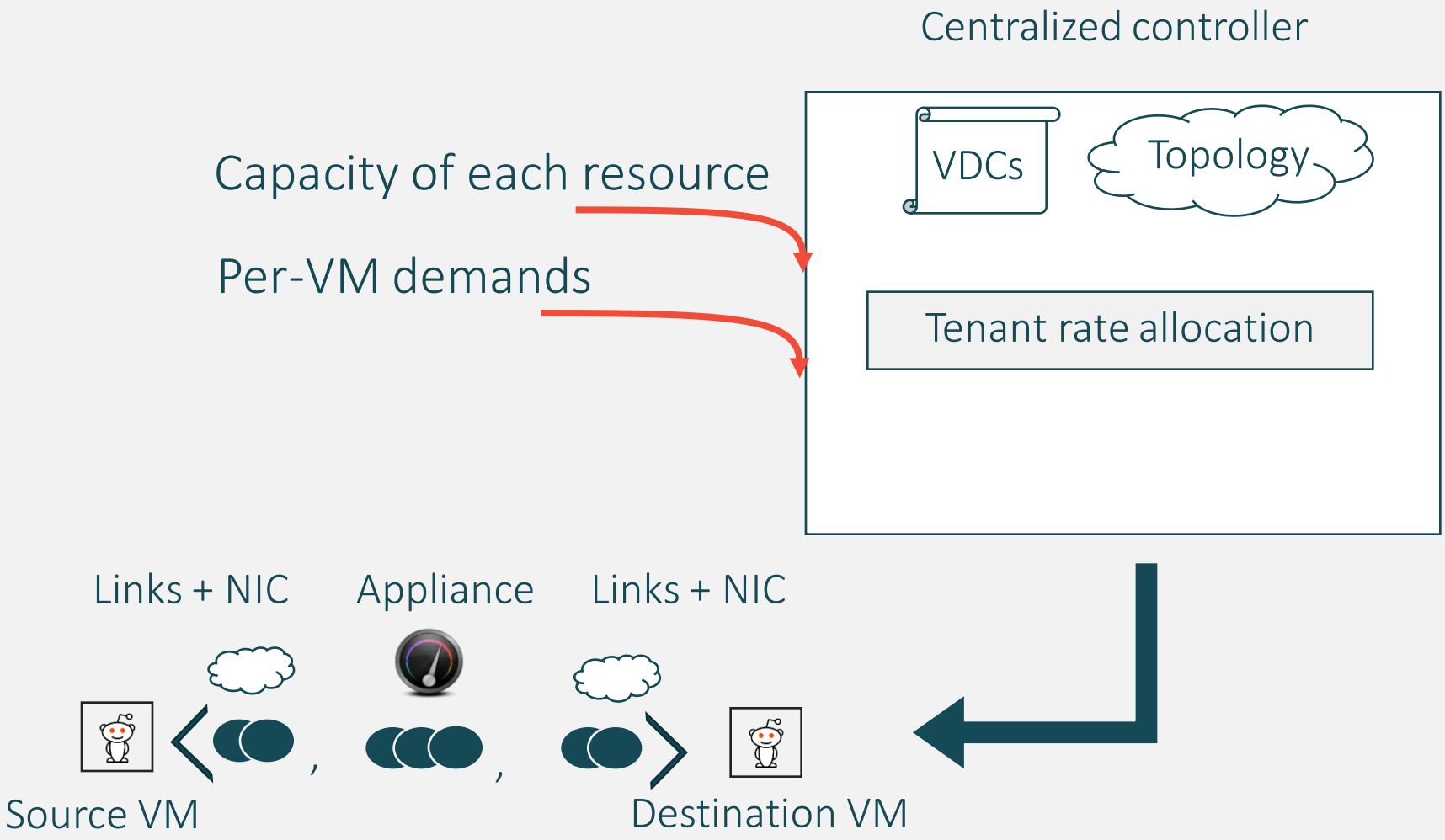
Pulsar allocates resources to each VM-VM pair



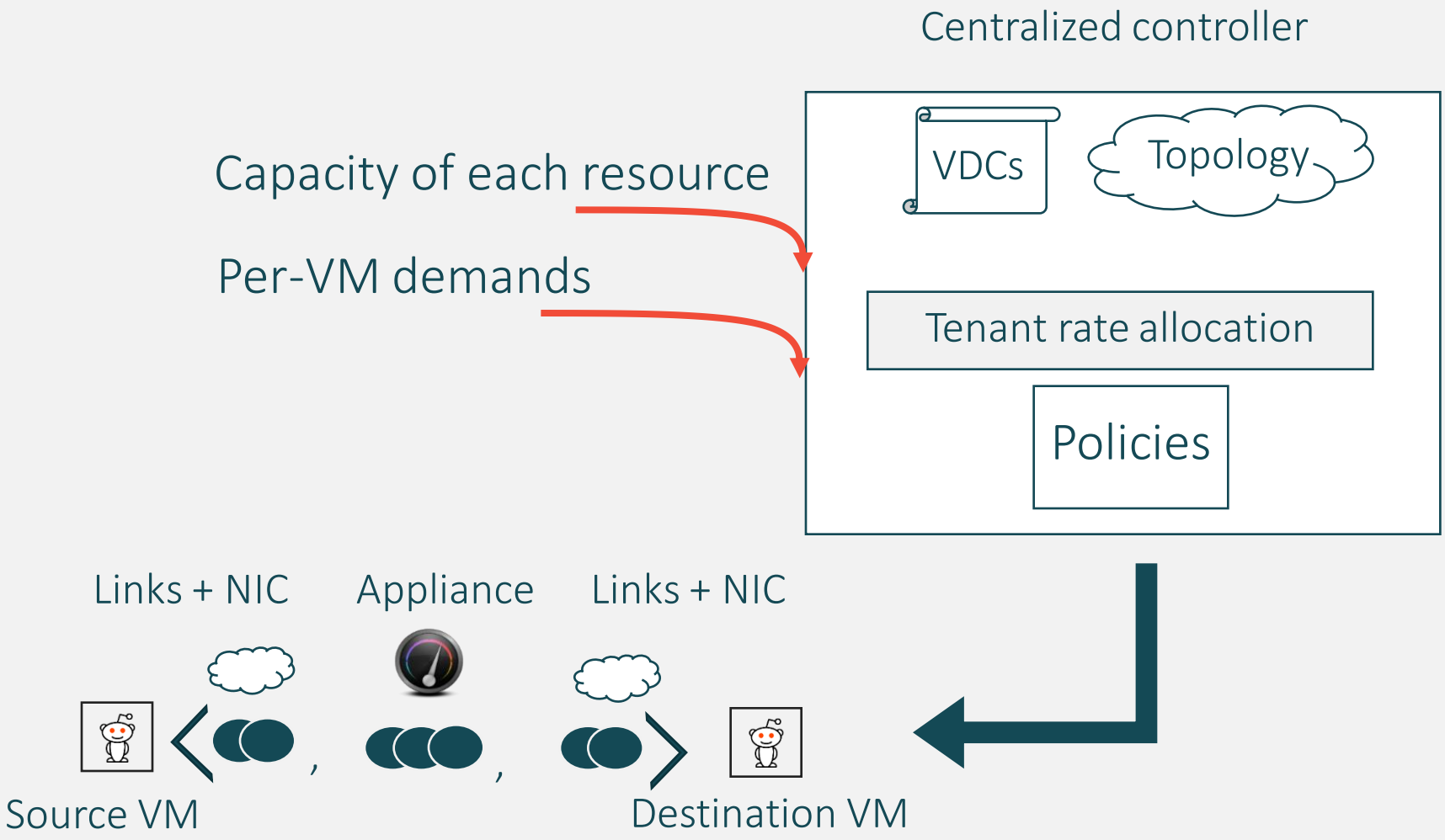
Pulsar allocates resources to each VM-VM pair



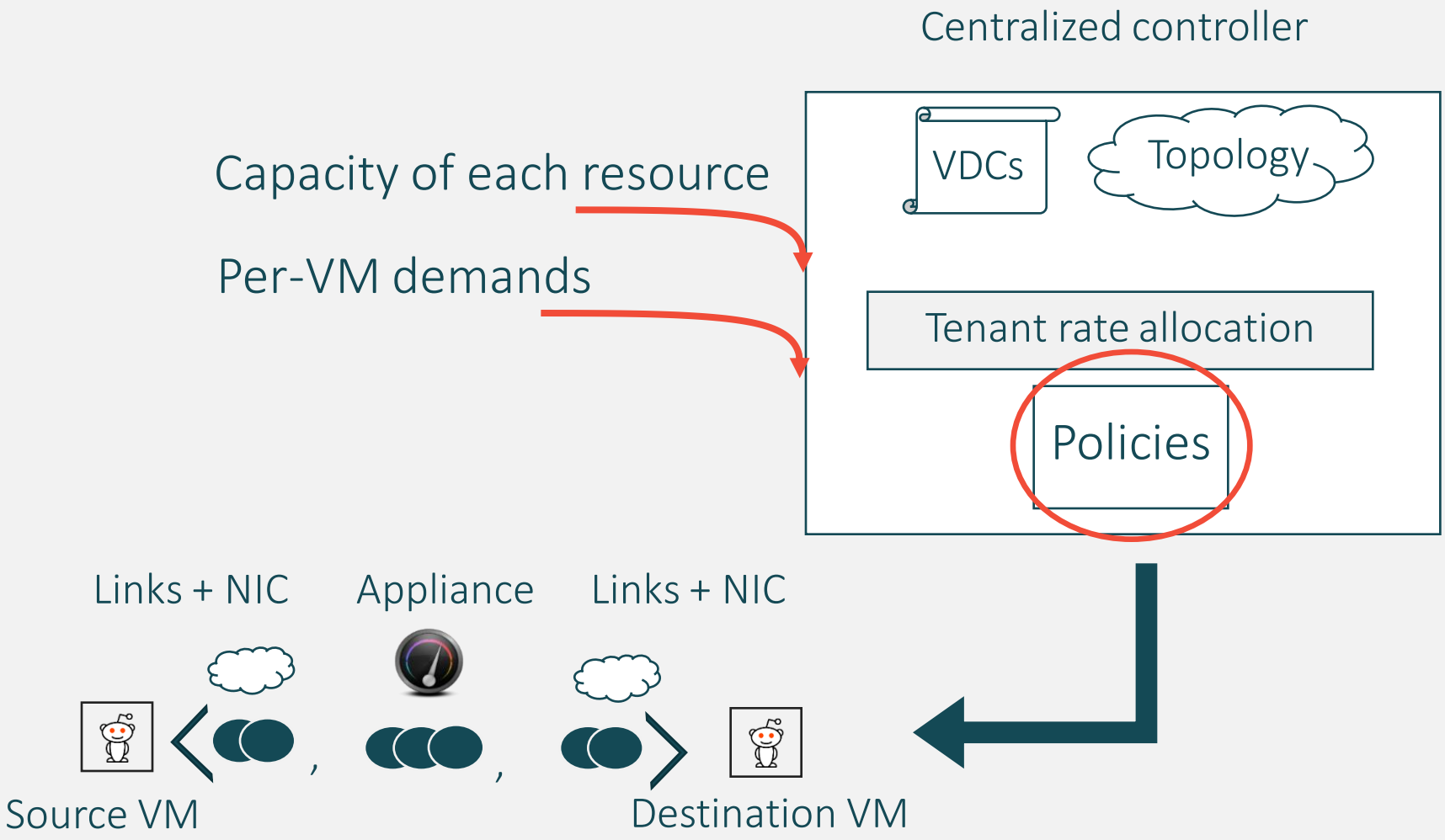
Pulsar allocates resources to each VM-VM pair



Pulsar allocates resources to each VM-VM pair



Pulsar allocates resources to each VM-VM pair

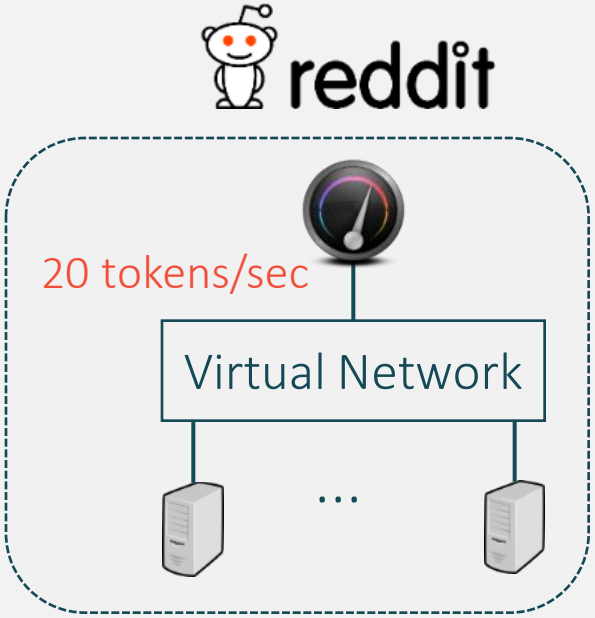


Allocations are based on two policies

- Tenant-specified policy
  - Specifies how VDC resources are divided to VMs

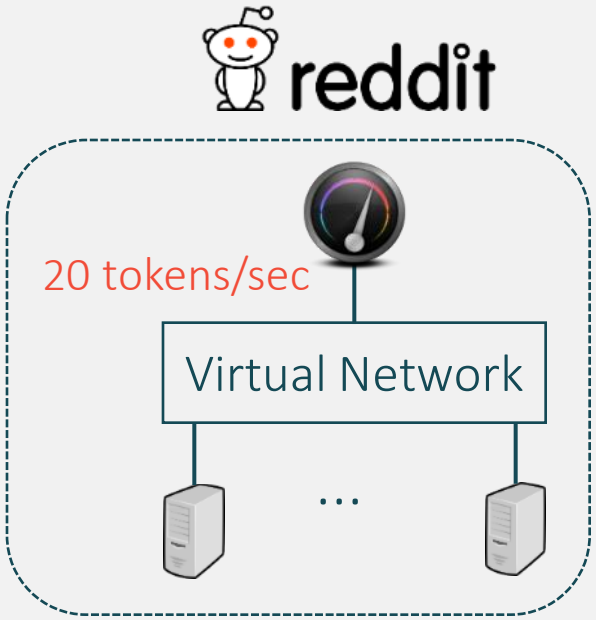
Allocations are based on two policies

- Tenant-specified policy
  - Specifies how VDC resources are divided to VMs



Allocations are based on two policies

- Tenant-specified policy
  - Specifies how VDC resources are divided to VMs



Example policies:

- Divide 20 tokens/sec fairly across all active VMs
- Give all 20 tokens/sec to VM 3 (whenever it is active)

Allocations are based on two policies

- Tenant-specified policy
  - Specifies how VDC resources are divided to VMs
- Provider-specified policy
  - Specifies how spare resources are given to tenants' VMs

Allocations are based on two policies

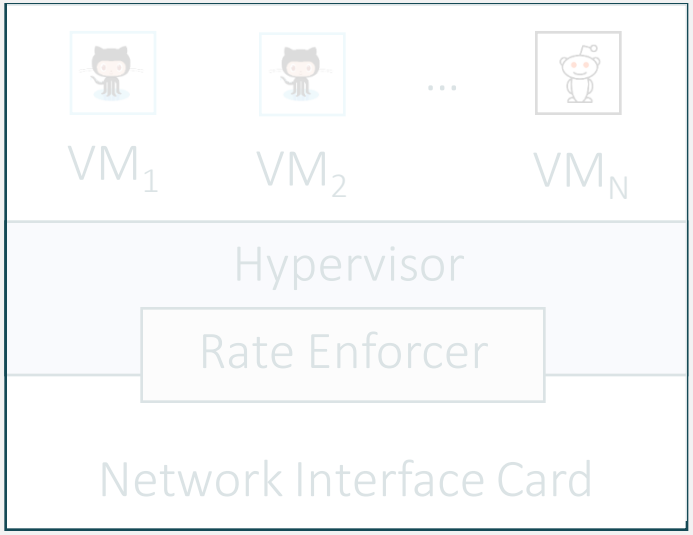
- Tenant-specified policy
  - Specifies how VDC resources are divided to VMs
- Provider-specified policy
  - Specifies how spare resources are given to tenants' VMs

  Example policies:

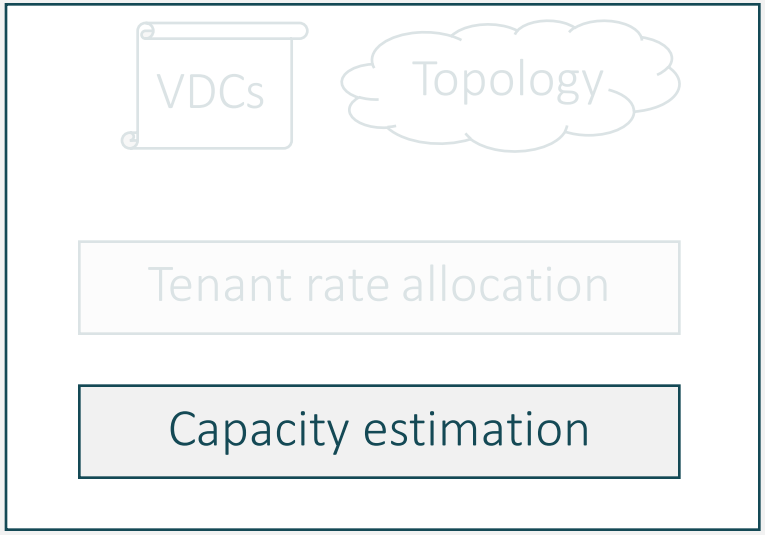
  - Distribute spare resource fairly across all tenants' VMs
  - Distribute spare resources in a way that maximizes profit

Pulsar's architecture

Compute server



Centralized controller



Congestion control protocols estimate network capacity

- Basic idea
  - Each network flow probes for a higher capacity estimate
  - Decreases allocation on observing congestion

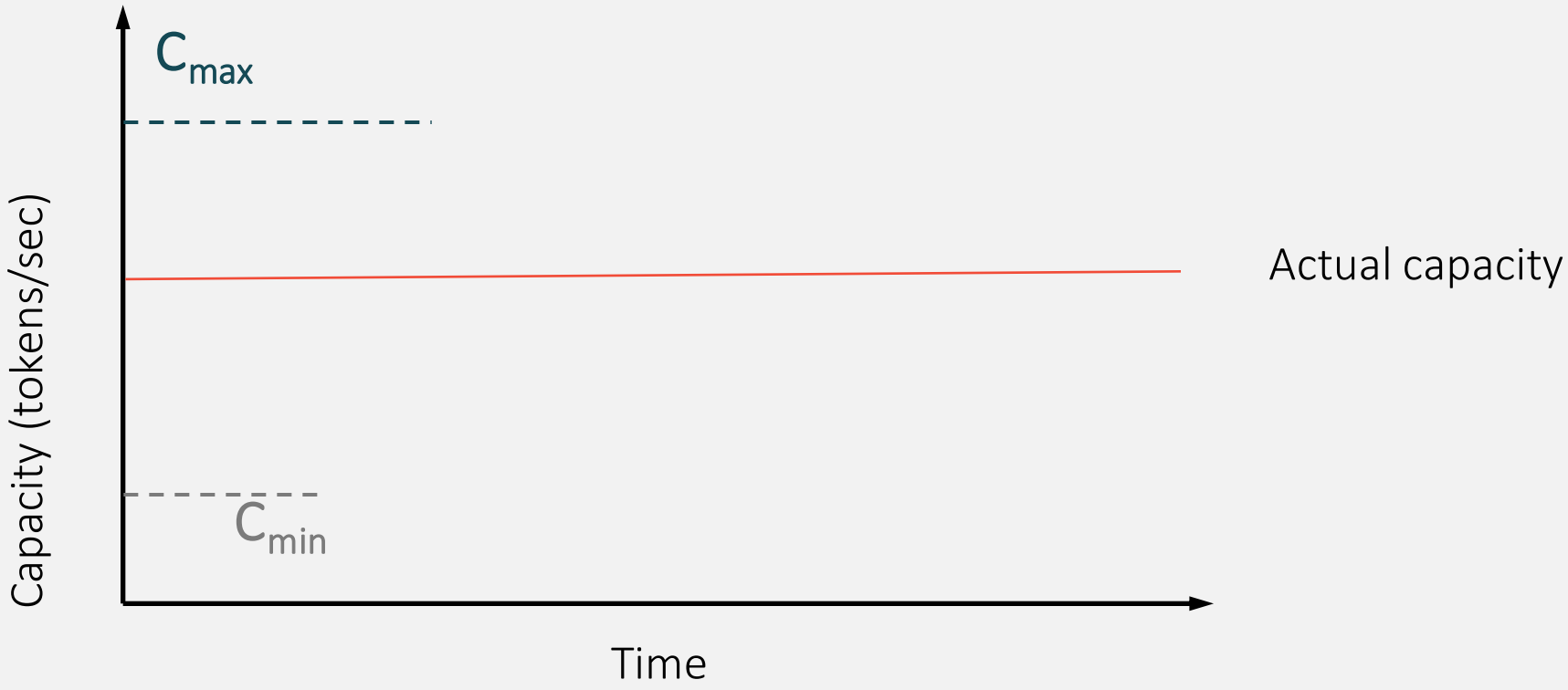
Congestion control protocols estimate network capacity

- Basic idea
  - Each network flow probes for a higher capacity estimate
  - Decreases allocation on observing congestion
- Challenges
  - Congestion signals are not present or noisy
  - Distributed operation is complex
  - Estimation is tightly coupled with allocation

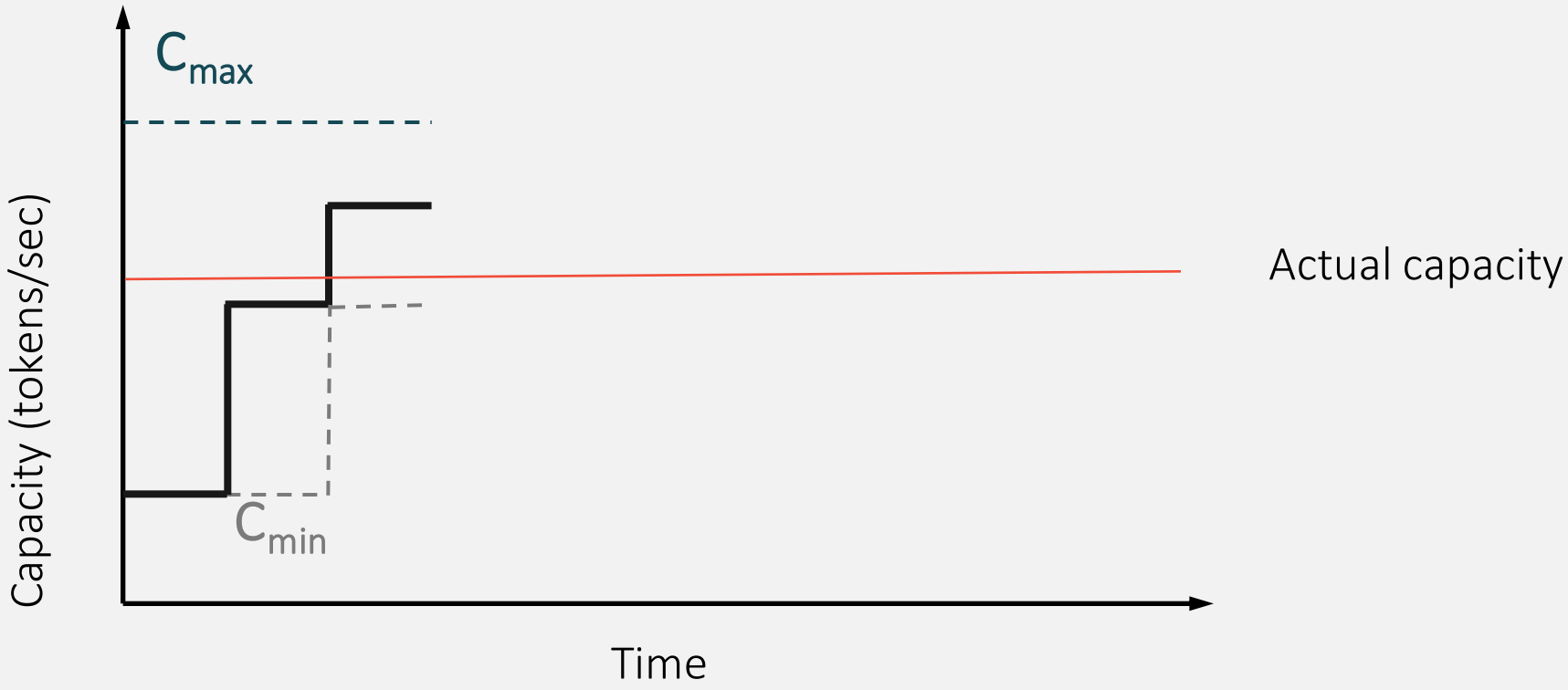
Centralized estimation algorithm

- Maintain window for capacity estimate
- Refine window based on congestion signals

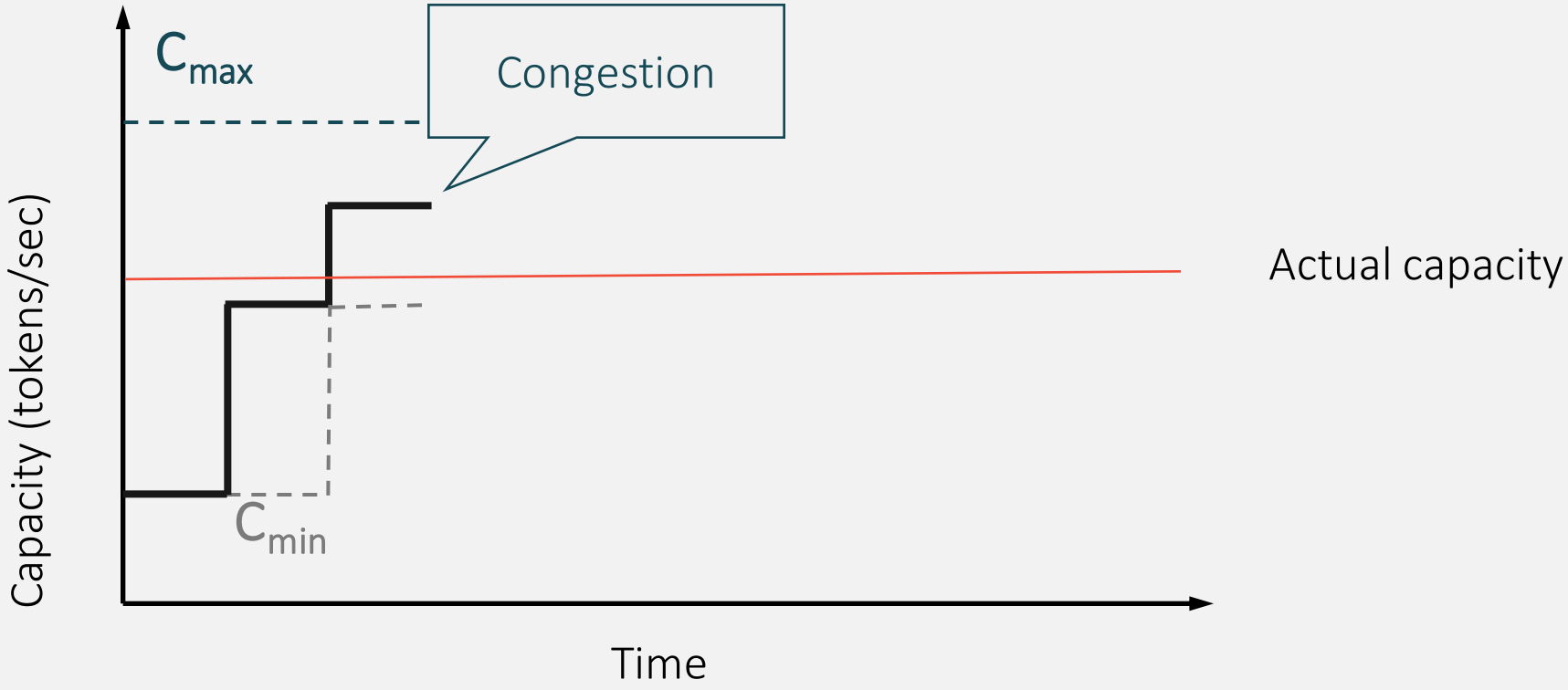
Controller-based capacity estimation



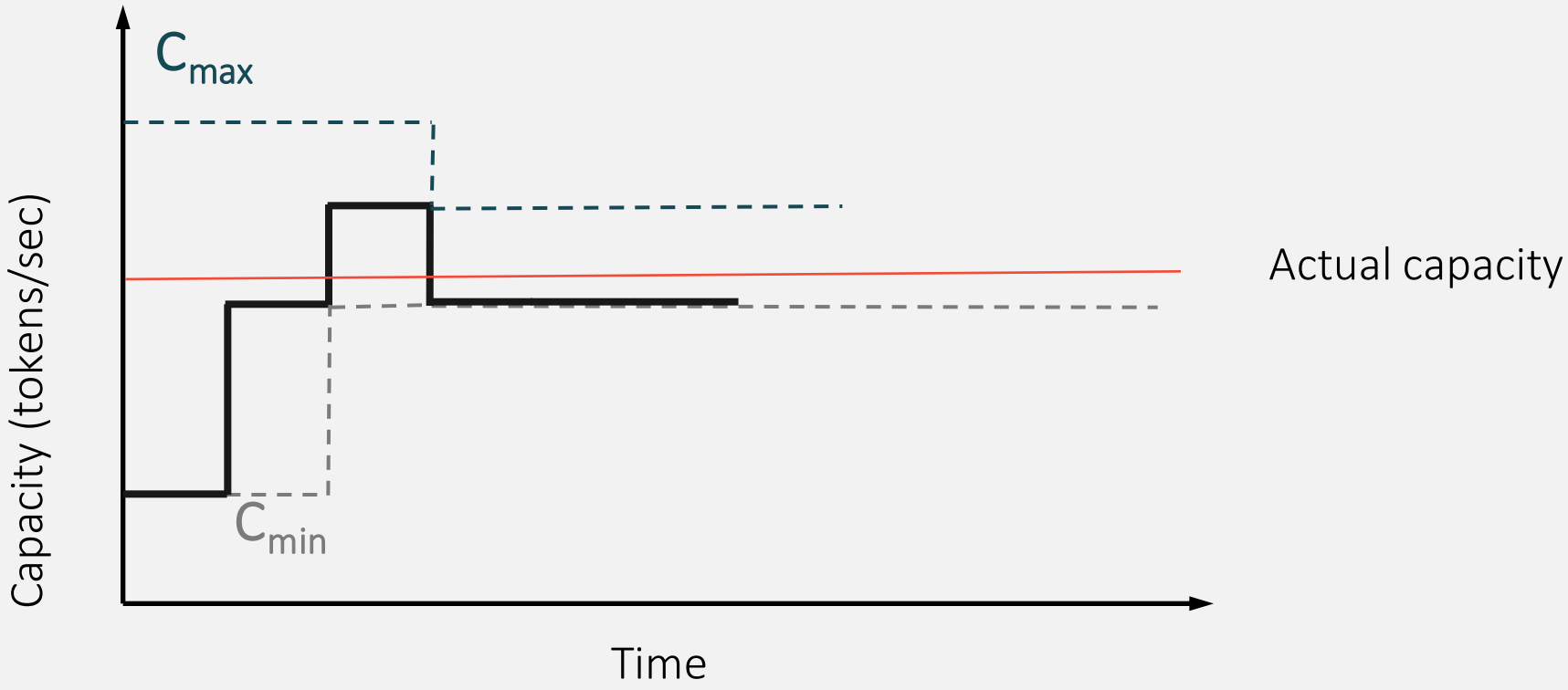
Controller-based capacity estimation



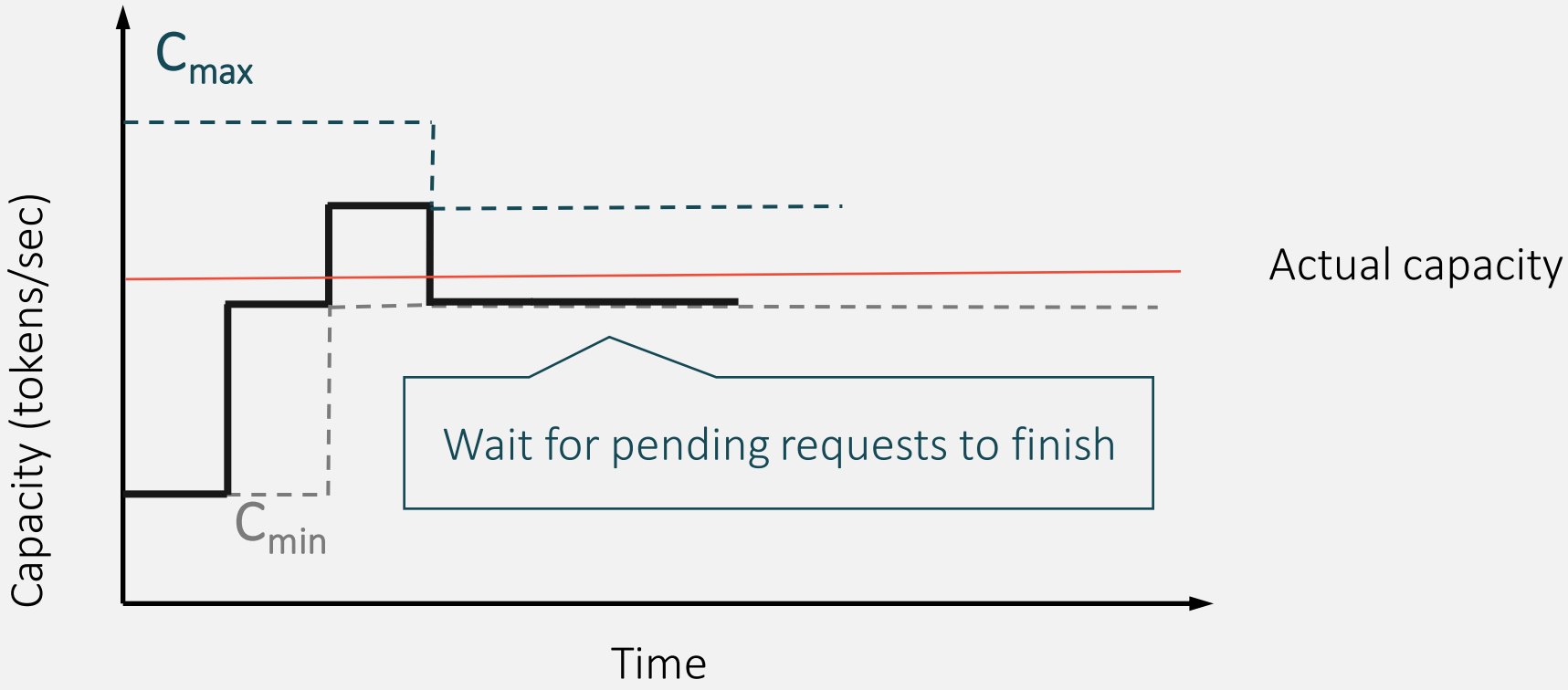
Controller-based capacity estimation



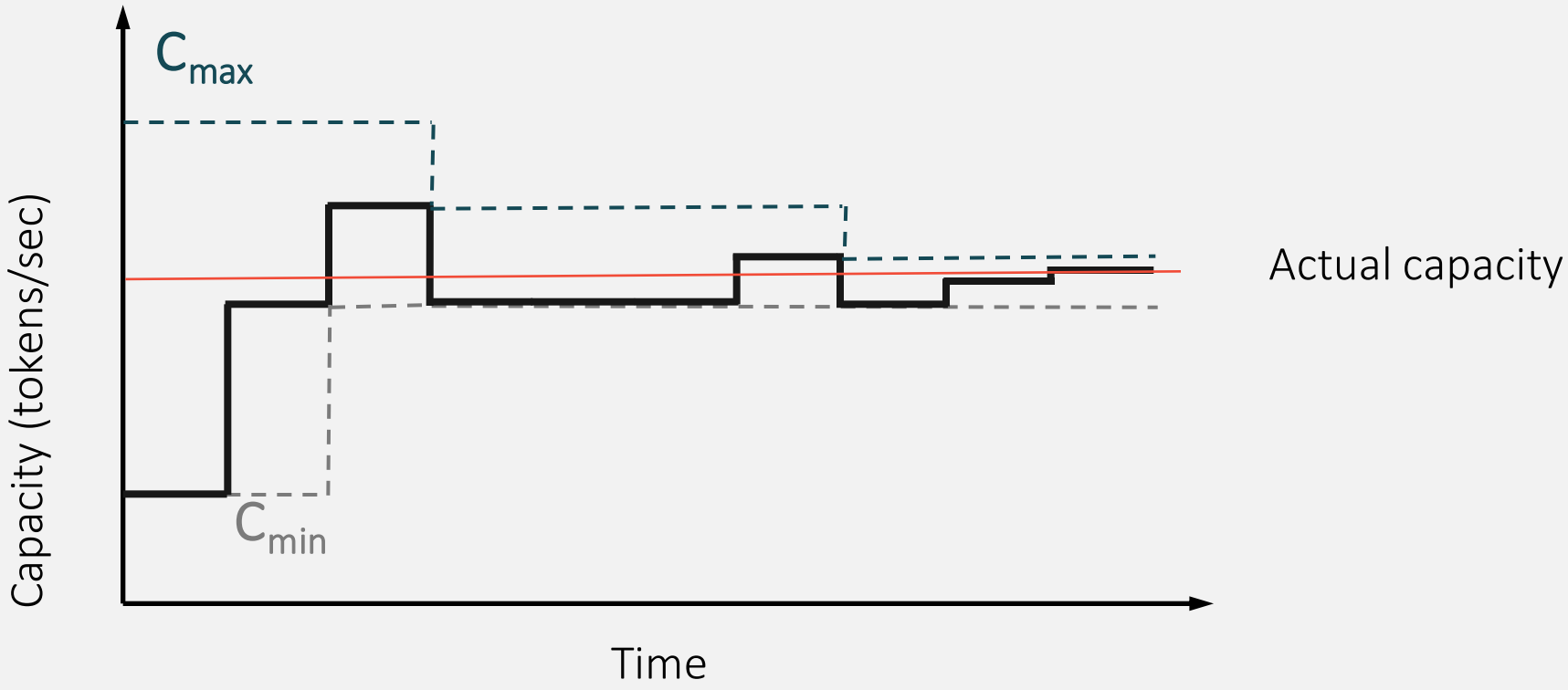
Controller-based capacity estimation



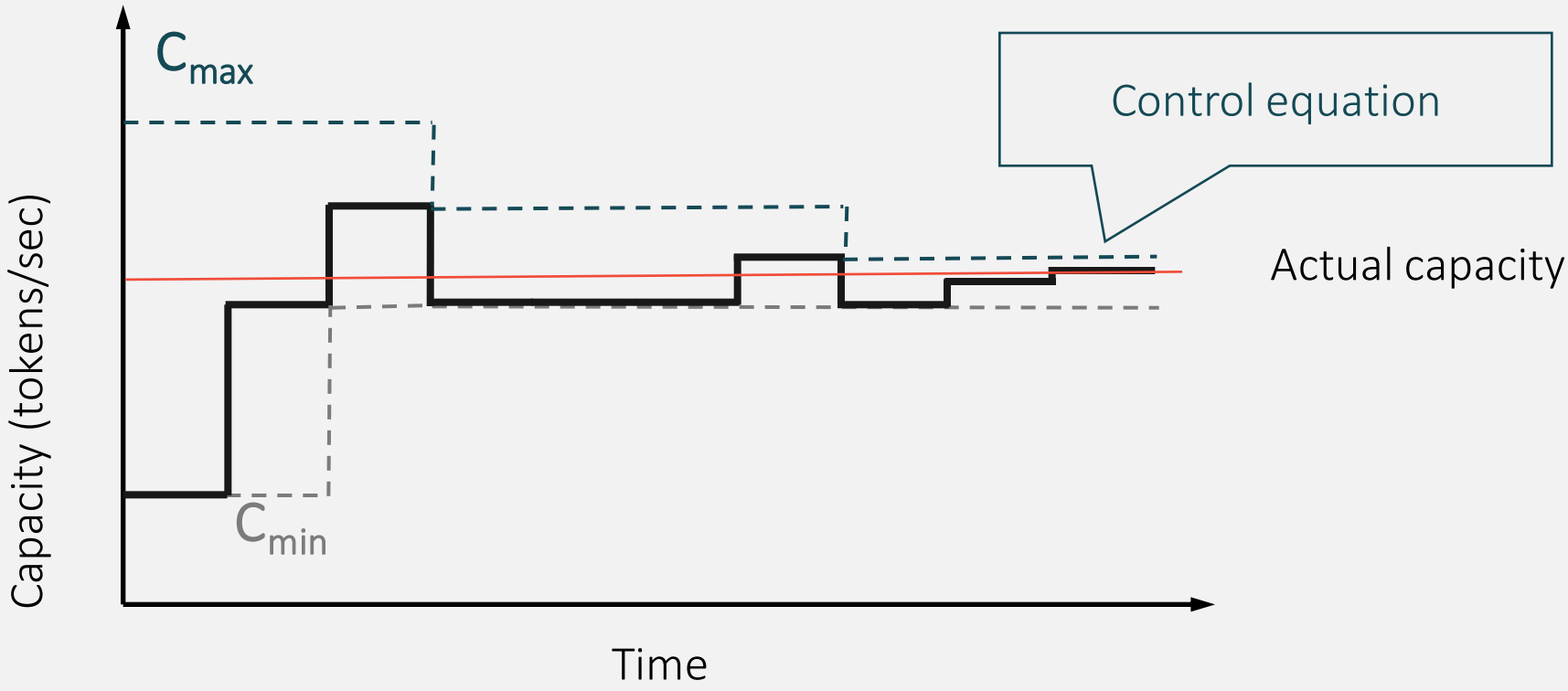
Controller-based capacity estimation



Controller-based capacity estimation



Controller-based capacity estimation



We rely on two congestion signals

- Aggregate throughput < Capacity estimate
- VDC-compliant workload
  - Helps find capacity for VDC-compliant workload
  - Detailed example in the paper!

Implementation

- Rate enforcer
  - Filter driver running on Hyper-v
  - Enforces allocations using a multi-resource token bucket
- Controller
  - Stand-alone server
  - Allocation mechanisms include DRF [NSDI'11], H-DRF [SOCC'13]
  - Installs relevant cost functions in rate enforcers

Evaluation questions

1. Can Pulsar isolate tenants and meet their guarantees?

2. Can Pulsar estimate appliance capacity?

3. What are the data- and control-plane overheads?

Experimental setup and testbed

- 3 Appliances:

In-memory key-value store Blockstore (6 SSDs) Encryption device

- Network: Mellanox 40 Gbps RDMA RoCE full-duplex

- 10 compute servers (total of 113 VMs)

Workloads and expected throughput

- 4 Tenants: A—D (tenants share at least one resource)
- Workloads are generated with parameters from Hotmail traces

Workloads and expected throughput

- 4 Tenants: A—D (tenants share at least one resource)
- Workloads are generated with parameters from Hotmail traces
- Expected throughput (tokens/second):

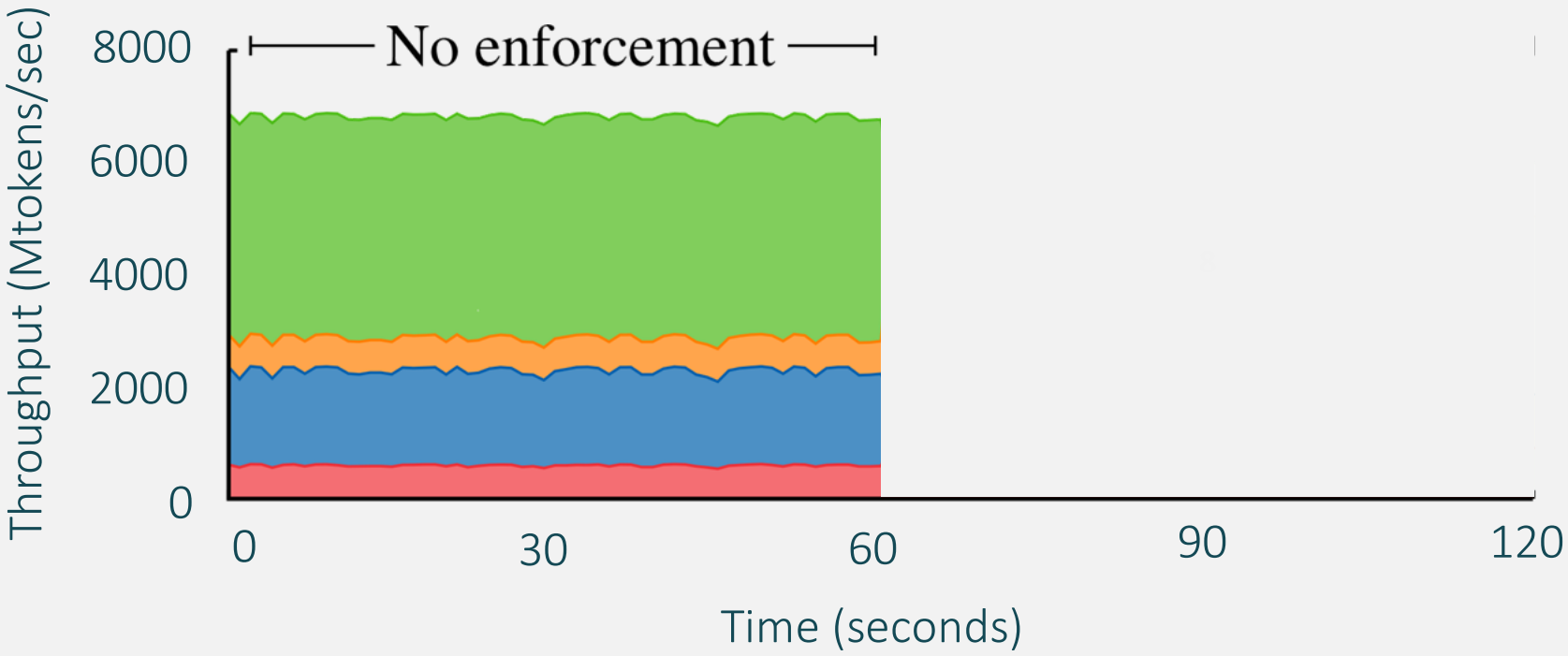
Tenant	A*	B	C*	D
Guarantee	400 M	1600 M	800 M	800 M
VMs	49 (many flows)	48	8 (large IO window)	8

Pulsar achieves tenants' guarantees

Tenant	A*	B	C*	D
Guarantee	400 M	1600 M	800 M	800 M
VMs	49 (many flows)	48	8 (large IO window)	8

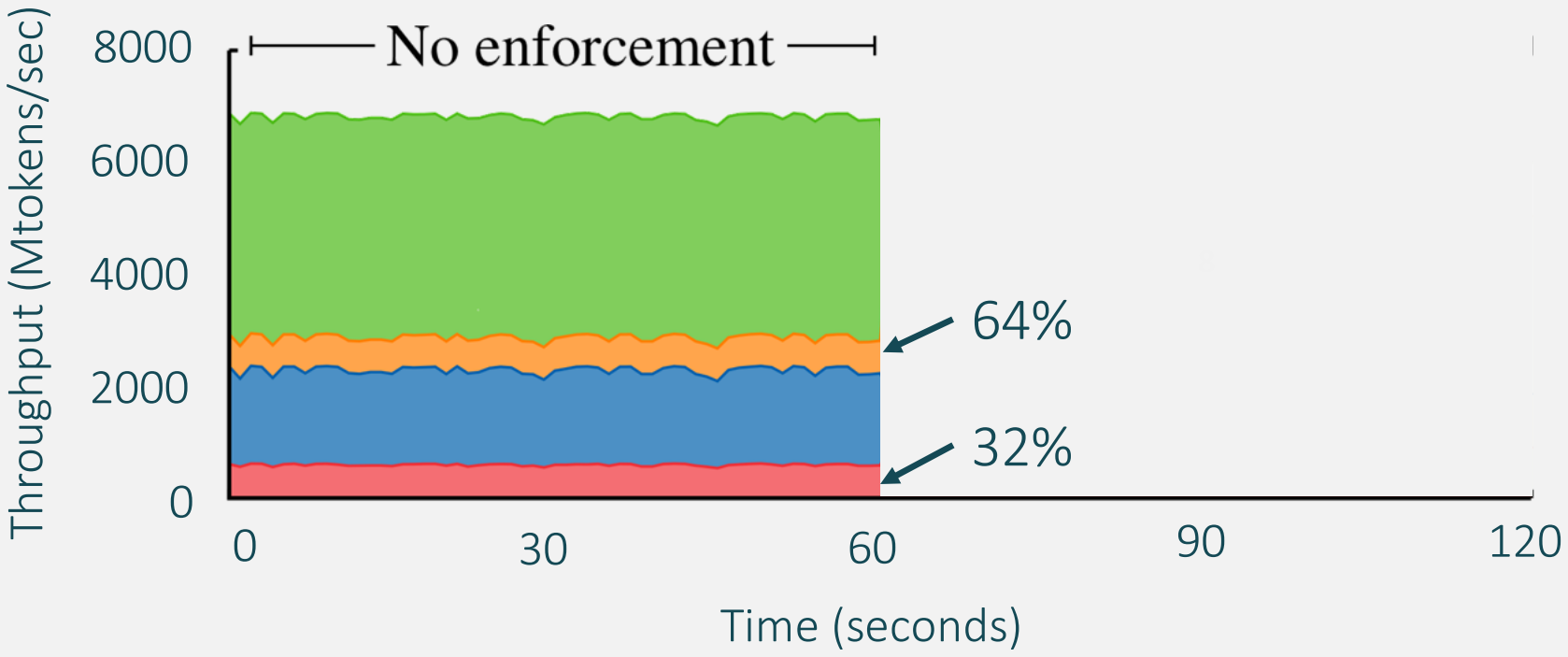
Pulsar achieves tenants' guarantees

Tenant	A*	B	C*	D
Guarantee	400 M	1600 M	800 M	800 M
VMs	49 (many flows)	48	8 (large IO window)	8



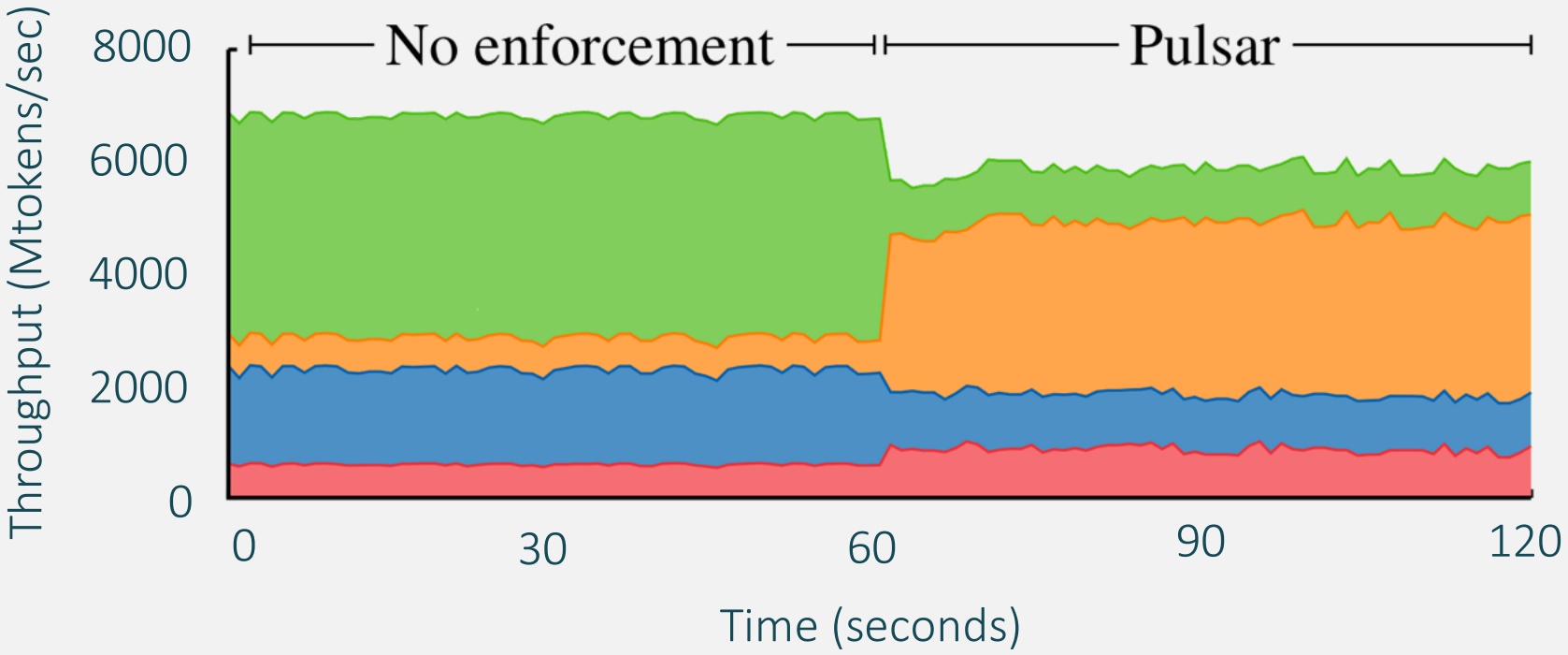
Pulsar achieves tenants' guarantees

Tenant	A*	B	C*	D
Guarantee	400 M	1600 M	800 M	800 M
VMs	49 (many flows)	48	8 (large IO window)	8



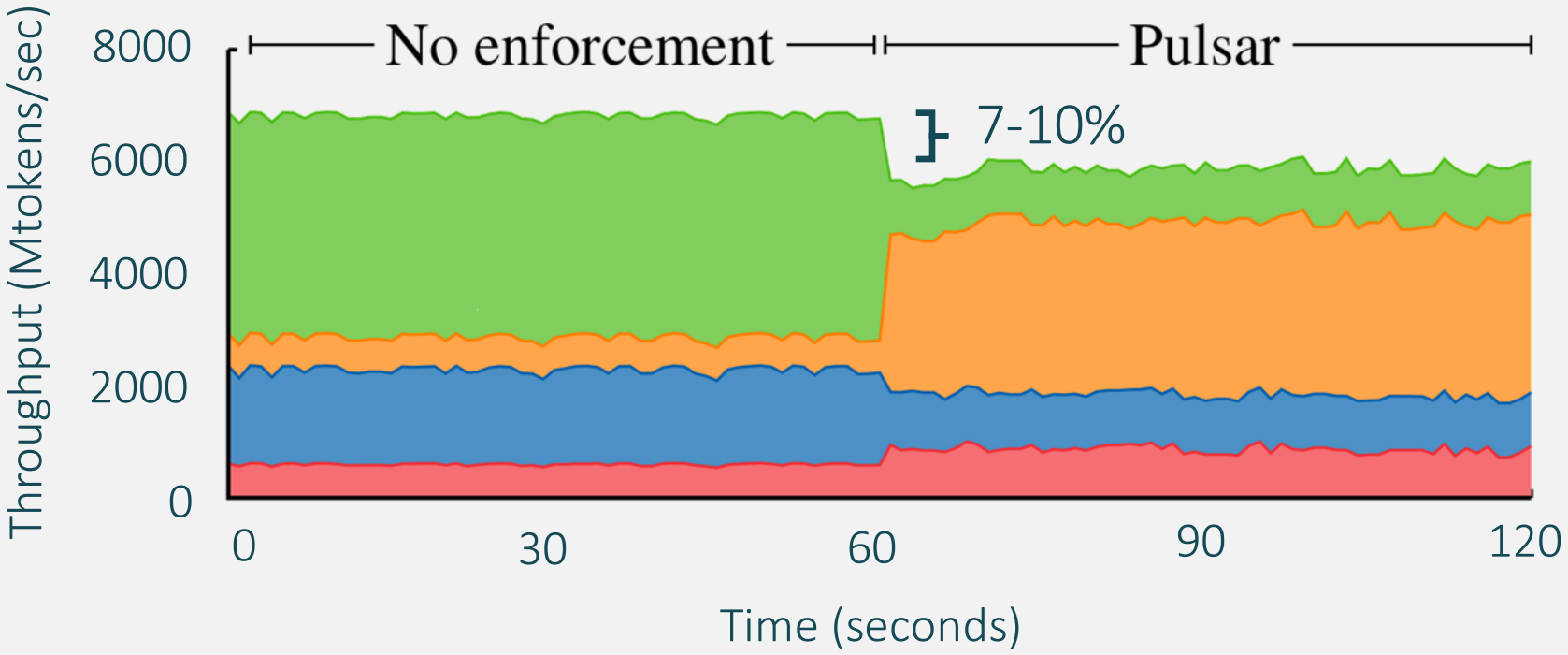
Pulsar achieves tenants' guarantees

Tenant	A*	B	C*	D
Guarantee	400 M	1600 M	800 M	800 M
VMs	49 (many flows)	48	8 (large IO window)	8



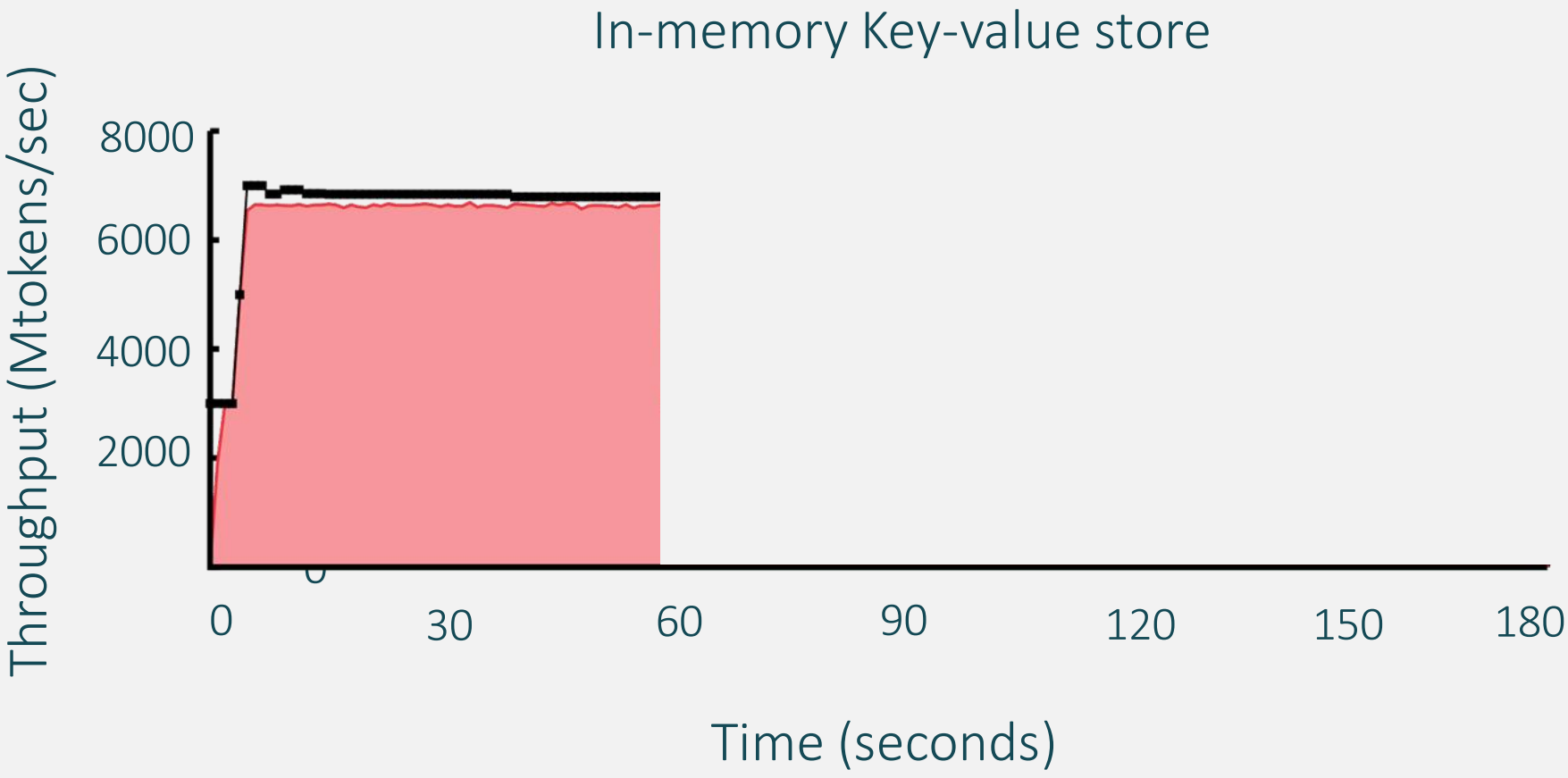
Pulsar achieves tenants' guarantees

Tenant	A*	B	C*	D
Guarantee	400 M	1600 M	800 M	800 M
VMs	49 (many flows)	48	8 (large IO window)	8

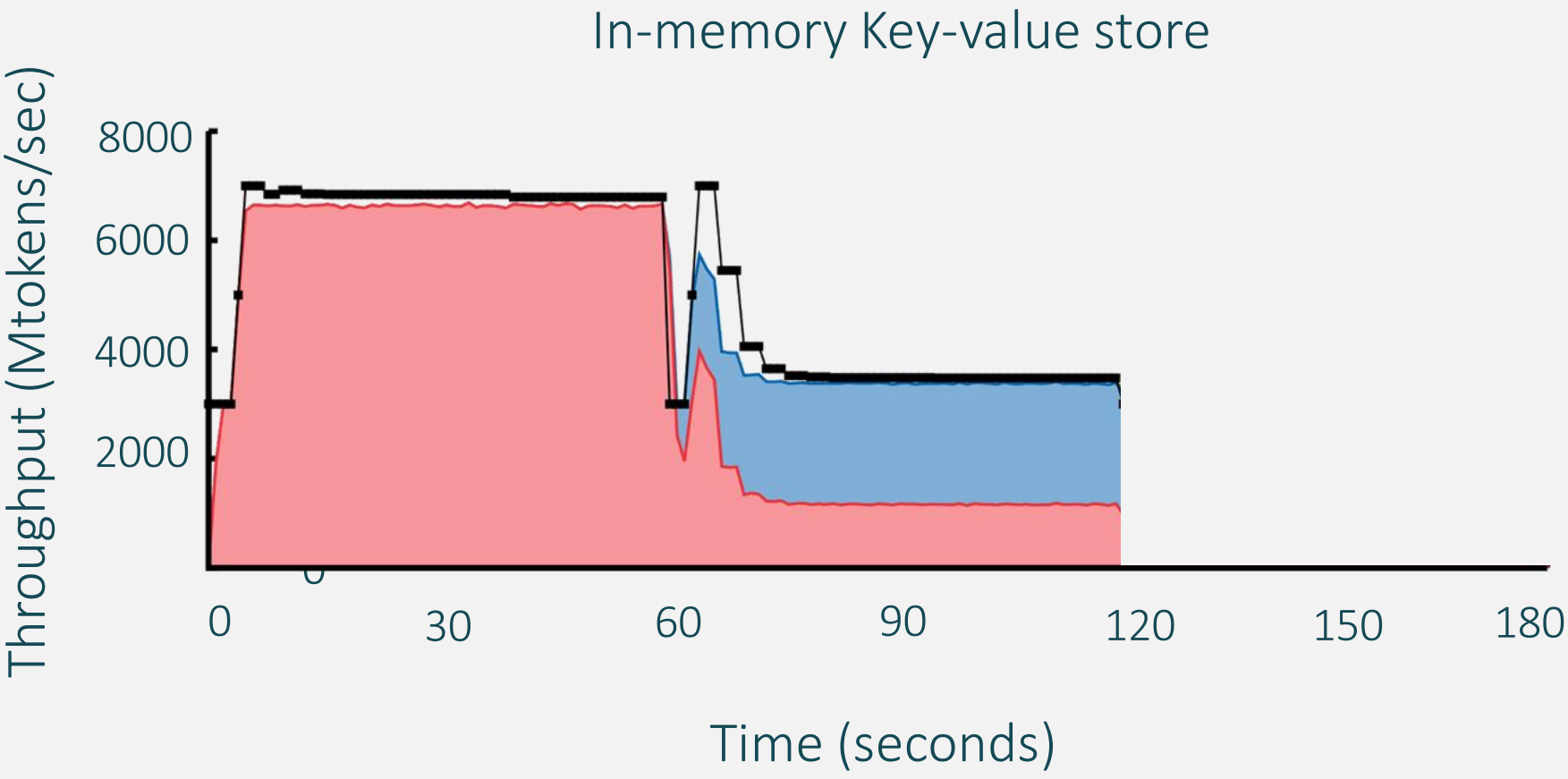


Can Pulsar estimate appliance capacity?

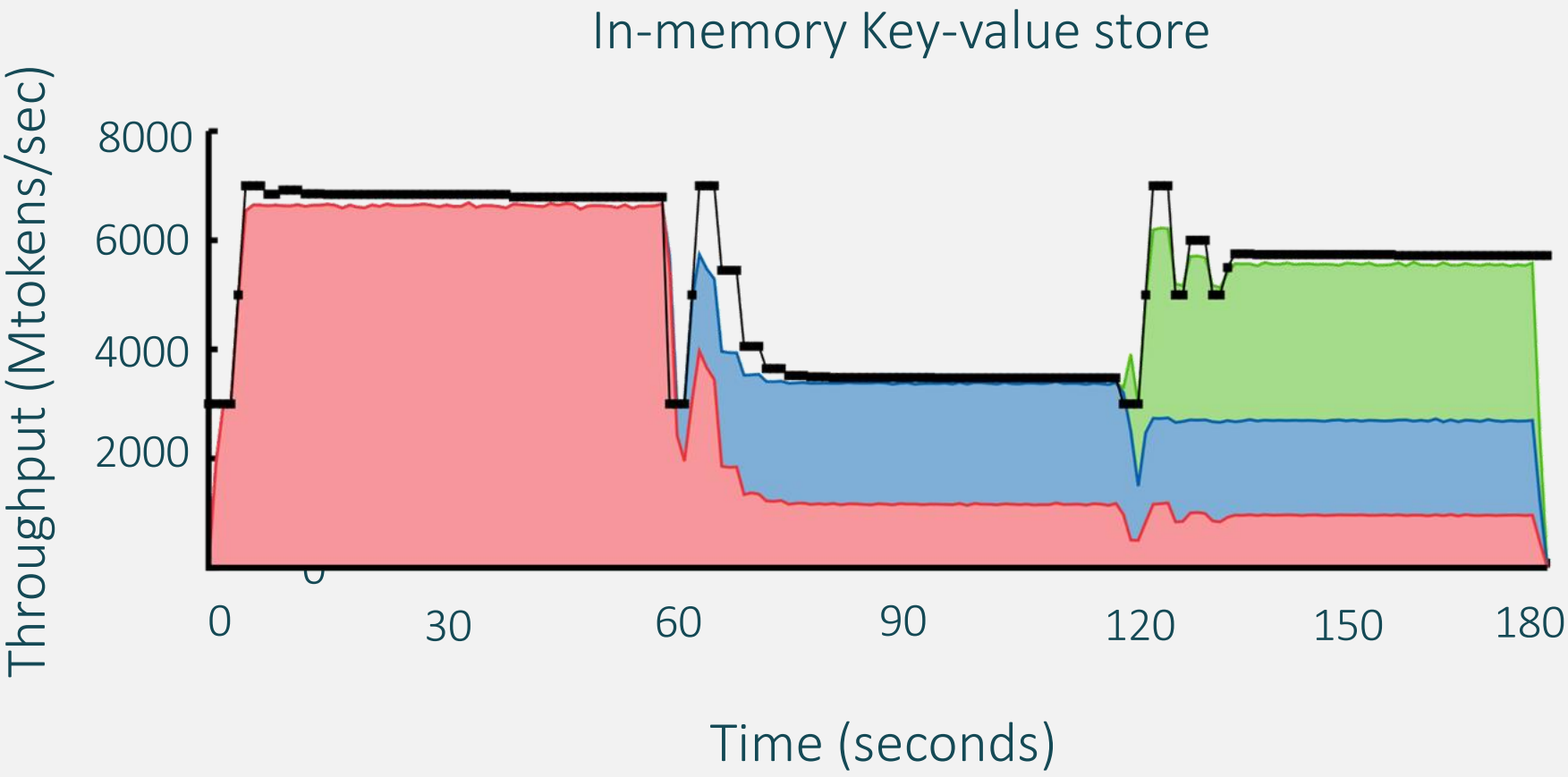
Pulsar estimates capacities and copes with changing workloads



Pulsar estimates capacities and copes with changing workloads



Pulsar estimates capacities and copes with changing workloads



What are the data- and control-plane overheads?

Data- and control-plane overheads are reasonable

Data-plane

- Overhead from rate enforcer < 2% (15% for small requests)

Control-plane

- 256 bytes/sec for each VM
- Setting up cost functions at rate enforcers takes 83 µs
- Can compute rich policies for 24K VMs and 200 appliances

Summary

- Virtual Datacenter (VDC) abstraction
  - Captures tenants' end-to-end throughput guarantees
- Pulsar implements the VDC abstraction
  - Simple data-plane rate limiting and centralized control-plane
  - No changes to appliances, switches, guest OSes, and apps
  - Reasonable data- and control-plane overheads
- See you at the poster session!