The Performance of Distributed Applications

A Traffic Shaping Perspective

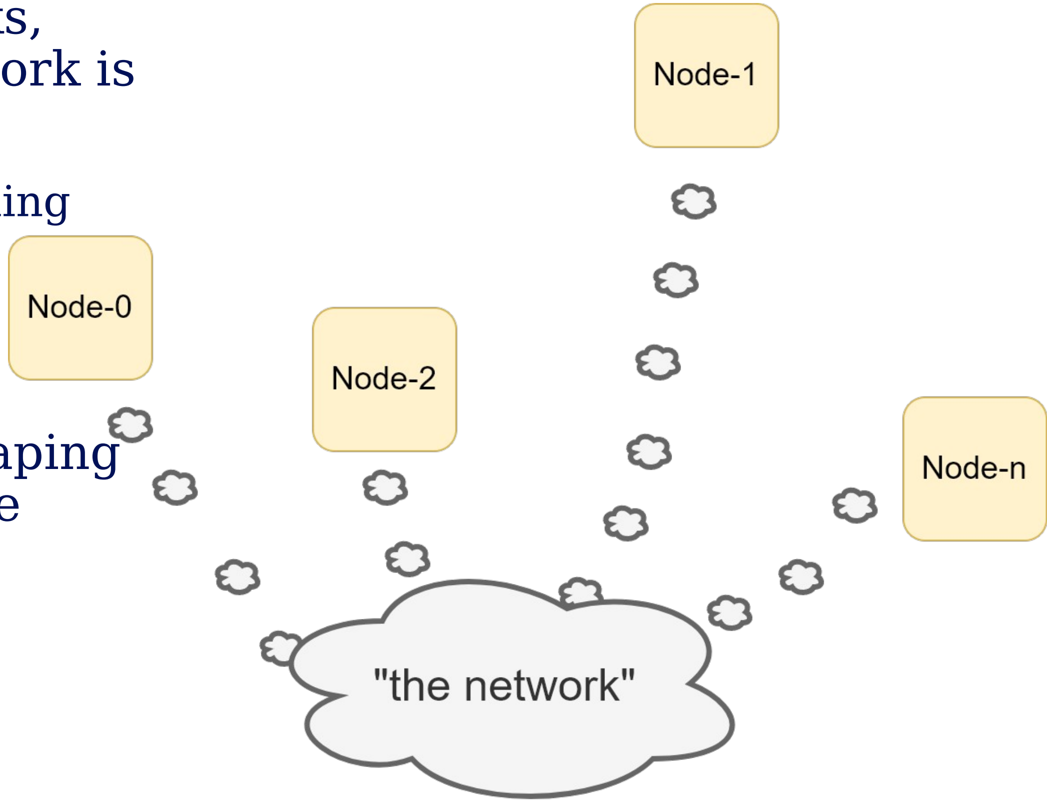
Jasper Hasenoot, Jan S. Rellermeyer, Alexandru Uta

ICPE ’23, April 15–19, 2023, Coimbra, Portugal

Universiteit Leiden
The Netherlands

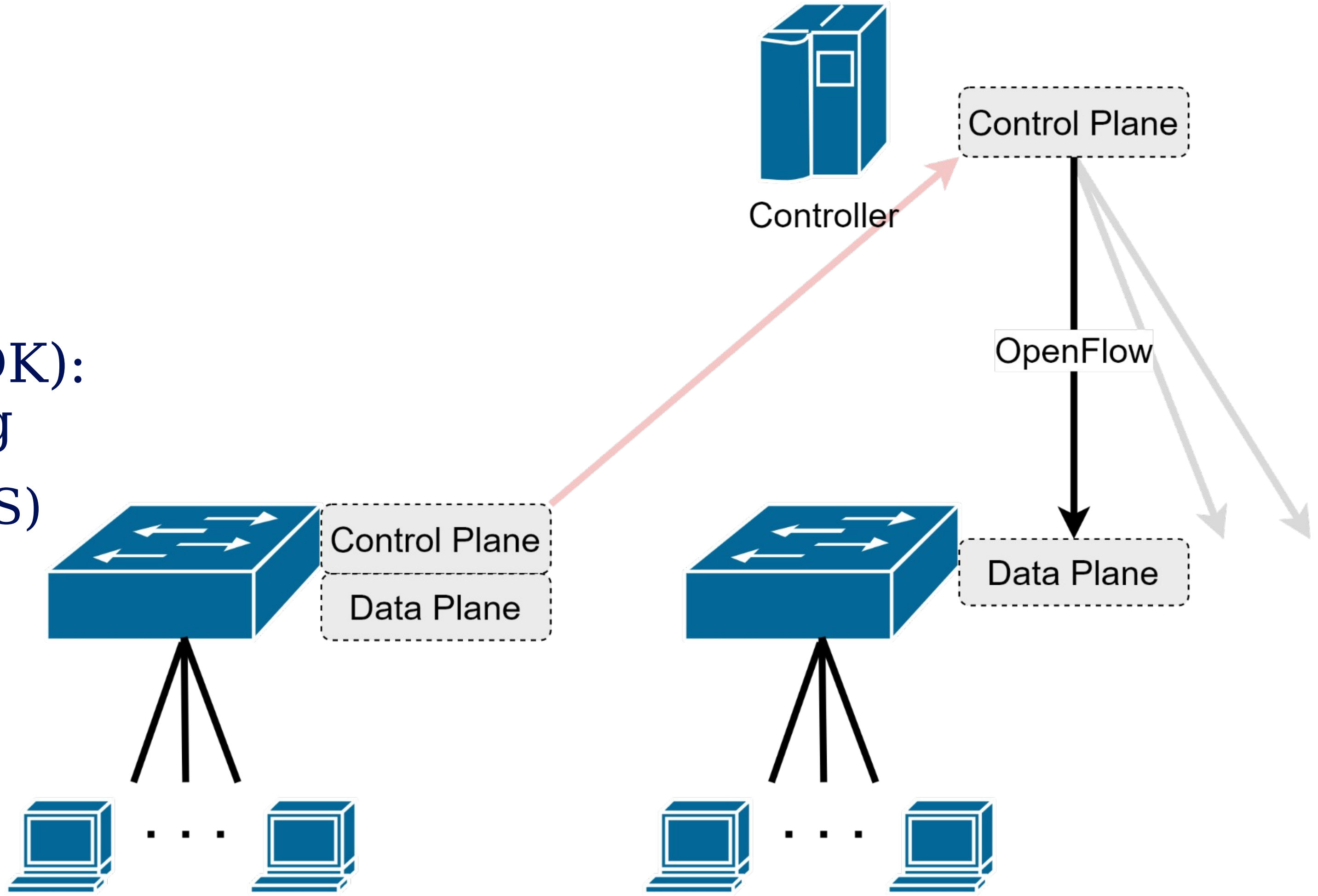
Context

- Network is integral to the functioning of *distributed* applications
- However: In (scientific) Cloud experiments & benchmarks, contention and performance impact concerning the network is often ignored
  - This is contrary to contention caused by other tenants concerning e.g. Disk or CPU
  - Cloud providers have no “Bandwidth guarantee”
- To alleviate the impact of network contention, Traffic Shaping is used, though this may also exacerbate the performance impact



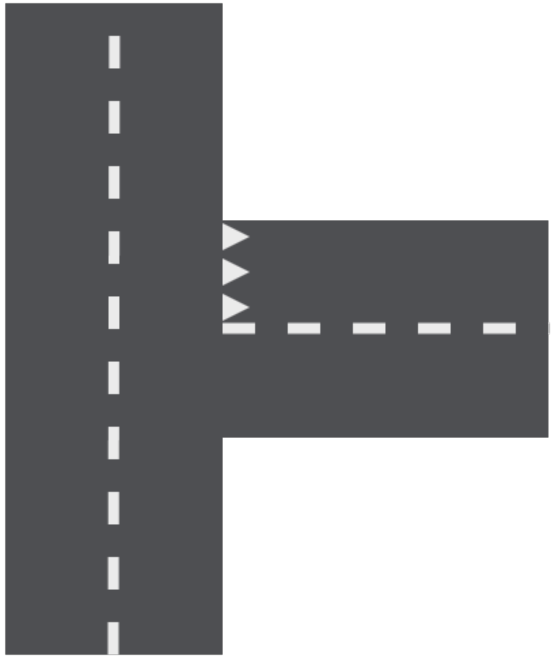
Background

- Software Defined Networking (SDN)
  - Control Plane
  - Data Plane
  - OpenFlow
  - Physical/Virtual Switch
- Data Plane Development Kit (DPDK): Kernel bypass network processing
  - Can be added to Open vSwitch (OVS)

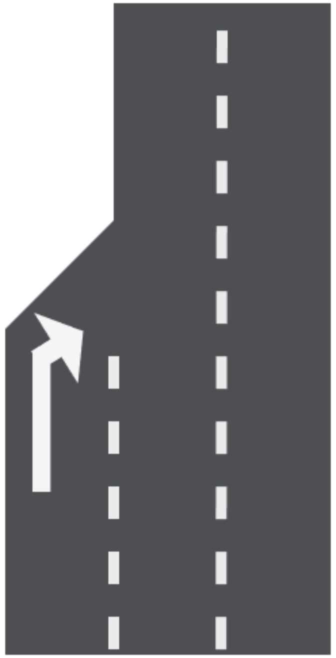


Background

- Traffic Shaping: A general idea.
  - Priority Queue
  - Token Bucket
- Port-by-port basis on switch
  - Allocated priority: Relative bandwidth
  - Allocated bandwidth: Absolute bandwidth



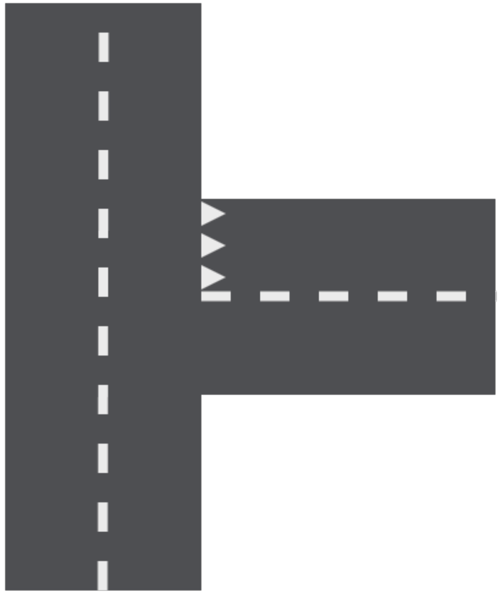
“Priority Queue”



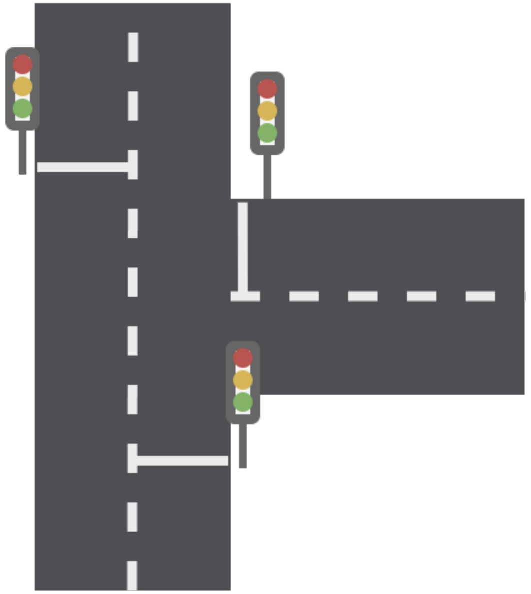
“Token Bucket”

Priority Queue

- Priority marked in IP headers using Differentiated Services Code Point (DSCP)
- Codes for:
  - 4 tiers in priority (has precedence)
  - 3 tiers in drop probability
- Priority:
  - Strict
  - Weighted



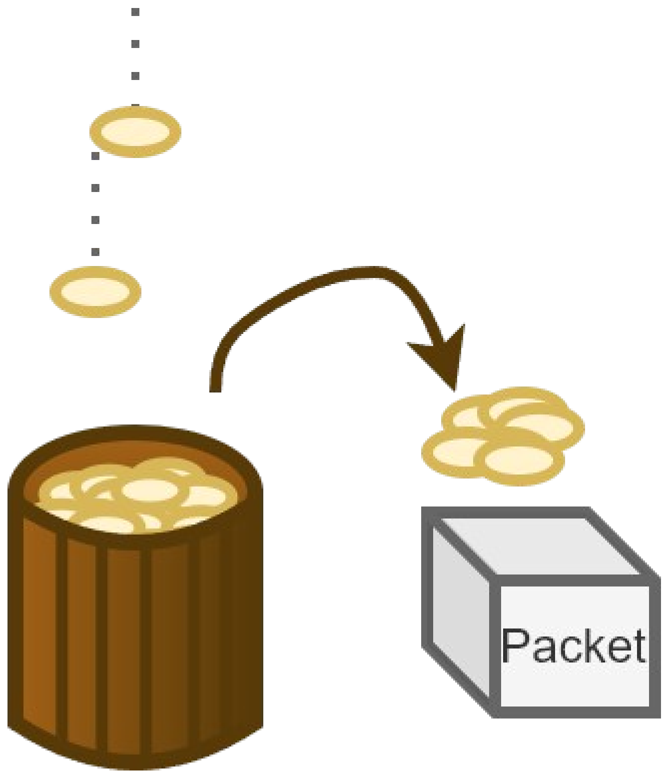
“Strict”



“Weighted”

Token Bucket

- Maximum bucket size in bytes/packets: the “tokens”
  - Consumed by packets when they are transmitted
- Refill rate of the bucket
  - Fills up to the bucket size, otherwise is discarded
- Maximum average bandwidth (guaranteed) is determined by refill rate



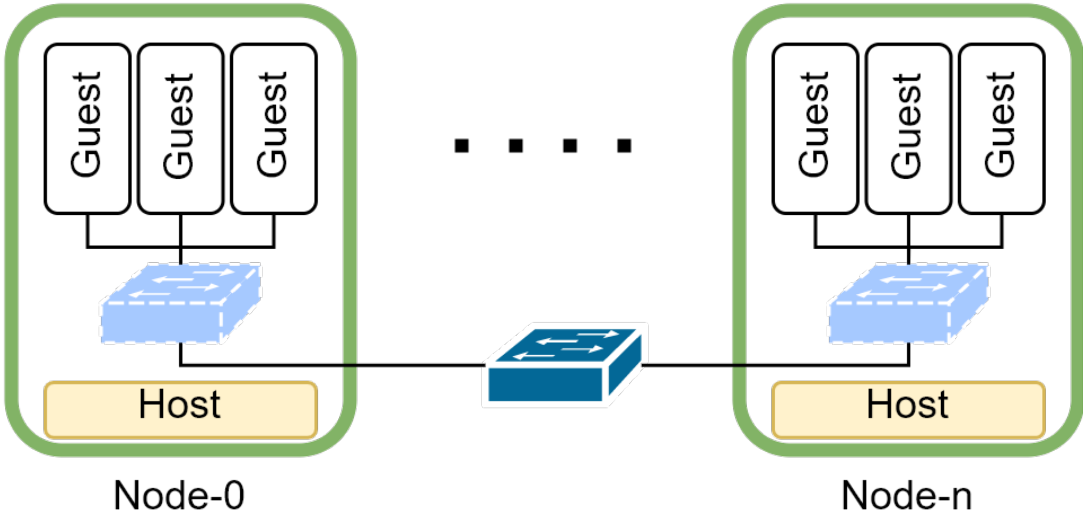
How to quantify the effects of traffic shaping on applications?

- Distributed Applications covering multiple domains
- Standardized benchmarks measuring multiple facets

Distributed Application	Specific Application	Benchmark
Key/Value Store	MongoDB	YCSB
Big Data Workload	Apache Spark	HiBench
HPC Workload	OpenMPI	HPCC

Cloud Model – Experiment setup

- Virtual & physical switches
- Docker overlay network connecting guests
- Separate control network
- Course/fine granularity traffic shaping depending on location
  - Trade-off between granular control & CPU usage

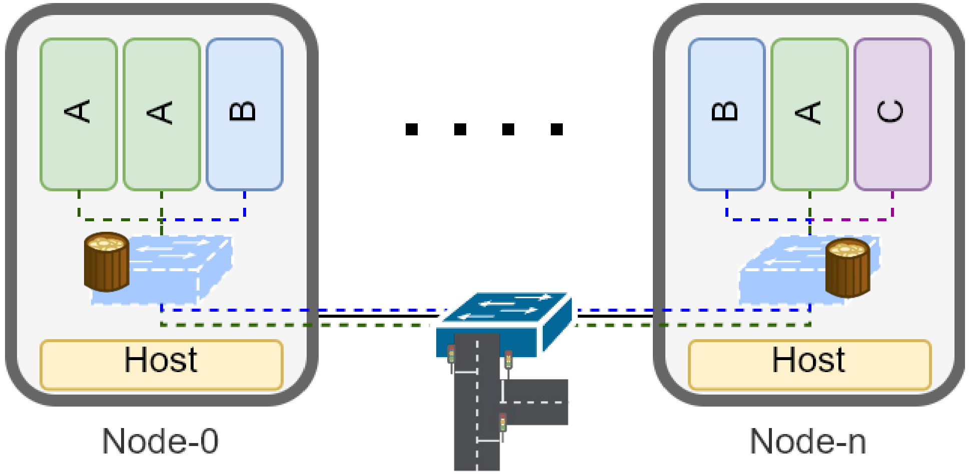


What is the effect of traffic shaping on distributed applications?

Experiment Design

- General idea:
  - Use standardized benchmarks on applications
  - Subject them to traffic shaping and network interference & contention

- This shows both the effect of network interference, and the added effect of traffic shaping



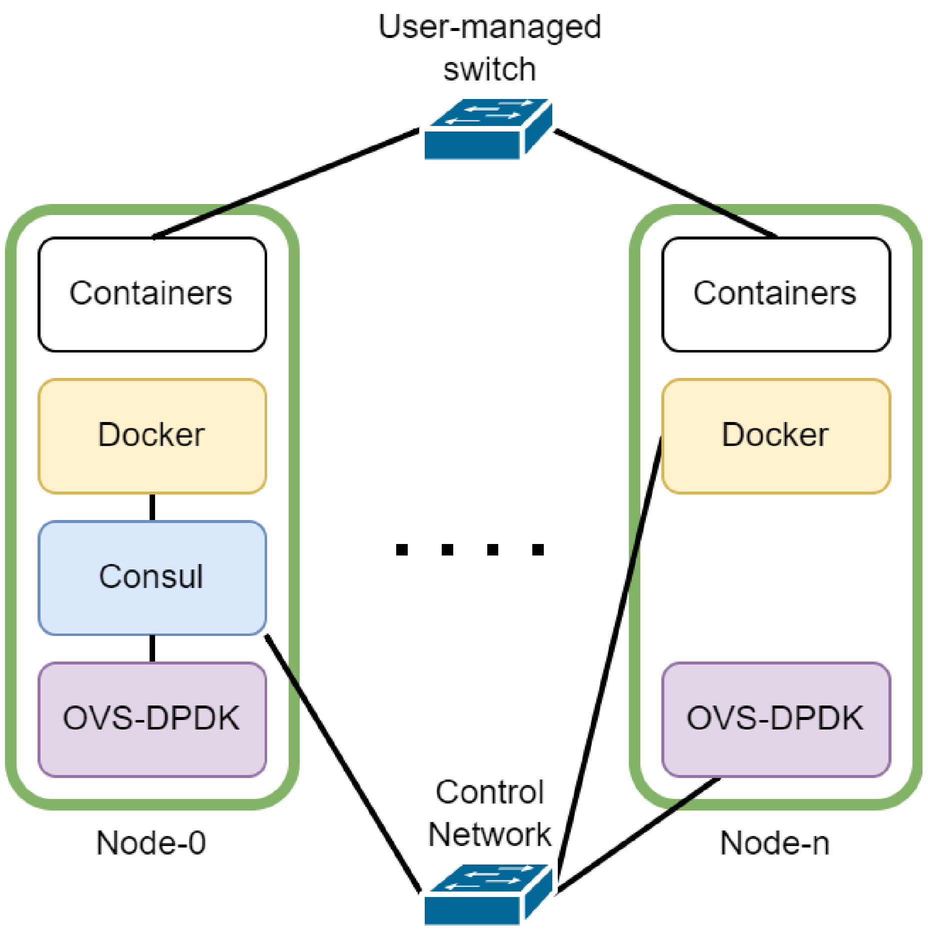
Traffic Shaping Benchmark Configurations

Host (vSwitch)	Switch (Physical)
None*	None*
None	None
None	Token Bucket
None	Priority Queue
None	Token Bucket & Priority Queue
Token Bucket	None
Priority Queue	None
Token Bucket & Priority Queue	None
Token Bucket	Priority Queue
Priority Queue	Token Bucket

* no interference traffic

Experiment Setup

- Based on simplified Cloud model (shown previously)
- Use Docker & Containers for virtualisation
  - Docker supports a virtual “Overlay Network” spanning multiple nodes, uses the *User-Managed* network
  - Containers are added to this network through *docker-compose*
  - MPI: Containers run in privileged mode in the host IPC namespace
- Consul Key/Value store keeps track of network state
  - Communicates with Docker instances over the *Control Network*
- OVN Docker Overlay Driver *(implied)* translates Docker commands to OpenFlow to program OVS-DPDK vSwitch



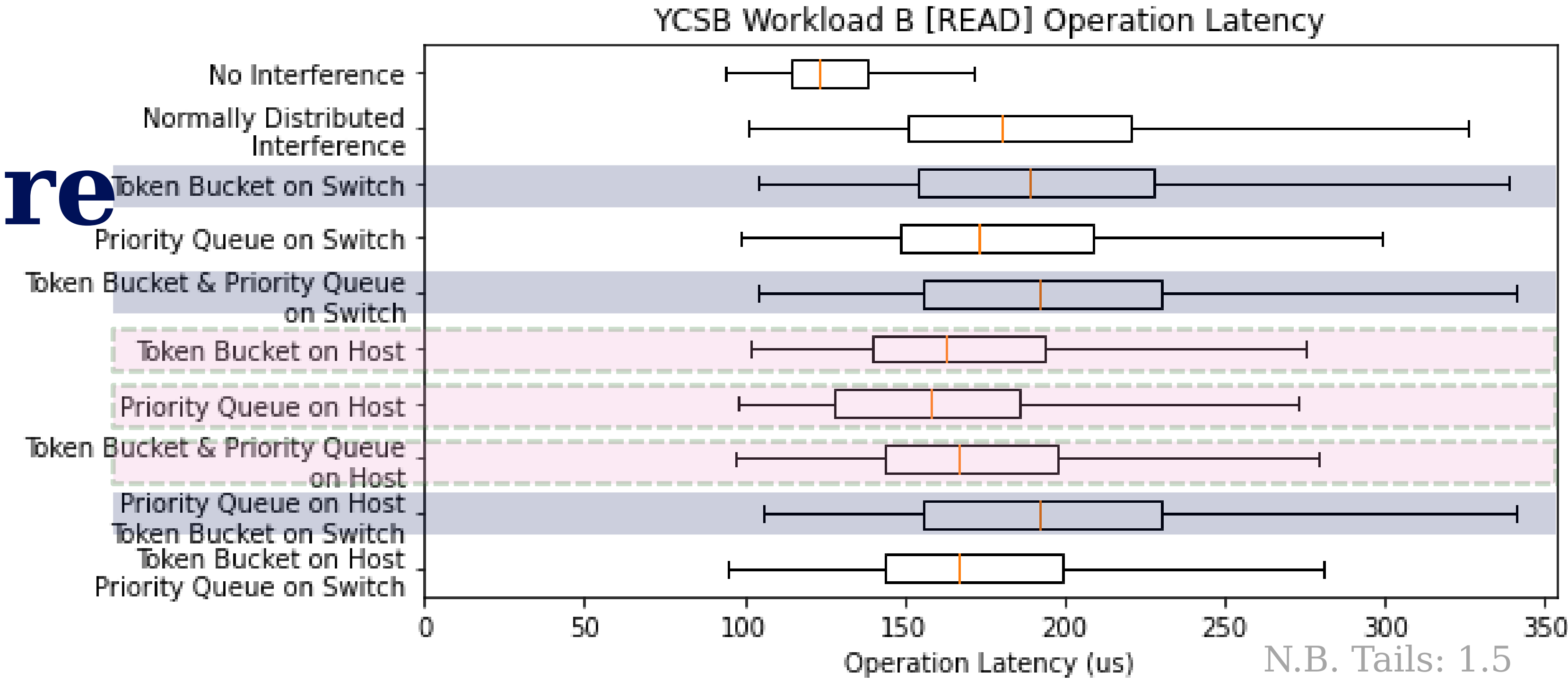
Experiment Setup

Category	Benchmark	Settings	Sub-benchmarks	Repeats
Key/Value Store	YCSB	Records: 1,000,000	A-F	1,000,000 operations
Big Data Workload	HiBench	Dataset size: 300,000,000	Terasort	10
HPC Workload	HPCC	Default	HPL, DGEMM, STREAM, PTRANS, RandomAccess, FFT, Latency/Bandwidth	100

Result Format

- Box plot
  - Whiskers: 1.5 IQR
  - No outliers, 95th, 99th and 99.9th percentiles shown in tables instead

Results: Key/Value Store



N.B. Tails: 1.5 IQR

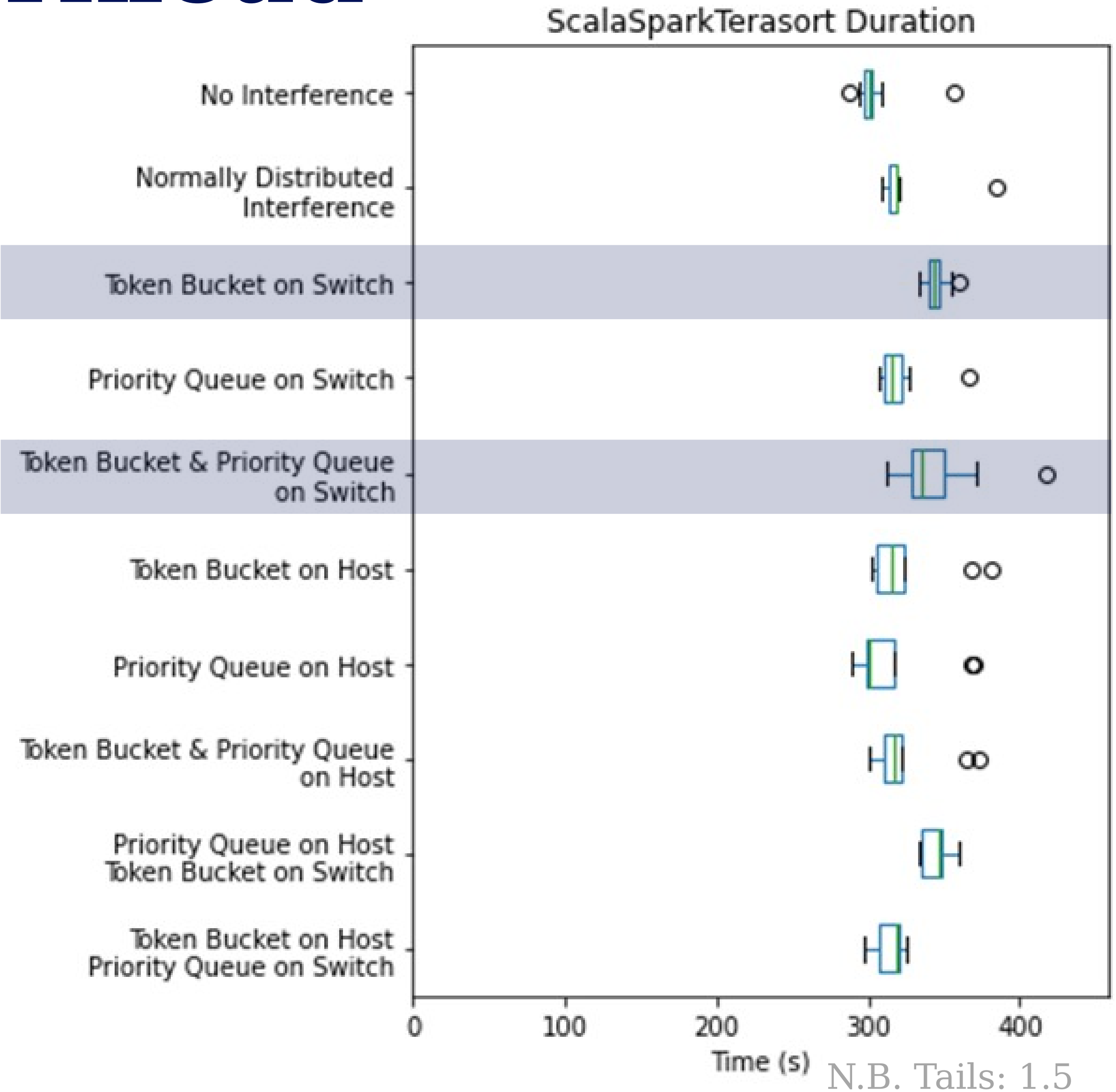
- Traffic shaping:
  - Overall reduction in variance
  - Increase in variance in on-switch token bucket
  - Decrease in variance in on-host shaping

Experiment Runtime (µs)	**P95**	**P99**	**P99.9**
No Interference	246	360	551
Normally Distributed Interference	440	633	937
Token Bucket on Switch	430	**205695**	**211199**
Priority Queue on Switch	389	536	759
Token Bucket & Priority Queue on Switch	429	**205823**	**211071**
Token Bucket on Host	394	856	1947
Priority Queue on Host	318	449	681
Token Bucket & Priority Queue on Host	391	653	1708
Priority Queue on Host, Token Bucket on Switch	428	**205695**	**210687**
Token Bucket on Host, Priority Queue on Switch	392	626	1610

- On-switch token bucket has very large tail latencies
- Only priority queues have a slight decrease in tail latency

Results: Big Data Workload

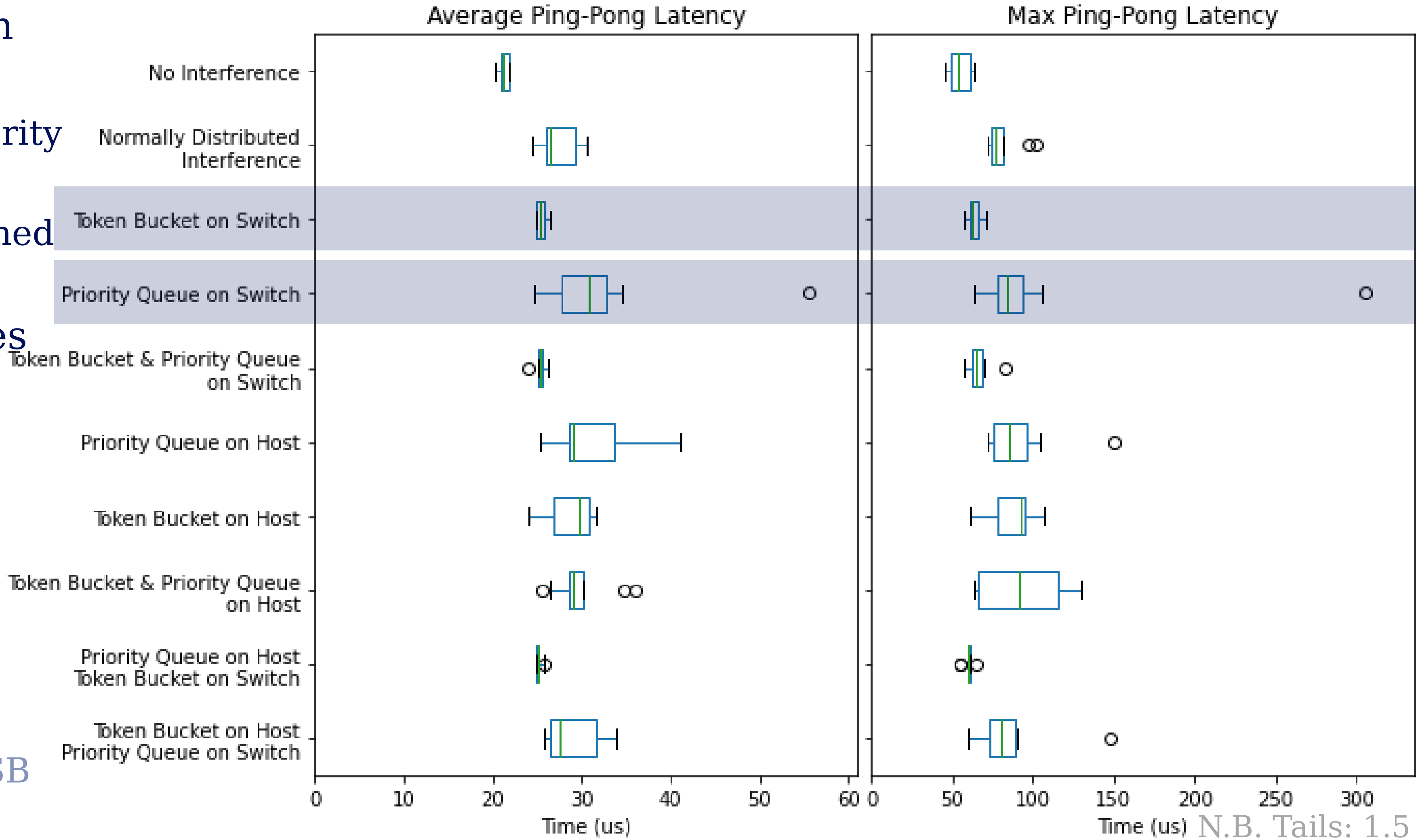
- On-switch token bucket increases duration of Terasort, variance is similar
- Increase in variance when using both on-switch priority queue and token bucket
- Other measures have little effect



N.B. Tails: 1.5 IQR

Results: HPC Workload

- Reduction in variance & overall latency in on-switch token bucket
  - Increased effect when a priority queue is added
  - YCSB: Token Bucket performed worse
- Priority queue by itself does not show this behaviour

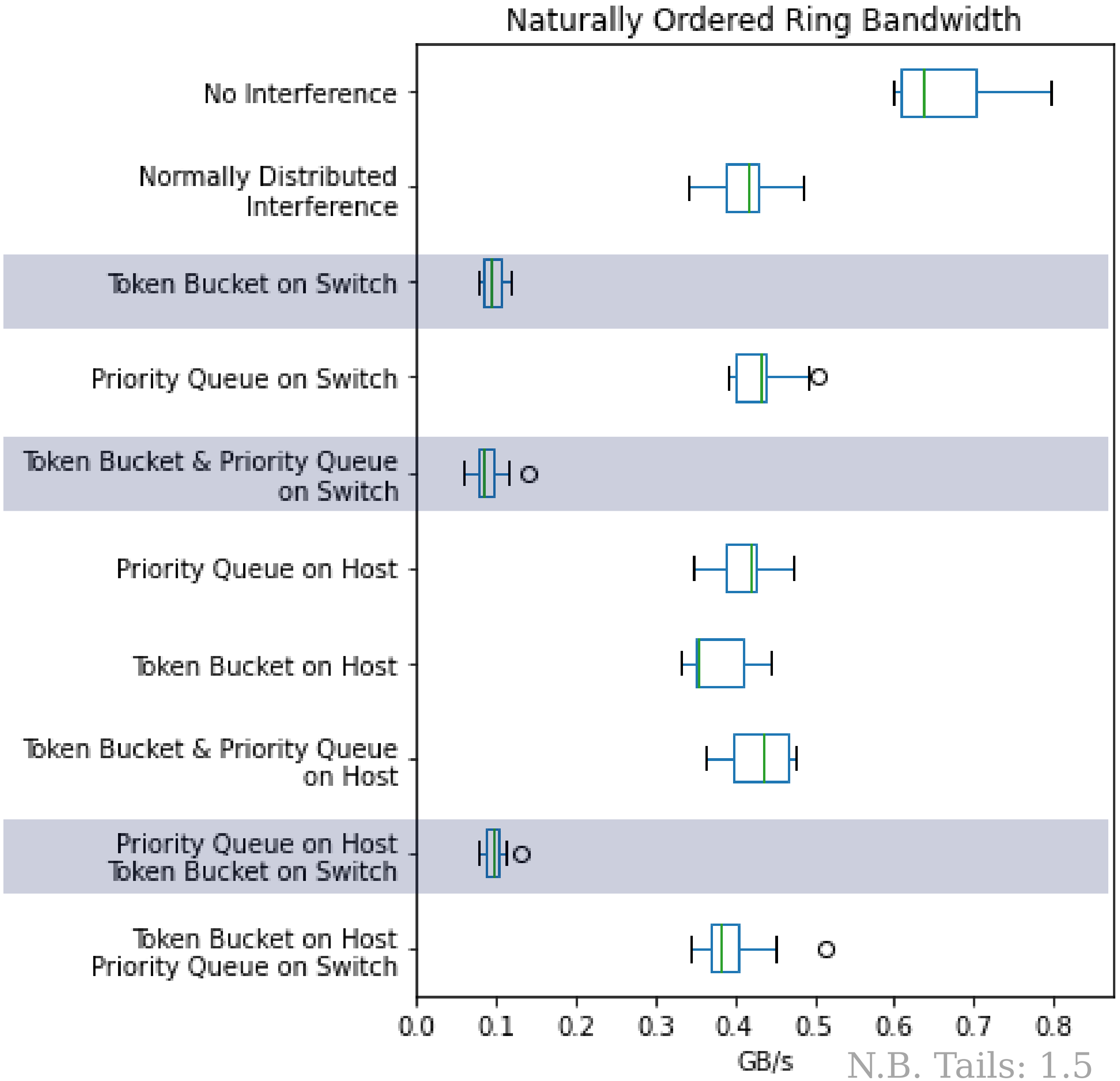


Ping-Pong between all node combinations, using 8 byte Ethernet frames compared to 1500 byte frames used by YCSB

N.B. Tails: 1.5 IQR

Results: HPC Workload

- On-switch token bucket causes a large decrease in bandwidth, due to contention with interference traffic
- This is not observed on the on-host token bucket: granular control can be important
  - N.B.: Token bucket settings are identical between the two switches



N.B. Tails: 1.5 IQR

Practical Implications

What is the effect of traffic shaping on distributed applications?

It depends on the application, its network usage and packet size, as well as the traffic shaping used.
There is no such thing as a free lunch.

Take aways and recommendations:

1. Benchmark the to-be-deployed application
   - Compare different (private) cloud environments, different node types
   - Exert any influence possible over the network
2. On-switch Token Buckets negatively impact tail latencies of *many* applications
3. Applications with small IP packets may benefit from Token Buckets
4. Consider the assumptions made about the network
5. Design experiments taking cloud variability into account

The Performance of Distributed Applications

A Traffic Shaping Perspective

Universiteit Leiden
The Netherlands