# Preventing Network Bottlenecks:
# Accelerating Datacenter Services with Hotspot-Aware Placement for Compute and Storage

NSDI 2025

contact: bazzaz@google.com

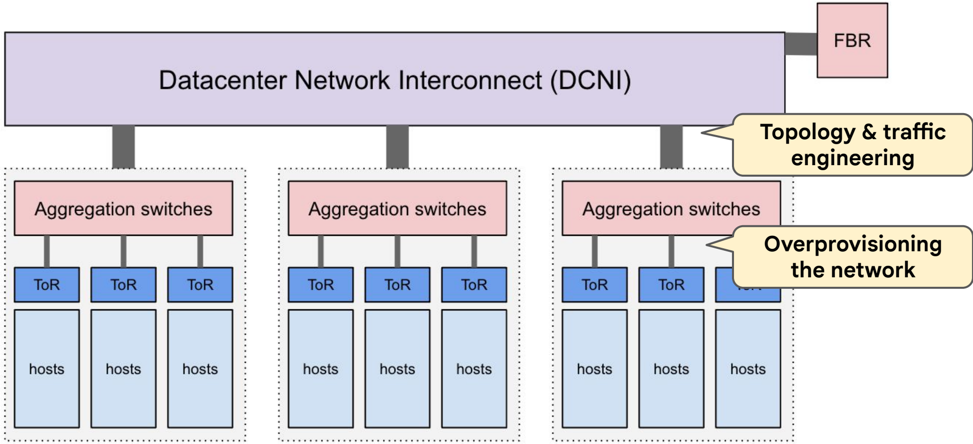
Google

# Introduction to Network Hotspots

- Goal: provide applications with the illusion of an unconstrained network.

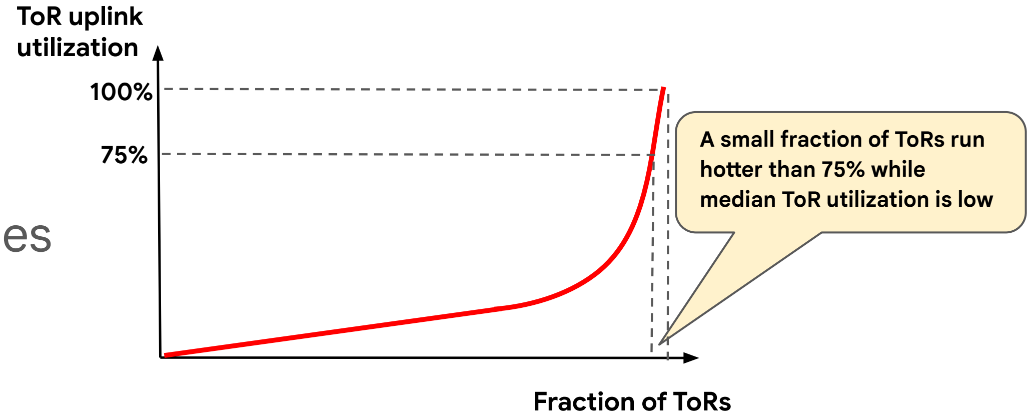
- In practice *network is not unconstrained always & everywhere.* Data centers have *hotspots*: ToRs with persistently high uplink utilization

- When all/most uplinks of such ToRs run at high utilization, congestion control and load balancing techniques are insufficient.



Topology & traffic engineering

Overprovisioning the network



A small fraction of ToRs run hotter than 75% while median ToR utilization is low

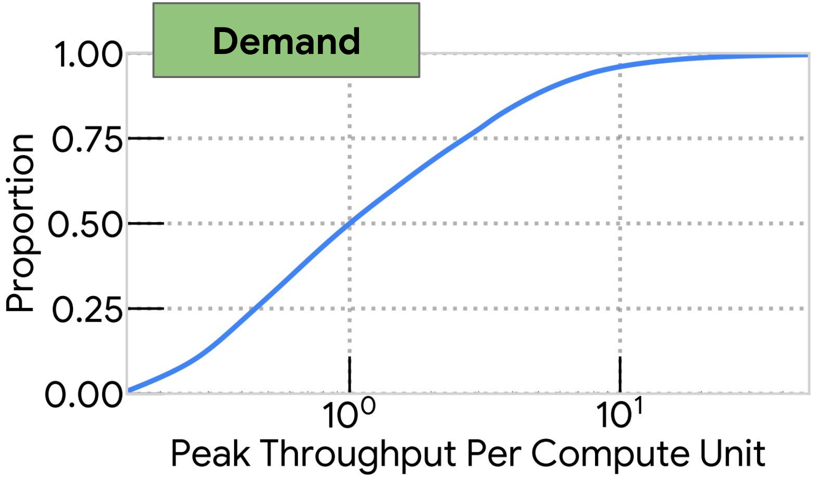
# This Research: “Landscape of Hotspots”

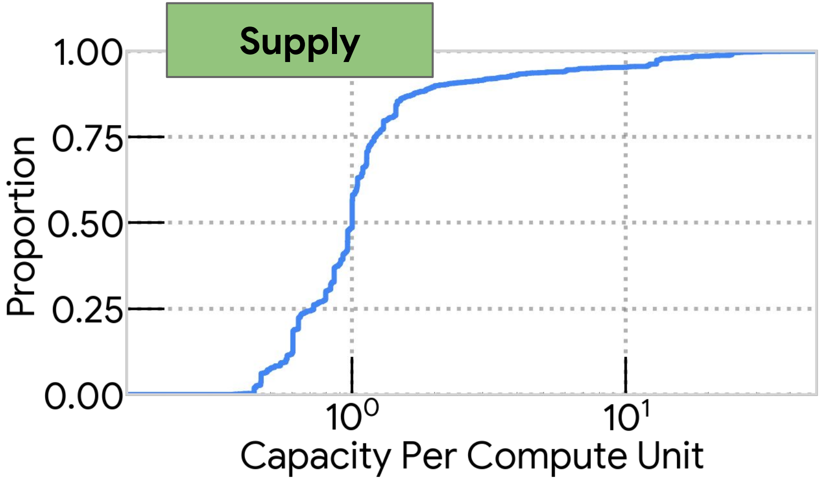
- **Characterisation**: Root causes, measurement & production incidents
- **Impact to Storage operations:** How much hotspots impact storage operations performance?
- **Hotspot-aware Placement:** Scheduling strategies for addressing hotspots

# Characterisation

Root causes, quantification & production incidents

# Why Hotspots Happen? An Intuitive Explanation





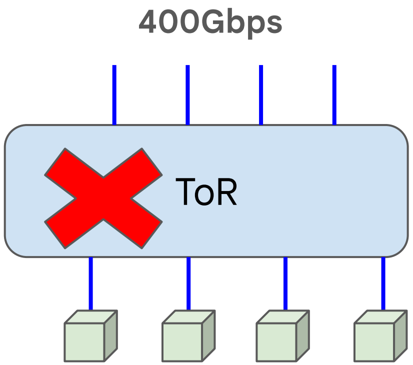
- Workload network demand and ToR network capacities exhibit heterogeneity.
- Bandwidth-agnostic task placement (scheduling) will inevitably creates **rack bandwidth demand-supply imbalance** -- leading to hotspots.
- A more detailed explanation lies in how ToR bandwidth is capacity planned.

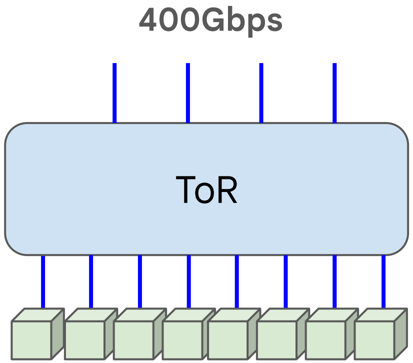
# Why Hotspots Occur? A Capacity Planning Perspective

- Datacenter capacity planning, for cost-efficiency, over-subscribes ToR uplinks, balancing ToR bandwidth supply with anticipated compute and storage traffic demand.

- ToR provisioning, fixed at deployment for its multi-years lifecycle in a dynamic Cloud, depends on workload and network usage predictions.

- Demand or supply deviations from initial forecasts create imbalances, leading to hotspots.
  - Demand
    - Inorganic growth or efficiency improvements of network-heavy workloads
    - Shifting workload mix
  - Supply
    - [Network] (Un)planned ToR uplink outages temporarily reducing bandwidth
    - [Compute/Storage] Deploying faster machines / appliance without matching ToR upgrades





ToR downlink to uplink ratio is oversubscribed

# Example Hotspot Incidents within Google

- Incidents illustrating how changes in demand or supply lead to hotspots.
  - Unintended capacity reduction causes imbalance
  - Borg constraint-induced network imbalance

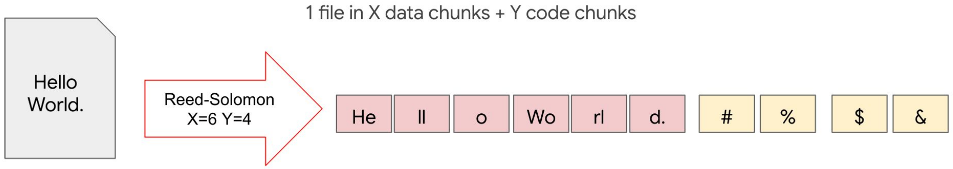
- Incidents illustrating the qualitative / quantitative impact of hotspots on applications.
  - Storage ToR hotspots cause high-level applications pain
  - Colossus degraded reads exacerbate ToR hotspots

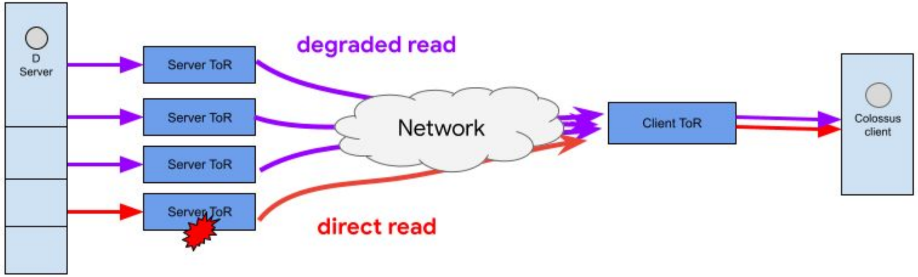
# Incident: Application’s High Latency Recovery Mechanism Could Worsen Hotspots

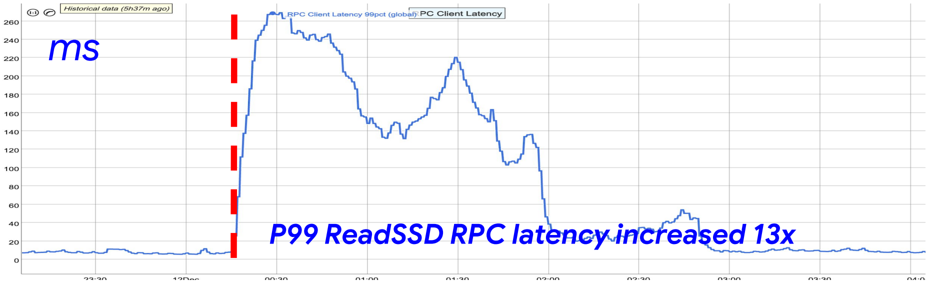
- Colossus high latency recovery mechanism:
  - Normal Read: Client directly reads relevant data chunks.
  - Degraded Read: Client reconstructs missing data chunks via reading erasure coding placed on other Storage (D) servers.

- Problem: Degraded reads initiated during client-side ToR hotspots amplify network congestion and latency.

- This underscores the necessity for a systematic solution to address hotspots.







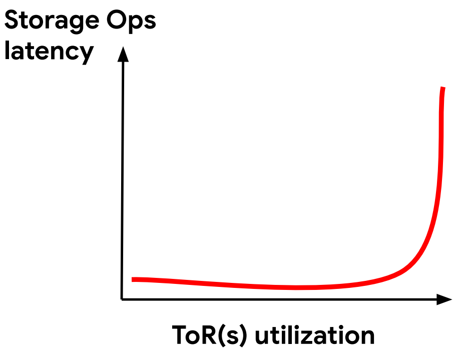
P99 ReadSSD RPC latency increased 13x

# Impact to Storage Operations

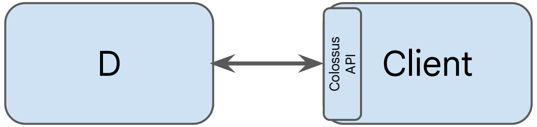
How much hotspots impact storage operations performance?

# Hotspot Latency Impact on Storage Operations

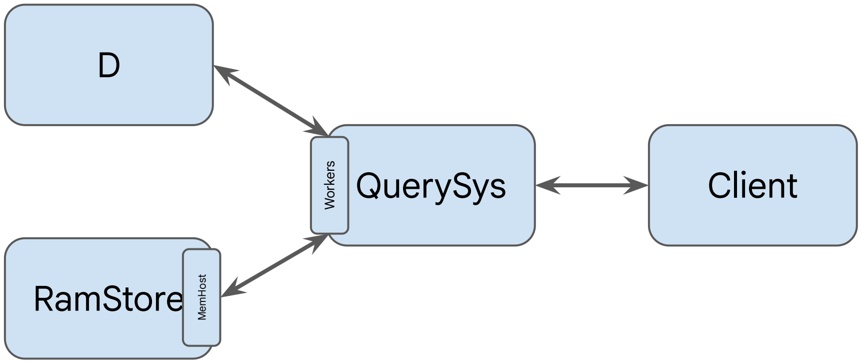
- Correlate the latency of storage operations with the corresponding ToR(s) utilization, generating a latency vs. utilization curve.

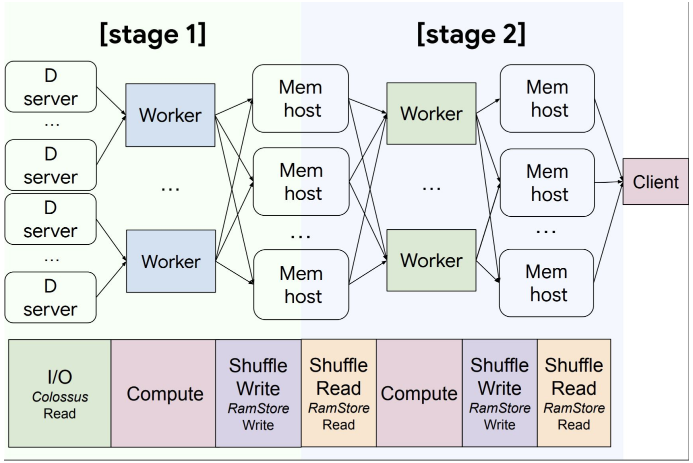


### File access read/write (Colossus)



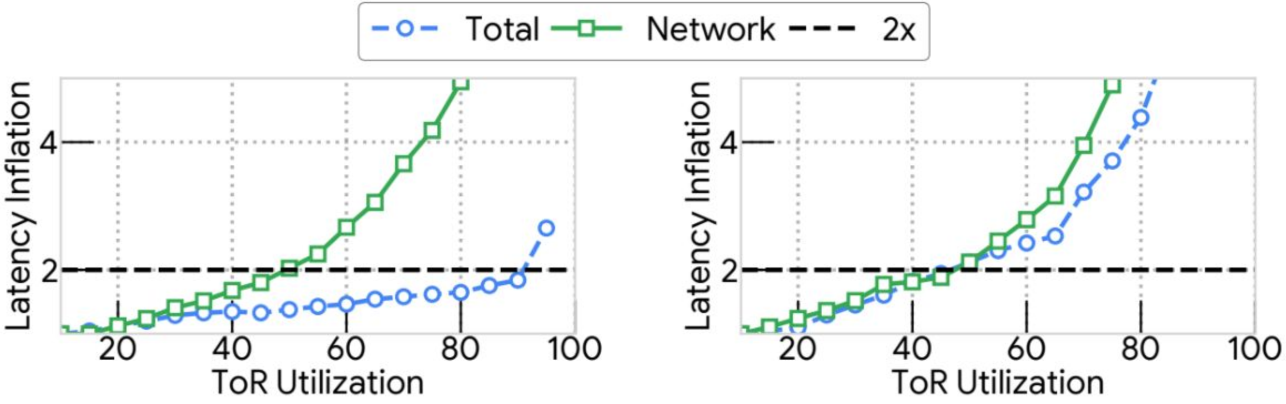
### Database queries (QuerySys)





# Colossus Read / Write Results

the ratio of the latency in a given util bucket to the latency of the lowest

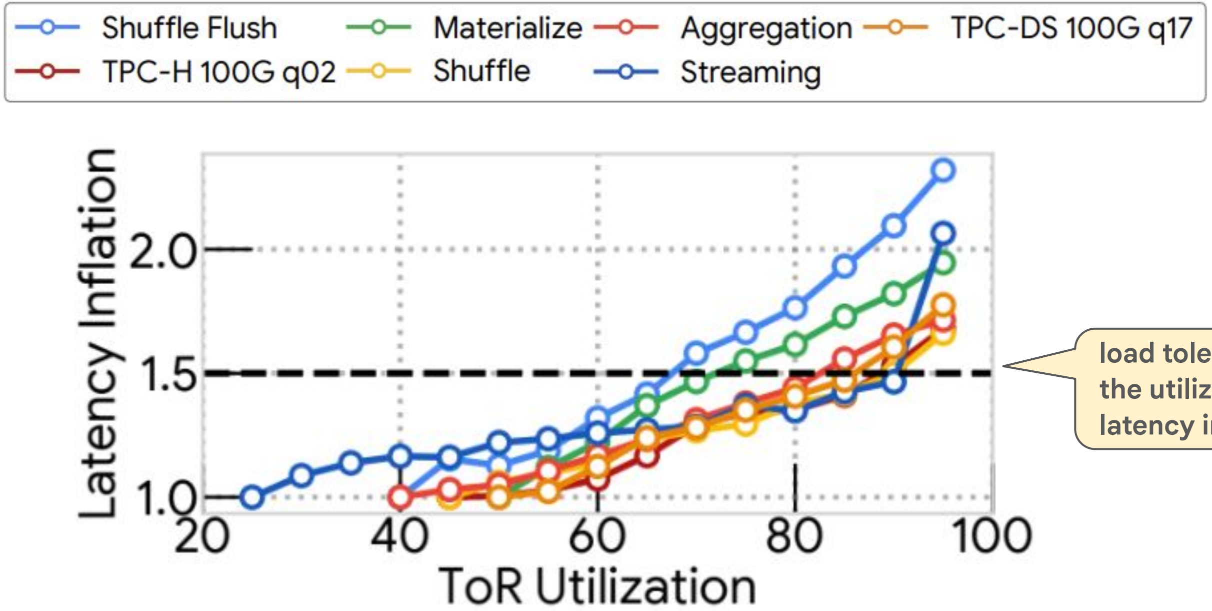


(a) HDD read.

(b) HDD write.

- **Read:** *Not too sensitive to high ToR utilization.*
  - Network latency accounts for <20% of total latency below 90% utilization, and ≤40% at 90%.
- **Write:** *Is network-dominated and significantly impacted by high ToR utilization.*
  - Writes are cached in battery-protected server-side memory and later flushed to disk.

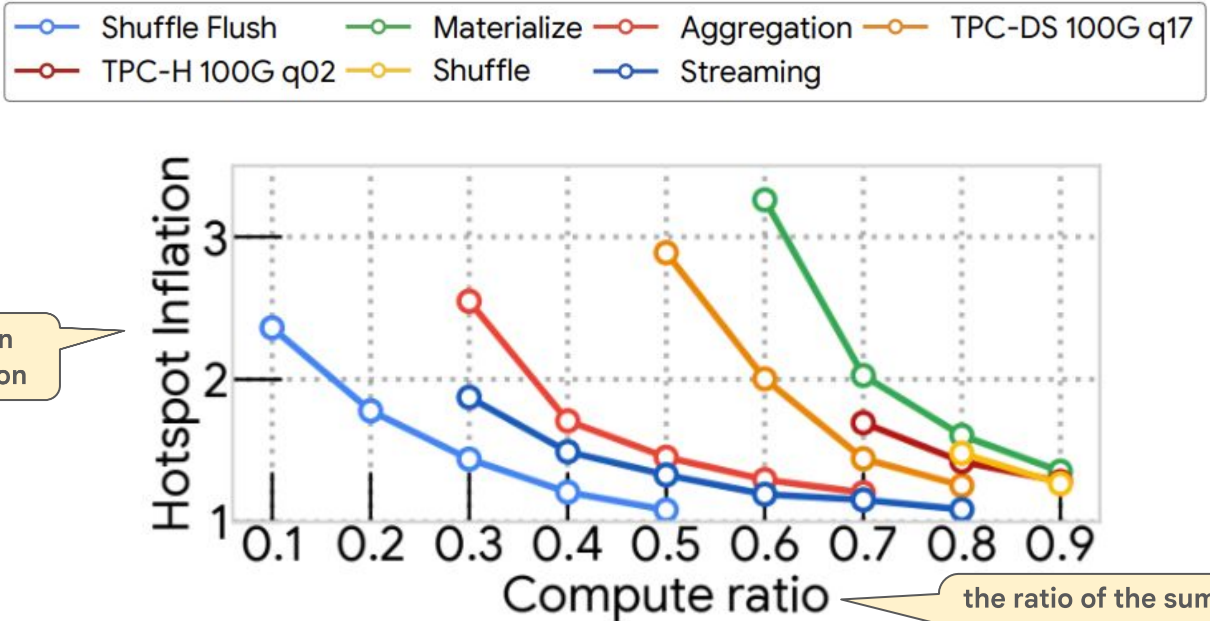
# QuerySys Benchmarks Results



load tolerance defined as the utilization at which the latency inflation is 1.5

Benchmarks show two sensitivity levels: two queries are sensitive to ToR utilization (load tolerance ~75%), while less network-intensive queries load tolerate is ~90–95%.

# QuerySys Benchmarks -- Understanding the Results



The latency inflation at 75% ToR utilization

the ratio of the sum of the processing time for the 10 slowest workers within each stage by the sum of their worker time

The more compute bound a query is the less sensitive it’s to hotspots.

# Hotspot-aware Placement

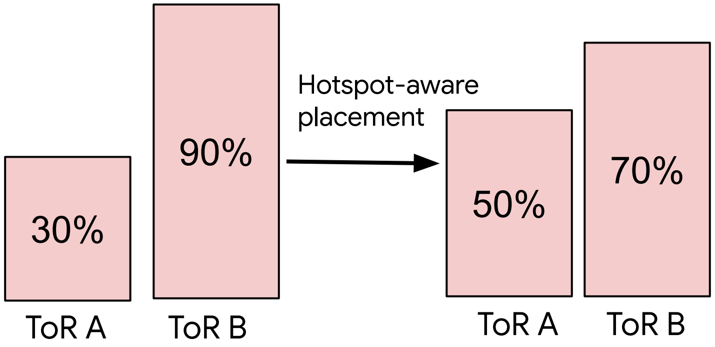
Scheduling strategies for addressing hotspots

# UTP: ToR Utilization Aware Placement and Migration

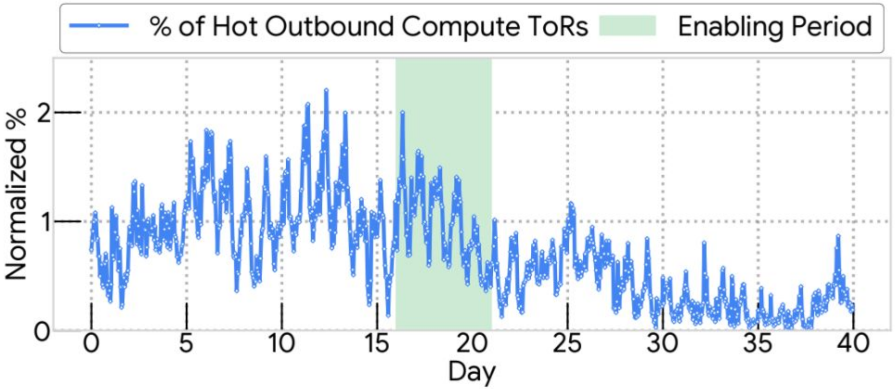
- Hotspots, arising from rack bandwidth demand-supply imbalances, can be addressed by scheduling systems through improved load balancing.

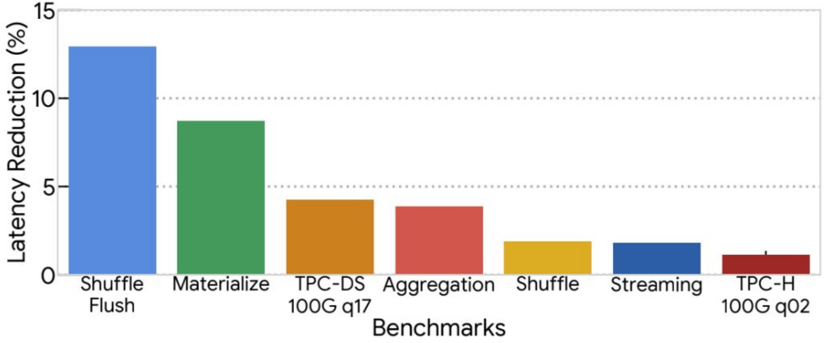
- Introduced network balancing as a scheduling objective for Borg (Google's job scheduler) & Colossus (Google's distributed file system)
  - Constrained by the goal of minimal adverse impacts on existing scheduling objectives

- Developed two scheduling capabilities for machines whose ToR are hot
  - **Proactive:** avoids putting new network-intensive tasks on them
  - **Reactive:** migrates existing network-intensive tasks away from them



# UTP Results & Insights





- Very effective
  - addressed **~90%** of hotspots.
  - Achieved up to **13%** p95 latency reduction across QuerySys benchmarks.
  - Reduced production incident rate by **~70%**.

- The scheduler has ample flexibility to mitigate hotspots without negatively impacting other scheduling objectives
  - Low average utilization leaves many ToRs far below this point providing ample headroom for balancing.
  - Most applications tolerate high network utilization allowing the scheduler to target only the tail of utilization.

# Conclusion & Future Work

# Conclusion and Future work

- ToR hotspots arise from *rack bandwidth demand-supply imbalances* in dynamic networks with evolving workloads, incremental resource upgrades, and ToR uplink outages.

- Hotspots can severely degrade application performance
  - Can double the latency of simple storage read/write and complex application operations
  - Operations *less compute-bound/disk-bound are more sensitive to hotspots*

- Best-effort hotspot-aware placement in compute and storage schedulers, to balance network load, *can significantly reduce hotspot frequency*.

- Fruitful future work directions
  - SLO-backed network-aware scheduling
  - Application-level hotspot-aware placement
  - Network fault-aware scheduling for ML workloads
  - Connecting placement and provisioning
  - Automated hotspot sensitivity measurement for diverse applications