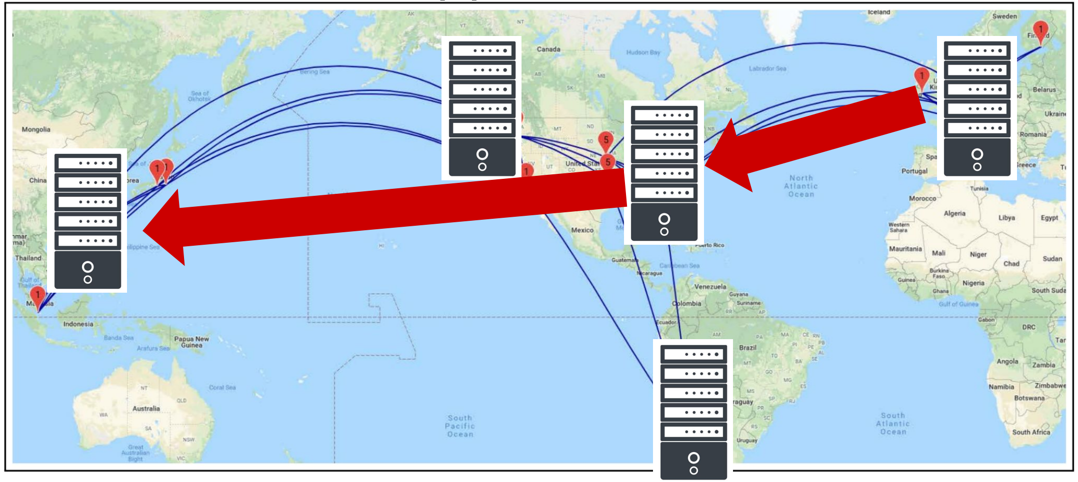
Google

A Cloud-Scale Characterization of Remote Procedure Calls

Korakit Seemakhupt, Brent Stephens, Samira Khan, Sihang Liu, Hassan Wessel, Soheil Hassas Yeganeh, Alex C. Snoeren, Arvind Krishnamurthy, David Culler, Henry M. Levy

Motivation

A cloud-scale application must be:



- Scalable
- Highly-available
- Failure tolerant
- Easy to maintain

Understanding RPC is a key to understanding global-scale distributed services

Google

RPC Study: Cloud-scale Workloads

This is a study of RPC at Google Scale

We include

- Google's **first party** web services
  - Search, Gmail, Youtube, ...
- Google's **internal services**
  - Spanner, Bigtable, F1, ...

We do **not** include

- RPCs serving **Cloud customers** (GCP)
- **RDMA** and software-based **RMA** communication (Snap/Pony Express)

RPC Study: Measurements

We collected and processed data using three Google internal monitoring systems

Overall we examined:

- Over **700 billion** RPC traces
- **10,000** different RPC methods from over **100** production clusters
- System statistics collected every **30 minutes for ~2 years**

Aggregated statistics include:

- Latency Components, Payload Size, Call Structure
- CPU Utilization, Memory Bandwidth, Scheduling Latency
- Requests/Second, Growth rate, ...

Agenda

- Motivations for studying RPC in the cloud
- RPC Study Results
  - What is the **source of RPCs**? Where do they go?
  - What is the **timescale** of RPC?
  - Which **latency component** affects RPC latency?
  - Which **RPC Latency Tax** component is the bottleneck?
  - How does **utilization** vary across datacenters?
- Implications from the study

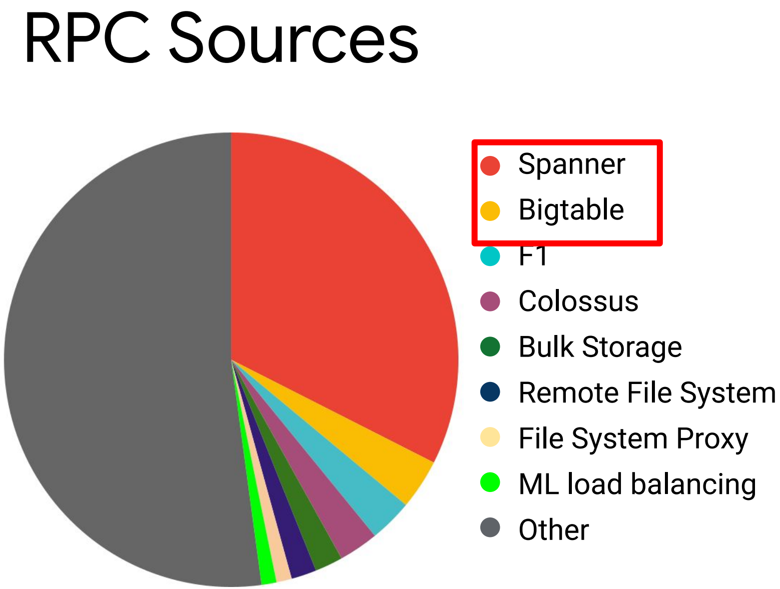
Google

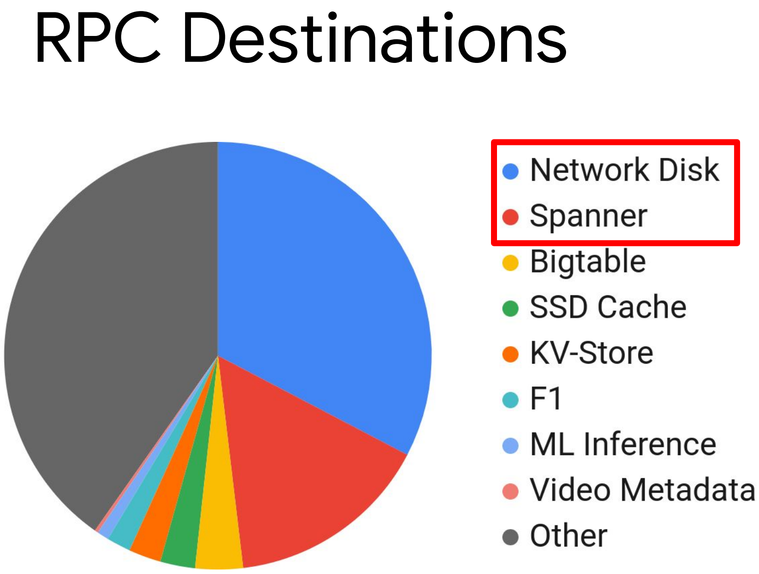
Agenda

- Motivations for studying RPC in the cloud
- RPC Study Results
  - What is the **source of RPCs**? Where do they go?
  - What is the **timescale** of RPC?
  - Which **latency component** affects RPC latency?
  - RPC Latency Tax
  - How does **utilization** vary across datacenters?
- Implications from the study

Google's RPC Environment

RPC Sources and Destinations

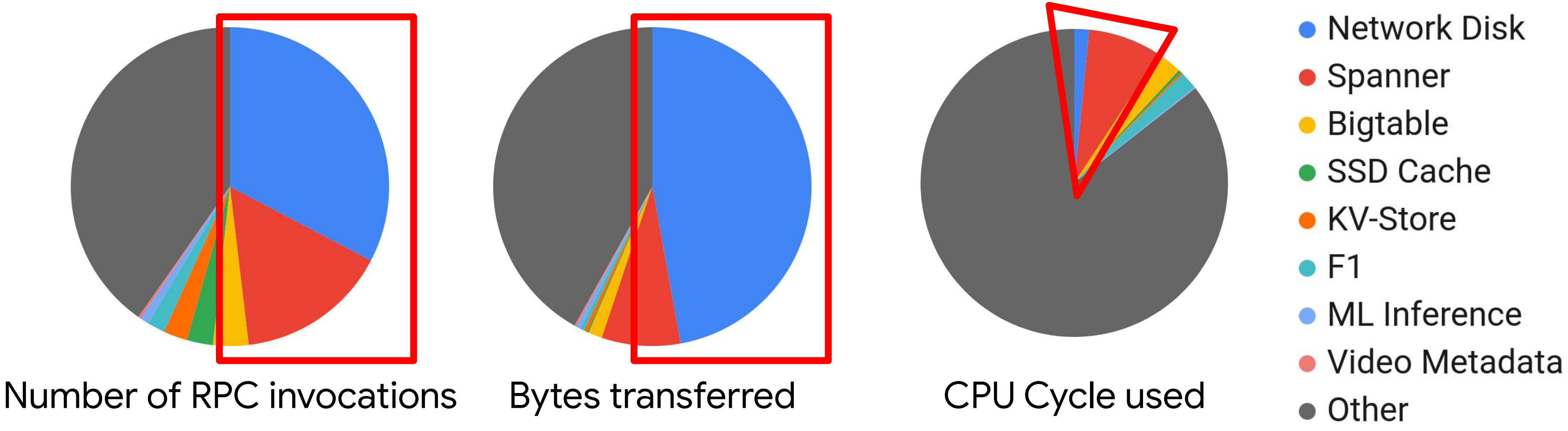




Google's Internal RPC is dominated by communication between **storage services**.

Google's RPC Environment

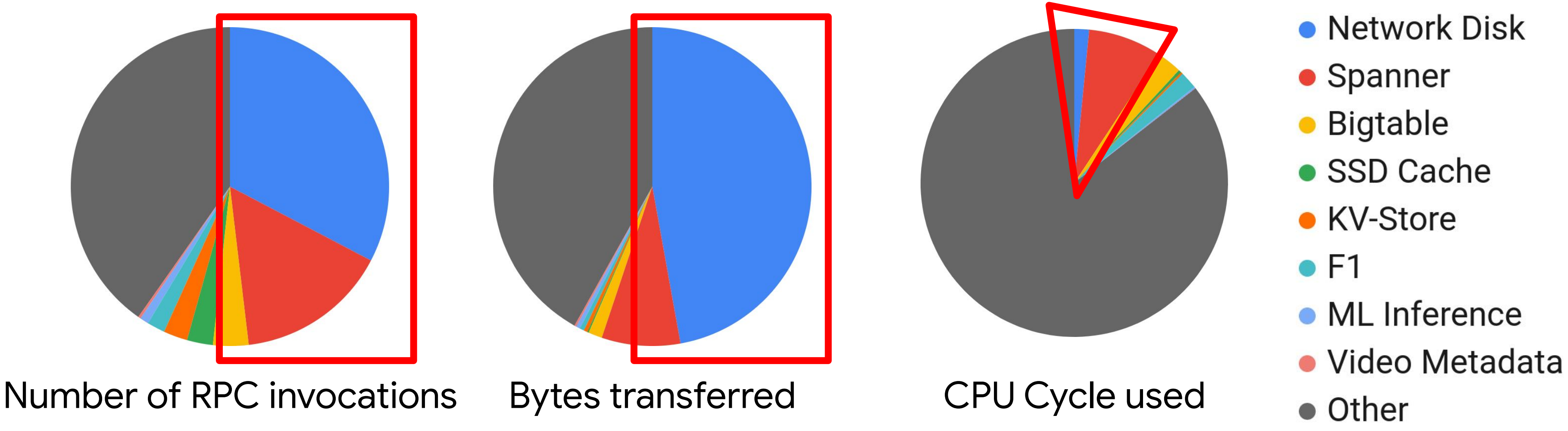
RPC Popularity and Resource Utilization (by destination)



Half of RPC invocations and data transferred are from **Spanner** and **Network Disk**

Google's RPC Environment

RPC Popularity and Resource Utilization (by destination)

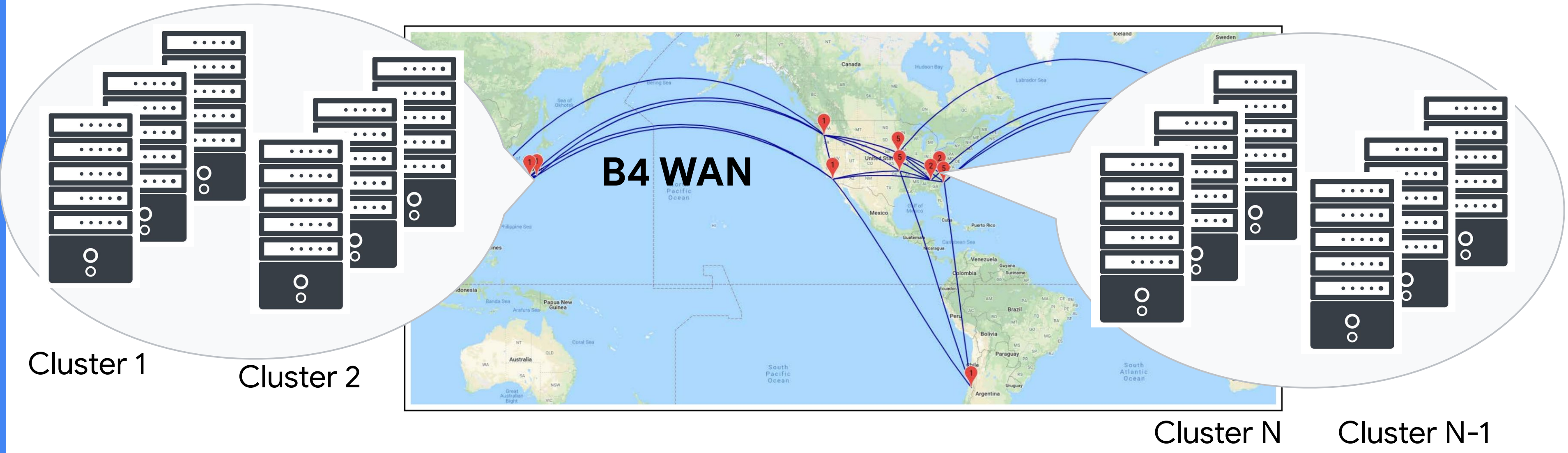


Takeaway: Storage RPC is by far the largest contributor to fleet-wide RPC and bytes transfer in the network.
This motivates for research on **data-movement acceleration**.

Google

Google's RPC Environment

Cross-cluster RPC and WAN



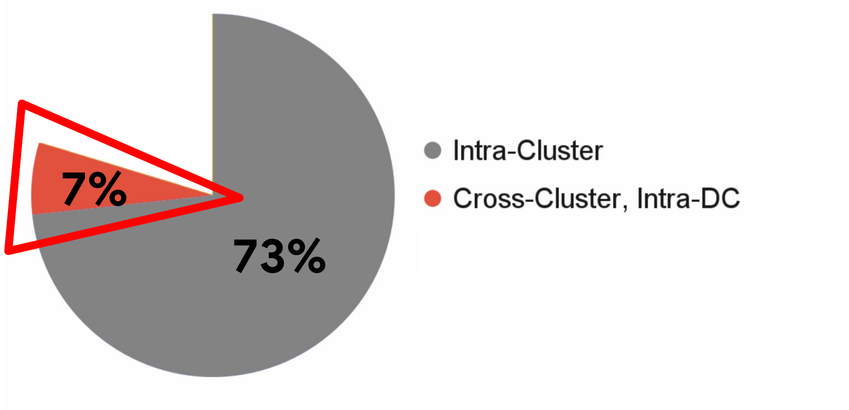
Google's geo-distributed datacenters

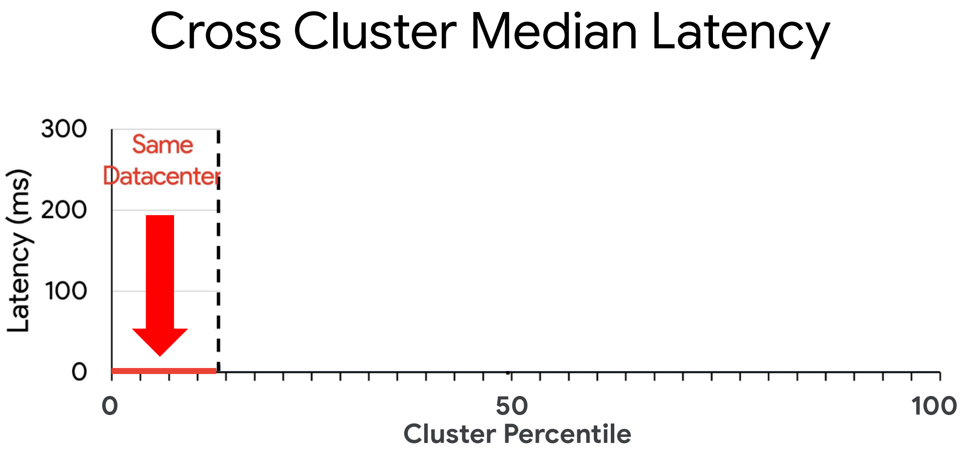
Each datacenter can consist of **multiple clusters**

Datacenters are connected through **WAN links (B4)**

Google's RPC Environment

Cross-cluster RPC and WAN





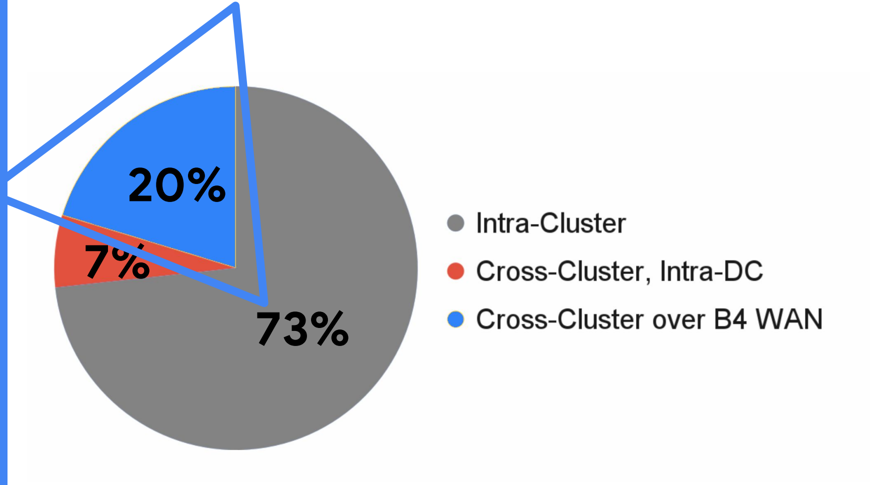
7% of RPCs are **cross-cluster**, but in **the same datacenter**

Same Datacenter RTT is under 10 ms

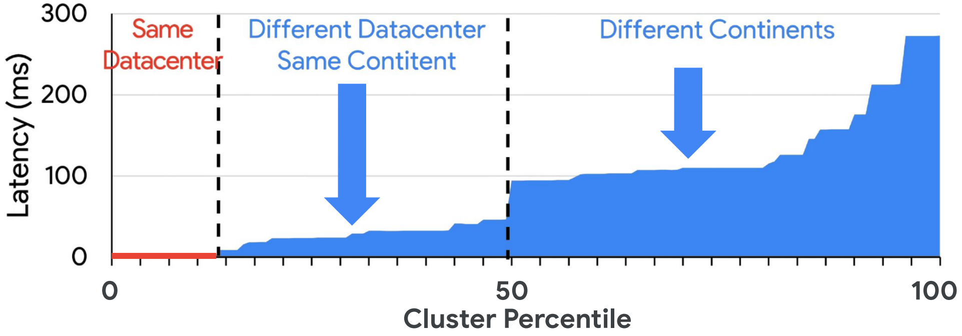
Google

Google’s RPC Environment

Cross-cluster RPC and WAN



Cross Cluster Median Latency



20% of RPCs are **cross-cluster** over **B4 WAN**

Cross-continent RTT can be over 200 ms

Takeaway: RPC locality significantly affects the latency. Cross-cluster RPCs over WAN introduces significant overhead

RPC Study Results:
Takeaways

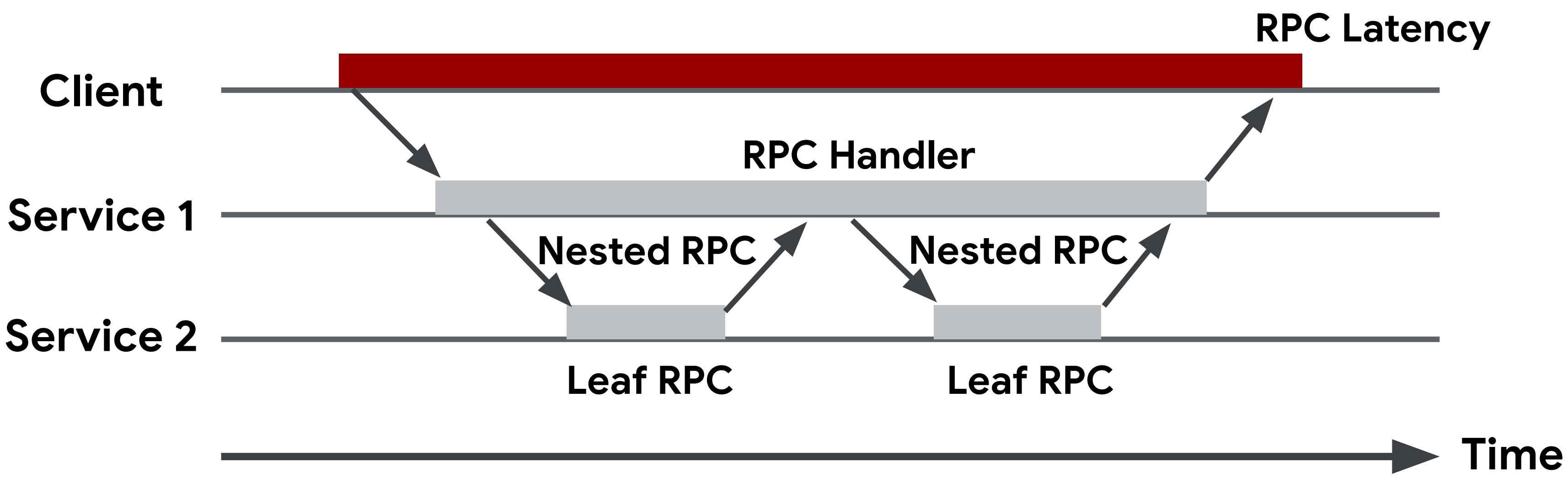
What is the **source of RPCs**? Where do they go?

- Storage RPCs are the largest contributor to fleet-wide RPCs
  - Motivates research on **data-movement acceleration**
- RPC locality significantly affects latency
  - Motivates research on **locality-aware scheduling**

Agenda

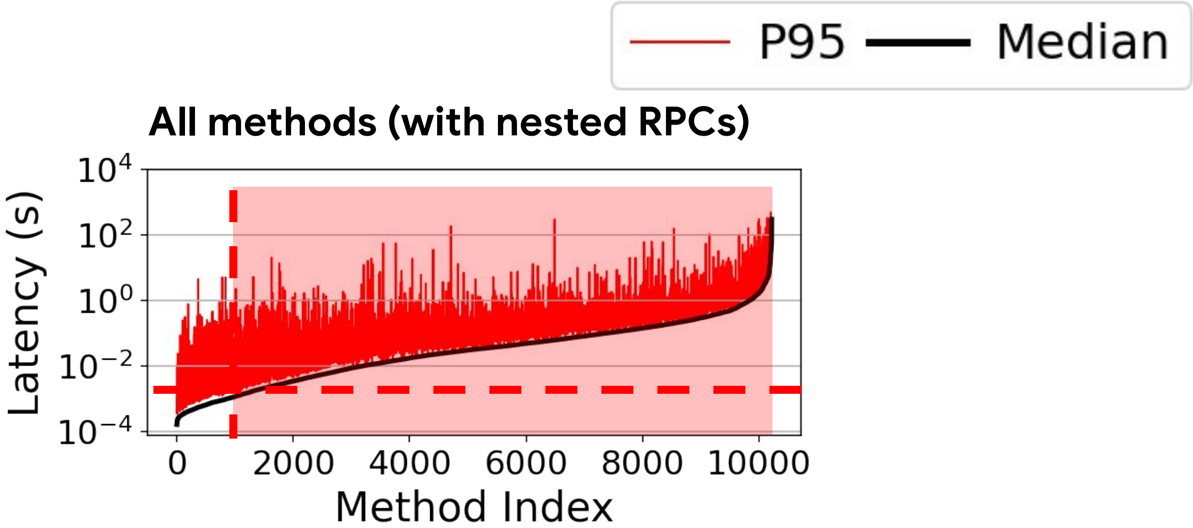
- Motivations for studying RPC in the cloud
- RPC Study Results
  - What is the **source of RPCs**? Where do they go?
  - What is the **timescale** of RPC?
  - Which **latency component** affects RPC latency?
  - Which **RPC Latency Tax** component is the bottleneck?
  - How does **utilization** vary across datacenters?
- Implications from the study

RPC completion time includes nested RPC calls



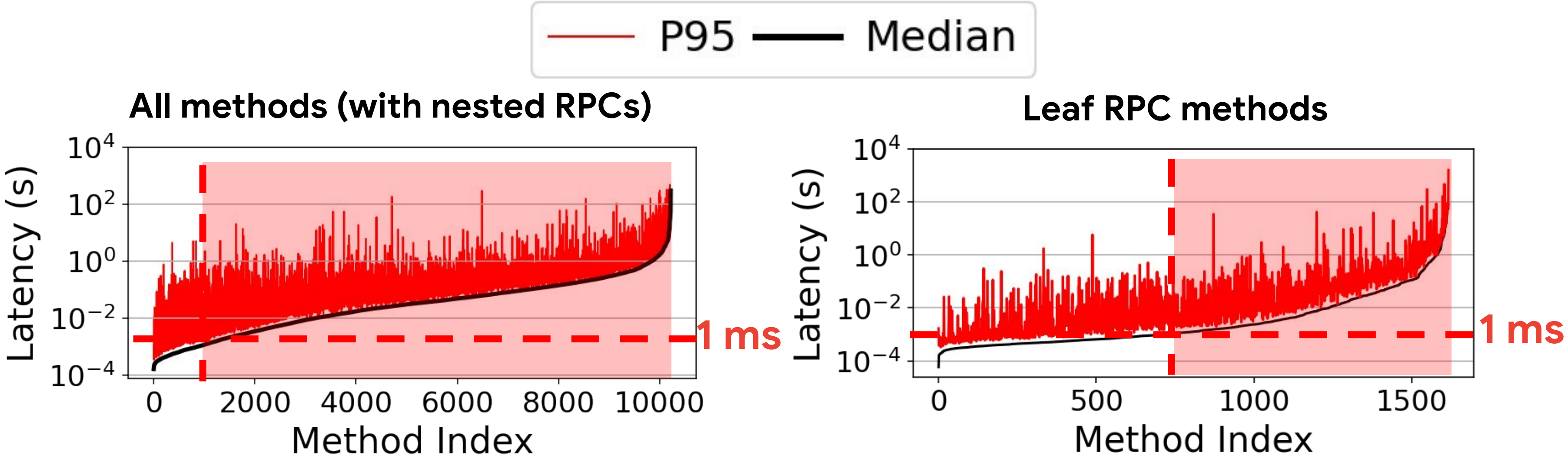
RPC Latency includes **RPC handler** and **nested RPC calls**.
We also show **leaf RPC latency**

What is the timescale of RPC?



90% of RPC methods have median latency **over a millisecond**.

What is the timescale of RPC?



90% of RPC methods have median latency **over a millisecond.**

53% of leaf RPC methods have median latency **over a millisecond**.

Takeaway: Majority of RPC methods in this environment are **millisecond**, not microsecond scale

RPC Study Results:
Takeaways

What is the **timescale** of RPC?

- Majority of RPC methods in this environment are **millisecond** scale

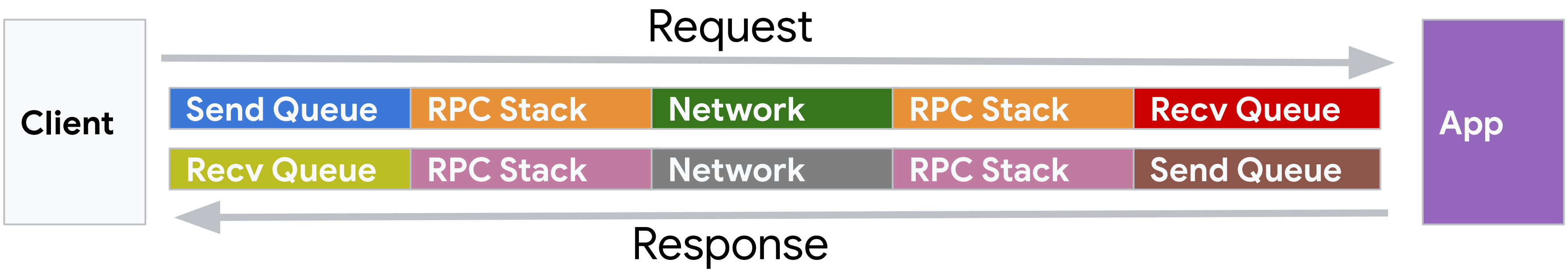
- But half of the **leaf RPC** methods have **sub-millisecond** latency
    - Optimizing for **latency** is still important for median **leaf RPCs**

Google

Agenda

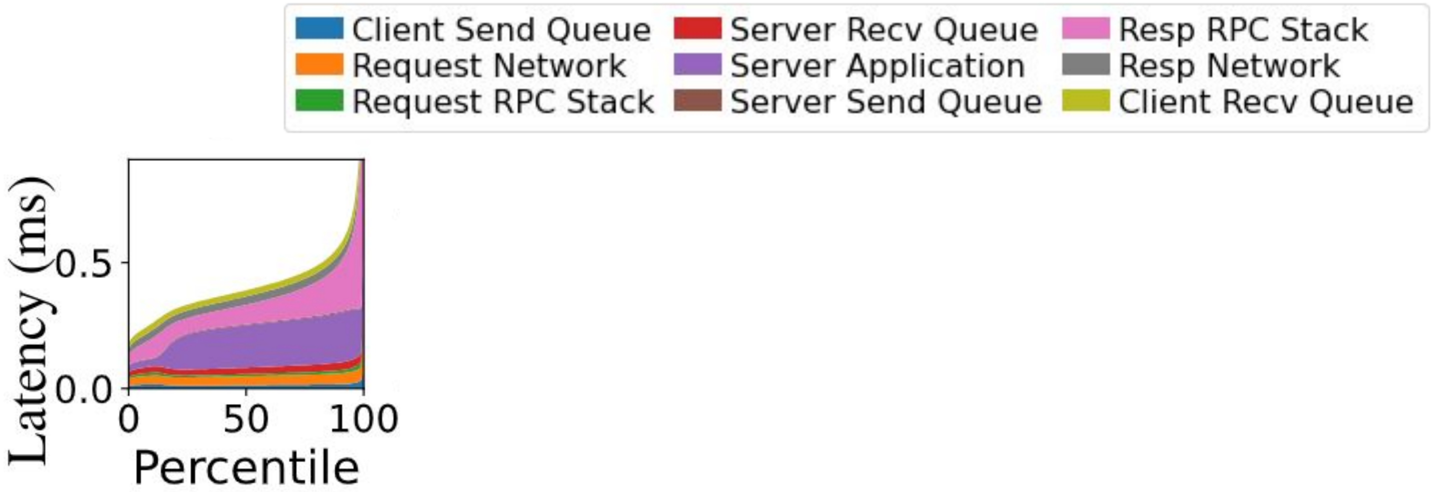
- Motivations for studying RPC in the cloud
- RPC Study Results
    - What is the **source of RPCs**? Where do they go?
    - What is the **timescale** of RPC?
    - Which **latency component** affects RPC latency?
    - Which **RPC Latency Tax** component is the bottleneck?
    - How does **utilization** vary across datacenters?
- Implications from the study

What are the **Latency components**?



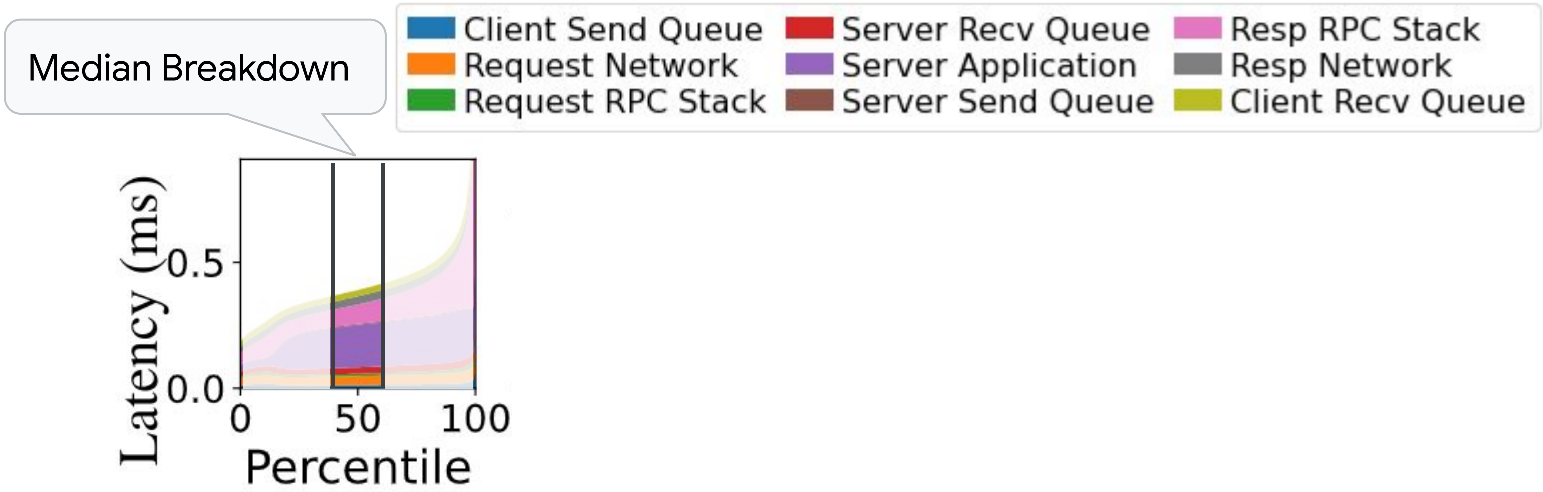
We measure time spent on **queues, RPC stack, network**, and the **application processing time** in leaf RPCs.

What are the different causes of latency?



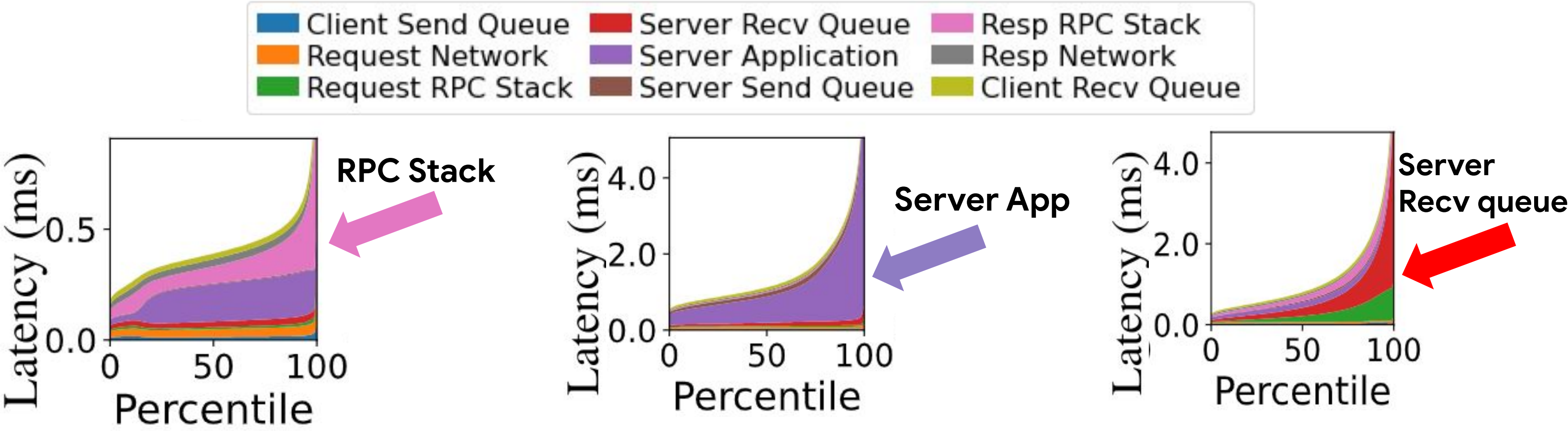
RPC Stack dominated:
(e.g., K/V-Store)

What are the different causes of latency?



RPC Stack dominated:
(e.g., K/V-Store)

What are the different causes of latency?



RPC Stack dominated:
(e.g., K/V-Store)

RPC method dominated:
(e.g., ML Inference)

Queueing dominated:
(e.g., SSD Cache)

Different dominant component for each application

Google

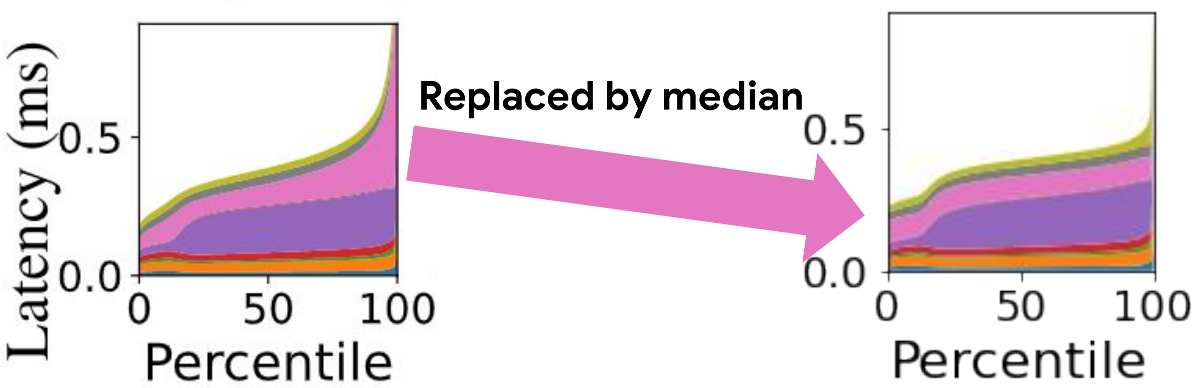
What-if analysis with causal modeling

Research question:

What is the latency component that is most responsible for tail latency?

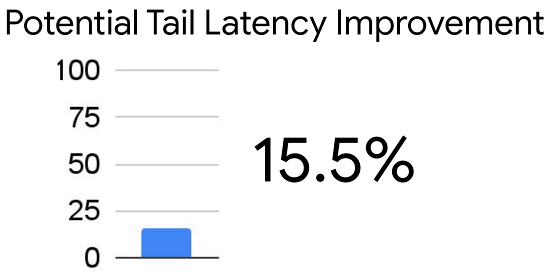
How much latency can we improve by optimizing that component?

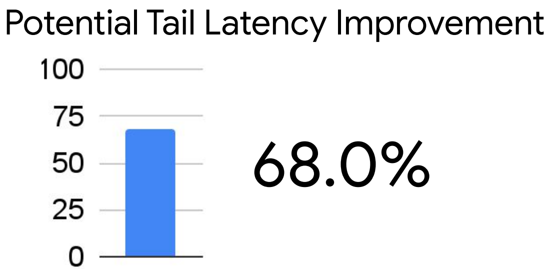
Methodology:

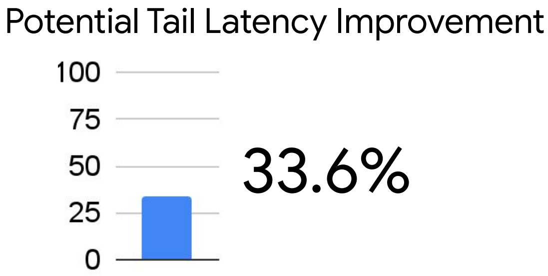


Goal: Understand impact of reducing the variation caused by that component

Potential latency improvement







RPC bottlenecked: K/V-Store

RPC optimization can improve tail latency by 15.5%

App bottlenecked: ML Inference

Accelerated application processing can improve tail latency by 68.0%

Queueing bottlenecked: SSD Cache

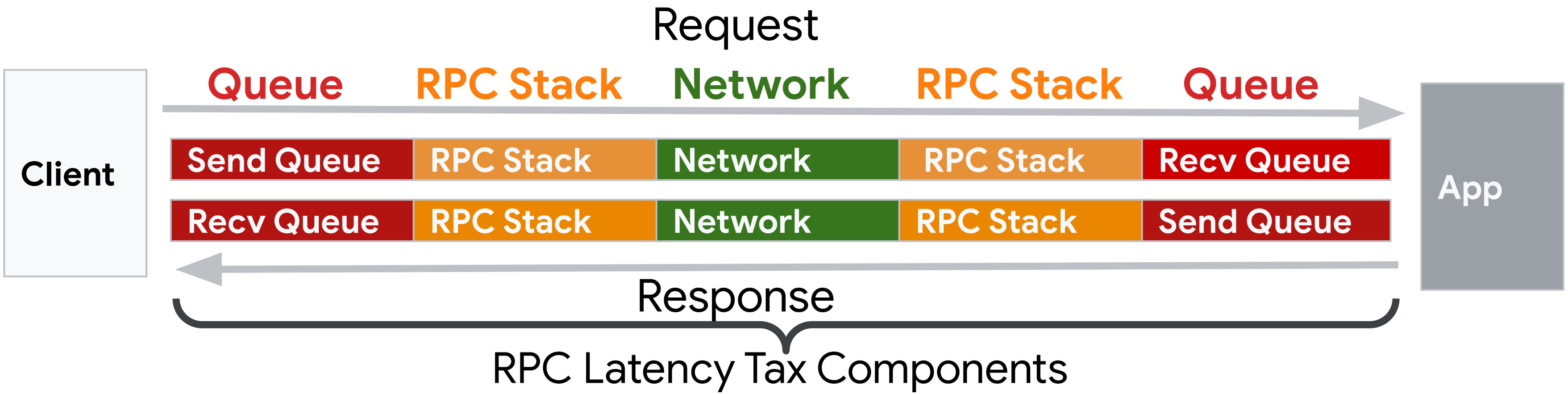
Scheduling or resource management can improve tail latency by 33.6%

Takeaway: There is a potential for a significant reduction to tail latency by **eliminating the variation** caused by the **dominant component**.

Agenda

- Motivations for studying RPC in the cloud
- RPC Study Results
  - What is the **source of RPCs**? Where do they go?
  - What is the **timescale** of RPC?
  - Which **latency component** affects RPC latency?
  - Which **RPC Latency Tax** component is the bottleneck?
  - How does **utilization** vary across datacenters?
- Implications from the study

RPC Latency Tax

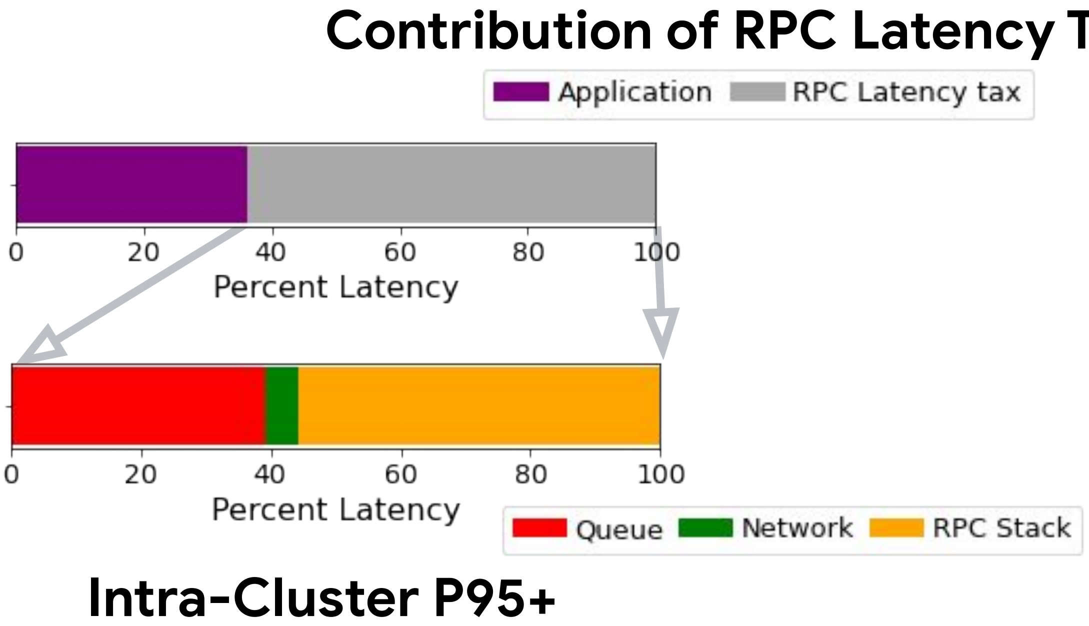


We define RPC Latency Tax as the overhead of running application over RPC, all latency components excluding the application processing time

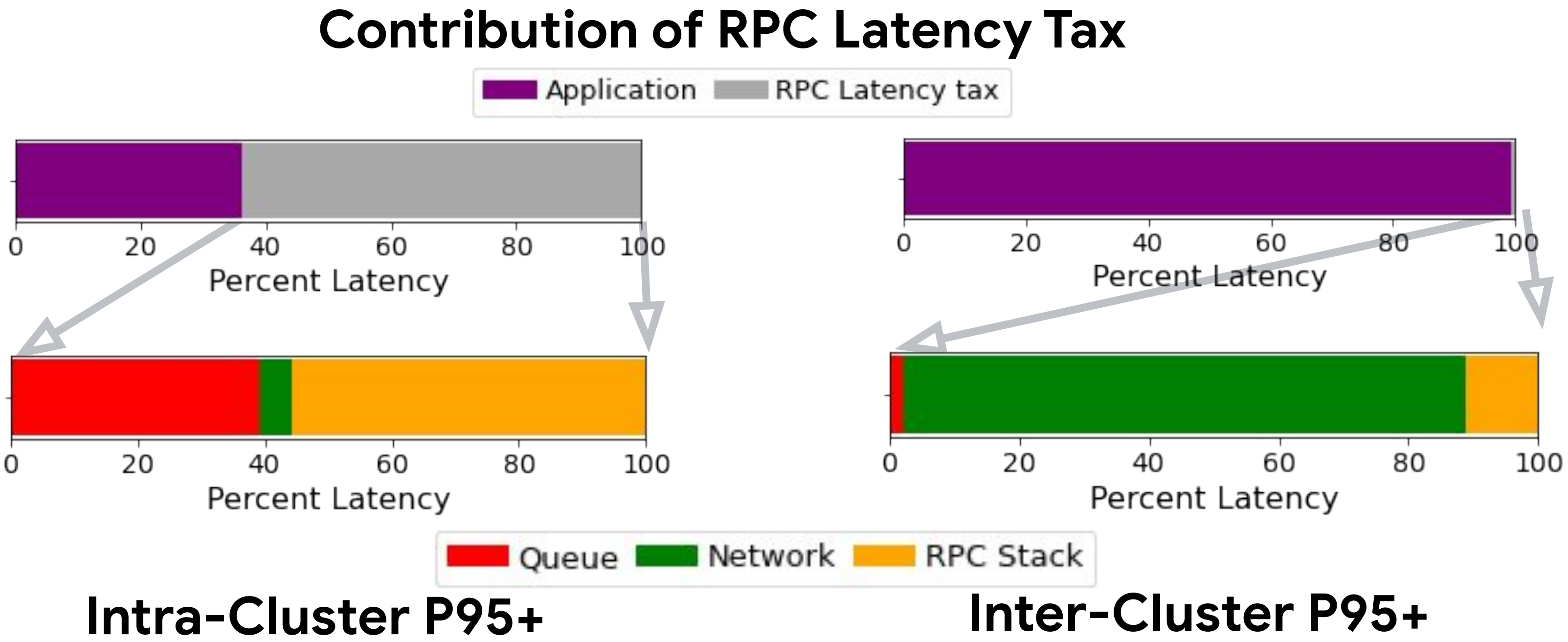
How significant is RPC Latency Tax for **within** and **across clusters**?

Google

RPC Latency Tax: Variation at Tail

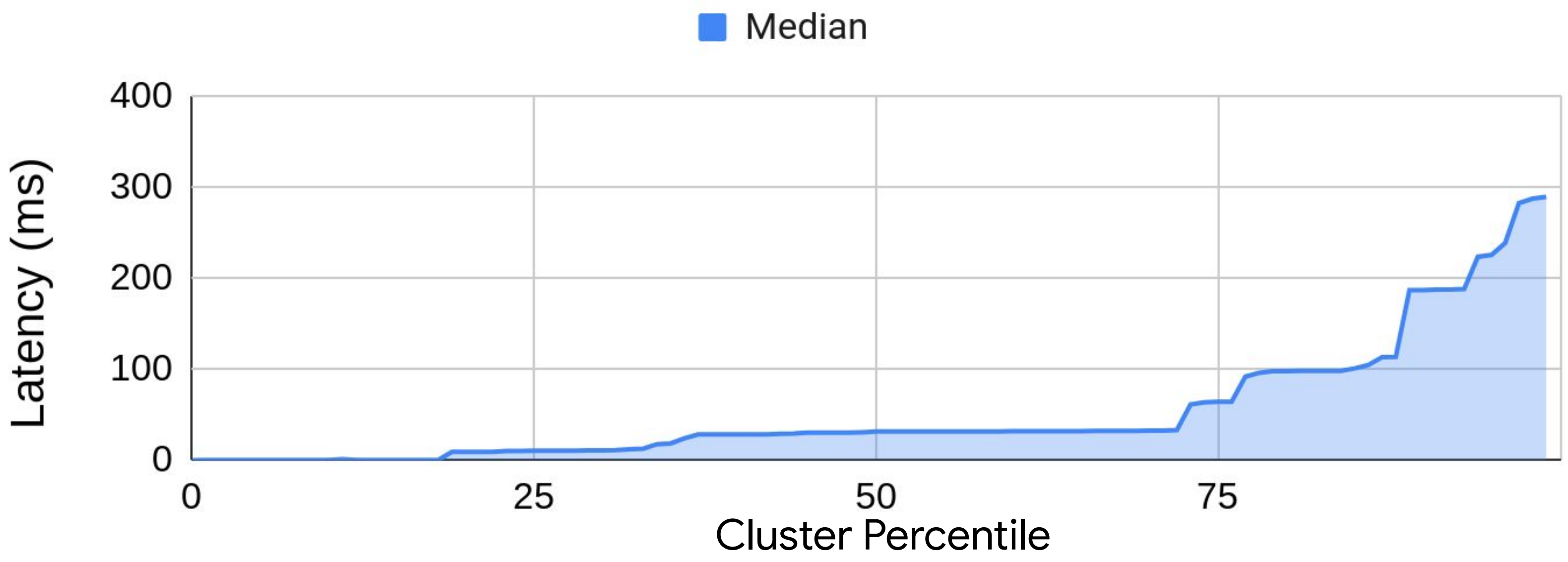


RPC Latency Tax: Variation at Tail



Takeaway: RPC Latency tax is **significant at tail**

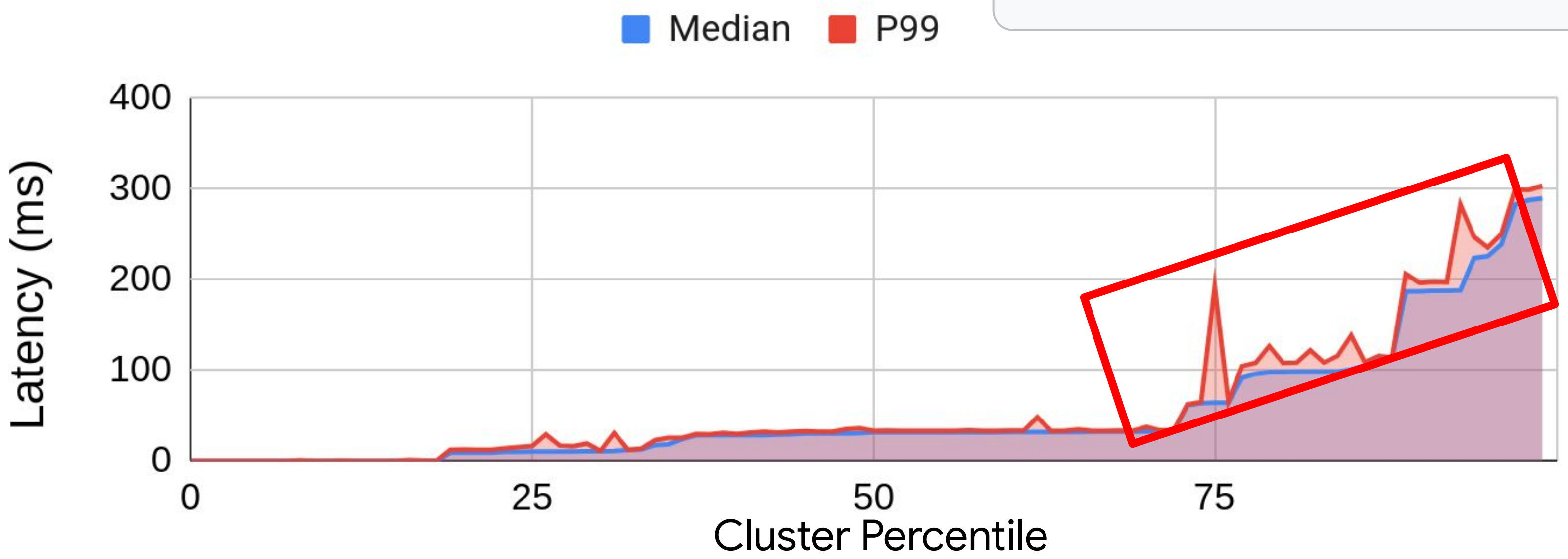
RPC Latency Tax: Inter-Cluster Variation at Tail



Google

RPC Latency Tax: Inter-Cluster Variation at Tail

Latency spike due to **congestion**



Takeaway: **Congestion on WAN** can have an impact on **tail latency across clusters**

RPC Study Results:
Takeaways

Which **latency component** affects RPC latency?

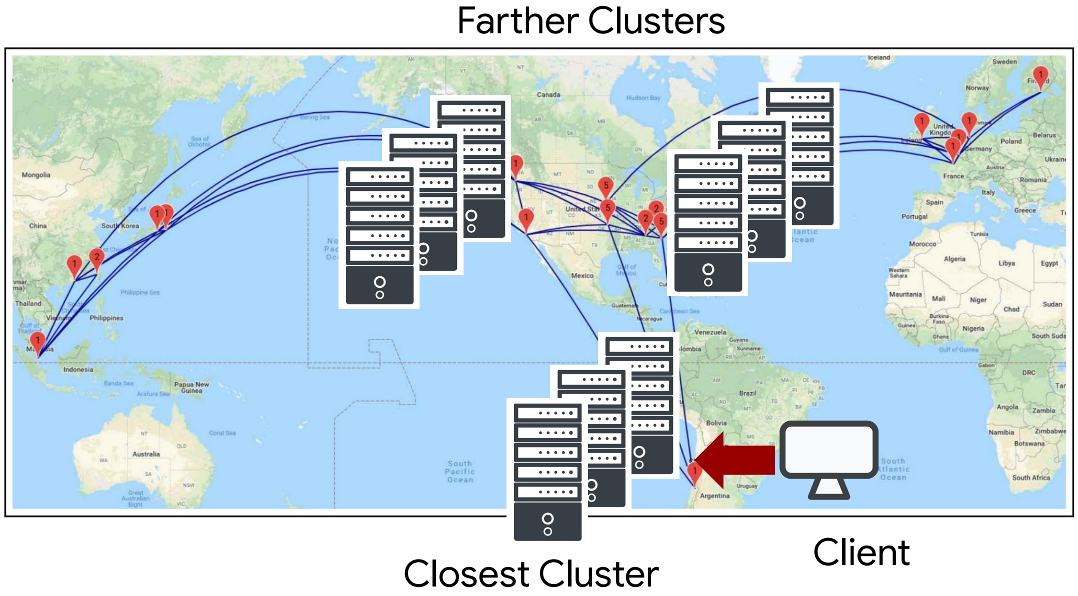
- **Dominant latency component** is different for each service
  - Optimize RPC stack, queueing or app processing
- **RPC Latency tax is significant at tail**
  - **Queueing matters** for intra-cluster RPCs.
  - **WAN congestion** matters for inter-cluster RPCs.

Agenda

- Motivations for studying RPC in the cloud
- RPC Study Results
    - What is the **source of RPCs**? Where do they go?
    - What is the **timescale** of RPC?
    - Which **latency component** affects RPC latency?
    - Which **RPC Latency Tax** component is the bottleneck?
    - How does **utilization** vary across datacenters?
- Implications from the study

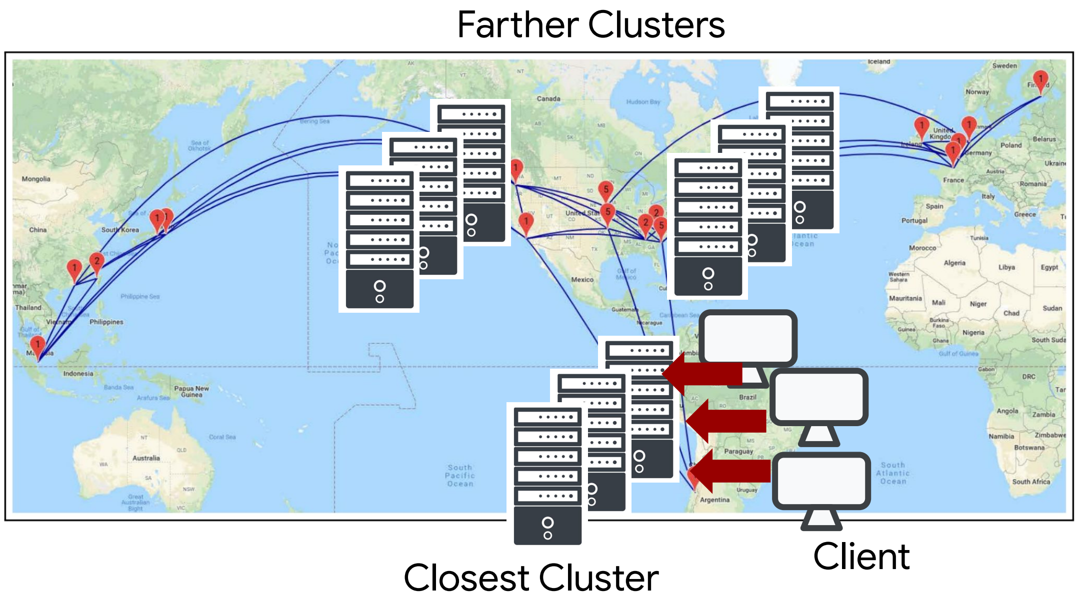
Load balancing

To avoid inter-cluster traffic, we could serve in the cluster **closest to the client**



Load balancing

To avoid inter-cluster traffic, we could serve in the cluster **closest to the client**



However, serving requests on the closest cluster could unbalance load.

Google

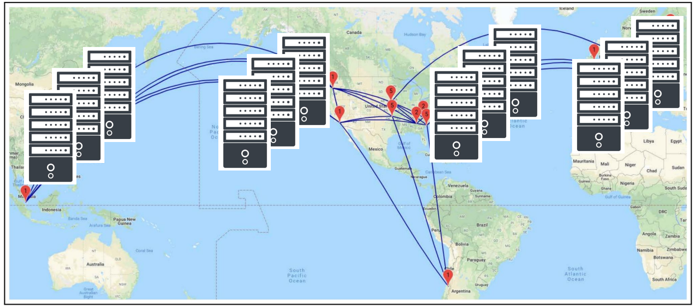
Load balancing: CPU Utilization Variation

Research question:

Is CPU utilization balanced across **different clusters**?

Is CPU utilization balanced across **different machines** within a cluster?

Methodology: Collect CPU Utilization



Google

Load balancing: CPU Utilization Variation

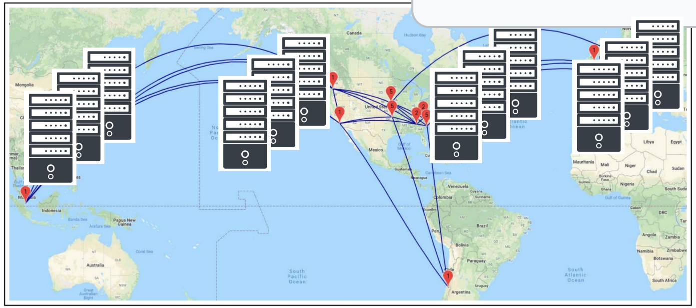
Research question:

Is CPU utilization balanced across **different clusters**?

Is CPU utilization balanced across **different machines** within a cluster?

Methodology: Collect CPU Utilization

1. Across different clusters



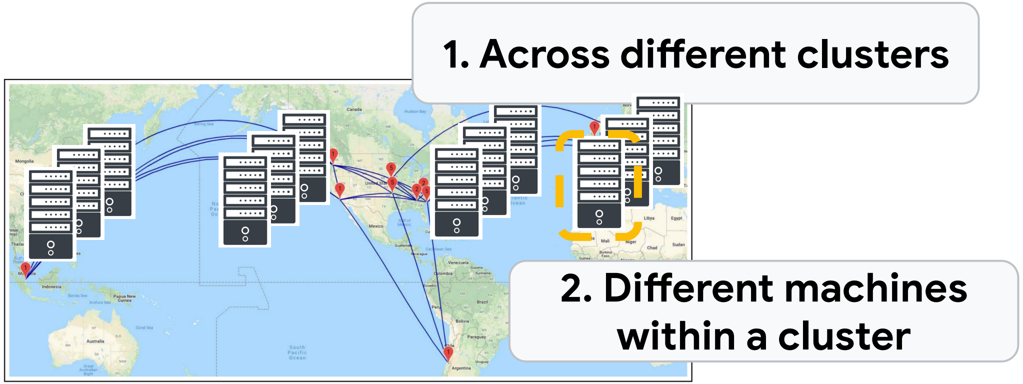
Load balancing: CPU Utilization Variation

Research question:

Is CPU utilization balanced across **different clusters**?

Is CPU utilization balanced across **different machines** within a cluster?

Methodology: Collect CPU Utilization



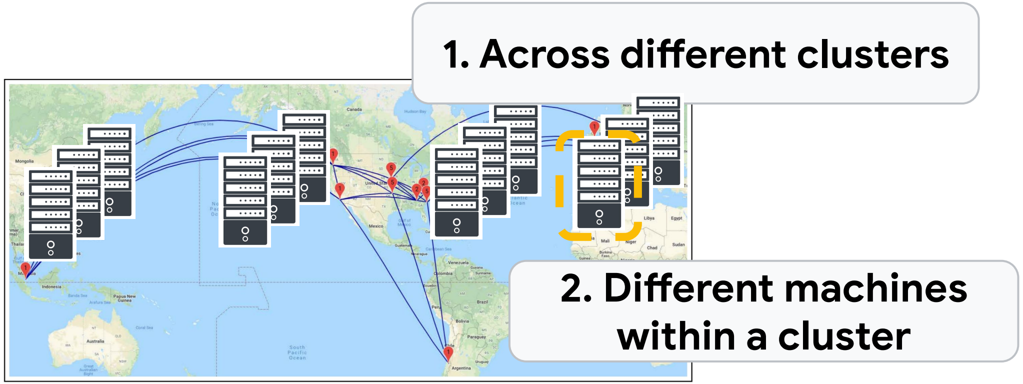
Load balancing: CPU Utilization Variation

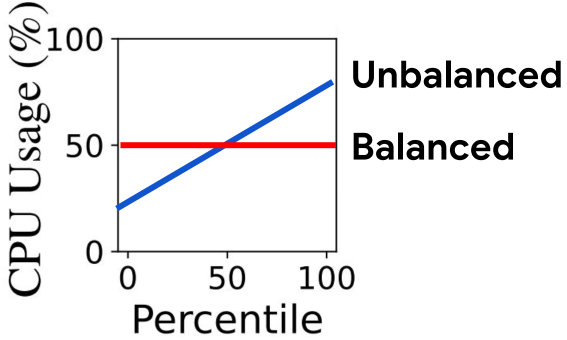
Research question:

Is CPU utilization balanced across **different clusters**?

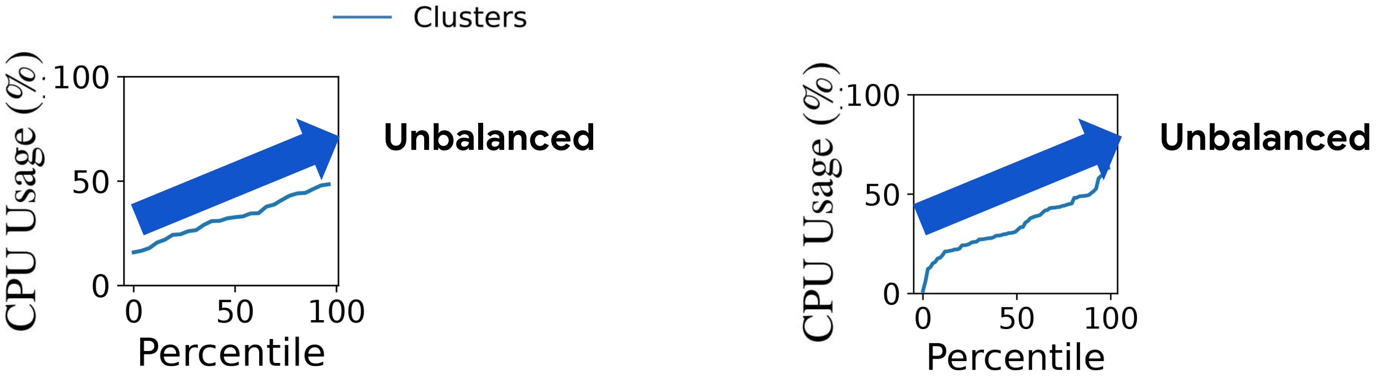
Is CPU utilization balanced across **different machines** within a cluster?

Methodology: Collect CPU Utilization





Load balancing: Cross-Cluster CPU Utilization Variation



SSD Cache **ML Inference**

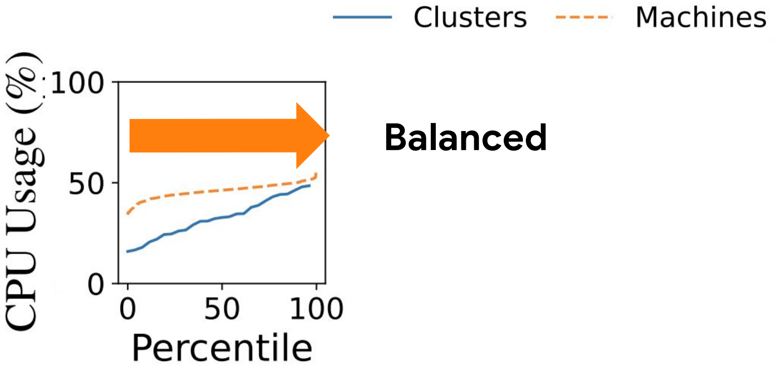
Unbalanced across clusters

Unbalanced across clusters

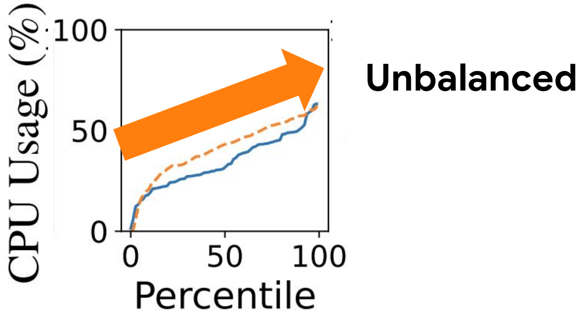
Takeaway: There are trade-offs between **network latency** and **load balancing across clusters**.

Google

Load balancing: Same-Cluster CPU Utilization Variation



SSD Cache



ML Inference

Unbalanced across clusters
Balanced across within a cluster

Unbalanced across clusters
Unbalanced within a cluster

Takeaway: Compute services with **unpredictable** latency are unbalance within a cluster.

RPC Study Results:
Takeaways

How do latency and utilization vary across datacenters?

- Hard to balance load for services with **varied computation**
- Inter-cluster RPC placement helps load balancing across clusters but **can increase latency**

Agenda

- Motivations for studying RPC in the cloud
- RPC Study Results
  - What is the **source of RPCs**? Where do they go?
  - What is the **timescale** of RPC?
  - Which **latency component** affects RPC latency?
  - Which **RPC Latency Tax** component is the bottleneck?
  - How does **latency and utilization** vary across datacenters?
- Implications from the study

Implications

- **Storage data flow optimization is important**
  - Majority of RPC invocations and data transfer are from storage applications
  - Optimizing data movement for storage RPCs can significantly improve resource efficiency
- **Millisecond, not just microsecond timescales**
  - Most RPCs operate in millisecond scale
  - Reducing CPU utilization can be more beneficial than saving a few microseconds
- **Host queuing matters**
  - Client & Server queuing latency are major contributors to the tail latency
  - Improving scheduling and placement is important
- **RPC Latency Tax is significant at tail**
  - Need to optimize RPC overhead at the tail requests
- **Load-balancing needs to account for latency**
  - Need research on predicting latency for RPCs with varied computational needs
  - Scheduling across cluster needs to co-optimize for latency and load-balancing

Google

Implications

- **Storage data flow optimization is important**
  - Majority of RPC invocations and data transfer are from storage applications
  - Optimizing data movement for storage RPCs can significantly improve resource efficiency
- **Millisecond, not just microsecond timescales**
  - Most RPCs operate in millisecond scale
  - Reducing CPU utilization can be more beneficial than saving a few microseconds
- **Host queuing matters**
  - Client & Server queuing latency are major contributors to the tail latency
  - Improving scheduling and placement is important
- **RPC Latency Tax is significant at tail**
  - Need to optimize RPC overhead at the tail requests
- **Load-balancing needs to account for latency**
  - Need research on predicting latency for RPCs with varied computational needs
  - Scheduling across cluster needs to co-optimize for latency and load-balancing

Implications

- **Storage data flow optimization is important**
  - Majority of RPC invocations and data transfer are from storage applications
  - Optimizing data movement for storage RPCs can significantly improve resource efficiency
- **Millisecond, not just microsecond timescales**
  - Most RPCs operate in millisecond scale
  - Reducing CPU utilization can be more beneficial than saving a few microseconds
- **Host queuing matters**
  - Client & Server queuing latency are major contributors to the tail latency
  - Improving scheduling and placement is important
- **RPC Latency Tax is significant at tail**
  - Need to optimize RPC overhead at the tail requests
- **Load-balancing needs to account for latency**
  - Need research on predicting latency for RPCs with varied computational needs
  - Scheduling across cluster needs to co-optimize for latency and load-balancing

Implications

- **Storage data flow optimization is important**
  - Majority of RPC invocations and data transfer are from storage applications
  - Optimizing data movement for storage RPCs can significantly improve resource efficiency
- **Millisecond, not just microsecond timescales**
  - Most RPCs operate in millisecond scale
  - Reducing CPU utilization can be more beneficial than saving a few microseconds
- **Host queuing matters**
  - Client & Server queuing latency are major contributors to the tail latency
  - Improving scheduling and placement is important
- **RPC Latency Tax is significant at tail**
  - Need to optimize RPC overhead at the tail requests
- **Load-balancing needs to account for latency**
  - Need research on predicting latency for RPCs with varied computational needs
  - Scheduling across cluster needs to co-optimize for latency and load-balancing

Implications

- **Storage data flow optimization is important**
  - Majority of RPC invocations and data transfer are from storage applications
  - Optimizing data movement for storage RPCs can significantly improve resource efficiency
- **Millisecond, not just microsecond timescales**
  - Most RPCs operate in millisecond scale
  - Reducing CPU utilization can be more beneficial than saving a few microseconds
- **Host queuing matters**
  - Client & Server queuing latency are major contributors to the tail latency
  - Improving scheduling and placement is important
- **RPC Latency Tax is significant at tail**
  - Need to optimize RPC overhead at the tail requests
- **Load-balancing needs to account for latency**
  - Need research on predicting latency for RPCs with varied computational needs
  - Scheduling across cluster needs to co-optimize for latency and load-balancing

Google

Conclusion

RPC Study:

- First ever study on **Google's fleet-wide RPC characteristics**
- **722 billion** RPC traces over **~2 years** running on **100 production clusters**
- Provides insights on the characteristics of Google's geo-distributed **internal services**

Key contributions and findings:

- Storage **data flow optimization** is important
- **Millisecond**-scale RPCs are common – need to **balance** CPU **utilization** vs. **latency**
- **RPC Queuing matters** – need to improve **scheduling** and **load balancing**
- **RPC Latency Tax is significant at tail** – need better optimization within and across clusters to reduce **tail latency variation**
- **Load-balancing needs to account for latency** - co-optimize latency and utilization

Our measurements can influence **future RPC research.**

Google

Google

A Cloud-Scale Characterization of Remote Procedure Calls

Korakit Seemakhupt, Brent Stephens, Samira Khan, Sihang Liu, Hassan Wessel,
Soheil Hassas Yeganeh, Alex C. Snoeren, Arvind Krishnamurthy, David Culler, Henry M. Levy