SC24-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis

Communication, I/O, and Storage at Scale on Next-Generation Platforms - Scalable Infrastructures
(SC24 IXPUG Workshop)

Can Current SDS Controllers Scale To Modern HPC Infrastructures?

Mariana Miranda, *Yusuke Tanimura†, Jason Haga†, Amit Ruhela*, Stephen Lien Harrell*, John Cazes*,*
Ricardo Macedo, José Pereira, João Paulo
INESC TEC & University of Minho
†AIST
**TACC & UTAustin*

November 18, 2024

Modern HPC Infrastructures

- Modern supercomputers comprise **thousands of computes nodes**.

System	Rank	Rmax (PFLops/s)	Number of Nodes	Year
Frontier	1	1,206	9,408	2021
Aurora	2	1,012	10,624	2023
Fugaku	4	442	158,976	2020
Summit	9	148.6	4,608	2018
Frontera	33	23.52	8,368	2019

Extracted from Top500 list (June 2024)

- Enables large-scale parallel applications to run at massive scale.

Modern HPC Infrastructures

- Modern supercomputers comprise **thousands of computes nodes**.
- Enables large-scale parallel applications to run at massive scale.
    - Data-centric workloads (e.g., DL, LLM)

Competition over **shared storage resources**

↓

I/O contention and performance degradation

Modern HPC Infrastructures

- Modern supercomputers comprise **thousands of computes nodes**.
- Enables large-scale parallel applications to run at massive scale.
    - Data-centric workloads (e.g., DL, LLM)

⚠️HPC centers struggle to manage the shared load on their PFS systems efficiently.

The HPC Storage Challenge

Existing Solutions

- **Intrusive to critical HPC components**

  Solutions tightly coupled and intrusive to the system implementation (*e.g.*, GIFT, CALCioM, TBF)

  ⚠️ Low portability and maintainability!

- **Static and uncoordinated control**

  Enabling QoS control from the application-side (*e.g.*, OOOPS)

  ⚠️ Isolated and uncoordinated QoS!

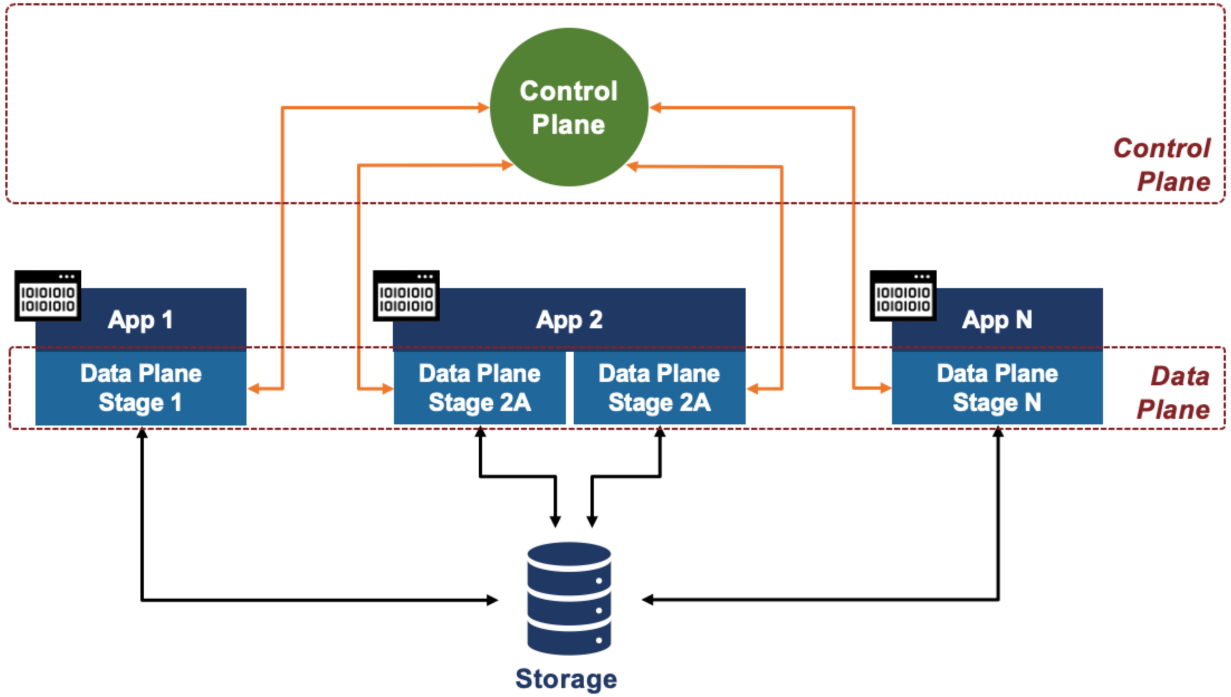
- **Software-defined storage solutions**

Software-Defined Storage

SDS solutions offer **non-intrusive**, **dynamic**, and **coordinated** control to manage storage **QoS across all HPC jobs**.

Software-Defined Storage

Overview



Control Plane

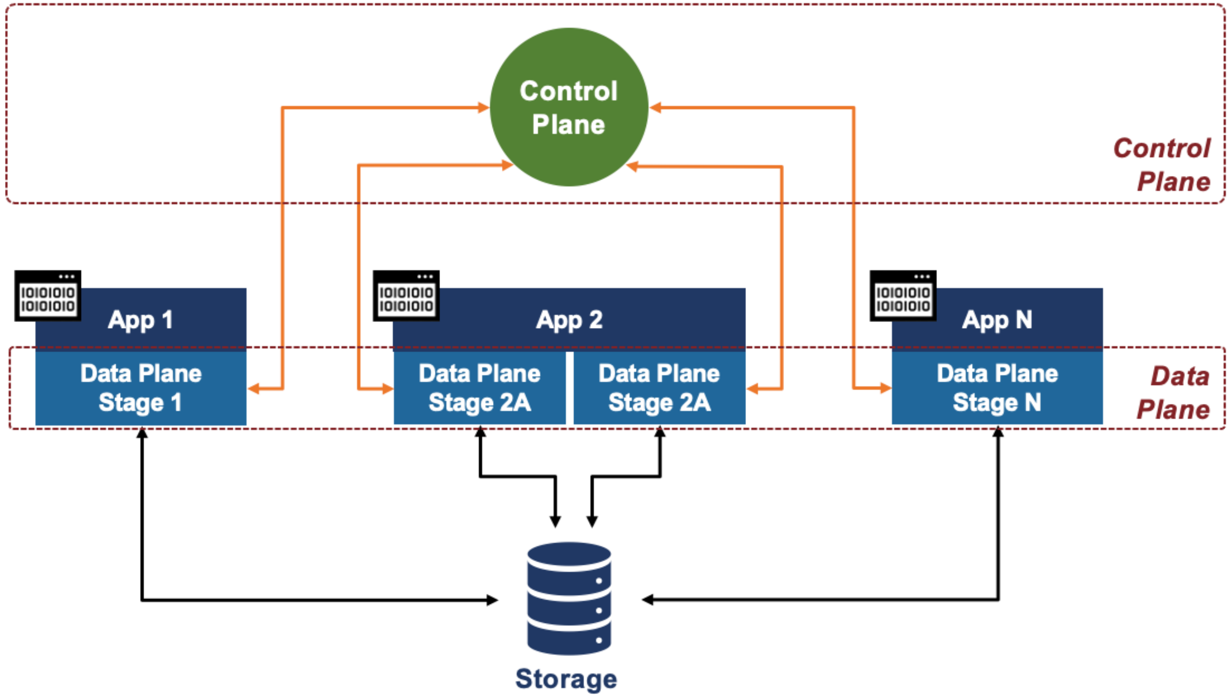
- Handles the **control logic.**
- **Logically centralized** and with **system-wide visibility.**

Data Plane

- **Intercepts I/O requests.**
- **Implements** and **executes** the policies defined by the control plane.

Software-Defined Storage

Overview

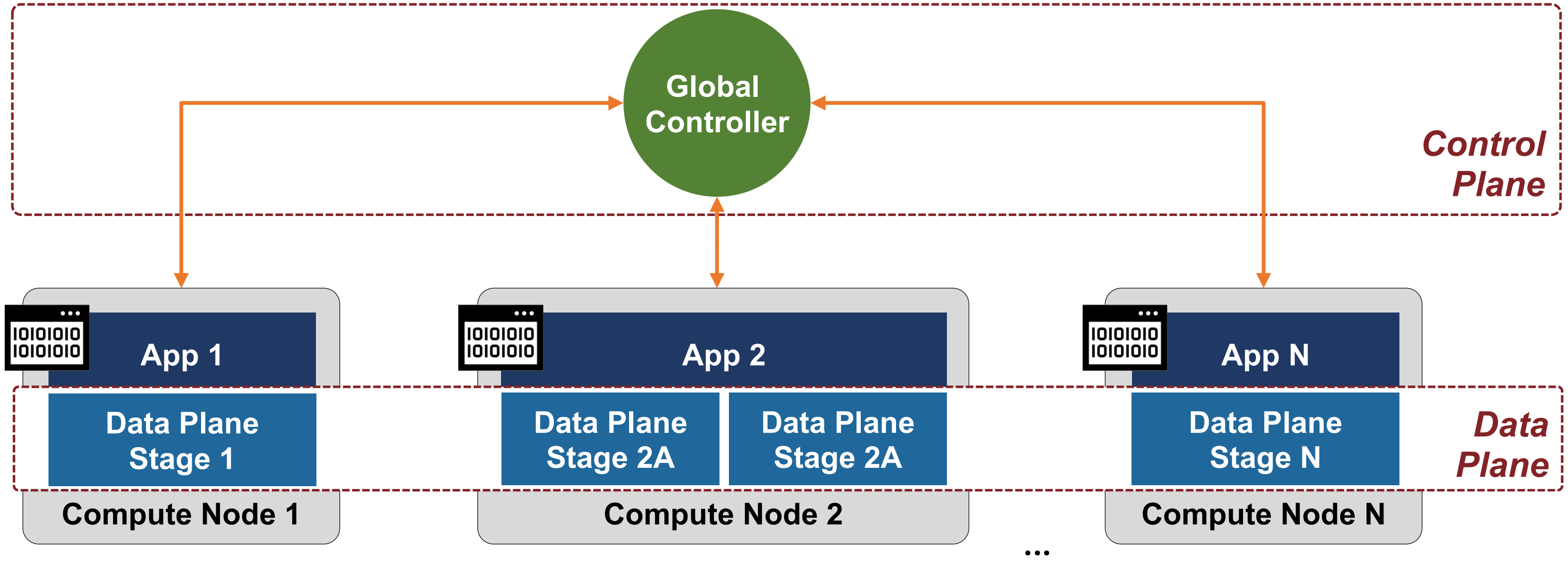


- Current SDS research **mainly focuses** on the **data plane**.
- SDS systems provide a very superficial control plane solution.
  - Generally overlook **scalability** and **dependability** properties.

This work focus on the **control plane's scalability.**

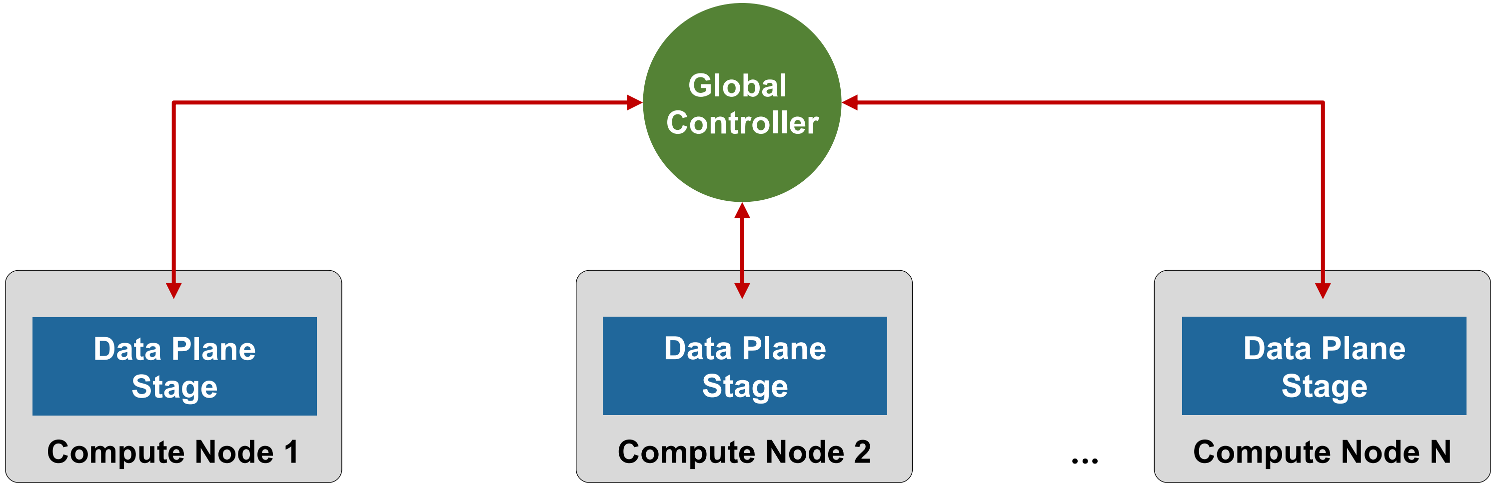
Control Plane Designs

Centralized



Control Plane Designs

Centralized[1]

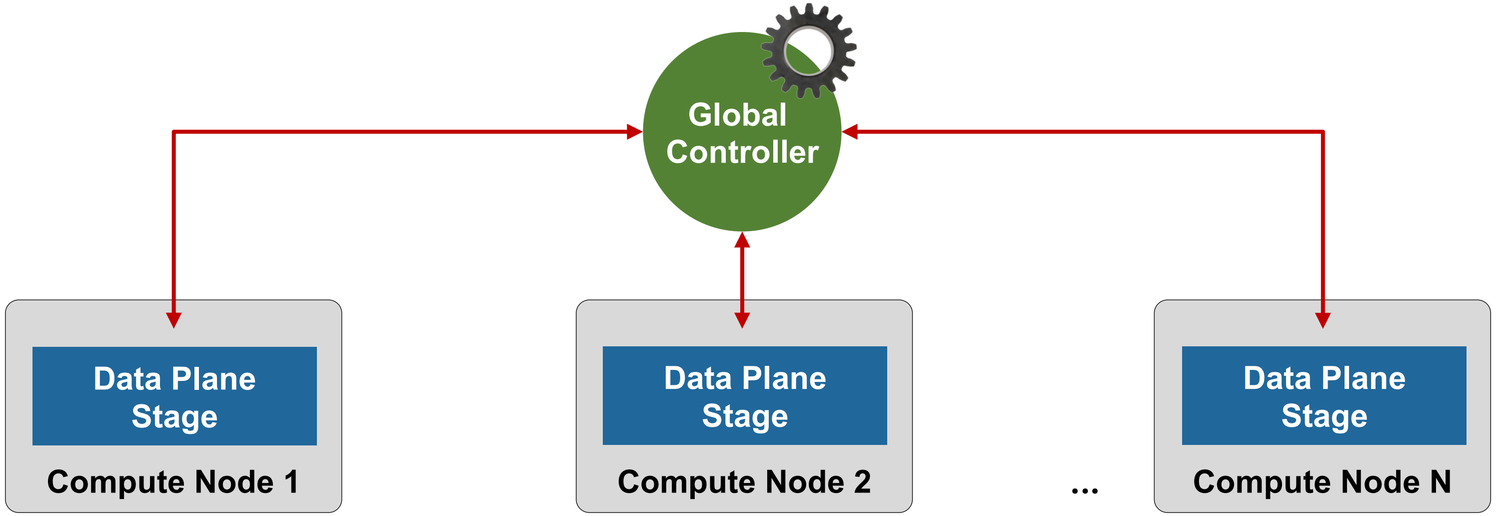


Implementation based on:

1. **Taming metadata intensive HPC jobs through dynamic, application-agnostic QoS control.**
   Ricardo Macedo, **Mariana Miranda**, Yusuke Tanimura, Jason Haga, Amit Ruhela, Stephen Lien Harrell, Richard Todd Evans, José Pereira, João Paulo.
   23rd IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGrid 23), 2023..

Control Plane Designs

Centralized[1]

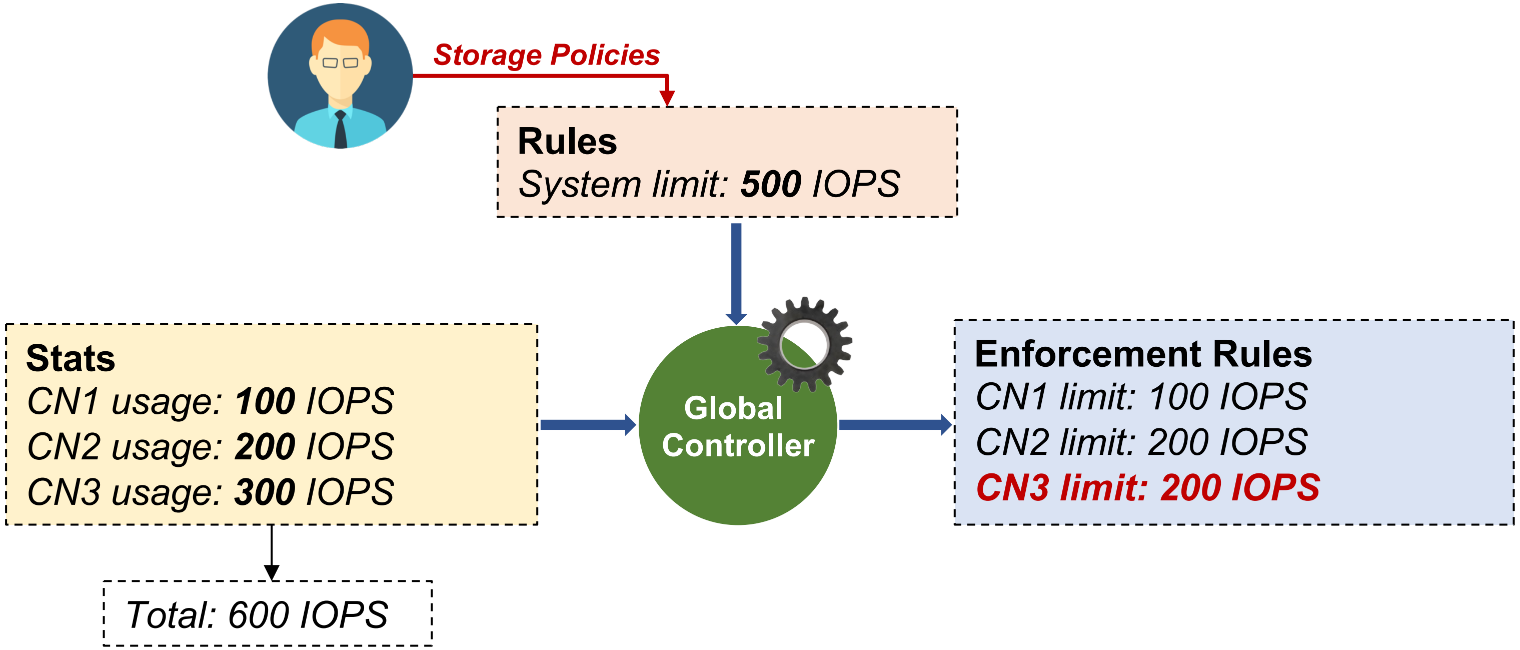


Feedback loop control cycle:

1. **Collects I/O metrics** from the data plane stages (*e.g.*, workflows’ rate).
2. **Computes** if all policies are being met.
3. **Enforces** new rates to respond to workload or system variations.

Control Plane Designs

Centralized[1]



Storage Policies

Rules
System limit: ***500*** *IOPS*

Stats
CN1 usage: ***100*** *IOPS*
CN2 usage: ***200*** *IOPS*
CN3 usage: ***300*** *IOPS*

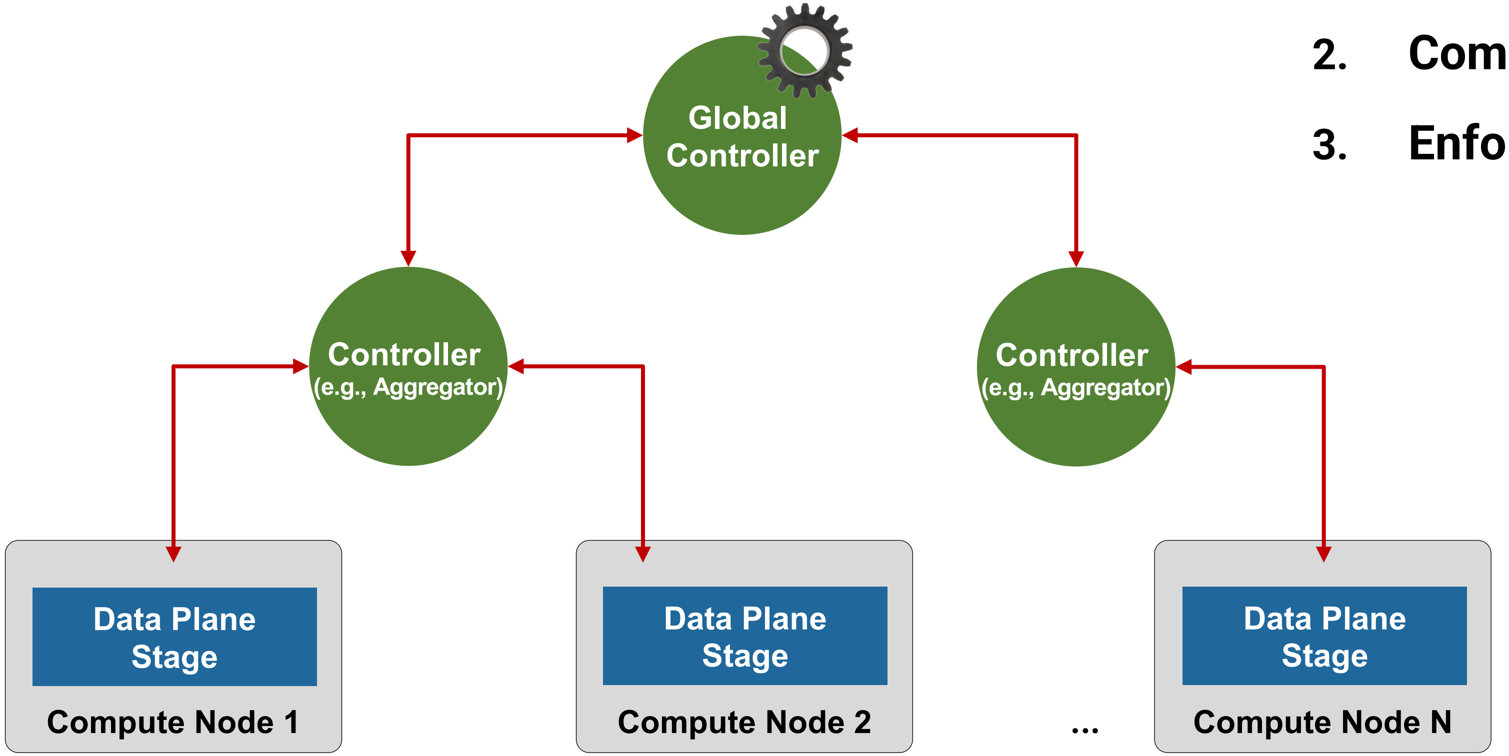
Total: 600 IOPS

Global Controller

Enforcement Rules
CN1 limit: 100 IOPS
CN2 limit: 200 IOPS
CN3 limit: 200 IOPS

Control Plane Designs

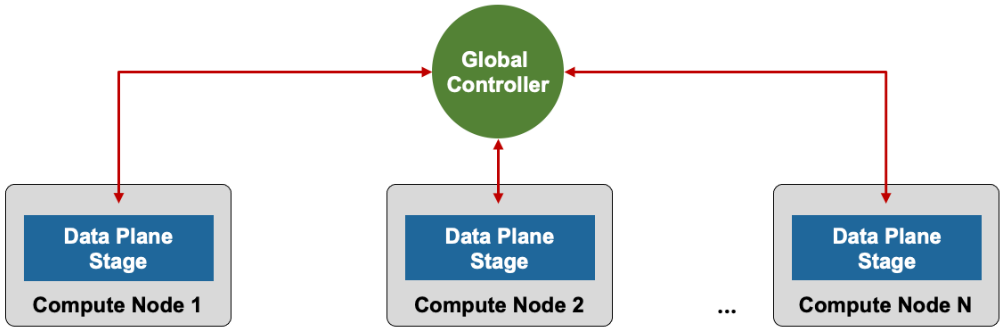
Hierarchical



1. **Collects I/O metrics**
2. **Compute**
3. **Enforce**

Control Plane Designs

Centralized



Global Controller

Data Plane Stage

Compute Node 1

Data Plane Stage

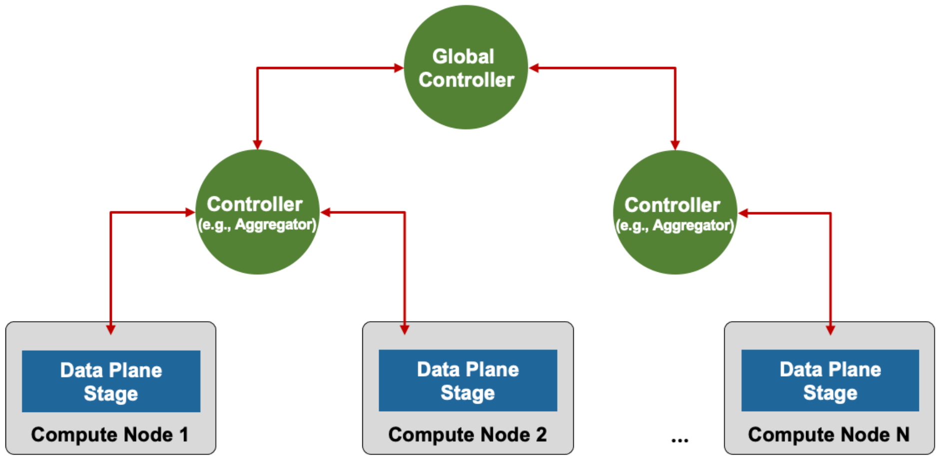
Compute Node 2

...

Data Plane Stage

Compute Node N

Hierarchical



Global Controller

Controller (e.g., Aggregator)

Controller (e.g., Aggregator)

Data Plane Stage

Compute Node 1

Data Plane Stage

Compute Node 2

...

Data Plane Stage

Compute Node N

Research Questions

- What is the scalability of single-node, flat-based control planes?
- Can hierarchical designs ensure better scalability than flat-based ones?
- What is the performance impact of adding more controllers across the hierarchy?
- How are the different phases of the control cycle impacted by each control plane design?

Study

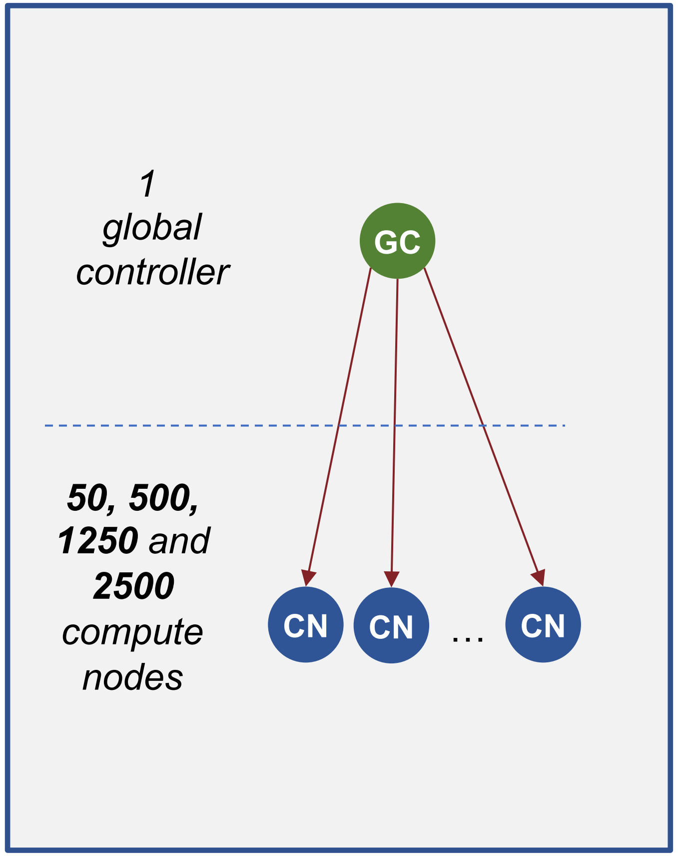
- **Experimental setup**
  - Compute nodes of the Frontera supercomputer.
    - Two 28-core Intel Xeon processors; 192 GiB of RAM; a single 240 GiB SSD; CentOS 7.9 with the Linux kernel v3.10.
    - Lustre file system as production PFS.

- **Workloads**
  - **Computation algorithm** *(at the control plane)*
    - Proportional sharing without false allocation (PSFA)
  - **Compute nodes workloads**
    - Synthetic workload that feeds statistics to the control plane.

- **Methodology**
  - Computation algorithm runs for 5+ minutes, repeated 3 times;
  - Data collected on the average control cycle latency, per control phase, and resource usage (CPU, memory, network) via REMORA tool.

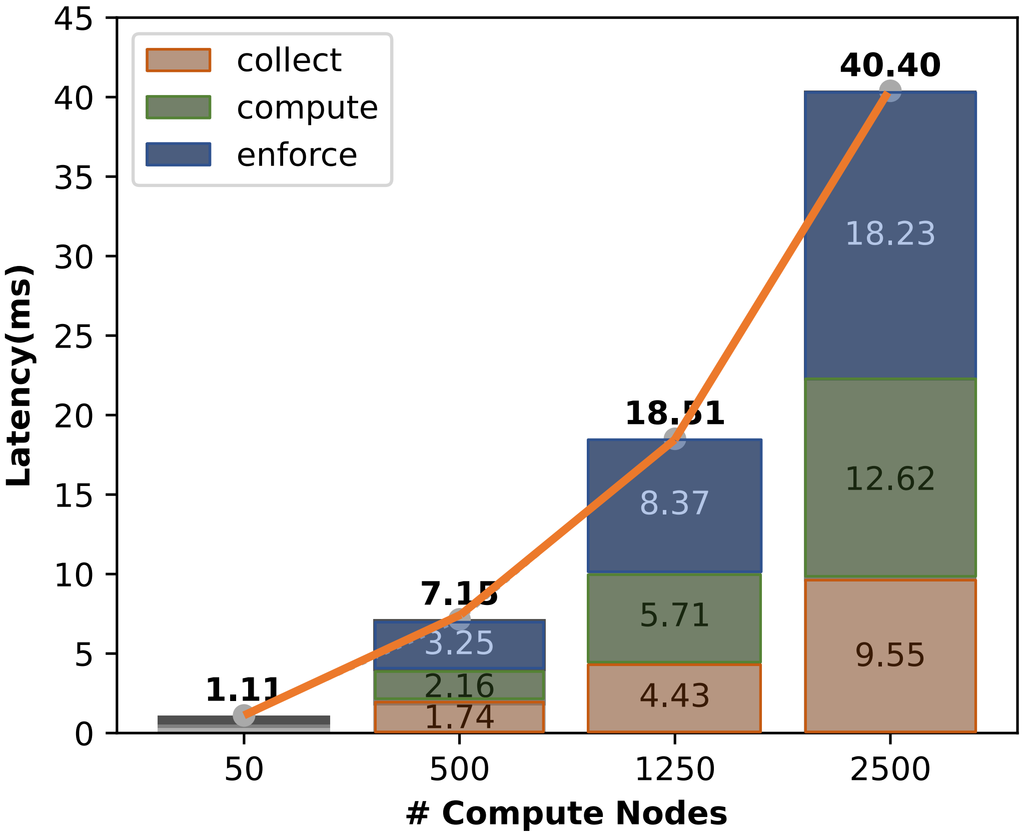
Evaluation

Centralized Design - Latency



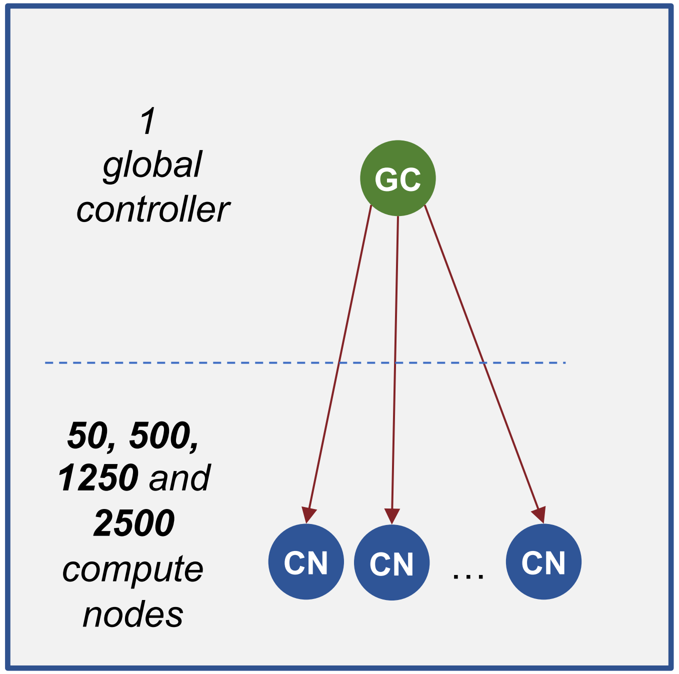
1 global controller

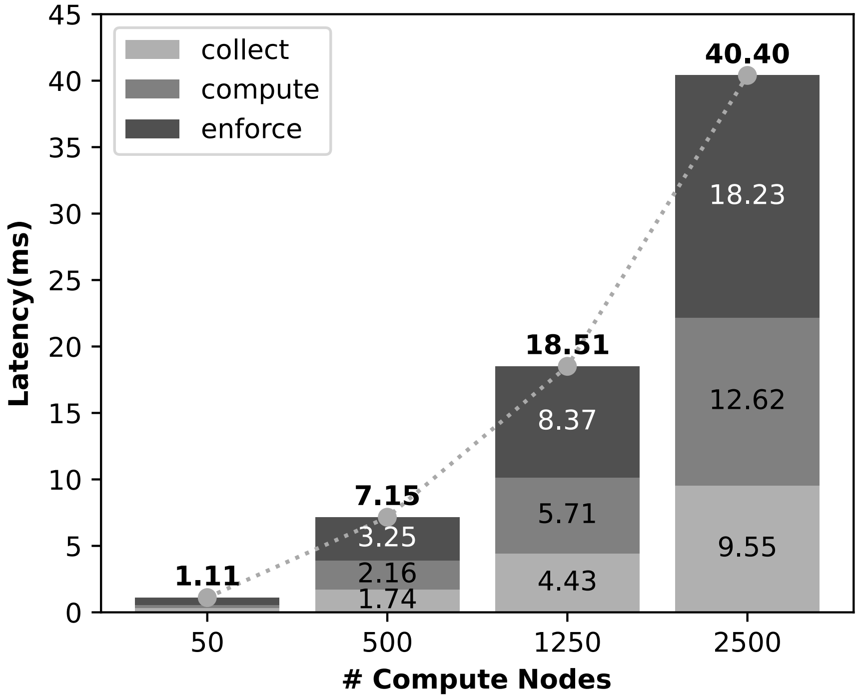
***50, 500, 1250** and **2500** compute nodes*



Evaluation

Centralized Design - Latency



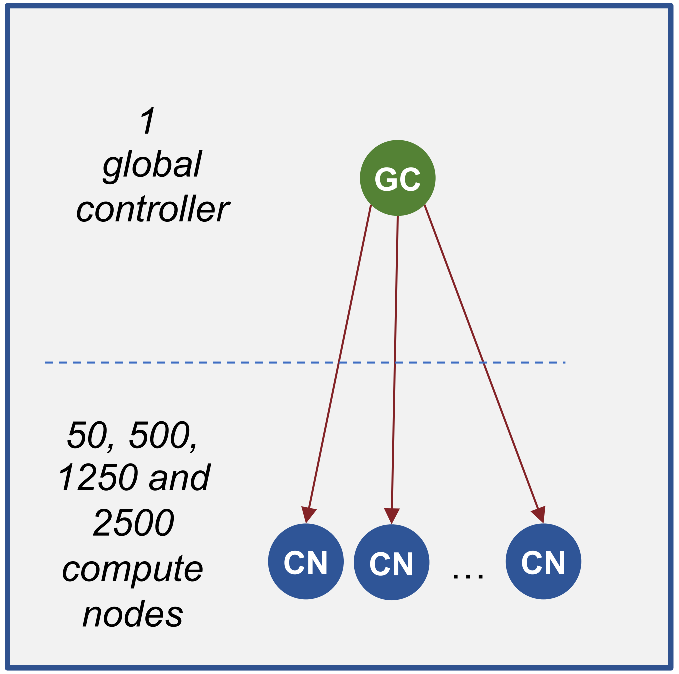


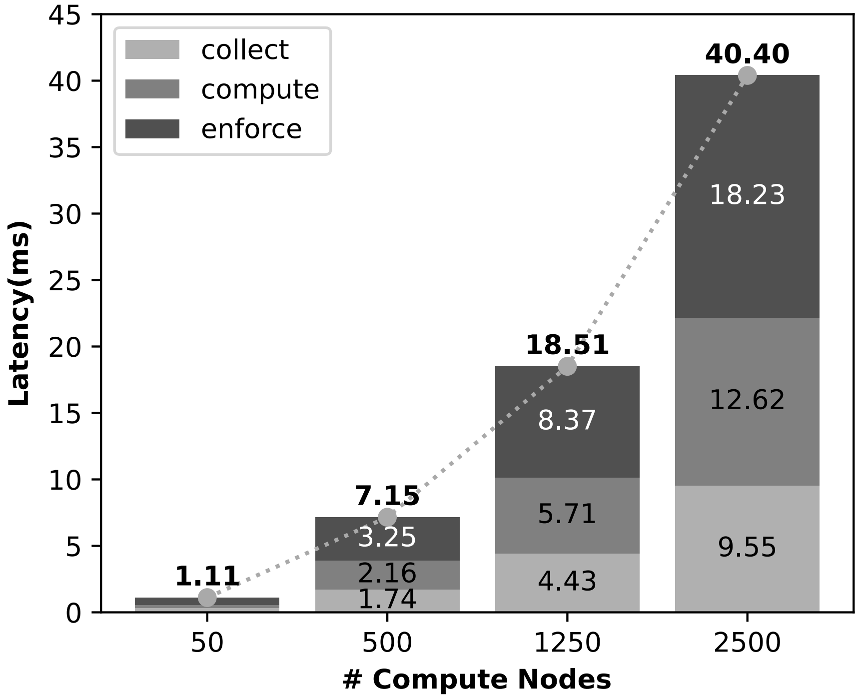
✓ Efficiently orchestrates up to 2,500 instances under 41 *ms*.

⚠ Network limitations at the global controller.

Evaluation

Centralized Design - Latency



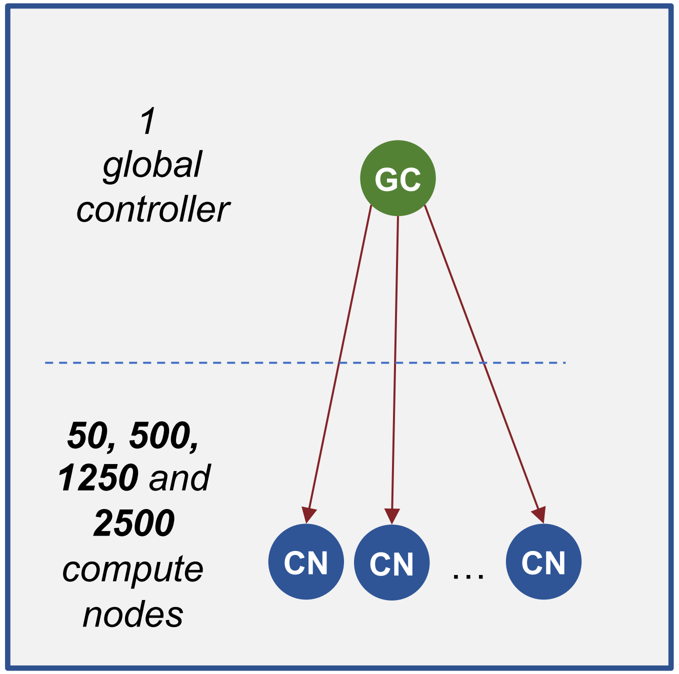


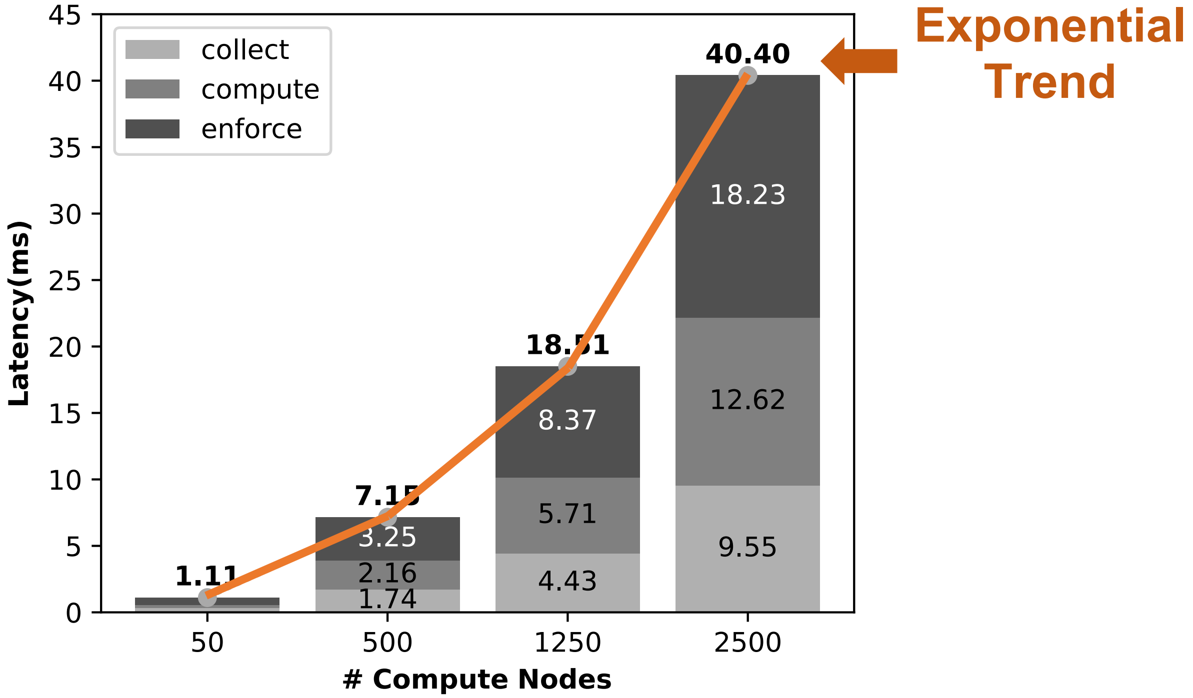
Observation #1:

A **flat control plane** with a single global controller suits **small to medium infrastructures**, adapting to I/O workload variations in tens of milliseconds.

Evaluation

Centralized Design - Latency



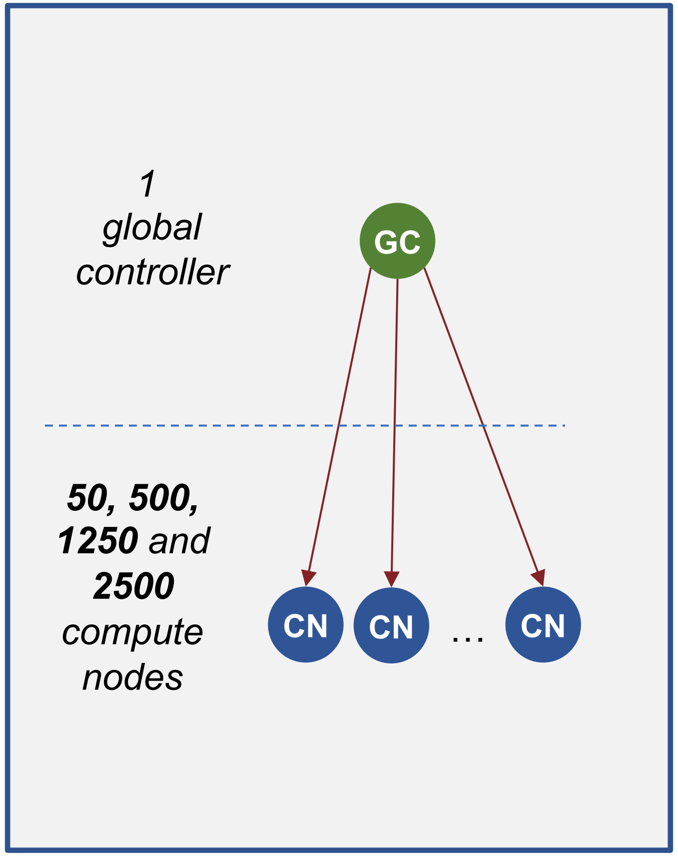


Observation #2:

A **single controller** handling computations and **network connections** creates a **scalability bottleneck**, mainly due to hardware limits.

Evaluation

Centralized Design – Resource Utilization



Controller	Resource	Setup			
		# Compute Nodes			
		50	*500*	*1250*	*2500*
Global	*CPU (%)*	6.07	9.58	10.39	10.34
	Memory (GB)	0.07	0.31	0.64	1.18
	Transmitted (MB/s)	5.67	8.74	8.74	9.73
	Received (MB/s)	3.74	5.75	5.74	5.36

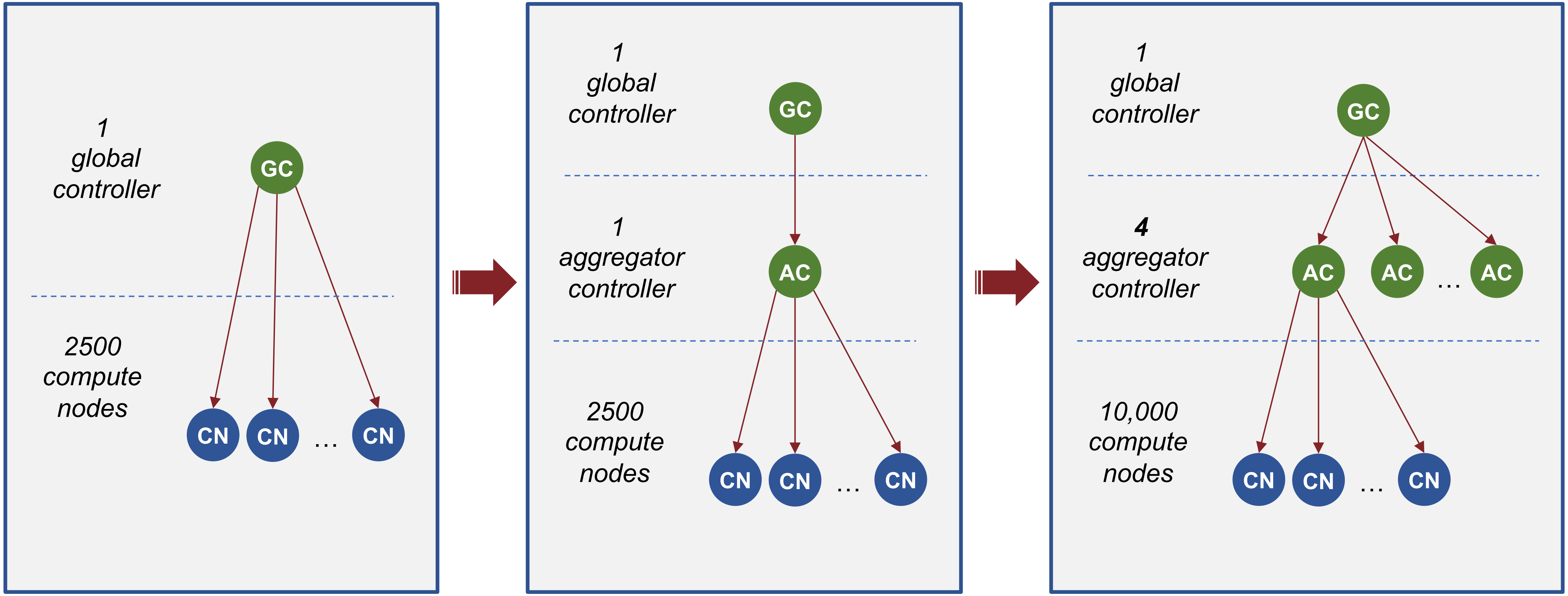
More Compute Nodes

⬇

Higher Resource Usage

Evaluation

Centralized Design To Hierarchical Design



1
global
controller

2500
compute
nodes

1
global
controller

1
aggregator
controller

2500
compute
nodes

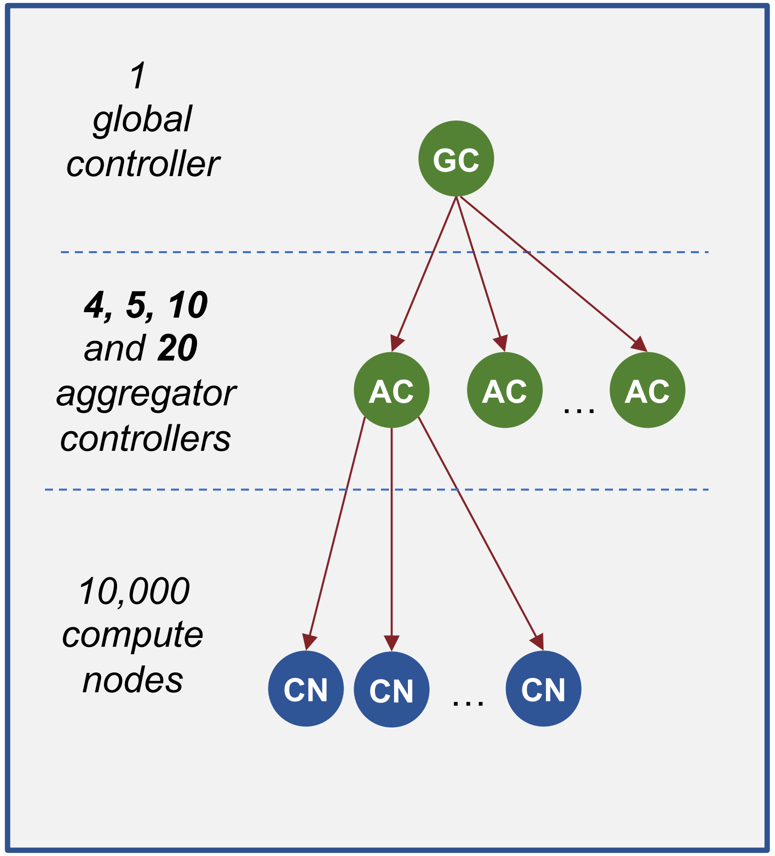
1
global
controller

4
aggregator
controller

10,000
compute
nodes

Evaluation

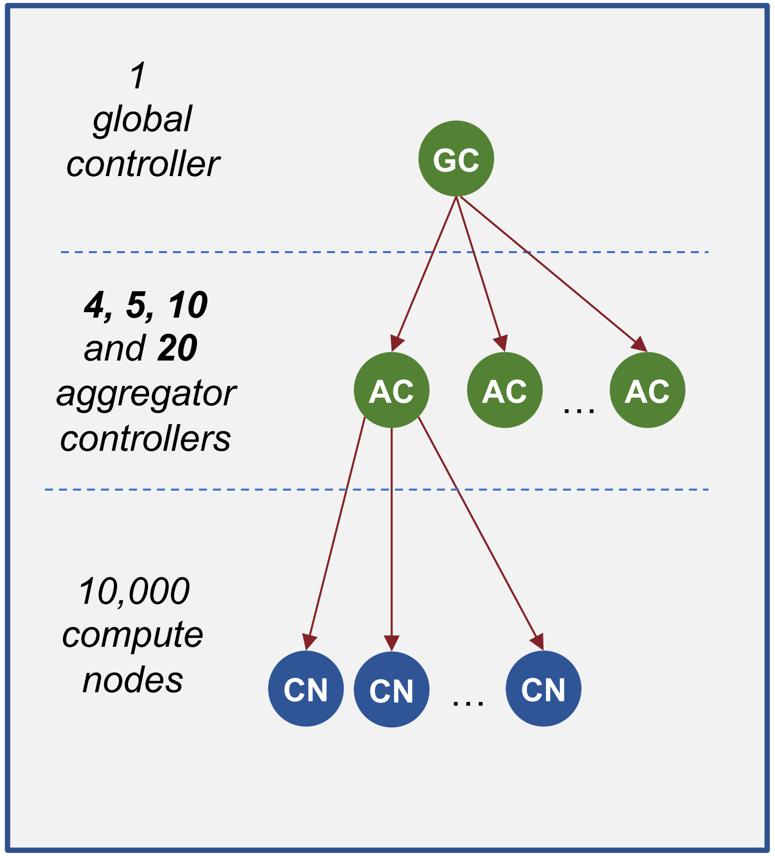
Hierarchical Design

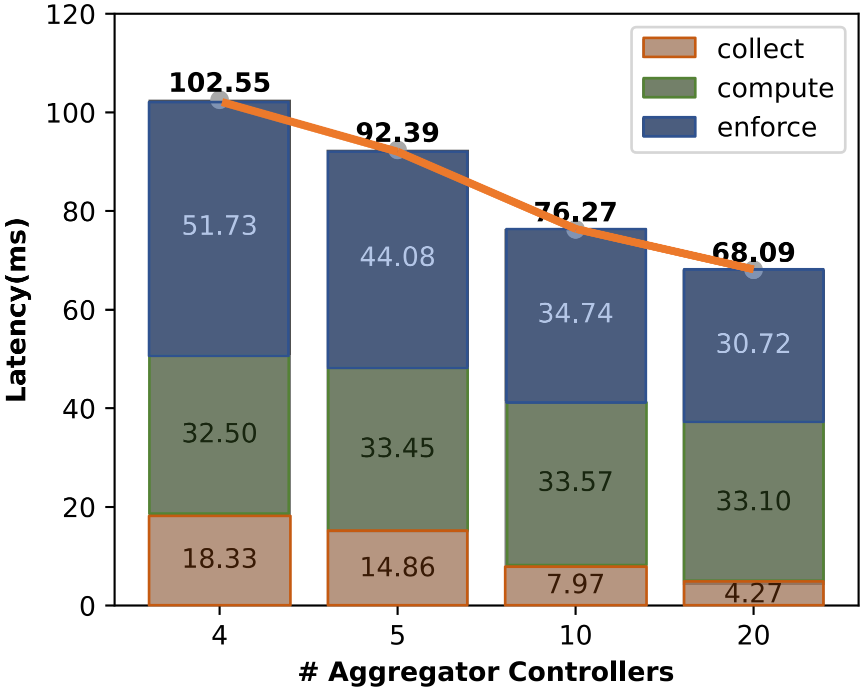


Aggregator controllers	Compute nodes per aggregator controller	Compute nodes
4	2,500	10,000
5	2,000	10,000
10	1,000	10,000
20	500	10,000

Evaluation

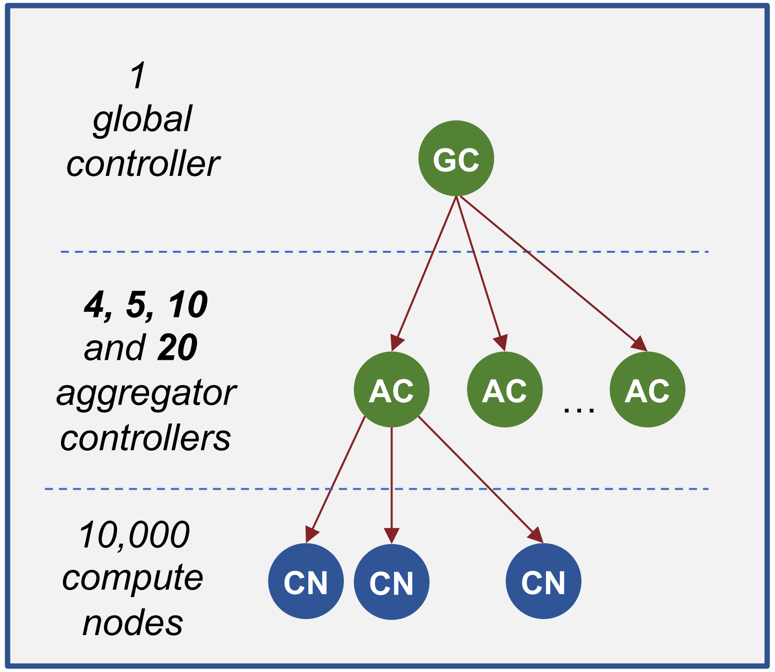
Hierarchical Design - Latency

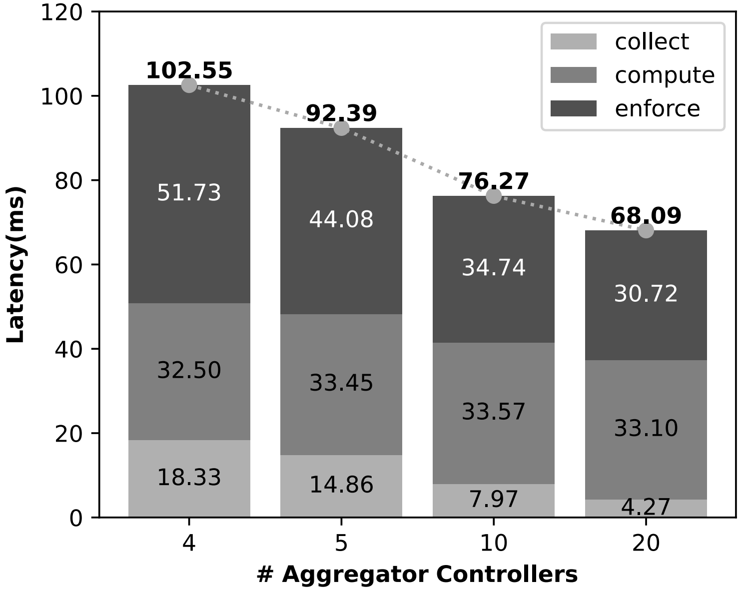




Evaluation

Hierarchical Design - Latency



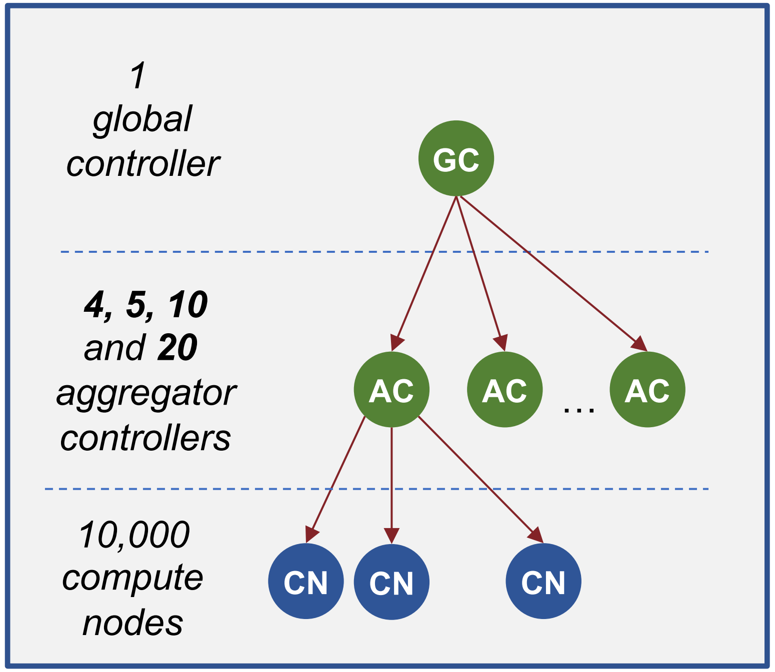


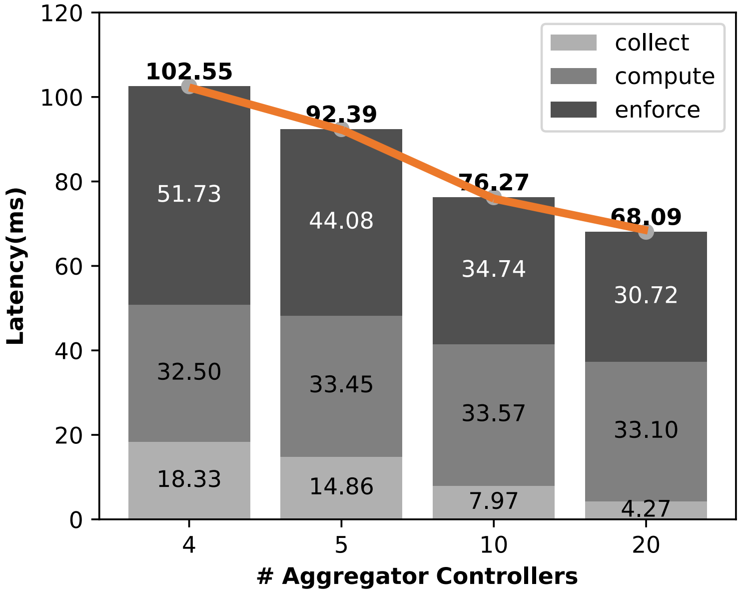
Observation #3:

A **hierarchical design** with **4 aggregator controllers** can scale to **10,000 nodes**, adapting to I/O workload variations in hundreds of milliseconds.

Evaluation

Hierarchical Design - Latency



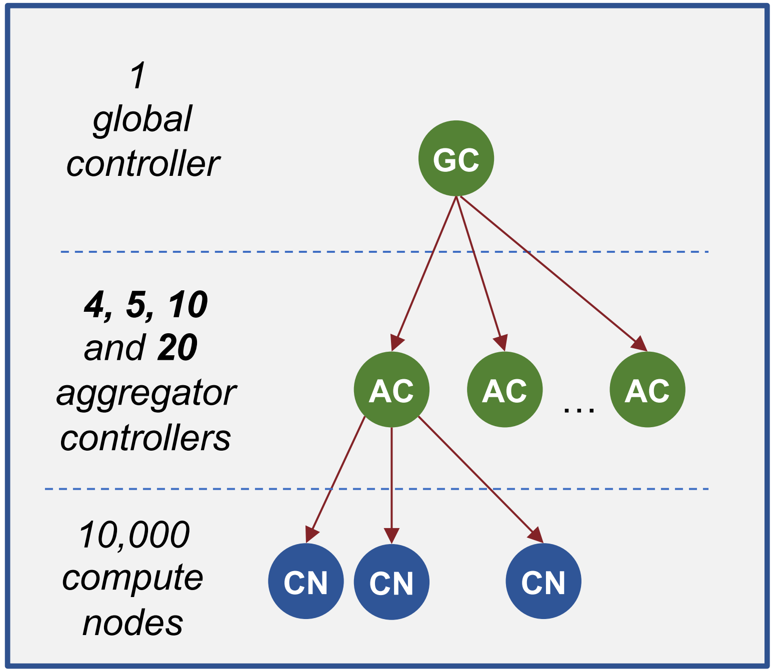


Observation #4:

Adding **more aggregator** controllers **lowers control cycle latency**, which is crucial for sustaining storage QoS in highly dynamic I/O workloads (e.g., burstiness).

Evaluation

Hierarchical Design - Resource Utilization



1 global controller

***4, 5, 10** and **20** aggregator controllers*

10,000 compute nodes

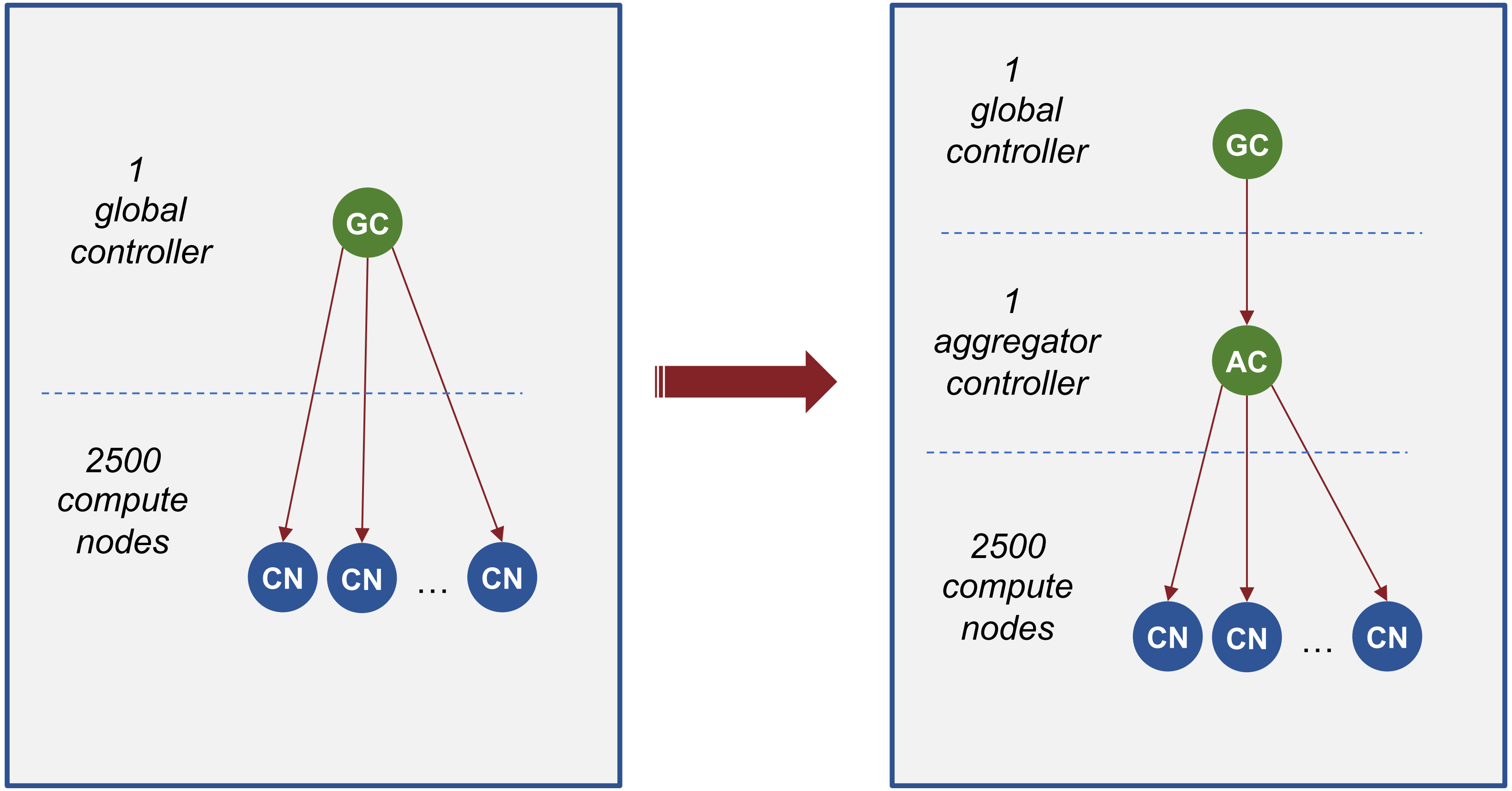
Controller	**Resource**	**Setup**			
		# Aggregator Controllers			
		4	***5***	***10***	***20***
Global	*CPU (%)*	2.55	2.81	3.22	3.52
	Memory (GB)	3.52	3.56	3.53	3.60
	Transmitted (MB/s)	4.39	4.73	5.66	6.08
	Received (MB/s)	1.45	1.58	1.82	1.98
Aggregator	*CPU (%)*	3.95	3.4	1.94	0.95
	Memory (GB)	0.16	0.13	0.08	0.04
	Transmitted (MB/s)	4.53	4.13	2.4	1.31
	Received (MB/s)	2.53	2.31	1.34	0.73

Observation #5:

There is a **trade-off** between the amount of **resources and control cycle latency** that must be chosen according to the needs of the targeted infrastructure.

Evaluation

Overhead of adding more control levels



1 global controller

2500 compute nodes

GC

CN CN … CN

1 global controller

1 aggregator controller

2500 compute nodes

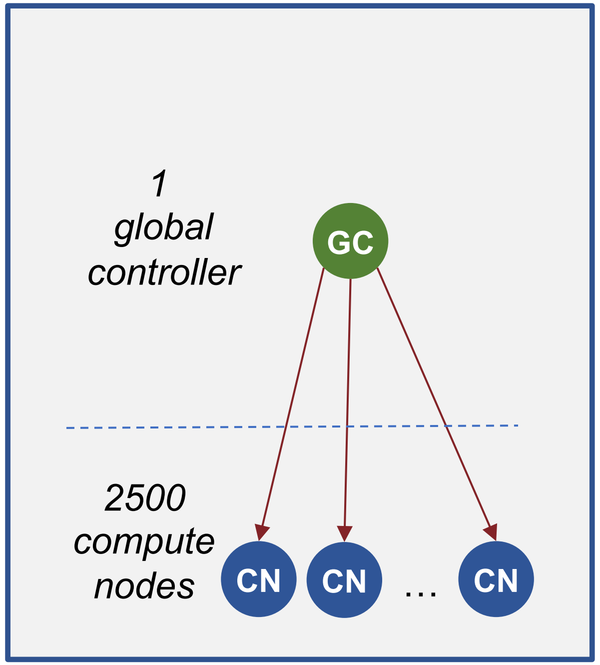
GC

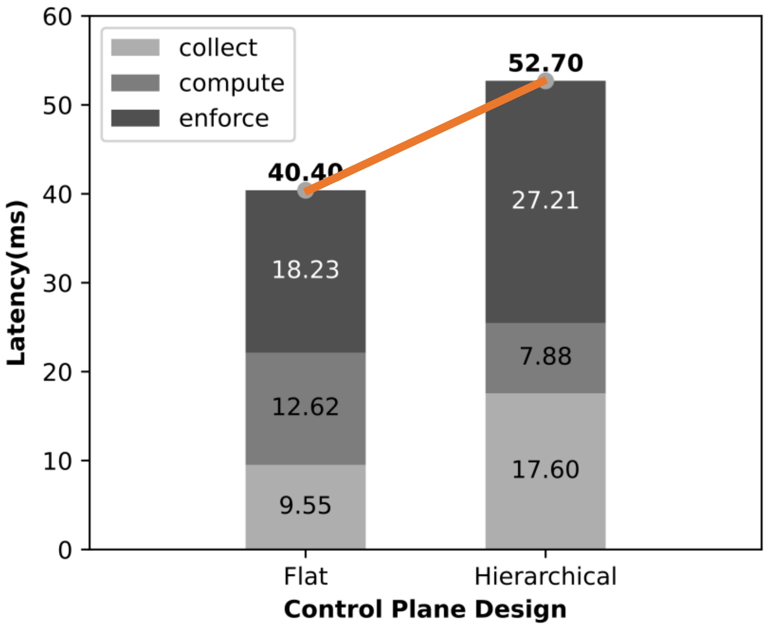
AC

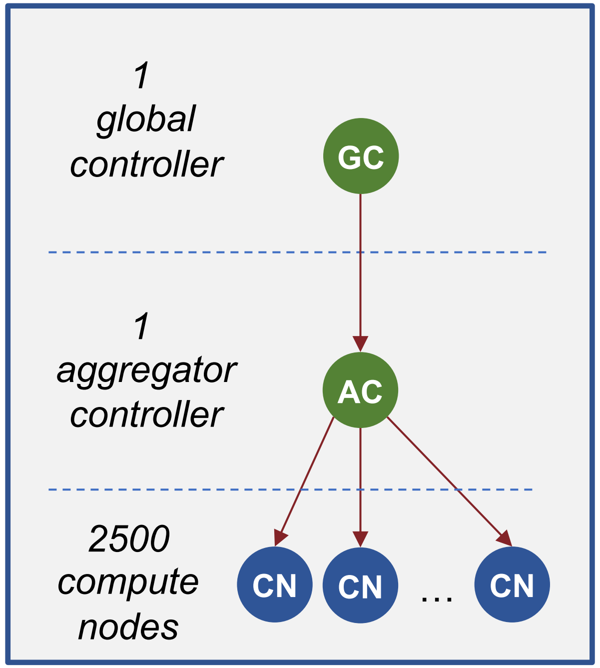
CN CN … CN

Evaluation

Overhead of adding more control levels





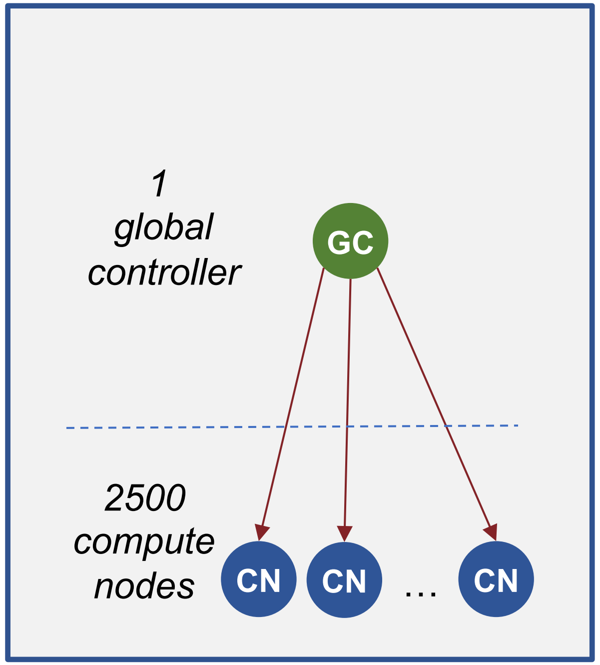


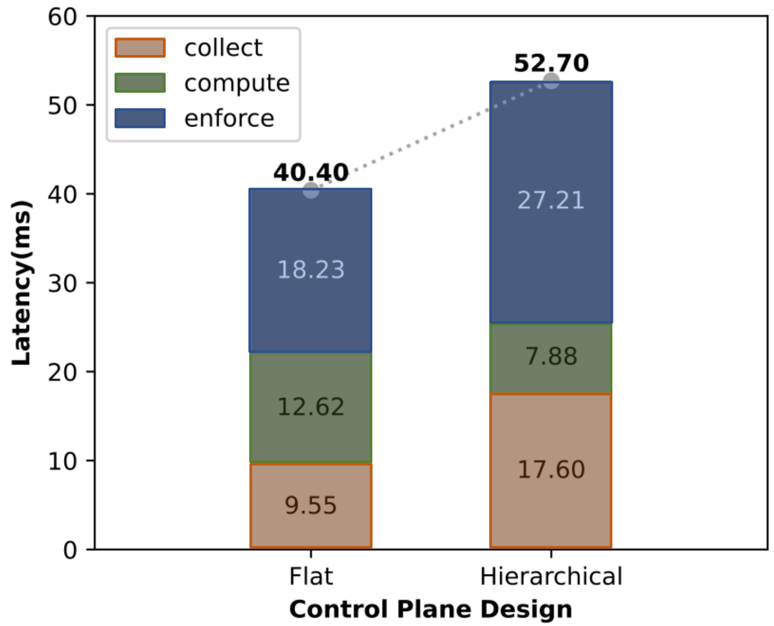
Observation #6:

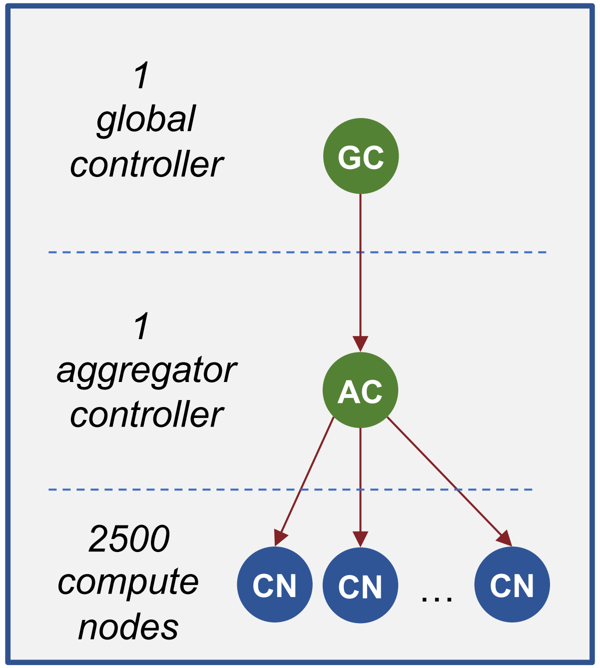
For up to 2,500 compute nodes, adding na **extra control level** raises the control cycle latency by up to **12.3 ms**, but keeps it below 53 ms.

Evaluation

Overhead of adding more control levels







Observation #7:

Adding a dedicated control level for aggregating metrics **reduces the latency** of the **compute phase**.

Outcomes

- This study is the first to assess the scalability of two common control plane designs.
- A **single flat controller** handles **2,500 nodes** with 41 ms average latency.
- Scaling beyond requires a **hierarchical design** to avoid scalability **limits** in **network connections**.

Outcomes

- Adding an **extra control layer** controller for metrics aggregation and rule enforcement **increases latency** by only 12 ms.
- With **4 aggregators**, the system **scales to 10,000 nodes**, keeping latency under 103 ms; **more aggregators** can **reduce the latency** to under 69 ms.
- Trade-off: **more aggregators improve latency** but require more resources; choices depend on workload needs.

Outcomes

- A 103 ms control cycle, even while managing 10,000 nodes, stays **well below** any anticipated **intervention threshold**.
- **Allows for a prompt response to bursty workloads!**

Future Work

- New research directions include:
    - **Flat designs with multiple coordinated controllers** managing different nodes.
    - **Hierarchical designs with processing offloaded to aggregators** for reduced top-level load.
- Test with **real workloads**.
- **Control plane dependability** is critical, as controller failures may result in outdated I/O rules.

SC24-W: Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis

Communication, I/O, and Storage at Scale on Next-Generation Platforms - Scalable Infrastructures
(SC24 IXPUG Workshop)

Can Current SDS Controllers Scale To Modern HPC Infrastructures?

Mariana Miranda, Yusuke Tanimura[†], Jason Haga[†], Amit Ruhela*, Stephen Lien Harrell*, John Cazes*,
Ricardo Macedo, José Pereira, João Paulo
INESC TEC & University of Minho
[†]AIST
*TACC & UTAustin

November 18, 2024