21st Workshop on Workflows in Support of Large-Scale Science (WORKS)
The International Conference for High Performance Computing, Networking, Storage & Analysis (SC 26)

How far can network congestion awareness improve scientific workflow scheduling?

Luis Antonio Momm Duarte, Guilherme Piêgas Koslovski

Laboratório de Processamento Paralelo e Distribuído (LabP2D)
Santa Catarina State University (UDESC), Joinville, SC, Brazil

guilherme.koslovski@udesc.br

UDESC UNIVERSIDADE DO ESTADO DE SANTA CATARINA | JOINVILLE CENTRO DE CIÊNCIAS TECNOLÓGICAS

LabP2D

Introduction

- Network congestion is a persistent phenomenon in HPC infrastructures
- Congestion arises from communication contention among tasks within the same workflow or across concurrent workflows
- Impact on makespan
- **It is a fact!**

End-to-end congestion control in datacenter networks: a survey

Zejia ZHOU[1*], Shan HUANG[1,2,3*], Dezun DONG[1]✉, Yang BAI[1], Liquan XIAO[1]

1. College of Computer Science and Technology, National University of Defense Technology, Changsha 410073, China
2. The 63rd Research Institute, National University of Defense Technology, Nanjing 210000, China
3. Laboratory for Big Data and Decision, Changsha 410073, China

Characterizing the Impact of Congestion in Modern HPC Interconnects

Lorenzo Piarulli*, Marco Faltelli†, Dirk Pleiter‡, Karthee Sivalingam‡§, Dancheng Zhang§, Kexue Zhao§, Matteo Turisini¶, Francesco Iannone†, Aldo Artigiani§, Daniele De Sensi*

*Sapienza University of Rome, {piarulli, desensi}@di.uniroma1.it
†ENEA, {marco.faltelli, francesco.iannone}@enea.it
ter@open-edge-hpc-initiative.org
haokexue, aldo.artigiani}@huawei.com
i@cineca.it

IEEE TRANSACTIONS ON CLOUD COMPUTING, VOL. 2, NO. 2, APRIL-JUNE 2014

Deadline Based Resource Provisioning and Scheduling Algorithm for Scientific Workflows on Clouds

Maria Alejandra Rodriguez and Rajkumar Buyya

ELSEVIER

Future Generation Computer Systems
Volume 89, December 2018, Pages 360-374



Congestion control in high-speed lossless data center networks: A survey

Shan Huang [a] ✉, Dezun Dong [a] ✉, Wei Bai [b] ✉

Guilherme P. Koslovski
guilherme.koslovski@udesc.br

Motivation

- Simulation: WRENCH, SimGrid, and Pegasus workflows
- Topologies: Tree, Fat-tree, and Dragonfly
  - 256 servers and 100Gbps links.

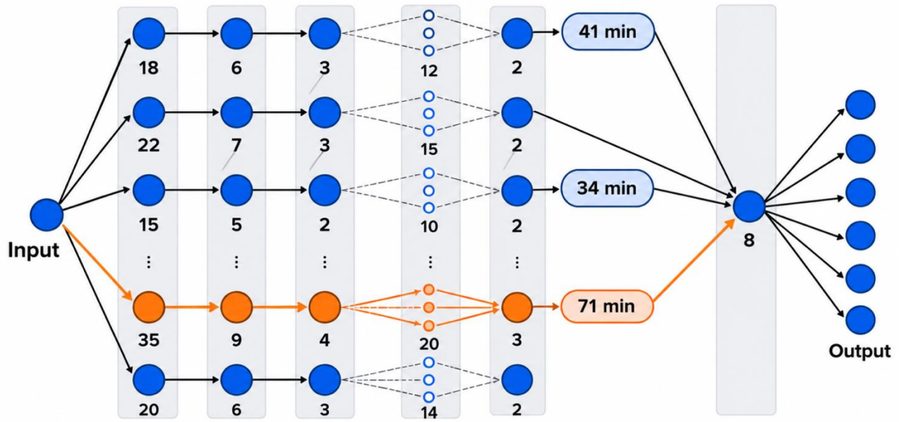
- The *Utopian* scenario: no network limits!

- Makespan-based metrics:
  Congestion Overhead (CO) = (Real – Utopian) / Real

	Tree		Fat-tree		Dragonfly	
Workflow	**CO**	**CI**	**CO**	**CI**	**CO**	**CI**
blast	35.8%	0.56	61.2%	1.58	91.1%	10.29
1000genome	26.9%	0.37	40.5%	0.68	81.9%	4.51
soykb	16.6%	0.20	30.6%	0.44	76.2%	3.20
montage	5.9%	0.06	9.0%	0.10	56.3%	1.29
epigenomics	1.7%	0.02	2.8%	0.03	30.1%	0.43
srasearch	1.8%	0.02	2.1%	0.02	27.5%	0.38
cycles	0.1%	0.00	0.1%	0.00	1.0%	0.01
seismology	0.1%	0.00	0.3%	0.00	0.0%	0.00

Takeaways:

- Even a mechanism that eliminated contention entirely is limited by CO
- The impact varies by target topology
- Critical path must be protected



UDESC UNIVERSIDADE DO ESTADO DE SANTA CATARINA | JOINVILLE CENTRO DE CIÊNCIAS TECNOLÓGICAS

LabP2D

Guilherme P. Koslovski
guilherme.koslovski@udesc.br

Objective

- Central question: When it is worthwhile for a workflow management system (WMS) to act upon it?

  - WMS cannot directly control network resources

  - WMS has complete knowledge and control over workflows: execution order and communication timing

- Opportunity: Workflow-level congestion avoidance

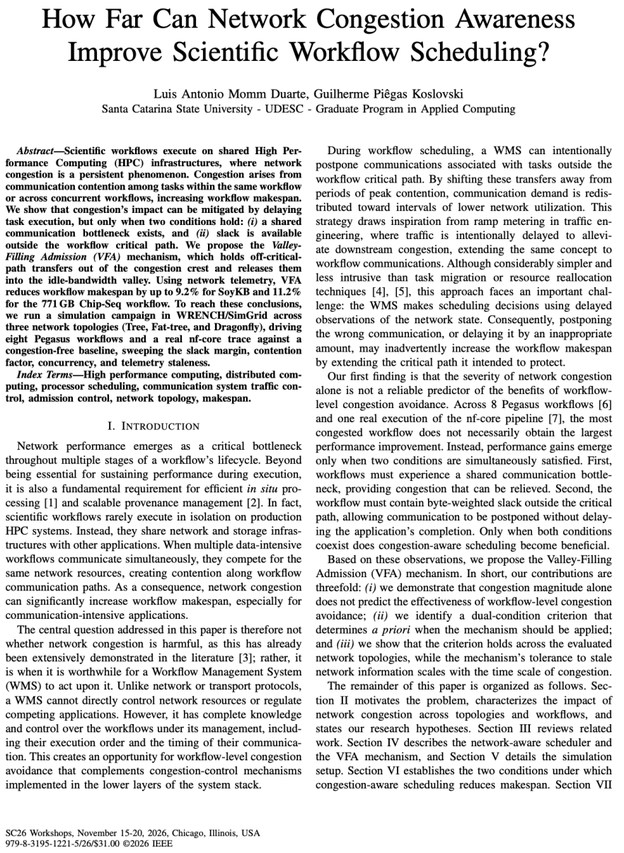
  - We can postpone communications outside the workflow critical path

  - It is intuitive, but how? Is it always useful?

Guilherme P. Koslovski
guilherme.koslovski@udesc.br

Agenda

- Related work
- **Network-aware scheduler and the Valley-Filling Admission (VFA) mechanism**
- Simulation setup
- **Conditions under which congestion avoidance reduces makespan**
- Testing the criterion on isolated workflows
- Generalizing to a real production trace
- **VFA gains with concurrency**
- How stale can the signal be before congestion awareness backfires?
- **Conclusion**



How Far Can Network Congestion Awareness Improve Scientific Workflow Scheduling?

Luis Antonio Momm Duarte, Guilherme Piêgas Koslovski
Santa Catarina State University - UDESC - Graduate Program in Applied Computing

***Abstract*—Scientific workflows execute on shared High Performance Computing (HPC) infrastructures, where network congestion is a persistent phenomenon. Congestion arises from communication contention among tasks within the same workflow or across concurrent workflows, increasing workflow makespan. We show that congestion's impact can be mitigated by delaying task execution, but only when two conditions hold: *(i)* a shared communication bottleneck exists, and *(ii)* slack is available outside the workflow critical path. We propose the *Valley-Filling Admission (VFA)* mechanism, which holds off-critical-path transfers out of the congestion crest and releases them into the idle-bandwidth valley. Using network telemetry, VFA reduces workflow makespan by up to 9.2% for SoyKB and 11.2% for the 771 GB Chip-Seq workflow. To reach these conclusions, we run a simulation campaign in WRENCH/SimGrid across three network topologies (Tree, Fat-tree, and Dragonfly), driving eight Pegasus workflows and a real nf-core trace against a congestion-free baseline, sweeping the slack margin, contention factor, concurrency, and telemetry staleness.**

***Index Terms*—High performance computing, distributed computing, processor scheduling, communication system traffic control, admission control, network topology, makespan.**

I. Introduction

Network performance emerges as a critical bottleneck throughout multiple stages of a workflow's lifecycle. Beyond being essential for sustaining performance during execution, it is also a fundamental requirement for efficient *in situ* processing [1] and scalable provenance management [2]. In fact, scientific workflows rarely execute in isolation on production HPC systems. Instead, they share network and storage infrastructures with other applications. When multiple data-intensive workflows communicate simultaneously, they compete for the same network resources, creating contention along workflow communication paths. As a consequence, network congestion can significantly increase workflow makespan, especially for communication-intensive applications.

The central question addressed in this paper is therefore not whether network congestion is harmful, as this has already been extensively demonstrated in the literature [3]; rather, it is when it is worthwhile for a Workflow Management System (WMS) to act upon it. Unlike network or transport protocols, a WMS cannot directly control network resources or regulate competing applications. However, it has complete knowledge and control over the workflows under its management, including their execution order and the timing of their communication. This creates an opportunity for workflow-level congestion avoidance that complements congestion-control mechanisms implemented in the lower layers of the system stack.

During workflow scheduling, a WMS can intentionally postpone communications associated with tasks outside the workflow critical path. By shifting these transfers away from periods of peak contention, communication demand is redistributed toward intervals of lower network utilization. This strategy draws inspiration from ramp metering in traffic engineering, where traffic is intentionally delayed to alleviate downstream congestion, extending the same concept to workflow communications. Although considerably simpler and less intrusive than task migration or resource reallocation techniques [4], [5], this approach faces an important challenge: the WMS makes scheduling decisions using delayed observations of the network state. Consequently, postponing the wrong communication, or delaying it by an inappropriate amount, may inadvertently increase the workflow makespan by extending the critical path it intended to protect.

Our first finding is that the severity of network congestion alone is not a reliable predictor of the benefits of workflow-level congestion avoidance. Across 8 Pegasus workflows [6] and one real execution of the nf-core pipeline [7], the most congested workflow does not necessarily obtain the largest performance improvement. Instead, performance gains emerge only when two conditions are simultaneously satisfied. First, workflows must experience a shared communication bottleneck, providing congestion that can be relieved. Second, the workflow must contain byte-weighted slack outside the critical path, allowing communication to be postponed without delaying the application's completion. Only when both conditions coexist does congestion-aware scheduling become beneficial.

Based on these observations, we propose the Valley-Filling Admission (VFA) mechanism. In short, our contributions are threefold: *(i)* we demonstrate that congestion magnitude alone does not predict the effectiveness of workflow-level congestion avoidance; *(ii)* we identify a dual-condition criterion that determines *a priori* when the mechanism should be applied; and *(iii)* we show that the criterion holds across the evaluated network topologies, while the mechanism's tolerance to stale network information scales with the time scale of congestion.

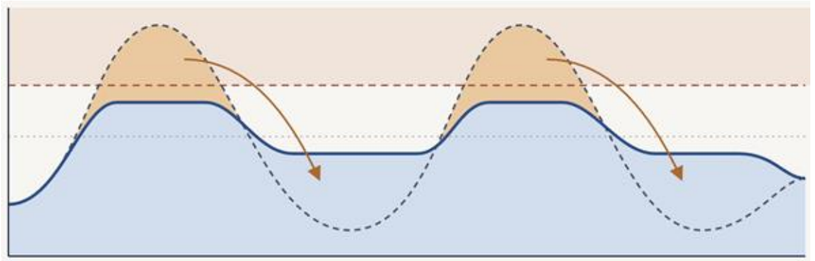
The remainder of this paper is organized as follows. Section II motivates the problem, characterizes the impact of network congestion across topologies and workflows, and states our research hypotheses. Section III reviews related work. Section IV describes the network-aware scheduler and the VFA mechanism, and Section V details the simulation setup. Section VI establishes the two conditions under which congestion-aware scheduling reduces makespan. Section VII

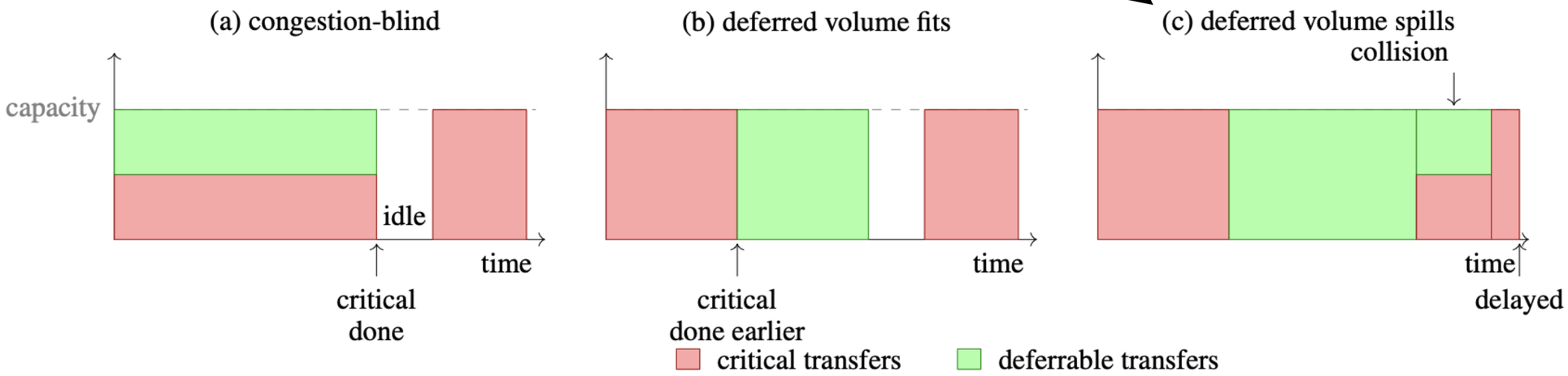
SC26 Workshops, November 15-20, 2026, Chicago, Illinois, USA
979-8-3195-1221-5/26/$31.00 ©2026 IEEE

Network-aware scheduler and the Valley-Filling Admission (VFA) mechanism

- Admission control between WMS scheduler and data transfer:
  - Tasks are scheduled by any traditional policy (e.g., HEFT), VFA is agnostic to it
  - VFA acts on the timing of data transfers
    - Critical-path transfers are released immediately

- It requires efficient (response time, fresh signal) network telemetry



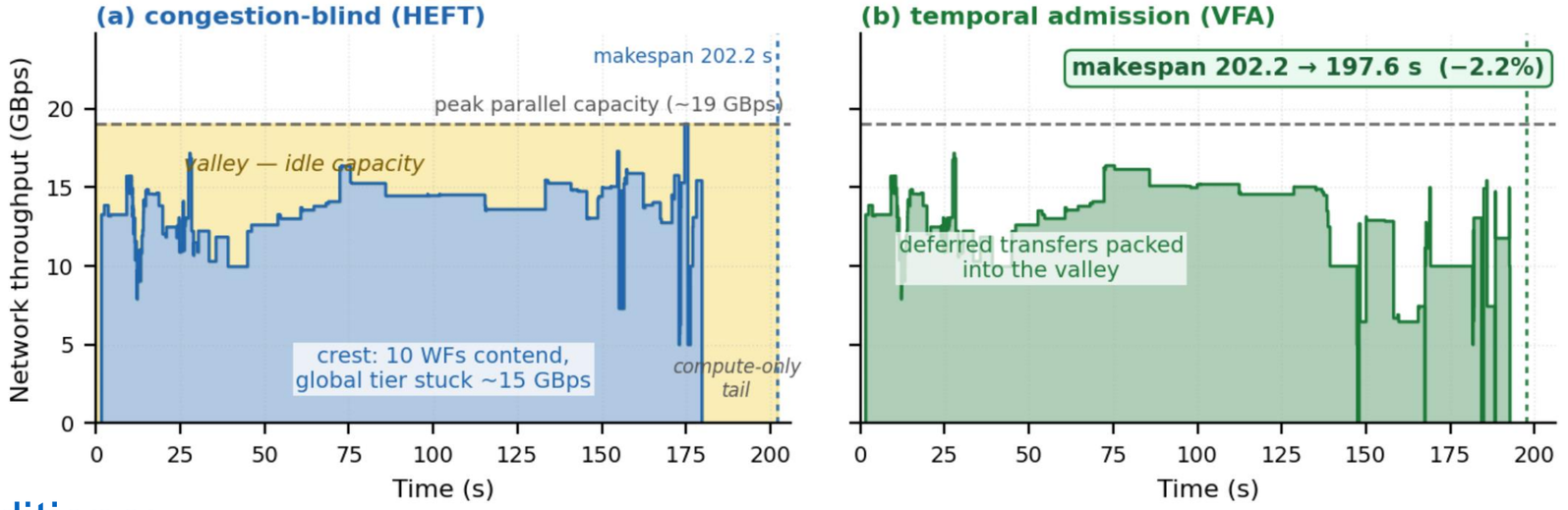


Guilherme P. Koslovski
guilherme.koslovski@udesc.br

Conditions under which congestion avoidance reduces makespan

- Intuitively, it depends on workflow structure: congestion crests and idle-bandwidth valleys

- We quantify the slack of tasks outside the critical path as difference between the latest time at which its data transfer may start without extending the critical path and its earliest possible start.



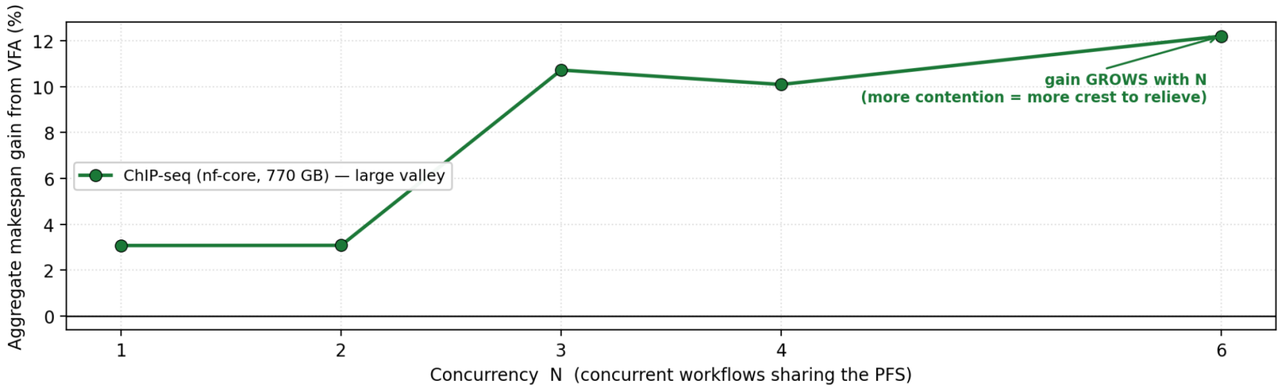
- 10 SoyKB workflows sharing a 256-servers Dragonfly

- Conditions:
  (1) A shared communication bottleneck must exists: online monitoring tools
  (2) Enough data outside critical path: must fit the valley

Guilherme P. Koslovski
guilherme.koslovski@udesc.br

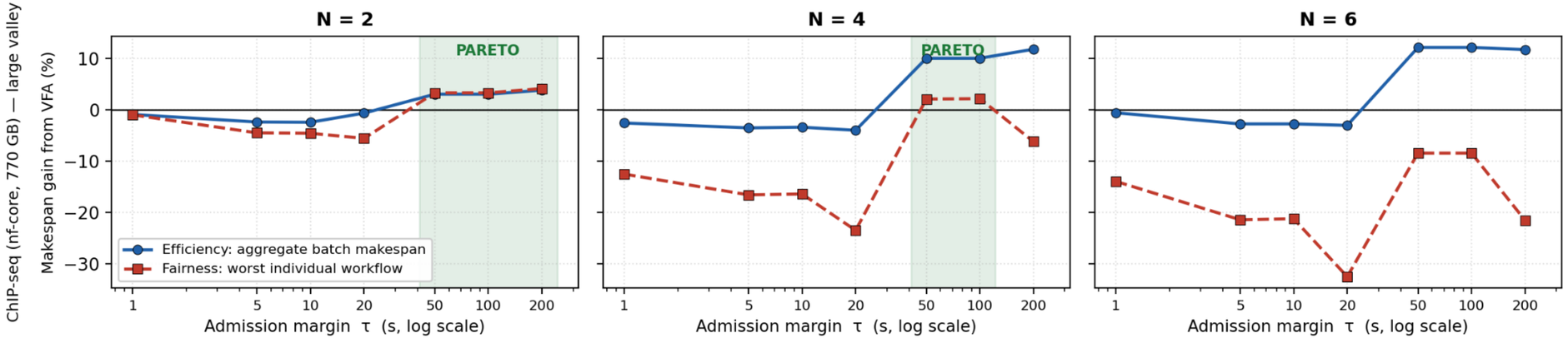
VFA gains with concurrency

- *N* copies of ChIP-seq trace sharing a 256-servers Dragonfly



Although some benefits can be observed…

… their extent is clearly limited.



slack of tasks outside the critical path

UDESC UNIVERSIDADE DO ESTADO DE SANTA CATARINA | JOINVILLE CENTRO DE CIÊNCIAS TECNOLÓGICAS

LabP2D

Guilherme P. Koslovski
guilherme.koslovski@udesc.br

Conclusion

- Network congestion is a fact!
- Workflow management systems have (full) knowledge on DAGs (critical paths)
- We can help making an admission control! But we have limits on it…
  - Network-aware scheduler and the Valley-Filling Admission (VFA) mechanism
  - Simulation setup
  - Conditions under which congestion avoidance reduces makespan
  - Testing the criterion on isolated workflows
  - Generalizing to a real production trace
  - VFA gains with concurrency
  - How stale can the signal be before congestion awareness backfires?
- Future:
  - Static to dynamic critical path
  - Remove distributed storage
  - Can we tolerate an increase in makespan to benefit all workflows?

How Far Can Network Congestion Awareness Improve Scientific Workflow Scheduling?

Luis Antonio Momm Duarte, Guilherme Piêgas Koslovski
Santa Catarina State University - UDESC - Graduate Program in Applied Computing

***Abstract*—Scientific workflows execute on shared High Performance Computing (HPC) infrastructures, where network congestion is a persistent phenomenon. Congestion arises from communication contention among tasks within the same workflow or across concurrent workflows, increasing workflow makespan. We show that congestion's impact can be mitigated by delaying task execution, but only when two conditions hold: *(i)* a shared communication bottleneck exists, and *(ii)* slack is available outside the workflow critical path. We propose the *Valley-Filling Admission (VFA)* mechanism, which holds off-critical-path transfers out of the congestion crest and releases them into the idle-bandwidth valley. Using network telemetry, VFA reduces workflow makespan by up to 9.2% for SoyKB and 11.2% for the 771 GB Chip-Seq workflow. To reach these conclusions, we run a simulation campaign in WRENCH/SimGrid across three network topologies (Tree, Fat-tree, and Dragonfly), driving eight Pegasus workflows and a real nf-core trace against a congestion-free baseline, sweeping the slack margin, contention factor, concurrency, and telemetry staleness.**

***Index Terms*—High performance computing, distributed computing, processor scheduling, communication system traffic control, admission control, network topology, makespan.**

I. Introduction

Network performance emerges as a critical bottleneck throughout multiple stages of a workflow's lifecycle. Beyond being essential for sustaining performance during execution, it is also a fundamental requirement for efficient *in situ* processing [1] and scalable provenance management [2]. In fact, scientific workflows rarely execute in isolation on production HPC systems. Instead, they share network and storage infrastructures with other applications. When multiple data-intensive workflows communicate simultaneously, they compete for the same network resources, creating contention along workflow communication paths. As a consequence, network congestion can significantly increase workflow makespan, especially for communication-intensive applications.

The central question addressed in this paper is therefore not whether network congestion is harmful, as this has already been extensively demonstrated in the literature [3]; rather, it is when it is worthwhile for a Workflow Management System (WMS) to act upon it. Unlike network or transport protocols, a WMS cannot directly control network resources or regulate competing applications. However, it has complete knowledge and control over the workflows under its management, including their execution order and the timing of their communication. This creates an opportunity for workflow-level congestion avoidance that complements congestion-control mechanisms implemented in the lower layers of the system stack.

During workflow scheduling, a WMS can intentionally postpone communications associated with tasks outside the workflow critical path. By shifting these transfers away from periods of peak contention, communication demand is redistributed toward intervals of lower network utilization. This strategy draws inspiration from ramp metering in traffic engineering, where traffic is intentionally delayed to alleviate downstream congestion, extending the same concept to workflow communications. Although considerably simpler and less intrusive than task migration or resource reallocation techniques [4], [5], this approach faces an important challenge: the WMS makes scheduling decisions using delayed observations of the network state. Consequently, postponing the wrong communication, or delaying it by an inappropriate amount, may inadvertently increase the workflow makespan by extending the critical path it intended to protect.

Our first finding is that the severity of network congestion alone is not a reliable predictor of the benefits of workflow-level congestion avoidance. Across 8 Pegasus workflows [6] and one real execution of the nf-core pipeline [7], the most congested workflow does not necessarily obtain the largest performance improvement. Instead, performance gains emerge only when two conditions are simultaneously satisfied. First, workflows must experience a shared communication bottleneck, providing congestion that can be relieved. Second, the workflow must contain byte-weighted slack outside the critical path, allowing communication to be postponed without delaying the application's completion. Only when both conditions coexist does congestion-aware scheduling become beneficial.

Based on these observations, we propose the Valley-Filling Admission (VFA) mechanism. In short, our contributions are threefold: *(i)* we demonstrate that congestion magnitude alone does not predict the effectiveness of workflow-level congestion avoidance; *(ii)* we identify a dual-condition criterion that determines *a priori* when the mechanism should be applied; and *(iii)* we show that the criterion holds across the evaluated network topologies, while the mechanism's tolerance to stale network information scales with the time scale of congestion.

The remainder of this paper is organized as follows. Section II motivates the problem, characterizes the impact of network congestion across topologies and workflows, and states our research hypotheses. Section III reviews related work. Section IV describes the network-aware scheduler and the VFA mechanism, and Section V details the simulation setup. Section VI establishes the two conditions under which congestion-aware scheduling reduces makespan. Section VII

SC26 Workshops, November 15-20, 2026, Chicago, Illinois, USA
979-8-3195-1221-5/26/$31.00 ©2026 IEEE

21st Workshop on Workflows in Support of Large-Scale Science (WORKS)
The International Conference for High Performance Computing, Networking, Storage & Analysis (SC 26)

Thanks!

How far can network congestion awareness improve scientific workflow scheduling?

Guilherme Piêgas Koslovski

guilherme.koslovski@udesc.br

UDESC UNIVERSIDADE DO ESTADO DE SANTA CATARINA | JOINVILLE CENTRO DE CIÊNCIAS TECNOLÓGICAS
LabP2D
fapesc Fundação de Amparo à Pesquisa e Inovação do Estado de Santa Catarina
CNPq