Research on HPC I/O in the Context of the PEPR NumPEx Project

Francieli Boito

Associate professor @ University of Bordeaux

Researcher @ LaBRI and Inria (TADaaM team)

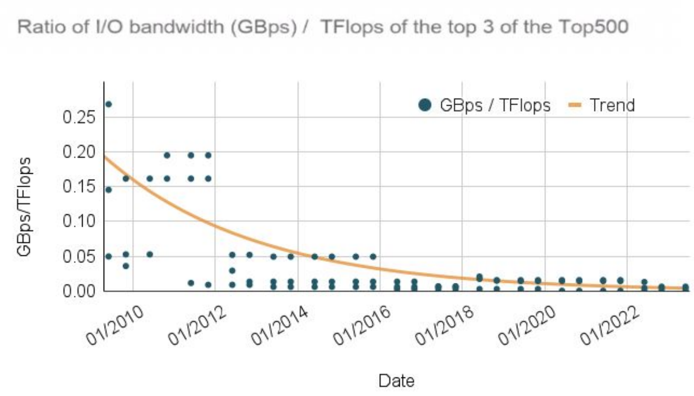
Per3S Workshop, May 2024

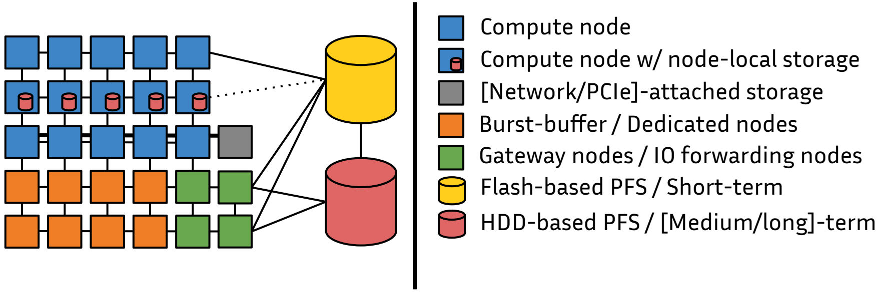
PC3 - Exa-DoST (Data-oriented Software and Tools for the Exascale)
Leaders: Gabriel Antoniu (Inria) and Julien Bigot (MdlS)

Data at exascale: a challenge in hardware

- Increasing **gap between compute and I/O** performance on large-scale systems
    - Ratio of I/O to computing power divided by ~10 over the last 10 years on the top 3 supercomputers
- … and data deluge!
    - At NERSC, **data volume x41** in 10 years

- New storage tiers and advanced architectures to try to mitigate this increasing bottleneck
    - More complex on-node memory layout
    - Emerging complex applications and workflows have to adapt





Trend in storage technologies available on extreme-scale systems

Our ambition

Fill the gaps in the existing software stack designed by previous projects (e.g. ECP)

Take into account French & European specificities

Ensure French & European needs are taken into account in roadmaps

Approach:

- **Research** on data-oriented tools for HPC
- Transverse, **re-usable tools**
- Usable **in production** at exascale

⇒ Exa-DoST will produce:

- **New approaches** to handle the data challenge at exascale
- Transverse **libraries & tools** that implement these approaches

Validated in illustrators at full scale

Fully application agnostic

Fully open-source

Work Packages in Exa-DoST

WP1: Exascale I/O and storage

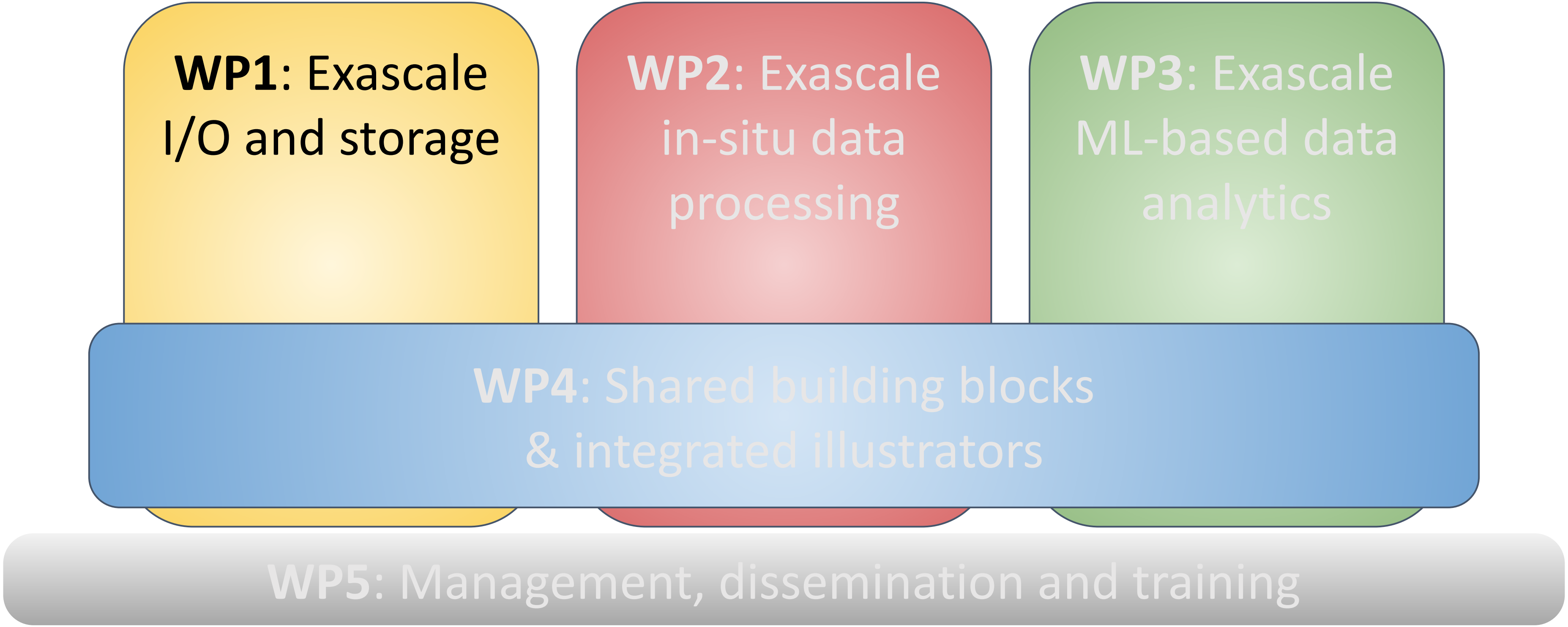
WP2: Exascale in-situ data processing

WP3: Exascale ML-based data analytics

WP4: Shared building blocks & integrated illustrators

WP5: Management, dissemination and training

Work Packages in Exa-DoST



WP Objectives

Optimize the I/O performance of applications and workflows, and leverage emerging storage technologies

- **Scale up modern I/O** and data storage methods and tools
- **Support the I/O and storage requirements** of complex simulation/analytics/AI workflows running on hybrid HPC (+cloud, +edge) systems
- Develop and integrate **new output formats** for checkpoint/restart and for scientific analysis

Participants

WP co-leaders: Francieli Boito (University of Bordeaux) and
François Tessier (Inria Rennes)

WP1: Exascale I/O and storage

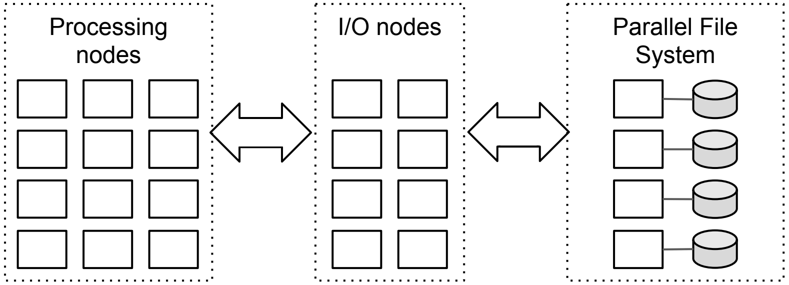
- **[T1.1]** What applications benefit from each solution?
  - In what conditions?
  - What are the problems (concurrent access, resource arbitration)?
- **[T1.2]** How can we detect the best strategy for an application?
- **[T1.4],[T1.5]** How to manage resources and tune the system for applications?
- **[T1.6]** How to represent applications' data? (Advanced data models)
- **[T1.3],[T1.7]** How to integrate these solutions in a software stack?

Scheduling Distributed I/O Resources in HPC Systems

Alexis Bandet, Francieli Boito, Guillaume Pallez

accepted for publication at Euro-Par 2024
available at https://inria.hal.science/hal-04394004

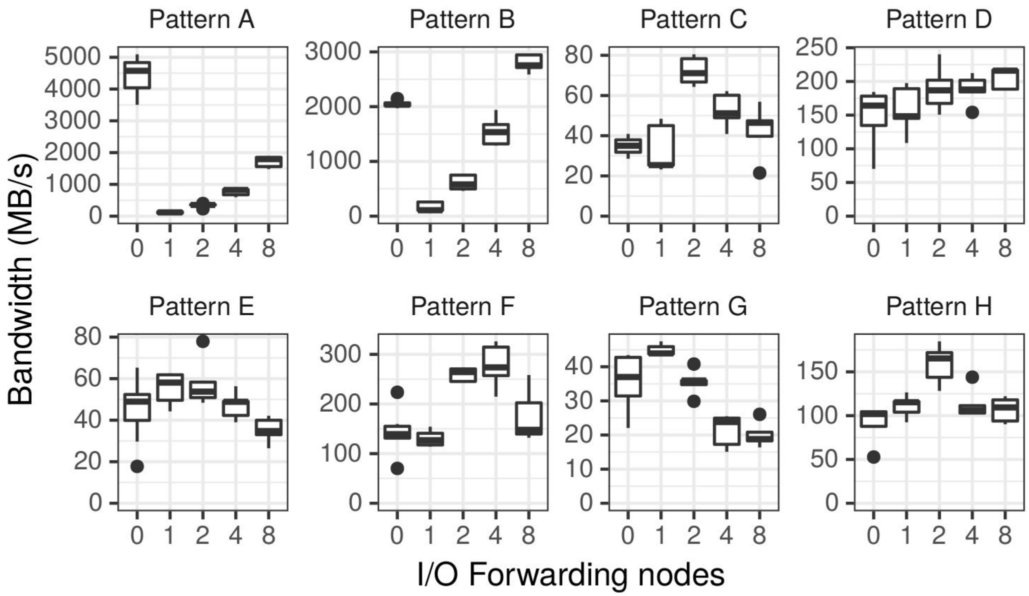
The problem of I/O in HPC



- HPC jobs are usually allocated exclusive compute resources
- The **I/O infrastructure is shared**
  - Variability: I/O performance depends on what others are doing
  - Contention: lower overall I/O performance
  - Lower utilisation: compute resources are usually “wasted” while waiting for I/O

Motivation

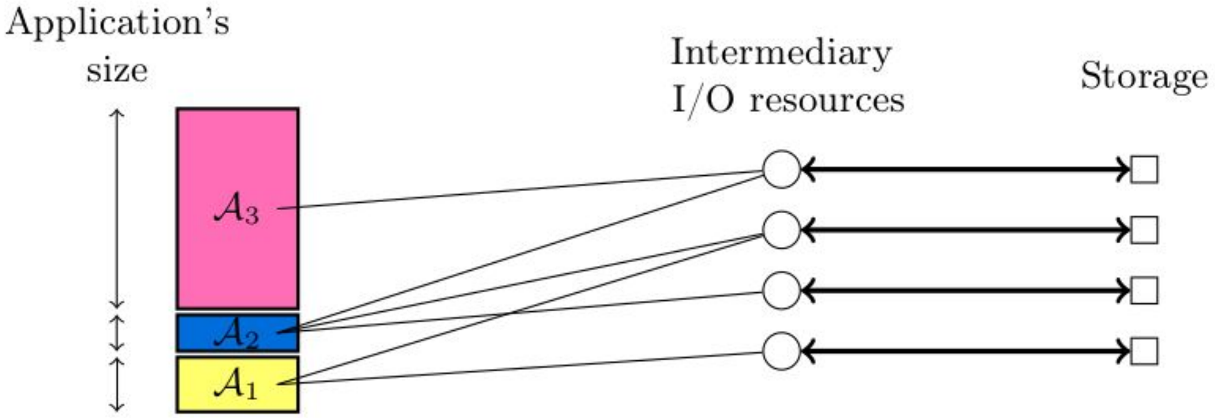
- The number of I/O nodes is usually static (similar for OSTs)
  - N compute nodes per I/O node, it depends on the placement
  - But it has a strong impact on performance



Graph from (Bez, Boito et al. PDSW 2020)

Scheduling of I/O resources in two steps

- **Allocation** = how many resources?
- **Placement** = which resources?



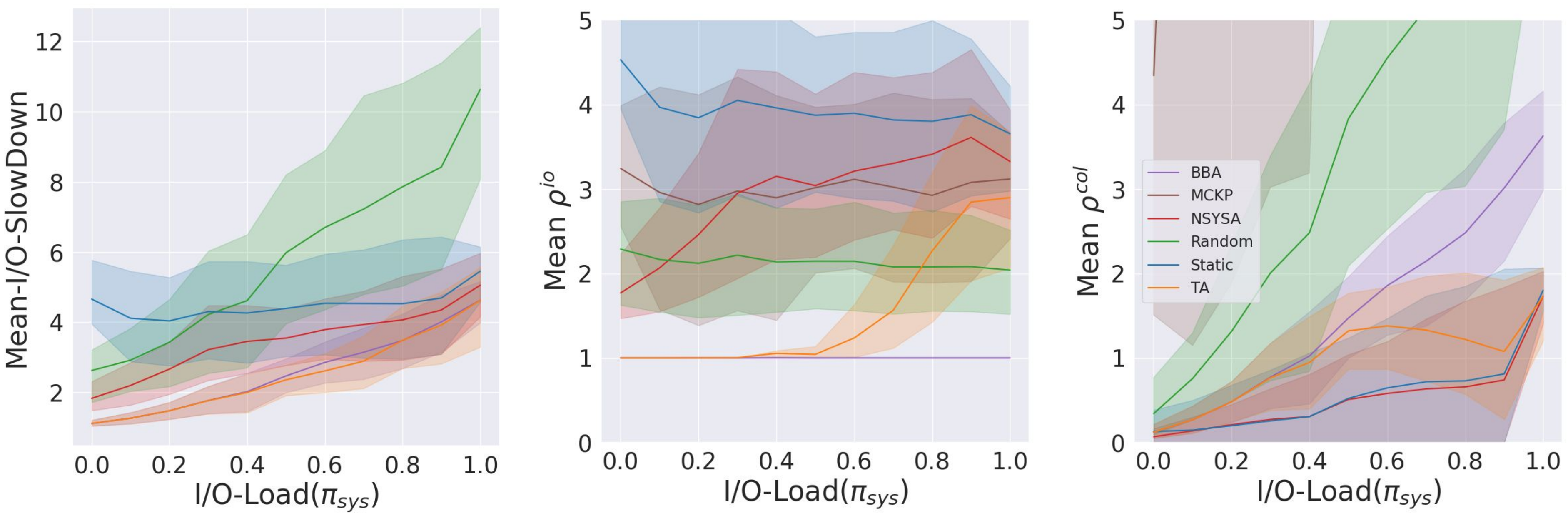
Algorithms

- Allocation:
  - Random and Static: baselines, +MCKP from previous work (Bez et al. IPDPS 2021)
  - NSYSA: each application receives the number that minimizes its I/O load
  - BBA: each application receives the number for its best I/O performance
  - TA: improve on NSYSA's solution by giving more resources to applications while respecting a maximum I/O load
- Placement:
  - Random: baseline
  - GNC: balance the number of applications per I/O resource
  - GC: balance the I/O load per I/O resource

Table 2: Heuristics and their input

		Allocation					Placement		
		Random	Static	BBA	NSYSA	TA	RandP	GNC	GC
Easy	Q_j		x			x			
Medium	V_{io}^j				x	x			x
	T_{cpu}^j				x	x			x
	n_{perf}			x					
Hard	b_j				x	x			x

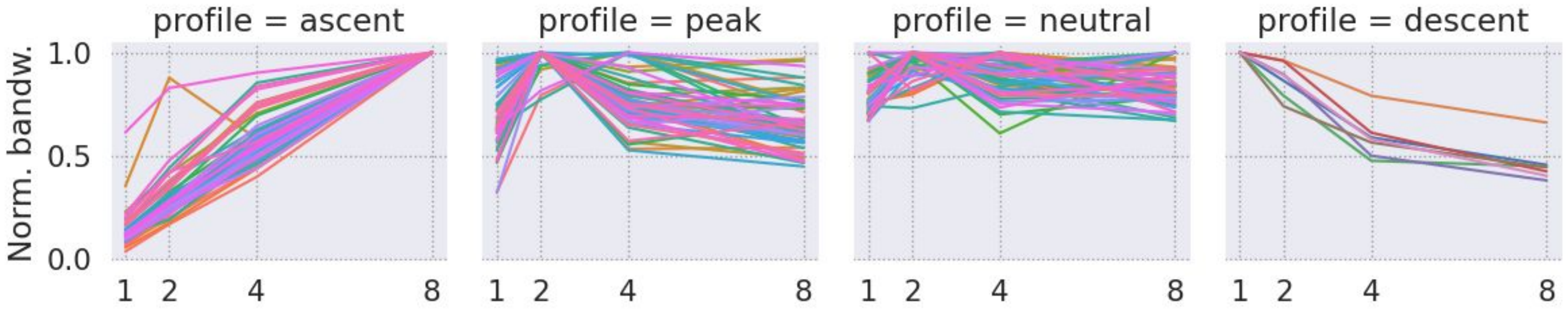
Results



Mean-I/O-SlowDown
Mean ρ^{io}
Mean ρ^{col}
I/O-Load(π_{sys})
I/O-Load(π_{sys})
I/O-Load(π_{sys})
BBA
MCKP
NSYSA
Random
Static
TA

Results with partial (imprecise) information

- BBA and TA are the best allocation policies
  - but as input they require the “profile” of the application
  - profile = performance as a function of number of I/O resources
- What if we just know the general shape?
  - Results get **< 1% worse**!



Ongoing work: classifying application behavior

(aka call for collaborations)

Perspectives

- First, to identify **classes of applications** regarding their behavior
  - example: the “I/O profile” from the work on scheduling of I/O resources
  - multi-dimensional classification
- Then, to identify what metrics allow for classification **at run time**
  - how fast can we do it?
  - ideally, very little overhead
- A challenge: temporal I/O behavior
  - publicly available traces are rare to non-existent

Capturing Periodic I/O Using Frequency Techniques

Ahmad Tarraf, Alexis Bandet, Francieli Boito,
Guillaume Pallez, Felix Wolf

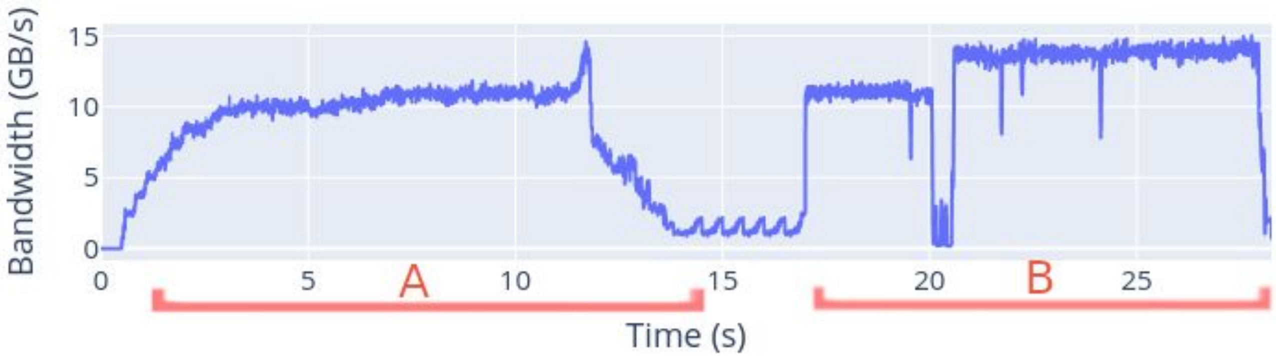
IPDPS 2024

available at
https://inria.hal.science/hal-04382142v1/

ADMIRE
malleable data solutions for HPC

Studying I/O periodicity

- A first step: **the time between the start of consecutive I/O phases**
  - and a measure of how much we trust that number (not all applications are periodic)
- it is actually much harder than it sounds…
  - an I/O phase = multiple I/O requests
  - where does it start and where does it end?
  - not all I/O is interesting

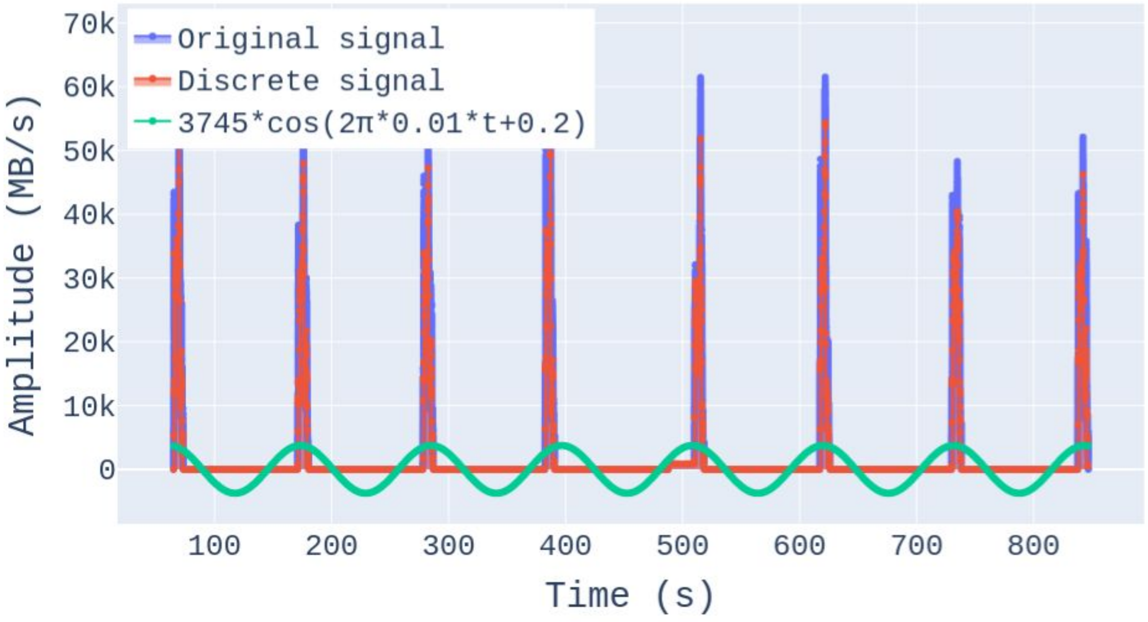


FTIO: frequency techniques for I/O

Collaboration between Inria Bordeaux and TU Darmstadt

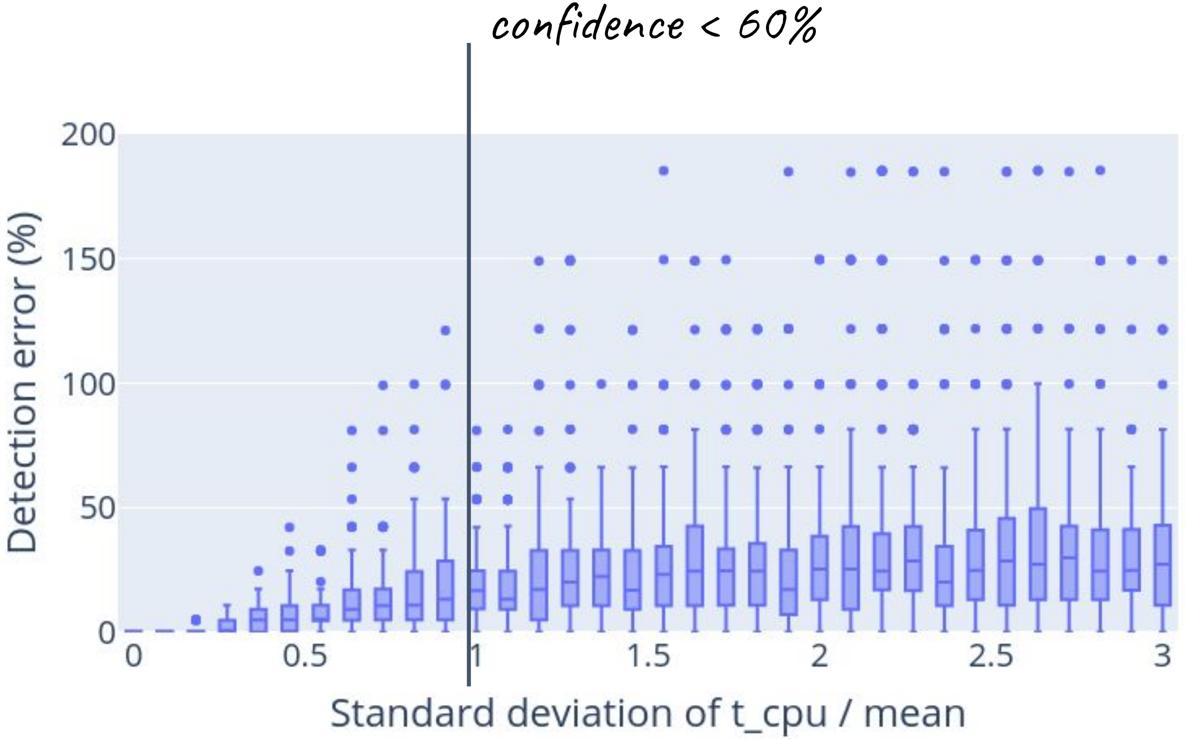
- Treat I/O bandwidth over time as a signal
  - Apply discrete Fourier transform (DFT) + z-score to find the dominant frequency(ies)
- It can be done online, working on a time window of recent activity
- Measures of periodicity: the standard deviation of the amount of transferred data (and time spent on I/O) per DFT-identified period

FTIO: frequency techniques for I/O

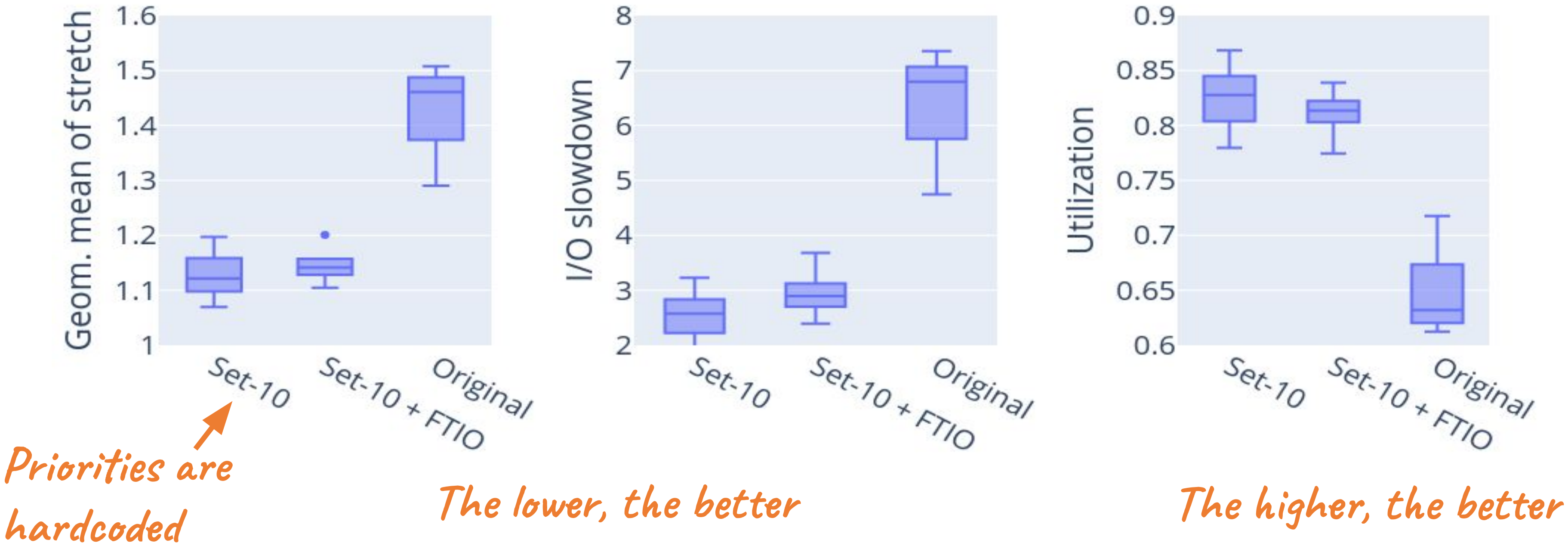


IOR on 9216 ranks

FTIO: frequency techniques for I/O



the more to the right = the "less periodic"



- **Stretch**: for each application, how much it was slowed-down by others compared to running by itself (minimum of 1, meaning no slow down). We take the geometric mean of the 16 applications.
- **IO-Slowdown**: for each application, how much slower its I/O was compared to running by itself (minimum of 1, meaning no slow down). We take the geometric mean of the 16 applications.
- **Utilization**: how much of the system time was spent on compute (NOT doing I/O or waiting for I/O), so between 0 and 1 (1 means no I/O at all).

We're hiring!

Research on HPC I/O in the Context of the PEPR NumPEx Project

Francieli Boito

francieli.zanon-boito@u-bordeaux.fr