



September 2, 2025 – Edinburgh, Scotland

I/O Simulation: From resources to complex workloads

Fred Suter

REX-IO 2025

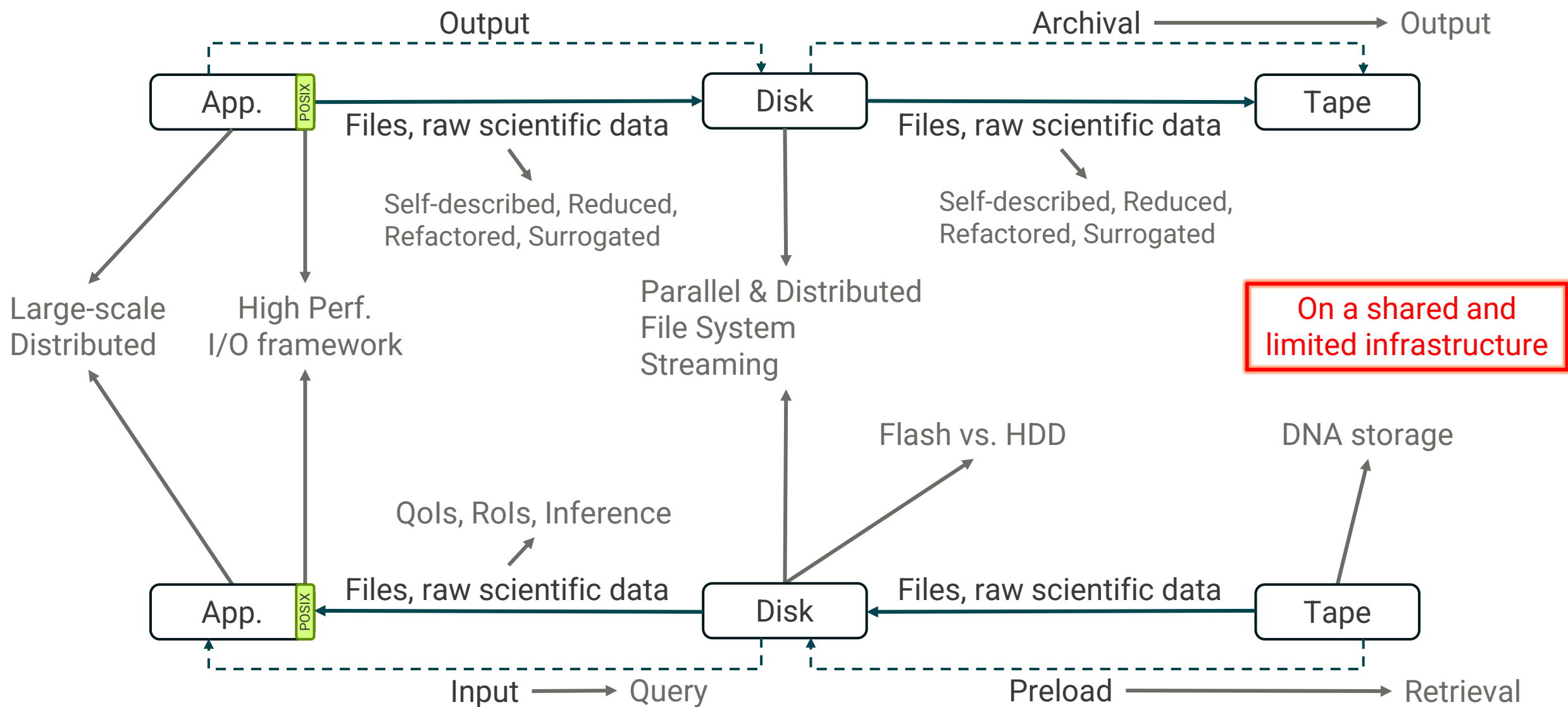


U.S. DEPARTMENT
of **ENERGY**

ORNL IS MANAGED BY UT-BATTELLE LLC
FOR THE US DEPARTMENT OF ENERGY



Evolution of I/O and Data Management



Overarching Questions in That Context

Can we

- Build better I/O and data management software to accelerate science?
- Optimize resource utilization and handle dynamic changes?
- Reduce manual interventions in complex data management tasks?

Yes, if we can take the right decisions and select the right levers at the right time

What is needed for that?

- Performance models that are **fast, scalable, dynamic / interactive**
- And can **capture the entire HW/SW stack and have predictive value**

In other words, we need a comprehensive Digital Twin

Why Simulate I/Os and Storage?

An important performance driver to understand

- Independent of scale and type of the computing infrastructure
- As much important as computing and networking

Specifics and concerns of storage subsystems may vary

- **Data Centers** ⇒ Hierarchical (mass) storage subsystems ⇒ Different types of media involved
- **Supercomputers** ⇒ Large-scale dedicated storage network ⇒ High-speed network interconnect
- **Clusters** ⇒ Specific and tuned file system ⇒ Reliable, scalable, and simple
- **Grids and Clouds** ⇒ Services offered by multiple data centers ⇒ Hidden underlying infrastructure

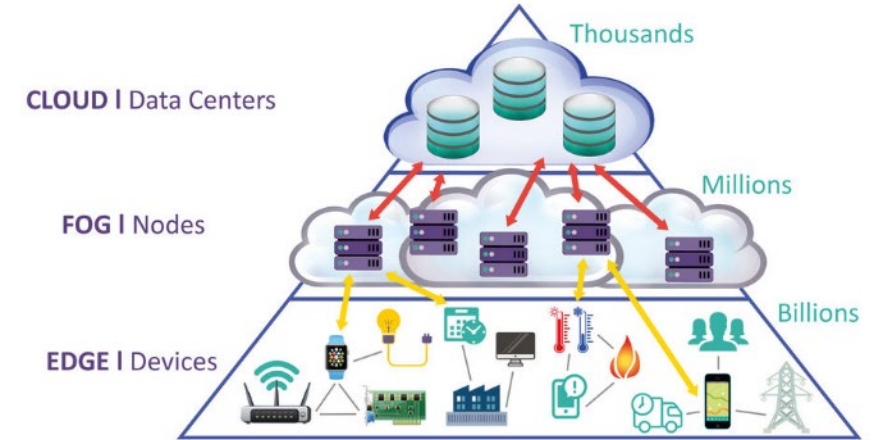
Versatility is key!

A Brief History of

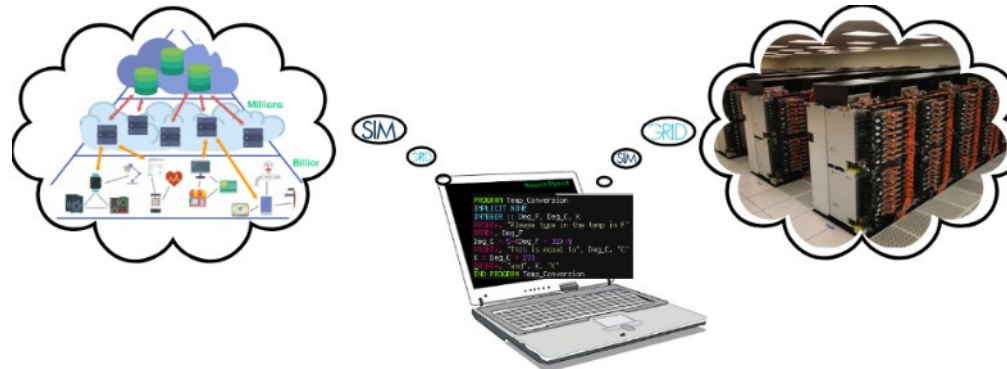


Distributed Systems as Scientific Objects to Study

Clusters, supercomputers, peer-to-peer systems, grids, clouds, . . .



How to study these systems and their applications on my laptop?



The SimGrid Toolkit

A scientific instrument on your laptop



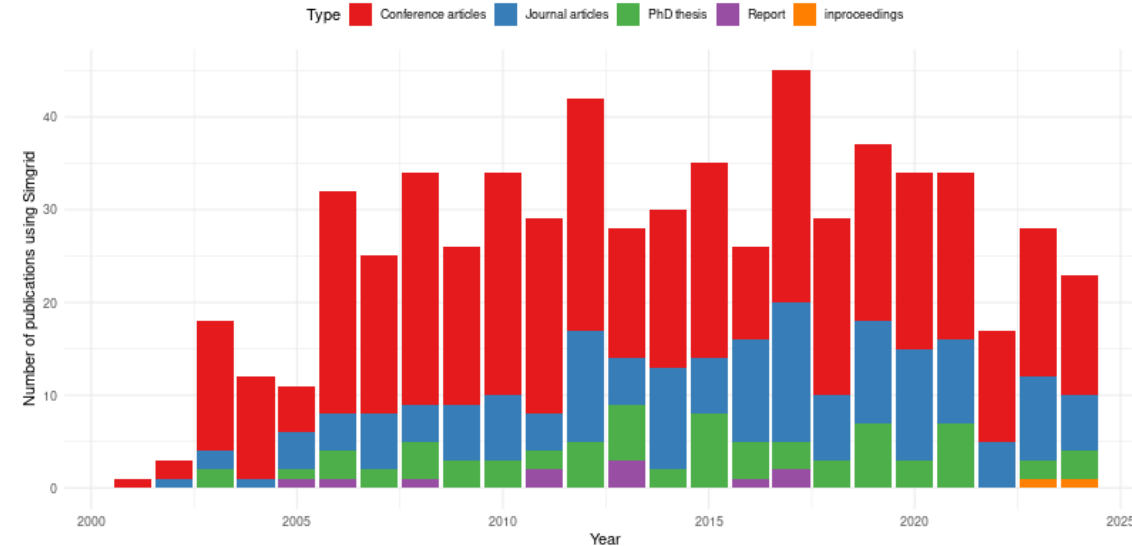
<https://simgrid.org>

Open Project since 1998

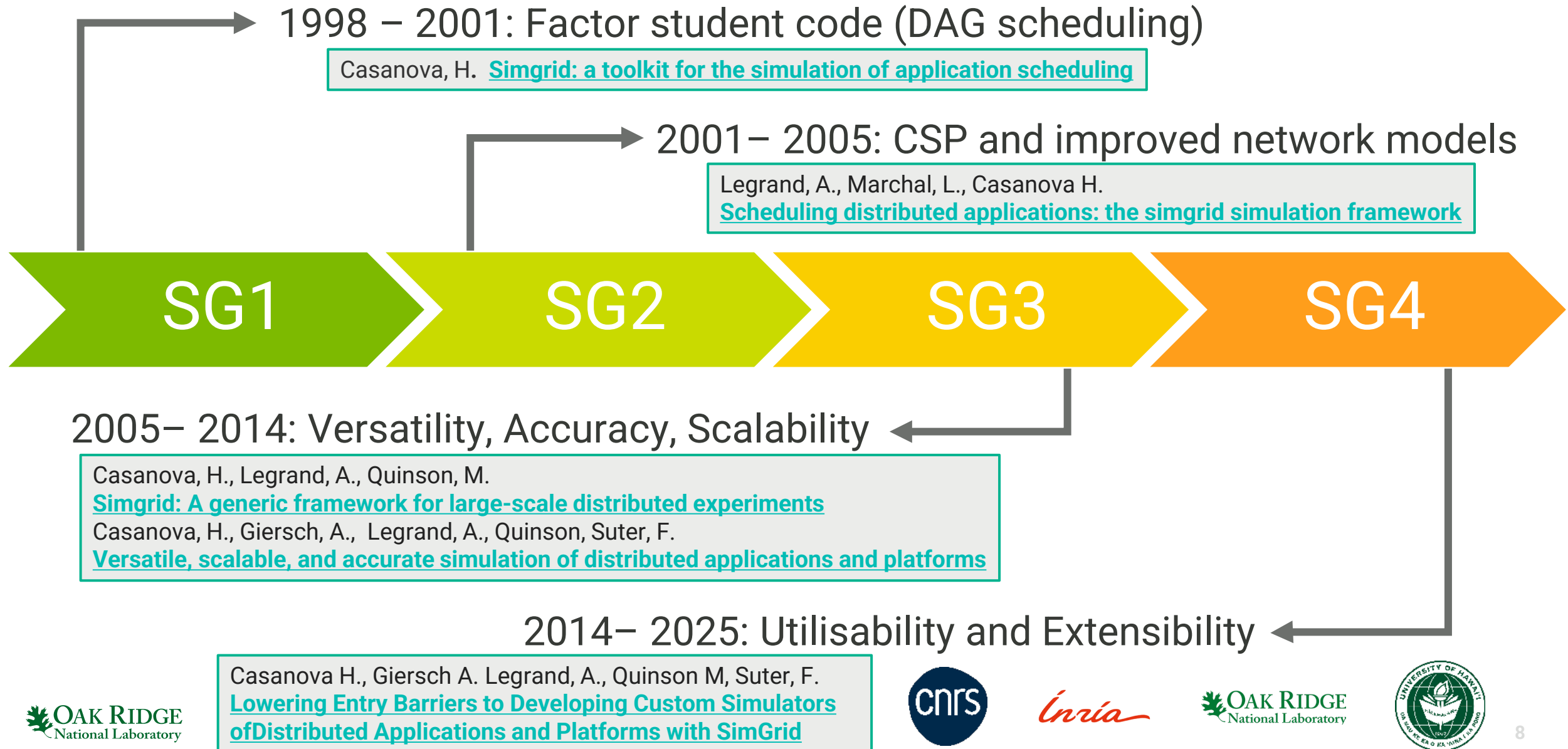
- 2,200+ citations and 665+ usages

Key strengths

- **Usability:** Fast, Reliable, User-oriented APIs
- **Validated performance models:** Open Science \Rightarrow Predictive Power
- **Versatility:** Grid, P2P, HPC, Cloud, Fog, ...



SimGrid History



SimGrid in a Nutshell



SimGrid in a Nutshell

Discrete Event Simulator (sequential, but fast)

Base Abstractions

Actors

Program anything you want/need

Activities

Computation, communication, I/O

Resources

CPUs, Links, Disks, ...

Mailboxes / MessageQueues

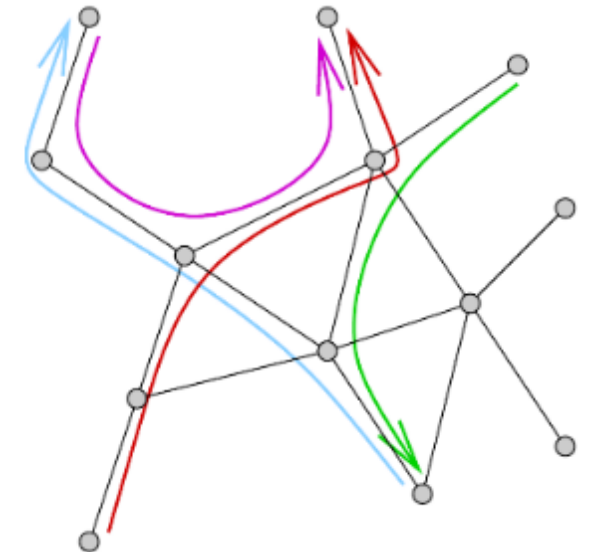
Rendez-vous points between actors

Flow-level models

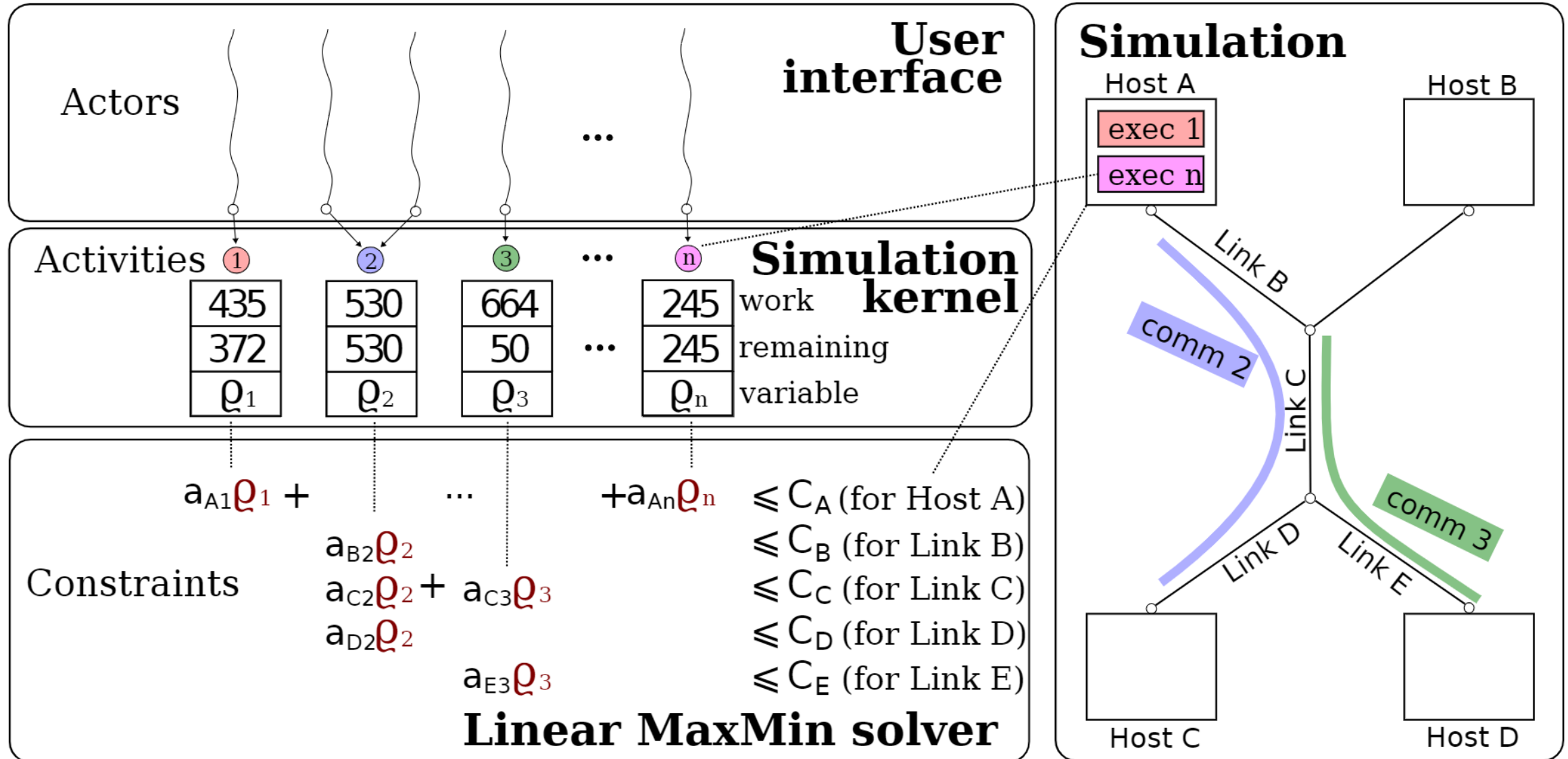
- Boils down to solve a **linear max min** problem
- Good tradeoff between **speed and accuracy**
- Multiple optimization techniques and specializations

Simulation kernel main loop

1. Compute **share** of resource allocated to every activity
2. Compute the **earliest finishing** activity, **advance** simulated time
3. Remove finished activity
4. Loop back to 2

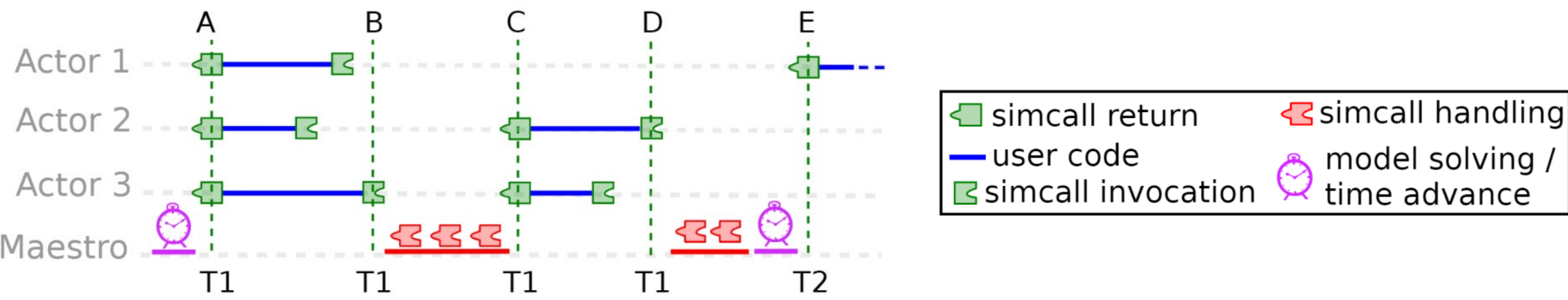


Overview of a SimGrid simulation

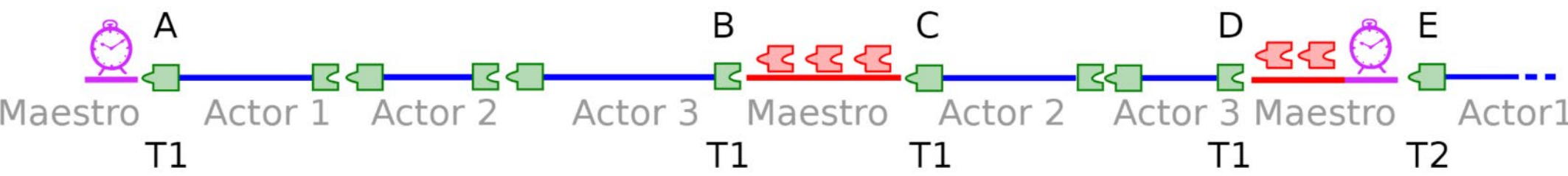


Execution Timelines

In Simulated Time



In Wallclock Time



SMPI

(**Sim/em**)ulation of (unmodified) MPI applications

- Support of (the essential of) MPI-3.1
 - Including a partial support of **MPI-IO**
- **Collective operations:** Borrowed selection logic of popular runtimes















































```
$ smpicc source.c -o application # The code is now compiled
$ smpirun -platform cluster.xml -hostfile hostfile.txt ./application # It starts
[...] # Some debug information about your data provenance
Got 42 from rank 0
```

Integration testing

- Compile and run **100+ proxy-apps** every night
- And full application (BigDFT) and runtime(StarPU)
- C, C++, F77, F90, kokkos, and some OpenMP

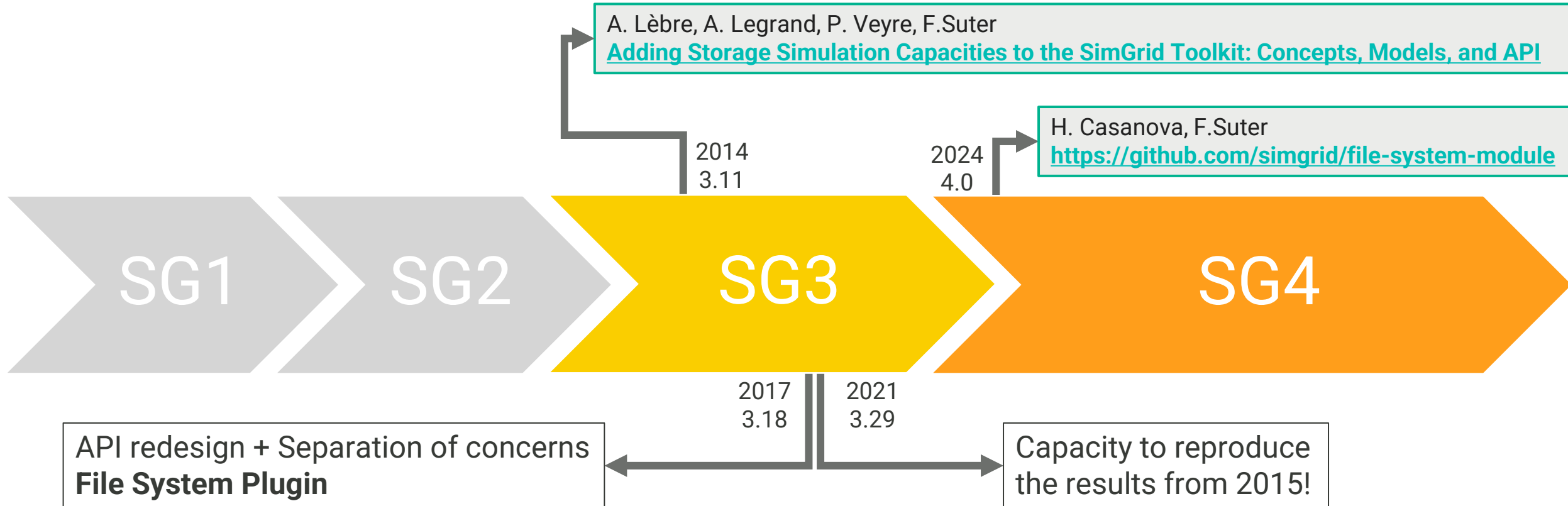
<https://framagit.org/simgrid/SMPI-proxy-apps/>

ECP Proxy Application

Benchmark	Lines	Lang	Script	Status	Build Patch	Code Patch
AMG	4,658	C		 	*	*
CLAMR	109,477	C++		 		
Chatterbug	1,050	C++		 	*	*
Comb	1,965	C++		 		
CoMD	4,658	C		 	*	*
CoSP2	2,199	C		 	*	
CloverLeaf	37,477	C, F90		 	*	*
CloverLeaf3D	11,200	C, F90		 	*	*
EBMS	841	C++, F90		 	*	*
Ember	1,300	C		 	*	*
ExaMiniMD	6,184	C++		 		only if sampling is needed : 
CabanaMD	33,000	C++		 		
HPCOG	1,548	C++		 	*	*

I/Os in SimGrid – Resource level

I/Os in SimGrid



Back in 2015 – Ground Truth Data Acquisition

Testbed

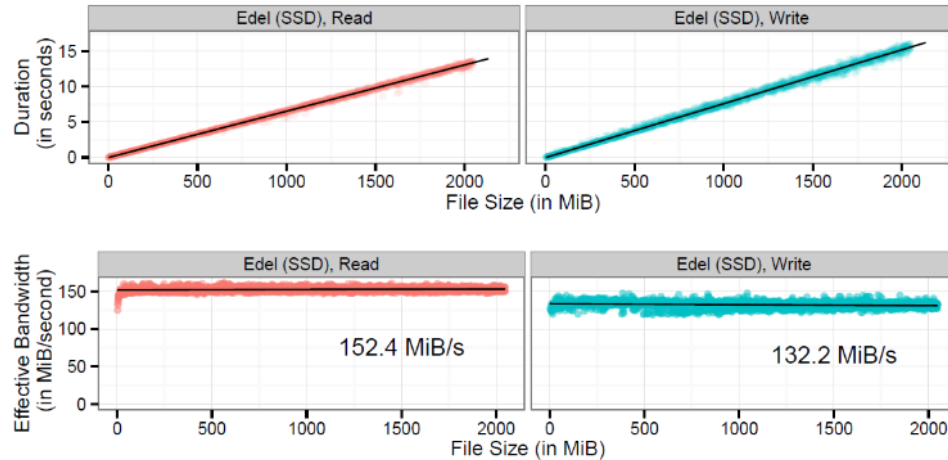
- Grid'5000 experimental platform (<https://www.grid5000.fr>)
- Three types of disk: SATA-II, SAS, and SATA/SSD

Methodology

- **Randomized FIO** benchmarks
- **Synchronous, non-buffered** I/O operations
 - **Independent:** From **32kiB** to **2GiB** with a **fixed block size of 32KiB**
 - **Concurrent: 1 to 15** operations
 - For 10, 50, 100, 500, 1024, and 2048 MiB files

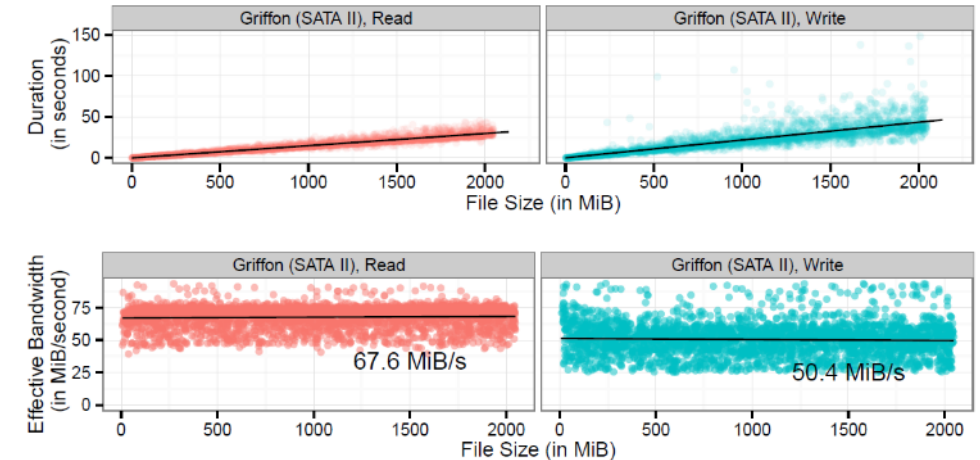
Deriving Models from Experimental Data

SSD



- Linear w.r.t. bandwidth
- No latency

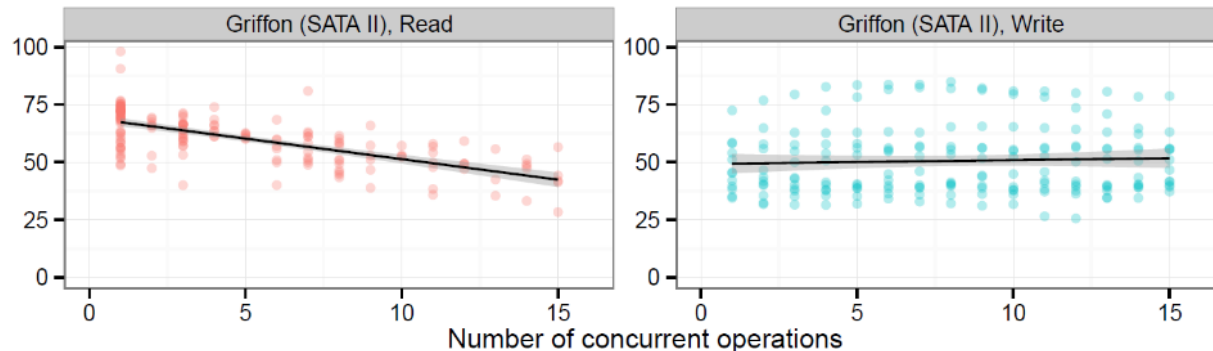
HDD



- **Heteroscedastic** behavior
 - Variability **proportional** to size

Concurrent accesses

- **Modify resource capacity** as concurrency increases
- **Reevaluate** each time a transfer **begins** or **ends**



File System Plugin, I/O Streams, and JBOD

File System Plugin (ca. 2017) – Better separation of concerns

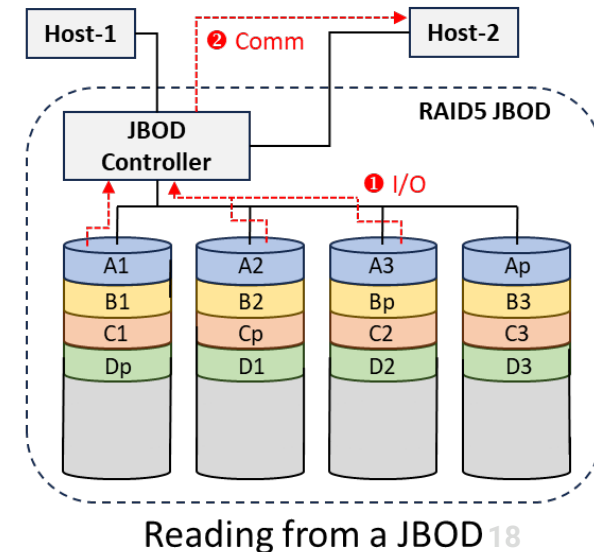
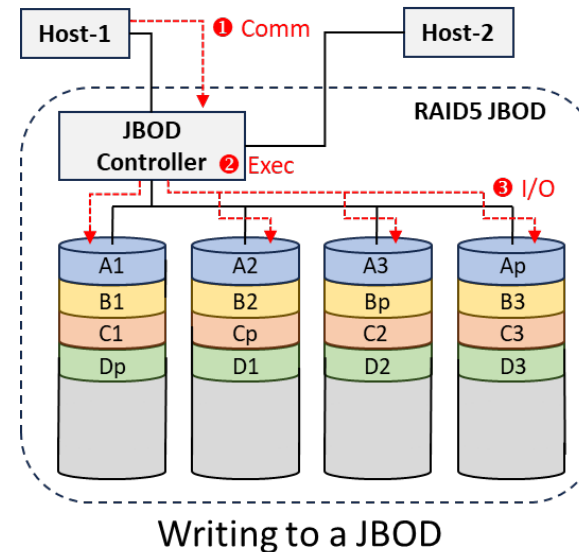
- **SimGrid models:** raw byte streams on disk (read/write bandwidth)
- **Plugin:** concept of **file** and standard operations, Posix-like operations on **file descriptors**

I/O streams – Speed up simulations

- Model **[read] – transfer – [write]** from a host to another as a **fluid activity**
 - Disk to disk, disk to memory, or memory to disk
- **Fluid?** Works as if doing store-and-forward at a very fine grain
 - I/O and Comm activities progress together at the **limiting bandwidth speed**

JBOD Plugin – Modeling RAID systems

- New concept of **compound activity**
- Combines several activities and wait for the completion of the last one

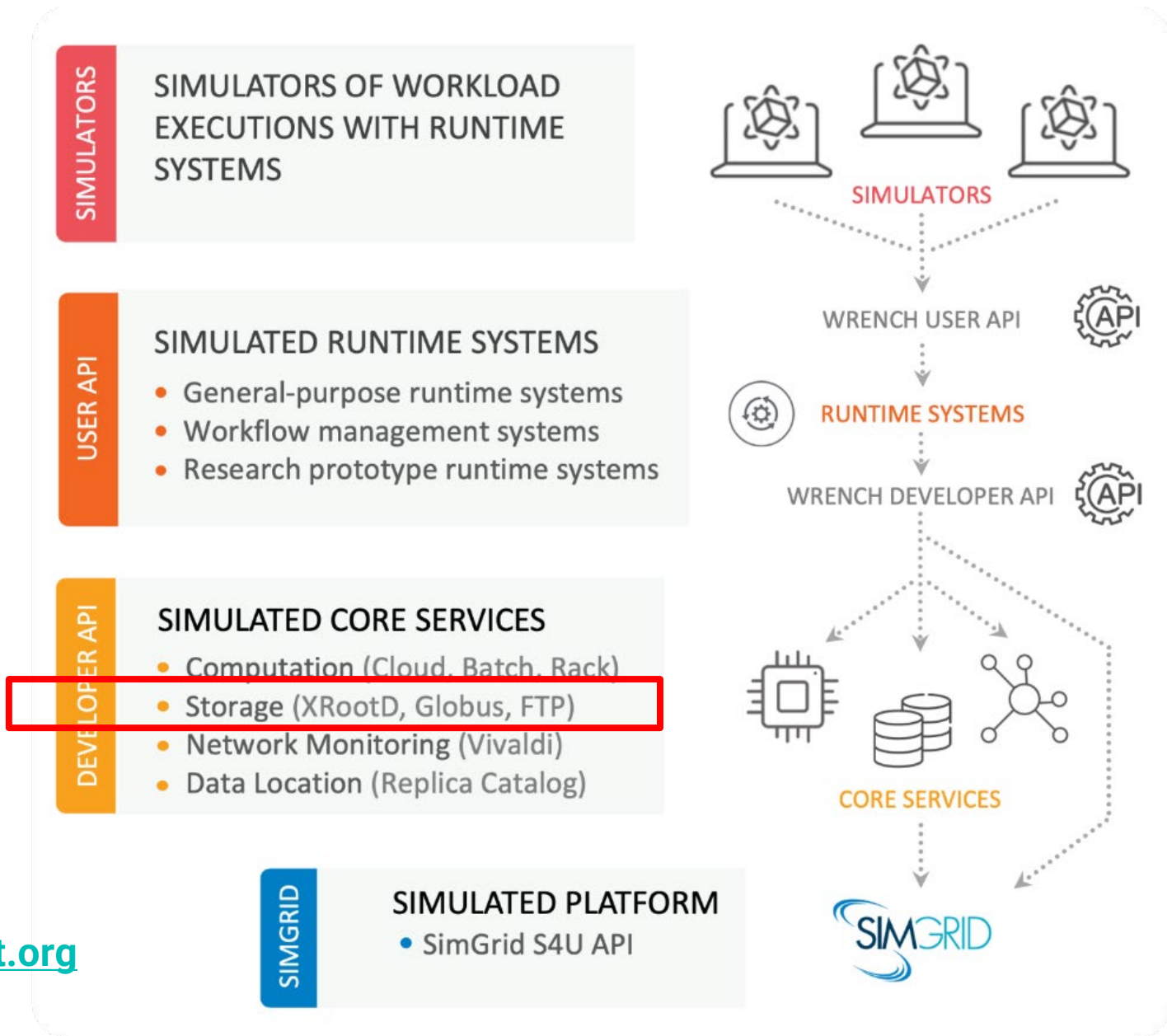


I/Os in SimGrid – File system level

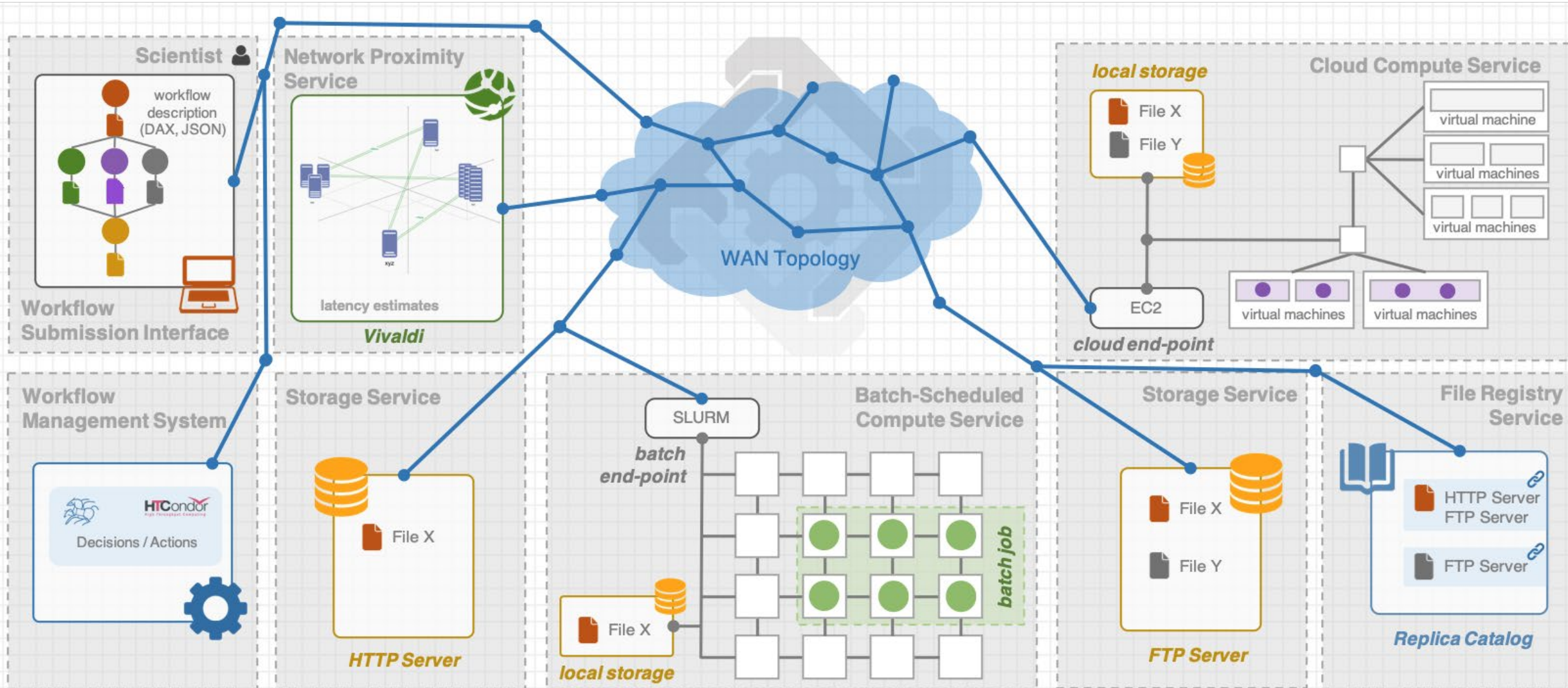


Wrench

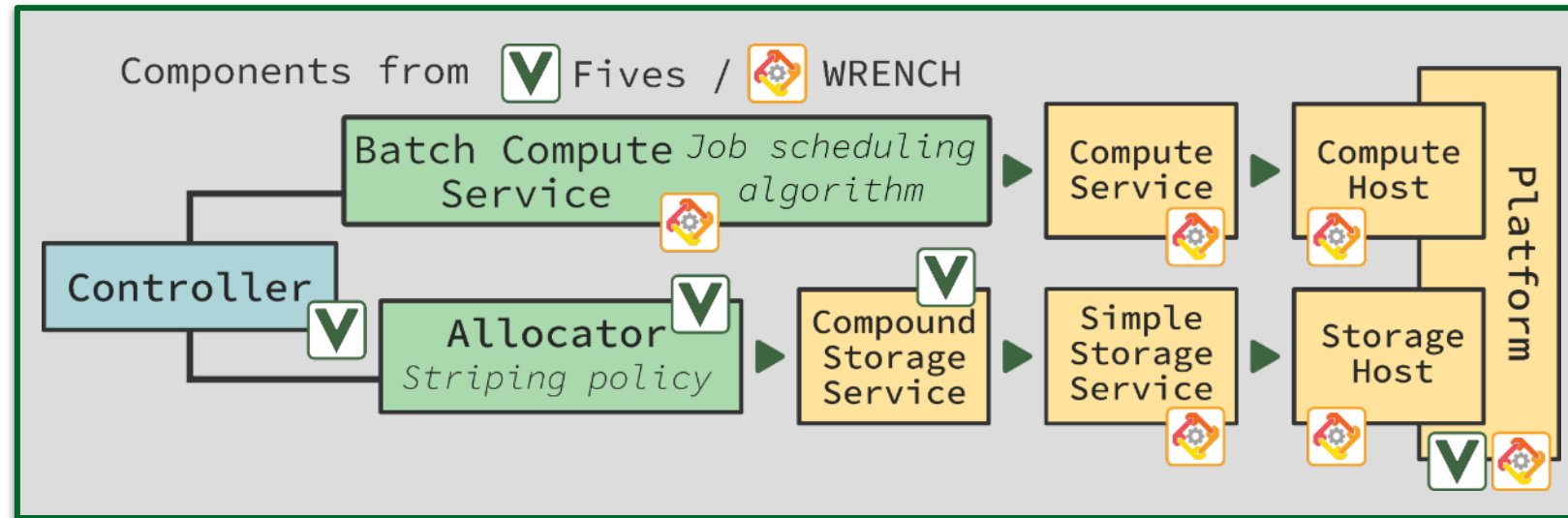
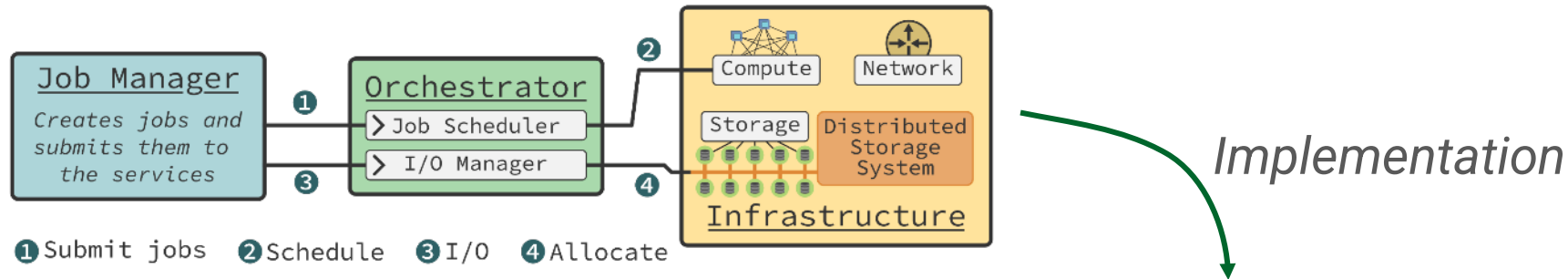
- Project initiated in 2016
 - With R. Ferreira da Silva (ORNL) and H. Casonova (UH Manoa)
- Objectives
 - A virtual lab to study WMS
 - Improve SimGrid expresiveness
- DSL-like approach:
 - High level concepts
 - Composable modules
 - Different levels of APIs



Wrench Overview



FIVES: a Simulator of High-Performance Storage Systems

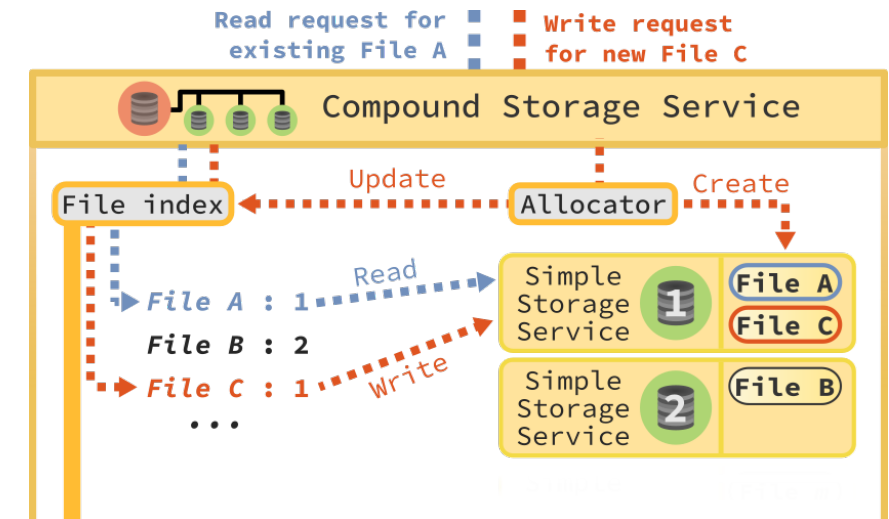


FIVES' Compound Storage Service:

- **Generic model** of a distributed file system
- Supports **splitting** a file in parts and **distributing** it on multiple *Storage Services*
- Integrated as a service into WRENCH

Internally

- **File Index** → Free MDS
- **Allocator** → User provided allocation policy
 - FIVES comes with the **Lustre round-robin/weighted policy**



File System Module

Motivations

- Factor development of similar capabilities between SimGrid and WRENCH
- Replace the old and simplistic file system plugin

Objective

- Implement a **simulated file system** on top of SimGrid
- Support the notion of **partitions** that store **directories** that store **files**
- **Standard operations**: create, move, unlink files, unlink directories, check for existence)
- Support the notion of a **file descriptor** with **POSIX-like operations** (open, seek, read, write, close).

Approach

- Developed as a **standalone library** to be used in **any SimGrid-based simulator**

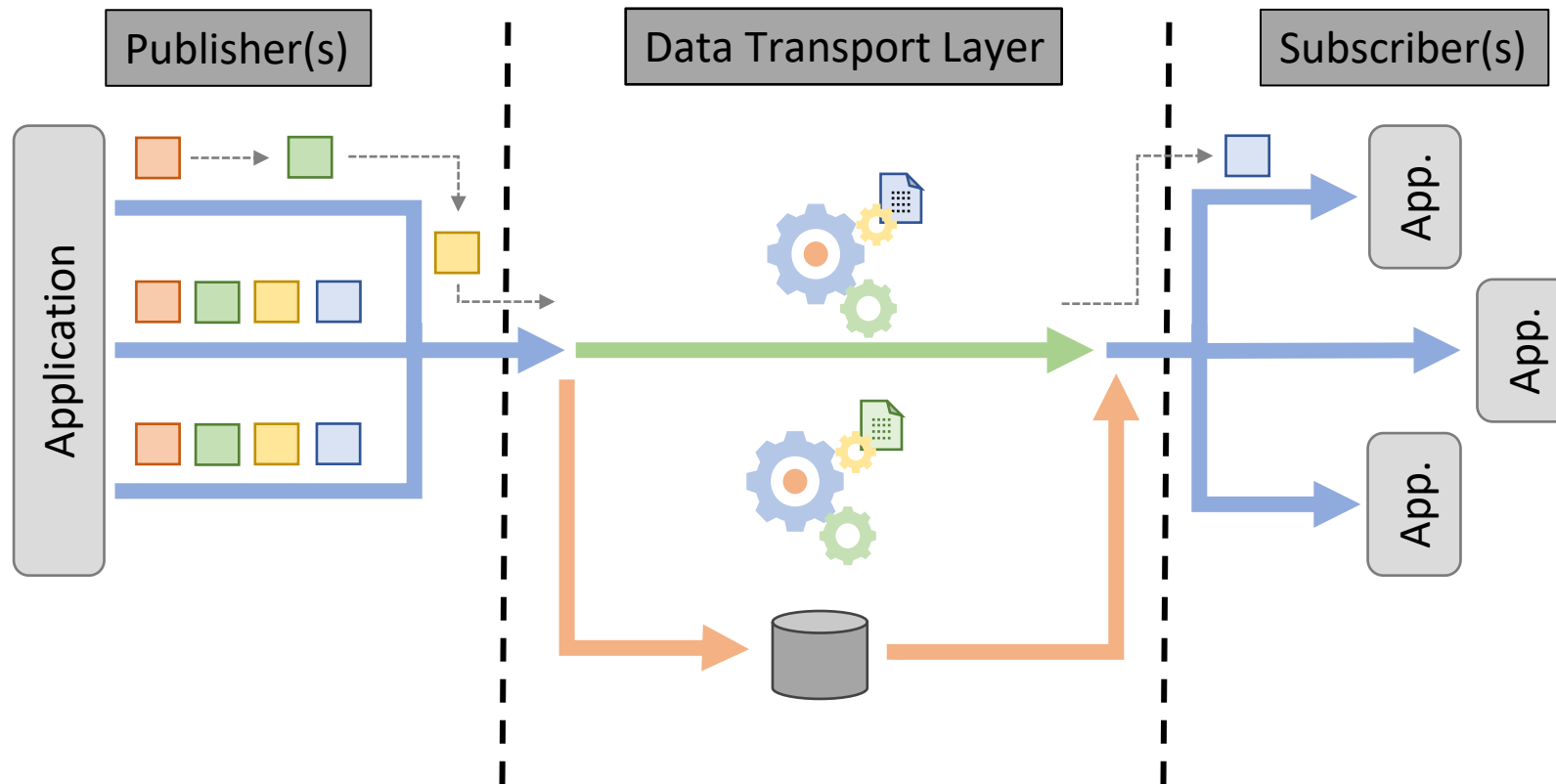
Future work

- Integrate the development of a **Lustre** file system made in [FIVES](https://github.com/simgrid/fives)

DTLMod – Versatile Simulated Data Transport Layer

More details on DTLMod
Thursday Sep 4th
Session 7 on Performance
Modelling and Optimisation
in Pentland room

Overview and Terminology



Building a Simulated In Situ Processing Workflow

```
int main(int argc, char** argv) {  
    sg4::Engine e(&argc, argv);  
    e.load_platform("./platform_description.so");  
  
    auto analysis_host      = e.get_host_by_name("node-0");  
    auto simulation_hosts =  
        e.get_hosts_from_MPI_hostfile("./hostfile");  
  
    // Create the data transport layer  
    DTL::create("./DTL_config_file.json");  
  
    // Start a simulated MPI code instance run by  
    // multiple actors  
    SMPI_app_instance_start("simulation",  
        simulation_main, simulation_hosts, argc, argv);  
  
    // Create a single in situ analysis actor  
    e.add_actor("analysis", analysis_host, analysis_main);  
  
    // Run the simulation  
    e.run();  
    return 0;  
}  
  
// Cleanup  
SMPI_SHARED_FREE(data);  
MPI_Finalize();
```

Connect to the DTL

'File' engine

V''

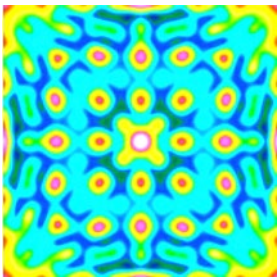
Variable 'V'

Operations per element
local_size() * 1e3);

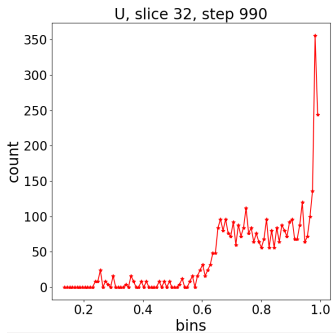
Communication

In Situ Simulation/Analysis Workflow

3D domain decomposition
MPI code

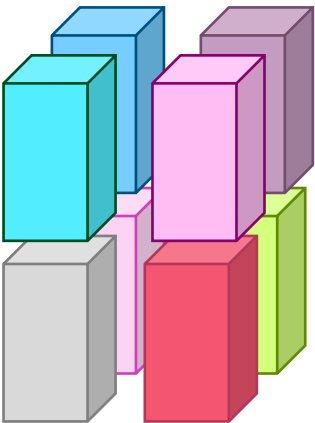


Gray-Scott
(chemical reaction)

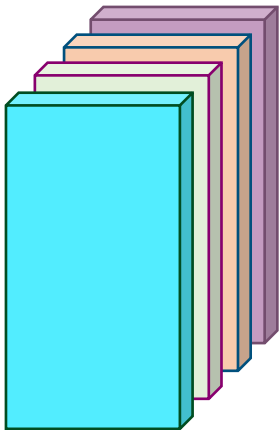


Embarrassingly parallel
MPI code

PDF Calculation



From 3D-variables
to 2D slices



Setup

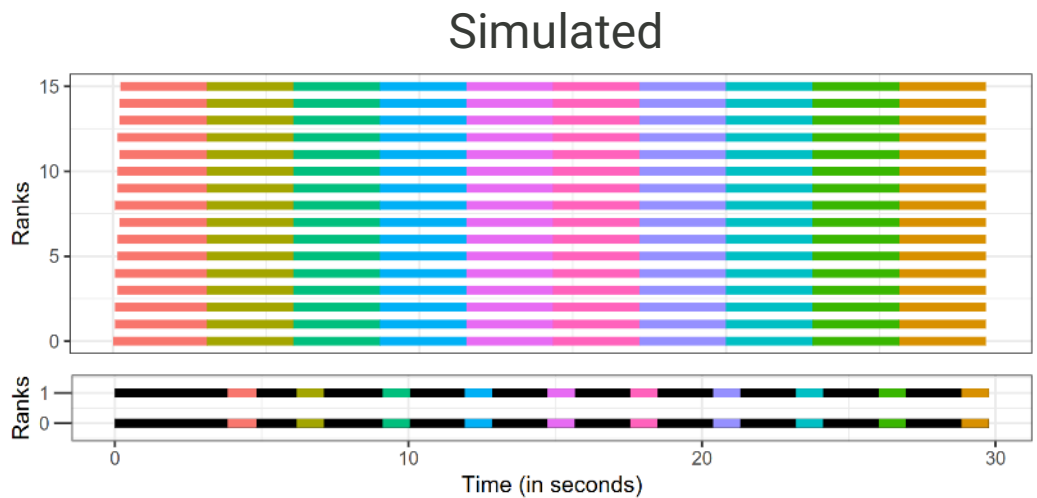
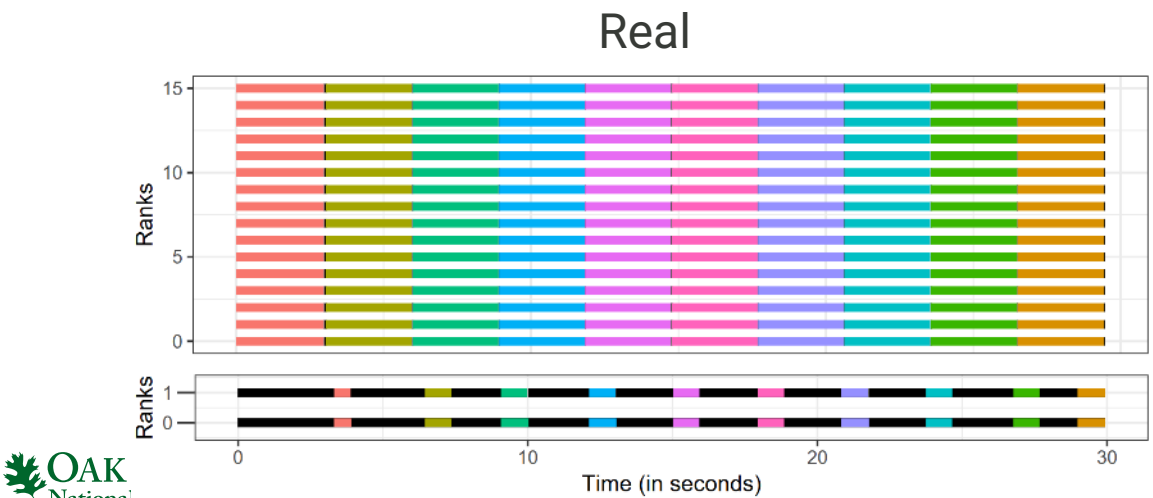
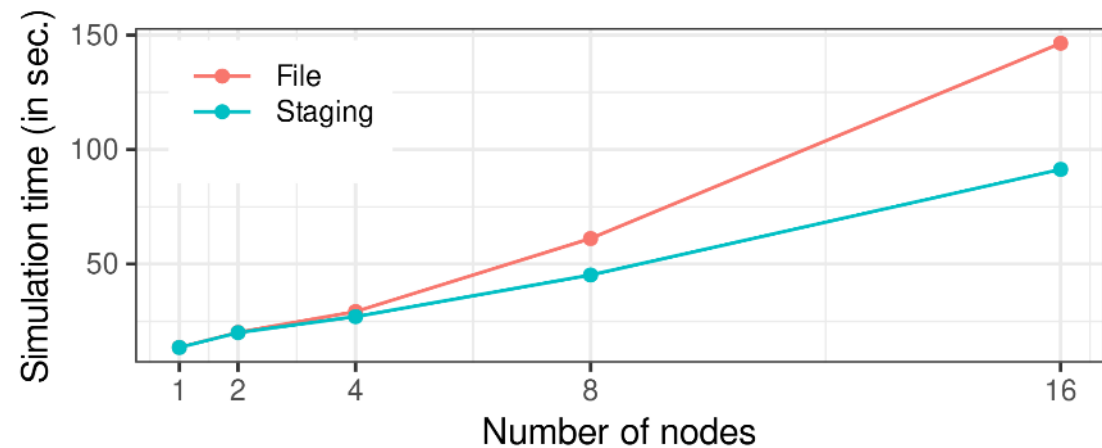
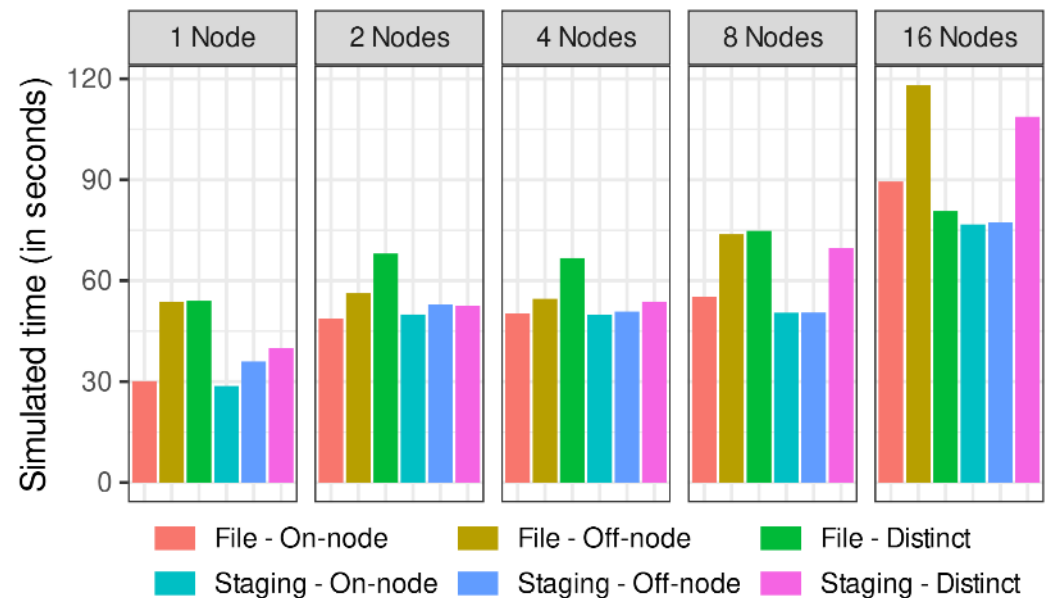
- Two 256^3 variables
- 4/1 Pub-Sub ratio
- 100 timesteps
- **Data transfer every 10 timesteps**
- 256 GB / transaction
- 2.5 TB total

Objective

- Can DTLMod simulate a **real in situ simulation-analysis workflow**

#Compute nodes	1	2	4	8	16
#Publishers	64	128	256	512	1,024
Pub. distribution	4^3	8×4^2	$8^2 \times 4$	8^3	16×8^2
GB / Pub. / Trans.	4	2	1	0.5	0.25
#Subscribers	4	8	16	32	64

Versatility, Scalability, and Accuracy

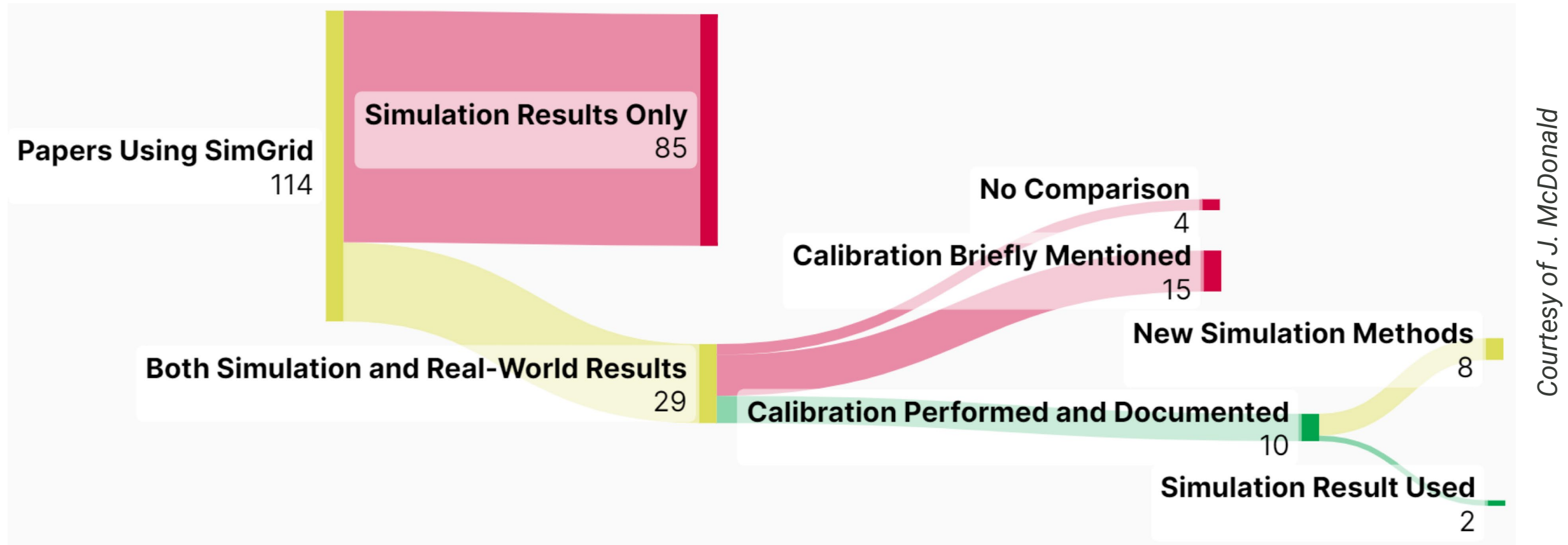


I/O simulation – Calibration and levels of detail



Simulation Calibration: State-of-the-Art

- Based on a study of 114 SimGrid-based papers published in 2017-2022



Courtesy of J. McDonald

McDonald, J., Horzela, M., Suter, F., and Casanova, H.
Automated Calibration of Parallel and Distributed Computing Simulators: A Case Study.
Proceedings of the 25th IEEE International Workshop On Parallel and Distributed Scientific and Engineering Computing (PDSEC)
DOI: [10.1109/IPDPSW63119.2024.00173](https://doi.org/10.1109/IPDPSW63119.2024.00173)

Approach: Automated Calibration Procedure

Ground-truth execution data from target system

Simulator of target system

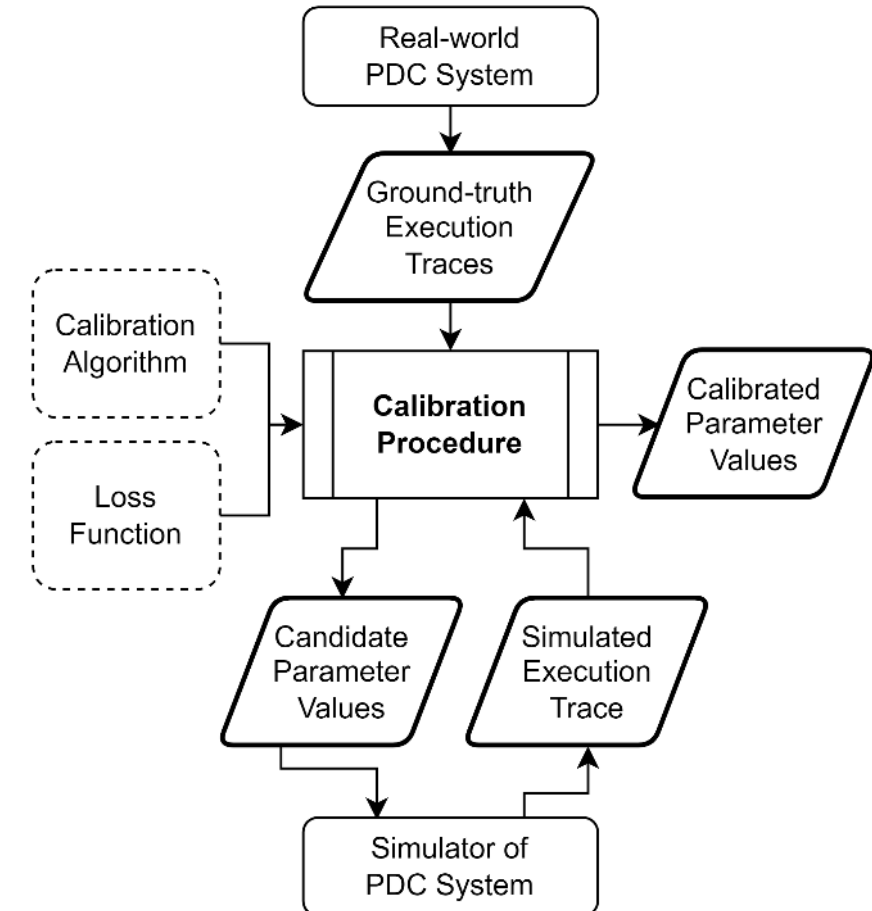
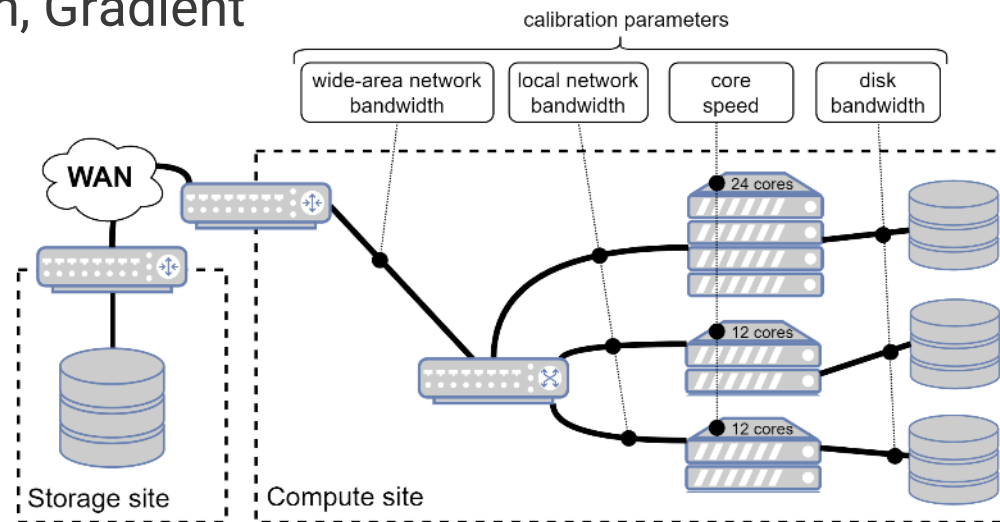
Parameter ranges to calibrate

- I/O bandwidth, RAM page cache, WAN bandwidth, ...

Loss function

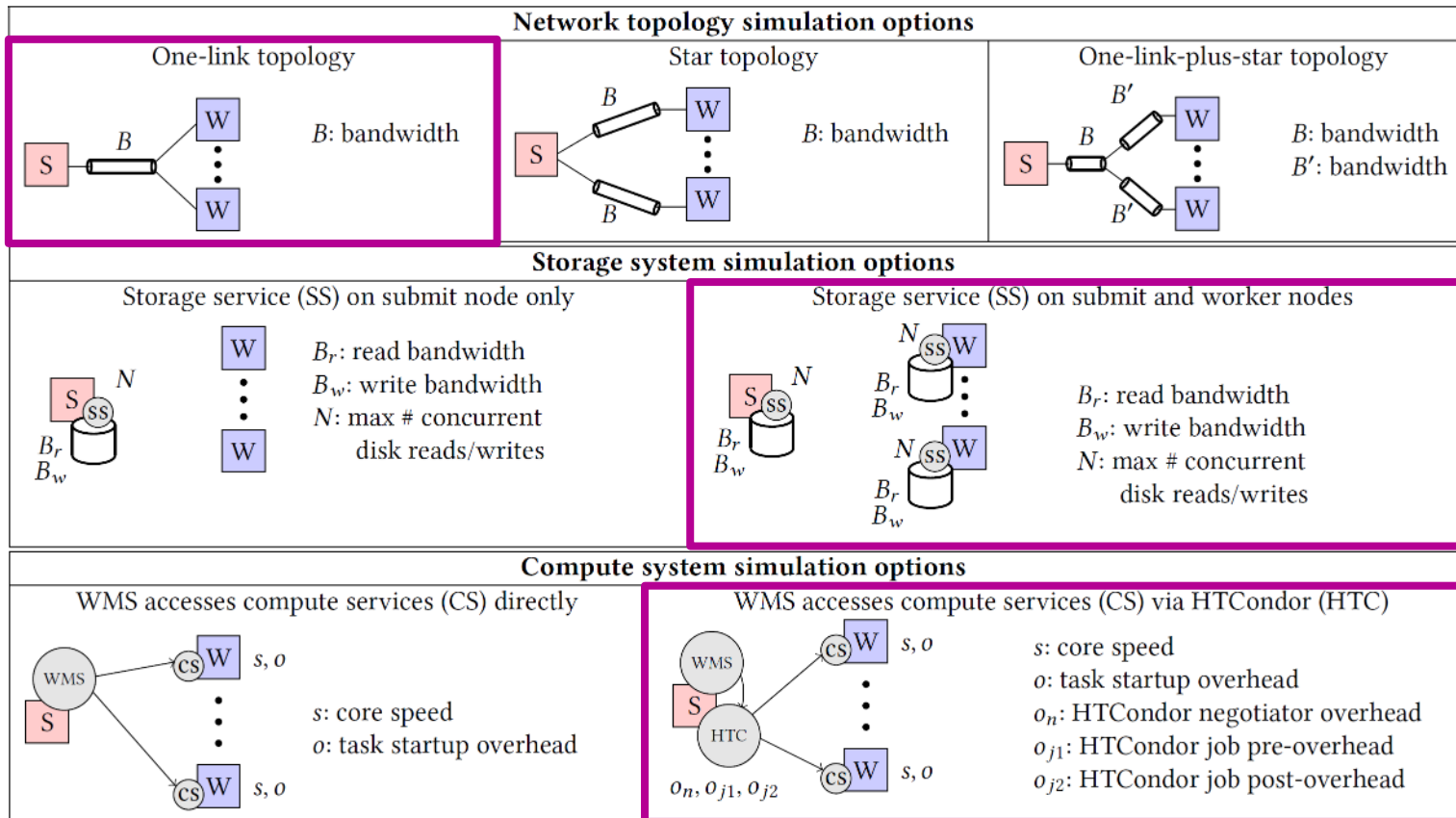
Calibration algorithm

- Grid, Random, Gradient



<https://github.com/wrench-project/simcal>

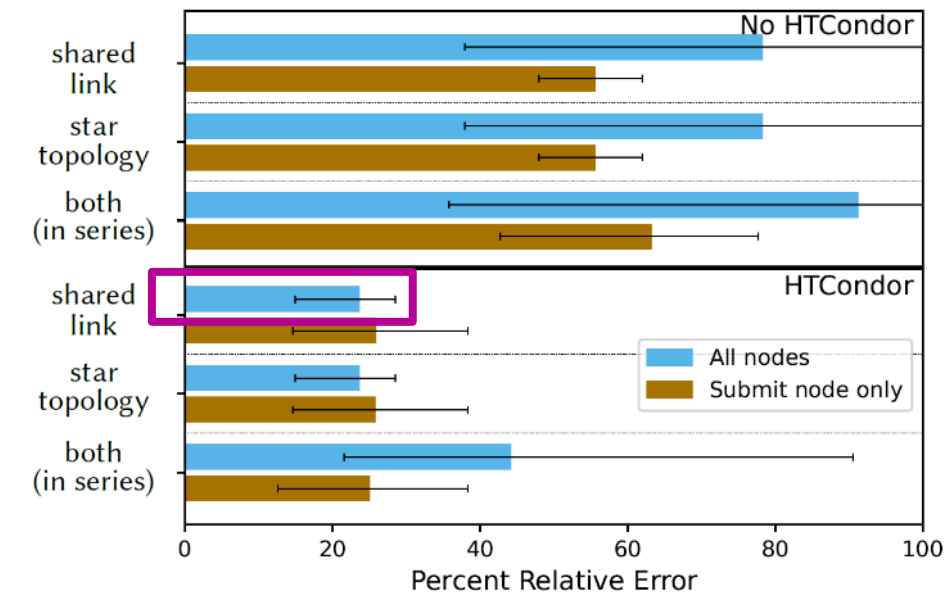
Determining Levels of Detail for Simulators



Evaluation of Workflow Scheduling strategies

Ground truth

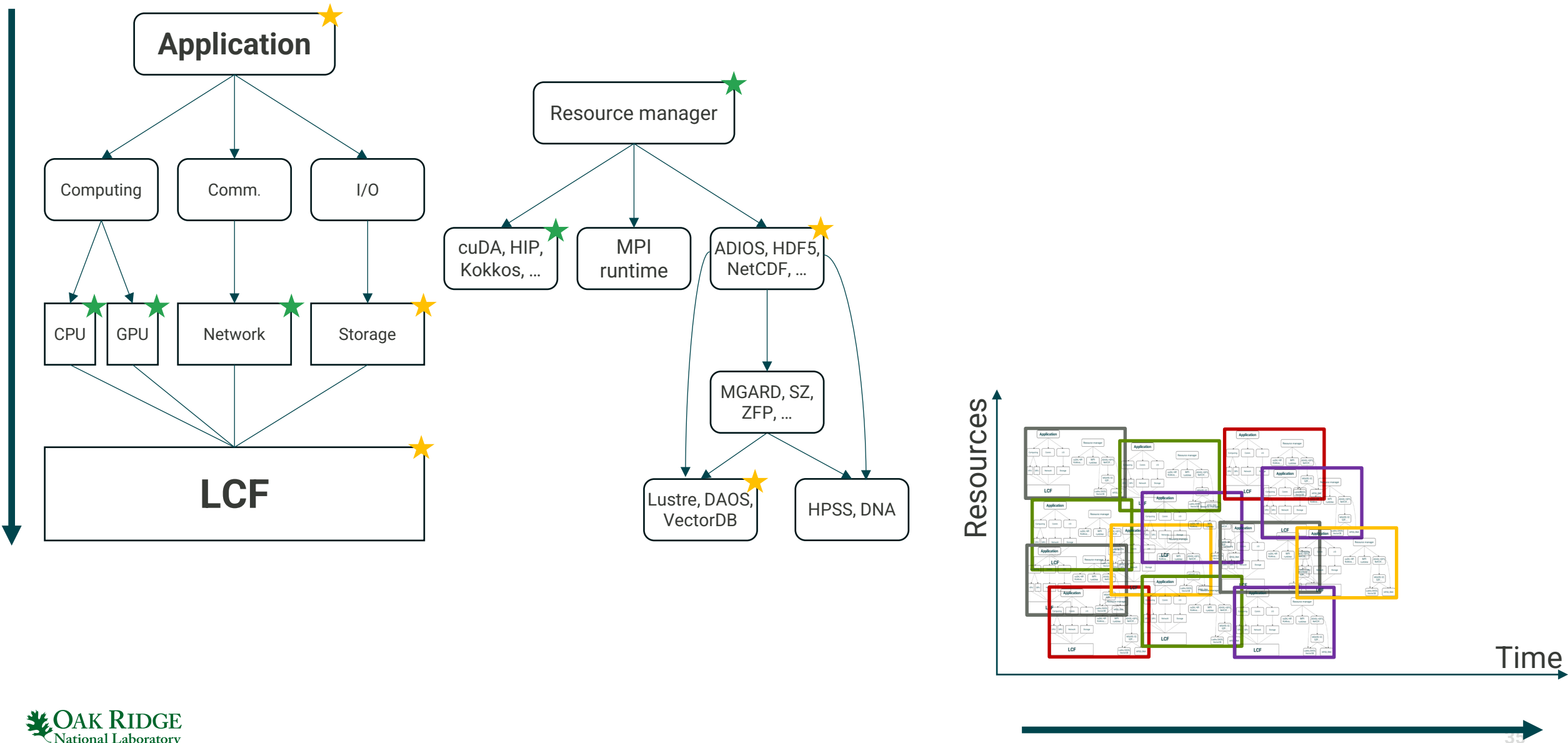
- 5 application workflows
- 5 sizes (#tasks)
- 5 per-task CPU work amounts
- 4 data footprints



I/O simulation – The ground truth challenge



There can't be any (useful) simulation model without data

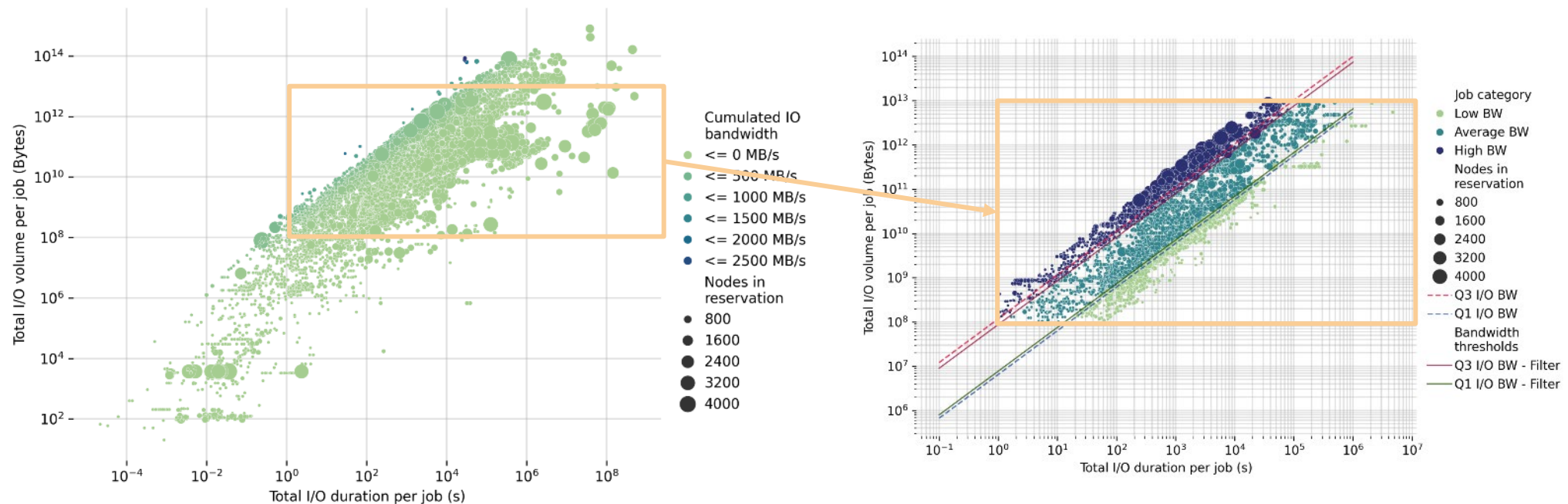


Finding Relevant Datasets is Hard and Requires Processing!

Darshan logs from **Theta** (ANL) → Year 2022, ~18,000 jobs

Need for **filtering** and **clustering** of monitored jobs:

- Removing jobs with no I/O activity, classification based on bw performance, etc
- Reducing heterogeneity / addressing technical limitations of simulators



Challenges

Performance data management faces the same challenges as **scientific data management**

Challenge 1: How can we **capture** and efficiently **export and store performance data**?

Design monitoring tools along the same principles as for science data

Challenge 2: How can disparate information from **multiple sources** regarding data management activities be **fused into useful knowledge**?

Build AI surrogates of DM workloads

Challenge 3: How can we **reproduce** Data Management behavior in a controlled fashion for **“what if” investigations**?

Design multi-scale, multi-fidelity performance evaluation tools

And some more ...

Data sharing vs. **policies**, integration of new **advanced storage**, coordination **beyond DM**, ...

Conclusion

Simulation can be used to assess the performance of I/Os and data management

- From the **resources** (a.k.a. disks)
- To application **workflows**
- Through (distributed) **file systems**

Models and **tools** exist

- SimGrid, FSMod, FIVES, WRENCH, DTLMod ... and many others outside the SimGrid ecosystem
- **Contributions welcomed!**

But **challenges** remain

- **Ground truth data** acquisition and accessibility
- **Calibration** of the simulation models
- Selection of the appropriate **level of detail**

Questions?

Thank you for your attention

