Uncovering I/O Usage in HPC Platforms

André Ramos Carneiro

Advisor: Prof. Dr. Philippe O. A. Navaux
Co-advisor: Prof. Dr. Carla Osthoff

LNCC

Instituto de Informática,
UFRGS, Brasil

.inf INSTITUTO DE INFORMÁTICA UFRGS

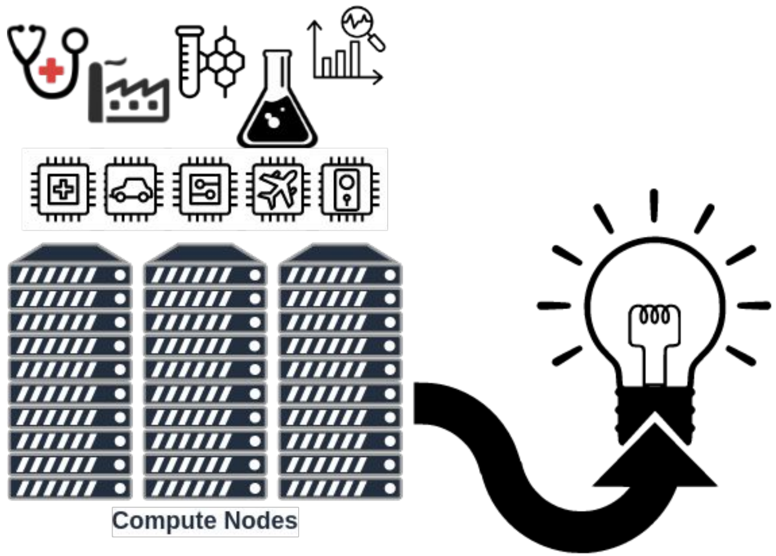
Agenda

- Introduction
- The Lustre Deployment on SDumont
- Related Work
- Analysis and Visualization Methodology
- Results - Glancing at the Lustre Filesystem
- Conclusion and Future Work

Introduction

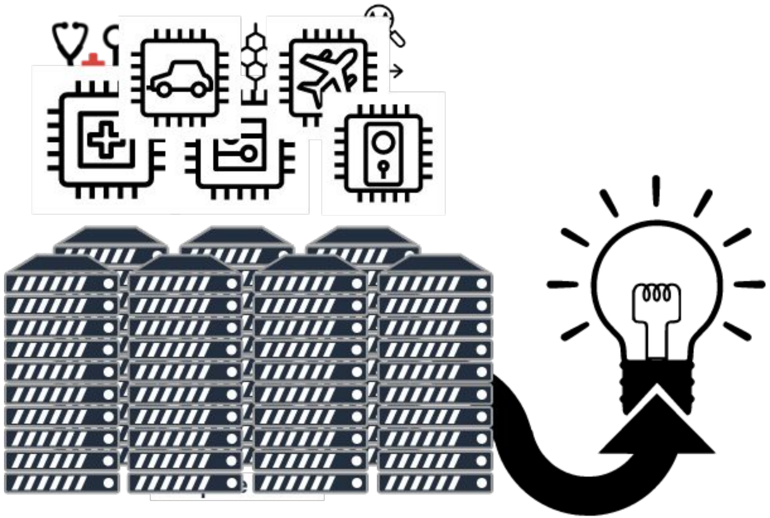
Introduction

- Supercomputers **dominate** the High-Performance Computing (**HPC**) environments.
- Used to **solve** the most diverse problems in **various fields**: biology, chemistry, physics, and health sciences.
- Each science domain use a **multitude of scientific software**.
- Supercomputers have to **handle mixed workloads**.



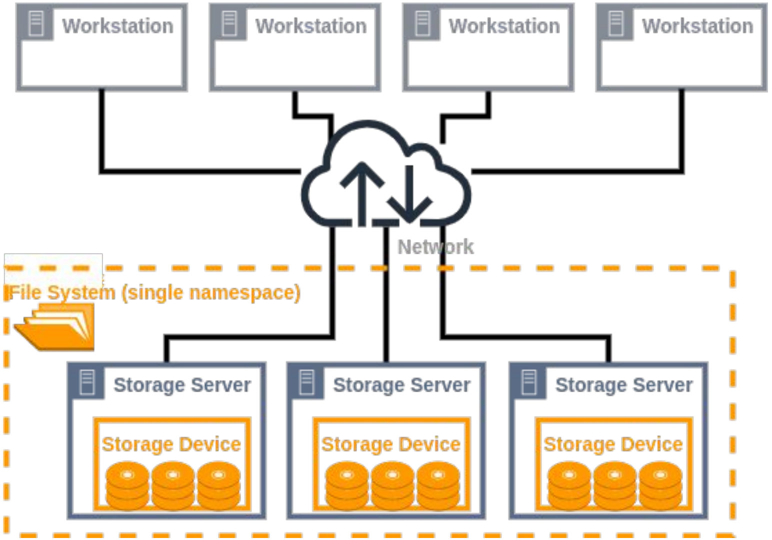
Introduction

- As the supercomputers **increase** in size (CPU and Mem.), so does the size of the **dataset used**.
- Data storage is one of the main **bottlenecks**
  - Performance gap between **CPU and I/O**
  - Rising **concurrency** and **interference**
  - **Metadata** operations
- Different scientific applications are **impacted** in diverse ways by storage system
- Performance limiting **factors**
  - Access patterns
  - Load imbalance between storage servers
  - Resource contention



Introduction

- **Parallel File Systems** (PFS) are the *de-facto* file system type for HPC systems.
- Decentralized Networked File System
- Provide
  - High-performance data access
  - Division of files in data blocks (*striping*)
  - Single namespace
  - Fault-Tolerant
  - Locking
  - Cache coherency
- **Lustre** is one of the most adopted PFS (≈ 30% of the file systems used on IO500 [SC21]).
  - Open-source
  - Client-server
  - Object-based



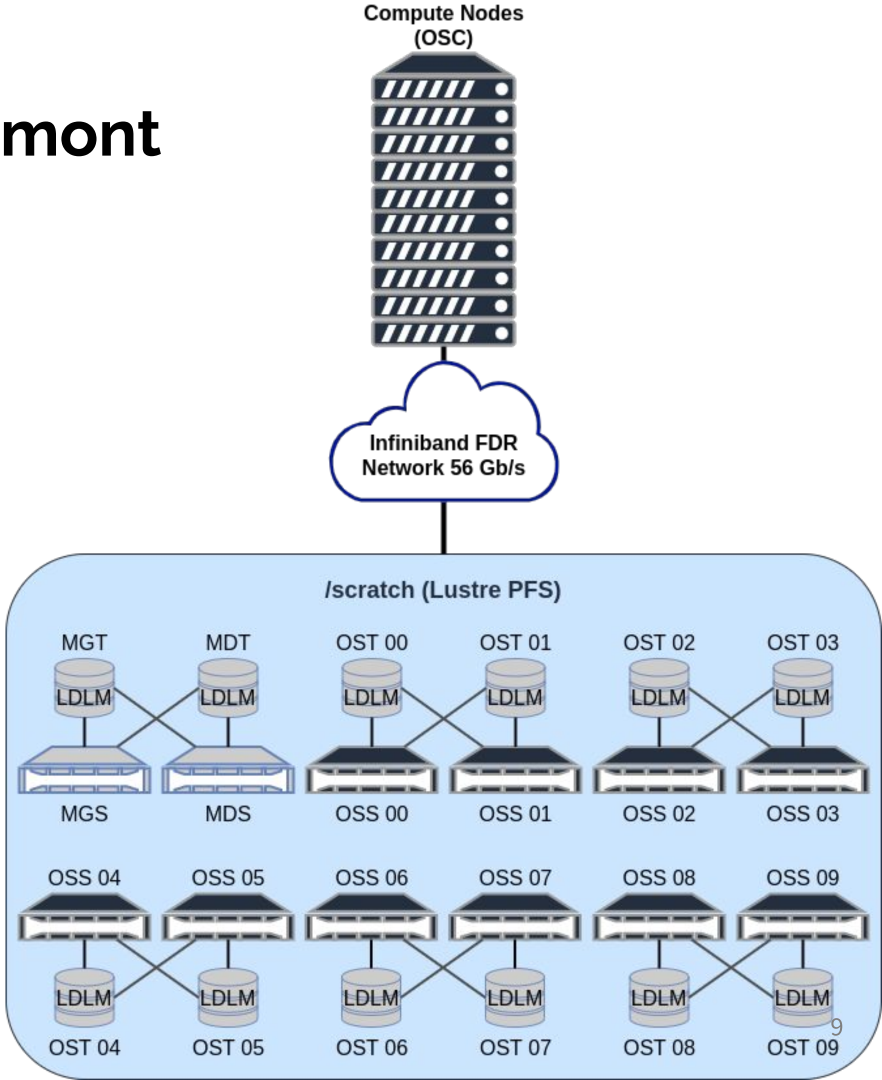
Introduction

- Our research aims to **understand** the **impact** and **uncover** data storage **needs** in a supercomputer by **evaluating** the Lustre's performance concerning the **varied workloads** from different domains.
- We provide a **methodology to visualize** performance factors, such as **small request sizes**, **load imbalance**, **resource contention**, and **metadata utilization**.
- We use the Santos Dumont Supercomputer (**SDumont**) as a case study.
- **Three months** of operational data (**March to May**) from two years (**2020 and 2021**).
- The study of the Lustre file system on SDumont was divided into **two parts**:
  - Analysis of the whole **three months** period
  - Focus on a **specific period** of interest

The Lustre Deployment on SDumont

The Lustre Deployment on SDumont

- A Supercomputer located at the National Laboratory for Scientific Computing (LNCC)
- Chemistry (21.3%), Physics (17.1%), Engineering (12.6%), Biological Sciences (10,1%), and Computer Science (9.1%).
- 758 nodes (18,424 CPU cores) - 1.1 petaflops
- **Lustre PFS ClusterStor 9000 v3.3**
    - 1 x MDS & 1 MDT + 10 OSS & 10 OST
    - Max Perf: 45 GiB/s (2,700 GiB/m)
    - `stripe_count = 1`
      `stripe_size = 1 MiB`



Related Works

Related Works

- **Luu et al.** (2015) analyzed **Darshan's** logs from more than one million jobs on three leading HPC supercomputer platforms: Intrepid and Mira at ALCF and Edison at NERSC.
  - Drawbacks: Only use Darshan, lack of server side information
- Lockwood et al. (2018) used **TOKIO**, benchmarks, and active probing on the PFS of two leadership-class HPC centers (NERSC and ALCF).
  - Drawbacks: Use Darshan, LMT (not supported), and active probing (may cause interference).
- Patel et al. (2019) developed a tool to analyze the log data of **LMT** from the Lustre PFS at NERSC HPC data center, shared by Edison and Cori supercomputers.
  - Drawbacks: Only use LMT (server side information), need a DBMS (not supported or allowed)
- Sivalingam et al. (2019) used **LASSi** to analyze application usage and contention caused by the use of shared resources on the Lustre PFS deployed at ARCHER supercomputer
  - Drawbacks: MySQL (not supported or allowed)
- Betke and Kunkel (2019) identify anomalies or high workloads from jobs' telemetric data through a workflow based on **Machine Learning**.
  - Drawbacks: Not mature yet (needs manual adjustment)

Related Works

- Luu et al. (2015) analyzed **Darshan's** logs from more than one million jobs on three leading HPC supercomputer platforms: Intrepid and Mira at ALCF and Edison at NERSC.
  - Drawbacks: Only use Darshan, lack of server side information
- **Lockwood et al.** (2018) used **TOKIO**, benchmarks, and active probing on the PFS of two leadership-class HPC centers (NERSC and ALCF).
  - Drawbacks: Use Darshan, LMT (not supported), and active probing (may cause interference).
- Patel et al. (2019) developed a tool to analyze the log data of **LMT** from the Lustre PFS at NERSC HPC data center, shared by Edison and Cori supercomputers.
  - Drawbacks: Only use LMT (server side information), need a DBMS (not supported or allowed)
- Sivalingam et al. (2019) used **LASSi** to analyze application usage and contention caused by the use of shared resources on the Lustre PFS deployed at ARCHER supercomputer
  - Drawbacks: MySQL (not supported or allowed)
- Betke and Kunkel (2019) identify anomalies or high workloads from jobs' telemetric data through a workflow based on **Machine Learning**.
  - Drawbacks: Not mature yet (needs manual adjustment)

Related Works

- Luu et al. (2015) analyzed **Darshan's** logs from more than one million jobs on three leading HPC supercomputer platforms: Intrepid and Mira at ALCF and Edison at NERSC.
    - Drawbacks: Only use Darshan, lack of server side information
- Lockwood et al. (2018) used **TOKIO**, benchmarks, and active probing on the PFS of two leadership-class HPC centers (NERSC and ALCF).
    - Drawbacks: Use Darshan, LMT (not supported), and active probing (may cause interference).
- ➢ **Patel et al.** (2019) developed a tool to analyze the log data of **LMT** from the Lustre PFS at NERSC HPC data center, shared by Edison and Cori supercomputers.
    - Drawbacks: Only use LMT (server side information), need a DBMS (not supported or allowed)
- Sivalingam et al. (2019) used **LASSi** to analyze application usage and contention caused by the use of shared resources on the Lustre PFS deployed at ARCHER supercomputer
    - Drawbacks: MySQL (not supported or allowed)
- Betke and Kunkel (2019) identify anomalies or high workloads from jobs' telemetric data through a workflow based on **Machine Learning**.
    - Drawbacks: Not mature yet (needs manual adjustment)

Related Works

- Luu et al. (2015) analyzed **Darshan's** logs from more than one million jobs on three leading HPC supercomputer platforms: Intrepid and Mira at ALCF and Edison at NERSC.
  - Drawbacks: Only use Darshan, lack of server side information
- Lockwood et al. (2018) used **TOKIO**, benchmarks, and active probing on the PFS of two leadership-class HPC centers (NERSC and ALCF).
  - Drawbacks: Use Darshan, LMT (not supported), and active probing (may cause interference).
- Patel et al. (2019) developed a tool to analyze the log data of **LMT** from the Lustre PFS at NERSC HPC data center, shared by Edison and Cori supercomputers.
  - Drawbacks: Only use LMT (server side information), need a DBMS (not supported or allowed)
- ➢ **Sivalingam et al.** (2019) used **LASSi** to analyze application usage and contention caused by the use of shared resources on the Lustre PFS deployed at ARCHER supercomputer
  - Drawbacks: MySQL (not supported or allowed)
- Betke and Kunkel (2019) identify anomalies or high workloads from jobs' telemetric data through a workflow based on **Machine Learning**.
  - Drawbacks: Not mature yet (needs manual adjustment)

Related Works

- Luu et al. (2015) analyzed **Darshan's** logs from more than one million jobs on three leading HPC supercomputer platforms: Intrepid and Mira at ALCF and Edison at NERSC.
  - Drawbacks: Only use Darshan, lack of server side information
- Lockwood et al. (2018) used **TOKIO**, benchmarks, and active probing on the PFS of two leadership-class HPC centers (NERSC and ALCF).
  - Drawbacks: Use Darshan, LMT (not supported), and active probing (may cause interference).
- Patel et al. (2019) developed a tool to analyze the log data of **LMT** from the Lustre PFS at NERSC HPC data center, shared by Edison and Cori supercomputers.
  - Drawbacks: Only use LMT (server side information), need a DBMS (not supported or allowed)
- Sivalingam et al. (2019) used **LASSi** to analyze application usage and contention caused by the use of shared resources on the Lustre PFS deployed at ARCHER supercomputer
  - Drawbacks: MySQL (not supported or allowed)

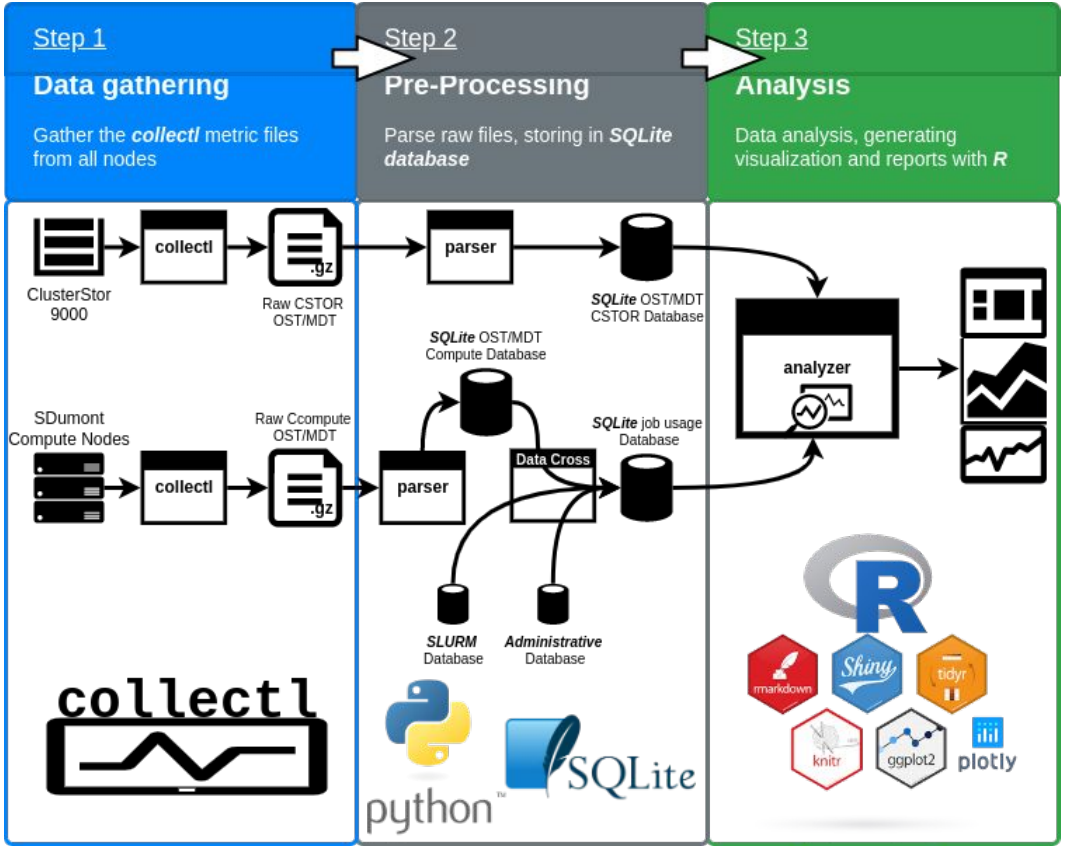
➢ **Betke and Kunkel** (2019) identify anomalies or high workloads from jobs' telemetric data through a workflow based on **Machine Learning**.
  - Drawbacks: Not mature yet (needs manual adjustment)

Related Works

- We propose:
    - **Broader** methodology to provide a **bigger picture** of the whole system's I/O utilization.
    - **Continuous analysis** from the **Storage Devices** to the **Compute Nodes**.
    - Characterize **data** and **metadata** usage.
    - Tracking **inefficient behavior**.
    - Adopted the use of **open-source** software that **does not require administrative privileges**.
    - Easily **implemented** and **reproduced**.

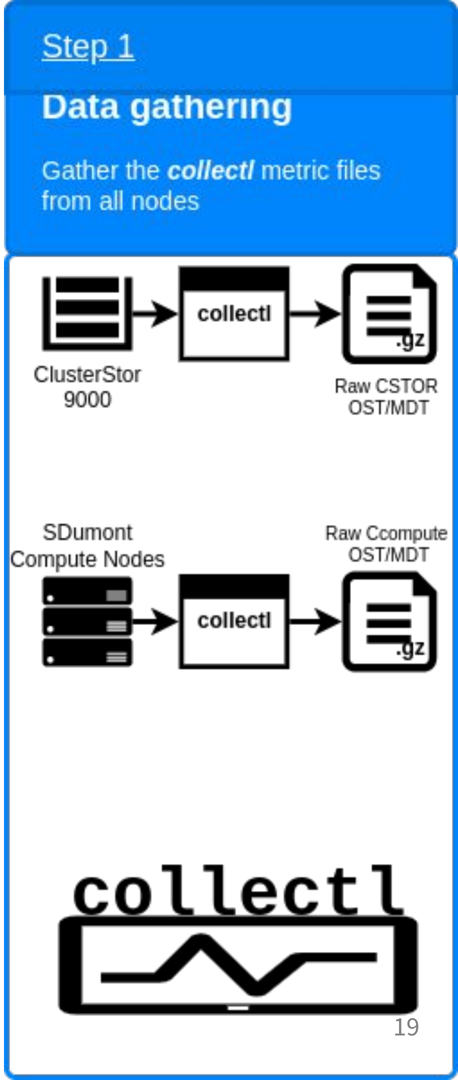
Analysis and Visualization Methodology

Analysis and Visualization Methodology



Analysis and Visualization Methodology

- *collectl*, an open-source system performance monitoring tool
- Special plugin for **Lustre PFS**
- Installed on **MDS** and **OSS** servers of ClusterStor
- Installed on **758** SDumont **Compute Nodes**
- **15 sec**. collection interval, stored on local `/tmp`
- Neglectable overhead (**0.1%** of CPU).



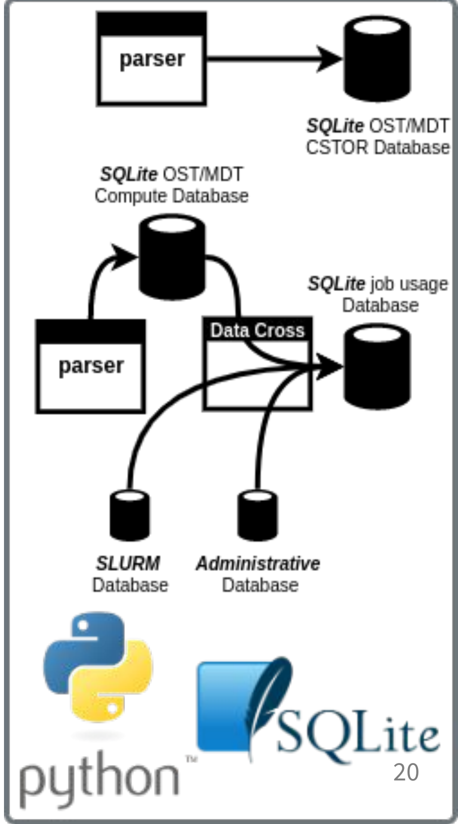
Analysis and Visualization Methodology

- Conversion of the **daily raw** *collectl* file to an easy to **use** and **transport** SQLite dataset
- Two datasets: *ClusterStor* and *Compute Nodes*
- **“Data Cross”** process to cross information from:
    - Compute Nodes dataset (utilization metrics) +
    - Slurm Database (job’s name, nodes, start and end) +
    - Administrative Database (Science Domain)
    - = ***Job Usage*** dataset: “who, how and why”

Step 2

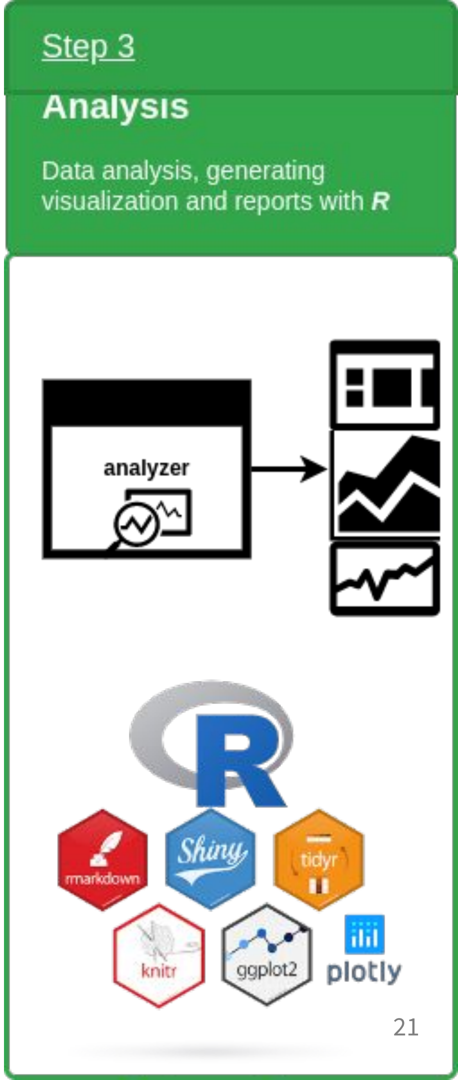
Pre-Processing

Parse raw files, storing in ***SQLite database***



Analysis and Visualization Methodology

- **Visualization and analysis** tool developed with R+Shiny
- **Reproduce** the process with dataset from **different periods**
- WebApp: https://arcarneiro.shinyapps.io/sdumont_lustre



Analysis and Visualization Methodology
I/O Metrics

Metric	Description	
reads	Number of read operations	Default collectl metrics Obtained at Step 1
$read_{kb}$	KiB data read	
writes	Number of write operations	
$write_{kb}$	KiB data written	
$read_{size}$	Block size of read operation ($read_{kb}/reads$)	
$write_{size}$	Block size of write operation ($write_{kb}/writes$)	
$read_{qo}$	**Q**uality of read **o**peration ($(reads * 1024)/read_{kb}$)	
$write_{qo}$	**Q**uality of write **o**peration ($(writes * 1024)/write_{kb}$)	
CF_{bw}	Bandwidth Coverage Factor of a job	
LI	Load Imbalance	
SMA_{3HR}	Simple Moving Averages of three hours	

Analysis and Visualization Methodology
I/O Metrics

Metric	Description
reads	Number of read operations
$read_{kb}$	KiB data read
writes	Number of write operations
$write_{kb}$	KiB data written
$read_{size}$	Transfer size of read operation ($read_{kb}/reads$)
$write_{size}$	Transfer size of write operation ($write_{kb}/writes$)
$read_{qo}$	**Q**uality of read **o**peration ($(reads * 1024)/read_{kb}$)
$write_{qo}$	**Q**uality of write **o**peration ($(writes * 1024)/write_{kb}$)
CF_{bw}	Bandwidth Coverage Factor of a job
LI	Load Imbalance
SMA_{3HR}	Simple Moving Averages of three hours

Derived metrics

Generated at Step 2

* The average **transfer size**

* **Q**uality of **O**peration (QO), based on the default striping policy of SDumont (1MiB).

1 50 100 …

Efficient -> Inefficient

Analysis and Visualization Methodology
I/O Metrics

Metric	Description	
reads	Number of read operations	Derived metrics Generated at Step 3
$read_{kb}$	KiB data read	
writes	Number of write operations	
$write_{kb}$	KiB data written	***CF_{bw}*** indicates the amount of bandwidth that can be attributed to a job.
$read_{size}$	Block size of read operation ($read_{kb}$/*reads*)	
$write_{size}$	Block size of write operation ($write_{kb}$ /*writes*)	
$read_{qo}$	**Q**uality of read **o**peration ((*reads* * 1024)/$read_{kb}$)	***LI*** measures the load imbalance among the OSTs
$write_{qo}$	**Q**uality of write **o**peration ((*writes* * 1024)/$write_{kb}$)	
CF_{bw}	Bandwidth Coverage Factor of a job	
LI	Load Imbalance	***SMA_{3HR}*** is calculated for all other metrics and is helpful during visualization
SMA_{3HR}	Simple Moving Averages of three hours	

$$CF_{bw}(job) = \frac{N_{bytes}(job)}{N_{bytes}(Lustre)}$$

$$LI = \frac{\sigma}{\mu}$$

$$SMA_{tf}(m) = \frac{1}{tf} \sum_{i=t-tf}^{t} m_i$$

Analysis and Visualization Methodology
Metadata Counters

Counter	Node	Description
fopen	MDS & Client	File open requests
fclose		File close requests
getattr		Operation that get file/dir attributes
setattr		Operation that set file/dir attributes
fsync		Operation that synchronizes data to the file system
getxattr	MDS	Operation that get file/dir extended attributes
setxattr		Operation that set file/dir extended attributes
unlink		File/dir removals
link		Hard or symbolic link creation
statfs		Operation that return statistics about the file system
mkdir		Directory creation requests
rmdir		Directory removal requests
seek	Client	Operation that change the file pointer

Results - Trimester Analysis

Results - Trimester Lustre Usage Analysis
I/O - OSS Nodes

- 3 months from the ***ClusterStor*** dataset, spanning from March to May, 2020 and 2021.
- Whole file system (**sum of all OSTs**)

	2020	2021	
Jobs	36,884	145,793	4× ↑
Total Read	1.8 PiB	7.95 PiB	4.7× ↑
Total Write	2.9 PiB	4.1 PiB	1.5× ↑
Read Ops	64.154 B	39.102 B	1.6× ↓
Write Ops	1.234 B	5.297 B	4.3× ↑
Peak Read Throughput	316 GiB/m (≈ 11.7% bw)	1,077 GiB/m (≈ 39.89% bw)	3.4x ↑
Avg. Read Throughput	15.825 GiB/m	66.953 GiB/m	4.2× ↑
Peak Write Throughput	1,127 GiB/m (≈ 41.74% bw)	1,145 GiB/m (≈ 42.41% bw)	-
Avg. Write Throughput	25.336 GiB/m	34.452 GiB/m	1.3× ↑

Results - Trimester Lustre Usage Analysis
I/O - OSS Nodes

Table 5.1 – Transfer Size (KiB) and Quality of Operations.

Year	Operation	Metric	Min.	1st Q.	Median	3rd Q.	Max.
2020	**Read**	*Size*	4.00	6.60	23.80	577.00	4096.00
		QO	0.25	1.77	43.00	155.00	256.00
	Write	*Size*	0.01	530.00	1458.00	2947.00	4096.00
		QO	0.25	0.35	0.70	1.93	525131.00
2021	**Read**	*Size*	4.00	271.00	816.00	1786.00	4096.00
		QO	0.25	0.57	1.25	3.78	256.00
	Write	*Size*	0.01	420.00	970.00	2149.00	4096.00
		QO	0.25	0.48	1.10	2.44	104865.00

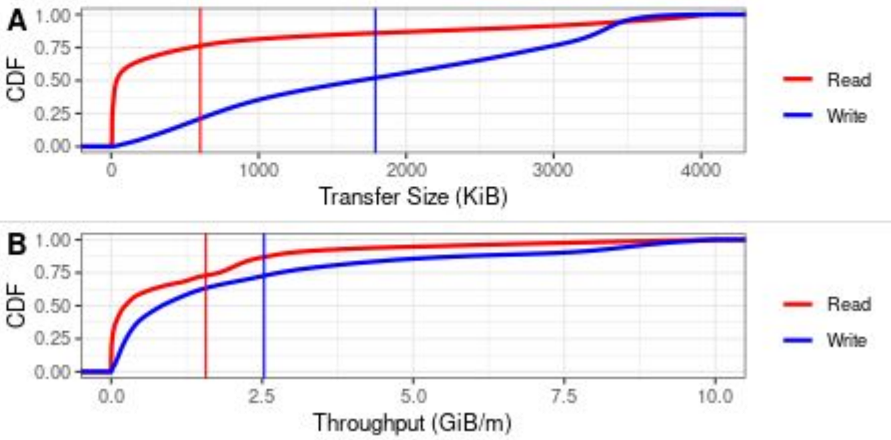
Source: Author

Results - Trimester Lustre Usage Analysis
I/O - OSS Nodes

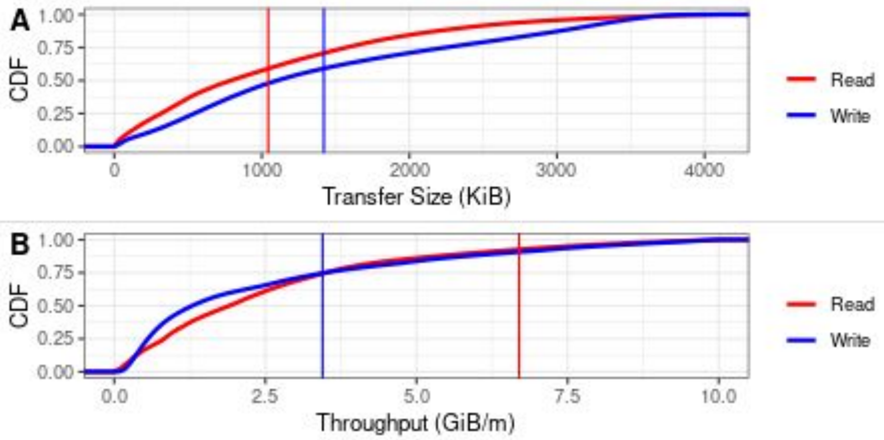
CDF of the Operation Size (A) and Throughput (B) for the Read (Red) and Write (Blue) operations **among OSTs**.

2020 avg: **652 KiB** Read and **1729 KiB** Write for Size (**≃3x**), and **1.5 GiB/m** Read and **2.2 GiB/m** Write for Throughput (**≃1.6x**).

2021 avg: **1043 KiB** Read and **1420 KiB** Write for Size, and **6.7 GiB/m** Read and **3.4 GiB/m** Write for Throughput (**≃2x**).



2020

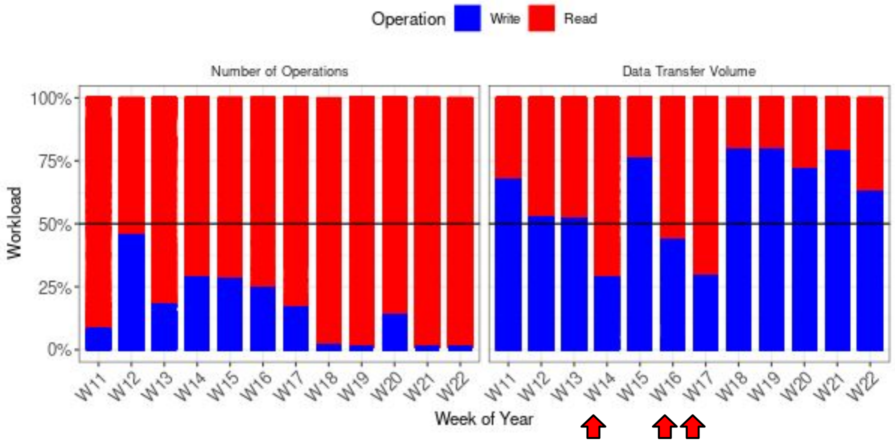


2021

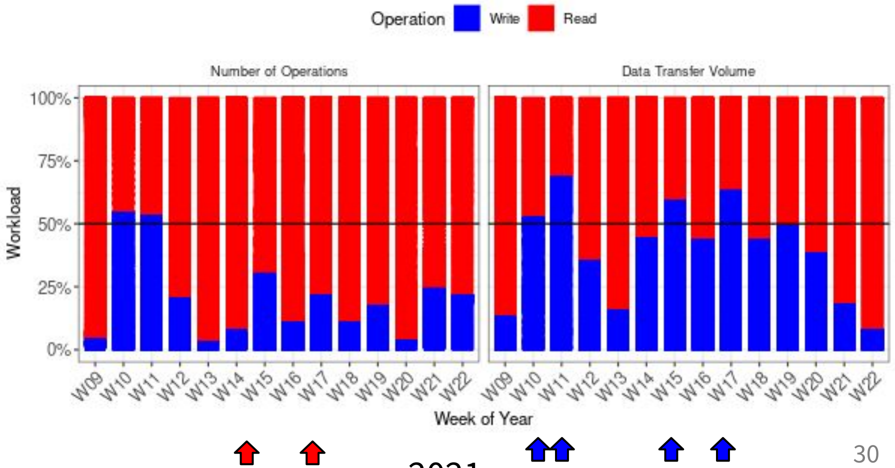
Results - Trimester Lustre Usage Analysis
I/O - OSS Nodes

Workload distribution by week.

- 2020: Write dominated data movement (**61%**), Read dominated number of operations (**98%**)
    - **≃1.6× write-to-read volume / ≃52× read-to-write requests**
- 2021: Read dominated both data movement (**66%**) and number of operations (**88%**)
    - **≃2× read-to-write volume / ≃7× read-to-write requests**



2020



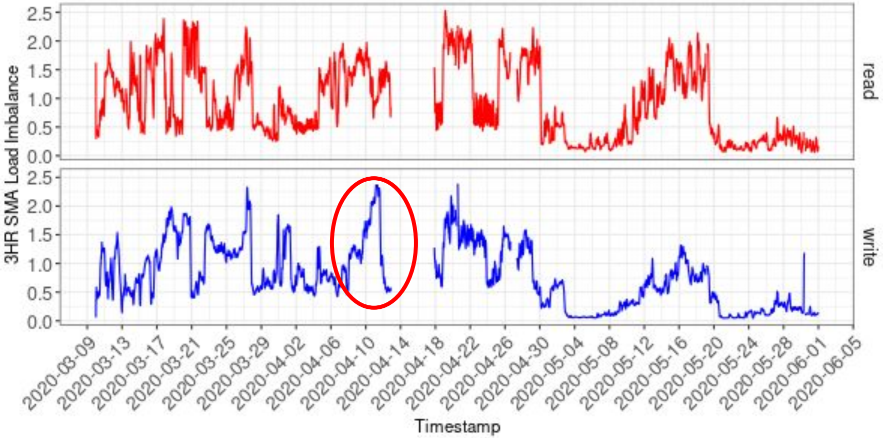
2021

Results - Trimester Lustre Usage Analysis
I/O - OSS Nodes

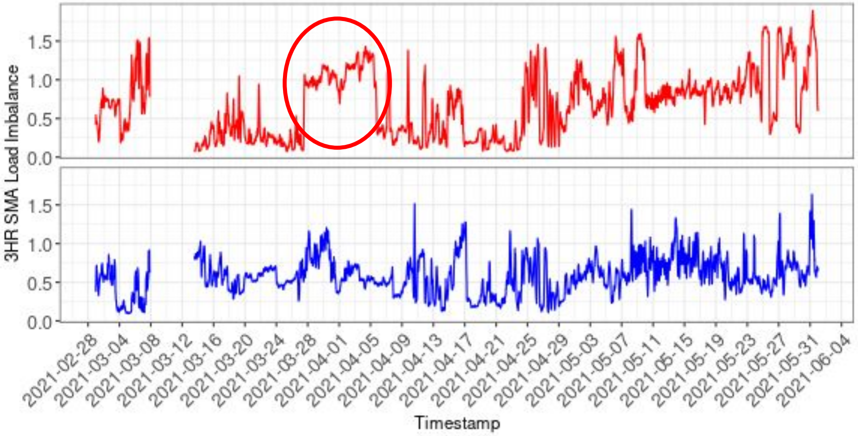
SMA$_{\mathbf{3HR}}$ of **LI** for the read (Red) and write (Blue) load. Values below **0.5** can be considered as **low** imbalance, values around **1** are considered as **moderate** imbalance, and values **above** are considered **severe** imbalance.

2020: 50% below 0.6 / 25% above 1. Avg for reading was **0.92** while for writing was **0.80**.

2021: 50% below 0.6 / 25% above 1. Avg for reading was **0.68** while for writing was **0.58**.



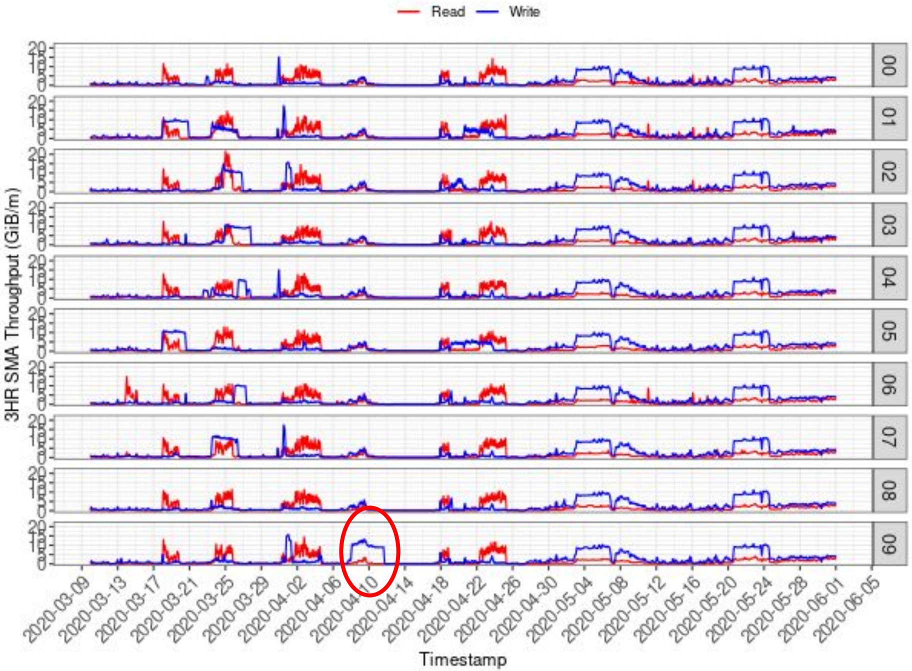
2020



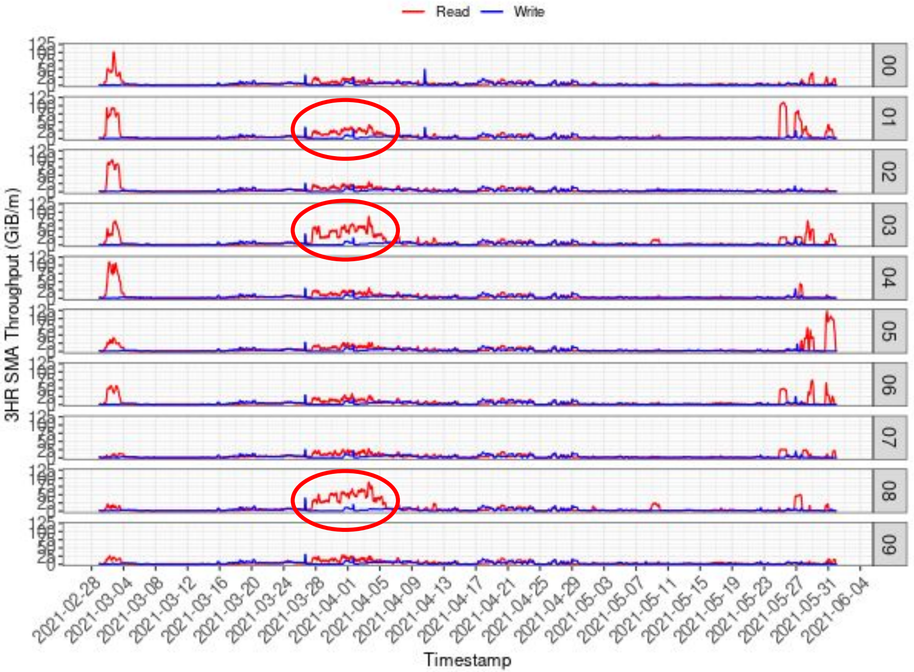
2021

Results - Trimester Lustre Usage Analysis
I/O - OSS Nodes

SMA_{3HR} of read and write throughput by OST.



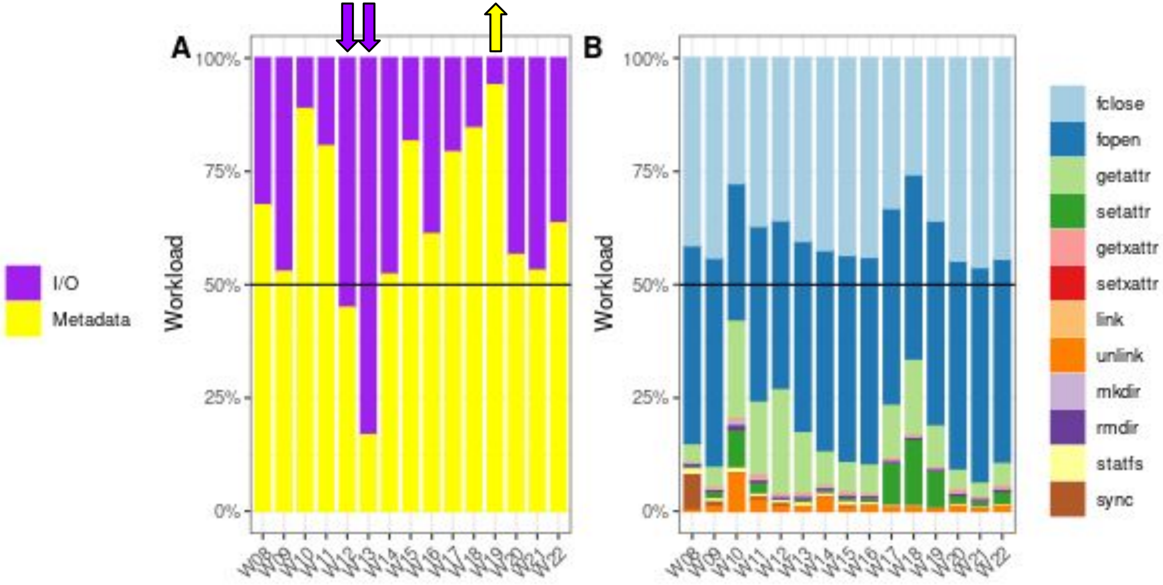
2020



2021

Results - Trimester Lustre Usage Analysis Metadata - MDS Node

- 3 months of data, spanning from March to May, **2021**.
- Avg 8,920 ops/s
  Max 205,016 ops/s.
- Metadata **60%** operations (67 B MD x 44 B I/O)
- **+ fopen, fclose, getattr, setattr**
- “Low” unlink operations



Results - Period of Interest

Results - Detailed View of a Region of Interest
I/O - Compute Nodes

- In-depth analysis with ***Job Usage*** dataset
- 2020 - Detailed on the dissertation
  - **March 24th** and **March 28th**
  - Read **peak throughput** of 2020
- 2021
  - **March 28th** and **April 1st**
  - Expressive **increase** in **read** activity, resulting in **load imbalance**
  - 845 jobs
- With the SLURM's information, we were able to identify eleven different applications:
  - DockThor (36.21%), **unknown (17.75%)**, QUANTUM ESPRESSO (10.06%), LHCB DIRAC (8.88%), AMBER (7.57%), GROMACS (6.98%), **OpenMPI mpiexec (4.62%)**, VASP (4.62%), **Bash Script (1.3%)**, LAMMPS (0.71%), ORCA (0.47%), SIESTA (0.47%), Python (0.24%), and BIE (0.12%).
- The system was used by twelve different Science Domains:
  - Astronomy, Biodiversity, Biological Sciences, **Chemistry**, Computer Science, Engineering, Geosciences, Health Sciences, Materials Science, Mathematics, **Physics**, Weather and Climate

Results - Detailed View of a Region of Interest
I/O - Compute Nodes

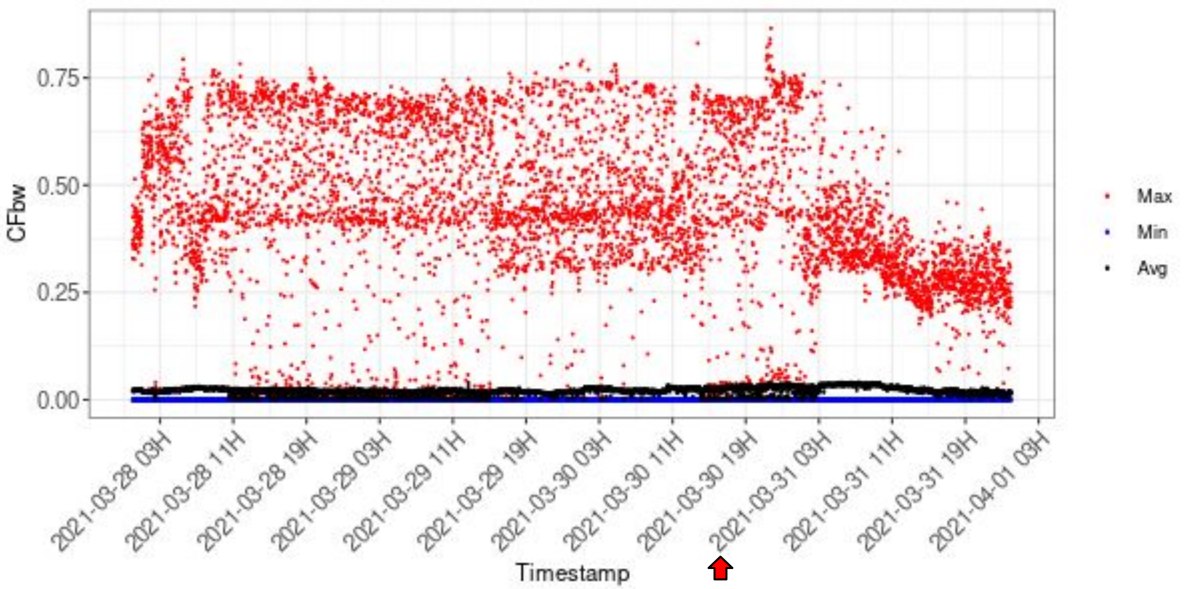
Table 5.3 – Individual application's throughput

Year	Application	Operation	GiB/m	CF_{bw}
2020	QUANTUM ESPRESSO	Read	290	0.84
	QUANTUM ESPRESSO	Write	353	0.94
2021	*unknown*	Read	153	0.70
	QUANTUM ESPRESSO	Write	90	0.31

Results - Detailed View of a Region of Interest I/O - Compute Nodes

CF_{bw} of the jobs. The dots in **red**, **black**, and **blue** represent the **Max**., **Avg**. and **Min**., respectively, of all jobs, observed on each timestamp.

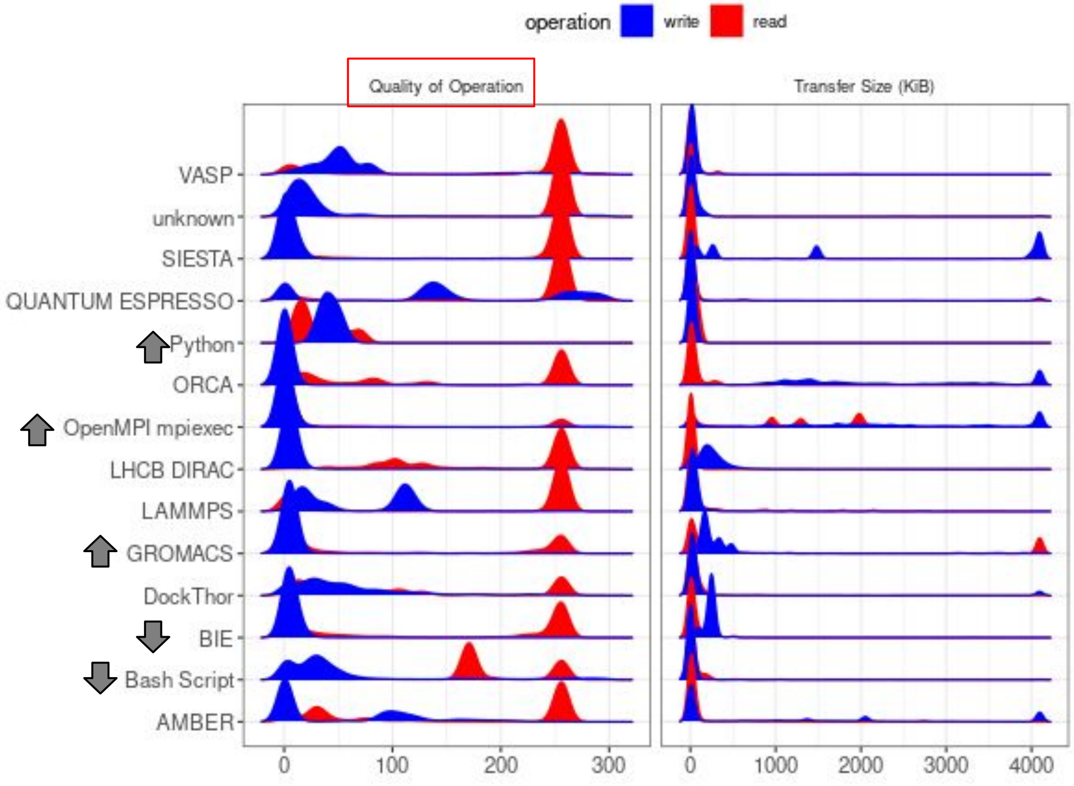
Few jobs with elevated throughput consume the bandwidth



Results - Detailed View of a Region of Interest
I/O - Compute Nodes

2021 Distribution of the **Quality of Operation (left)** and Transfer Size (right).

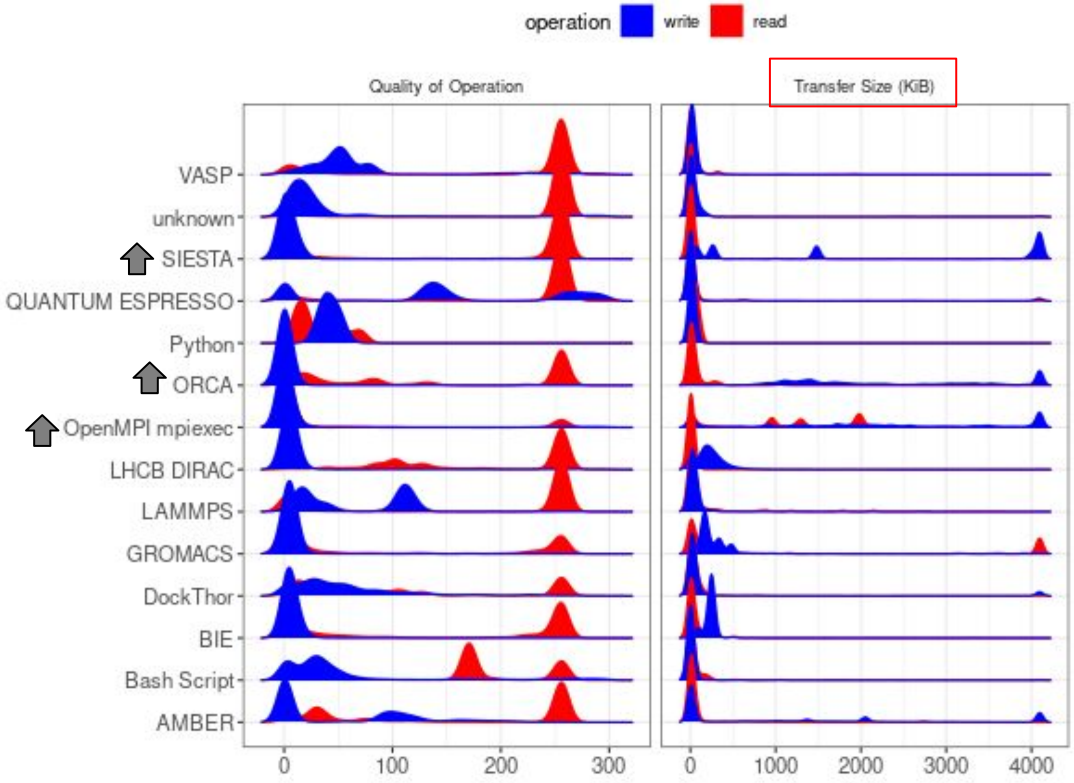
- Most applications are read inefficient
- **“Efficient”**
    - GROMACS, OpenMPI mpiexec, and Python
- **Inefficient**
    - Bash Script and BIE



Results - Detailed View of a Region of Interest I/O - Compute Nodes

2021 Distribution of the Quality of Operation (left) and **Transfer Size (right)**.

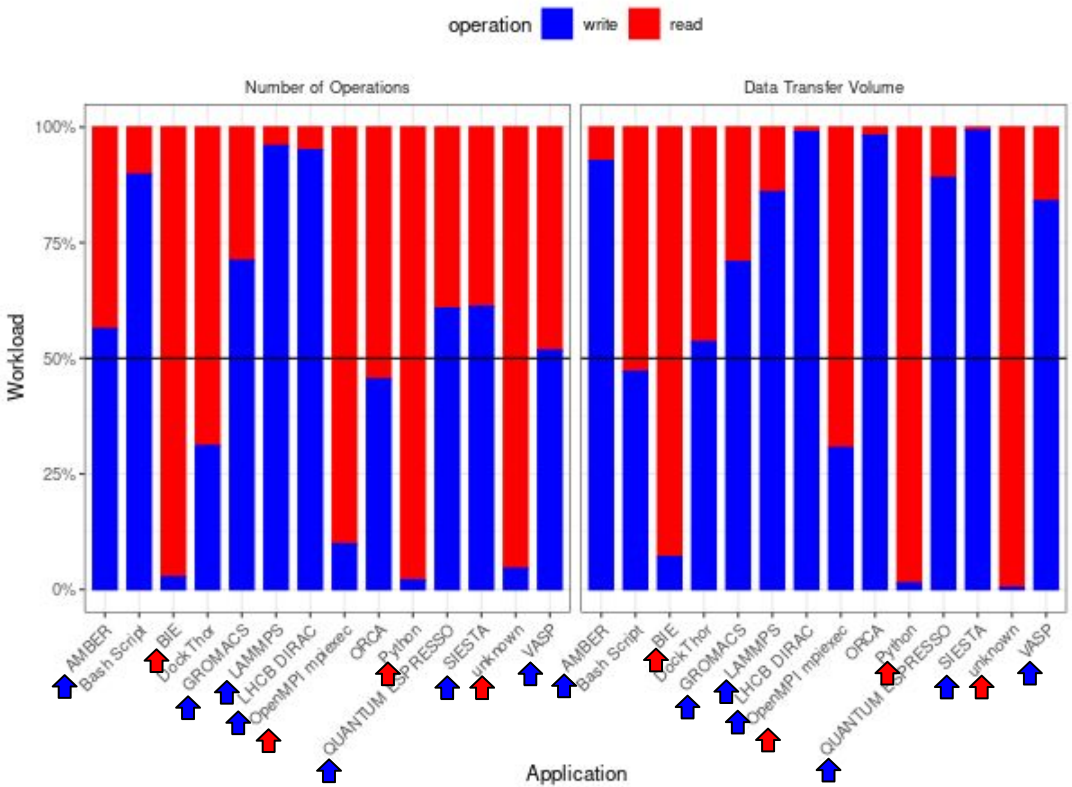
- Seldom use sizes larger than **1 MiB**.
- **< 100 KiB** for **75%** of the time.
- **4 MiB** limit
  - Default maximum bulk I/O RPC
  - Up to **16 MiB**
- OpenMPI biggest sizes
  - Reads (**50% above 1 MiB**)
    Writes (**75% above 1 MiB**)
- ORCA and SIESTA
  - Write above **1.5 MiB for 50%**



Results - Detailed View of a Region of Interest
I/O - Compute Nodes

2021 applications' workload distribution.

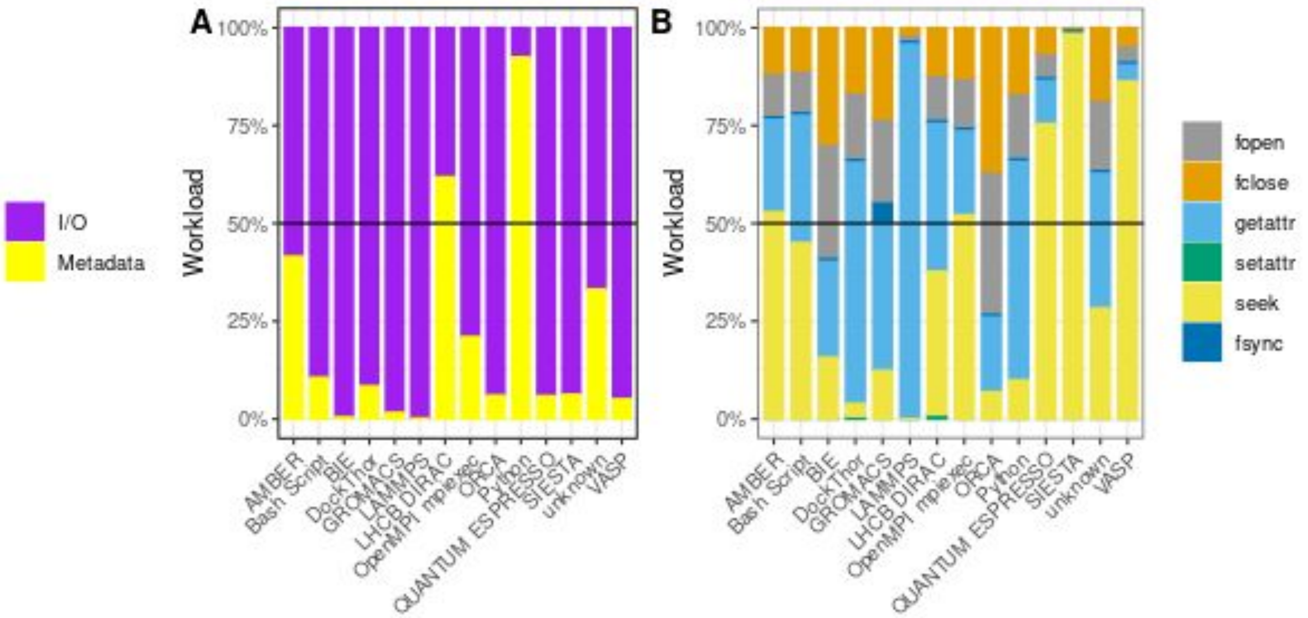
- Most applications are **write-intensive**
  - AMBER, GROMACS, LAMMPS, LHCB DIRAC, QE, SIESTA, VASP
- **4 Read-intensive**
  - BIE, OpenMPI mpiexec, Python, and *unknown*
- Others **mixed** in terms of **number of operations** and **data transferred**
  - Bash: Lots of smaller writes
  - ORCA: Lots of smaller reads



Results - Detailed View of a Region of Interest
Metadata - Compute Nodes

2021 applications' I/O and metadata load distribution

- Metadata **intensive**
  - LHCB DIRAC and Python
- **Heavy** metadata use
  - AMBER and *unknown*
- High ***seek***
  - AMBER, Bash, OpenMPI, QE, SIESTA, VASP
- High ***fopen*** and ***fclose***
  - BIE, ORCA, GROMACS
- High ***getattr***
  - DockThor, LAMMPS, Python



Conclusion

Conclusion

- Proposed a **methodology to visualize and analyze** performance factors on a Lustre PFS.
- The study used **metrics** collected from **storage servers** and **compute nodes**.
- Provided insights into **understanding Lustre's usage and the I/O needs**.
- Identified:
  - Requirements evolution: How the needs and demands **change** from one year to another
  - Inefficient read operations: ≈ **52×** read-to-write requests / ≈ **3×** write-to-read size
  - Demand for Low latency: peak throughput not reaching **50%**, but high demand for **small random operations**
  - Imbalance among resources: some severe and lasting cases where the overload corresponds to **3×** the average OSTs' load.
  - High-level libraries: applications seems to not make full use of libraries to aggregate requests
  - Problematic applications: BIE, which exhibits the **worst read**$_{\mathbf{qo}}$ and is **read-intensive**.
  - Demand for metadata operations: **60%** of all file system operations.

Conclusion - Suggestions

- Inefficient read operations:
  - Adopt I/O forwarding layer
- Demand for Low latency:
  - Use SSDs (client of servers), Lustre’s DoM (Data On Metadata)
- Imbalance among resources:
  - Revise the default striping policy, adopt an automatic load balancer
- High-level libraries and Problematic applications:
  - “Task force” to overhaul the performance, implement a framework to auto-tune the I/O stack
- Demand for metadata operations:
  - Use the Lustre’s DNE (Distributed Namespace)

Conclusion - Future work

- Improving the application identification:
  - Bash Scripts, OpenMPI mpiexec, Python, and *unknown* (**≃24%**)
- Revise some processes to increase the scalability and performance
  - The data cross process is **very time consuming**
- Integrate the metrics collection with SLURM
  - Reduce **space** requirements
- Assess the performance implications of implementing new strategies

Conclusion - Publications

- **CARNEIRO, A. R.**; BEZ, J. L.; BOITO, F. Z.; FAGUNDES, B. A.; OSTHOFF, C.; NAVAUX, P. O. A. Collective I/O Performance on the Santos Dumont Supercomputer. In: 2018 26th <u>Euromicro International Conference on Parallel, Distributed and Networkbased Processing</u> (PDP), 2018.
- BEZ, J. L.; **CARNEIRO, A. R.**; PAVAN, P. J.; GIRELLI, V. S.; BOITO, F. Z.; FAGUNDES, B. A.; OSTHOFF, C.; SILVA DIAS, P. L.; MEHAUT, J.-F.; NAVAUX, P. O. A. I/O Performance of the Santos Dumont Supercomputer. In: <u>The International Journal of High Performance Computing Applications</u>, 2019.
- **CARNEIRO, A. R.**; BEZ, J. L.; OSTHOFF, C.; SCHNORR, L. M.; NAVAUX, P. O. A. HPC Data Storage at a Glance: The Santos Dumont Experience. In: 2021 IEEE 33rd <u>International Symposium on Computer Architecture and High Performance Computing</u> (SBAC-PAD), 2021.
- **CARNEIRO, A. R.**; BEZ, J. L.; OSTHOFF, C.; SCHNORR, L. M.; NAVAUX, P. O. A. Uncovering I/O Demands on HPC Platforms: Peeking Under the Hood of Santos Dumont. In: <u>Journal of Parallel and Distributed Computing</u>, 2022 (*Submitted*).
- **CARNEIRO, A. R.**; SERPA, M. S.; NAVAUX, P. O. A. Lightweight Deep Learning Applications on AVX-512. In: <u>2021 IEEE Symposium on Computers and Communications</u> (ISCC), 2021.
- HERRERA, S.; RIBEIRO, W.; TEIXEIRA, T.; **CARNEIRO, A. R.**; CABRAL F.; BORGES, M.; OSTHOFF, C. Avaliação de Desempenho no Supercomputador SDumont de uma Estratégia de Decomposição de Domínio usando as Funcionalidades de Mapeamento Topológico do MPI para um Método Numérico de Escoamento de Fluidos. In: <u>Anais da VI Escola Regional de Alto Desempenho do Rio de Janeiro</u>, 2020.

References

- LUU, H. et al. A multiplatform study of i/o behavior on petascale supercomputers. In: Proceedings of the 24th International Symposium on High-Performance Parallel and Distributed Computing. New York, NY, USA: Association for Computing Machinery, 2015. (HPDC '15), p. 33–44. ISBN 9781450335508. Available from Internet: <https://doi.org/10.1145/2749246.2749269>.
- LOCKWOOD, G. K. et al. A year in the life of a parallel file system. In: SC18: International Conference for High Performance Computing, Networking, Storage and Analysis. [S.l.: s.n.], 2018. p. 931–943. Available from Internet: <https://doi.org/10.1109/SC.2018.00077>
- PATEL, T. et al. Revisiting i/o behavior in large-scale storage systems: The expected and the unexpected. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. New York, NY, USA: Association for Computing Machinery, 2019. (SC '19). ISBN 9781450362290. Available from Internet: <https://doi.org/10.1145/3295500.3356183>.
- SIVALINGAM, K. et al. Lassi: Metric based i/o analytics for hpc. In: 2019 Spring Simulation Conference (SpringSim). [s.n.], 2019. p. 1–12. Available from Internet: <https://doi.org/10.23919/SpringSim.2019.8732903>.
- BETKE, E.; KUNKEL, J. M. Footprinting parallel i/o – machine learning to classify application's i/o behavior. In: International Conference on High Performance Computing. [S.l.]: Springer International Publishing, 2019. p. 214–226.Available from Internet: <https://doi.org/10.1007/978-3-030-34356-9_18>.

ACKNOWLEDGMENT

This study was financed by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) - Finance Code 001.

This work has been partially supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

The authors acknowledge the National Laboratory for Scientific Computing (LNCC/MCTI, Brazil) for providing HPC resources of the SDumont supercomputer, which have contributed to the research results reported within this paper. URL: http://sdumont.lncc.br.

Uncovering I/O Usage in HPC Platforms

André Ramos Carneiro

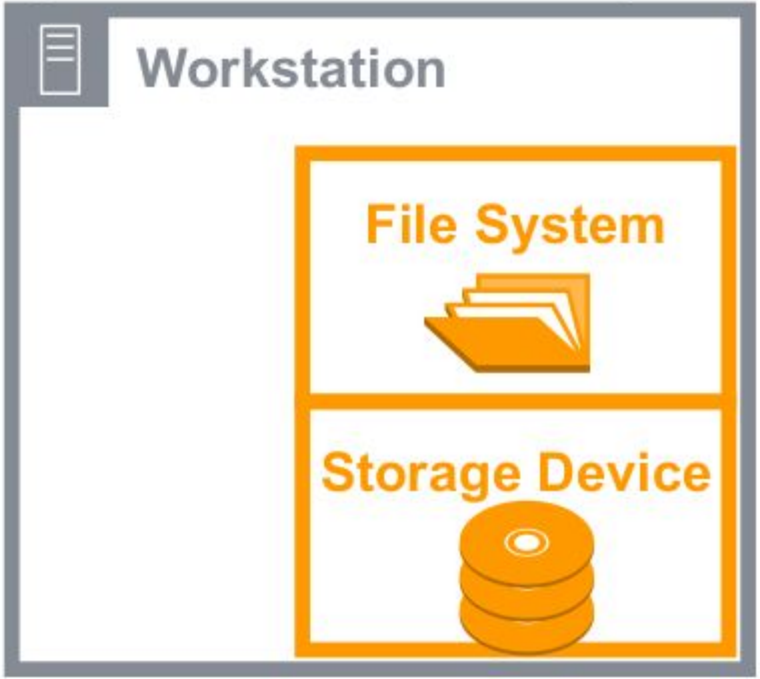
Advisor: Prof. Dr. Philippe O. A. Navaux
Co-advisor: Prof. Dr. Carla Osthoff

Instituto de Informática,
UFRGS, Brasil

L N C C

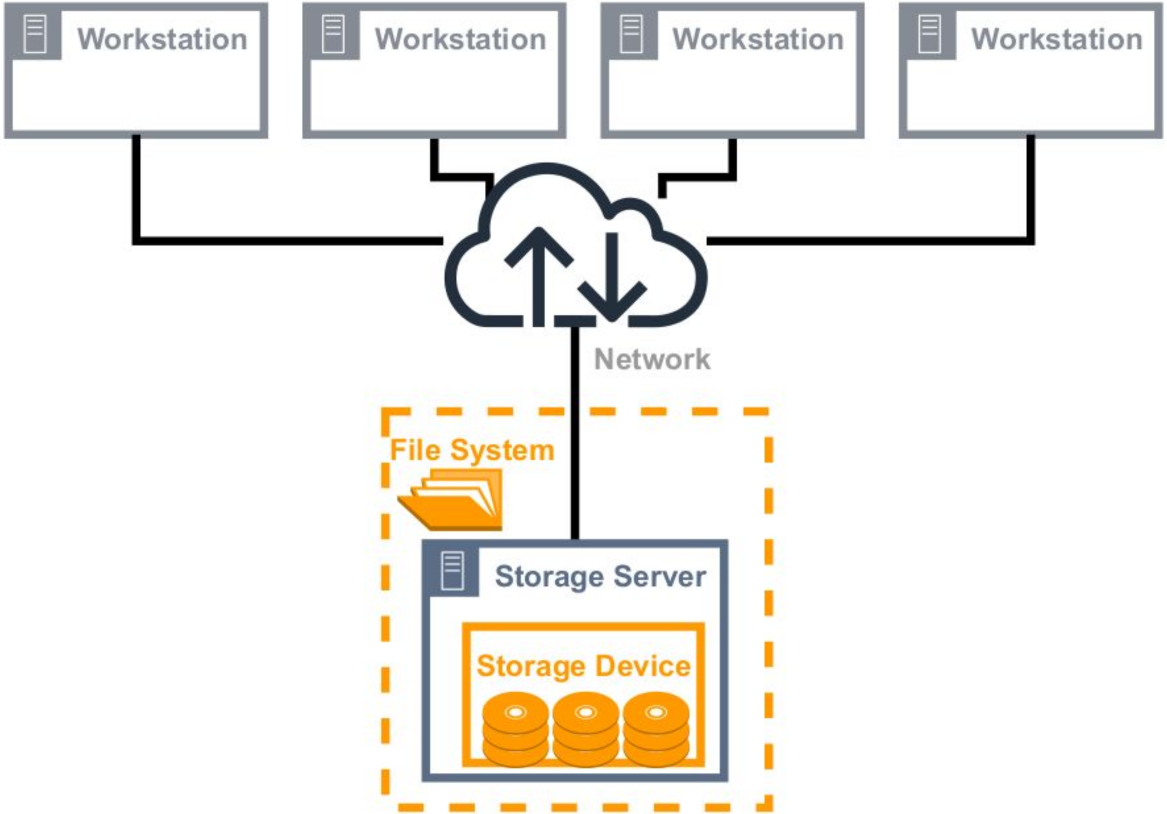
.inf INSTITUTO DE INFORMÁTICA UFRGS

Figure 2.1 – Local File System.



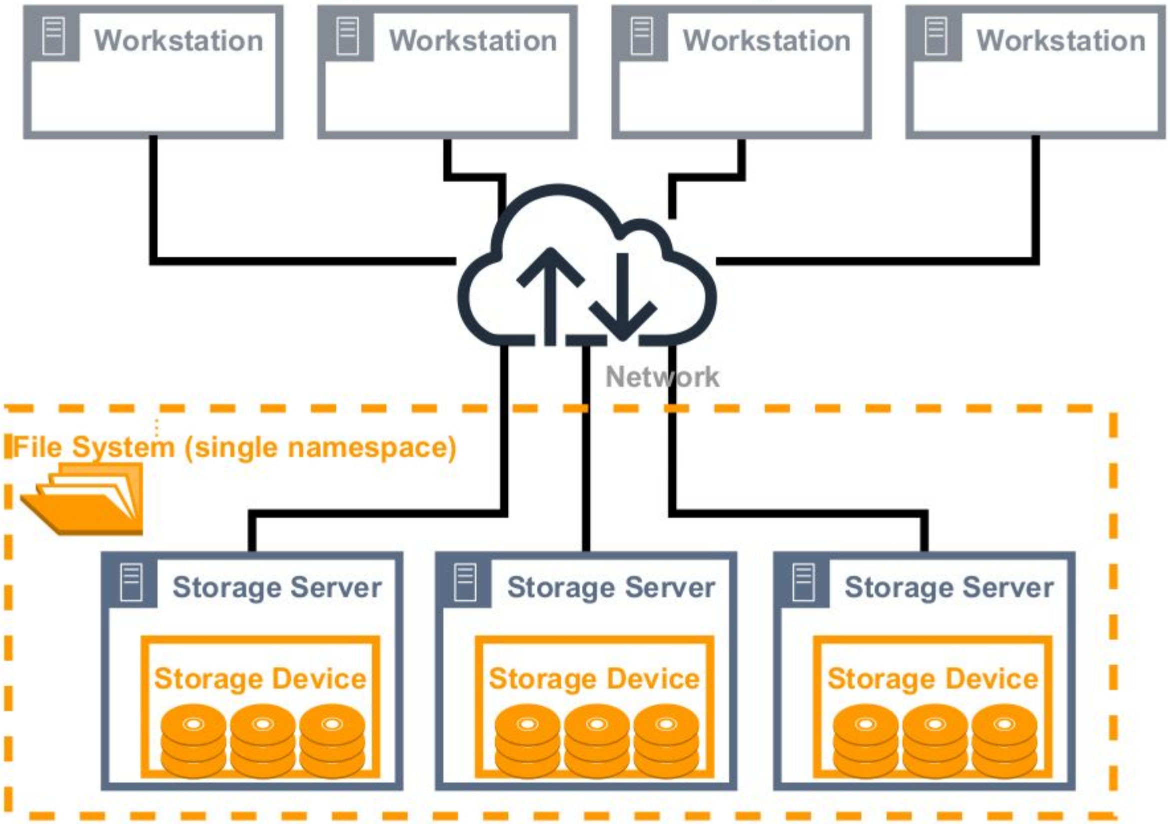
Source: Author

Figure 2.2 – Networked File System.



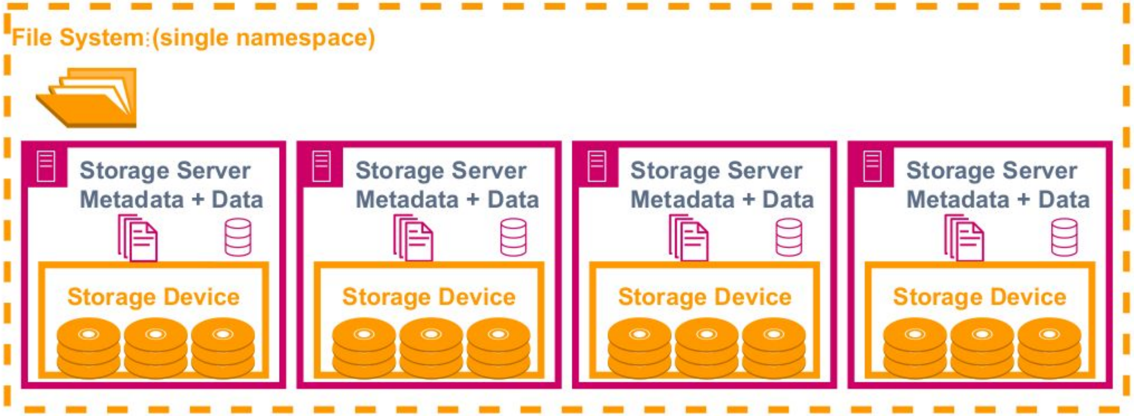
Source: Author

Figure 2.3 – Parallel File System.

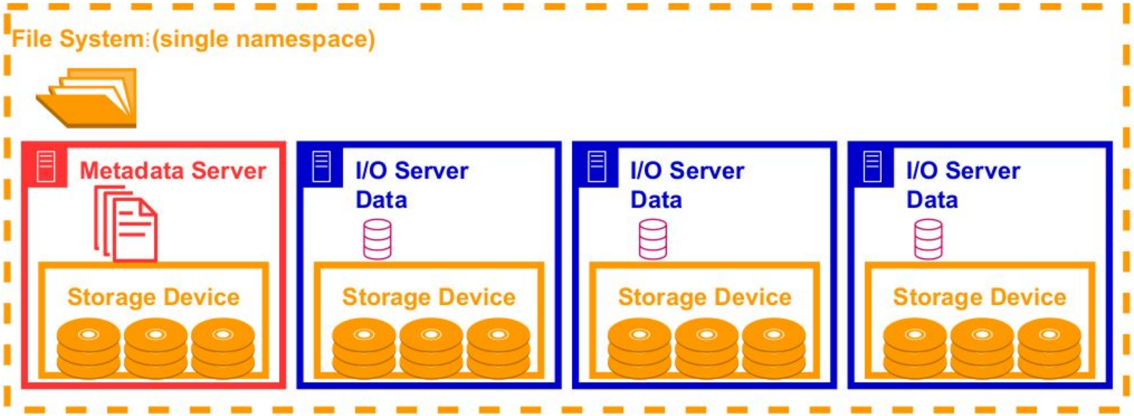


Source: Author

Figure 2.4 – Metadata Management Representation.

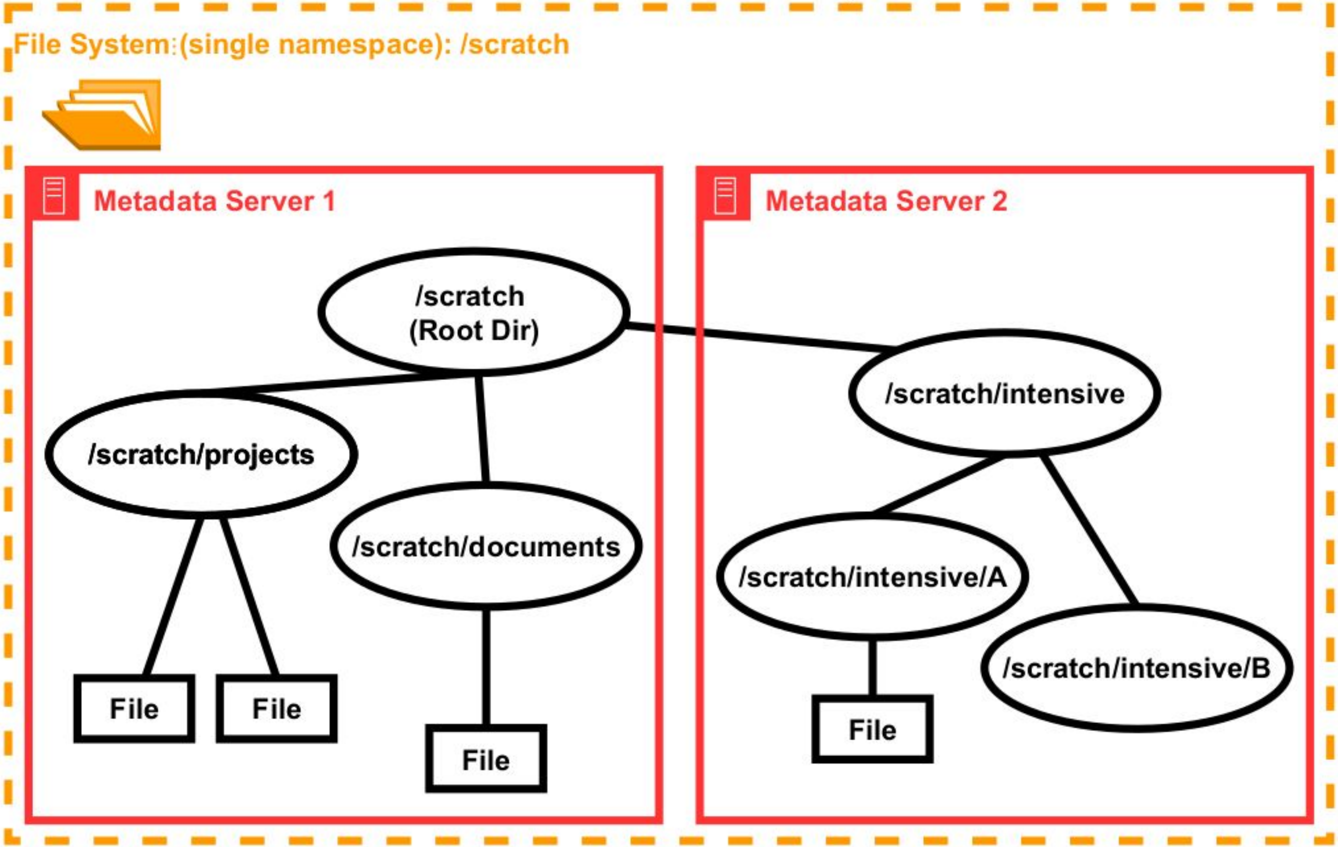


(a) Decentralized



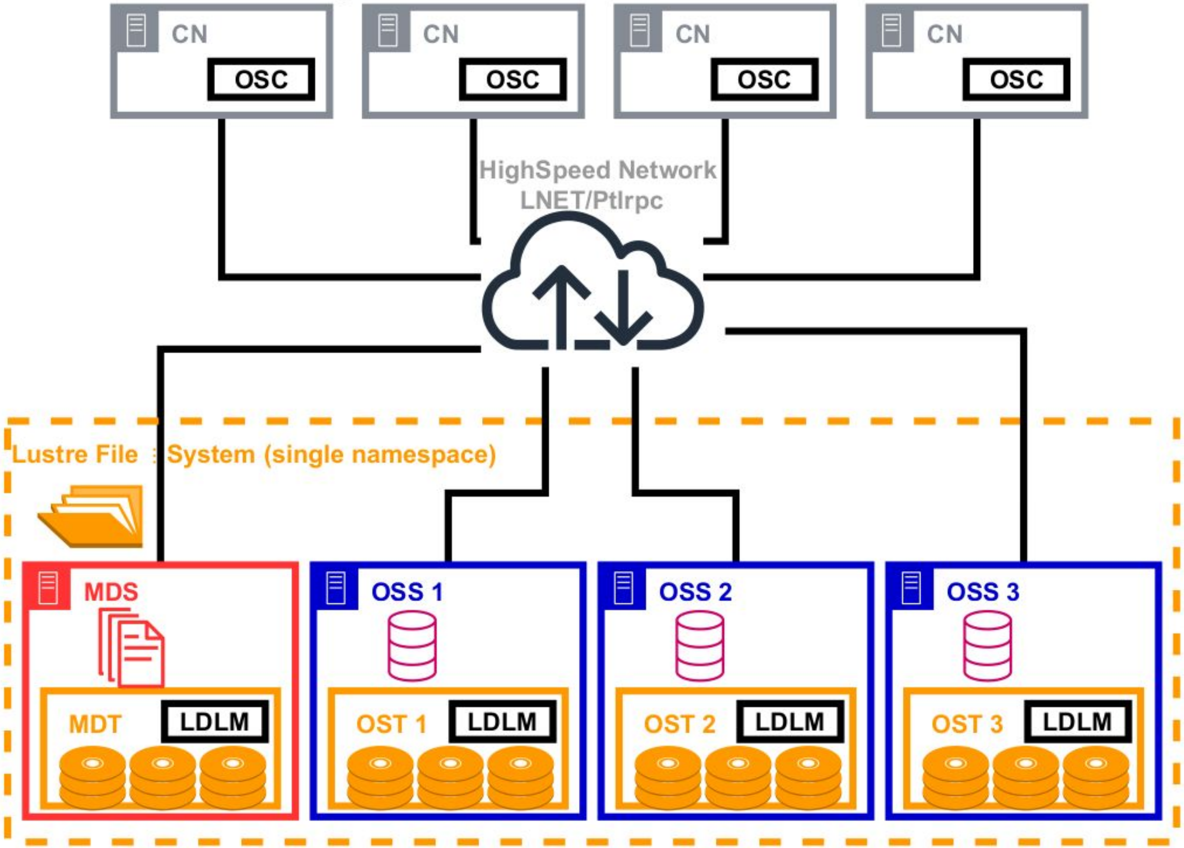
(b) Centralized

Figure 2.5 – Parallel File System.



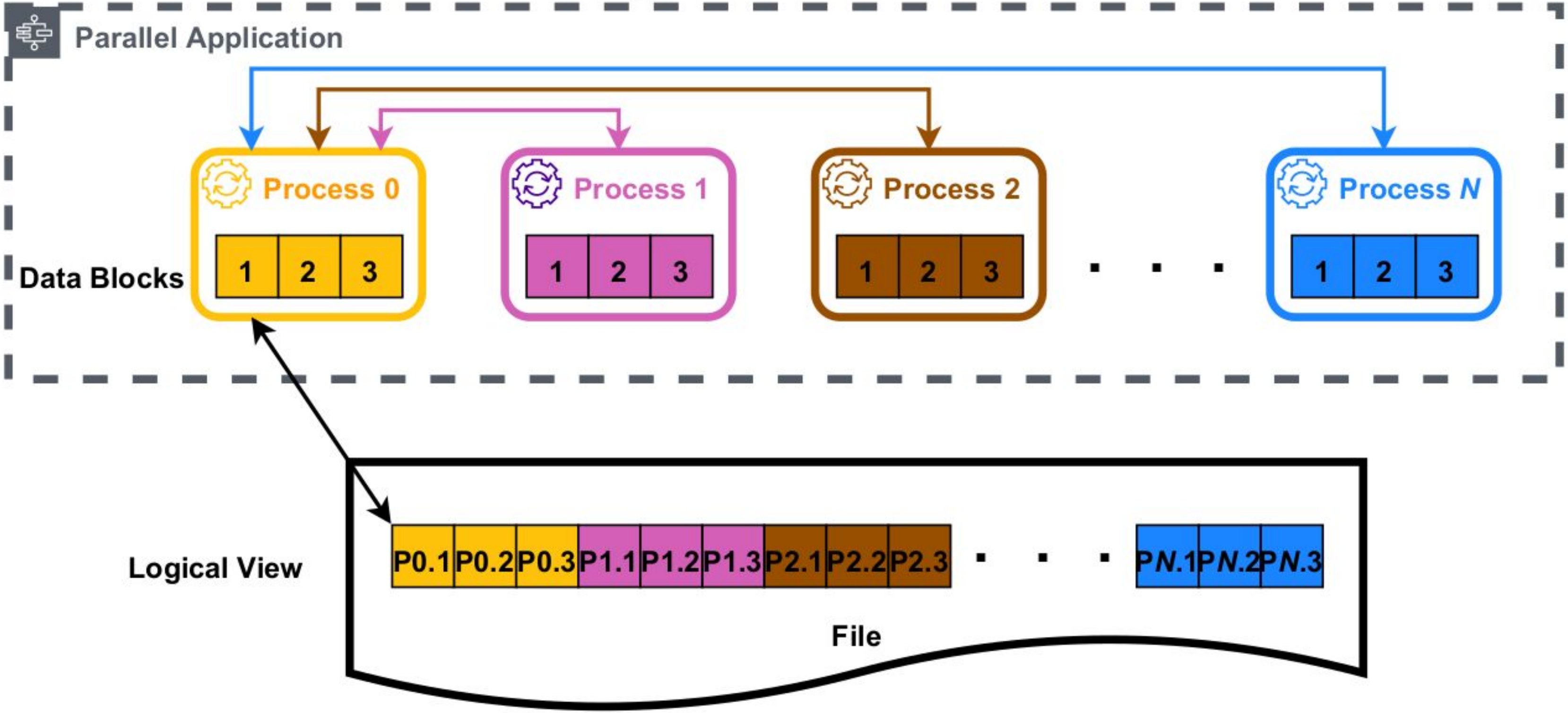
Source: Author

Figure 2.6 – Lustre PFS Architecture.



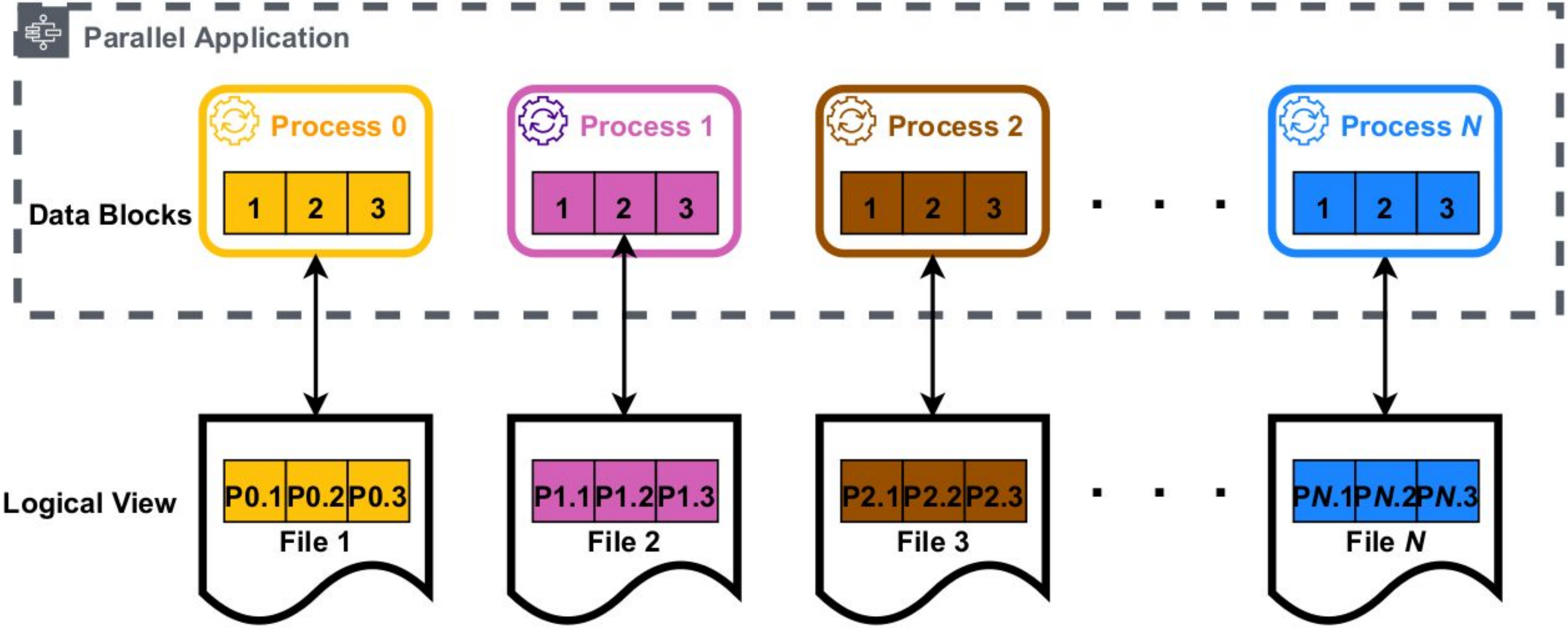
Source: Author

Figure 3.1 – Serial I/O.



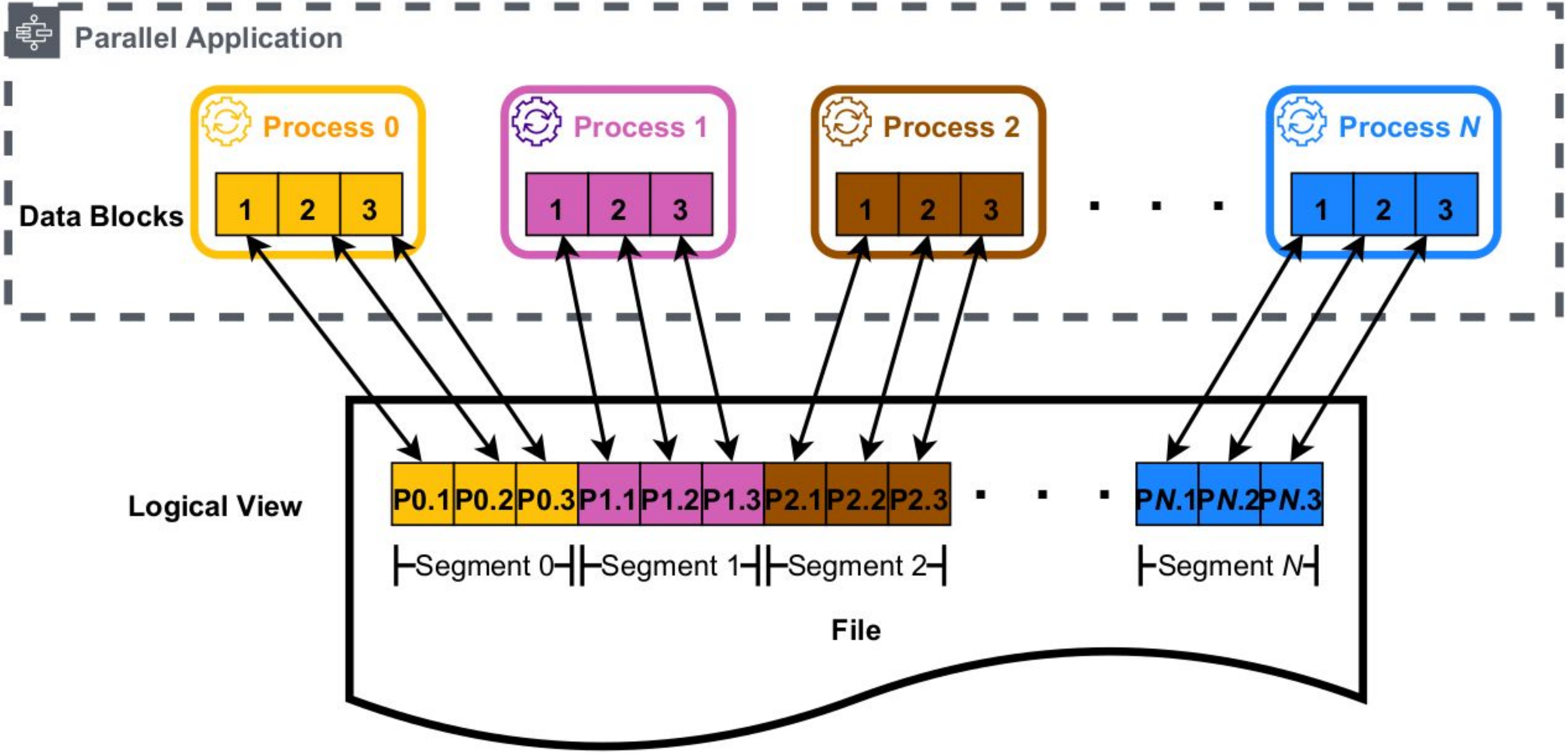
Source: Author, inspired by Ching et al. (2007)

Figure 3.2 – Parallel I/O - File-Per-Process.

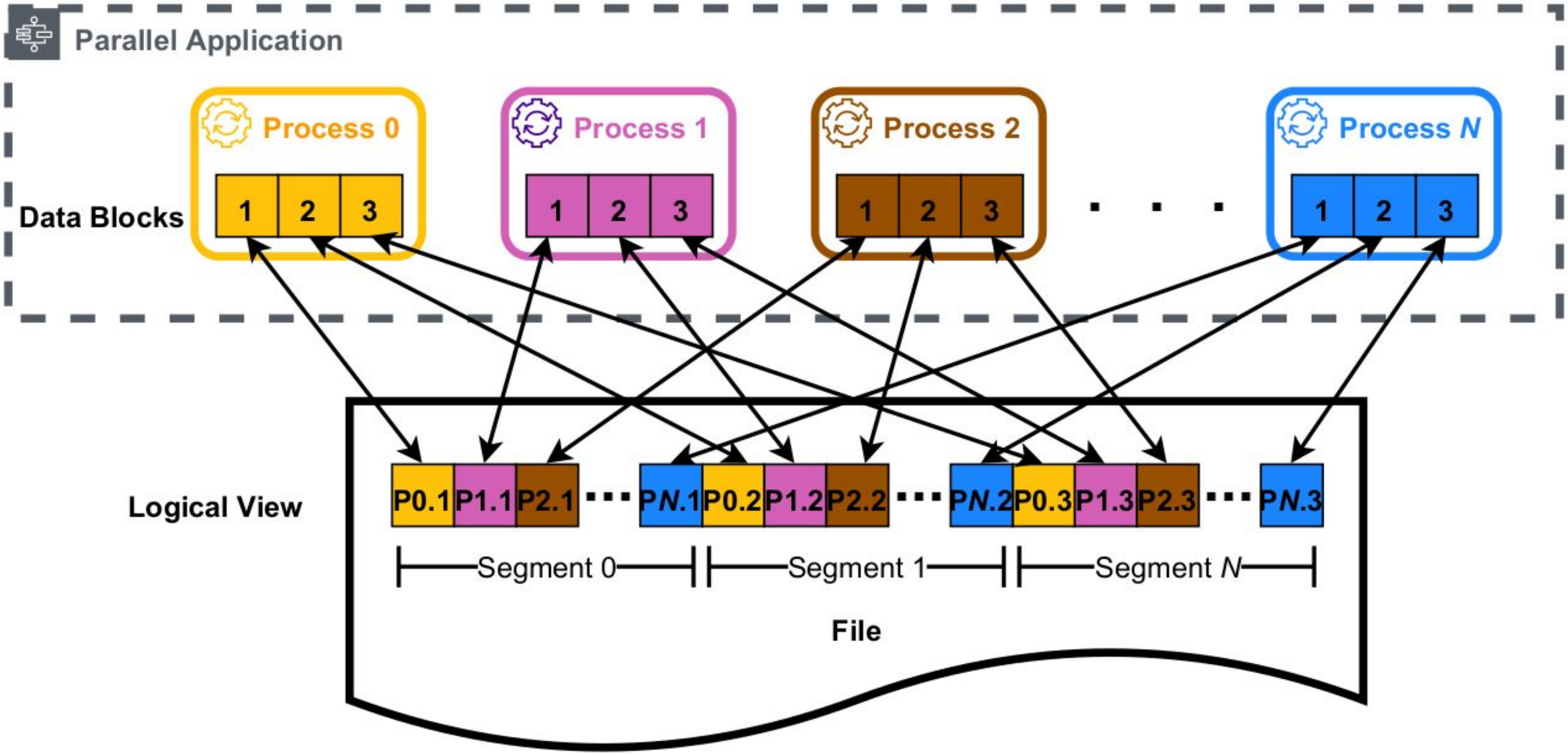


Source: Author, inspired by Ching et al. (2007)

Figure 3.3 – Parallel I/O - Shared-File.

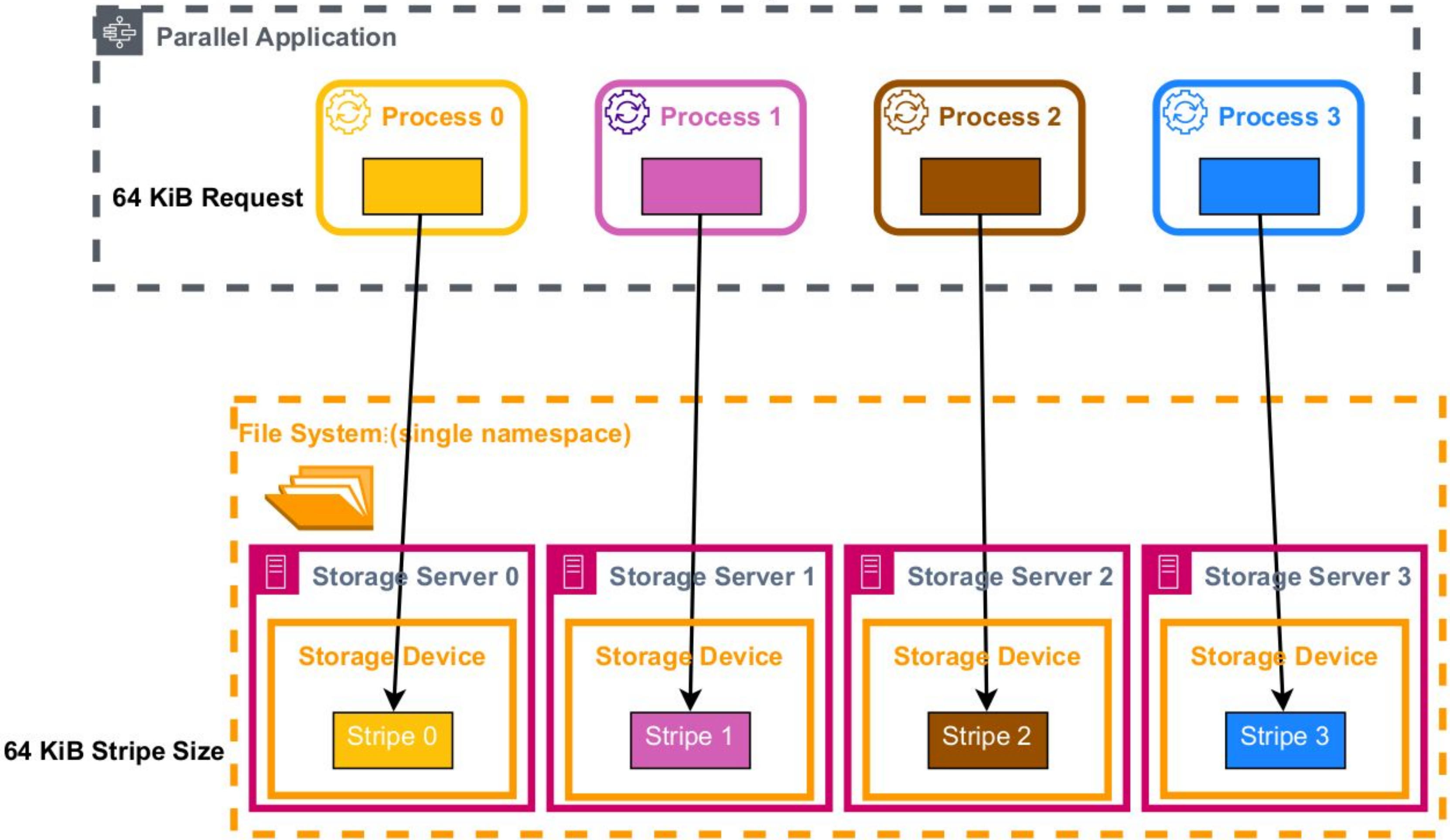


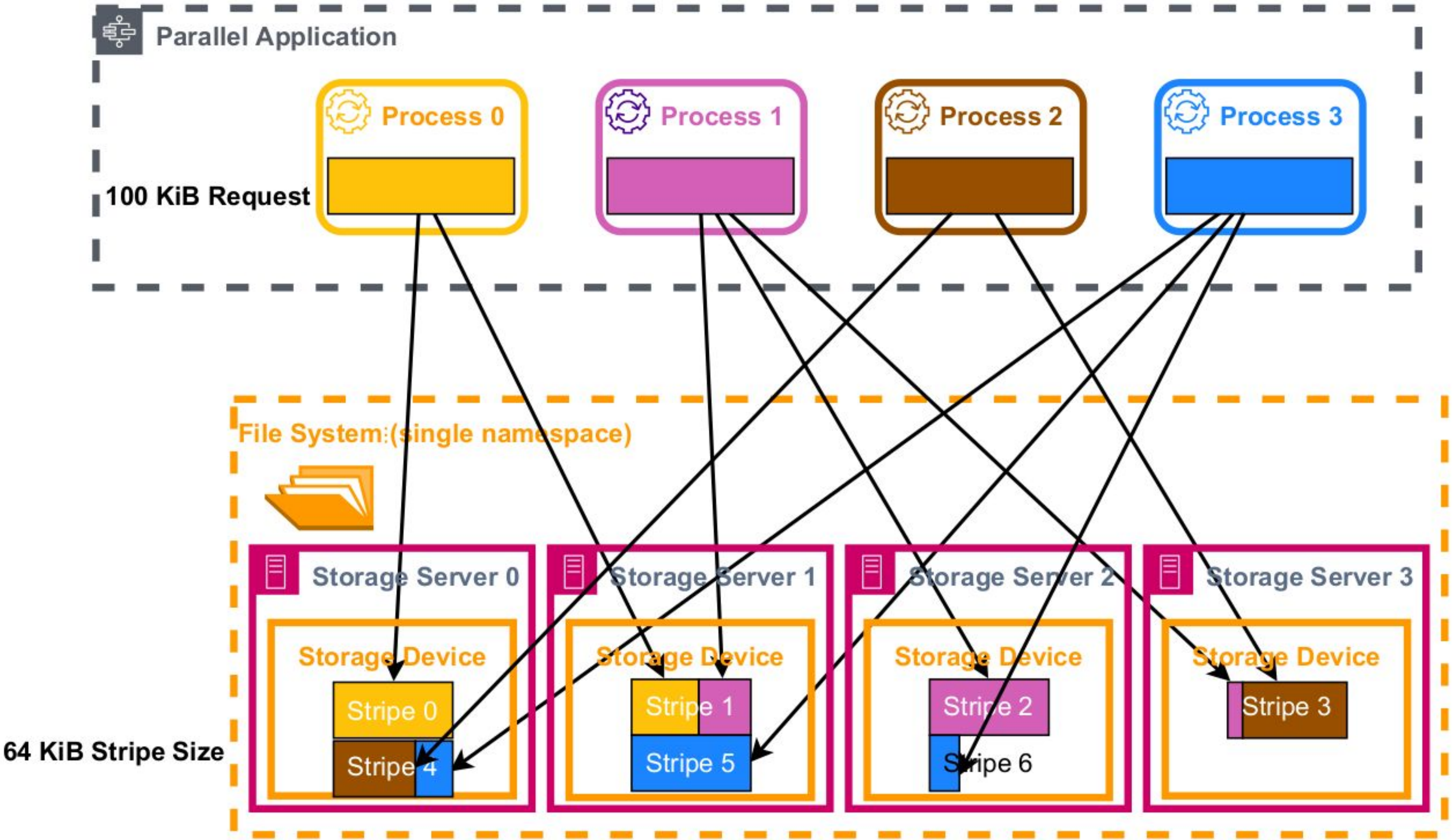
(a) Contiguous Access



(b) Strided Access

Figure 3.4 – Stripe Access.





(b) Misaligned Access

Growth in Max Score per Client
IO500 List

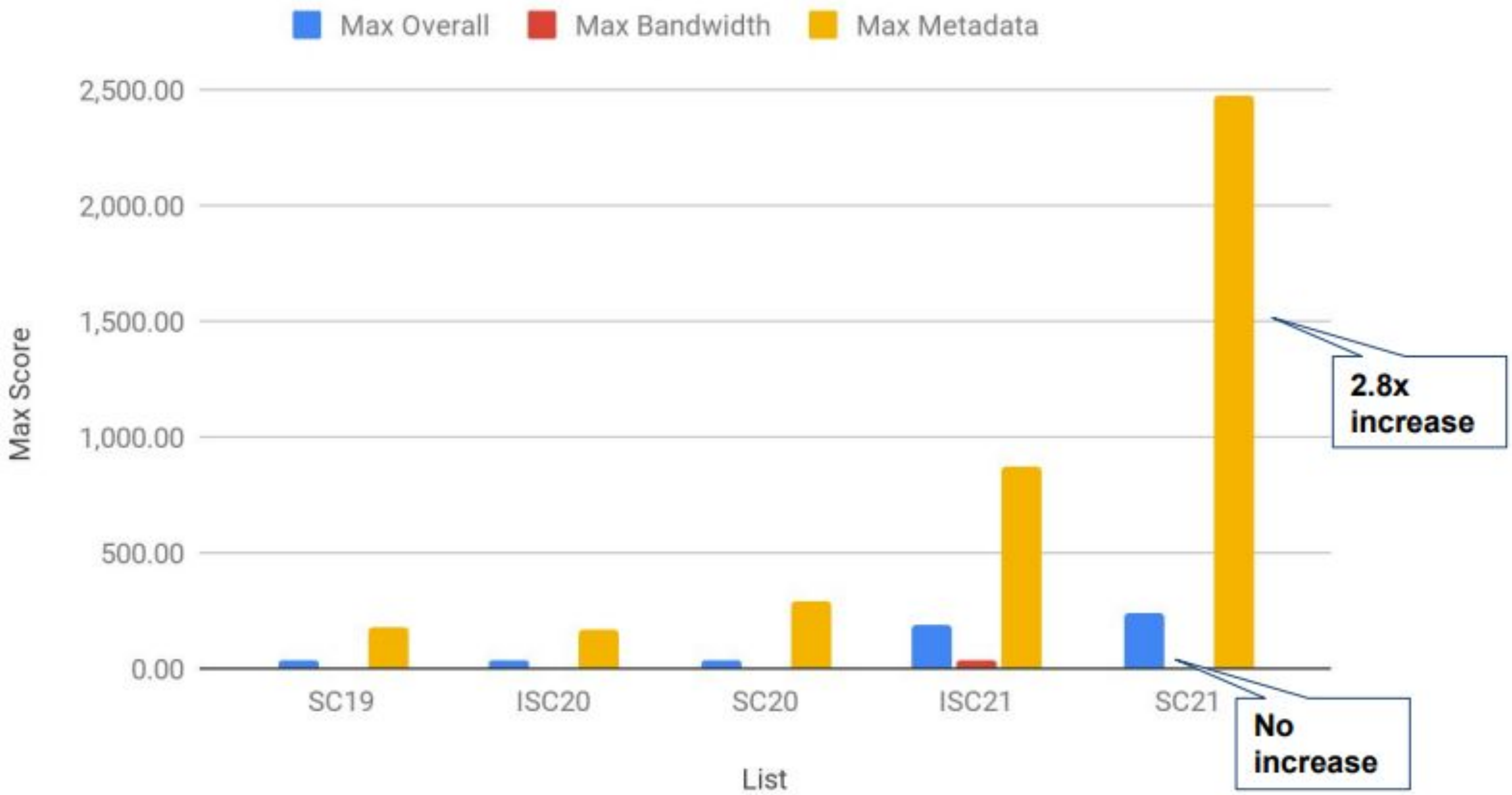


Table 5.2 – Amount of Metadata Operations

Operation	Total	Min ops/s	Avg. ops/s	Max. ops/s
fopen	28,812,381,450	1	3,859	102,291
fclose	25,369,943,340	1	3,398	102,132
getattr	6,733,374,960	1	902	32,698
setattr	3,451,979,850	1	462	8,406
unlink	593,117,055	1	87	2,357
getxattr	345,187,575	1	47	7,833
statfs	280,998,450	1	38	62
sync	125,075,625	1	76	1,618
mkdir	94,034,205	1	14	1,228
rmdir	41,638,320	1	34	1,041
setxattr	4,354,485	1	83	1,061
link	1,649,205	1	139	2,357

Source: Author

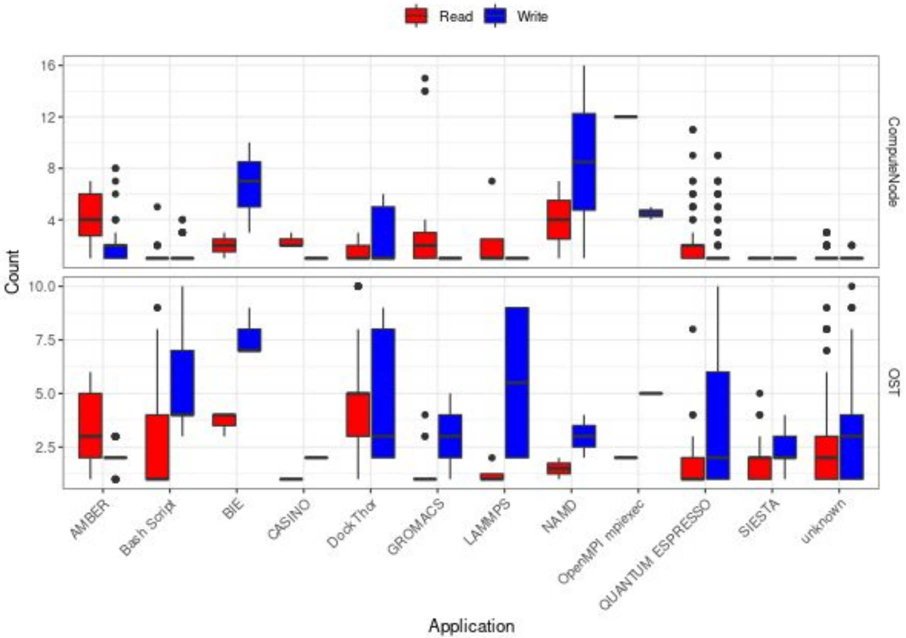
Results - Detailed View of a Region of Interest
I/O - Compute Nodes

Table 5.4 – Average Data Transfer per Job

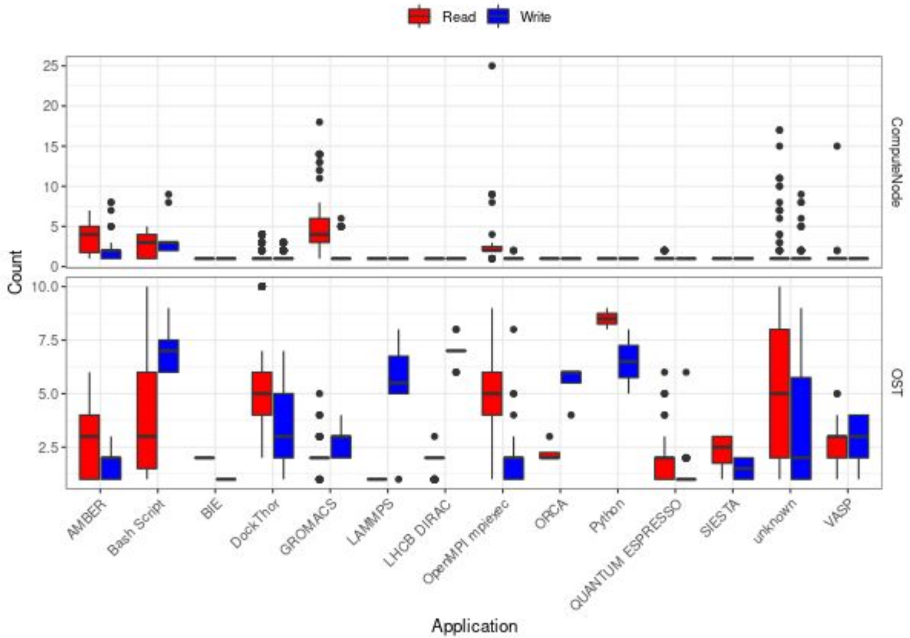
Application	**Read (GiB)**	**Write (GiB)**
unknown	5,394	23
BIE	793	60
OpenMPI mpiexec	95	42
AMBER	3	44
QUANTUM ESPRESSO	2	22

Source: Author

Results - Trimester Lustre Usage Analysis I/O - OSS Nodes



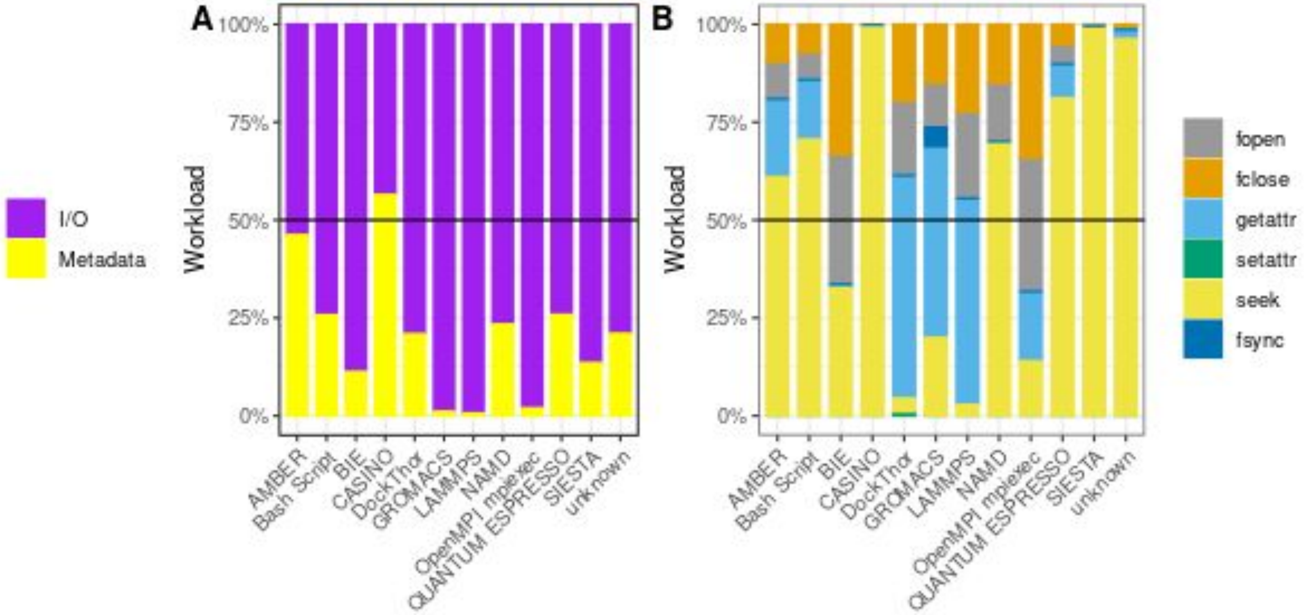
2020



2021

Distribution of the Simultaneous Resource Used by each application in readv(red) and write (blue).

Results - Detailed View of a Region of Interest
Metadata - Compute Nodes



2020 applications' metadata load distribution. (A) presents the load division between I/O (purple) and metadata operations (yellow). (B) presents the division among each metadata operation type.