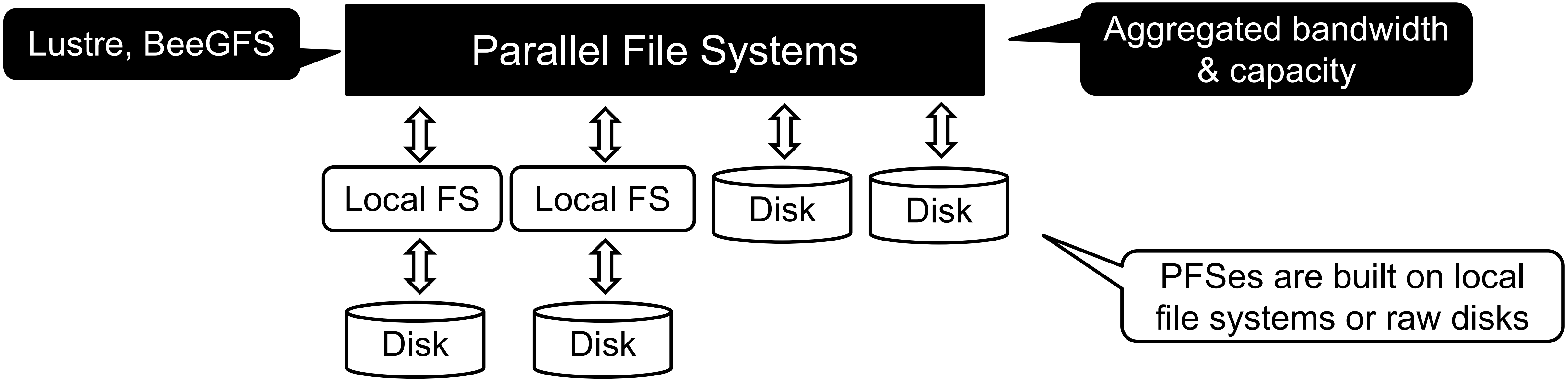
Pinpointing Crash-Consistency Bugs in the HPC I/O Stack: A Cross-Layer Approach

Jinghan Sun, Jian Huang, Marc Snir

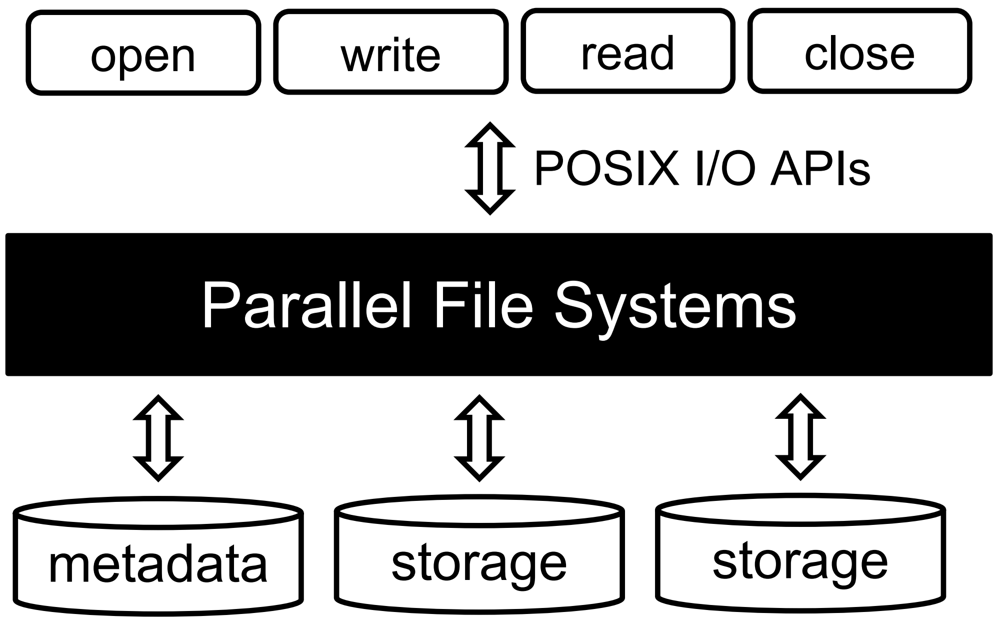
University of Illinois Urbana-Champaign

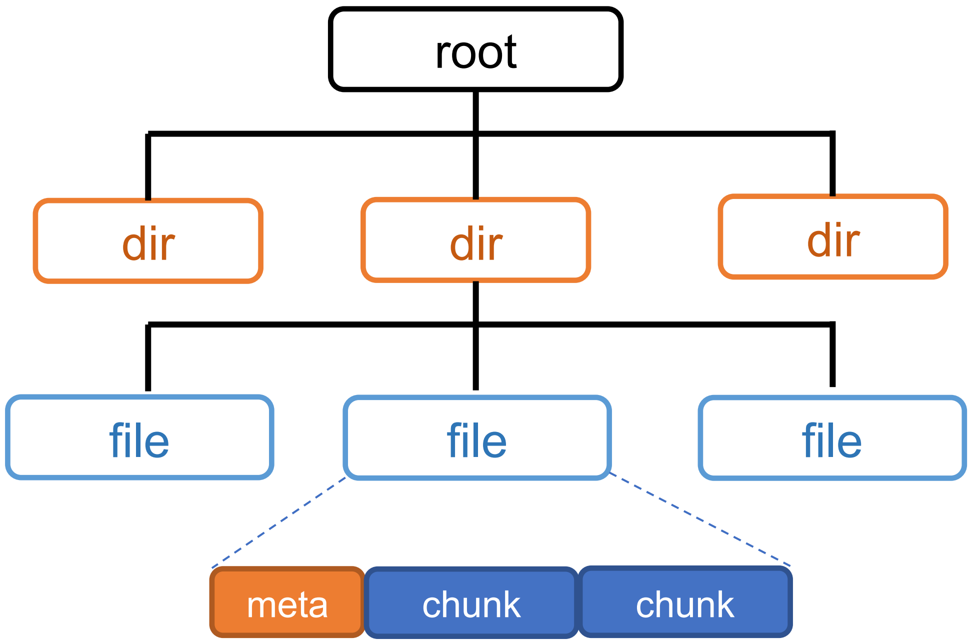
UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

The Complexity of Modern HPC I/O Stack



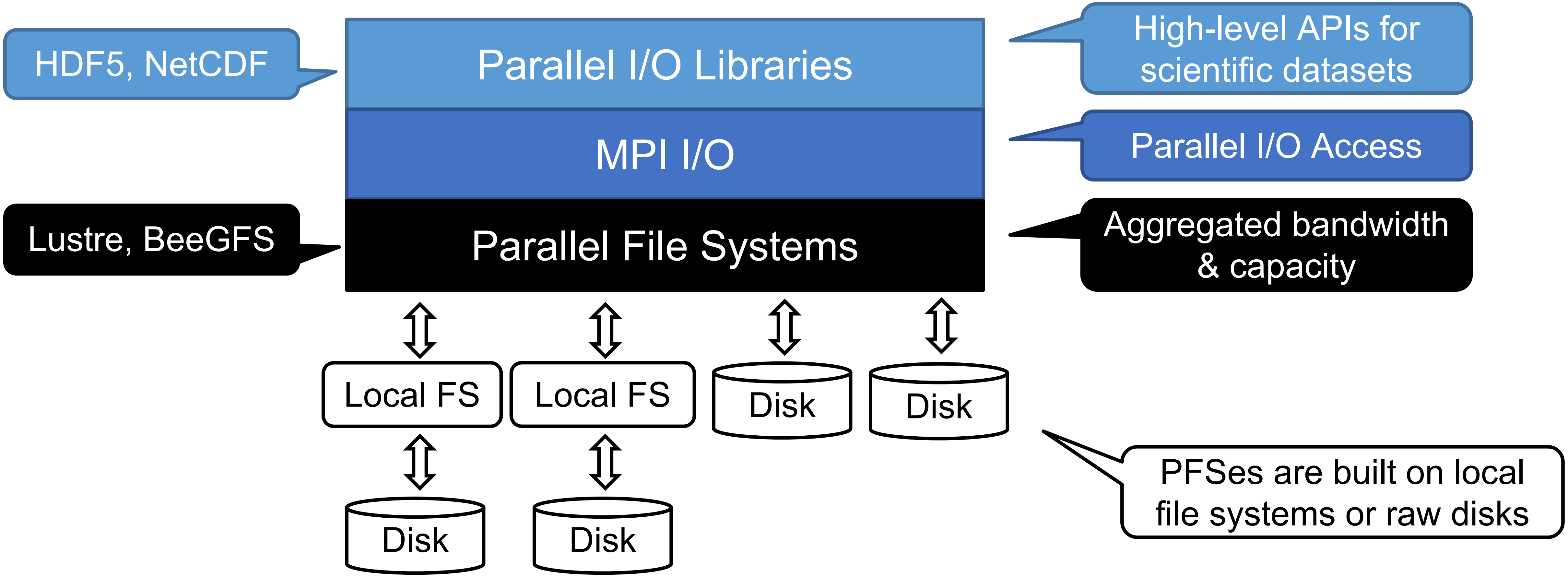
The Core Components of the HPC I/O Stack – I/O Library



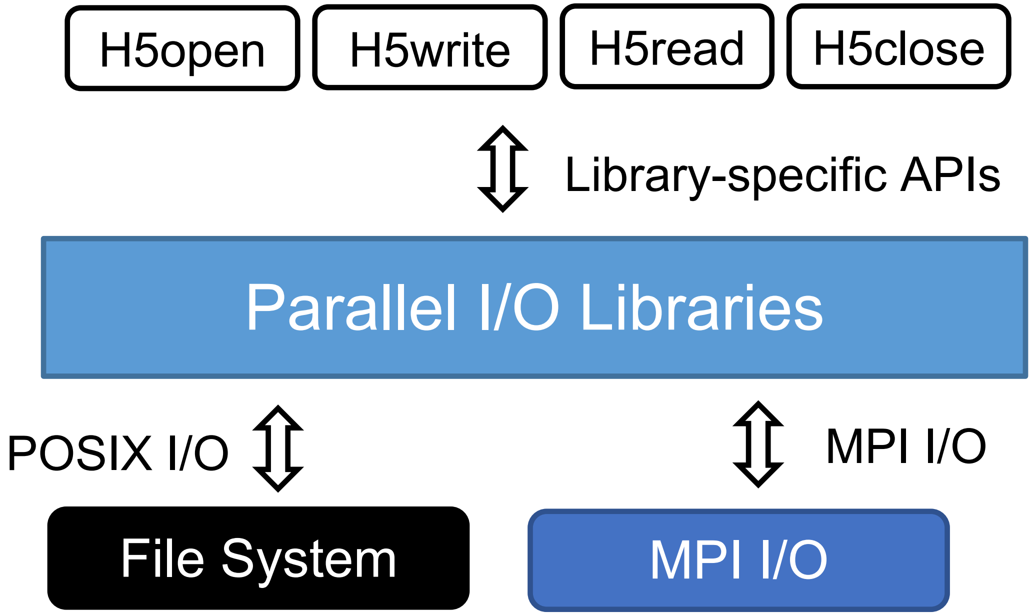


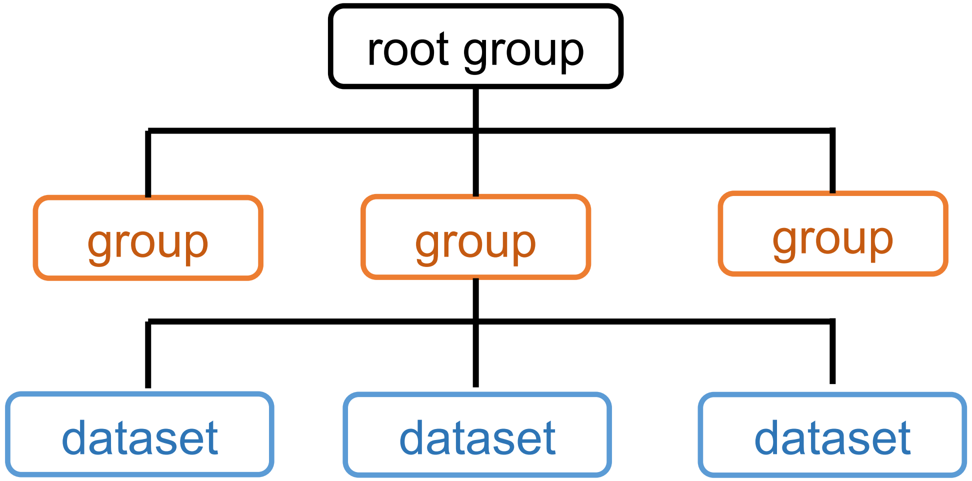
PFS stripes data across servers to provide high bandwidth and capacity

The Complexity of Modern HPC I/O Stack



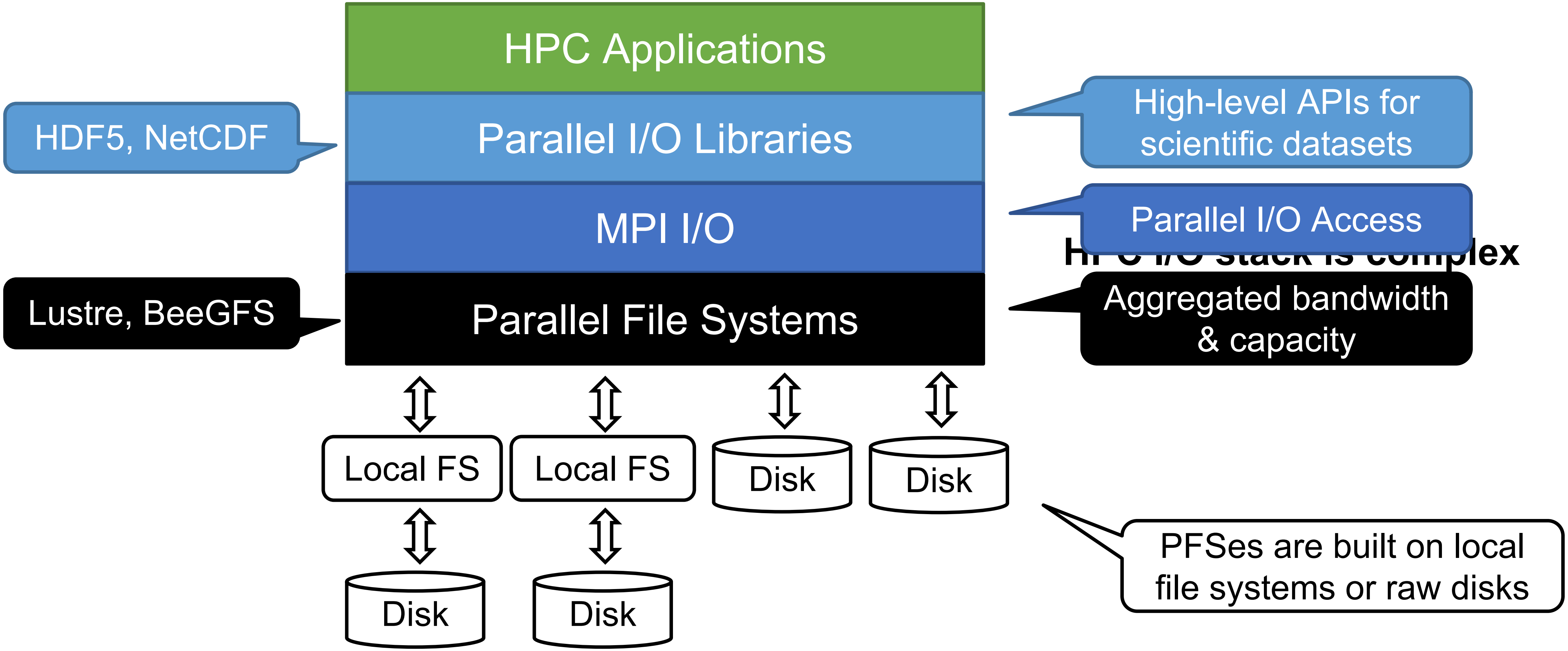
The Core Components of the HPC I/O Stack – I/O Library



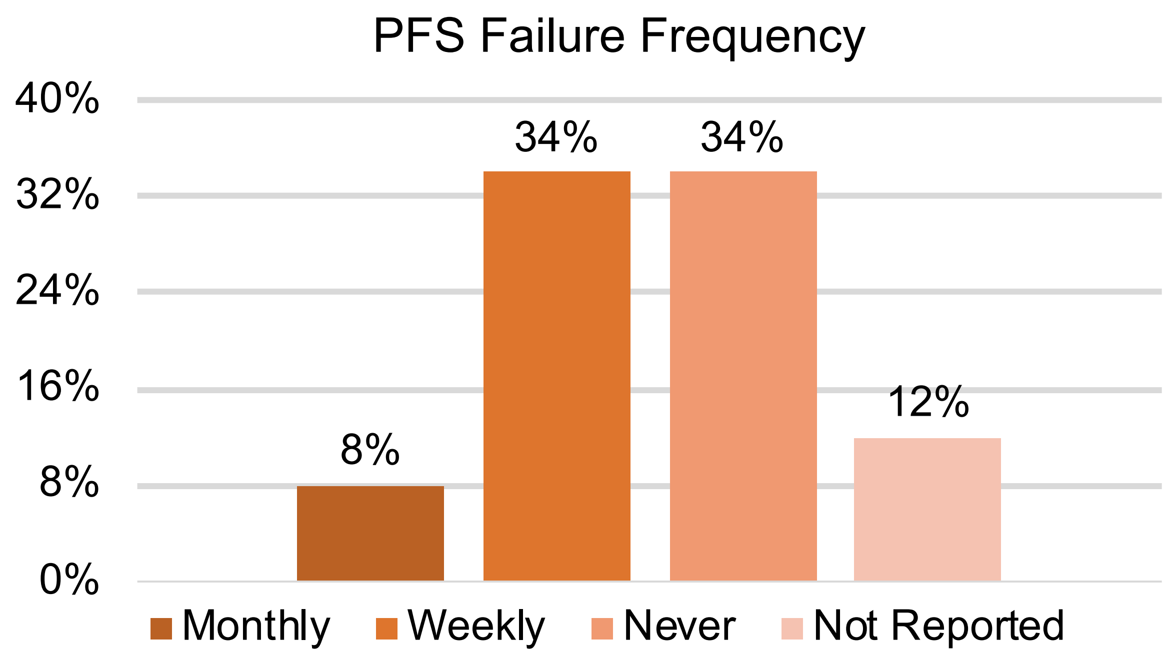


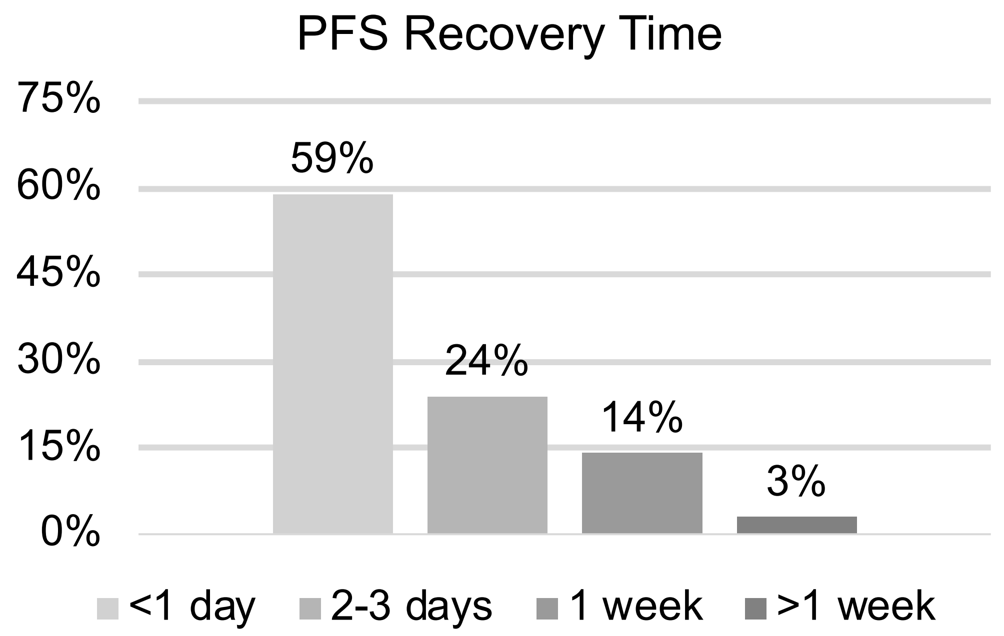
Parallel I/O library organizes scientific datasets into a similar hierarchical structure

The Complexity of Modern HPC I/O Stack



HPC I/O Stack Fails Frequently And Takes Long to Recover





42% of PFSes suffer from monthly or weekly failures and their recovery is time-consuming

Source: Hyperion Research 2019

Severe Errors Could Occur After HPC I/O Failures

Subject: Update: HPCC Power Outage
Date: Monday, January 11, 2016 at 8:50:17 AM Central Standard Time
From: HPCC - Support

To All HPCC Customers and Partners,

As we have informed you earlier, the Experimental Sciences Building experienced a major power outage Sunday, Jan. 3 and another set of outages Tuesday, Jan. 5 that occurred while file systems were being recovered from the first outage. As a result, there were major losses of important parts of the file systems for the work, scratch and certain experimental group special Lustre areas.

The HPCC staff have been working continuously since these events on recovery procedures to try to restore as much as possible of the affected file systems. These procedures are extremely time-consuming, taking days to complete in some cases. Although about a third of the affected file systems have been recovered, work continues on this effort and no time estimate is possible at present.

Recovering from power loss and timeline for journaling
Hdf-forum archives

HDF5 file state in case of crash
Hdf-forum archives

Recover a corrupt HDF5 file
Hdf-forum archives

Corrupt files when creating HDF5 files without closing them (h5py)

Metadata loss during Lustre recovery

Dataset corruption after HDF5 crashes

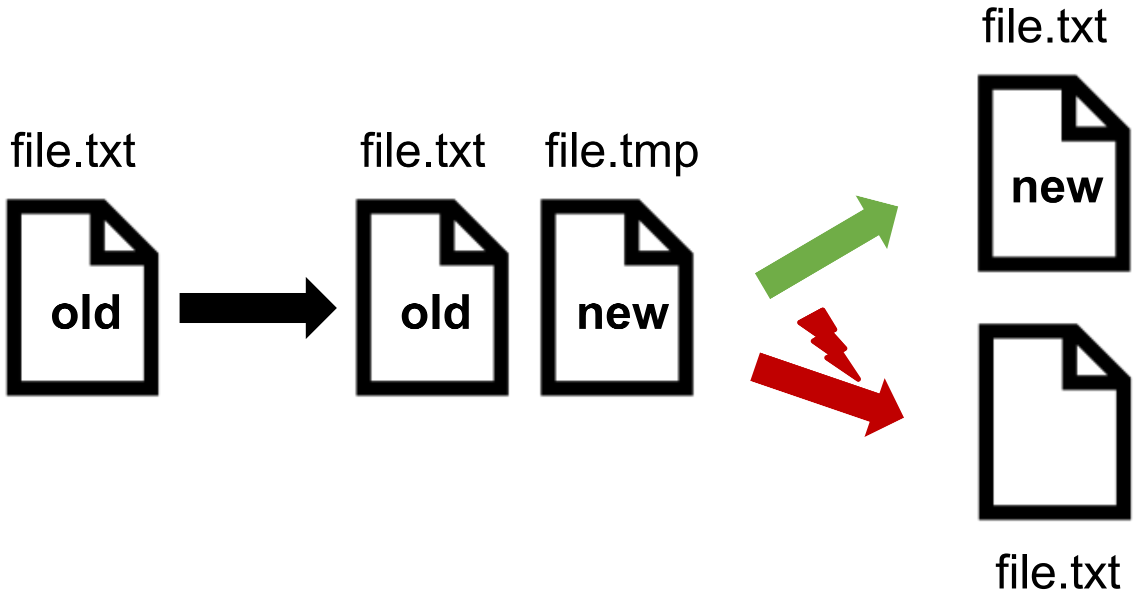
It is important to have a bug detection framework for the HPC I/O stack!

Crash Consistency of the HPC I/O Stack

A crash consistency bug is an unrecoverable storage state error after a system crash

```
void init(){
        int fd = open("file.txt", ...);
        write(fd, “old content”, size);
        close(fd);
}
// atomic replace via rename (ARVR)
bool atomic_update(){
        int fd = creat("file.tmp");
        write(fd, “new content”, size);
        close(fd);
        rename("file.tmp","file.txt");
}
```

Atomic replace via rename (ARVR)



Violation of rename atomicity may cause data loss

Existing Tools Are Insufficient For the HPC I/O Stack

Existing frameworks for crash consistency detection

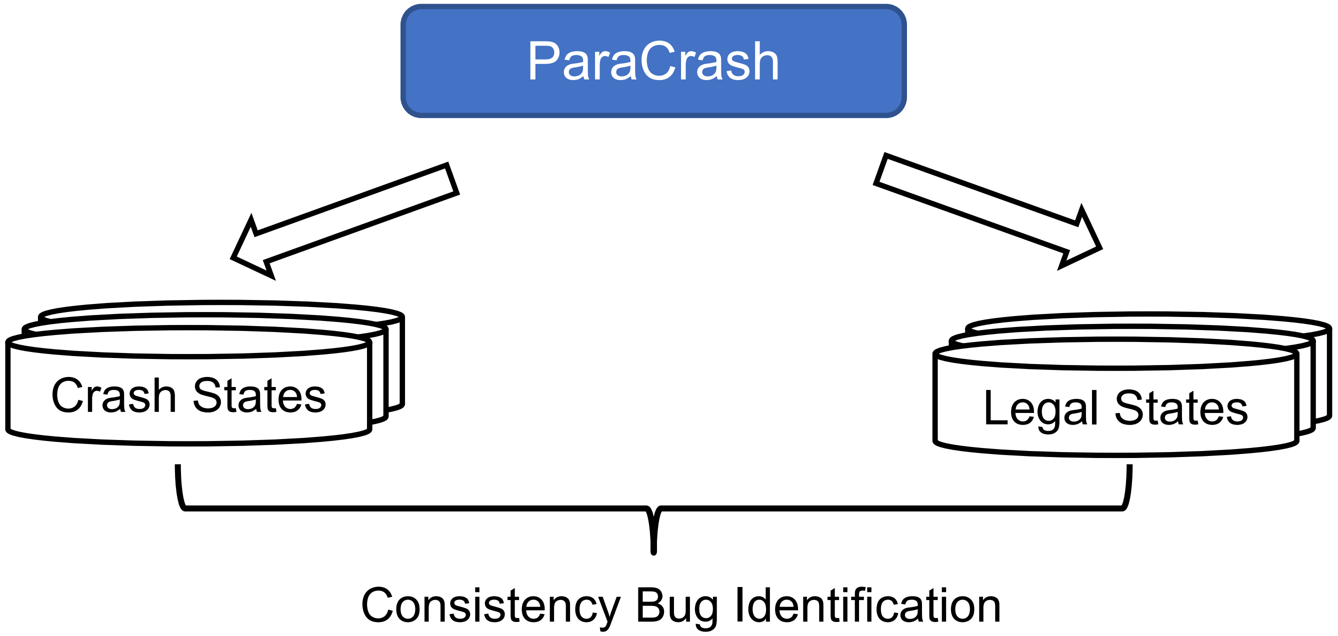
- Local file system: **CrashMonkey** and **ACE**
- Distributed database: **PACE**
- Application-level: **ALICE**
- Model checking: **FERRITE**

They lack support to parallel programs and do no handle multi-layered storage

ParaCrash

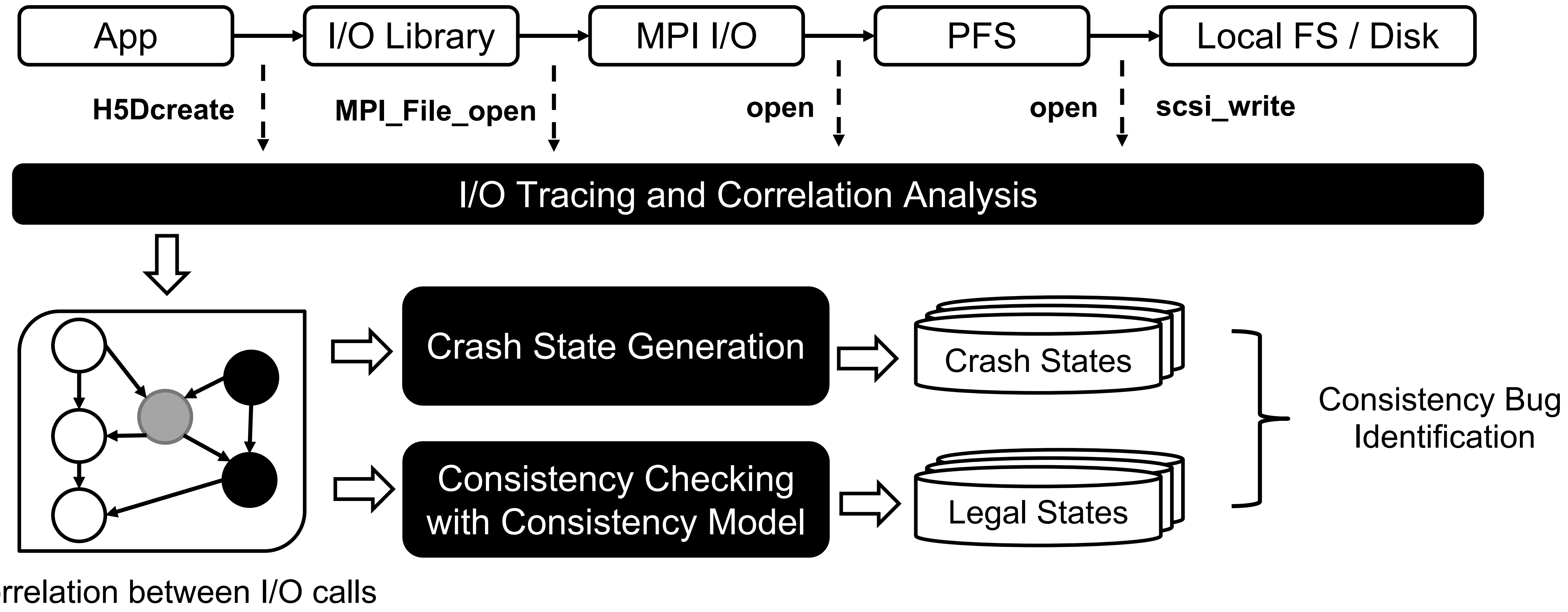
A new framework for detecting crash-consistency bugs in the HPC I/O Stack

ParaCrash uses the golden master testing approach

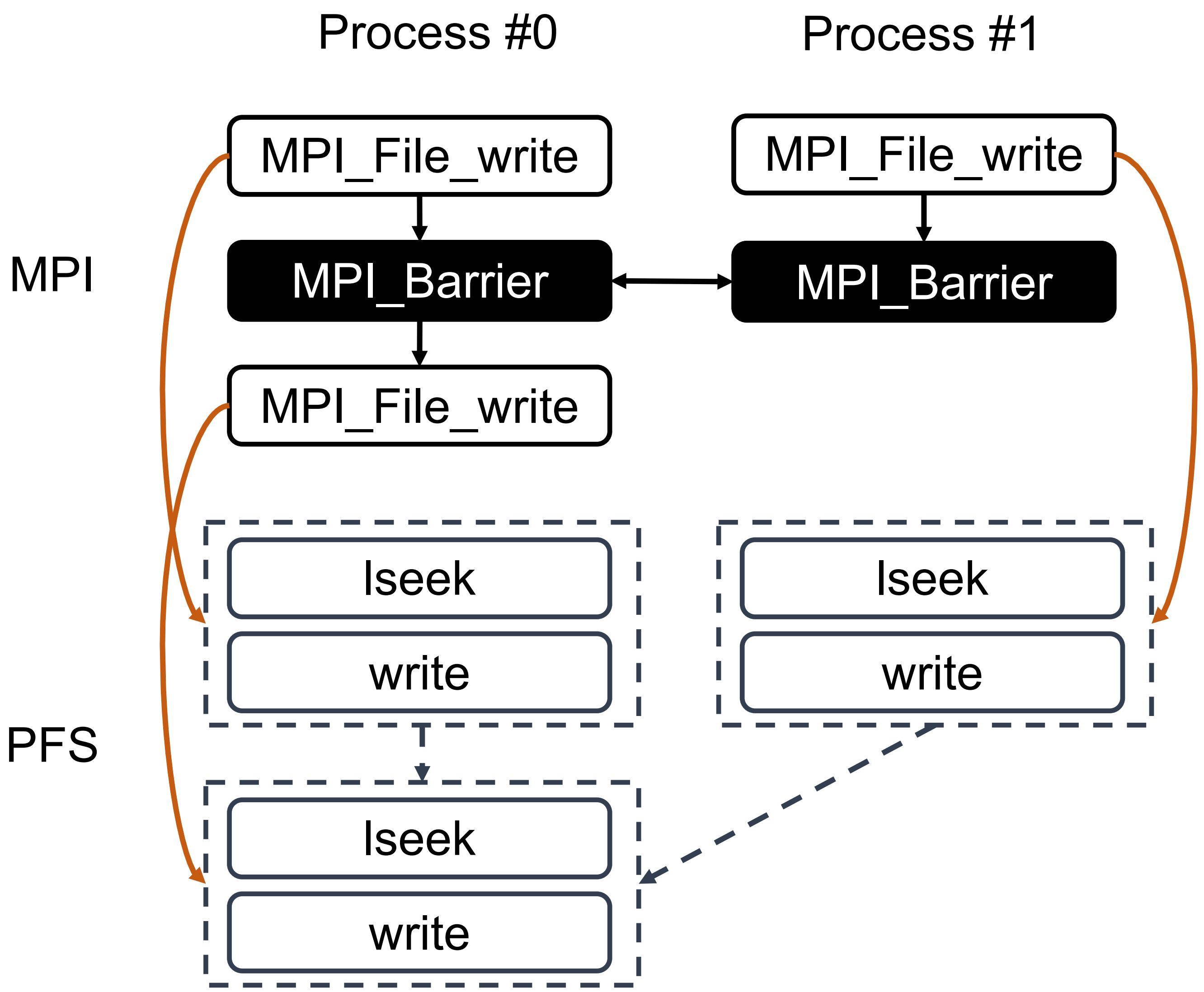


ParaCrash

A new framework for detecting crash-consistency bugs in the HPC I/O Stack



Analyzing Partial Order of Operation Execution



Trace I/O calls and communications

+

Identify synchronization operations

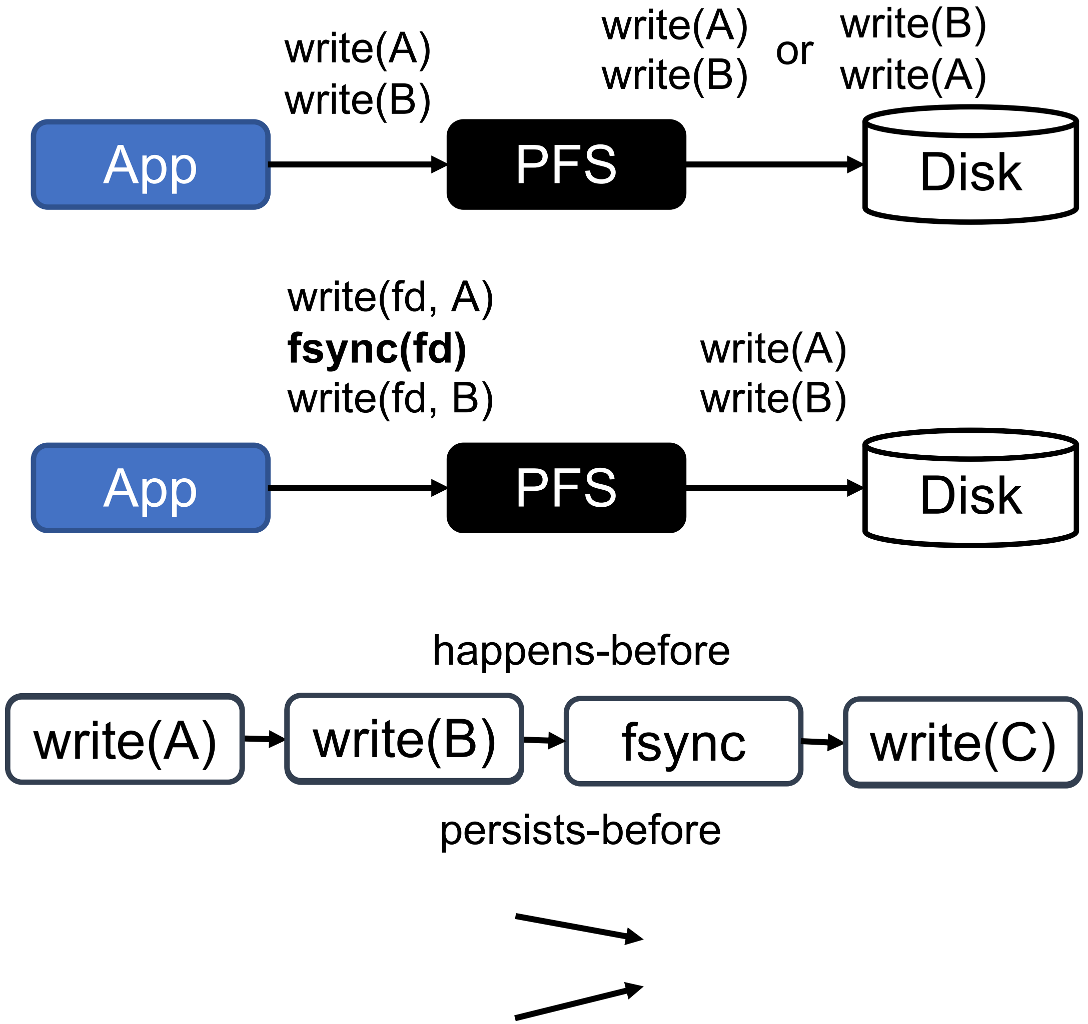
+

Identify caller-callee relationships

+

Build happens-before graph

From Execution Orderings to Possible Persistency Orderings



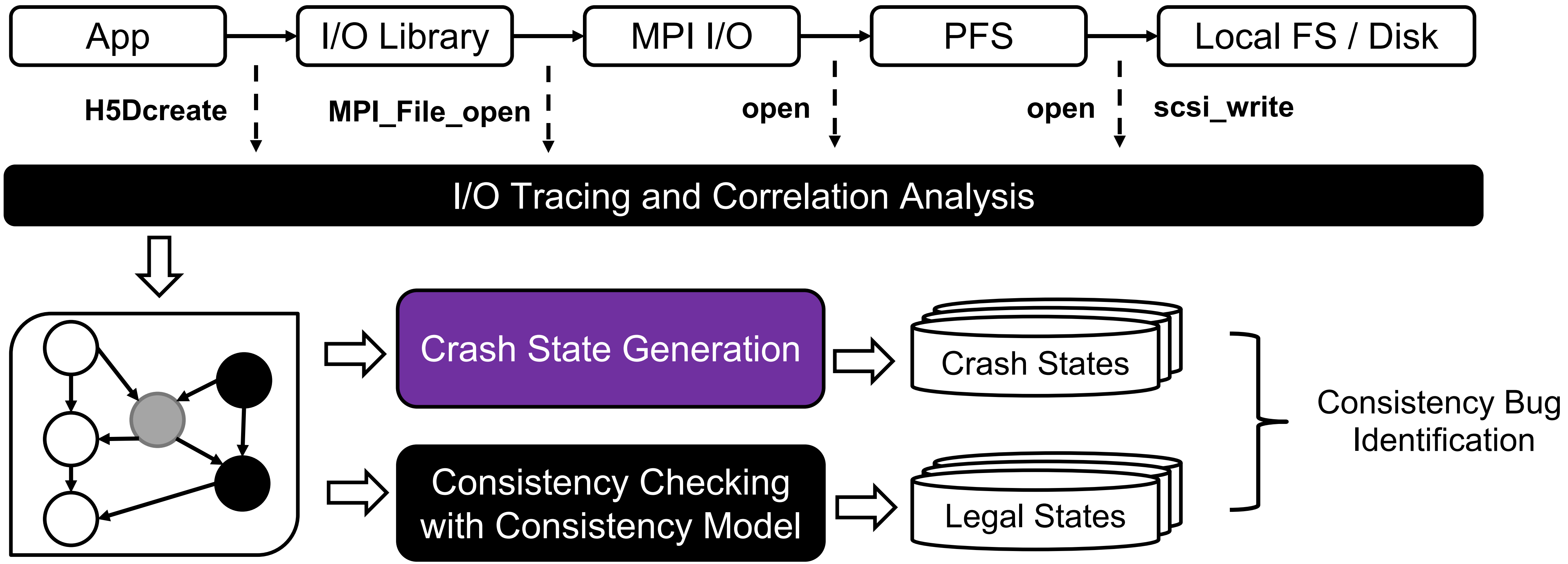
I/O calls can be persisted to disks different from their execution order

Persistency are enforced by fsync & barriers

Induce persists-before from happens-before

ParaCrash

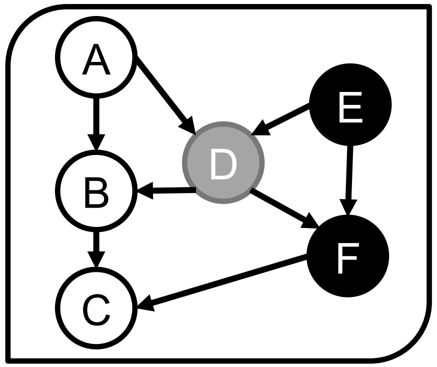
A new framework for detecting crash-consistency bugs in the HPC I/O Stack



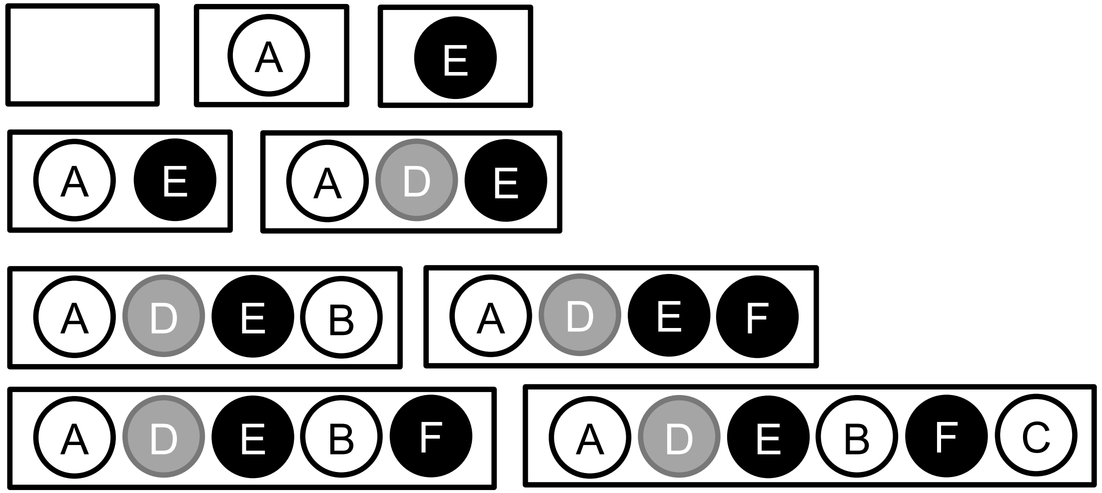
correlation between I/O calls

Crash State Generation

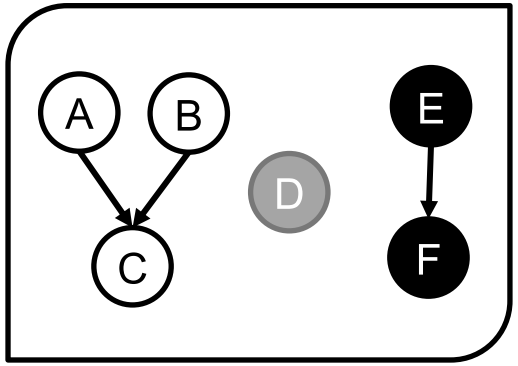
A normal state is a storage state in normal program execution



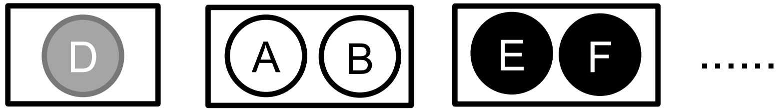
Happens-before graph



A crash state is a possible storage state after a system crash



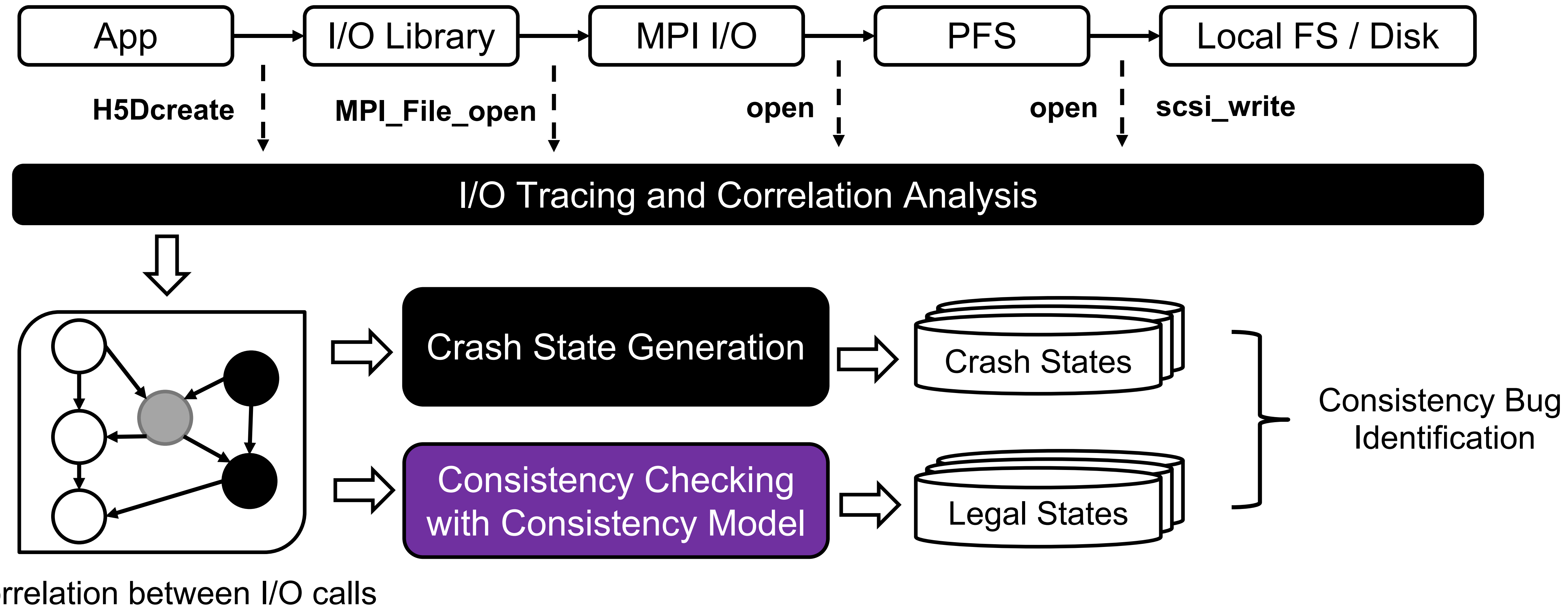
Persists-before graph



More storage states could exist!

ParaCrash

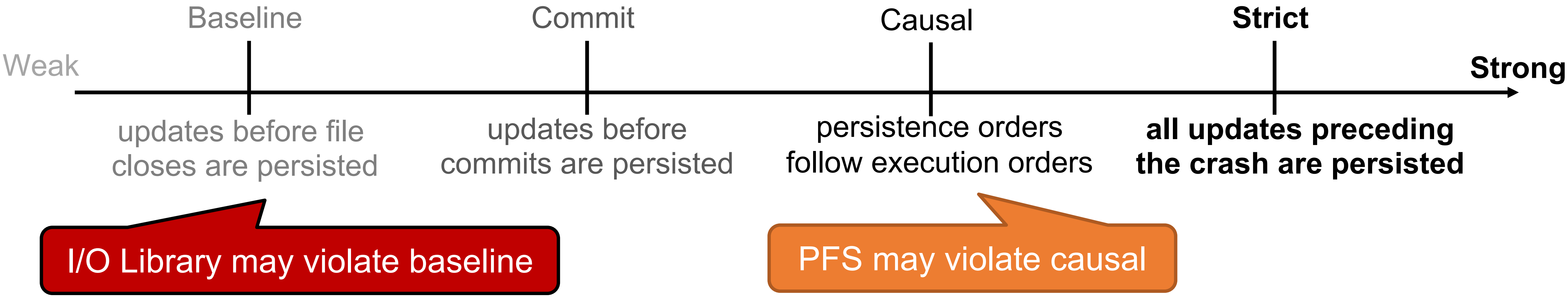
A new framework for detecting crash-consistency bugs in the HPC I/O Stack



Crash Consistency Models for HPC I/O

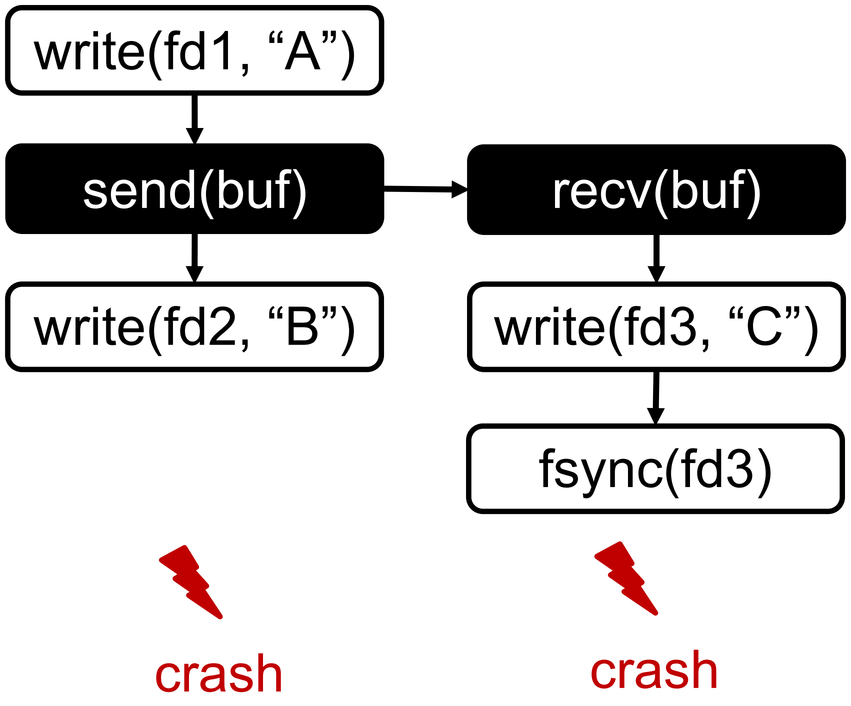
How to check if a crash state is consistent or not?

We define different levels of crash-consistency for HPC I/O



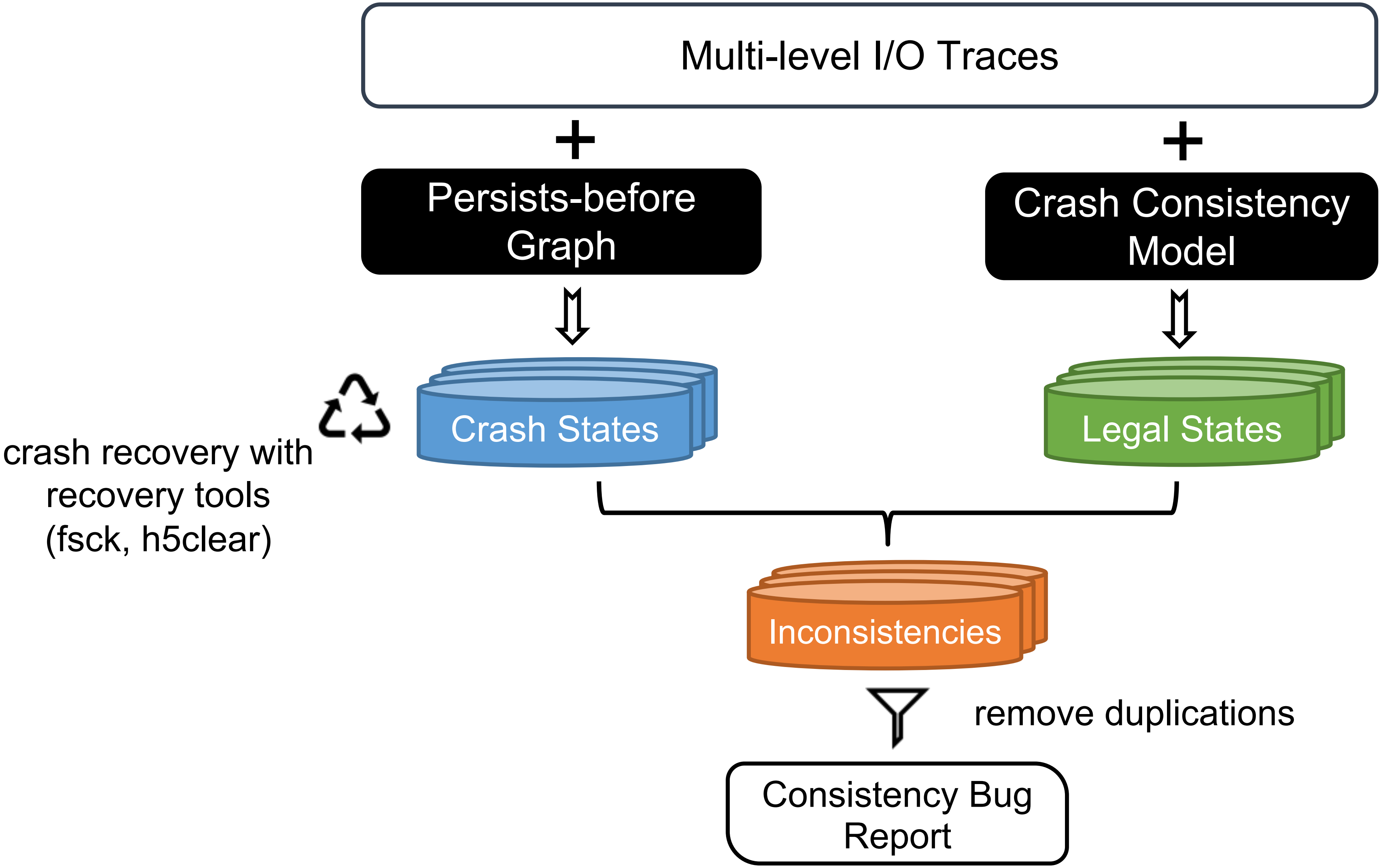
Examples of Crash Consistency Models

An example I/O Trace

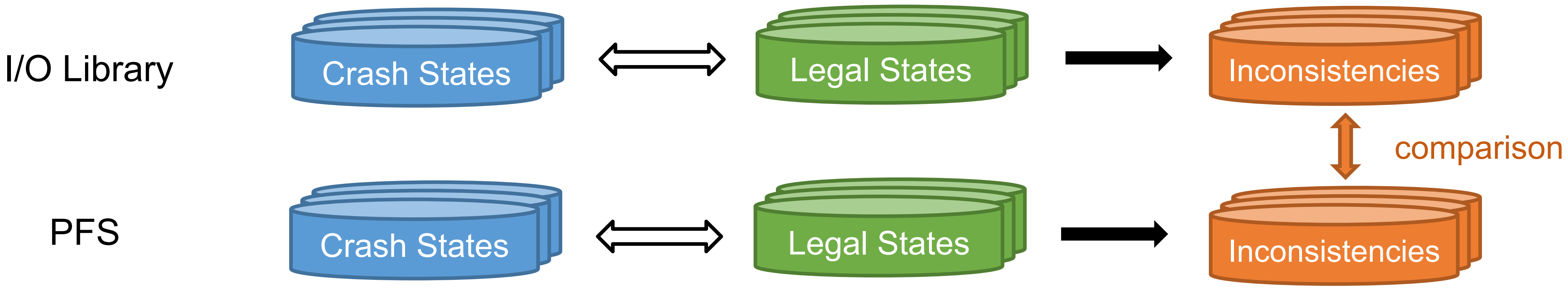


Model	Legal States
Strict	ABC
Causal	AC, ABC
Commit	C, AC, BC, ABC
Baseline	Any combinations of writes

Consistency Checking with Crash Consistency Models



Consistency Checking Across Layers



How to pinpoint a crash consistency bug to its corresponding layer?

We check consistency for a crash state at multiple levels -- the bug is attributed to the lowermost inconsistent layer.

ParaCrash Optimizations

How to reduce the state exploration time?

- Heuristic-based state pruning
- Incremental state construction

How to help developers better understand inconsistency issues?

- Disassembles the lower-level data structures of HDF5 files

More detailed explanations in the paper

ParaCrash Implementation

I/O Tracing: Recorder[1], strace, open-iscsi

HDF5 Support: h5inspect

Experimental Setup

PFSes: BeeGFS, Lustre, GPFS, OrangeFS, GlusterFS

I/O Libraries: HDF5, NetCDF

[1] Recorder 2.0: Efficient parallel I/O tracing and analysis

Test Cases

POSIX Programs

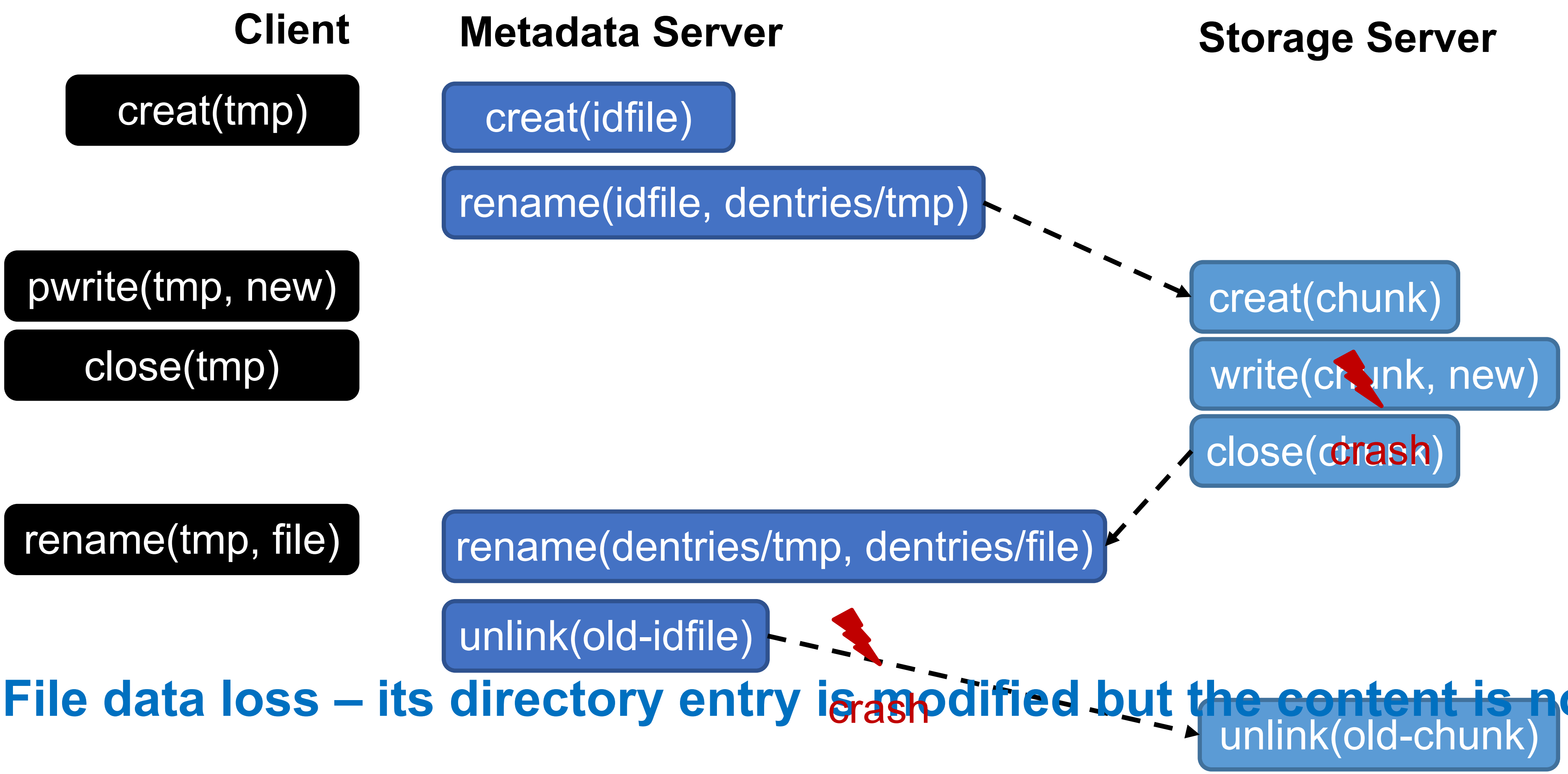
- Atomic-Replace-via-Rename (ARVR)
- Create-and-Rename (CR)
- Rename-and-Create (RC)
- Write-Ahead-Logging (WAL)

HDF5 and NetCDF Programs

- Dataset creation (H5create, CDFcreate)
- Dataset deletion (H5delete)
- Dataset rename (H5rename)
- Dataset resize (H5resize)
- Parallel HDF5 programs

Case Study

The consistency bugs of ARVR Program on BeeGFS



File data loss – its directory entry is modified but the content is not updated

New Crash Consistency Bugs Identified by ParaCrash

No.	Program	Root Cause Layer	Consequence	Sensitivity
1.	ARVR	BeeGFS, OrangeFS	Data loss	N/A
2.	ARVR	BeeGFS	Data loss	N/A
3.	ARVR	GPFS	Data and metadata loss	N/A
4.	CR	BeeGFS, OrangeFS, GPFS	File created in both directories	N/A
5.	RC	BeeGFS, GPFS	File created in a wrong directory	file distrib.
6.	WAL	BeeGFS	No logs written after file modification	file distrib.
7.	WAL	BeeGFS	No logs created after file modification	N/A
8.	WAL	BeeGFS, GlusterFS	No logs created after file modification	N/A
9.	H5-parallel-create	HDF5	Cannot open an unmodified dataset	# of clients
10.	H5-create	PFS	Cannot open an unmodified dataset	N/A
11.	H5-delete	HDF5	Cannot open an unmodified dataset	N/A
12.	H5-rename	HDF5	The renamed dataset is lost	N/A
13.	H5-resize	PFS	Cannot read data from the resized dataset	h5clear options
14.	H5-resize	HDF5	Cannot read data from the resized dataset	dim. of dataset
15.	CDF-create	PFS	Cannot open the file	N/A

New Crash Consistency Bugs Identified by ParaCrash

- **Three inconsistencies of HDF5 programs are attributed to PFS**

No.	Program	Root Cause Layer	Consequence	Sensitivity
1.	ARVR	BeeGFS, OrangeFS	Data loss	N/A
2.	ARVR	BeeGFS	Data loss	N/A
3.	ARVR	GPFS	Data and metadata loss	N/A
4.	CR	BeeGFS, OrangeFS, GPFS	File created in both directories	N/A
5.	RC	BeeGFS, GPFS	File created in a wrong directory	file distrib.
6.	WAL	BeeGFS	No logs written after file modification	file distrib.
7.	WAL	BeeGFS	No logs created after file modification	N/A
8.	WAL	BeeGFS, GlusterFS	No logs created after file modification	N/A
9.	H5-parallel-create	HDF5	Cannot open an unmodified dataset	# of clients
10.	H5-create	PFS	Cannot open an unmodified dataset	N/A
11.	H5-delete	HDF5	Cannot open an unmodified dataset	N/A
12.	H5-rename	HDF5	The renamed dataset is lost	N/A
13.	H5-resize	PFS	Cannot read data from the resized dataset	h5clear options
14.	H5-resize	HDF5	Cannot read data from the resized dataset	dim. of dataset
15.	CDF-create	PFS	Cannot open the file	N/A

No.	Program	Root Cause Layer
9.	H5-parallel-create	HDF5
10.	H5-create	PFS
11.	H5-delete	HDF5
12.	H5-rename	HDF5
13.	H5-resize	PFS
14.	H5-resize	HDF5
15.	CDF-create	PFS

New Crash Consistency Bugs Identified by ParaCrash

- **Many crash consistency bugs may cause severe data loss**

No.	Program	Root Cause Layer	Consequence	Sensitivity
1.	ARVR	BeeGFS, OrangeFS	Data loss	N/A
2.	ARVR	BeeGFS	Data loss	N/A
3.	ARVR	GPFS	Data and metadata loss	N/A
4.	CR	BeeGFS, OrangeFS, GPFS	File created in both directories	N/A
5.	RC	BeeGFS, GPFS	File created in a wrong directory	file distrib.
6.	WAL	BeeGFS	No logs written after file modification	file distrib.
7.	WAL	BeeGFS	No logs created after file modification	N/A
8.	WAL	BeeGFS, GlusterFS	No logs created after file modification	N/A
9.	H5-parallel-create	HDF5	Cannot open an unmodified dataset	# of clients
10.	H5-create	PFS	Cannot open an unmodified dataset	N/A
11.	H5-delete	HDF5	Cannot open an unmodified dataset	N/A
12.	H5-rename	HDF5	The renamed dataset is lost	N/A
13.	H5-resize	PFS	Cannot read data from the resized dataset	h5clear options
14.	H5-resize	HDF5	Cannot read data from the resized dataset	dim. of dataset
15.	CDF-create	PFS	Cannot open the file	N/A

Consequence
Data loss
Data loss
Data and metadata loss
Cannot open an unmodified dataset
Cannot open an unmodified dataset
The renamed dataset is lost
Cannot read data from the resized dataset
Cannot read data from the resized dataset
Cannot open the file

New Crash Consistency Bugs Identified by ParaCrash

- **Some crash consistency bugs are configuration-dependent**

No.	Program	Root Cause Layer	Consequence	Sensitivity
1.	ARVR	BeeGFS, OrangeFS	Data loss	N/A
2.	ARVR	BeeGFS	Data loss	N/A
3.	ARVR	GPFS	Data and metadata loss	N/A
4.	CR	BeeGFS, OrangeFS, GPFS	File created in both directories	N/A
5.	RC	BeeGFS, GPFS	File created in a wrong directory	file distrib.
6.	WAL	BeeGFS	No logs written after file modification	file distrib.
7.	WAL	BeeGFS	No logs created after file modification	N/A
8.	WAL	BeeGFS, GlusterFS	No logs created after file modification	N/A
9.	H5-parallel-create	HDF5	Cannot open an unmodified dataset	# of clients
10.	H5-create	PFS	Cannot open an unmodified dataset	N/A
11.	H5-delete	HDF5	Cannot open an unmodified dataset	N/A
12.	H5-rename	HDF5	The renamed dataset is lost	N/A
13.	H5-resize	PFS	Cannot read data from the resized dataset	h5clear options
14.	H5-resize	HDF5	Cannot read data from the resized dataset	dim. of dataset
15.	CDF-create	PFS	Cannot open the file	N/A

Sensitivity
file distrib.
file distrib.
of clients
h5clear options
dim. of dataset

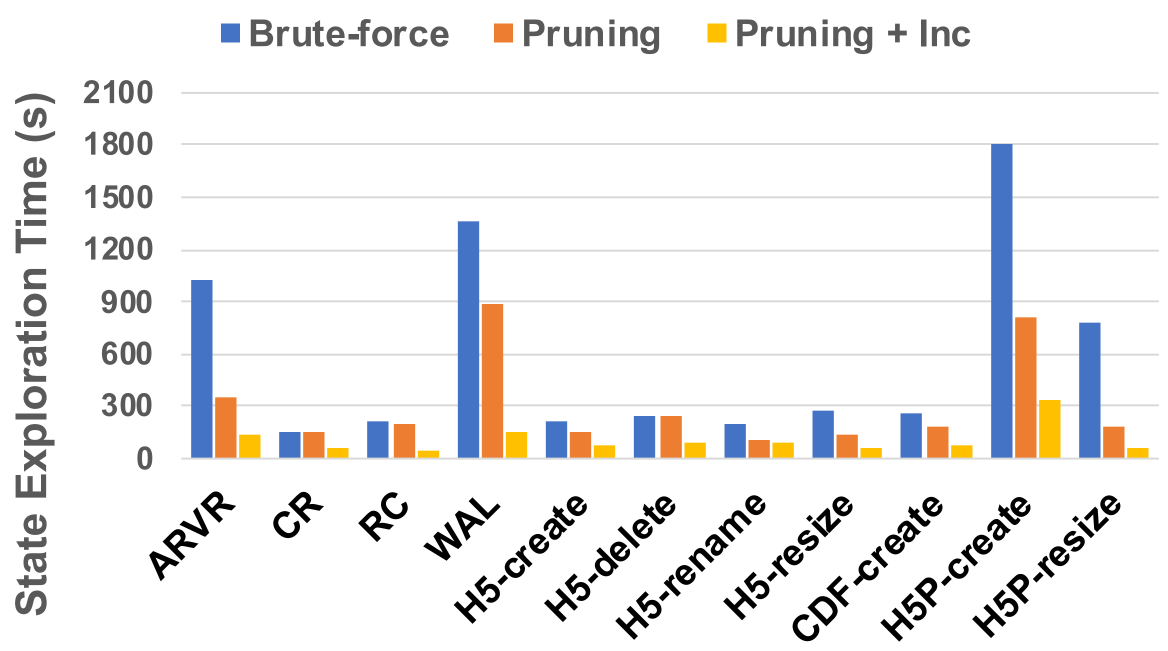
New Crash Consistency Bugs Identified by ParaCrash

No.	Program	Root Cause Layer	Consequence	Sensitivity
1.	ARVR	BeeGFS, OrangeFS	Data loss	N/A
2.	ARVR	BeeGFS	Data loss	N/A
3.	ARVR	GPFS	Data and metadata loss	N/A
4.	CR	BeeGFS, OrangeFS, GPFS	File created in both directories	N/A
5.	RC	BeeGFS, GPFS	File created in a wrong directory	file distrib.
6.	WAL	BeeGFS	No logs written after file modification	file distrib.
7.	WAL	BeeGFS	No logs created after file modification	N/A
8.	WAL	BeeGFS, GlusterFS	No logs created after file modification	N/A
9.	H5-parallel-create	HDF5	Cannot open an unmodified dataset	# of clients
10.	H5-create	PFS	Cannot open an unmodified dataset	N/A
11.	H5-delete	HDF5	Cannot open an unmodified dataset	N/A
12.	H5-rename	HDF5	The renamed dataset is lost	N/A
13.	H5-resize	PFS	Cannot read data from the resized dataset	h5clear options
14.	H5-resize	HDF5	Cannot read data from the resized dataset	dim. of dataset
15.	CDF-create	PFS	Cannot open the file	N/A

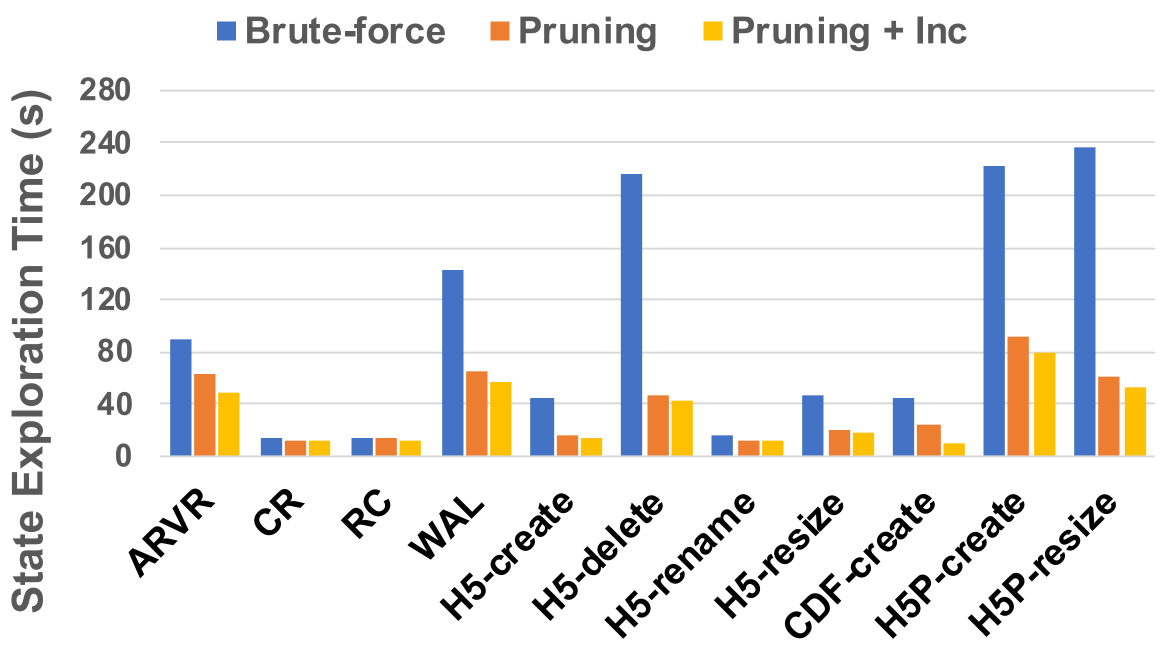
- **Three inconsistencies of HDF5 programs are attributed to PFS**
- **Many crash consistency bugs may cause severe data loss**
- **Some crash consistency bugs are configuration-dependent**

8 new bugs identified with POSIX programs and 7 new bugs with HDF5 programs

Performance of ParaCrash



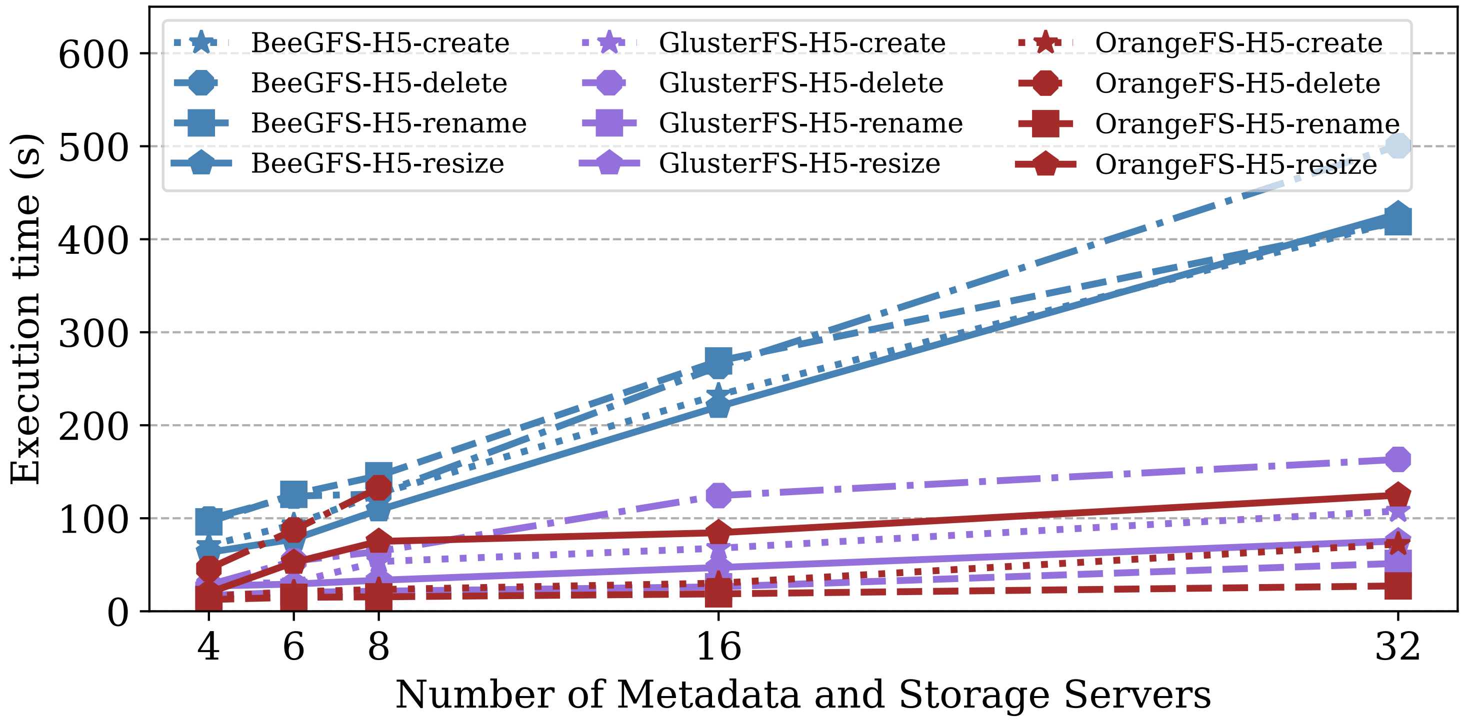
BeeGFS



OrangeFS

ParaCrash performs up to 7.3× faster with crash state pruning strategy

Scalability of ParaCrash



ParaCrash scales linearly with an increasing number of servers

Conclusion

- We present ParaCrash, the 1st crash consistency testing framework for HPC I/O
- We define different levels of crash consistency models for HPC I/O
- ParaCrash identifies 15 crash consistency bugs for 5 PFSes and 2 I/O libraries
- ParaCrash is up to 7.3 times faster in bug detection with its state-pruning policy

We open-sourced ParaCrash on GitHub!

https://github.com/my-HenryS/ParaCrash

Thank you!

Jinghan Sun, Jian Huang, Marc Snir

js39@illinois.edu

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN