

Lustre
Admin
Dev
2
6

Parallel Tools for Large-Scale File Migration and Verification on HPC Storage Systems

Sergey Noskov
01.10.2026

JOHANNES GUTENBERG
UNIVERSITÄT MAINZ



OUTLOOK FOR THE NEXT 25 MINUTES

- Motivation – Why the Tool Was Created
- Problem Statement
- Architecture and operation
 - Scheduler and worker model
 - Worker operations: rsync, copy, diff, lustre OST migration
 - Lustre file migration helper
- Benchmarks
- Conclusion

MOTIVATION

- Copy very large directory trees
 - 10+ million files
 - 100+ TiB
- Preserve file metadata
 - permissions, ownership and timestamps
 - extended attributes
- Exploit filesystem parallelism
- Verify data during or after the copy
- Continue processing despite individual file errors

MOTIVATION

EXISTING APPROACHES

Fpsync	partitions the directory tree runs multiple copy/sync processes in parallel
gnu parallel	+ rsync/cp/...
parsync	a parallel rsync wrapper in Perl
xcp	a rust program
mpiFileUtils	MPI-based, designed for large HPC datasets
pftool	Parallel File Tool, multi-process / multi-node architecture
copy2 ¹	parallel data transfer for HPC environments

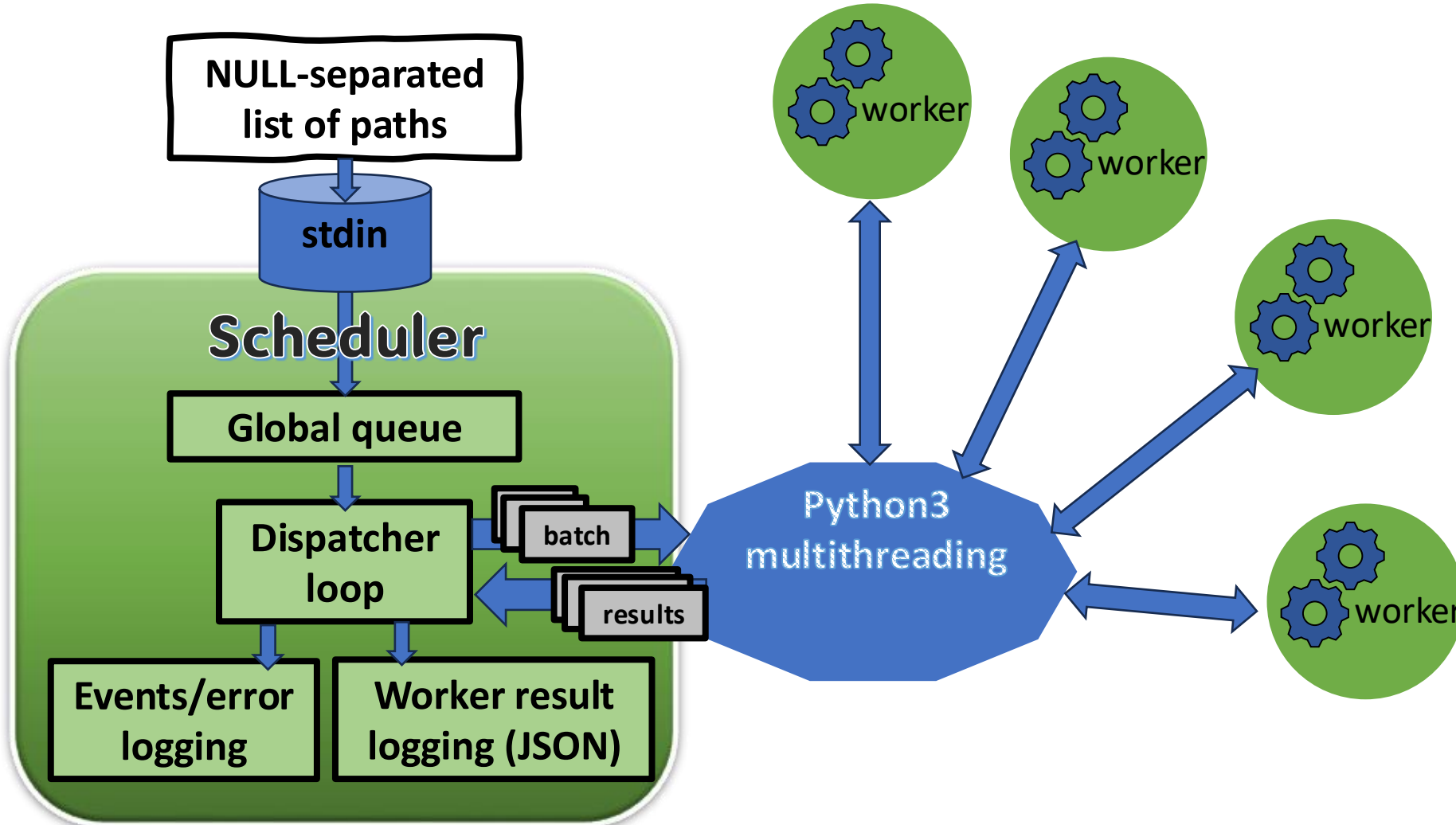
¹ A. Lambert and D. Panish, “Copy2: Efficient Data Transfer in High Performance Environments,” PEARC ’26, ACM, 2026. doi:10.1145/3785462.3815832

PROBLEM STATEMENT

We needed a tool that could:

- process millions of files continuously
- use multiple workers without requiring MPI
- dynamically tune the workload profile by changing
 - worker count
 - batch size
- handle very long paths
- preserve filesystem metadata
- verify copied data
- survive individual file and filesystem errors
- produce machine-readable per-file status information
- support Lustre-specific operations such as OST migration
 - One scheduler, multiple filesystem operations

ARCHITECTURE



Operation type
(selected at startup)

rsync process

internal copy
function

External copy
program

Internal diff
function

Lustre OST
migration helper

OPERATION

```
parallel_tools.py METHOD PROJECT_NAME CONFIGFILE # write a default configuration  
$EDITOR CONFIGFILE
```

```
find /src/root -print0 | parallel_tools.py start CONFIGFILE
```

```
parallel_tools.py start myjob1.cfg < filelist1.txtz 2> myjob1.log 1> myjob1.jsonlog
```

```
parallel_tools.py start myjob2.cfg < filelist2.txtz 2> myjob2.log 1> myjob2.jsonlog
```

```
parallel_tools.py -p METHOD PROJECTNAME SRC DST [INPUT] [-j N]
```

SCHEDULER EVENTS

Log file contains:

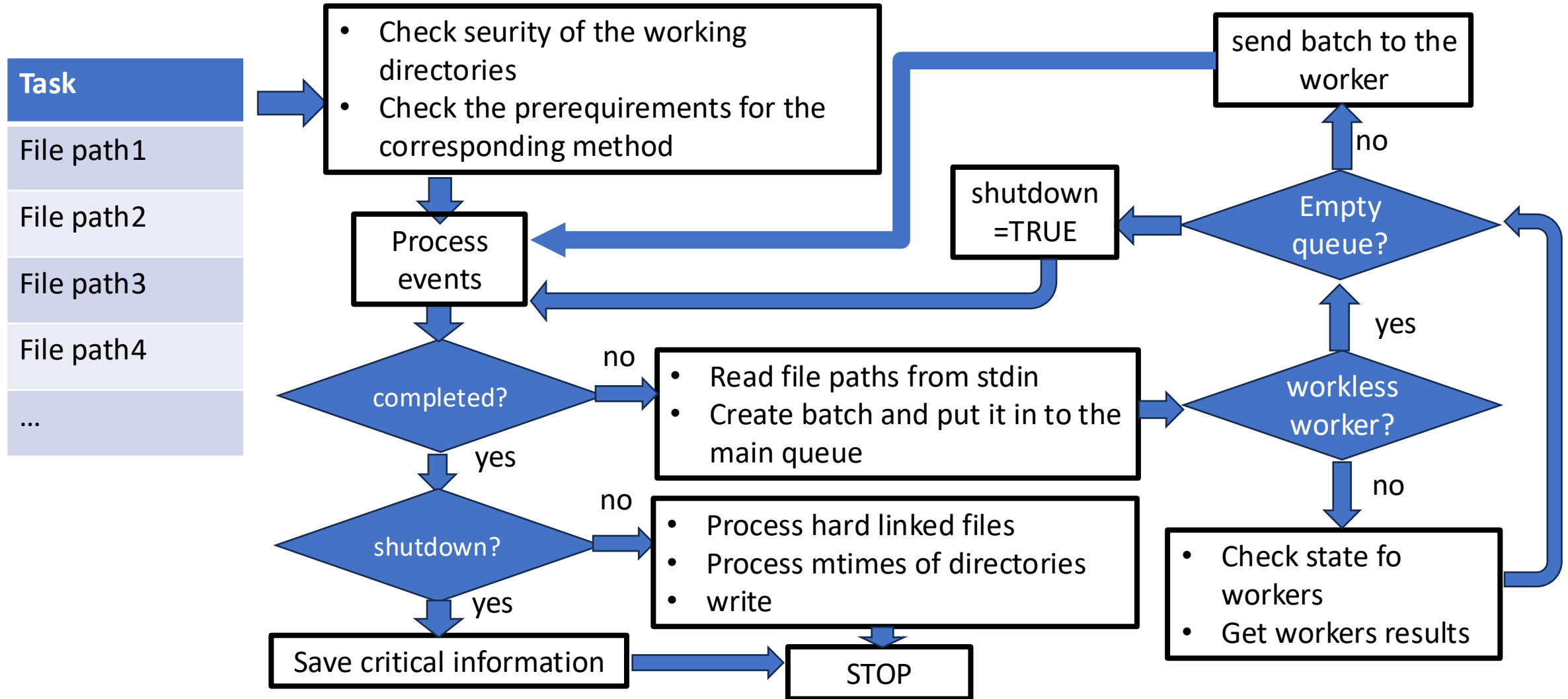
2026-09-22 20:58:15.203098 Scheduler PID: 4096426 status: workers=8 active=8 ...

```
parallel_tools.py pause /srv/jobs/myjob.log # SIGUSR1
parallel_tools.py resume /srv/jobs/myjob.log # SIGUSR2
parallel_tools.py reload /srv/jobs/myjob.log # SIGHUP
parallel_tools.py stop /srv/jobs/myjob.log # SIGTERM
```

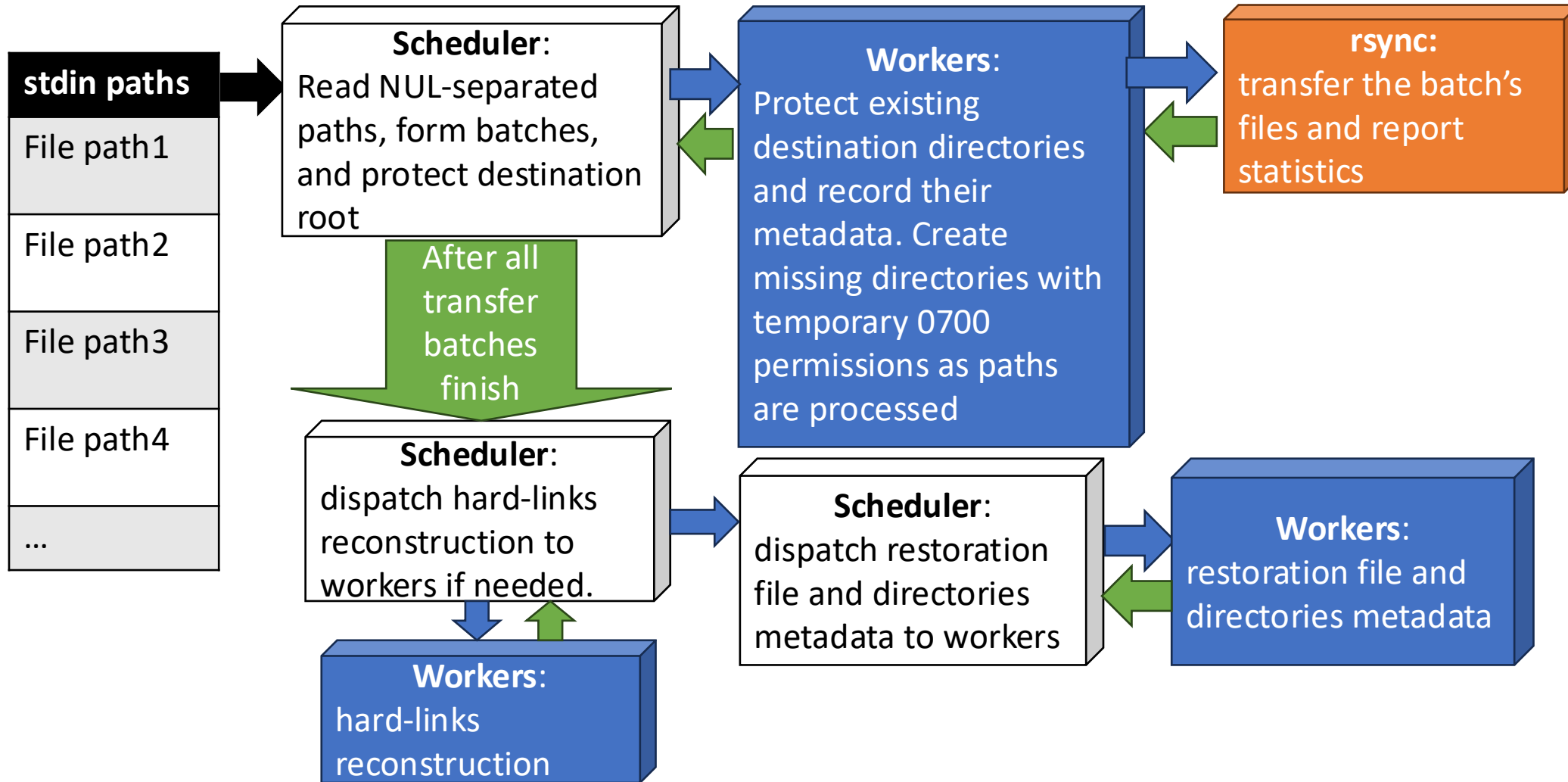
or

```
parallel_tools.py pause Scheduler-PID # SIGUSR1
parallel_tools.py resume Scheduler-PID # SIGUSR2
parallel_tools.py reload Scheduler-PID # SIGHUP
parallel_tools.py stop Scheduler-PID # SIGTERM
```

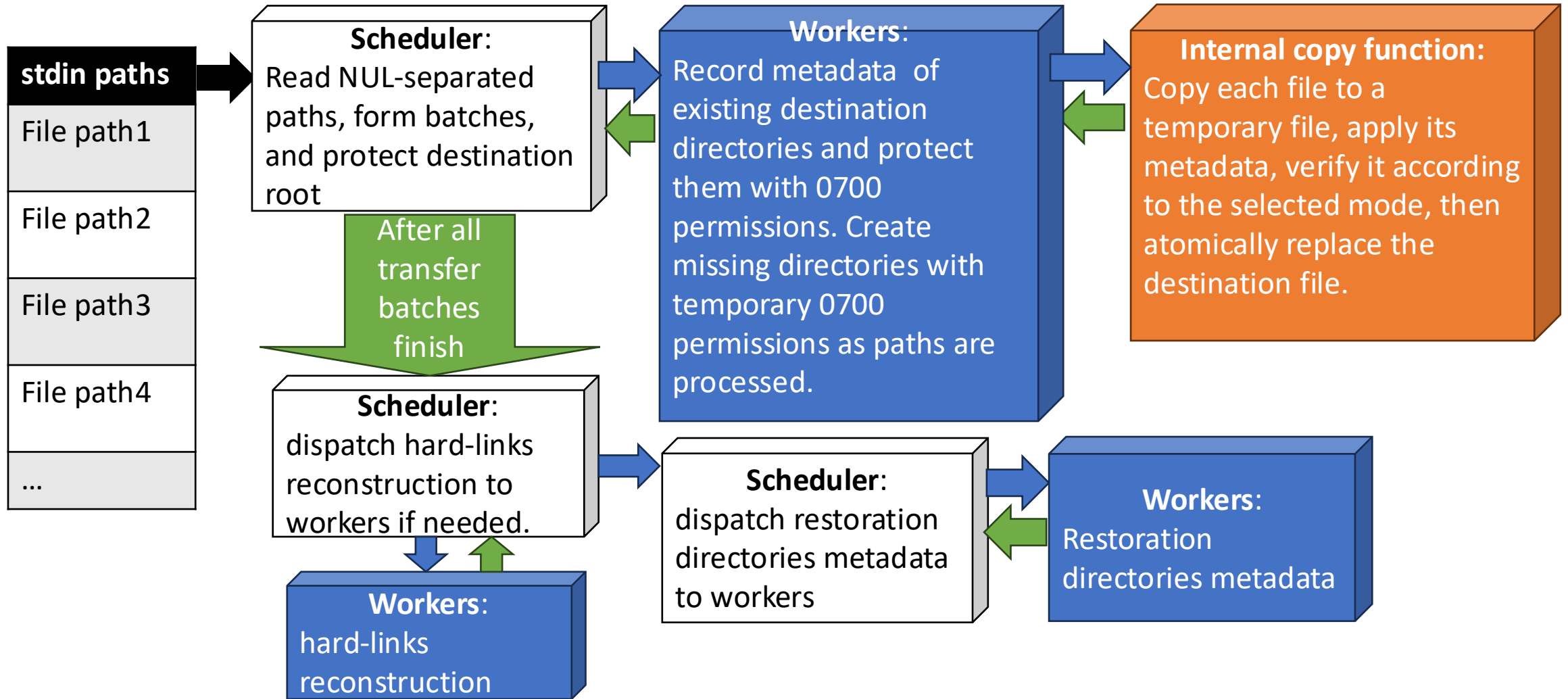
SCHEDULER WORKFLOW



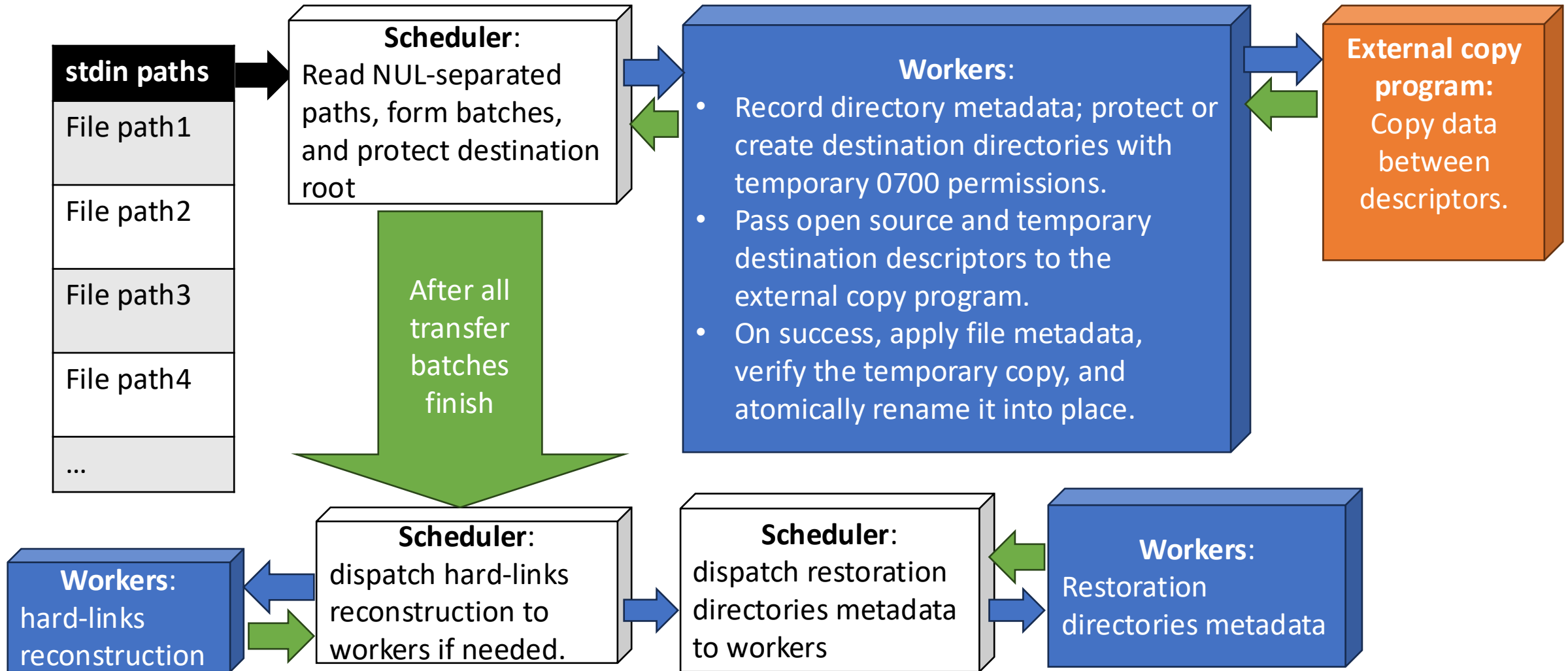
COPY OPERATION WITH RSYNC



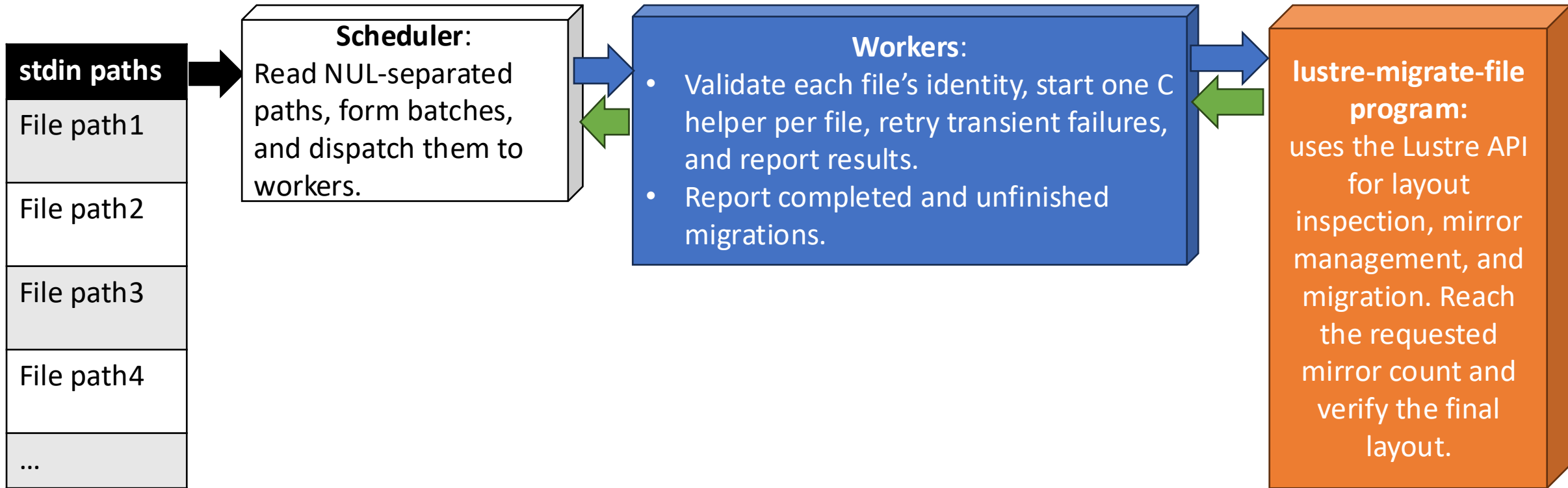
COPY OPERATION WITH THE INTERNAL FUNCTION



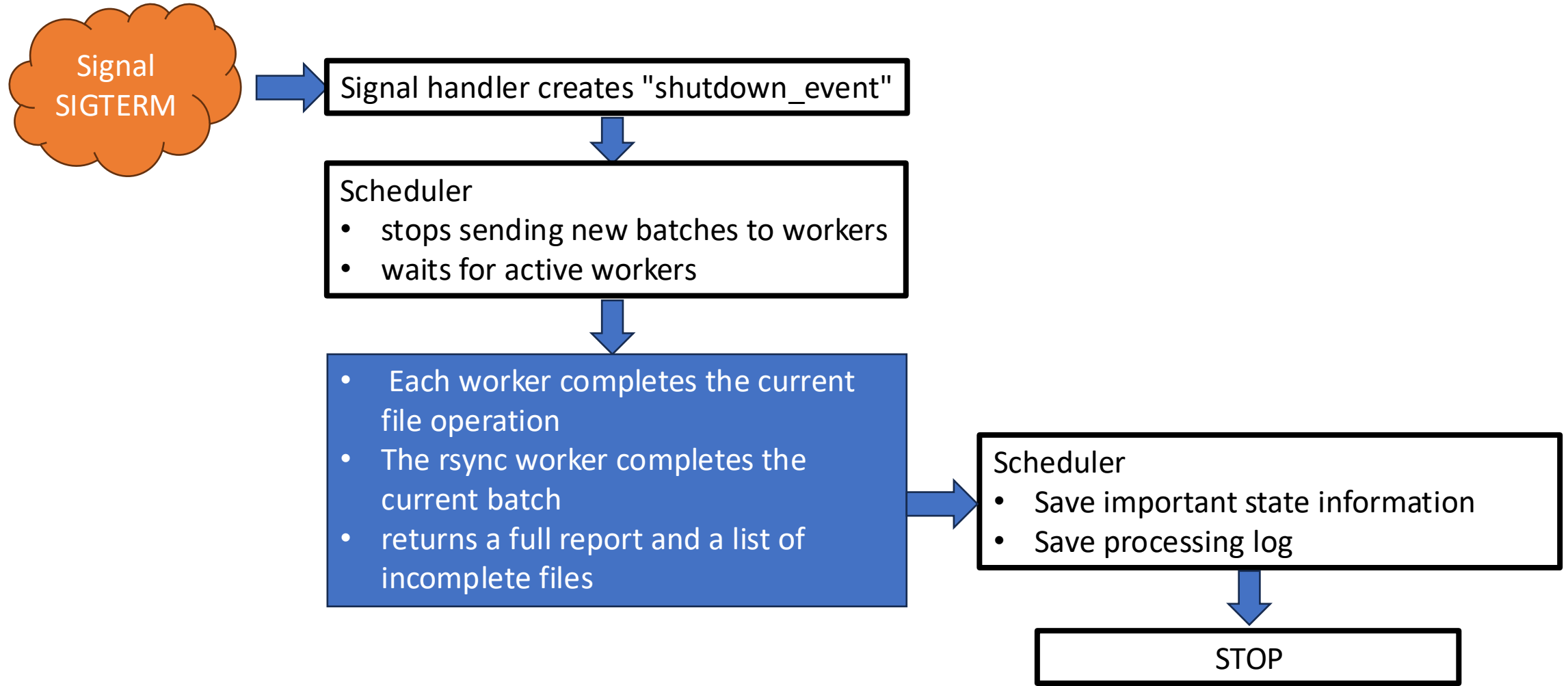
COPY OPERATION WITH AN EXTERNAL PROGRAM



OST MIGRATION OPERATION



SCHEDULER EVENTS: SHUTDOWN



COPY OPERATIONS PARAMETERS

Parameter	Meaning
max_path_length	"engine" : PATH_MAX-sized paths are copied with the internal openat engine; normal paths remain in the rsync batch. "rsync" : All allowed paths, including PATH_MAX-sized paths, are passed directly to rsync. N : Paths longer than N bytes are rejected before processing and recorded as path_length_exceeded in the JSON log.
stdin_chunk_size	integer ≥ 1 , default 16777216 (16 MiB) Read size used for the NUL-terminated stdin byte stream.
max_input_record_bytes	integer ≥ 1 , default 8388608 (8 MiB) Maximum length of one NUL-terminated input path. An unterminated last path is always an input error.

MIGRATION PARAMETERS

Parameter	Meaning
banned_osts	list of OST indexes, e.g. [10, 11, 16] OSTs from which existing allocations must be removed
allowed_osts	list of OST indexes, default [] Where a NEW mirror may be placed. Empty means Lustre chooses.
migrate_helper	absolute path or null, default null Where to find lustre-migrate-file. null looks next to this script first and then along PATH. The executable is opened, checked and run through its own descriptor, so the file that was inspected is the file that runs. It must not be writable by group or other, and under root it must be owned by root.

BENCHMARK PLATFORM: LUSTRE 2.15.8

Metadata storage	Object storage
HA pair 2x PowerEdge R650 (256Gb RAM)	HA pair 2x PowerEdge R650 (256Gb RAM)
RAID Controller Dell PowerVault ME5024	EMC Storage Expansion Enclosure ME484
SSD RAID-based MGT and MDTs	ZFS dRAID-based OSTs
MGT 50 GB 2x MDT 7 TB	4x OST 171 TB draid3:6d:21c:1s
InfiniBand bond 2x100Gbit (active-backup mode) Ethernet LACP bond (L3+4) 2x100Gbit	InfiniBand bond 2x100Gbit (active-backup mode) Ethernet LACP bond (L3+4) 2x100Gbit

Lustre client
PowerEdge R650 (256Gb RAM)
InfiniBand bond 2x100Gbit (active-backup mode) Eth LACP bond (L3+4) 2x100Gbit

BENCHMARK PLATFORM: LUSTRE 2.15.8

TUNING

All lustre default parameter

zfs_arc_max=128 GiB

zfs_dirty_data_max=2 GiB

zfs_vdev_async_read_min_active=1

zfs_vdev_async_read_max_active=8

zfs_vdev_async_write_min_active=2

zfs_vdev_async_write_max_active=8

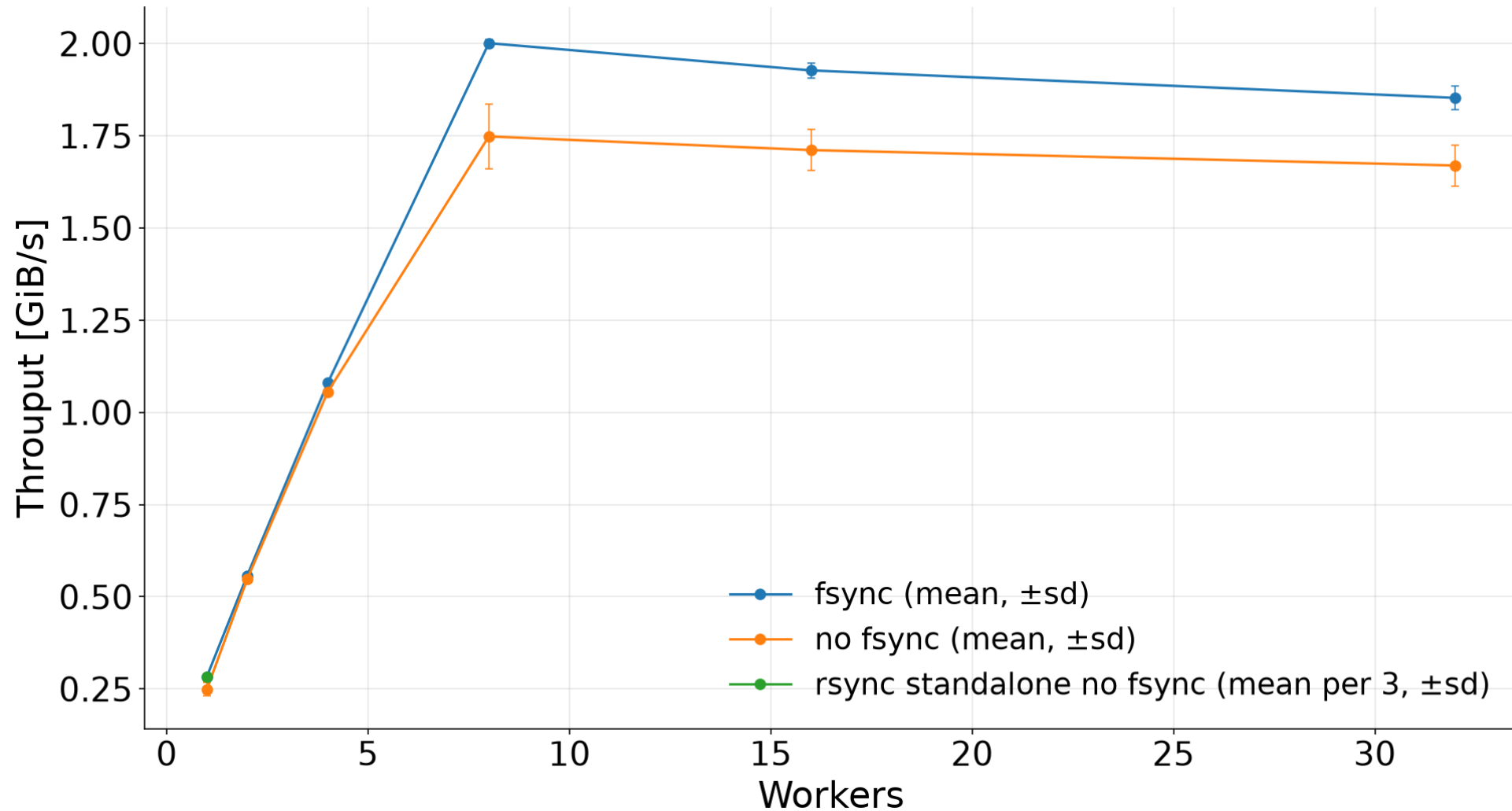
zfs_vdev_sync_read_min_active=4

zfs_vdev_sync_read_max_active=8

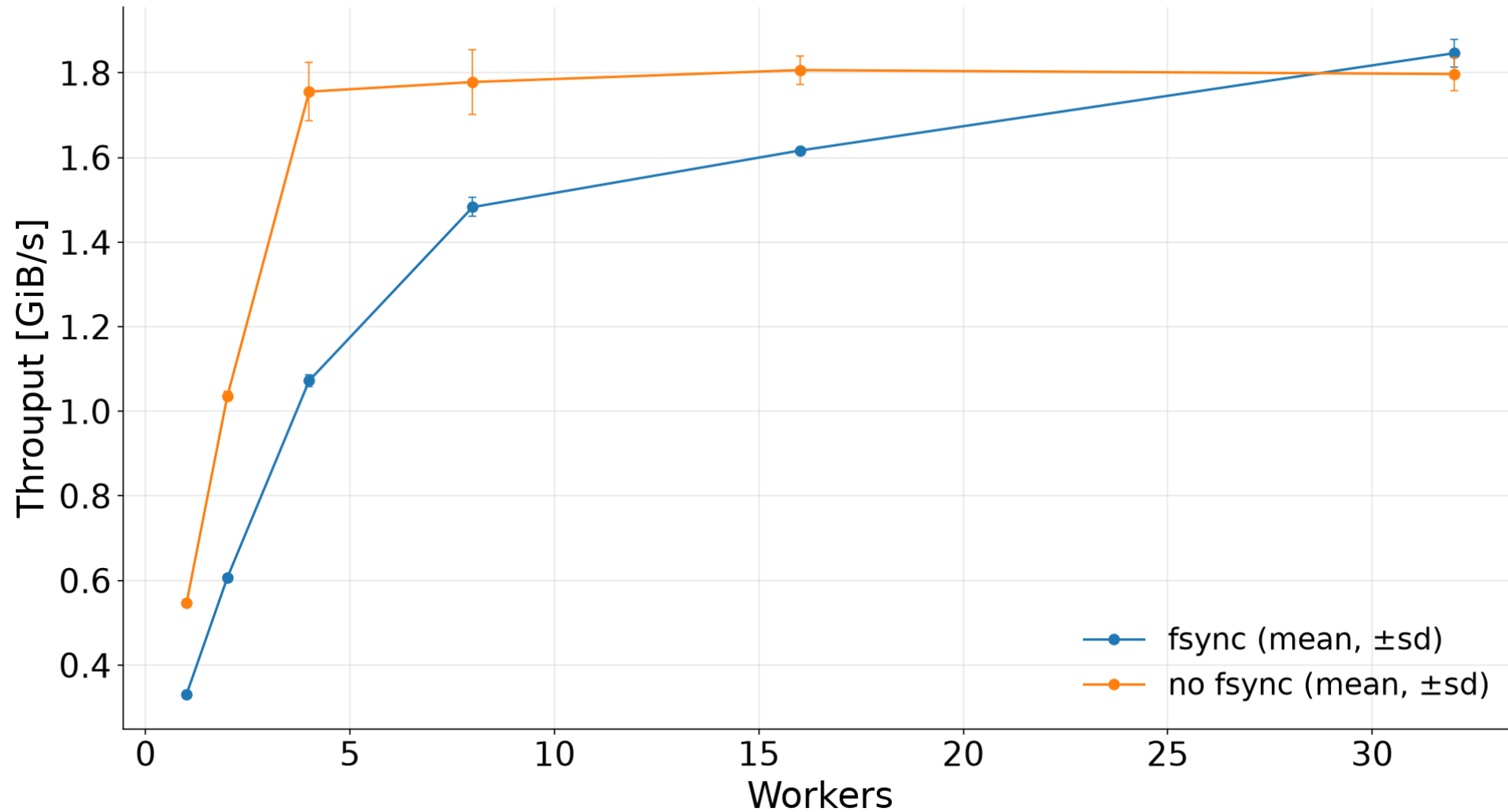
zfs_vdev_sync_write_min_active=4

zfs_vdev_sync_write_max_active=8

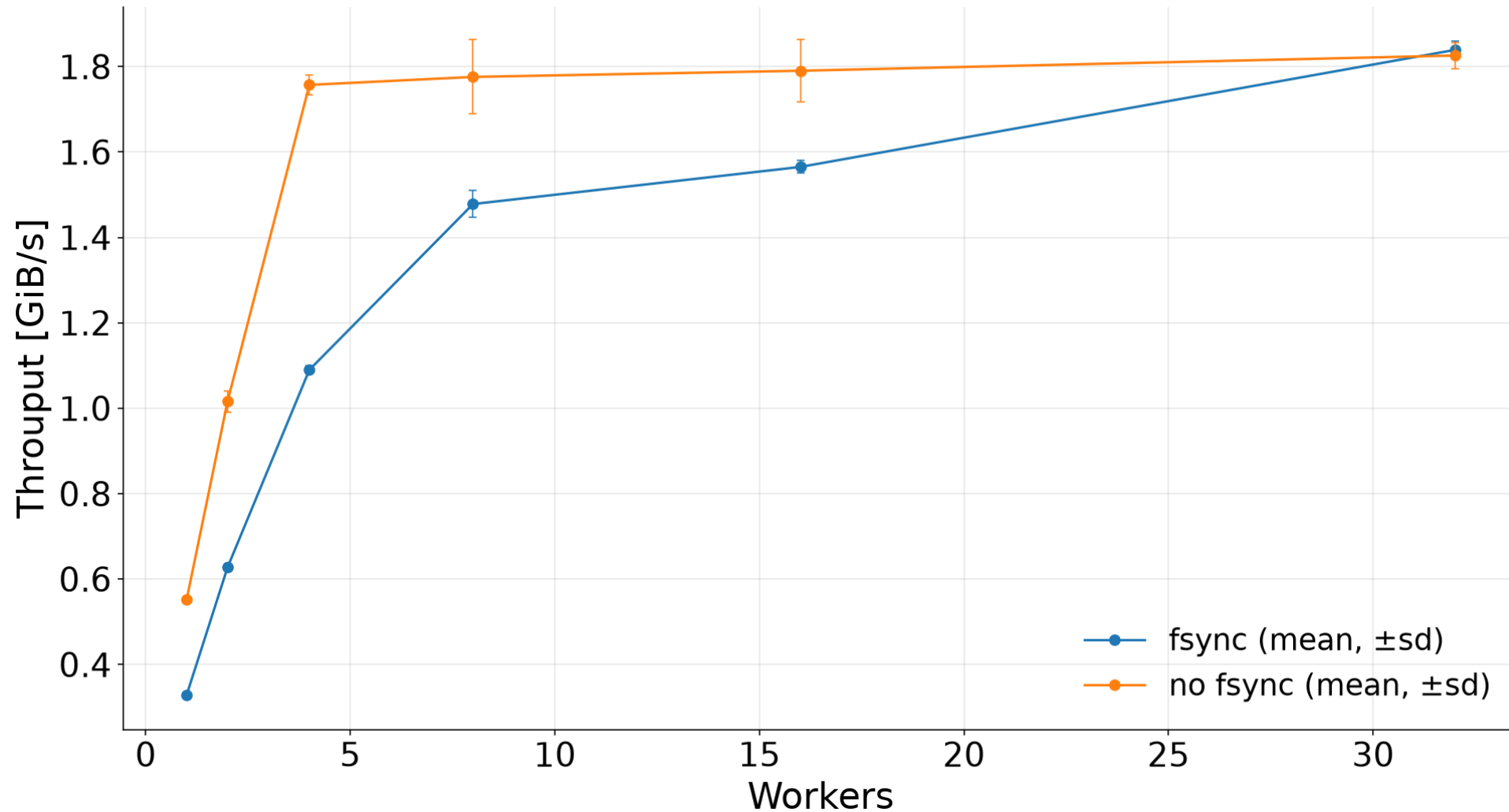
SCALING THE RSYNC OPERATION (1.18 TIB)



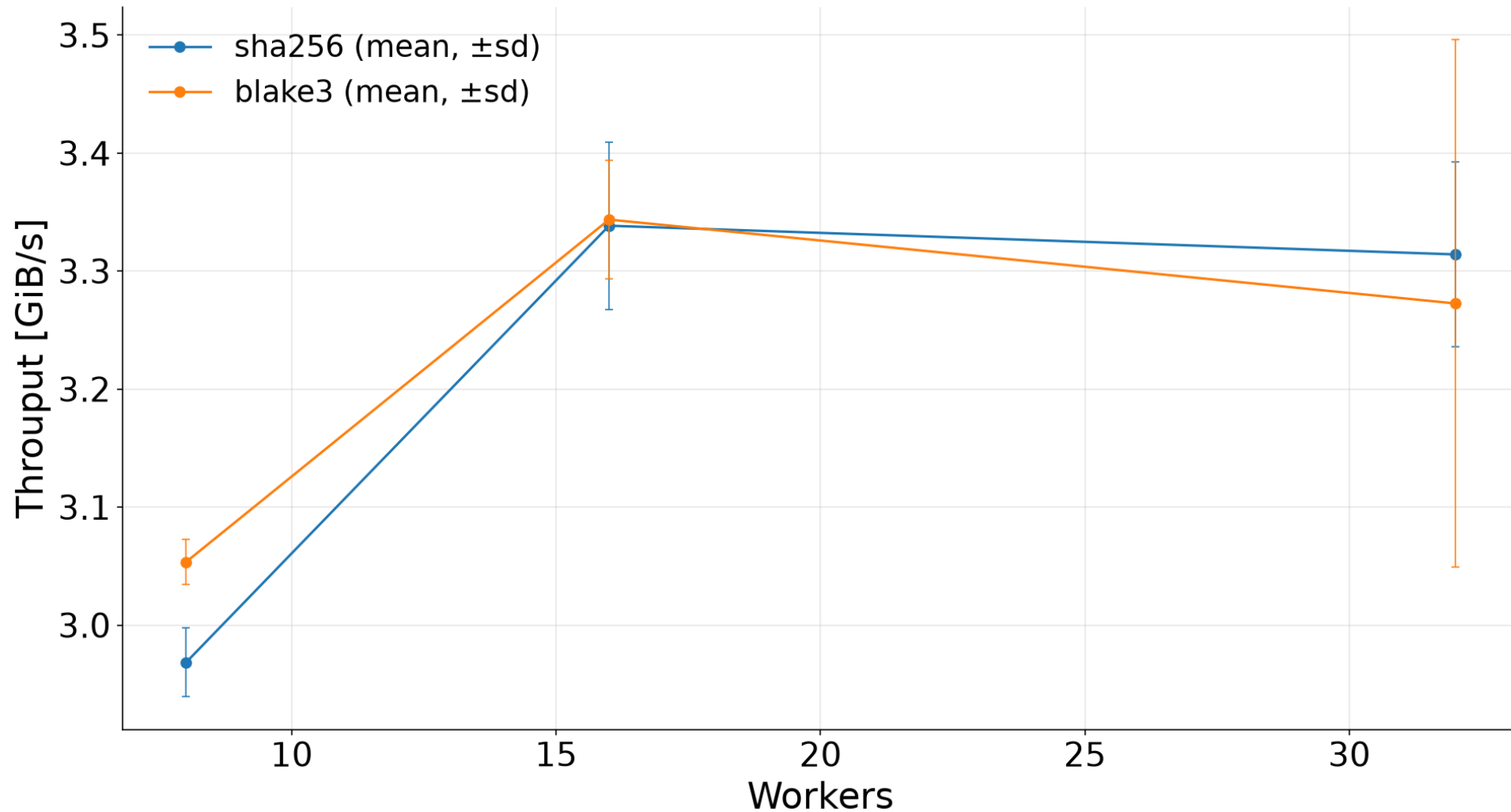
SCALING INTERNAL COPY OPERATION (1.18 TIB)



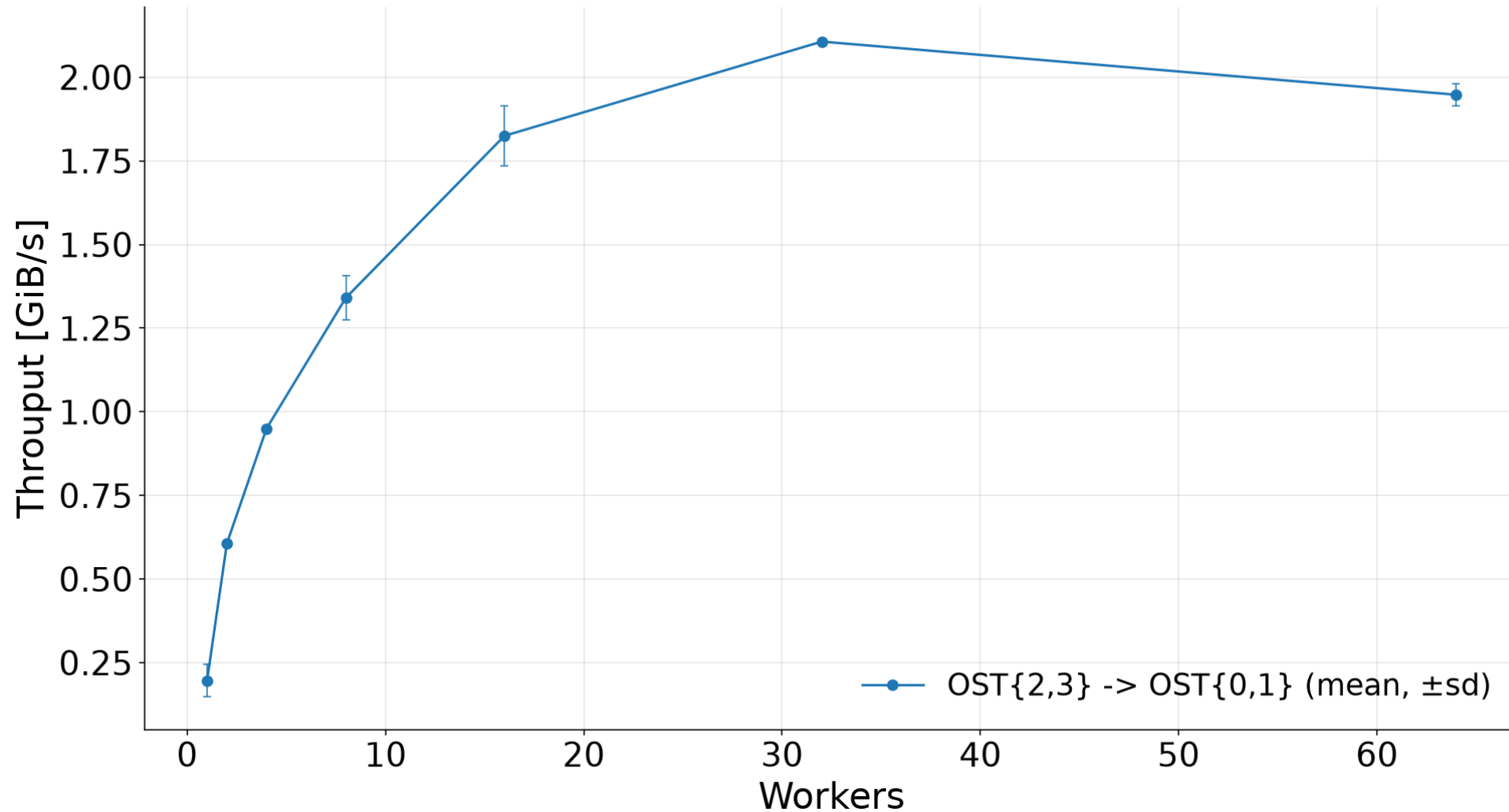
SCALING LINUX CP OPERATION (1.18 TIB)



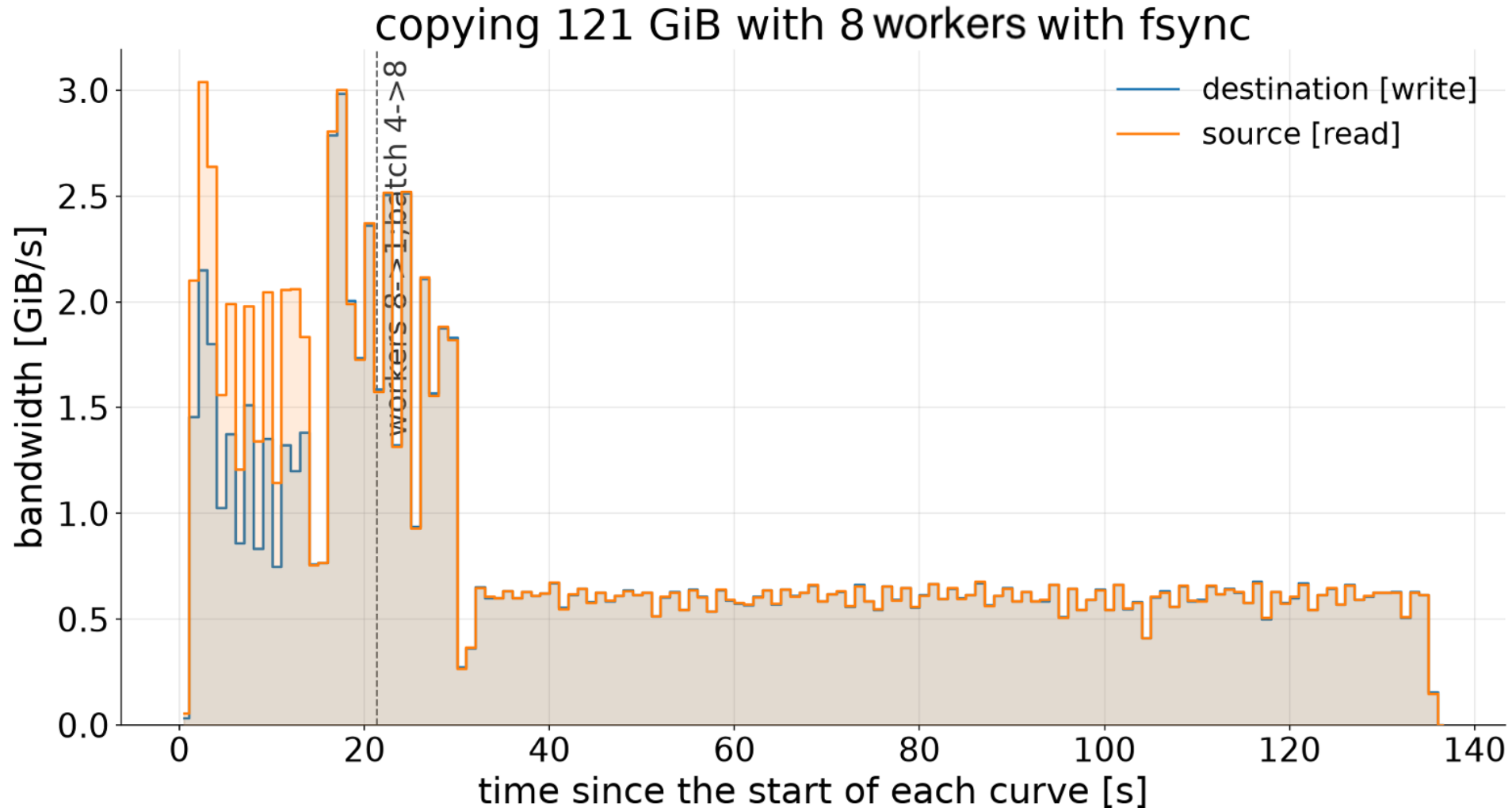
SCALING PARALLEL VERIFICATION (1.18 TiB)



SCALING LUSTRE OST MIGRATION (1.18 TiB)

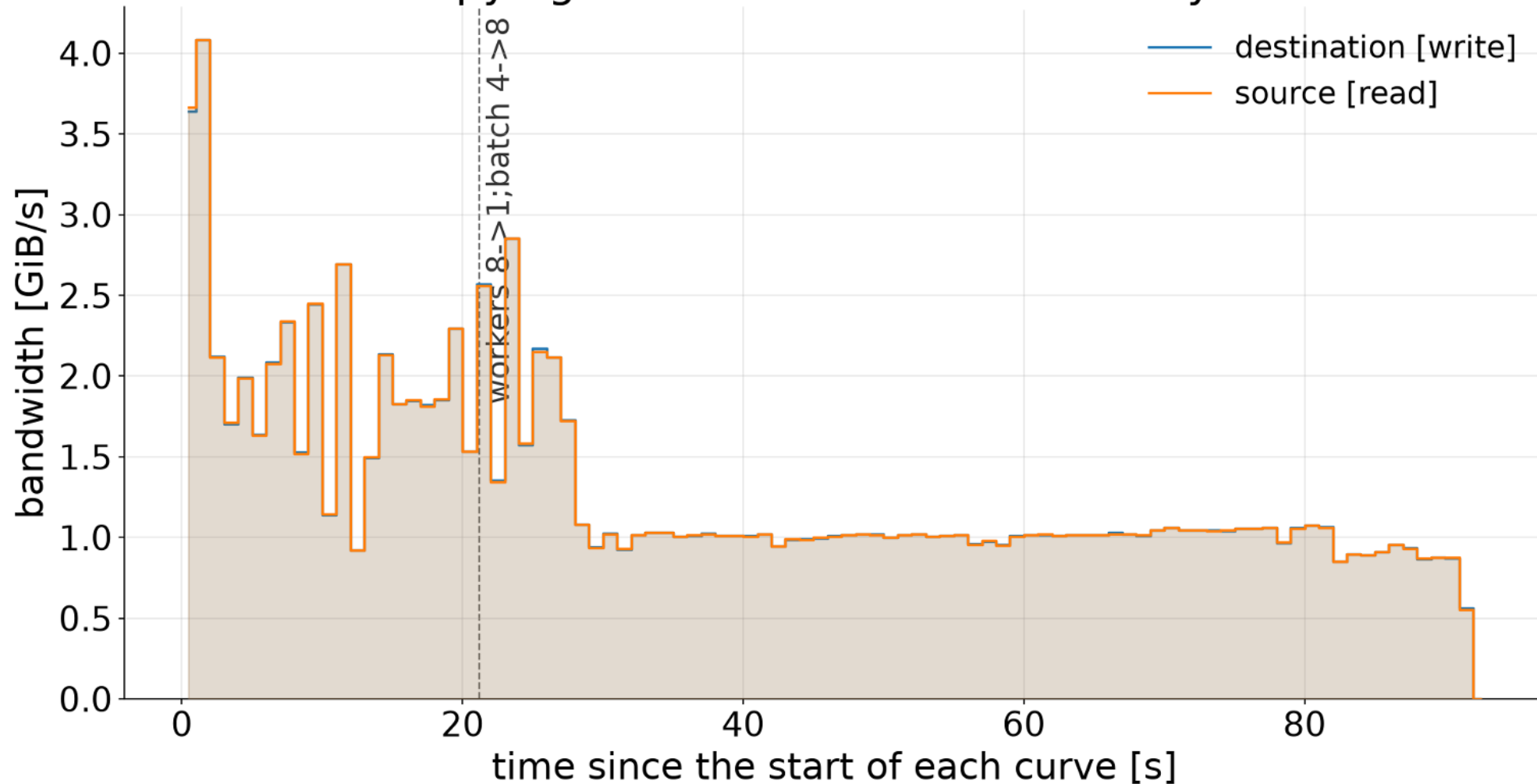


DYNAMIC WORKER SCALING (FSYNC ON)

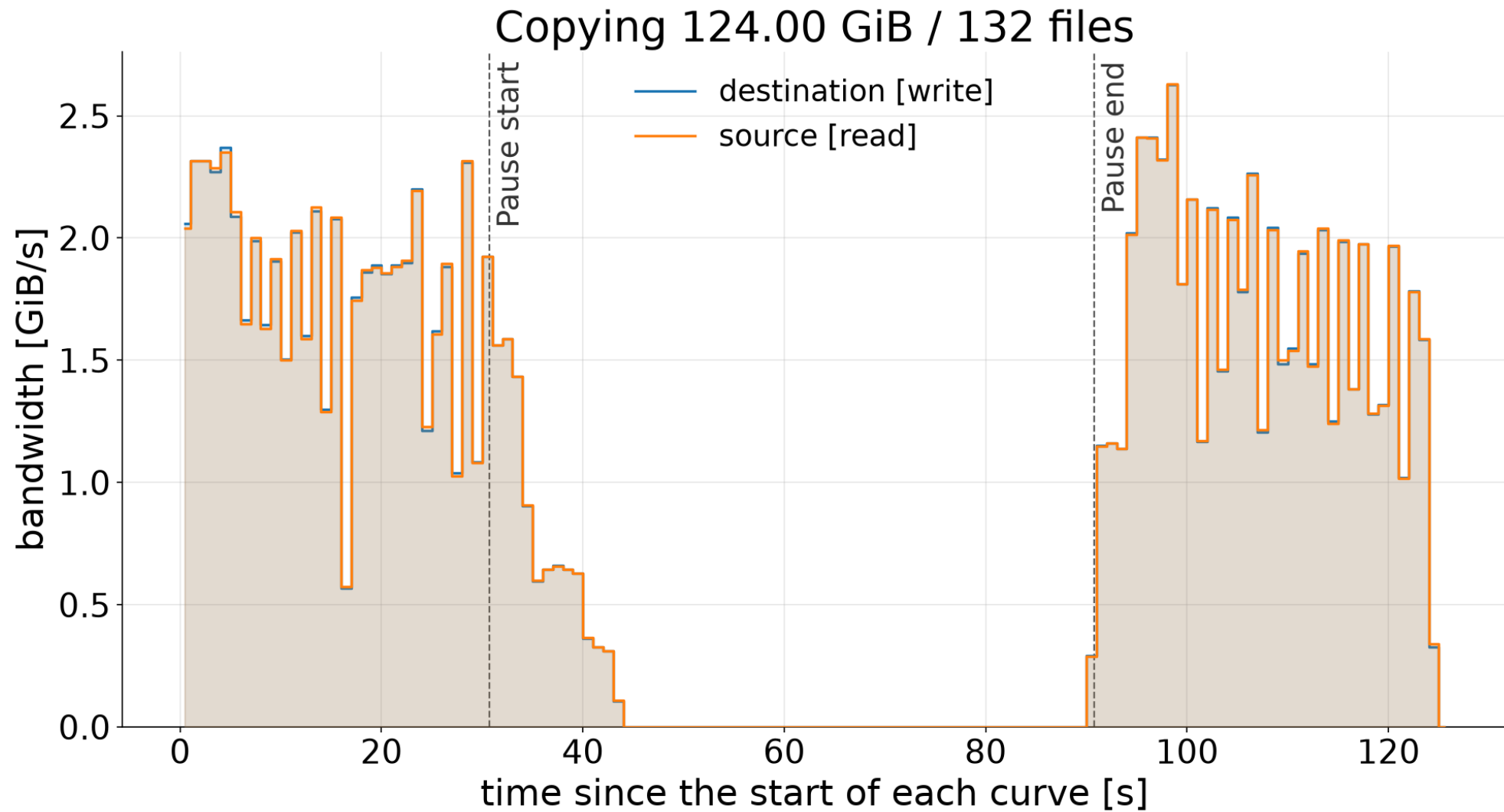


DYNAMIC WORKER SCALING (FSYNC OFF)

copying 121 GiB with 8 workers no fsync



PAUSE TEST



CONCLUSION

- One scheduler for different file operations
- Robust handling of complex directory trees: sparse files, hard links, symbolic links, and paths beyond PATH_MAX

```
ng_path/very_long_path/very_long_path/very_lo  
/very_long_path/very_long_path/very_long_path  
[root@m3dm02 v]# pwd|wc  
      1      1  17714  
[root@m3dm02 v]#
```

- Lustre-specific migration helper using the native Lustre API
- Different operations reach their optimal parallelism at different worker counts
- Worker count and batch size allow dynamic workload tuning
- Beyond a certain point, performance is limited by the storage system rather than the scheduler

THANK YOU FOR YOUR ATTENTION

I would like to express my sincere thanks to my colleagues for their support, valuable discussions and ideas.

<https://github.com/jguzdv/parallel-tools>

