# ADA-FS

## A job-temporal, ad-hoc file system for HPC

Marc-André Vef[1], Nafiseh Moti[1], André Brinkmann[1],
Mehmet Soysal[2], Achim Streit[2],
Sebastian Oeste[3], Andreas Knüpfer[3], Wolfgang E. Nagel[3]

March 21th 2018
SPPEXA Annual Plenary Meeting 2018
Garching


TECHNISCHE UNIVERSITÄT DRESDEN [3]

KIT Karlsruhe Institute of Technology [2]

JG|U [1]

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

# Motivation



- The shared I/O subsystem is the slowest component in HPC systems
- Compute jobs fight for I/O resources (job interference)
- Larger I/O pressure on the shared medium with larger clusters
- Node-local storage is available but often unused by applications

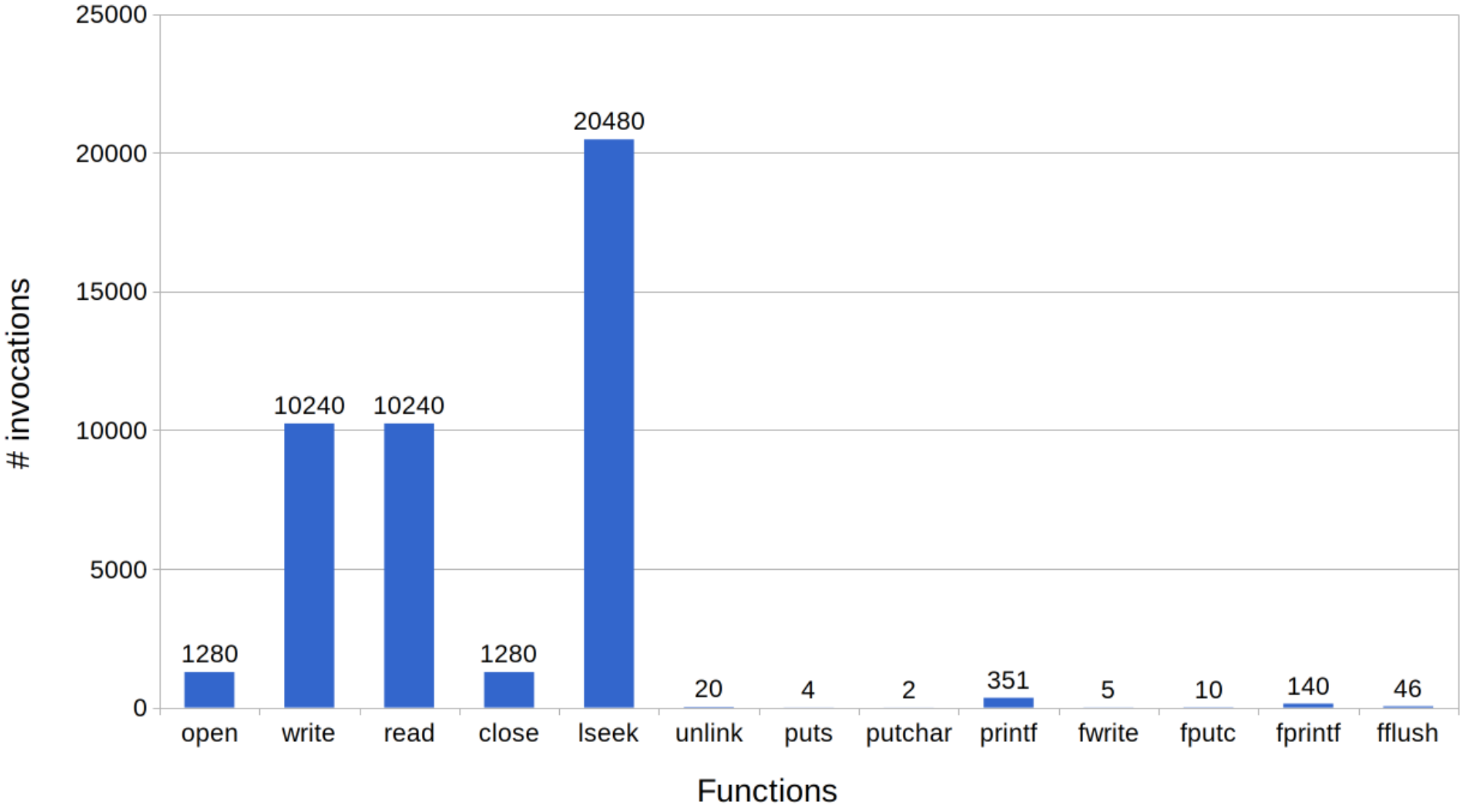
# Motivation



- Goal: Deploy a lightweight file system per job across all allocated nodes
  - Use unused node-local storage (RAMDisk, SSD, NVRAM, …)
  - Inputs are staged into the FS before job starts (Output vice versa)
  - Optimize the FS for the application's requirements
- Key assumptions:
  - Each FS object is accessed by a single application
  - Working data set fits into available node-local storage
  - Startup within seconds
  - Relax POSIX semantics, e.g., no sequentialized creates

# Function distribution





# Reducing the PFS load

- Stage in (read) and stage out (write) from/to PFS
- Absorb potentially harmful application I/O patterns from PFS in ADA-FS

“Bad” I/O pattern

Efficient I/O pattern

# Stage-in

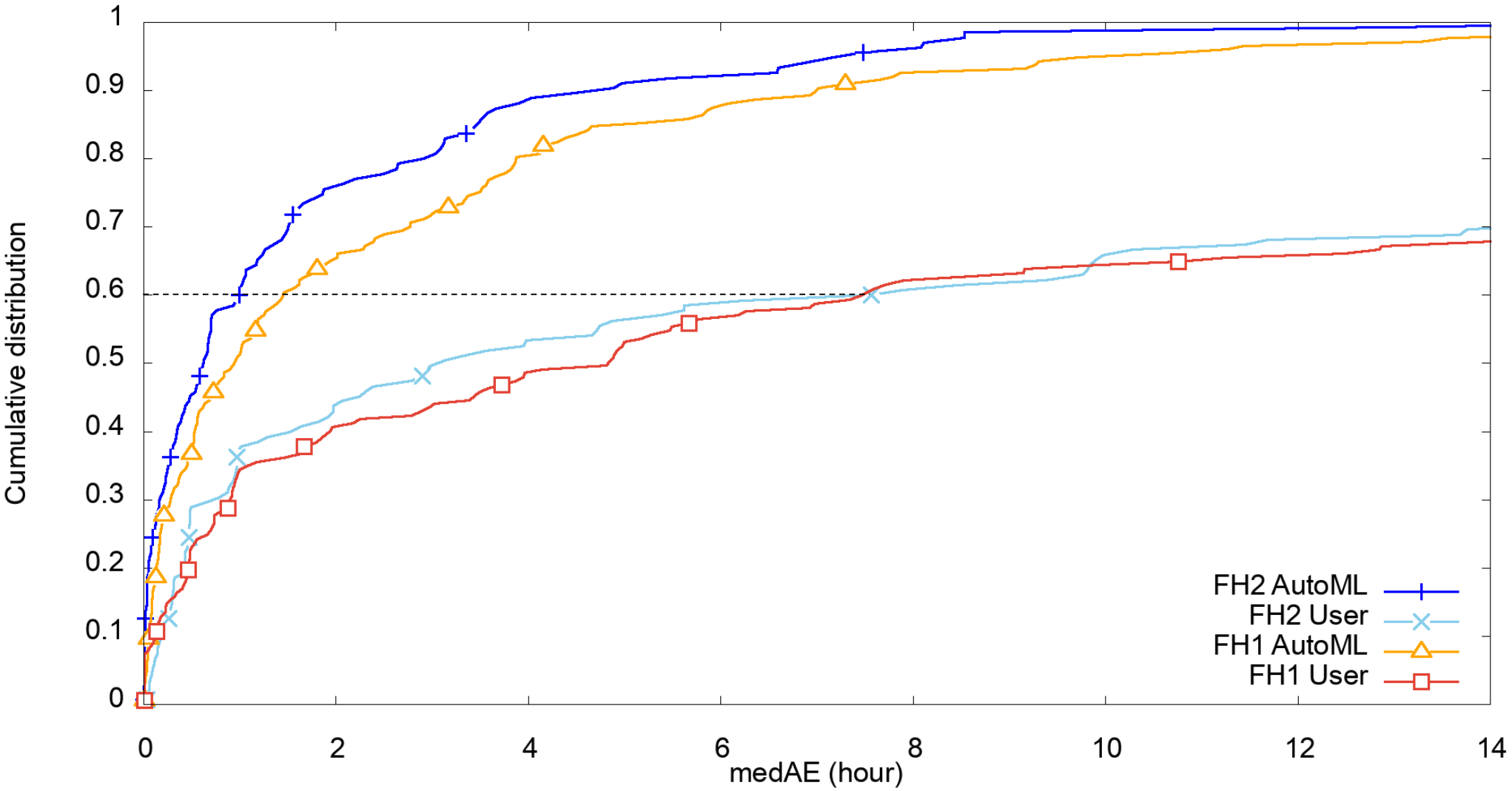


- Goal: Stage input data into ADA-FS before the job starts
  - The process must not disrupt running applications
- Accurate node allocation prediction is required
  - This requires accurate walltime predictions
- We implemented machine learning models for prediction (AutoML)
  - No further knowledge of an application is required
- Walltime accuracy increased significantly
  - E.g., 7.5 hours to 1 hour (medAE) for 60% of the users

*M. Soysal, M. Berghoff, A. Streit. Analysis of job metadata for enhanced walltime prediction. In Job Scheduling Strategies for Parallel Processing, JSSPP 2018 (accepted)*

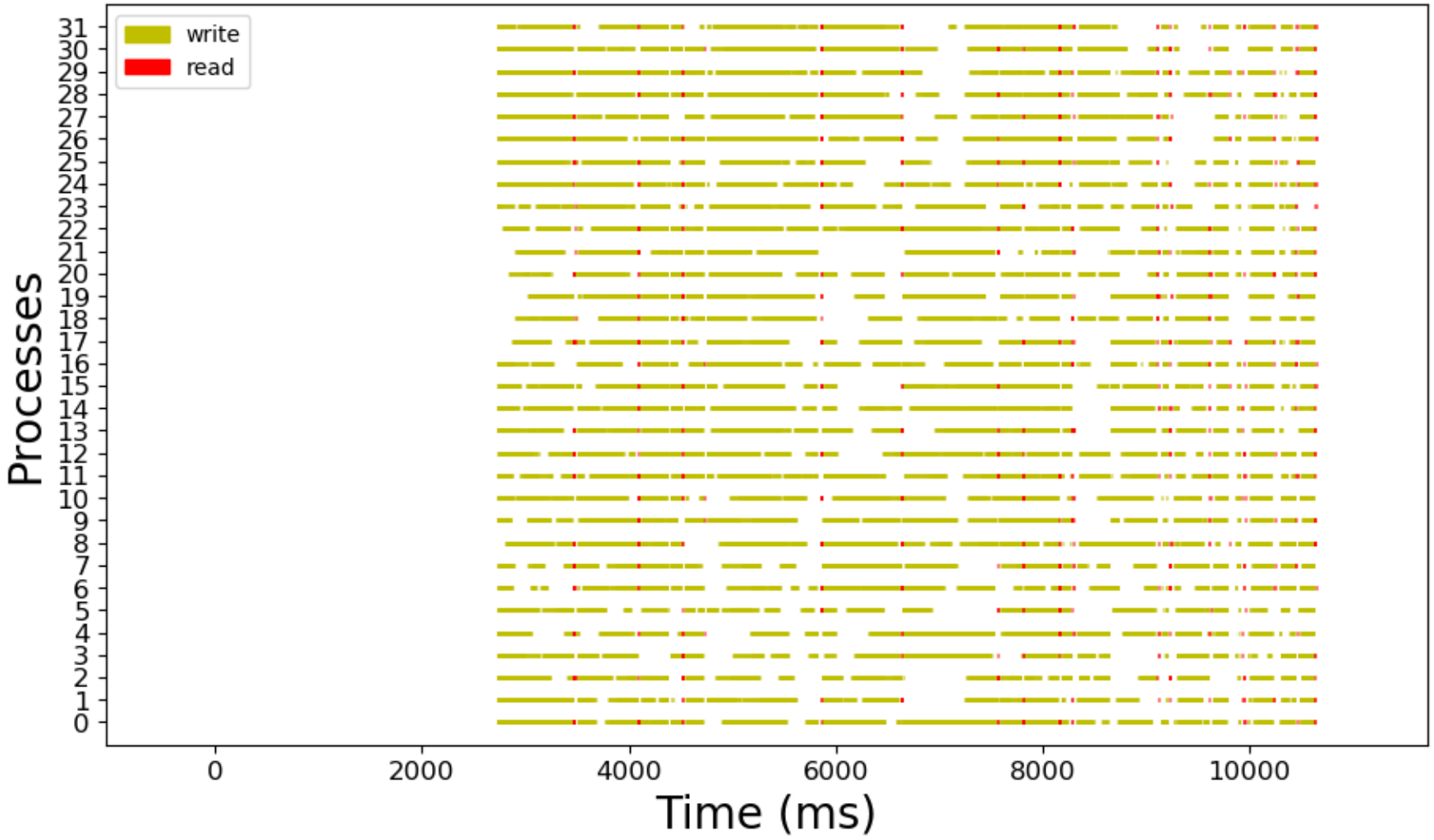
# Walltime prediction

## Median absolute error (h)



# I/O distribution over time

## IOR example



# File system design



- Strict consistency for direct operations on FS objects
- Key-Value store (one per node) handles metadata
- Node-local file system is used for data
- Data and metadata are distributed evenly across all job nodes
- No locking, no permission handling, and no fault tolerance

# File system design

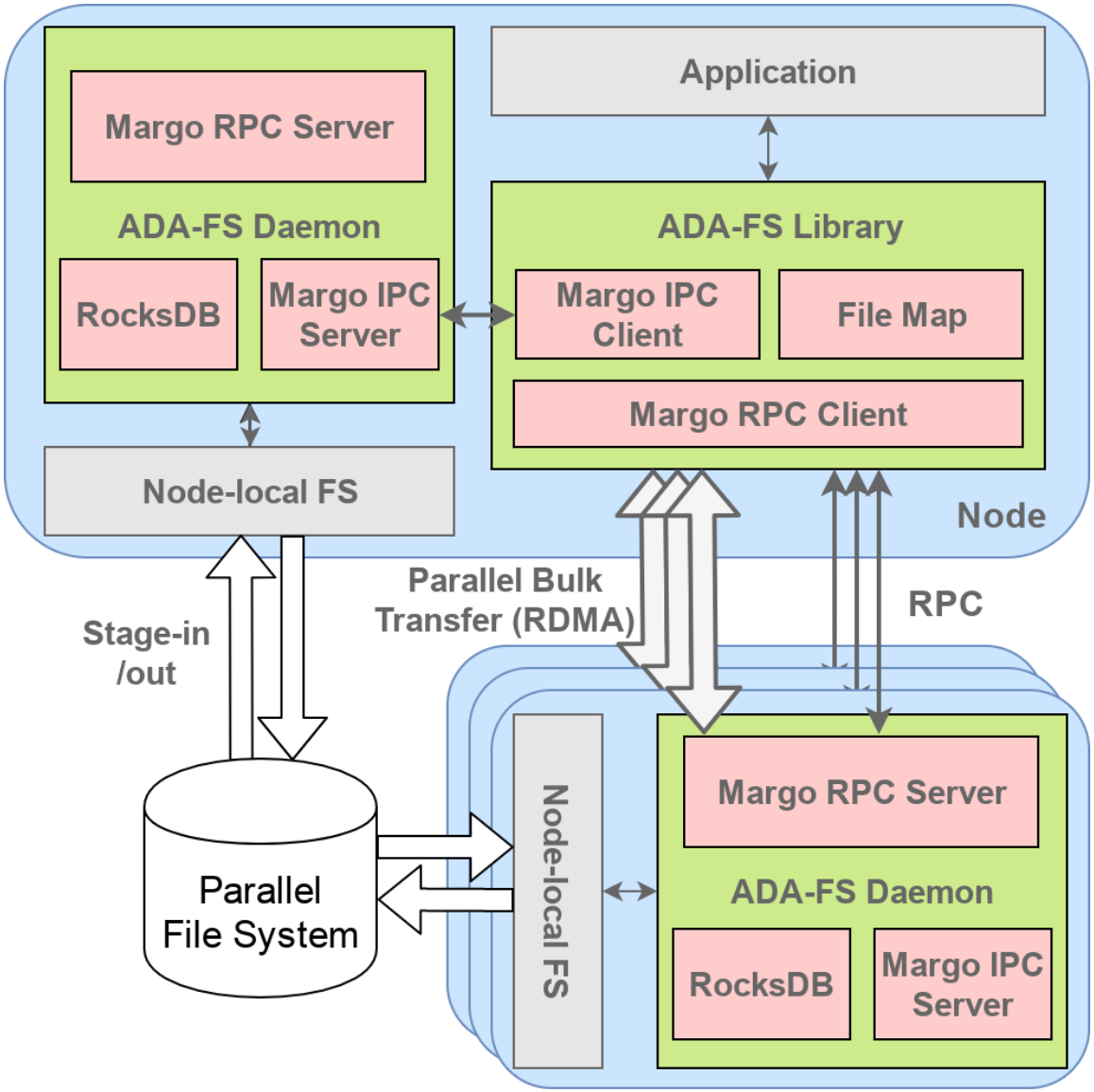


Let's rethink metadata handling in distributed file systems

- Directory/indirect blocks and inodes are *not* designed for parallel access
  - Leads to high code complexity, heavy communication, and intricate locking
  - Result: poor scalability (see common parallel file systems)
- Instead,
  - remove most metadata (timestamps, permissions, …),
  - compute metadata destinations on the fly, and
  - let the target node handle the request independently.

*M.-A. Vef, V. Tarasov, D. Hildebrand, A. Brinkmann. Challenges and solutions for tracing storage systems: A case study with Spectrum Scale. In ACM Transactions on Storage, 2018 (accepted)*

# File system architecture

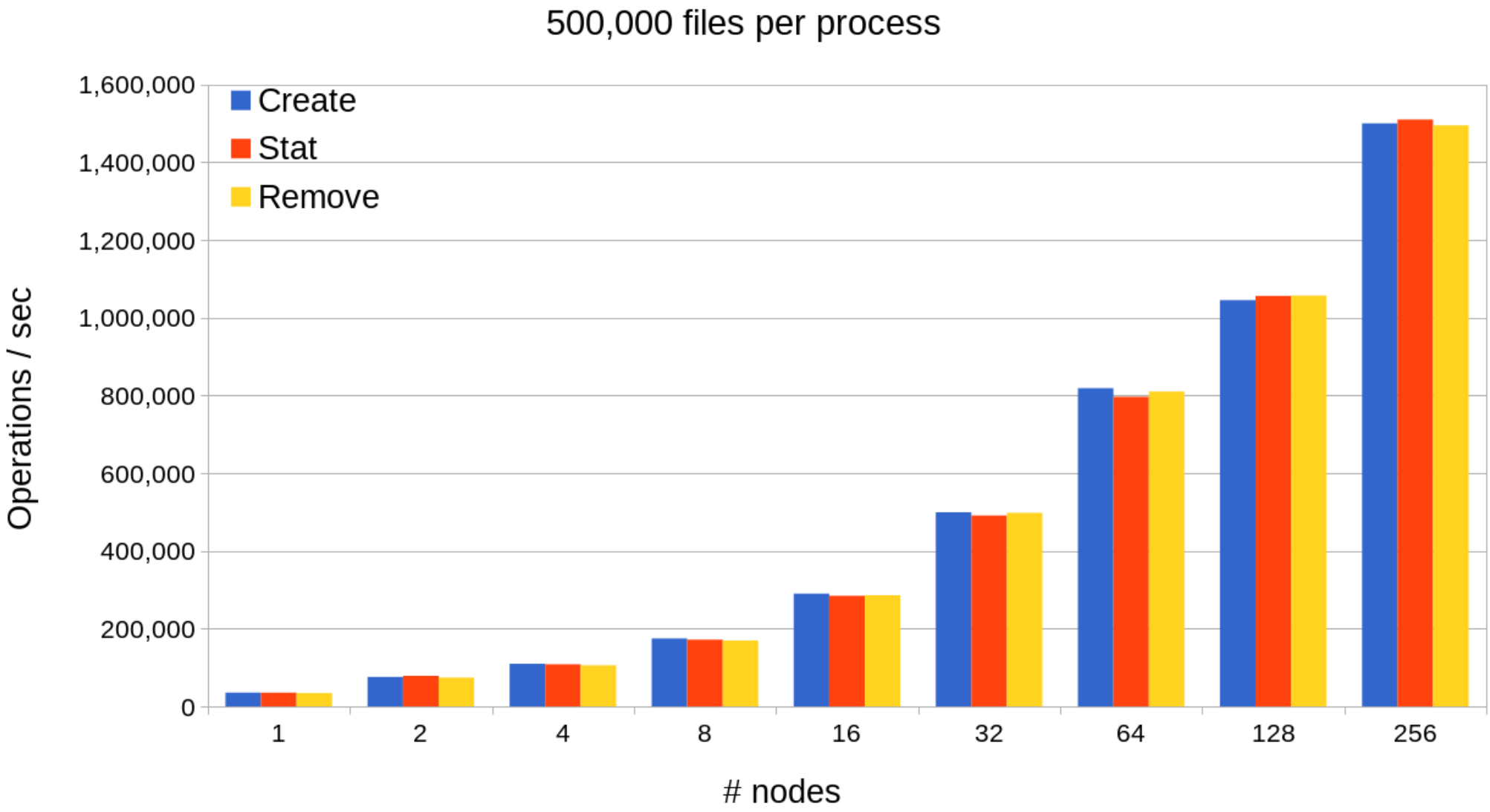


# Experimental Setup

- Experiments on *MOGON I* at JGU
  - 555 AMD Opteron 6272 (4·16 cores) available nodes
  - 128 GiB up to 512 GiB memory per node with Infiniband interconnect
- MDTest – metadata microbenchmark
  - 500k files are created/stated/removed per process in a single directory
- IOR – data microbenchmark
  - 256 MiB per process with a 4 MiB chunksize
- A RAMDisk on each node stores the metadata and data
- All operations are synchronous with no ADA-FS caching

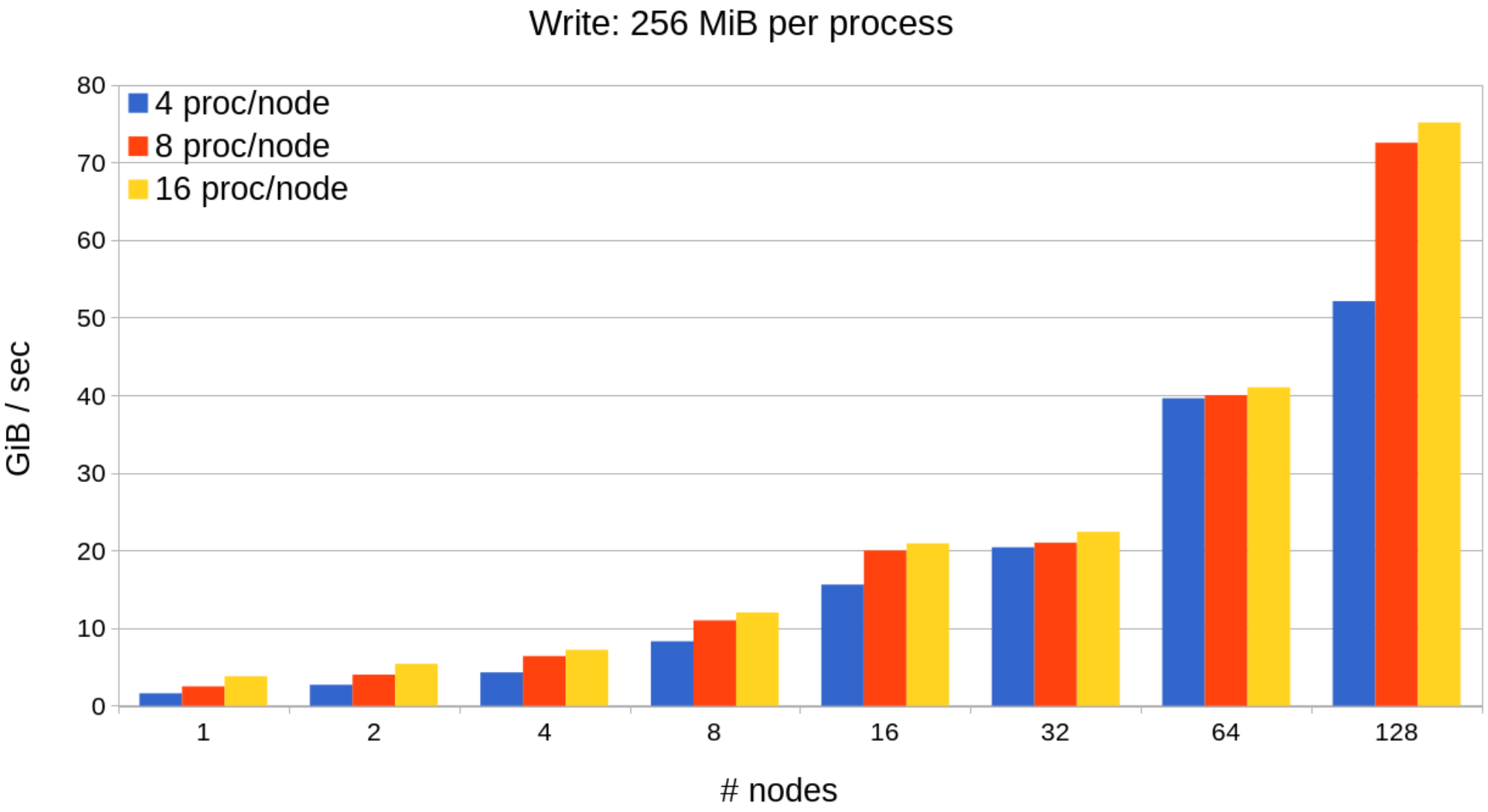
# Evaluation

## Metadata performance



# Evaluation

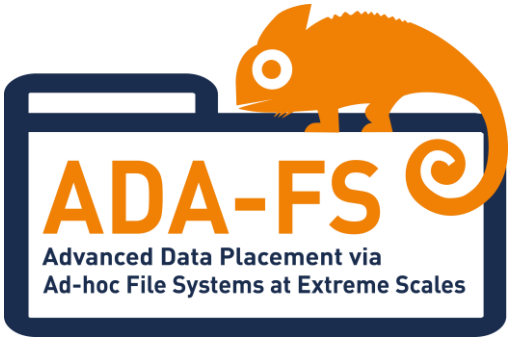
## Write throughput



# Future work

- Evaluate applications with ADA-FS
- Integrate monitoring into ADA-FS
- I/O phase characterization
- Integration of ADA-FS into batch system

# Thank you




Contact:


- Mailinglist: <ada-fs-all@fusionforge.zih.tu-dresden.de>
- Marc-André Vef <vef@uni-mainz.de>
- Sebastian Oeste <sebastian.oeste@tu-dresden.de>
- Mehmet Soysal <mehmet.soysal@kit.edu>

# Acknowledgements


- DFG for funding ADA-FS project under the SPPEXA
- We gratefully acknowledge being able to use several compute clusters:
  - Steinbuch Centre for Computing (SCC) @KIT
  - Johannes Gutenberg University
  - TU Dresden