Vef MA, Moti N, S¨uß T et al. GekkoFS — A temporary burst buffer file system for HPC applications. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 35(1): 72–91 Jan. 2020. DOI 10.1007/s11390-020-9797-6

# GekkoFS

## A temporary burst buffer file system for HPC applications

Marc-André Vef*, Nafiseh Moti, Tim Süß*, Markus Tacke, André Brinkmann*, Tommaso Tocci†, Alberto Miranda†, Ramon Nou†, Toni Cortes† §

* Johannes Gutenberg University Mainz, Center for Data Processing, Mainz, Germany

† Barcelona Supercomputing Center, Barcelona, Spain

§ Universitat Politenica de Catalunya, Computer Architecture Department, Barcelona, Spain

JOHANNES GUTENBERG UNIVERSITÄT MAINZ
JG|U
BSC Barcelona Supercomputing Center Centro Nacional de Supercomputación
UPC Departament d'Arquitectura de Computadors UNIVERSITAT POLITÈCNICA DE CATALUNYA
DFG Deutsche Forschungsgemeinschaft German Research Foundation
SPPEXA
nextgenio
ADA-FS Advanced Data Placement via Ad-hoc File Systems at Extreme Scales

# Motivation

- HPC is increasingly more used by data-driven science applications
- Data-driven workloads impose new requirements on HPC file systems
  - Many metadata operations, random access, small I/O requests …
- Solutions are software (e.g., ADIOS) or hardware (burst buffers) based
- Burst buffer file systems use node-local storage to accelerate I/O
  - But, they are often full or near POSIX-compliant

# Goal

- Goal: Deploy a lightweight file system per job across all allocated nodes
  - Temporary lifetime, e.g., for HPC job or campaigns
  - Use unused node-local storage (RAMDisk, SSD, NVRAM, …)
  - Inputs are staged into the FS before job starts (Output vice versa)
  - Relax POSIX semantics, e.g., no sequentialized creates
  - Offer only FS features which are actually required by most (not all) applications
- Application assumptions:
  - Each FS object is accessed by a single application
  - Working data set fits into available node-local storage

# File system design

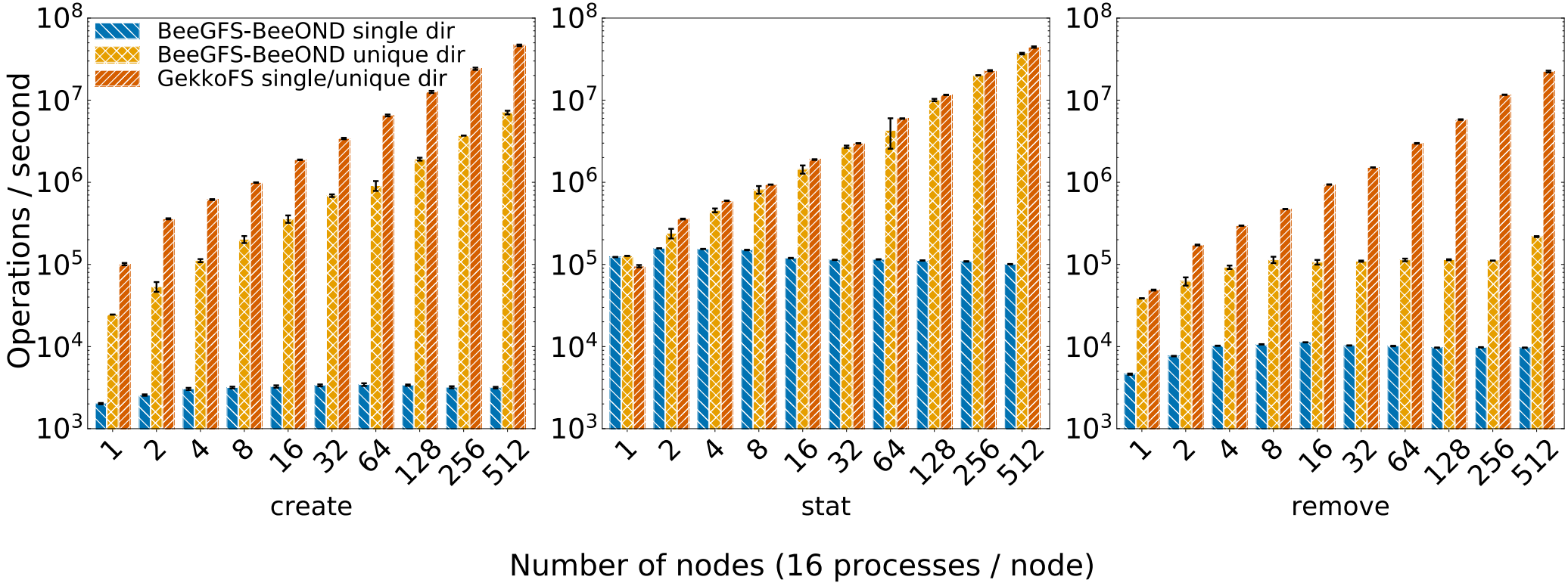
## Key properties

- Data and metadata are distributed evenly across all job nodes
  - Key-Value store (one per node) handles metadata
  - Node-local file system is used for data
- Strong consistency for direct operations on FS objects
- Synchronous and no cache mechanisms
- No internode-locking and no permission handling
- File I/O is split into equally sized chunks (configurable)
- The node destination of each chunk is computed on the fly
- Chunks are mapped to files in the node-local file system

# Evaluation

## Metadata performance

- GekkoFS and BeeGFS weakly scaled (100K files per process)
  - More than 819 million files in total with 512 nodes



# Evaluation

## Sequential I/O throughput vs. BeeGFS

Seq. write

Seq. read