JOHANNES GUTENBERG
UNIVERSITÄT MAINZ

ErUM Data – Application from Mainz

Prof. Dr.-Ing. André Brinkmann

Dr. Dalibor Djukanovic

Prof. Dr. Frank Maas

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

JG|U

From FLOPS to IOPS

- Proposal from Bonn focuses on using opportunistic resources from huge HPC centers
- Most HPC Clusters built for high FLOPS
- ErUM Data workload needs high IOPS

- SSD can deliver high IOPS, however
  - usually only used for Metadata if at all (Euro/Tbyte) or as
  - "OS" disk for fast boot of compute nodes

- Aim of the application from Mainz: Opportunistic use of resources for our workloads -> run at scale at HPC centers -> use opportunistic file systems

JG|U

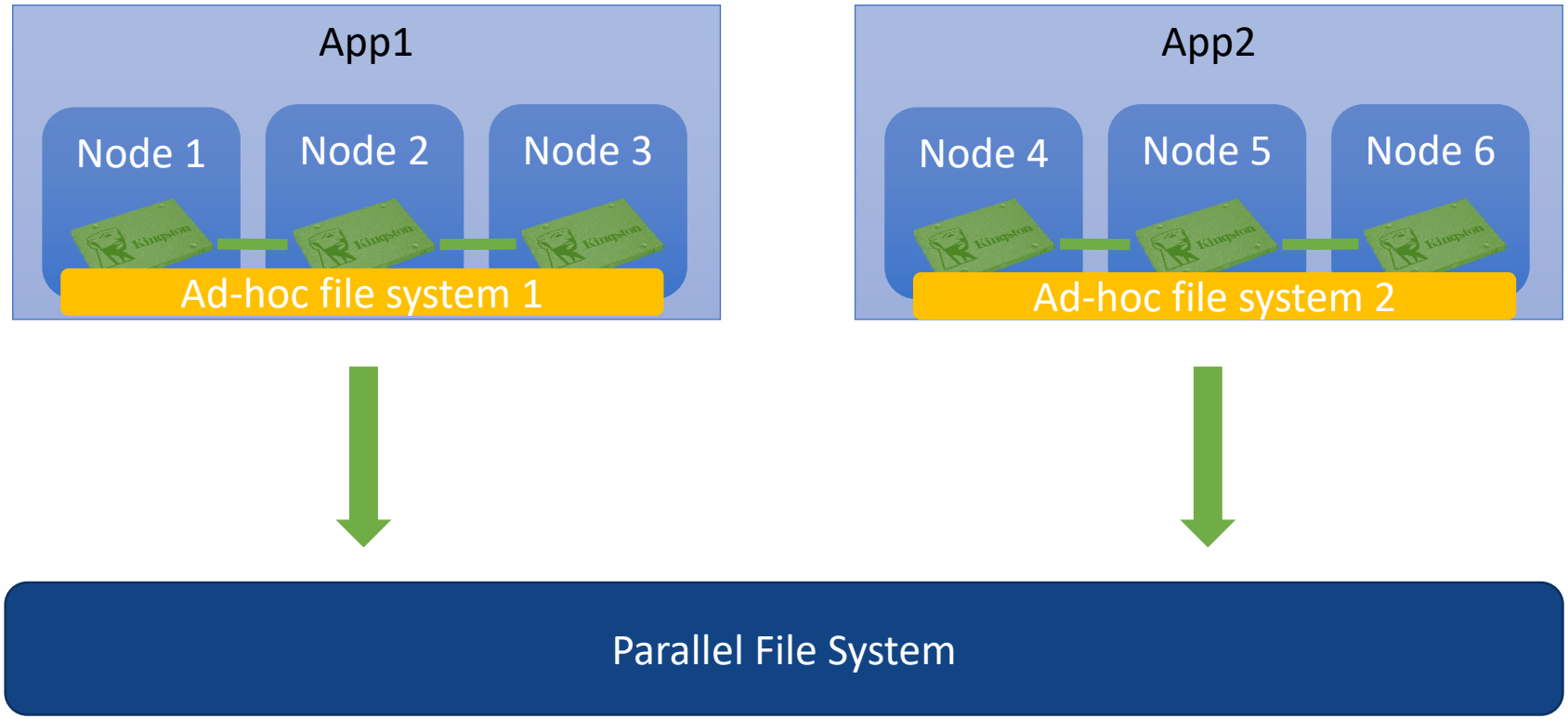
Workload

- Trivially parallelizable jobs
  - typically few calculations on a lot of data
  - sometimes problematic access patterns, i.e. random I/O
- Scaling to thousands of jobs makes even sequential I/O look like random I/O for the central storage system
- Solutions:
  - Build machine with high throughput in mind (does not apply to classic HPC centers)
  - Copy files to local disk space -> Problem getting the jobs to the correct nodes
  - Use parallel FS built from compute nodes resources -> Ad-hoc parallel FS

Benefits of the Ad-hoc FS using compute nodes

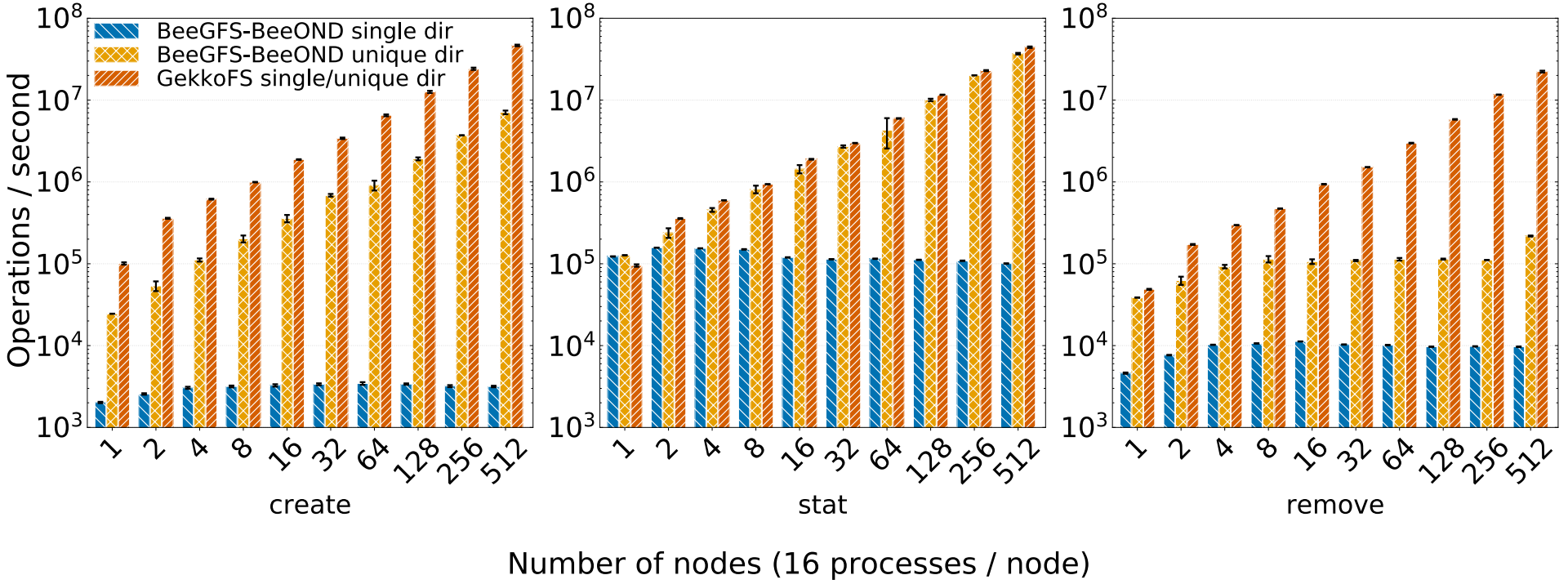
- Building FS using compute nodes resources, e.g. local SSDs, leverages these resources for trivially parallelizable work loads
- Remove load from central long-term storage
- Scaling properties of Ad-hoc FS better because of (potentially) large number of storage targets with huge aggregated IOPS performance

Ad-hoc File Systems



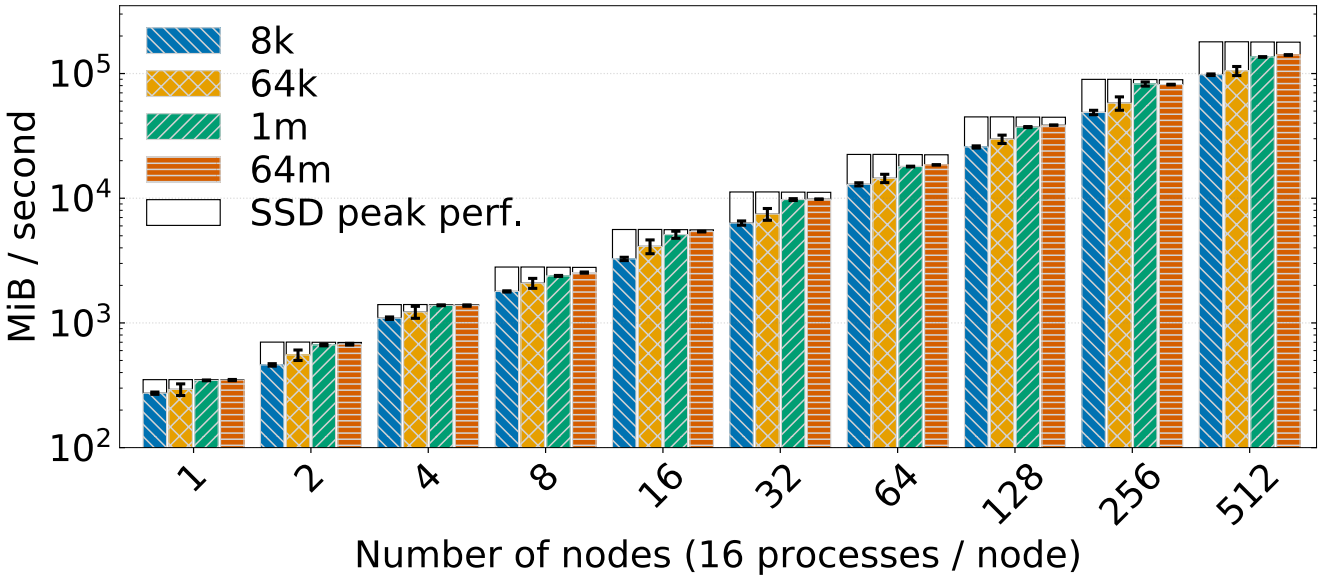
Metadata Performance

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

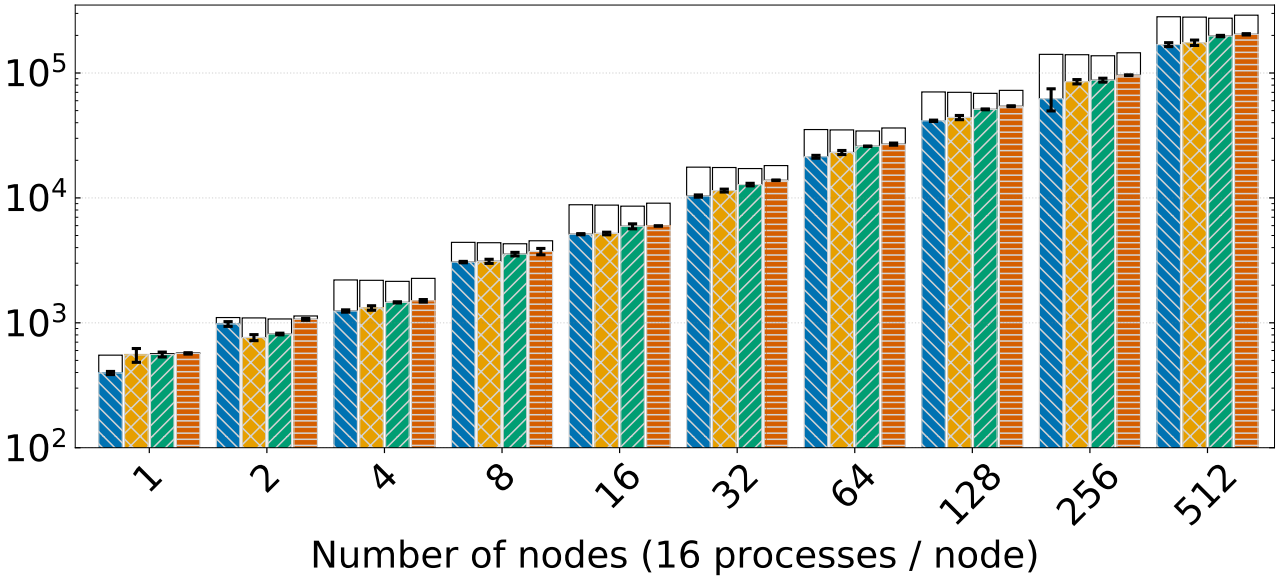


Sequential I/O throughput (file per process)

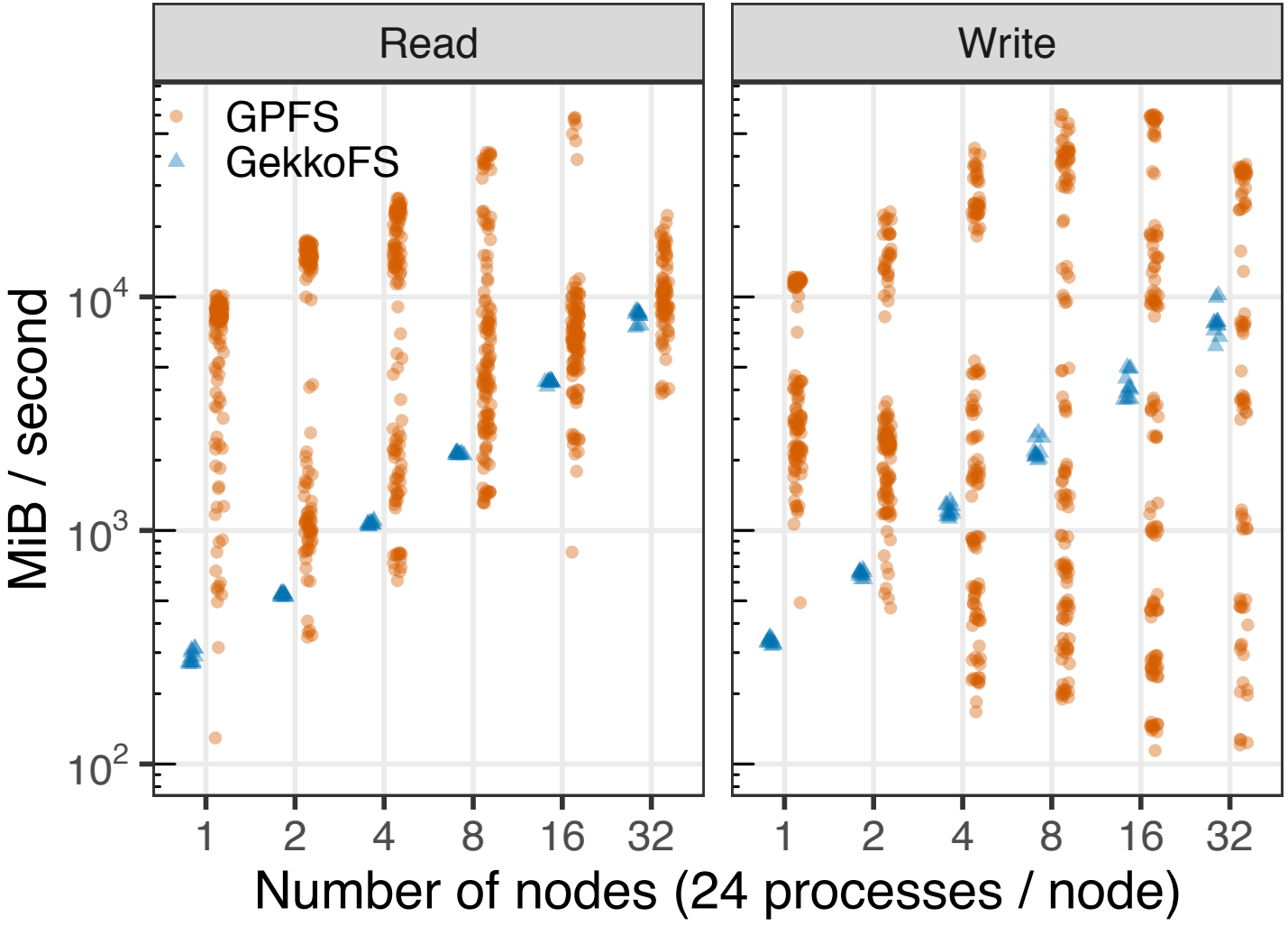
Seq. write



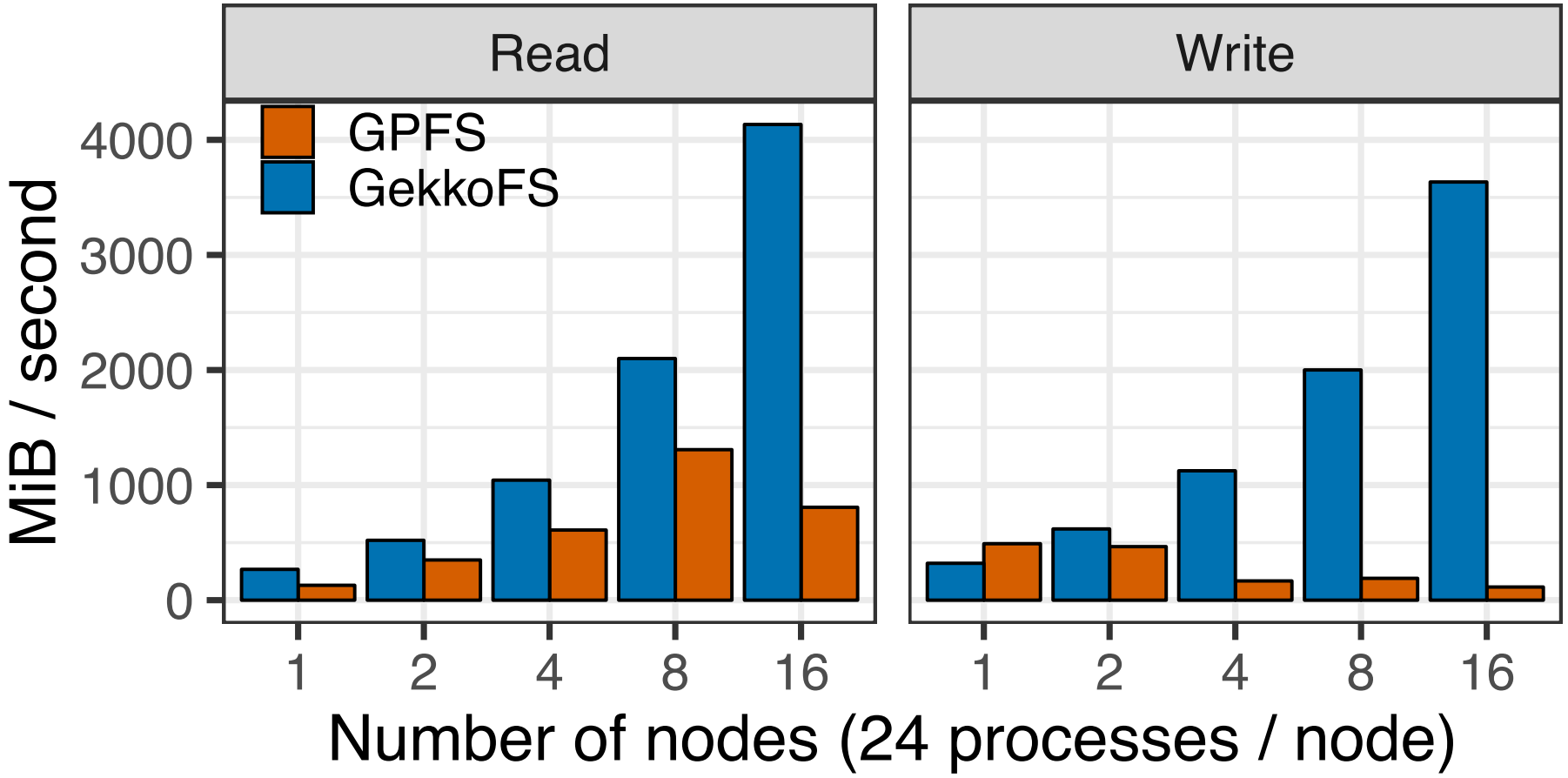
Seq. read



Test at BSC: Variability



Test at BSC: Worst-case performance



Ad-hoc file systems and Opportunistic Resources

- Additions from Mainz proposal:
  - Extend schedulers to stage in useful data, while job is already accumulating priority via wait time
  - Make the Ad-hoc FS less ad-hoc, i.e. keep the user generated PFS alive over several jobs

Thank you for your attention

JOHANNES GUTENBERG
UNIVERSITÄT MAINZ

JG|U