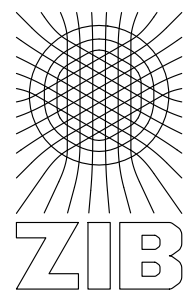
Improving I/O Performance Through Colocating Interrelated Input Data and Near-Optimal Load Balancing

Felix Seibert, Mathias Peters, and Florian Schintke

4th HPBDC Workshop 2018, Vancouver

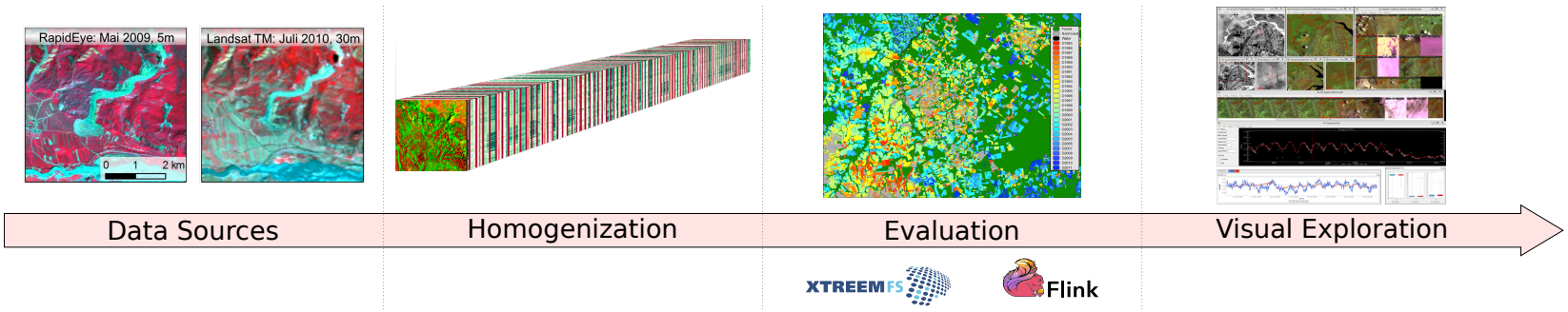


seibert@zib.de

Overview

Background: Application

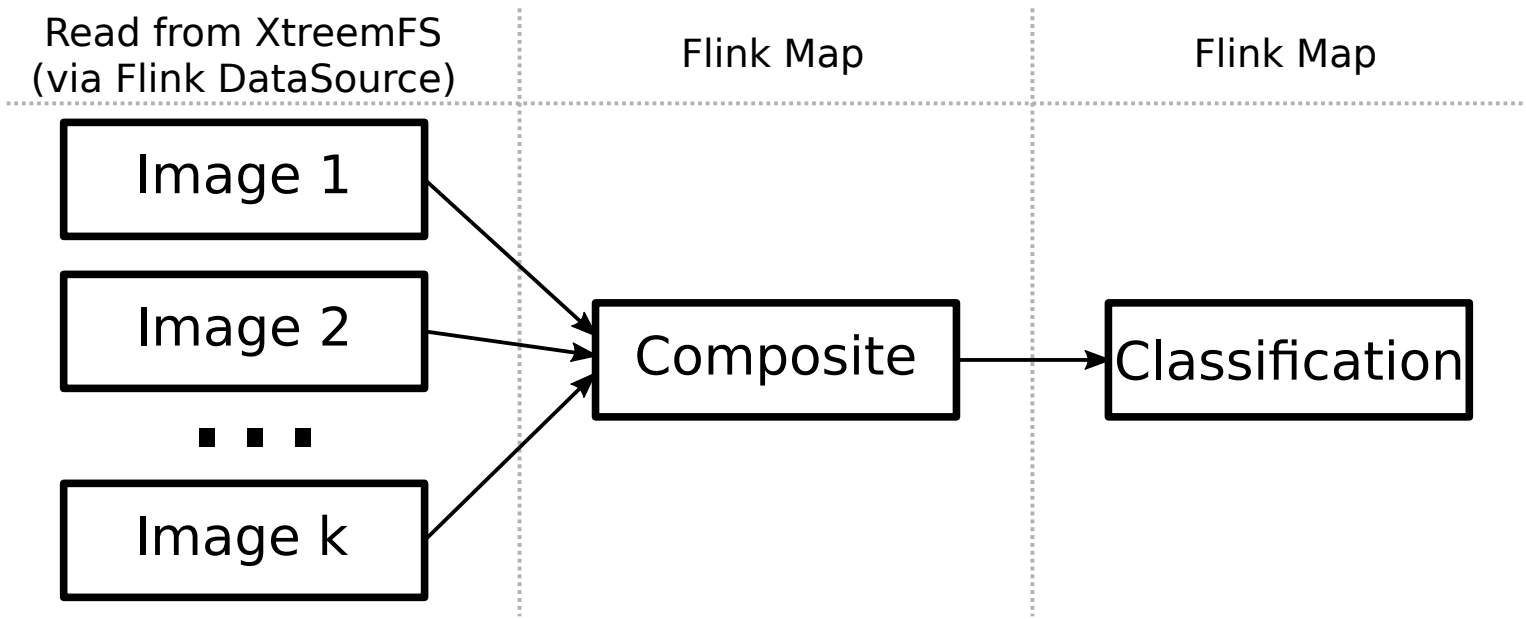
- GeoMultiSens: Analyze the Earth's surface based on remote sensing images



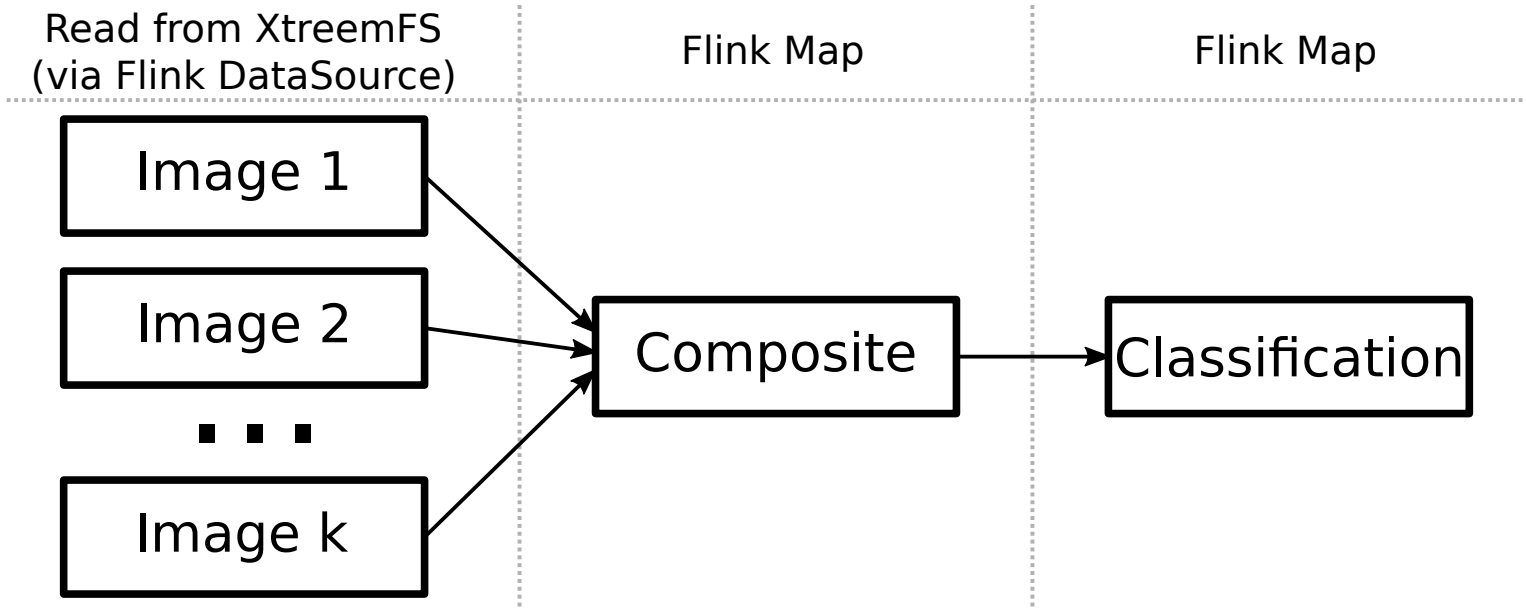
- The more images, the better the results (several PB available)
- Implementation as MapReduce job in Flink
- **Goal: distribute files in the distributed file system (XtreemFS) such that the computation is efficient and performant**

Background: Application Details (1)

- Data parallel application
- Parallelization along geographical regions (UTM grid)



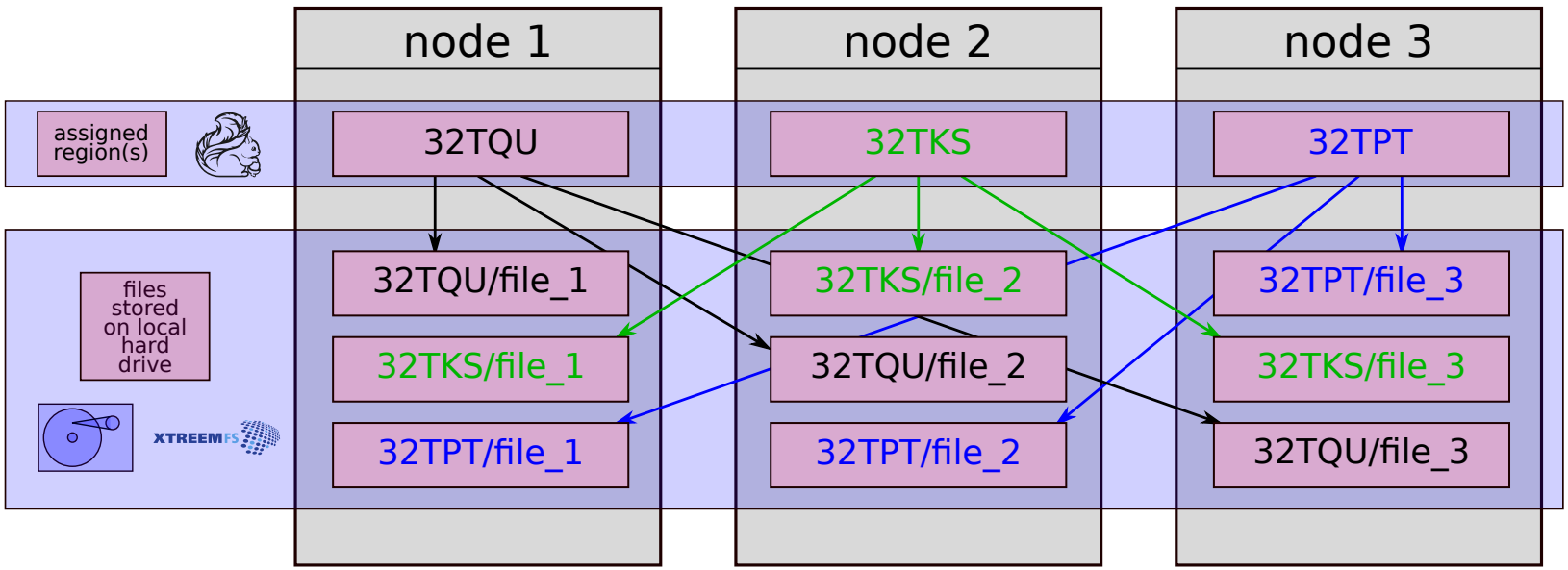
Background: Application Details (2)



- composites have large memory footprint
- (de)serialization is expansive (python $\leftrightarrow$ java)
- $\Rightarrow$ composites should not be moved
- $\Rightarrow$ analysis of one region (group) should be on one node

File Placement Issues (1)

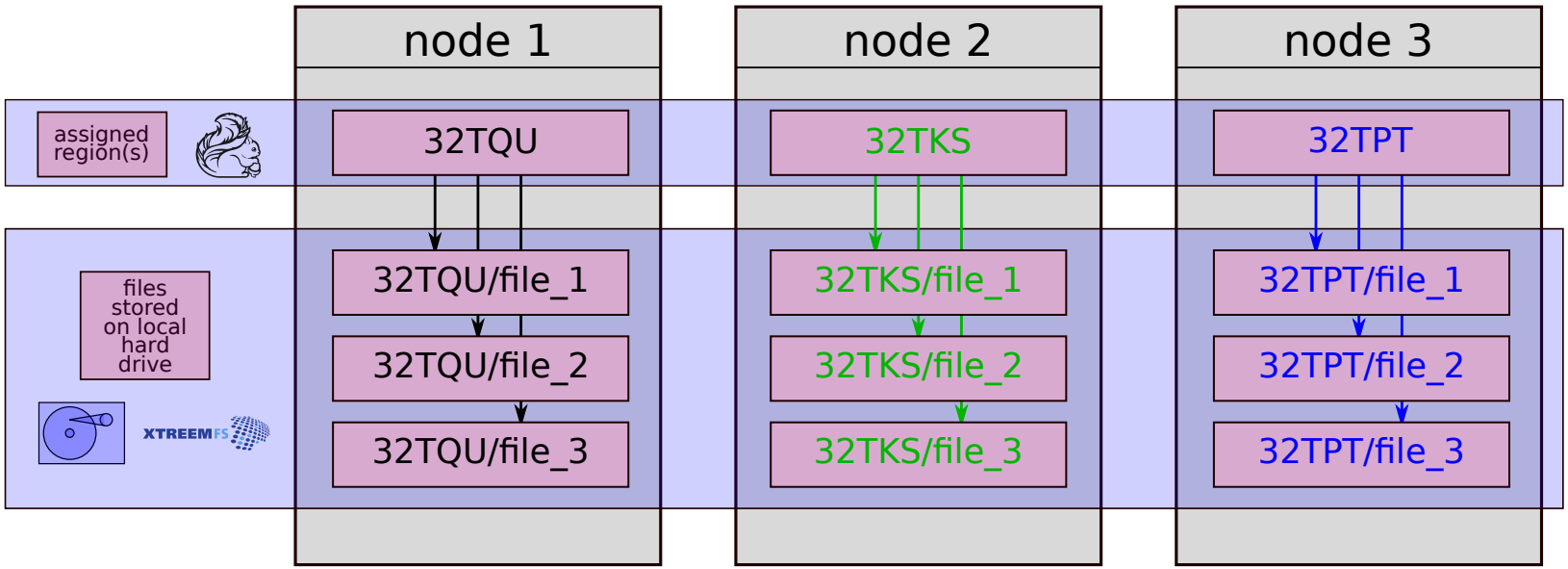
- State of the art: more or less random distribution
  $\Rightarrow$ no data local processing possible



- Network traffic
- Disk scheduling

File Placement Issues (2)

- Goal: colocated UTM regions (groups) for local access



- Network traffic
- Disk scheduling
- $\Rightarrow$ local grouping
- $\Rightarrow$ load balancing issues (Europe: 1400 groups between 3 MB and 25 GB)

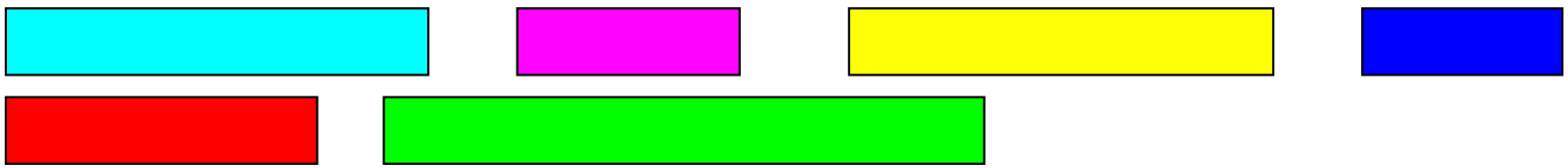
Hybrid placement optimization strategy

- place all files of the same group in same, tagged folder
- distributed file system places all files of same group on same server

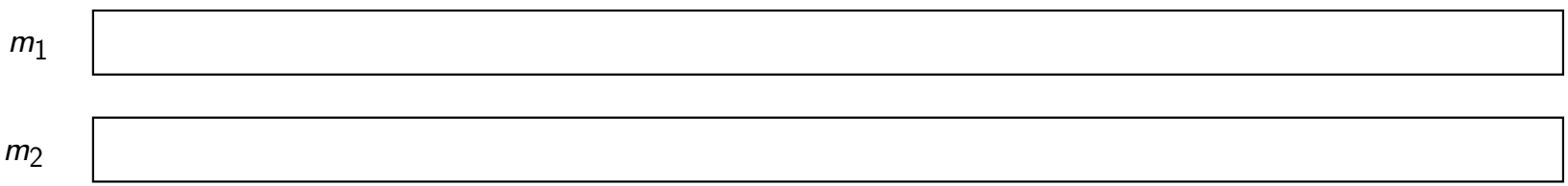
- load balancing (Storage Server Assignment Problem) is NP hard
- thus, use approximation algorithm

Storage Server Assignment Problem (1)

File groups (regions)

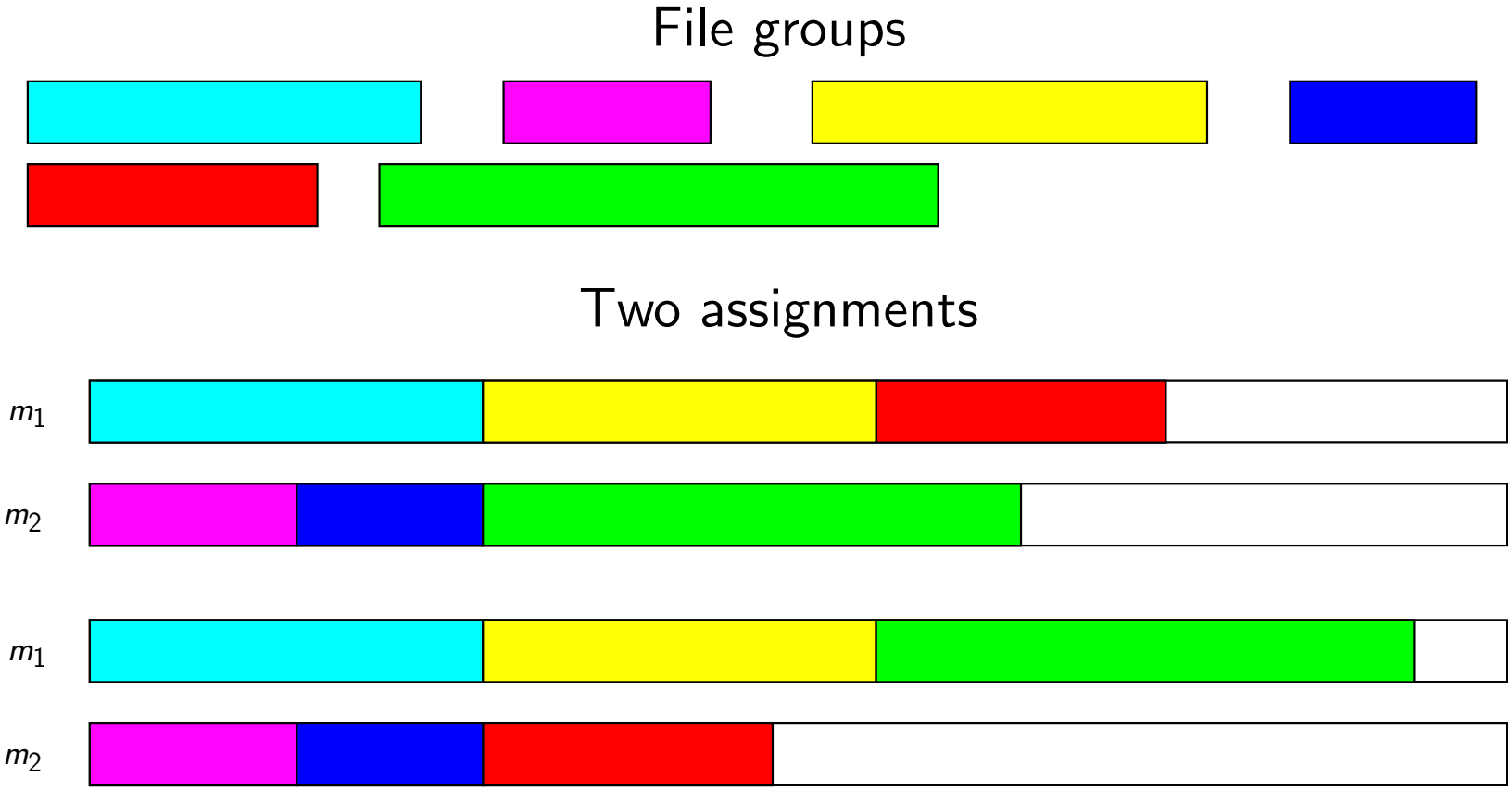


Storage machines (OSDs)



Goal: Assign file groups to machines such that the most loaded machine is loaded as little as possible

Storage Server Assignment Problem (2)



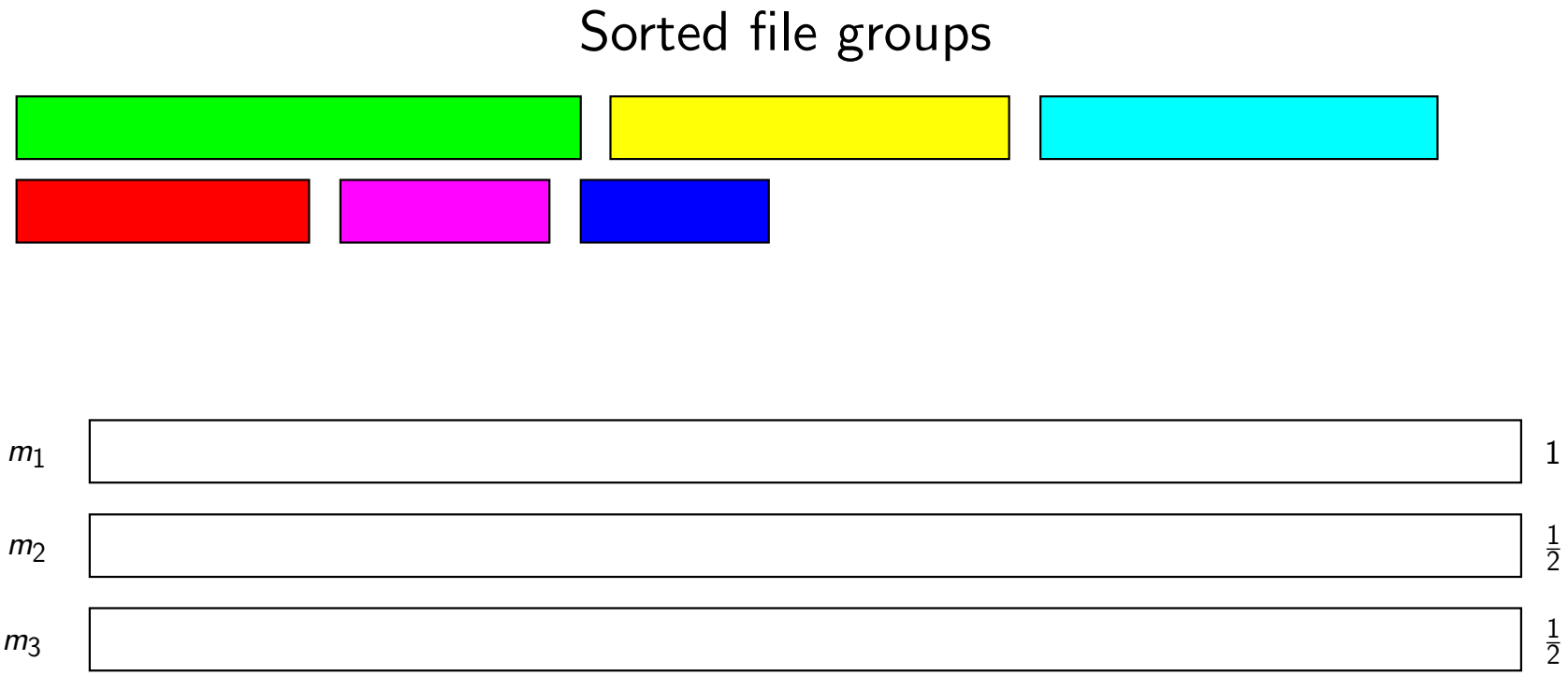
- our assignment problem is equivalent to multi-processor scheduling
- Approximation algorithm: Largest Processing Time first (LPT) becomes largest group size first in our scenario

LPT - Formal Description

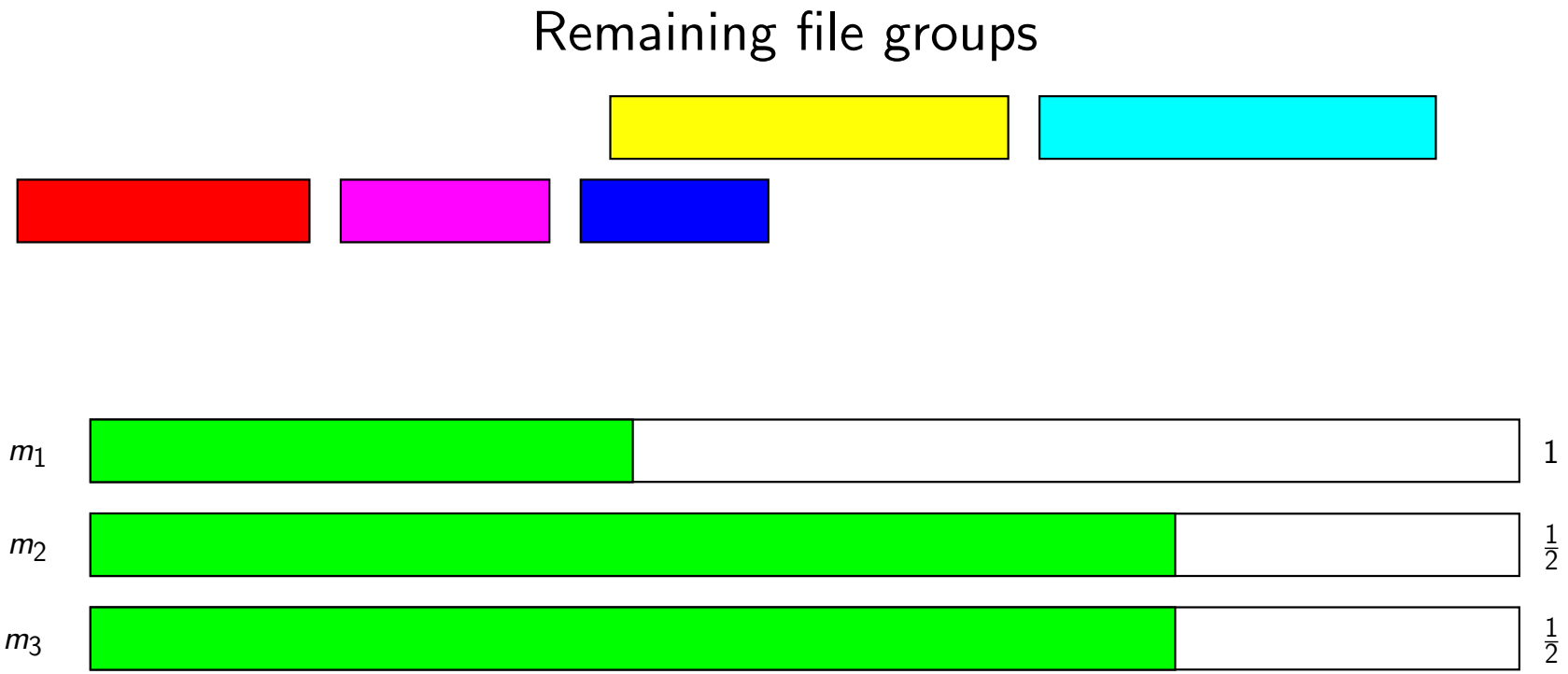
- $\mathcal{S}$ set of storage servers (OSDs) with capacities $c : \mathcal{S} \to \mathbb{N}$
- $\mathcal{F}$ set of file groups with sizes $s : \mathcal{F} \to \mathbb{N}$
- Sort $\mathcal{F} = \{f_1, \ldots, f_n\}$
  such that $s(f_i) \geq s(f_j)$ for all $1 \leq i < j \leq n$
- S_i denotes the storage server assigned to group f_i
- for $i = 1, \ldots, n$, S_i is given by

$$S_i = \arg\min_{S \in \mathcal{S}} (\ell(S) + \frac{s(f_i)}{c(S)})$$

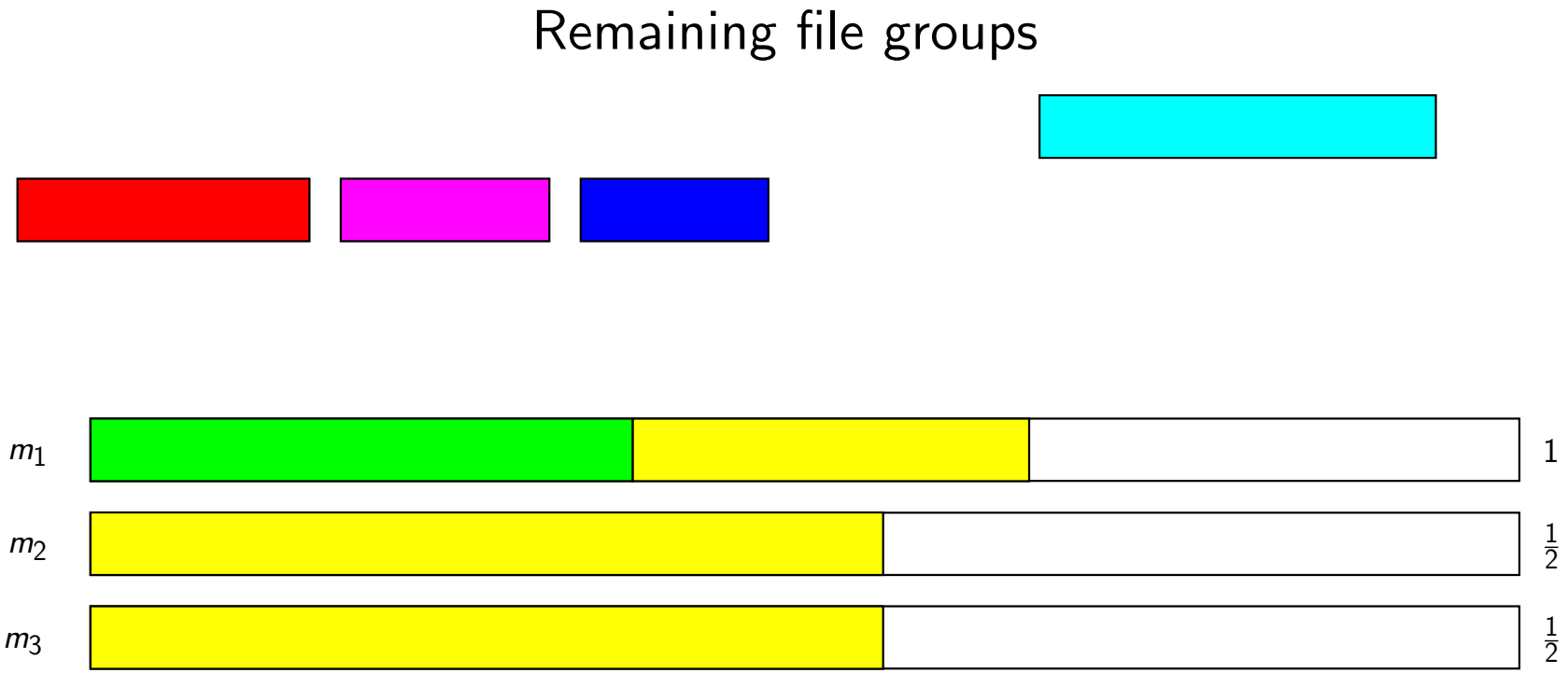
LPT - Step by Step Example (1)



LPT - Step by Step Example (2)



LPT - Step by Step Example (3)

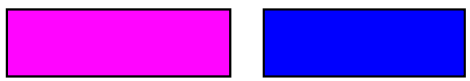


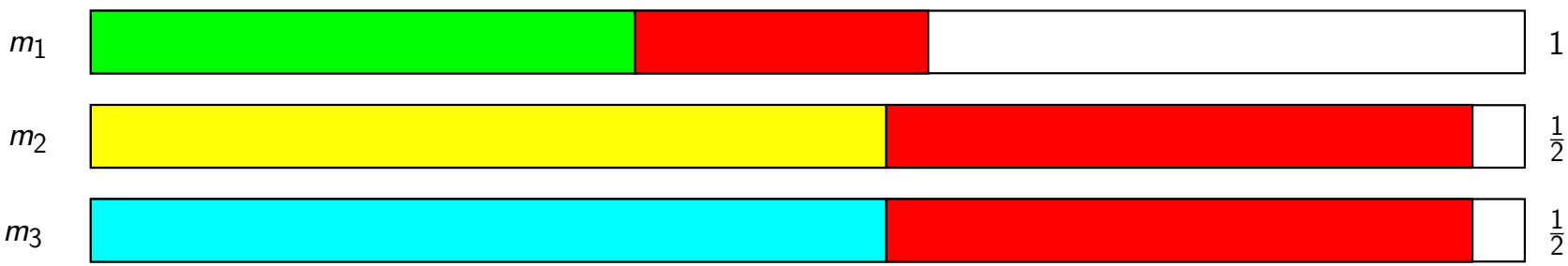
LPT - Step by Step Example (4)

Remaining file groups

LPT - Step by Step Example (5)

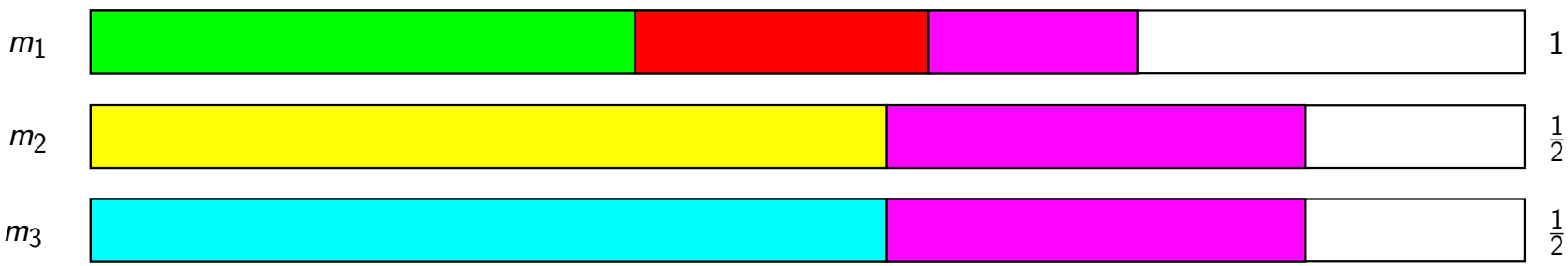
Remaining file groups





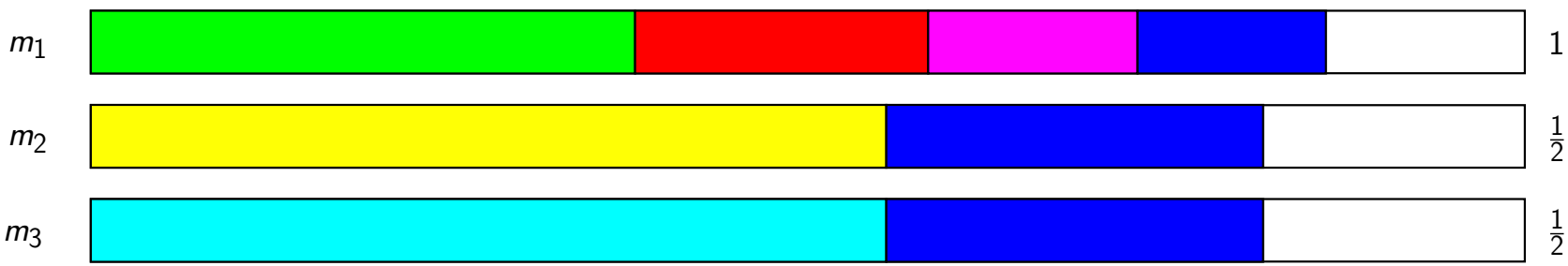
LPT - Step by Step Example (6)

Remaining file groups



LPT - Step by Step Example (7)

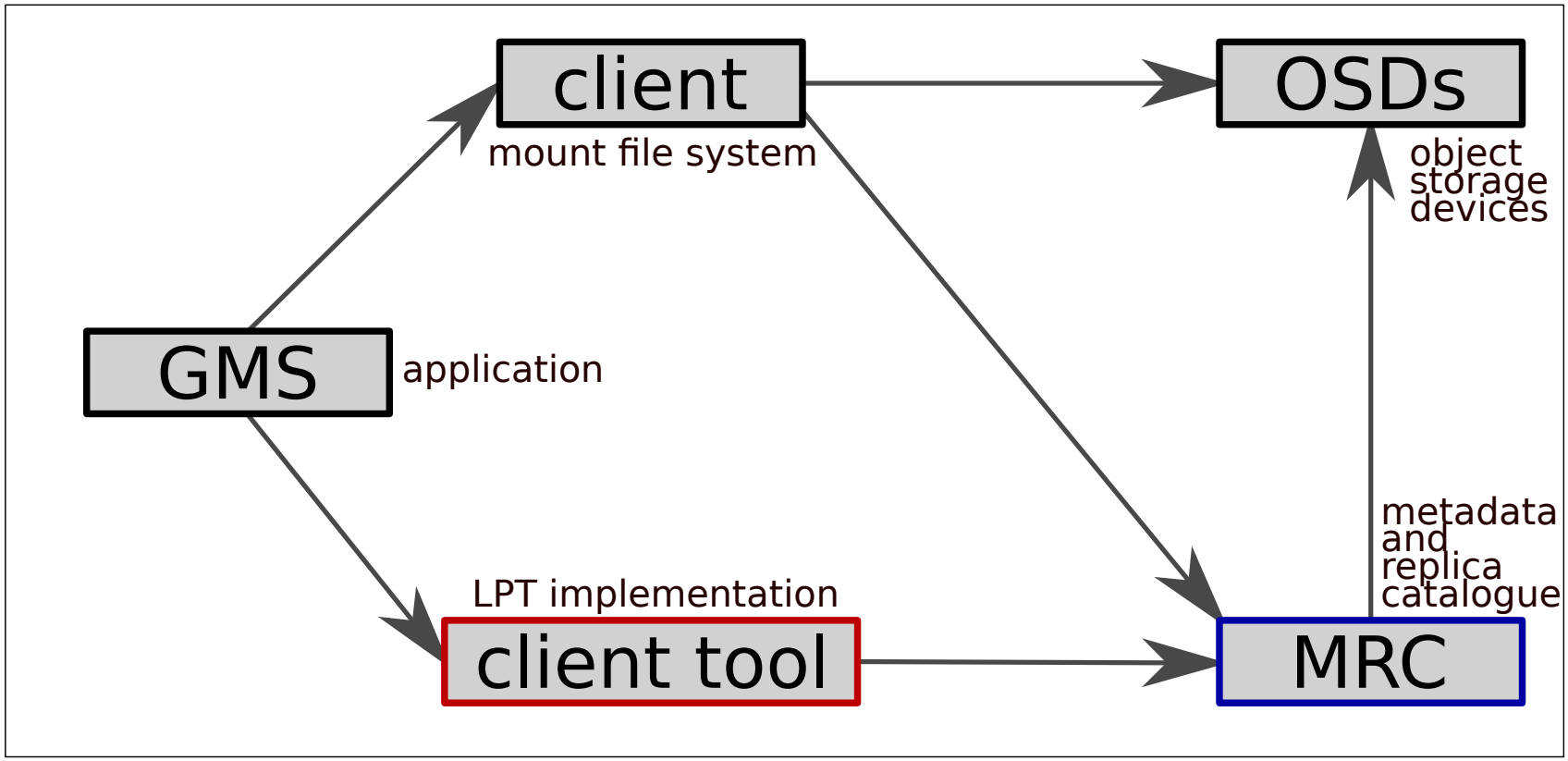
Remaining file groups



LPT: Key Properties

- simple and fast algorithm
- suitable for offline and online problems
- good theoretical performance
- practical evaluation: differs less than 1 % (offline) / less than 5 % (online) from the optimal solution

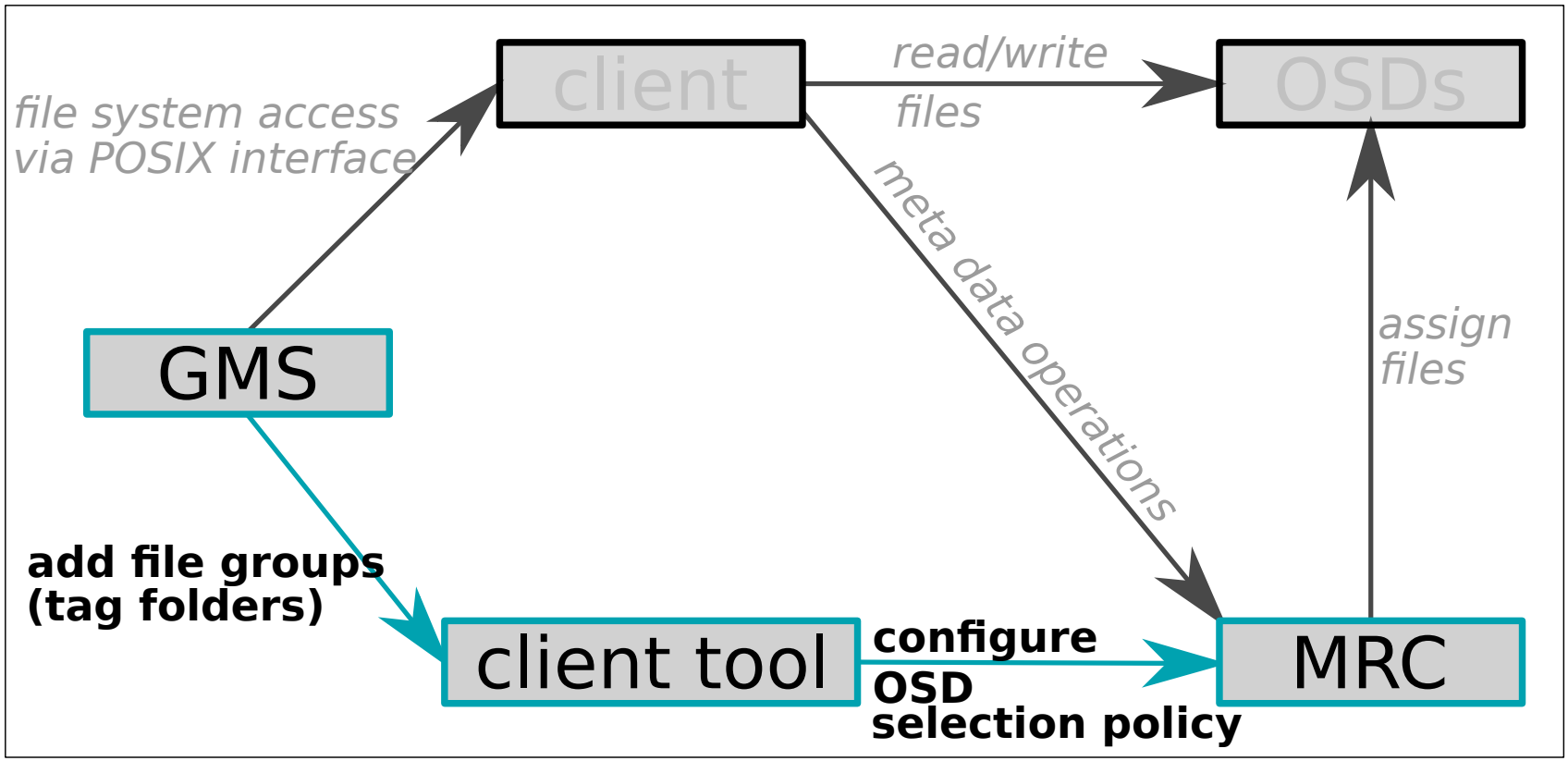
Implementation: Architecture



new code:

- client tool
- OSD selection policy for MRC

Implementation: Add Group(s)

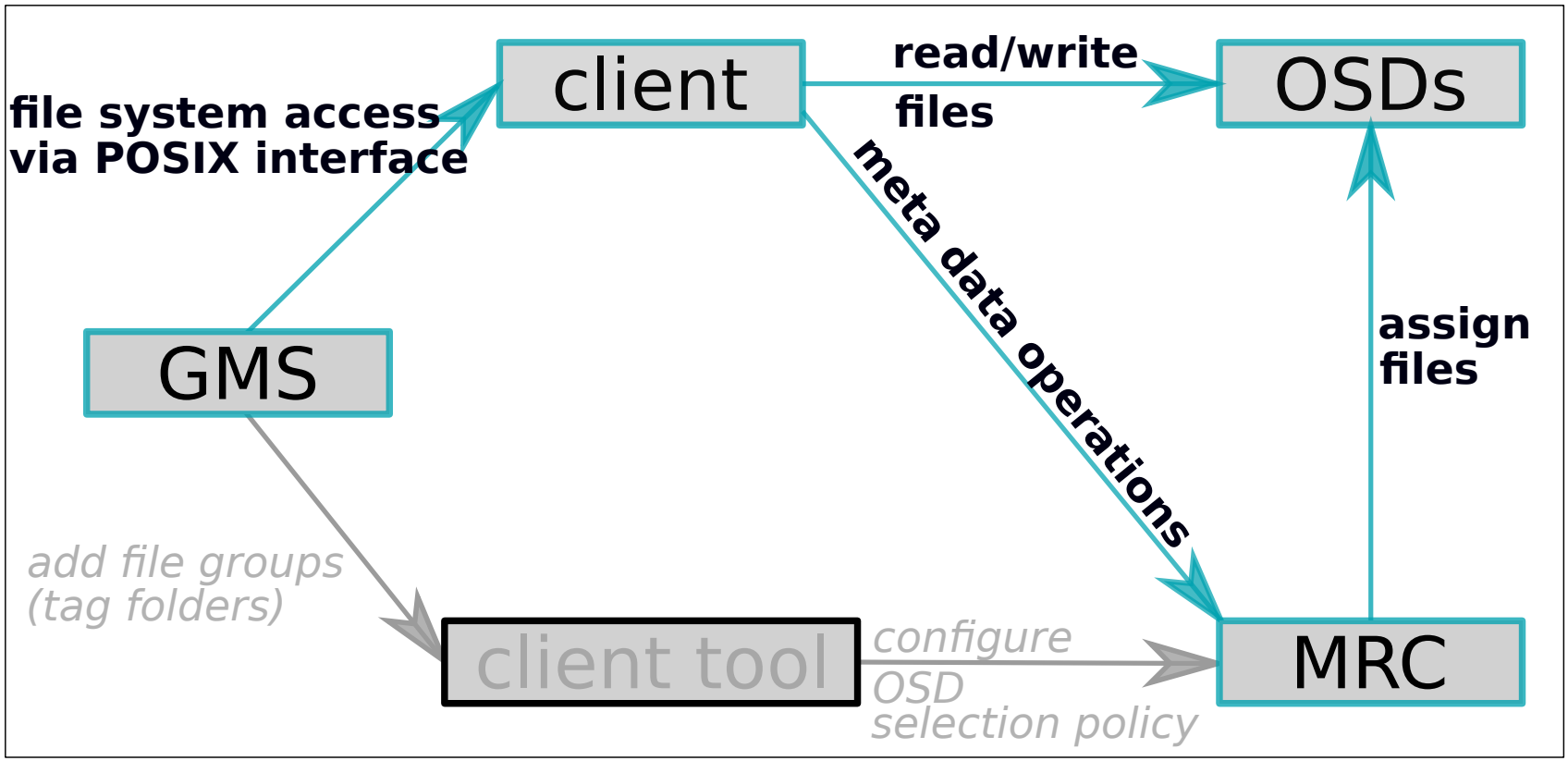


add_folders(/path/to/xtreemfs_mount/some/subdirs/32TQU)

$\Rightarrow$ MRC adds mapping entry:

some/subdirs/32TQU $\rightarrow$ OSD_17

Implementation: Add File(s)



open(/path/to/xtreemfs_mount/**some/subdirs/32TQU/LC8/file.tif**)

$\Rightarrow$ MRC finds match for prefix **some/subdirs/32TQU**

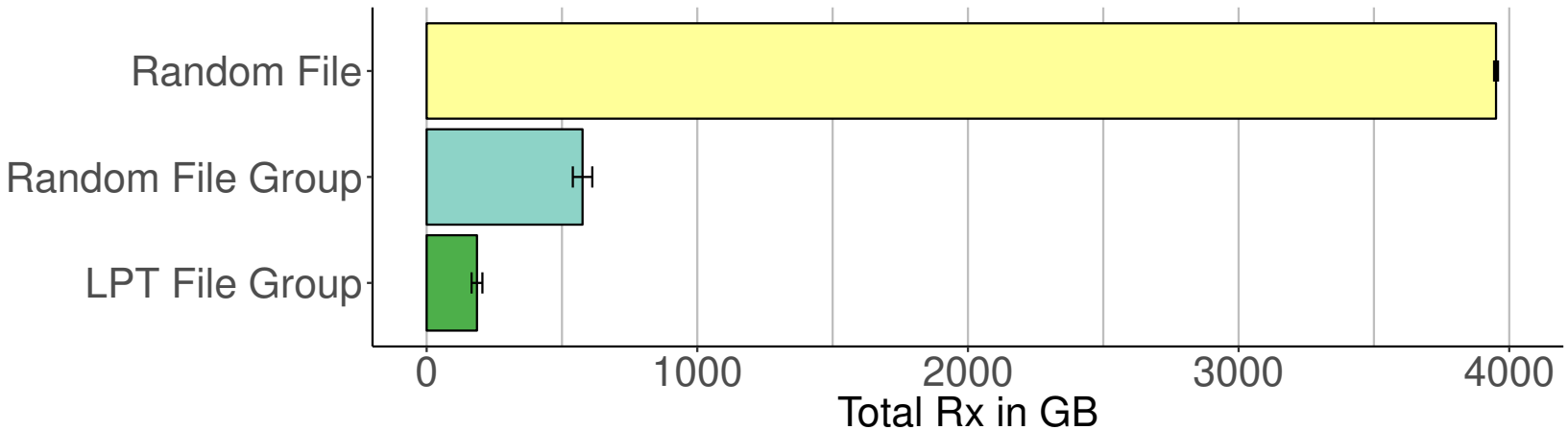
$\Rightarrow$ file.tif is stored on OSD_17

Experimental setup

- input data: 3.3 TB of satellite images in 355 groups
- hardware: one master, 29 worker/storage nodes, each with 16 CPU nodes and 10 Gb network
- job: read and decompress all data, in the same way as for land cover classification
- tested file distributions:
  - Random File (state of the art default)
  - Random File Group (e.g., CoHadoop)
  - LPT File Group (our strategy)
- each tested with HDDs and SSDs
- 10 repetitions for each setup

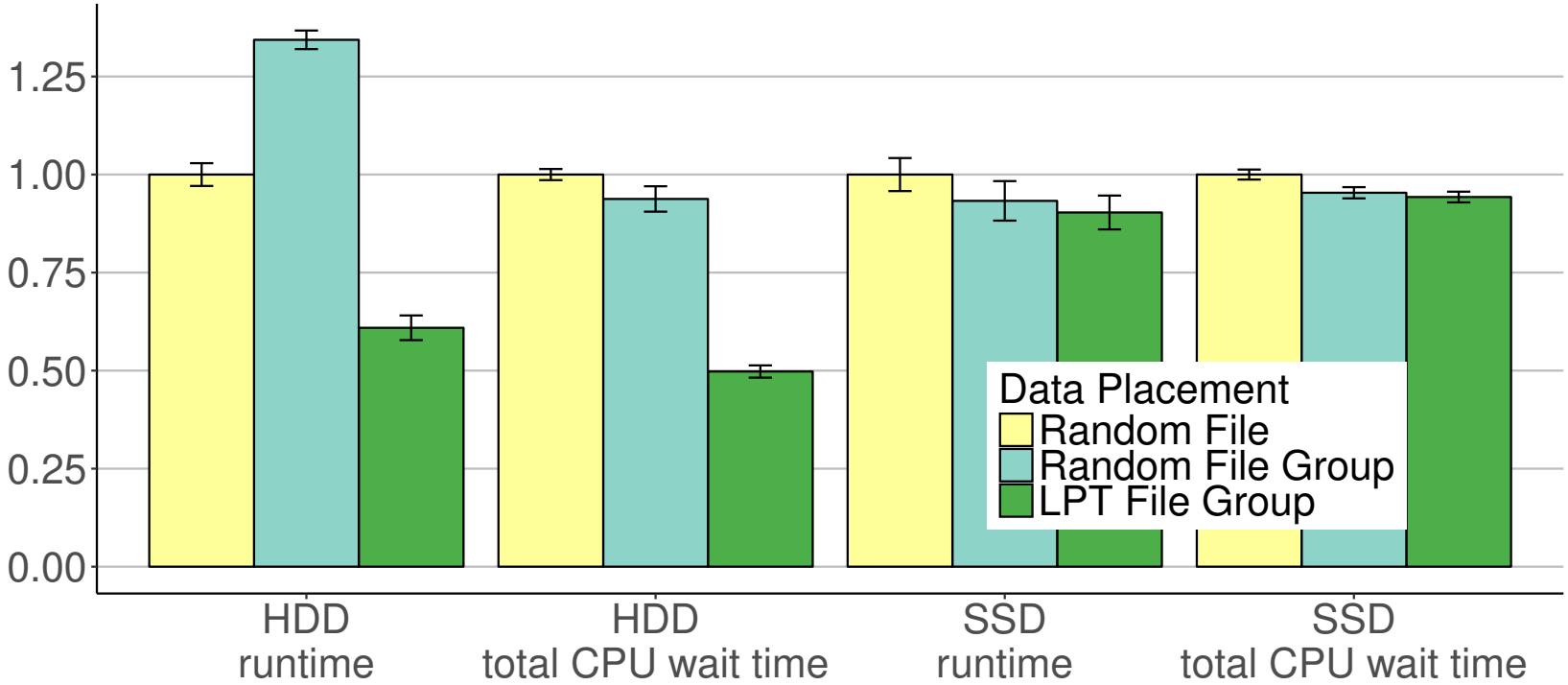
Network traffic

- measure total (incoming) network traffic of the whole job



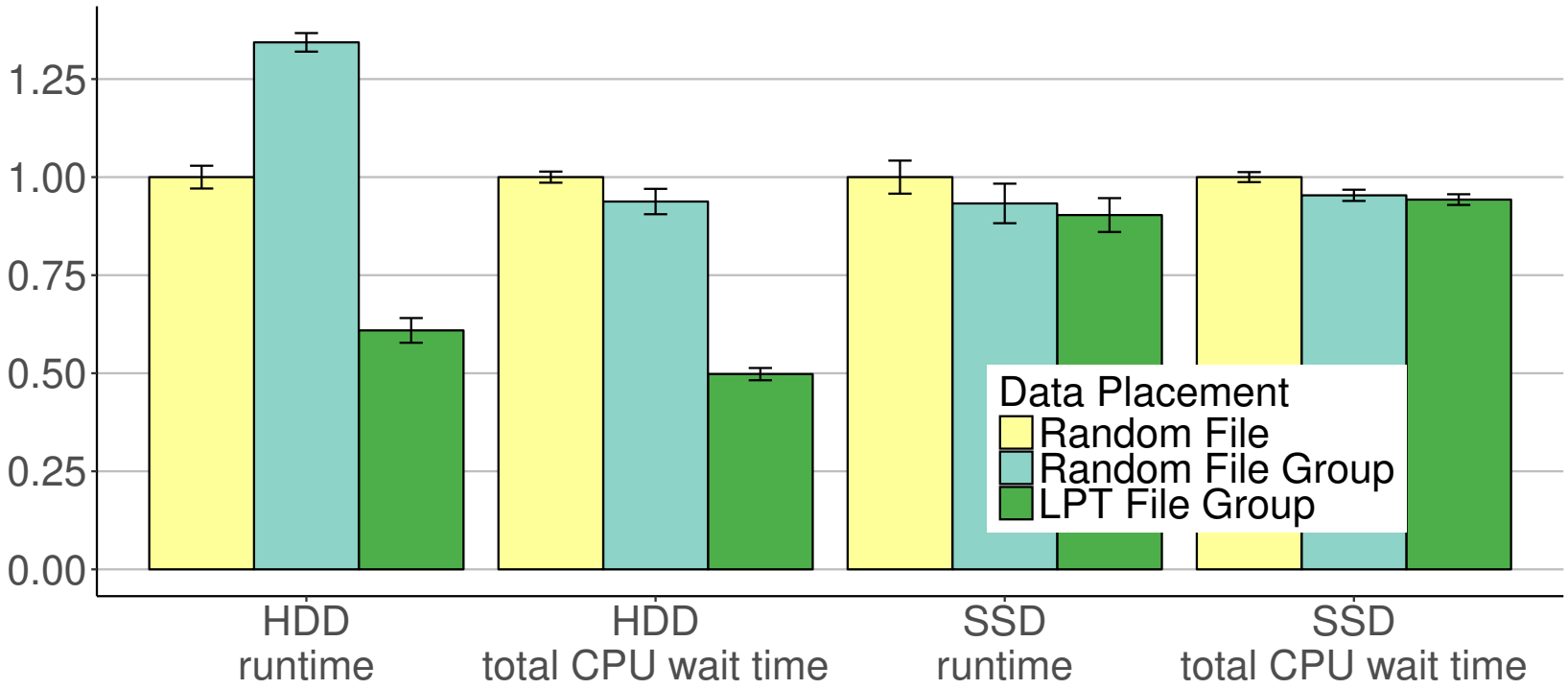
- 95 % decrease compared to Random File
- 68 % decrease compared to Random File Group

Running times and CPU wait times (relative values)



- baseline: Random File takes 40 min with HDDs
- 39 % running time and 50 % CPU wait time reduction compared to Random File
- 65 % running time and 47 % CPU wait time reduction compared to Random File Group

Running times and CPU wait times (relative values)



- baseline: Random File takes 16 min with SSDs
- file placement has no significant impact with SSD setups
- low network usage seems to have little impact
  $\Rightarrow$ HDD speedup mostly due to better scheduling

Conclusions/Summary

- Lightweight file placement mechanism that combines:
  - colocation of related input files for local performance
  - nearly optimal storage server selection for global perfomance (load balancing)

- Empirically verified benefits of colocated LPT placement:
  - network traffic reduced by around 95% compared to Random File placement
  - time to read input reduced by 39% / 65% compared to Random File / Random Group placement
  - difference to optimal solution less than 5% of the optimal solution

$\Rightarrow$ XtreemFS is ready for efficient large-scale analysis of the Earth's surface

References

- XtreemFS: `www.xtreemfs.org`, `https://github.com/xtreemfs/xtreemfs`
- client tool: `https://github.com/felse/xtreemfs_client`
- Application (GeoMultiSens): `http://www.geomultisens.de/`
- Many thanks to the GeoMultiSens team!

- Felix Seibert: `https://www.zib.de/members/seibert`

- Funding: *GeoMultiSens* (grants 01IS14010C and 01IS14010B) and the *Berlin Big Data Center (BBDC)* (grant 01IS14013B).

SPONSORED BY THE
Federal Ministry of Education and Research

geo multisens

BBDC BERLIN BIG DATA CENTER