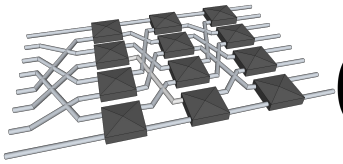


UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Performance Analysis of Parallel File System: Application Characterization and Benchmarks

Rodrigo Kassick, Francieli Boito (UFRGS/UdG)
Yves Deneulin (UdG)
Philippe O. A. Navaux (UFRGS).

L I G

.inf
INSTITUTO
DE INFORMÁTICA
UFRGS

Outline

- PFS Research at UFRGS
- Introduction
- Application Characterization
- MPI-IO based instrumentation
- Future Works

L I G

.inf INSTITUTO DE INFORMÁTICA UFRGS

Parallel File System Research at UFRGS

- Francieli Zanon Boito
  - Request Scheduling
  - Improving PFS performance by reordering requests and performing aggregations
- Rodrigo Virote Kassick
  - Mixed SSD/HDD file systems
  - Using applications' access pattern to guide data distribution on mixed SSD/HDD parallel file systems
- Yves Denneulin – LIG
- Philippe O. A. Navaux – UFRGS

L I G

.inf INSTITUTO DE INFORMÁTICA UFRGS

Cluster Computing

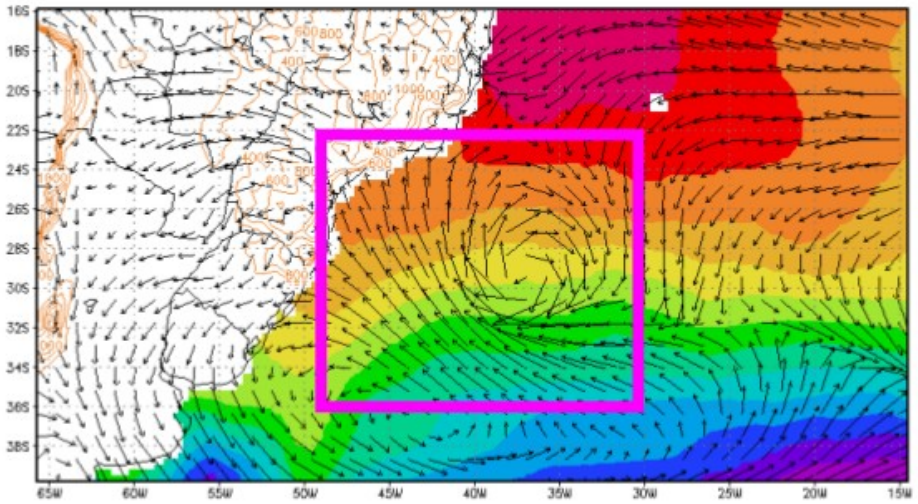
- Distributed Computing
- O(1000) individual processing units

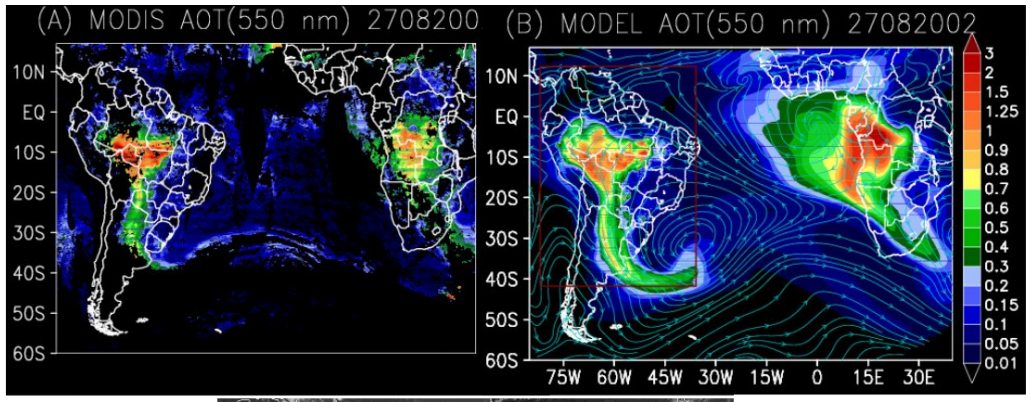
L I G

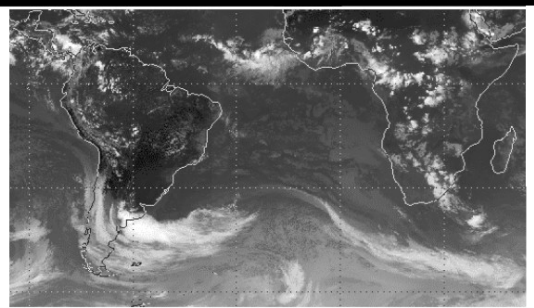
INSTITUTO DE INFORMÁTICA UFRGS

Applications

- Weather / Climate
- Materials
- Chemistry
- Biology
- Galaxy simulation
- High Energies
- Quantum
- Seismic





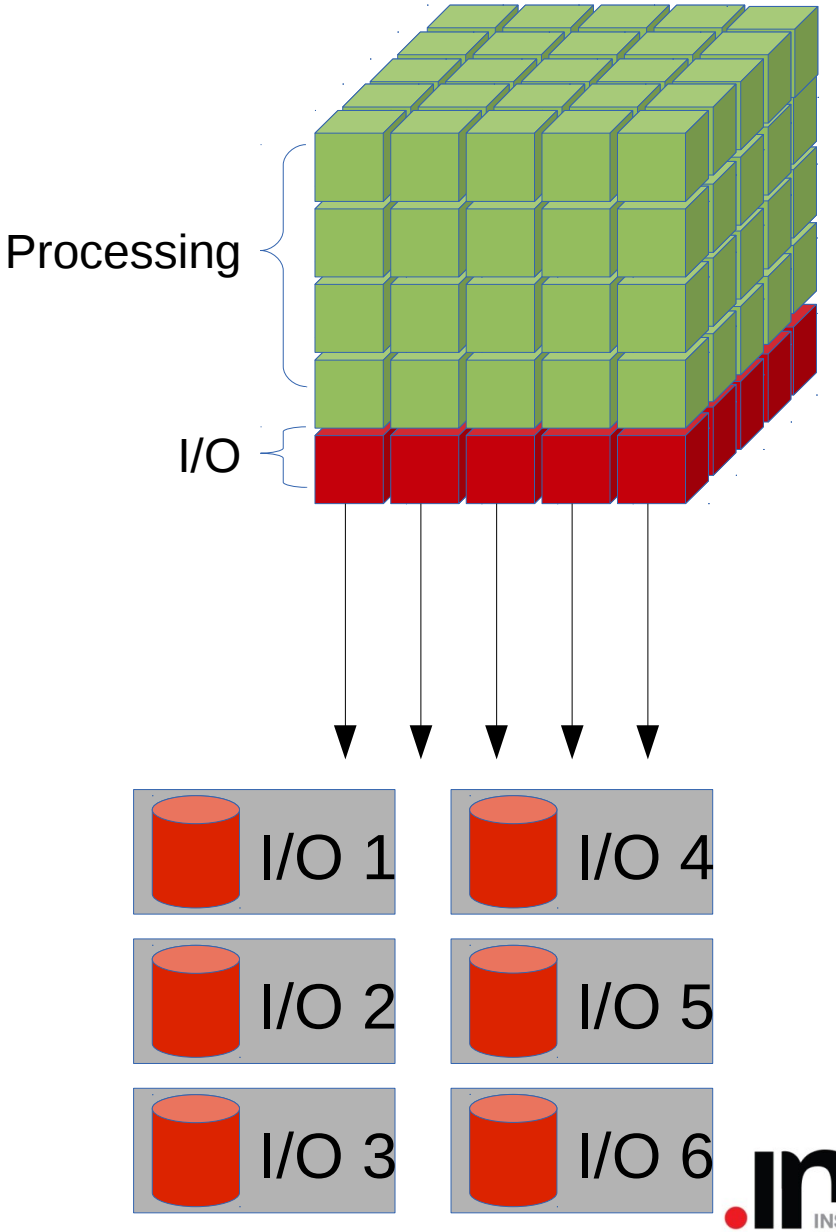


L I G

INSTITUTO DE INFORMÁTICA UFRGS

Storage & Data Management

- High precision simulations
  ↔ Amount of data
  - **Input**
    - E.g. Satellite Imagery
  - **Output**
    - Partial and final results
    - *Checkpoints*
  - **Limited Bandwidth**
    - Network + Disks



L I G

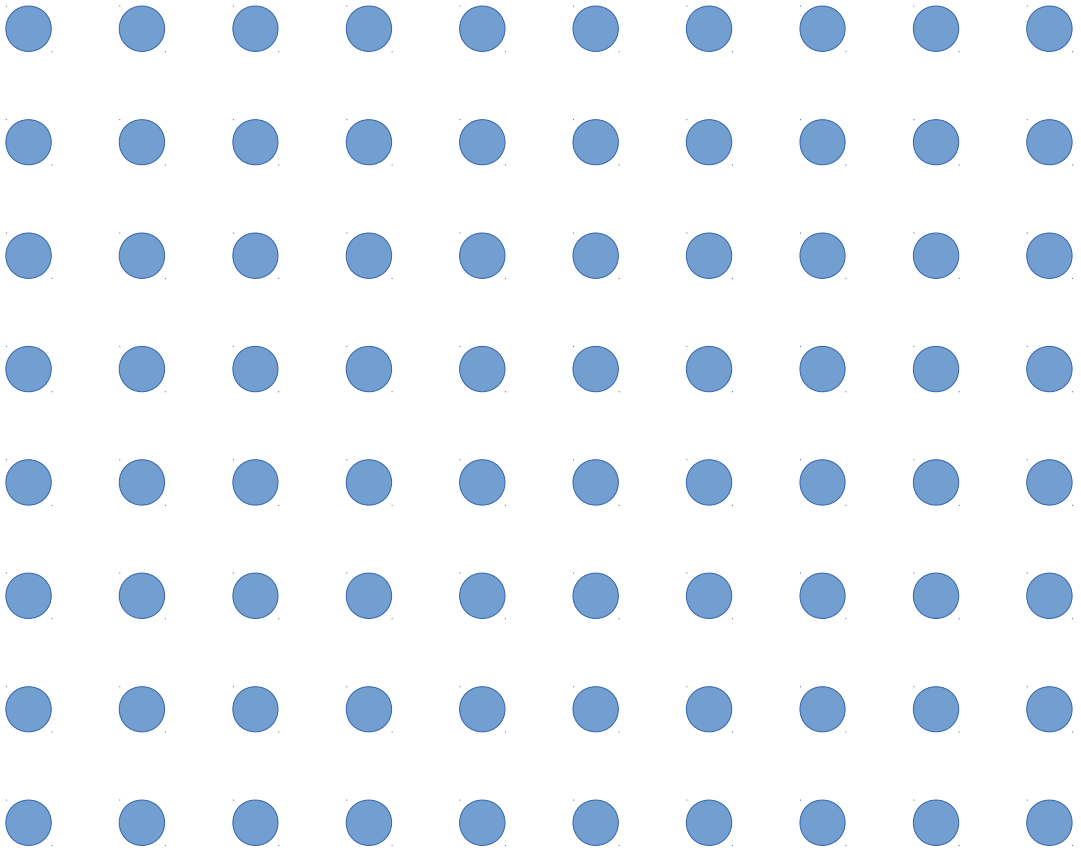
INSTITUTO DE INFORMÁTICA UFRGS

Chalenges for I/O on HPC

- Scalability
- Performance
- Fine-tuning for applications characteristics
- Benchmarking

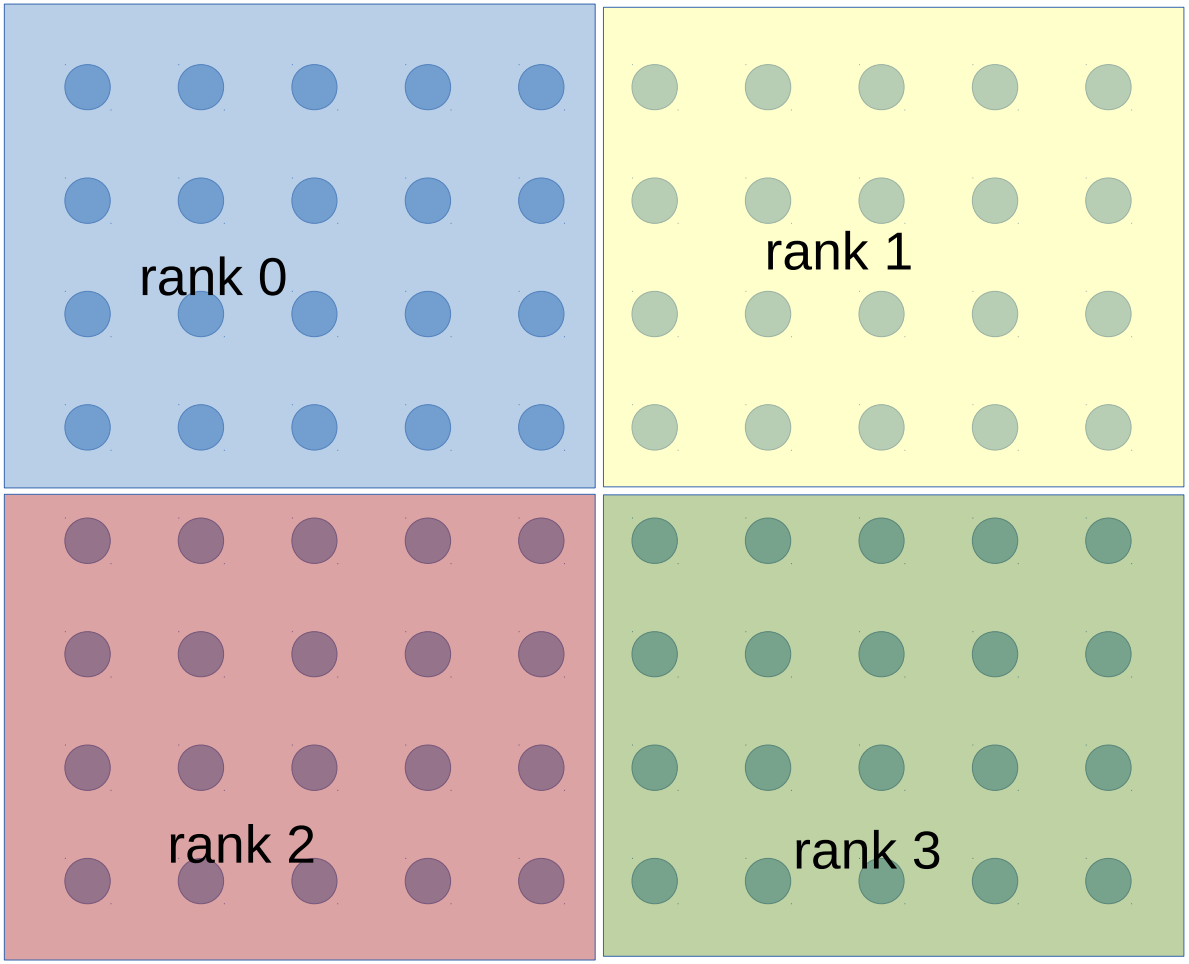
Application Characterization

- I/O Characteristics
  - Sparse / Contiguous
  - Coarse / Small grain
  - Temporal characteristics
  -
- **Consequence of the application's task, development strategy and intent**

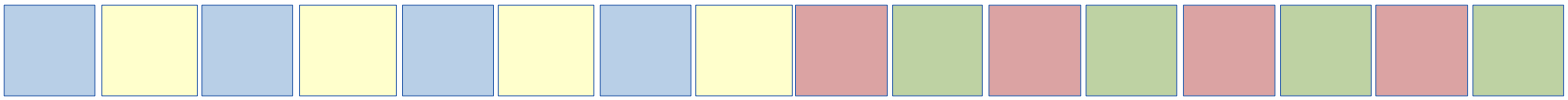


L I G

INSTITUTO DE INFORMÁTICA UFRGS

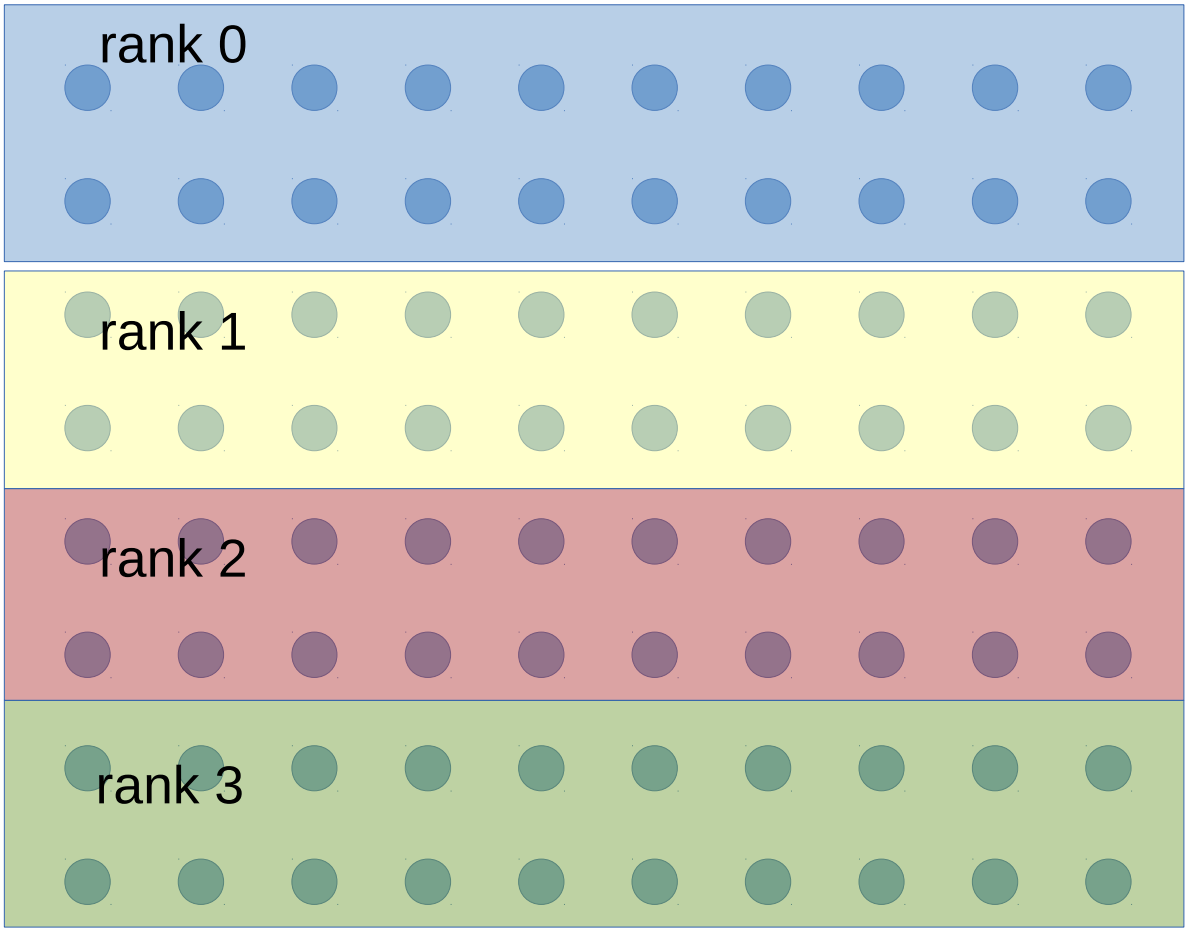


File Space:

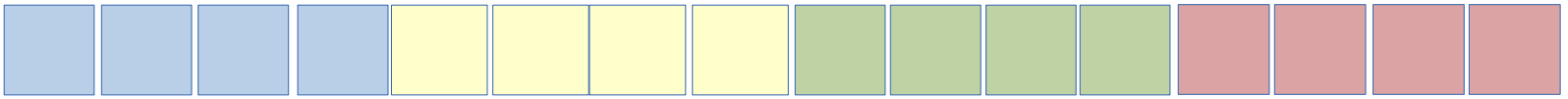


L I G

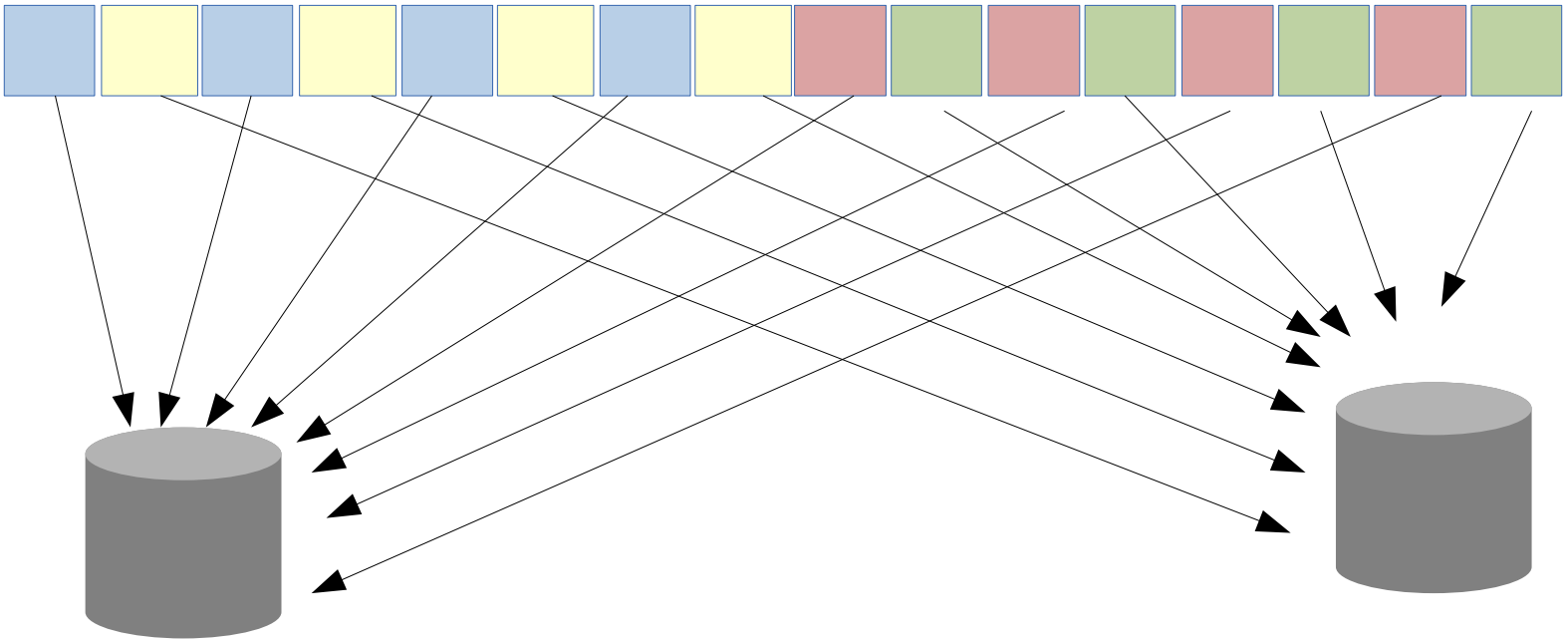
INSTITUTO DE INFORMÁTICA UFRGS



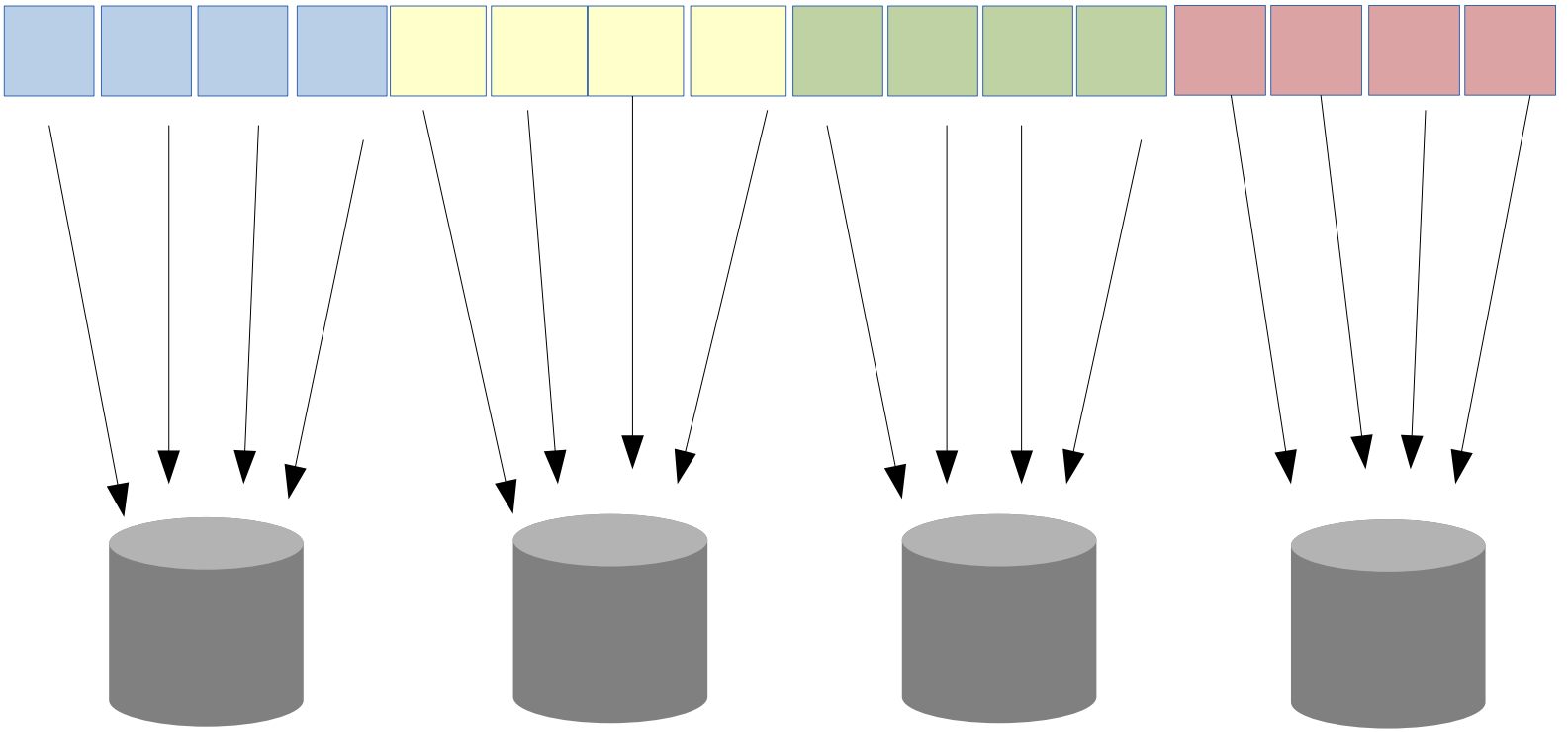
File Space:



Optimize Data Distribution



Optimize Data Distribution

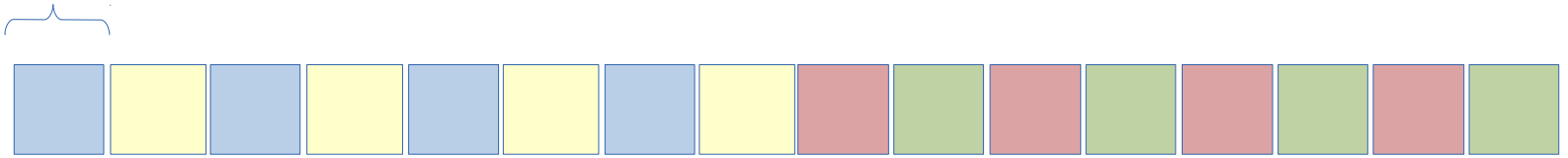


L I G

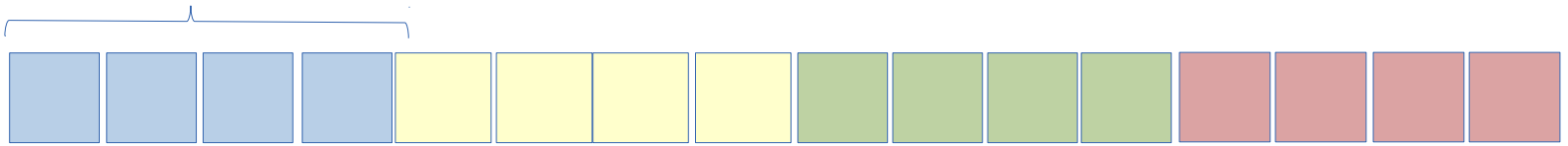
INSTITUTO DE INFORMÁTICA UFRGS

Tune Server's Request Scheduler

Contiguous requests from rank 0



Contiguous requests from rank 0



L I G

INSTITUTO DE INFORMÁTICA UFRGS

MPI-IO Based I/O Characterization

- Why MPI-IO ?
  - **Datatypes**
  - Parallel I/O
  - Used as low-level I/O library for parallel HDF5 and pNetCDF
- What for ?
  - Replay – Full I/O benchmark
  - µBenchmarks – Only some selected operations
  - Workload Characterization

L I G

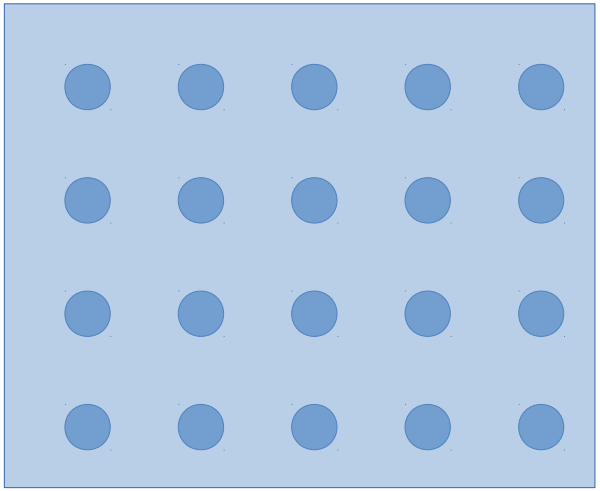
.inf INSTITUTO DE INFORMÁTICA UFRGS

MPI Datatype

- One single “write” call (application)
- Buffer of 10∗4 elements of a type
- A view of 5 elements followed by 5 empty spaces

MPI Datatype

Buffer:
4x5 Elements of type
struct (diretcion, magnitude)



File View:

5 elements followed by 5 empty spaces, repeat 4 times

MPI-IO – Views on Files

```
etype  = MPI_DOUBLE;
extent = sizeof(double) * matrix_n;
d      = default_n * sizeof(double) * myrank;

MPI_Type_contiguous(default_n, etype, &contig);
MPI_Type_create_resized(contig, 0, extent, &ftype);
MPI_Type_commit(&ftype);
MPI_File_set_view(thefile, d , etype,
                          ftype, "native",
                          MPI_INFO_NULL);

bufsize = my_n * matrix_m;

MPI_File_write_all(thefile, my_vector_cont, bufsize,
                       MPI_DOUBLE, MPI_STATUS_IGNORE);
```

MPI-IO – Views on Files

```
etype  = MPI_DOUBLE;
extent = sizeof(double) * matrix_n;
d      = default_n * sizeof(double) * myrank;

MPI_Type_contiguous(default_n, etype, &contig);
MPI_Type_create_resized(contig, 0, extent, &ftype);
MPI_Type_commit(&ftype);
MPI_File_set_view(thefile, d, etype,
```

- Single I/O operation gives a lot of hints on what parts of the file will be requested/written to

```
MPI_File_write_all(thefile, my_vector_cont, bufsize,
                   MPI_DOUBLE, MPI_STATUS_IGNORE);
```

Implementation

- Use PMPI to overload MPI-IO functions
  - + DataType functions, + Async management
- Generate trace with information on the parameters of the functions
- Use *strace* to complement the trace

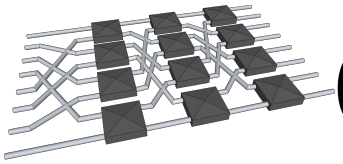
Replay

- Parallel replay of a previously extracted trace
  - Improve reproducibility of tests with I/O optimizations
- Uses MPI and mimics the I/O operations from the target application

- Currently capable of replaying simple parallel HDF5 codes

Future Works

- Trace Collection
  - Collect and publish a set of traces from parallel applications for use in future tests and by the community
- µBenchmarks collection
  - Investigate frequent access patterns on the traces and generate small traces for testing file systems for their specific characteristics



UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Performance Analysis of Parallel File System: Application Characterization and Benchmarks

Rodrigo Kassick, Francieli Boito (UFRGS/UdG)
Yves Deneulin (UdG)
Philippe O. A. Navaux (UFRGS).

L I G

.inf
INSTITUTO
DE INFORMÁTICA
UFRGS