UNIVERSITÀ DI TORINO

UNIVERSITÀ DI PISA

ICSC
Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

CAPIO
Cross application Programmable I/O

CAPIO: Cross Application Programmable IO

Scientific HPC in the pre-Exascale era - ITADATA 2024

Marco Edoardo Santimaria **marcoedoardo.santimaria@unito.it**
Iacopo Colonnelli iacopo.colonnelli@unito.it
Massimo Torquati massimo.torquati@unipi.it
Marco Aldinucci marco.aldinucci@unito.it

ACROSS
HPC Big Data Artificial intelligence cross Stack Platform Toward Exascale

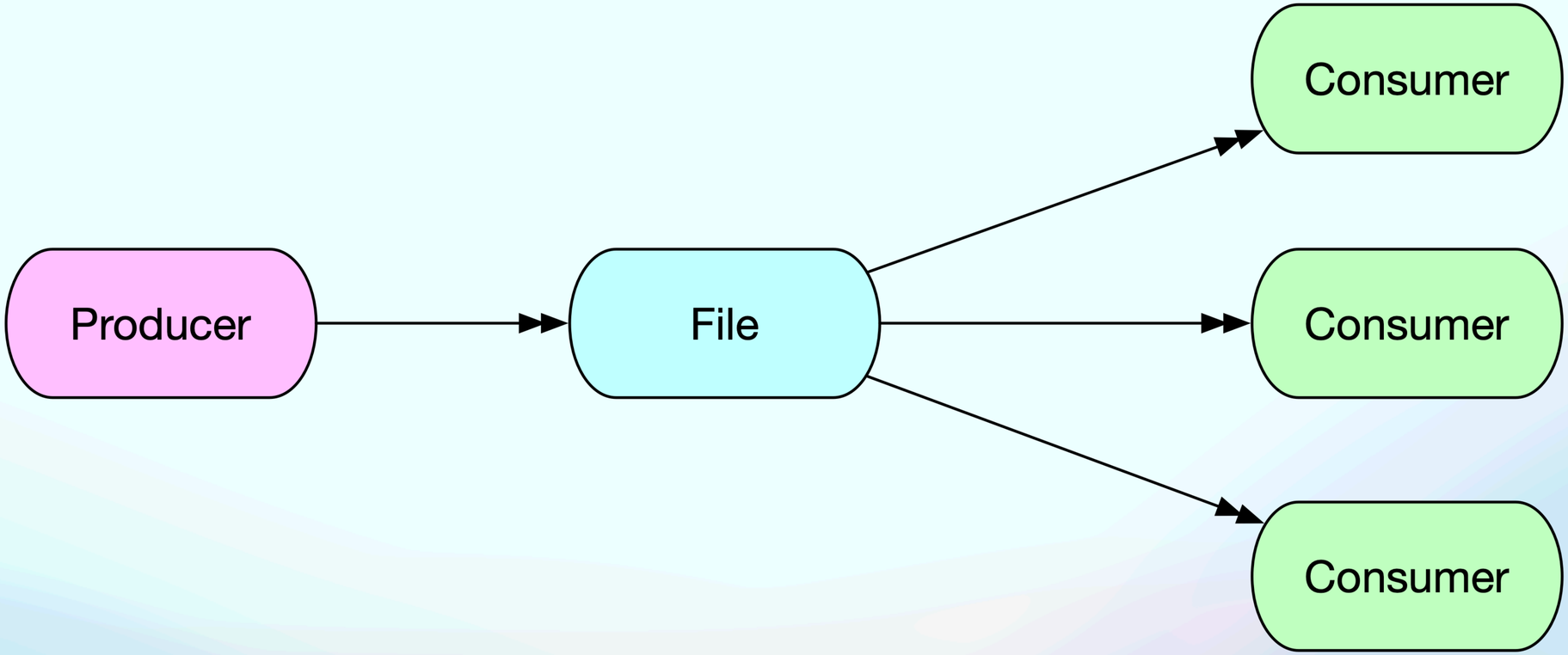
EUPEX
European Pilot for Exascale

ADMIRE
malleable data solutions for HPC
ADAPTIVE MULTI-TIER INTELLIGENT DATA MANAGER FOR EXASCALE

Motivations

- The I/O performance gap is still a significant bottleneck in large-scale HPC workflows.
- Parallel file systems struggle to scale linearly with the aggregate disk bandwidth in real scenarios.
- Computation and I/O overlapping are crucial in minimizing I/O overhead.
- Thousands of existing file-based workflows w/o streaming capabilities.

File based workflows



What is CAPIO

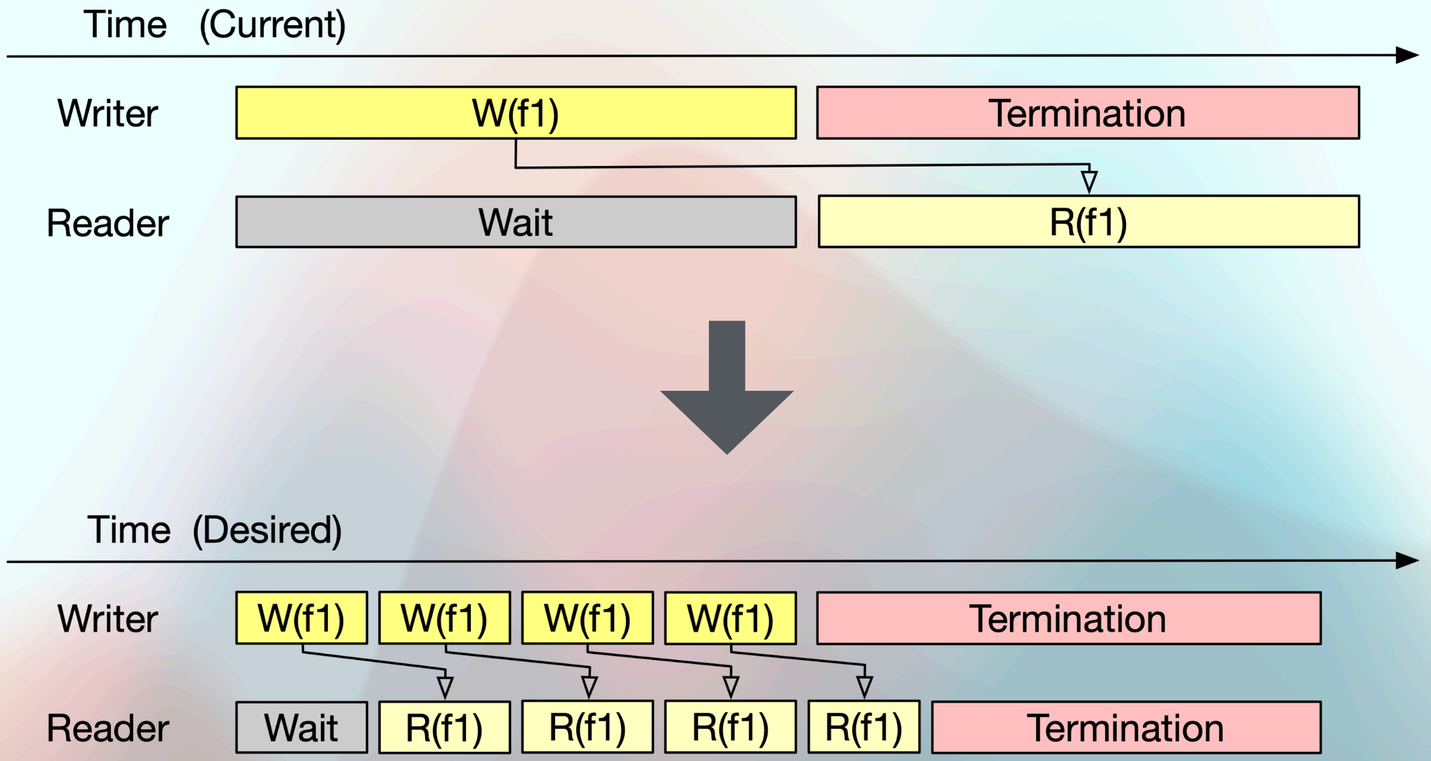
- User-space data transport middleware enabling I/O coordination in scientific Workflows

- No need to modify the workflow's step

- POSIX I/O System Calls (read/write/seek/stat/...) are transparently intercepted using dynamic linker features (i.e., LD_PRELOAD)

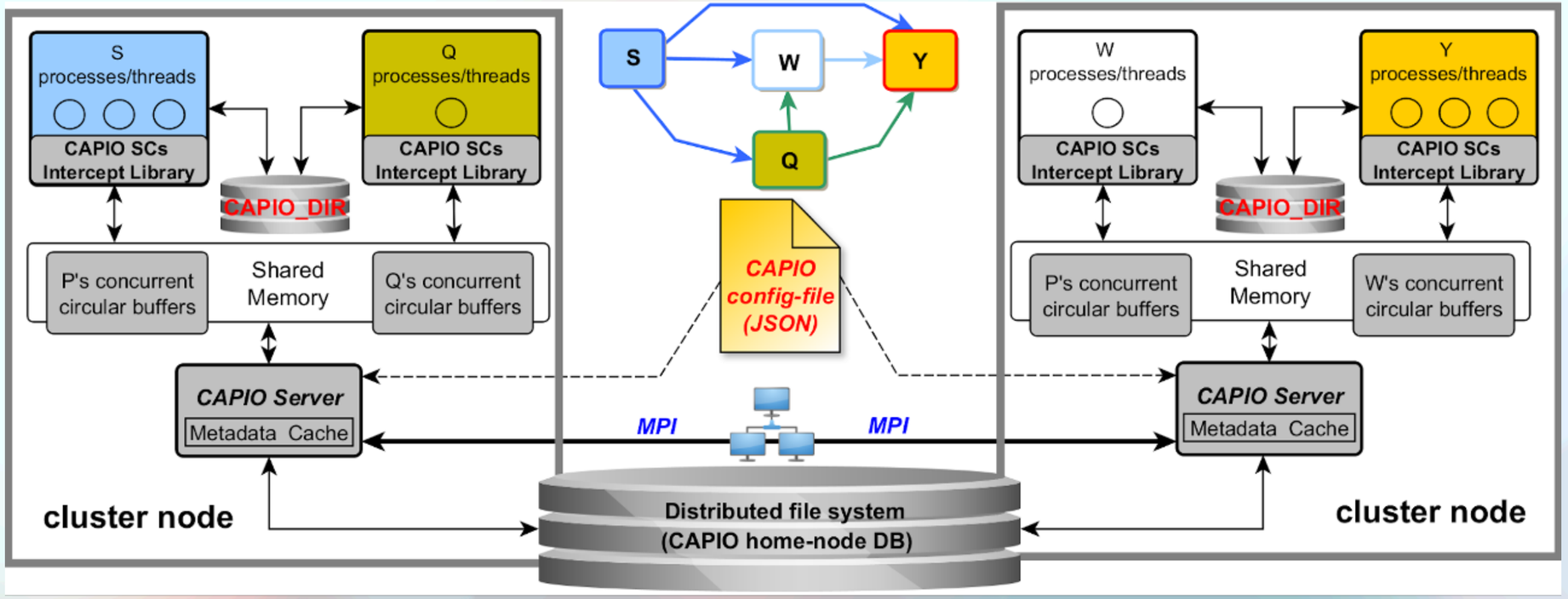
capio.hpc4ai.it

What is CAPIO



capio.hpc4ai.it

CAPIO Deployment



capio.hpc4ai.it

What is CAPIO-CL

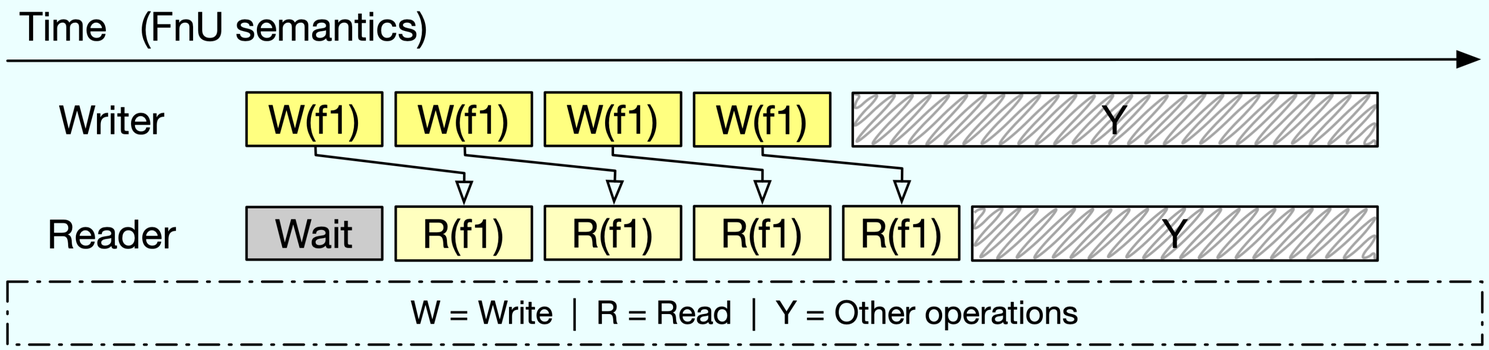
- An innovative, fully declarative, I/O coordination language

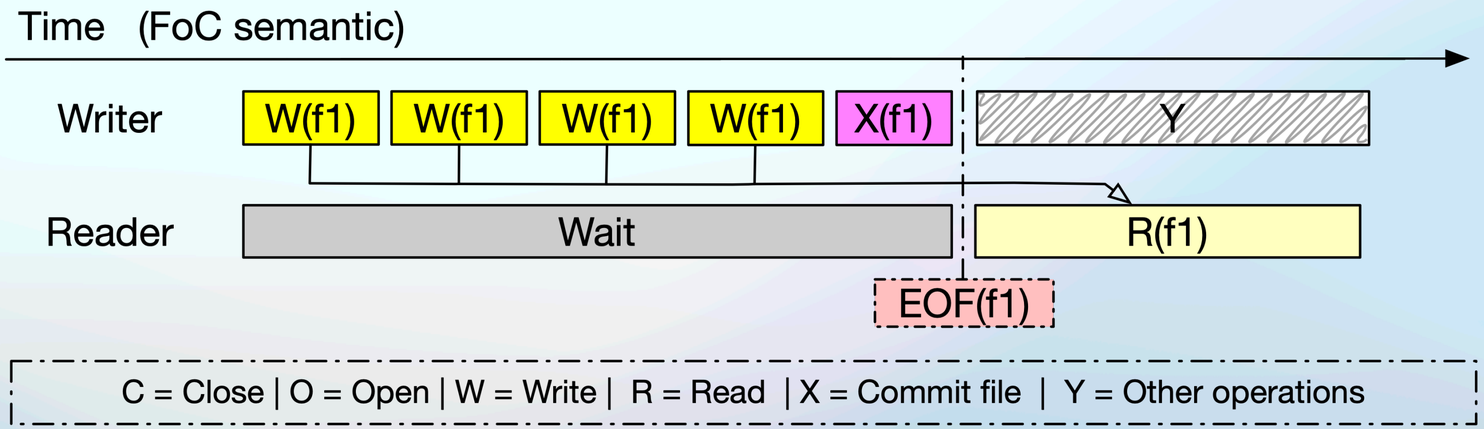
- Annotates data dependencies within file-based workflows with synchronization semantics

- Allows CAPIO to coordinate streaming injection

capio.hpc4ai.it

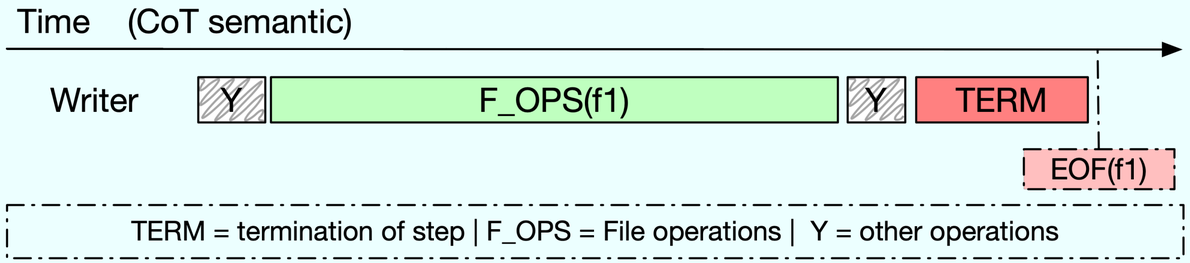
Firing rules

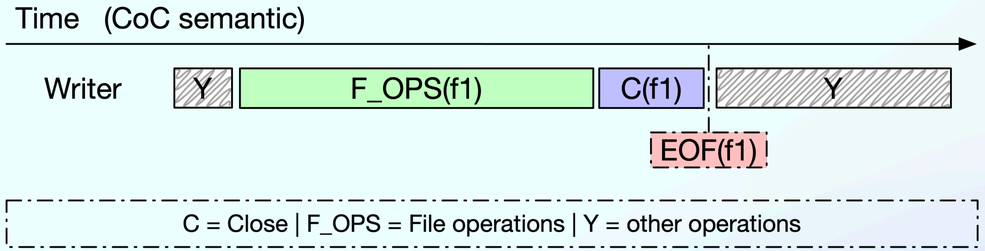


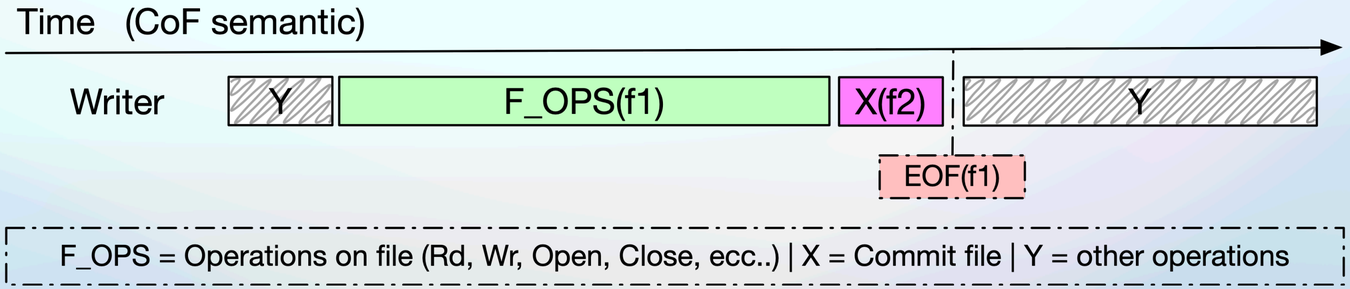


capio.hpc4ai.it

Commit rules

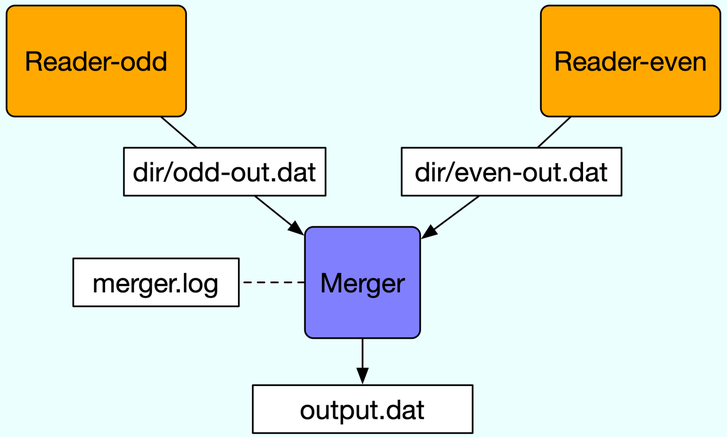






capio.hpc4ai.it

Usage example




```
export CAPIO_DIR=$CAPIODIR

srun -N 3 --overlap capio_server -c config.json &

srun -n 1 --export=ALL,LD_PRELOAD=libcapio_posix.so ./Reader-odd &
PID1=$!

srun -n 1 --export=ALL,LD_PRELOAD=libcapio_posix.so ./Reader-even &
PID2=$!

srun -n 1 --export=ALL,LD_PRELOAD=libcapio_posix.so ./Merger

wait $PID1 && wait $PID2
```

```
{
  "name" : "my_workflow",
  "permanent" : [ "output.dat"],
  "exclude" : [ "merger.log"],
  "IO_Graph": [
    {
      "name":"reader-even",
      "input_stream": ["group-even"],
      "output-stream": ["even-out.dat"],
      "streaming" : [{
          "name":["even-out.dat"],
          "committed":"on_close",
          "mode":"no_update"
        }]
    },{
      "name":"reader-odd",
      "input_stream": ["group-odd"],
      "output-stream": ["odd-out.dat"],
      "streaming" : [{
        "name":["odd-out.dat"],
        "committed":"on_file:even-out.dat",
        "mode":"update"}]
    },{
      "name": "merger",
      "input_stream":  ["odd-out.dat", "even-out.dat"],
      "output_stream" : ["output.dat"]
    }
  ]
}
```

CAPIO vs ADIOS2

```
...
#include <adios2.h>

void __write__(){
  std::ofstream out("test.txt");
  out << "Hello world with ADIOS2";
  out.close();
}

void __read__(){
   std::ifstream in("test.txt");
   std::cout << in.rdbuf();
}

void main(){
    std::thread reader_thread(__read__);
    std::thread writer_thread(__write__);
    reader_thread.join();
    writer_thread.join();
}
```

```
mpirun capio_server -c <config.json> &
CAPIO_DIR=... LD_PRELOAD=libcapio_posix.so ./demo_program
```

```
...
#include <adios2.h>

void __write__(){
    std::string msg = "Hello world with ADIOS2";
    adios2::fstream oStream(filenameString, adios2::fstream::out);
    oStream.write<std::string>("data", msg);
    oStream.close();
}

void __read__(){
    adios2::fstream istream("test.bp", adios2::fstream::in);
    for (adios2::fstep iStep; adios2::getstep(istream, iStep);)
        std::string msg = istream.read<std::string>("data").front();
    istream.close()
}

void main(){
    std::thread reader_thread(__read__);
    std::thread writer_thread(__write__);
    reader_thread.join();
    writer_thread.join();
}
```

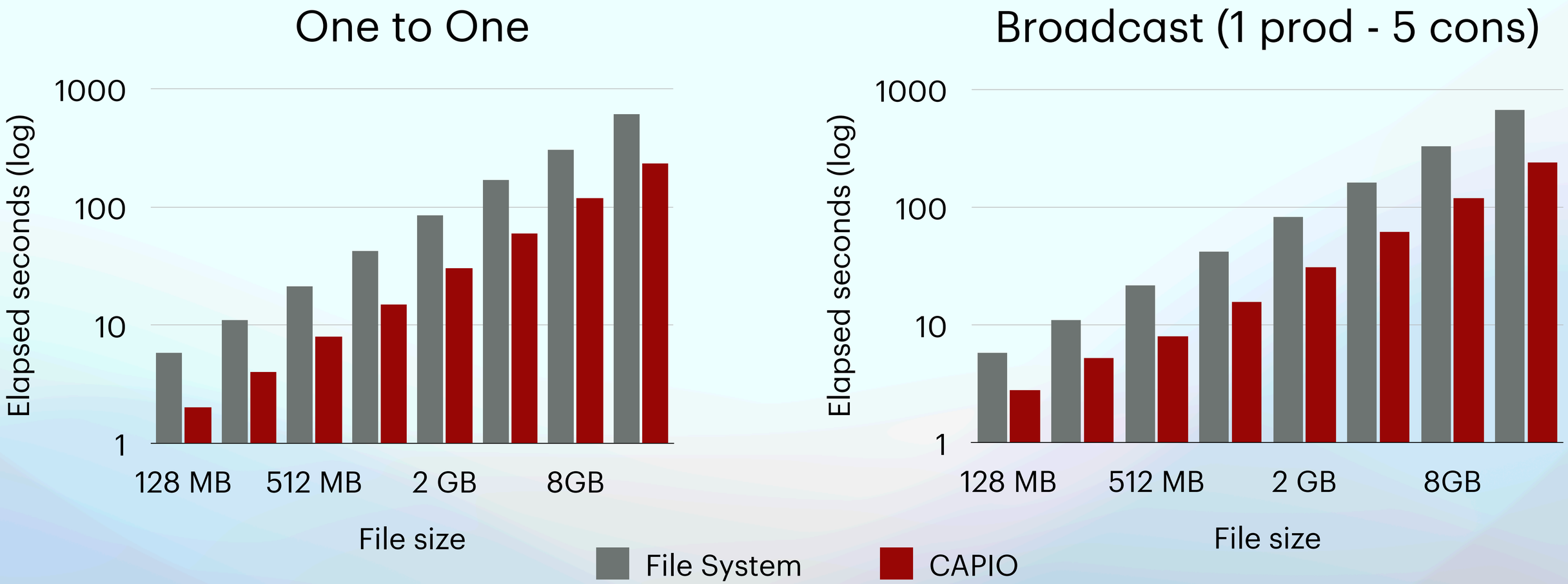
capio.hpc4ai.it

Marco Edoardo Santimaria @ CNR Pisa - 18 sept. 2024

Performance

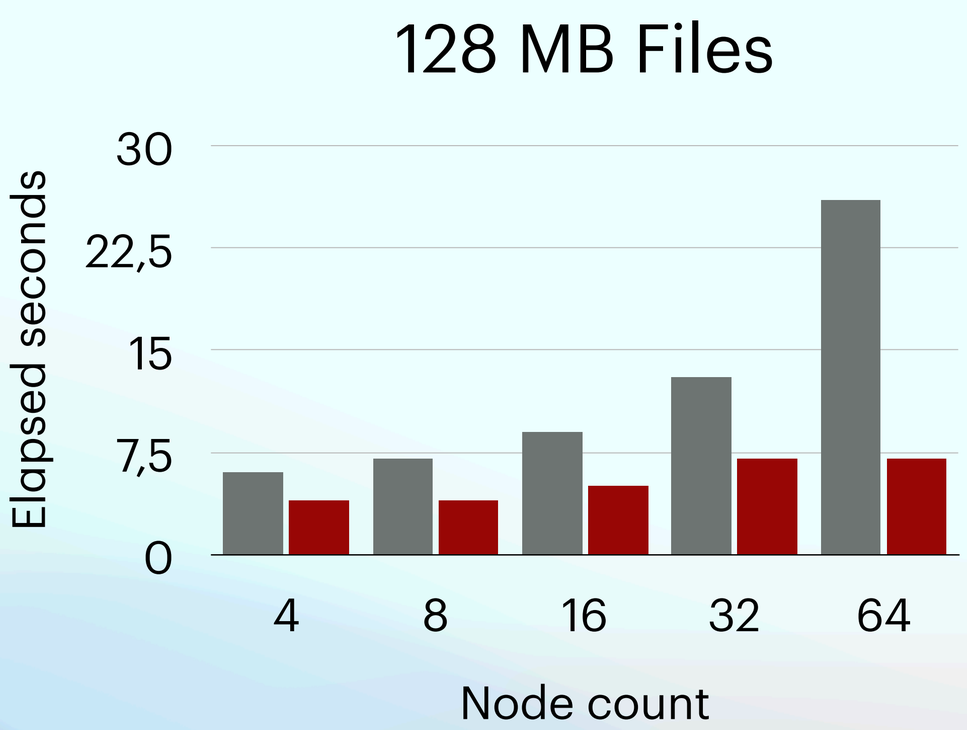
Basic IO patterns

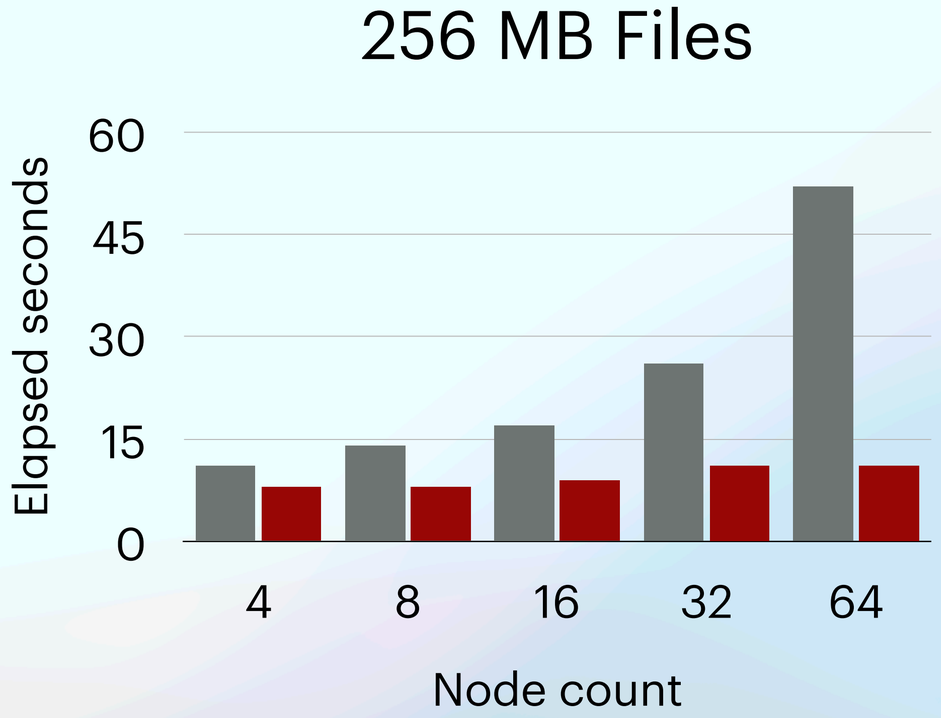
(5 iterations per test)



Scaling

(Broadcast, 5 iterations per test, 1 producer - n consumers)

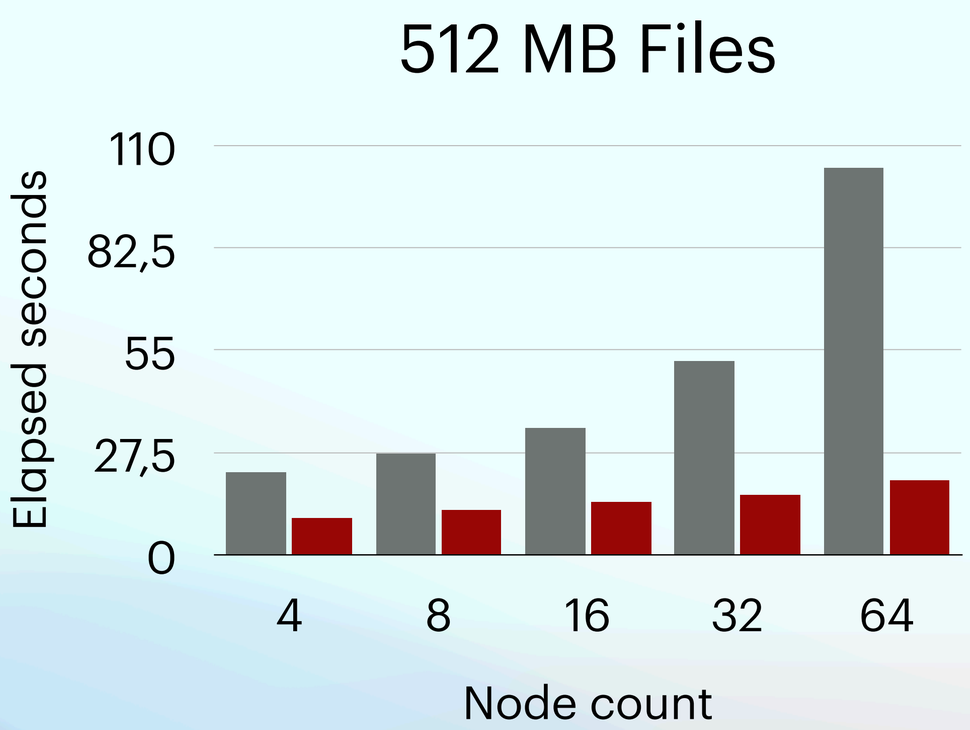


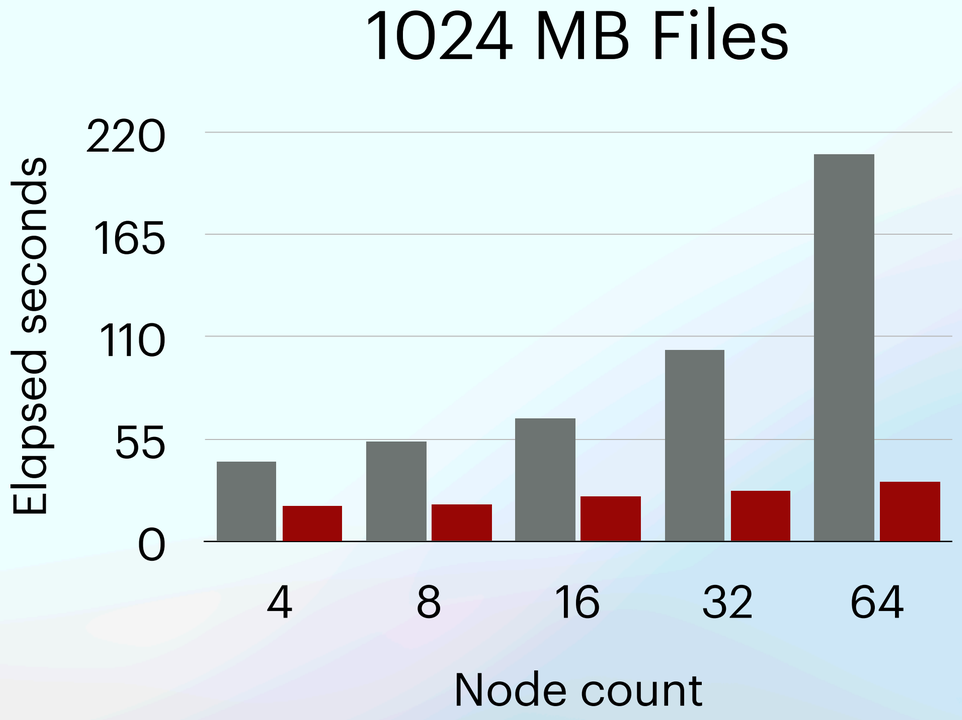


File System CAPIO

Scaling

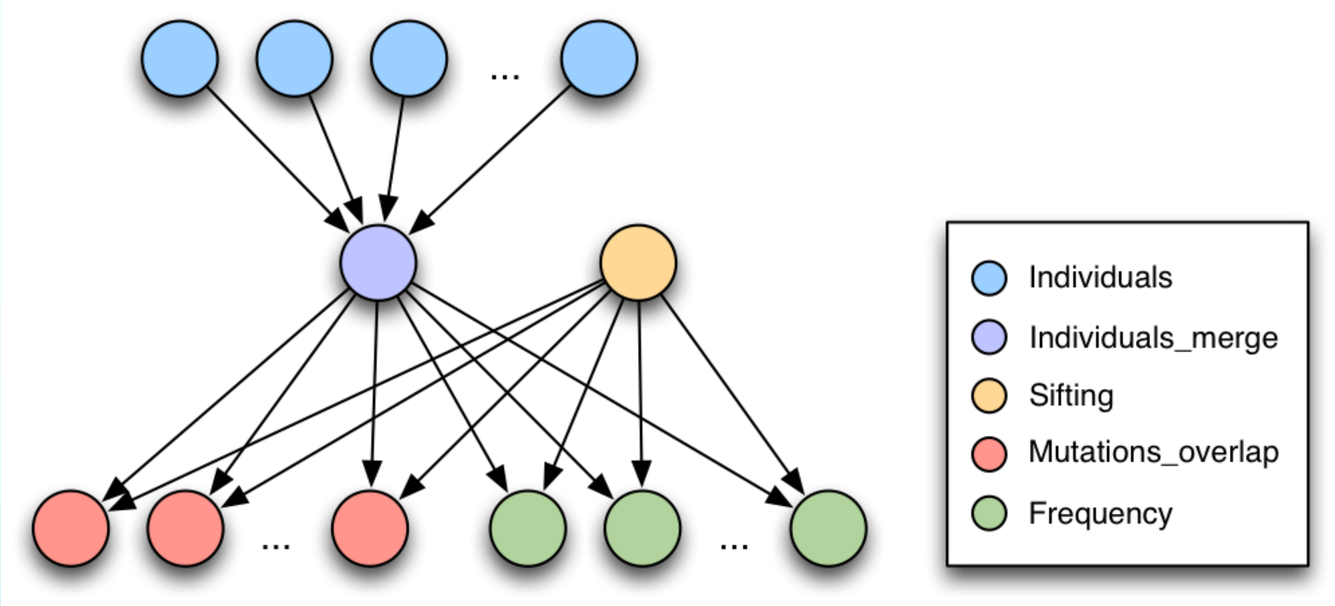
(Broadcast, 5 iterations per test, 1 producer - n consumers)

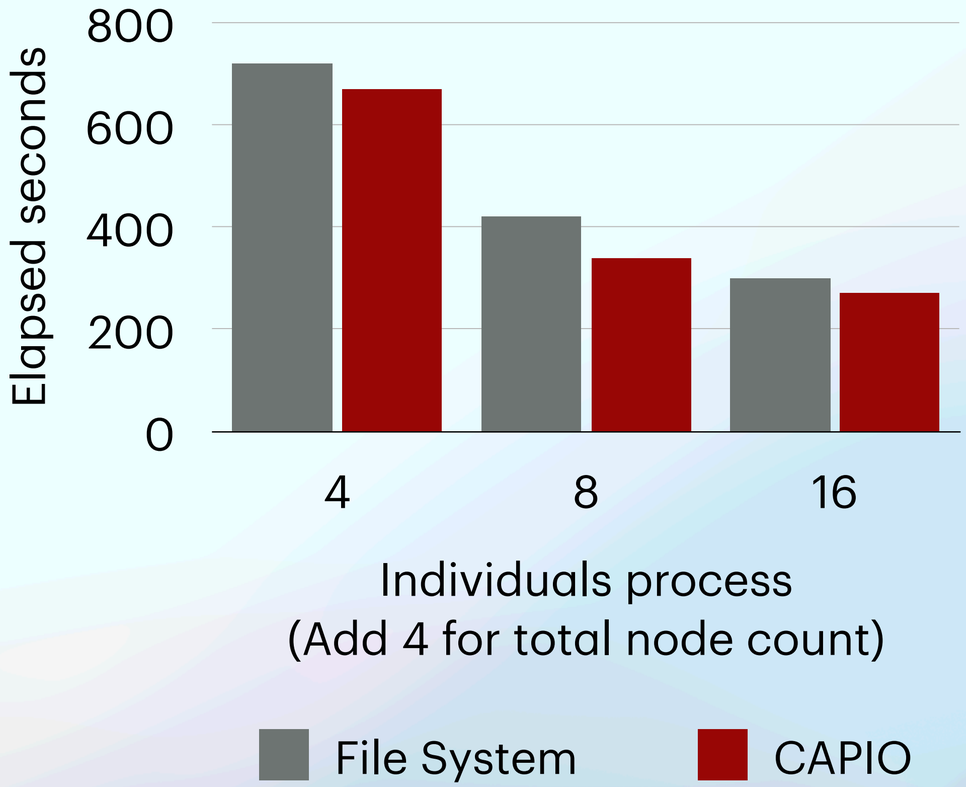




File System CAPIO

1000 Genome workflow





Future works

- Remove MPI from CAPIO
- Integration with WFMs

UNIVERSITÀ DI TORINO

UNIVERSITÀ DI PISA

ICSC
Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

CAPIO
Cross application Programmable I/O

Questions?

Marco Edoardo Santimaria marcoedoardo.santimaria@unito.it
Iacopo Colonnelli iacopo.colonnelli@unito.it
Massimo Torquati massimo.torquati@unipi.it
Marco Aldinucci marco.aldinucci@unito.it

capio.hpc4ai.it