RAMSES use case : self-gravitating astrophysical plasma flows

BOURNAUD Frédéric – frederic.bournaud@cea.fr
CHAPON Damien - damien.chapon@cea.fr
STRAFELLA Loïc - loic.strafella@cea.fr

RAMSES : introduction

RAMSES [(a)] code

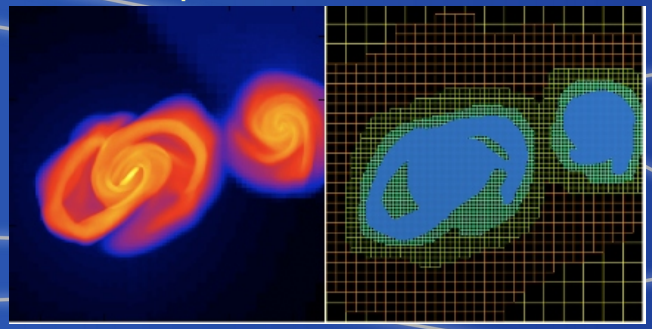
- Developped since 2002,
- Open source code to model self-gravitating astrophysical magnetised fluids,
- Community of ~ 300 users all around the world. One of the 4 most used code in astrophysics,
- Fortran 90 standard, use OpenMP and MPI, compilation with Intel and GNU compilers,

(a) R. Teyssier, 2002, Cosmological hydrodynamics with adaptive mesh refinement: a new high-resolution code called RAMSES, A&A, 385, 337-364

RAMSES : introduction

RAMSES [(a)] code

- Developped since 2002,
- Open source code to model self-gravitating astrophysical magnetised fluids,
- Community of ~ 300 users all around the world. One of the 4 most used code in astrophysics,
- Fortran 90 standard, use OpenMP and MPI, compilation with Intel and GNU compilers,
- About 50k lines of codes, adaptive mesh refinement (AMR) technique

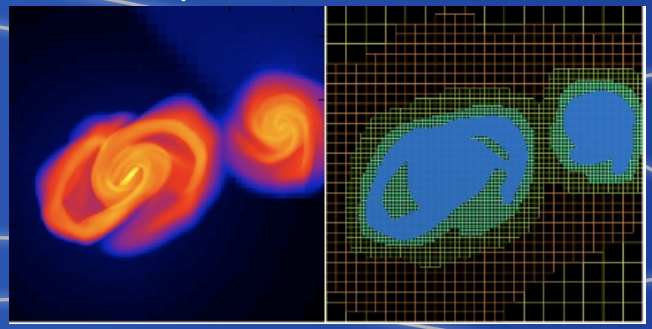


(a) R. Teyssier, 2002, Cosmological hydrodynamics with adaptive mesh refinement: a new high-resolution code called RAMSES, A&A, 385, 337-364

RAMSES : introduction

RAMSES [(a)] code

- Developped since 2002,
- Open source code to model self-gravitating astrophysical magnetised fluids,
- Community of ~ 300 users all around the world. One of the 4 most used code in astrophysics,
- Fortran 90 standard, use OpenMP and MPI, compilation with Intel and GNU compilers,
- About 50k lines of codes, adaptive mesh refinement (AMR) technique
- **I/O engine** : custom POSIX files
  - Checkpoints only, used for restarting the code and for analysis
  - Code-specific Fortran binary files
  - N files per MPI process : 10k cores simulation → **40000 – 80000 files / snapshot**

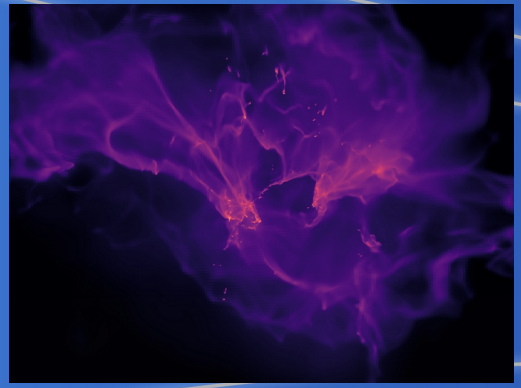
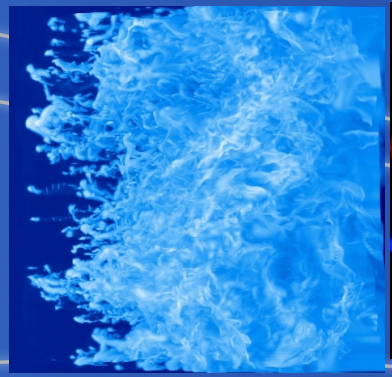
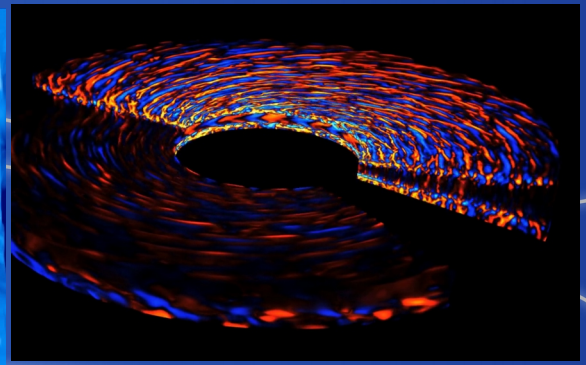
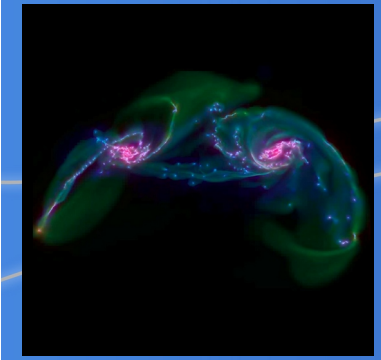
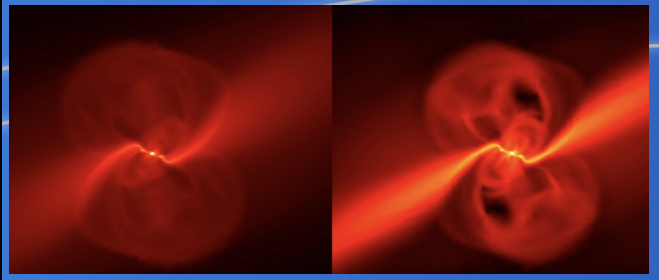
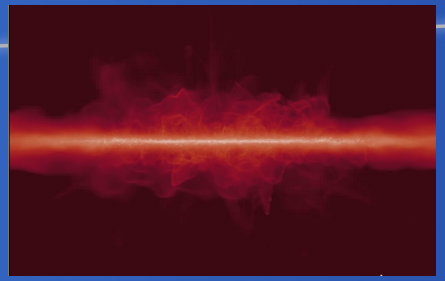
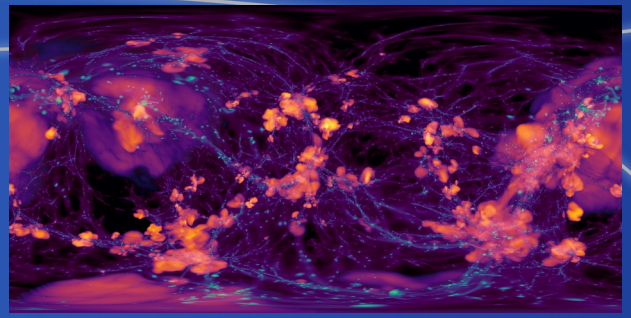


(a) R. Teyssier, 2002, Cosmological hydrodynamics with adaptive mesh refinement: a new high-resolution code called RAMSES, A&A, 385, 337-364

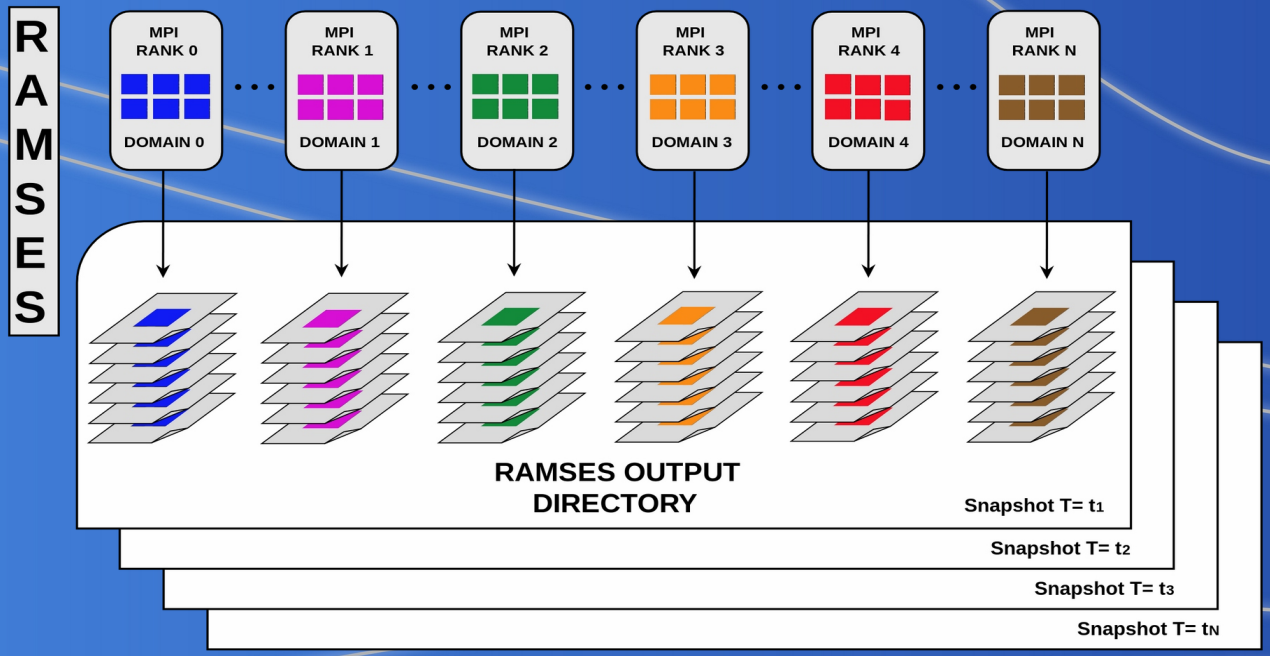
RAMSES : applications

RAMSES is used in

- Cosmology, Galactic dynamics, Galaxy interactions,
- Interstellar medium modeling, stellar formation, planetary system formation, accretion disks,
- Supernovae explosions,

RAMSES : I/O pattern



1 MPI process = n files (depending on the activated physics modules)

1 checkpoint = 1 directory = $P_{process}$ * n files

1 simulation = N snapshots = N directories = N * $P_{process}$ * n files (small files < 1GB)

➡ RAMSES limit of scalability : ~ 8,000 MPI processes

IO-SEA : expected benefits

Main objectives:

- Improve RAMSES I/O scalability,
- Improve post-processing productivity and data management

IO-SEA : expected benefits

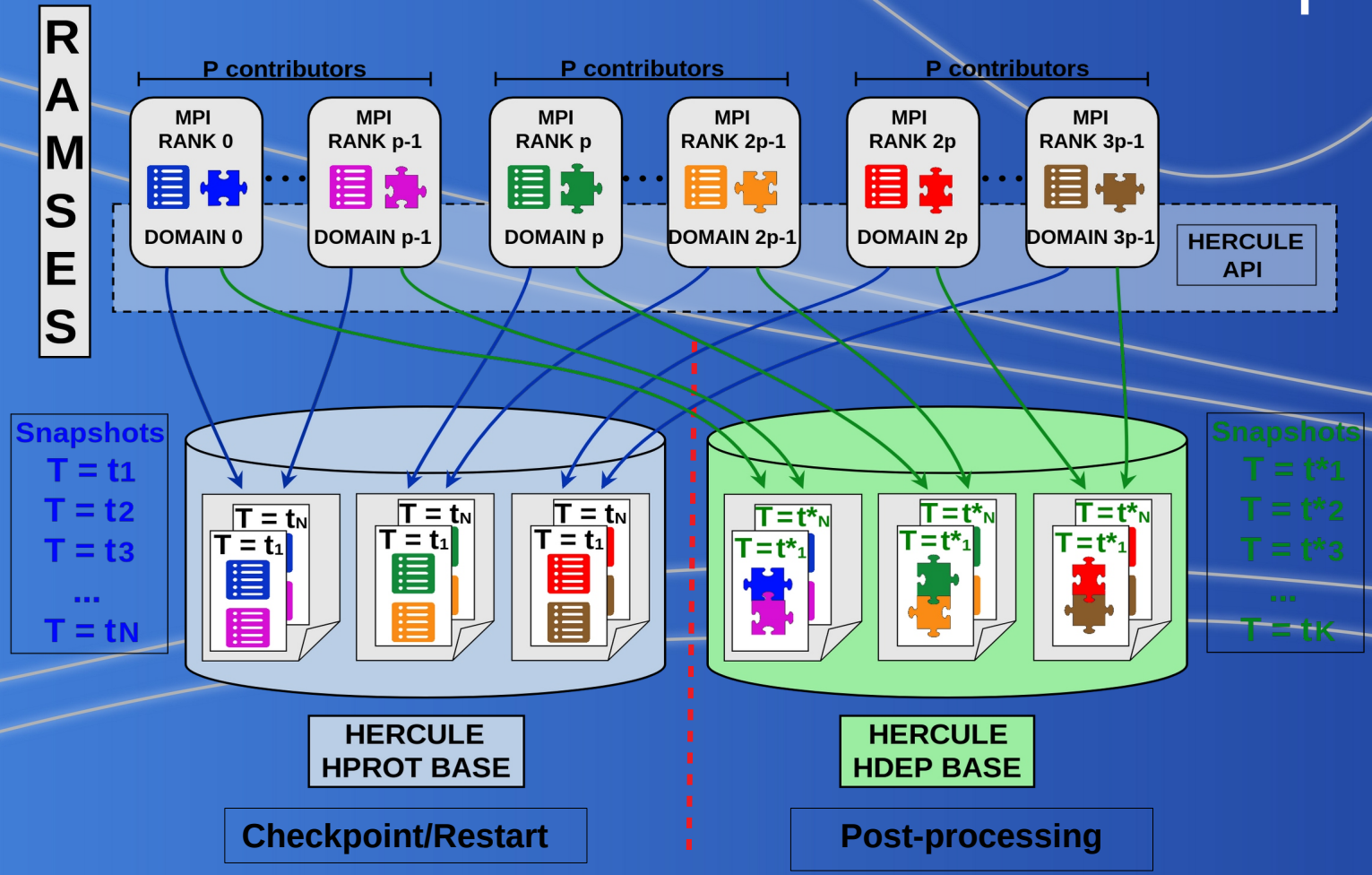
Main objectives:

- Improve RAMSES I/O scalability,
- Improve post-processing productivity and data management

Integration of Hercule (WP5) parallel I/O library in RAMSES :

- Reduced number of files (inode limitations on Lustre filesystem
  - contribution of multiple MPI processes to a single shared file (I/O concurrency)
- Improved scalability of RAMSES I/Os over 8,000 MPI processes,
- Taking advantage of supercomputer storage architecture
  - Multiple OSS, ...
- Semantic approach : compatible with DASI interface,
- Reduce data volume and improve simulation data management

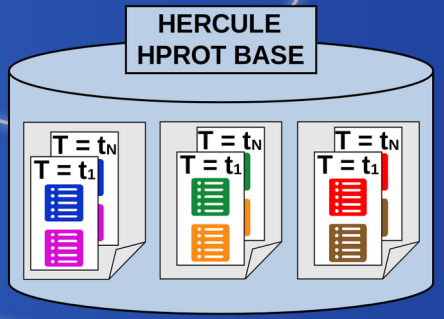
New HERCULE I/O pattern



RAMSES : dataflow separation

HProt database

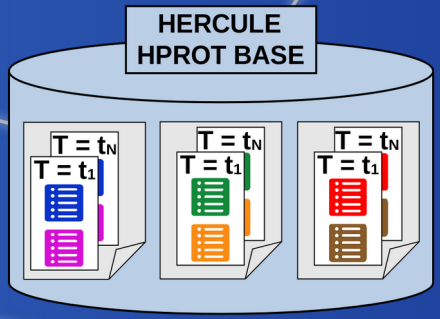
- Dedicated exclusively to **checkpoints/restarts**
- Provide a basic binary storage
  - Understandable only by the code that produced it
- Takes advantage of asynchronous I/O (No copy of data)
  - No additional memory cost



RAMSES : dataflow separation

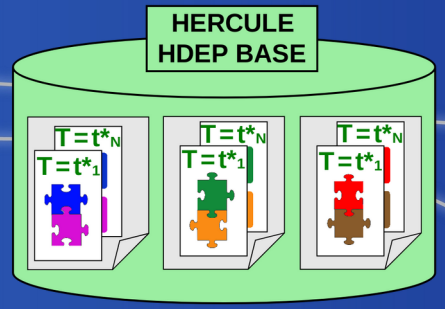
HProt database

- Dedicated exclusively to **checkpoints/restarts**
- Provide a basic binary storage
  - Understandable only by the code that produced it
- Takes advantage of asynchronous I/O (No copy of data)
  - No additional memory cost

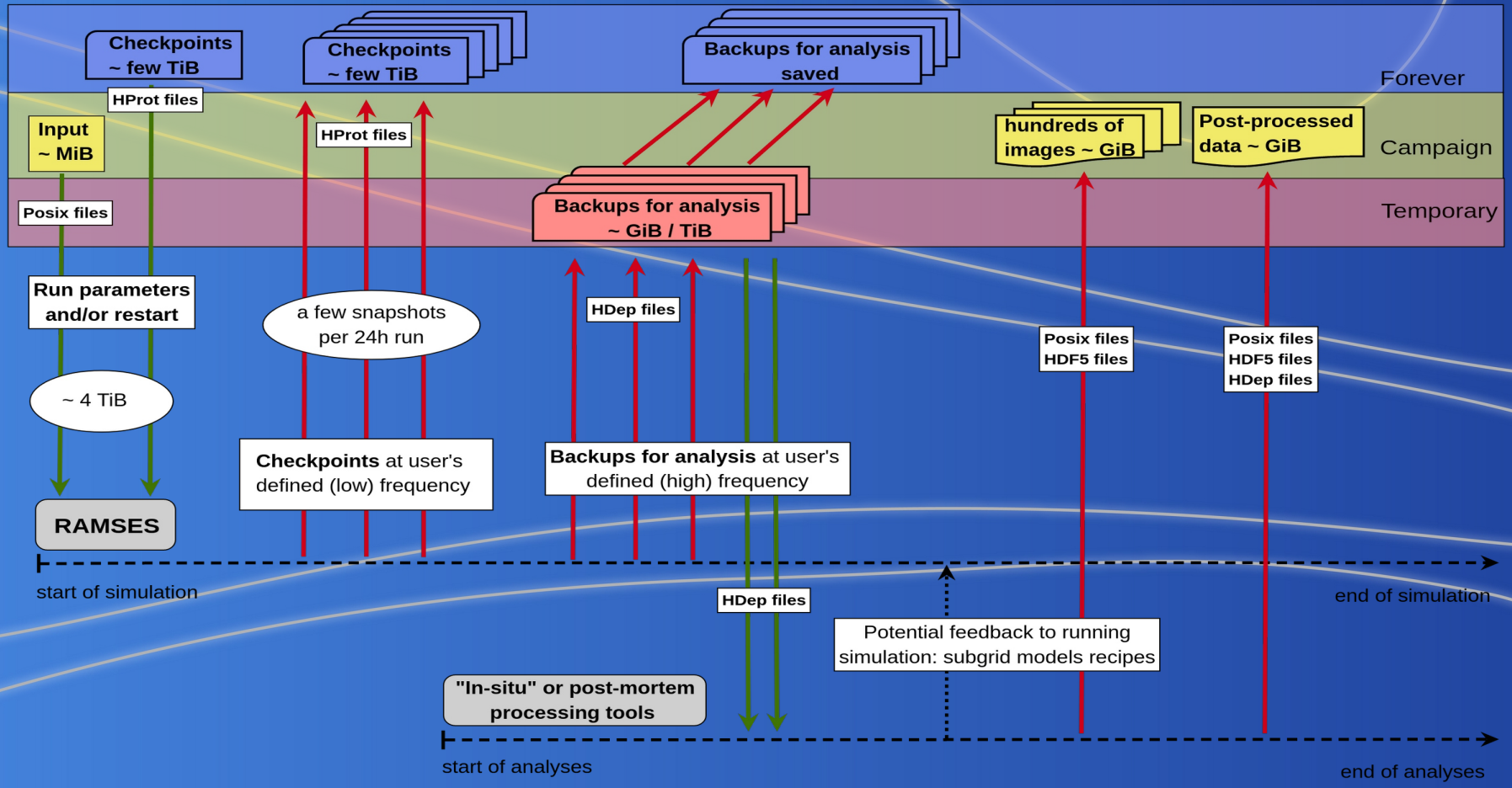


HDep database

- Dedicated exclusively to **post-processing** (analysis, visualization)
- Provide optimized data models
  - Self-described and understandable by other (post-processing codes)
- Possible restriction to a subset of the physical quantities
- Lightweight (compare to HProt and RAMSES legacy)
  - Data redundancy removal + data compression



RAMSES : workflow diagram



RAMSES : ephemeral services

NFS interface

- Read initial conditions files (few MB), once per simulation,
- Read configuration files (few kB), at each step of the simulation run session,
- Write output log files (few MB).

RAMSES : ephemeral services

NFS interface

- Read initial conditions files (few MB), once per simulation,
- Read configuration files (few kB), at each step of the simulation run session,
- Write output log files (few MB).

DASI interface (I/O intensive)

- Run session
  - Read/write checkpoint Hercule database (Hprot) : read upon restart, write upon checkpoint.
  - Write analysis Hercule databases (Hdep) : light-weight, but at high frequency.
- Analysis session
  - Read analysis Hercule databases (Hdep).

RAMSES : simulation workflow

```
# start a session for my workflow (described in the ramses_workflow.yaml file)
iosea-wf start WORKFLOW=ramses_workflow.yaml SESSION=$session

# run the workflow steps => first launch run from initial conditions
iosea-wf run SESSION=$session STEP=init NPROCS=$NCORES

# Restart simulation until final completion
for i in $(seq 1 $NRESTARTS) ; do
    iosea-wf run SESSION=$session STEP=restart NPROCS=$NCORES
done

# check status of jobs
iosea-wf status SESSION=$session

# stop session
iosea-wf stop SESSION=$session
```

Thank you for your attention.
Questions ?