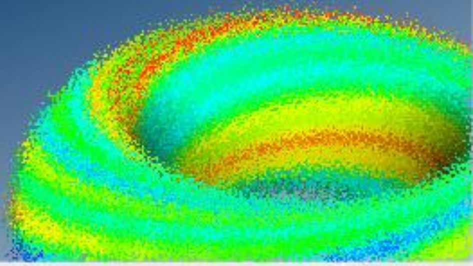
Fusion Group

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA



Progress of containerized HPC workflows with IMAS

Albert Gutiérrez Millà

Barcelona Supercomputing Center (BSC)

21/Oct/2020

Tech Dev meeting, Online

Initial consideration

- The present work is built upon **Tomek Żok's work**.
- For past TechDev slides on Tomek's work, please visit his July's 1st presentation at Indico:

https://indico.euro-fusion.org/event/236/contributions/928/attachments/366/731/imas-environment-zok.pdf

Outline

- Introduction
- Containerization
- HPC Performance
- Workflow case 1: H&CD
- Demo H&CD
- Workflow case 2: TGLF
- Thoughts & Future work

Introduction

fusion group

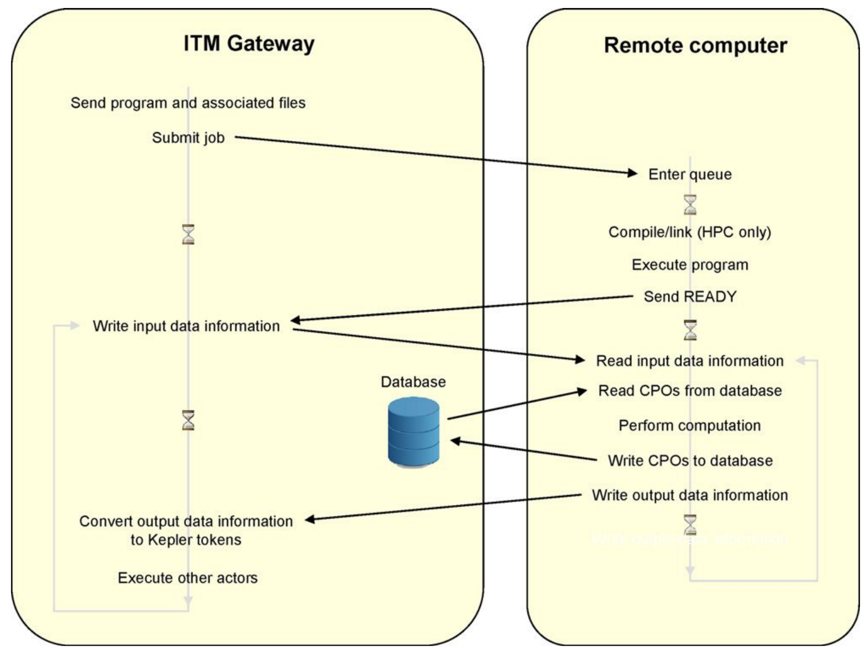
The situation

- Some codes of IMAS have **HPC requirements**, but they depend on a **supercomputer** that **does not have the framework installed**.
- Supercomputers are **complex systems** and moving HPC code needs consideration because **scalability** and **performance** could be affected.
- **IMAS** is a **heavy framework** under constant change and may not be installed in European supercomputers.

Goal: Implement **capabilities** for the execution of **integrated modelling fusion workflows** including **HPC** components on Tier-0 and Tier-1 **supercomputers**.

Previous attempts (and why they did not work)

- Built a solution based on **UNICORE**: HPC2K.
- Agreement to **install** UNICORE in coming supercomputers.
- But system administrators **refused to install** it on new machines.
- Moral: **do not rely the solution on external software**.



Requirements

- IMAS framework needs to be **moved** to the used supercomputers.
- The system should be **transparent** and **do no require installation by the system administrators.** It has the be built on the user space.
- The system needs to be **secure** and have a model that does not have a daemon running on the supercomputer login node.
- **Minimise the performance degradation** of the codes.

Idea: **use containers in user space**.

Containerization

Docker & uDocker

- Since its foundation **Docker** has been a growing **popular** tool.
- The usage of Docker **has been evaluated for HPC**.
- However, it has showed that it **can lead to escalation of privileges to get root access**.
- **uDocker** solves this issue by relying on the containers on the user space.

Singularity

- **Singularity** is a tool developed by LBL at Stanford.
- Uses **container technology**, aimed for HPC.
- Aimed for **reproducibility** and move single images where the file system is contained.
- Singularity software **is usually installed by the system administrators** but can be installed in the user space.
- List of reported clusters supporting Singularity.

uDocker vs Singularity

- Singularity has disadvantages: it is intended for **reproducibility** and not for **interaction**.
- This approach can make Singularity less flexible.
- uDocker is **simpler** and accessing the file system of the image is **straightforward**.
- While they show similar performance due to same container technologies **uDocker is more suitable for our needs**.
- Even though we have been using uDocker, it hasn't had **much activity lately** while Singularity is **constantly updated**.

HPC Performance

1. Singularity
2. uDocker - Single node
3. uDocker - Multi-node
4. uDocker - ASCOT & BIT1

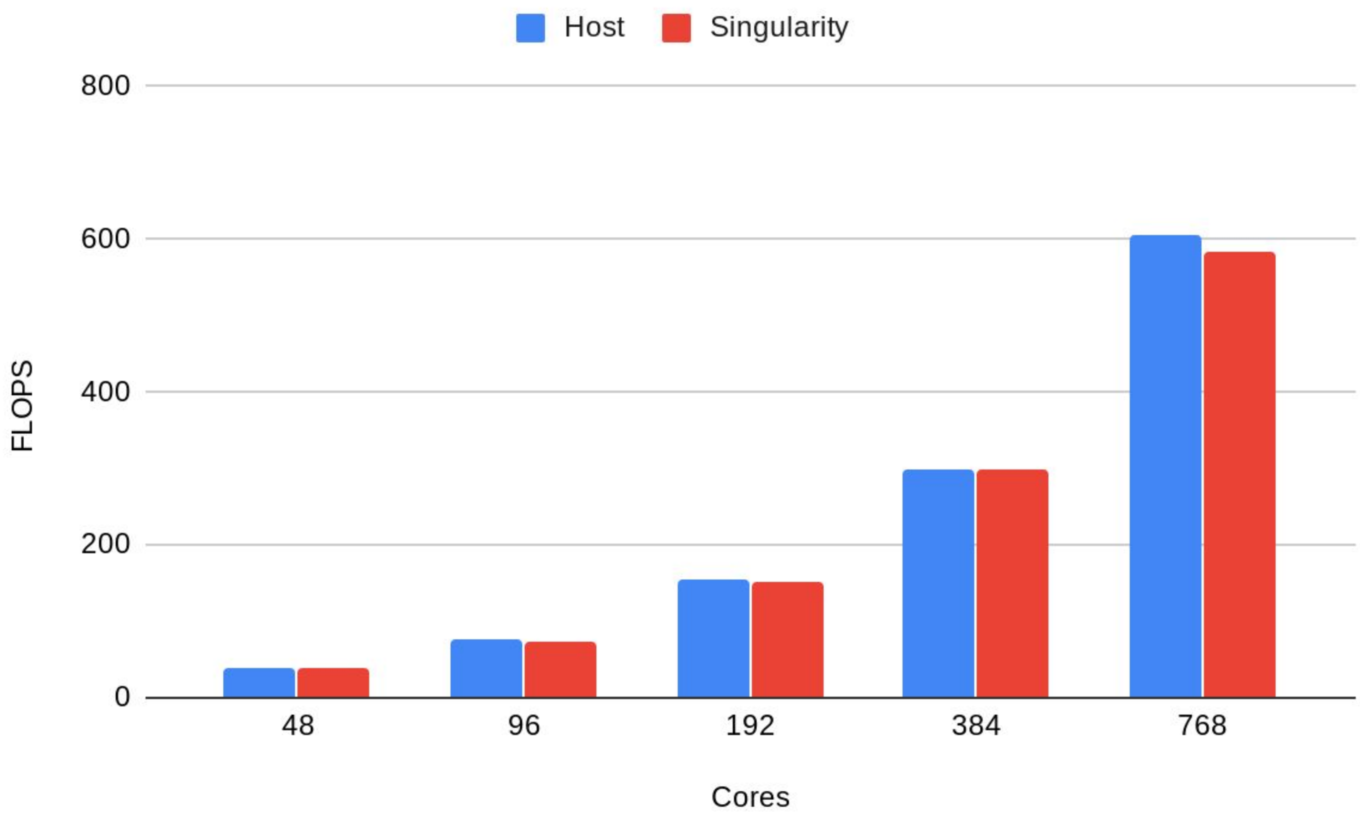
BSC fusion group

HPCG

- **HPCG** is a benchmark that performs basic operations: sparse matrix-vector multiplication, vector updates, global dot products, local symmetric Gauss-Seidel smoother, etc.
- One of the **two reference benchmark codes** to calculate the performance on top500 supercomputers list.
- Performance analysed with **Singularity** on **MareNostrum** with **Intel** drivers.

Performance Singularity HPCG

- Inter and intranode performance Intel SKL.
- Singularity shows to degrade slightly the performance no further than 4%.



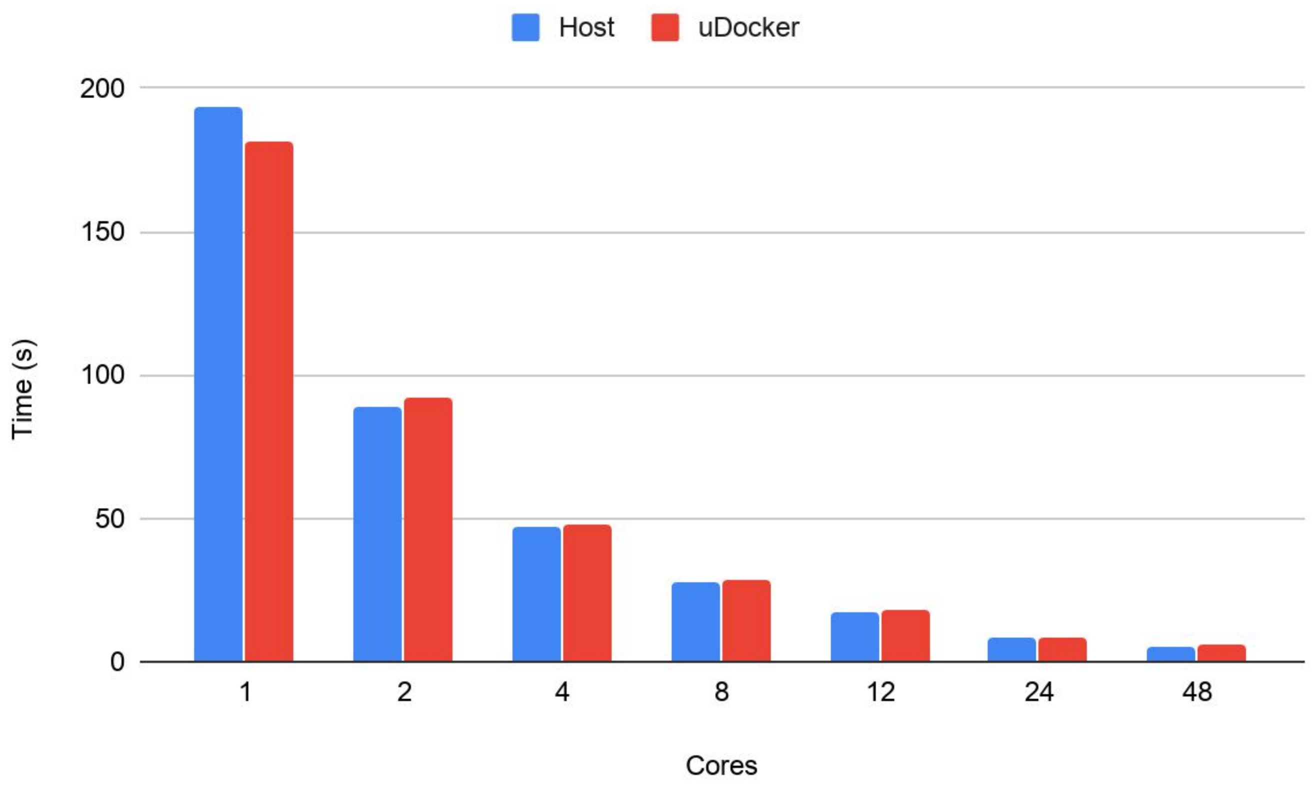
Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Performance uDocker MiniFE

- **MiniFE** is a Finite Element application for benchmarking HPC systems.
- Computation of: element-operators, assembly, sparse matrix-vector product, vector operations.
- The application has requirements **similar to applications in fusion** and it is **representative** of the workload that will be used.
- Performance performed with **uDocker** on **Marconi**.

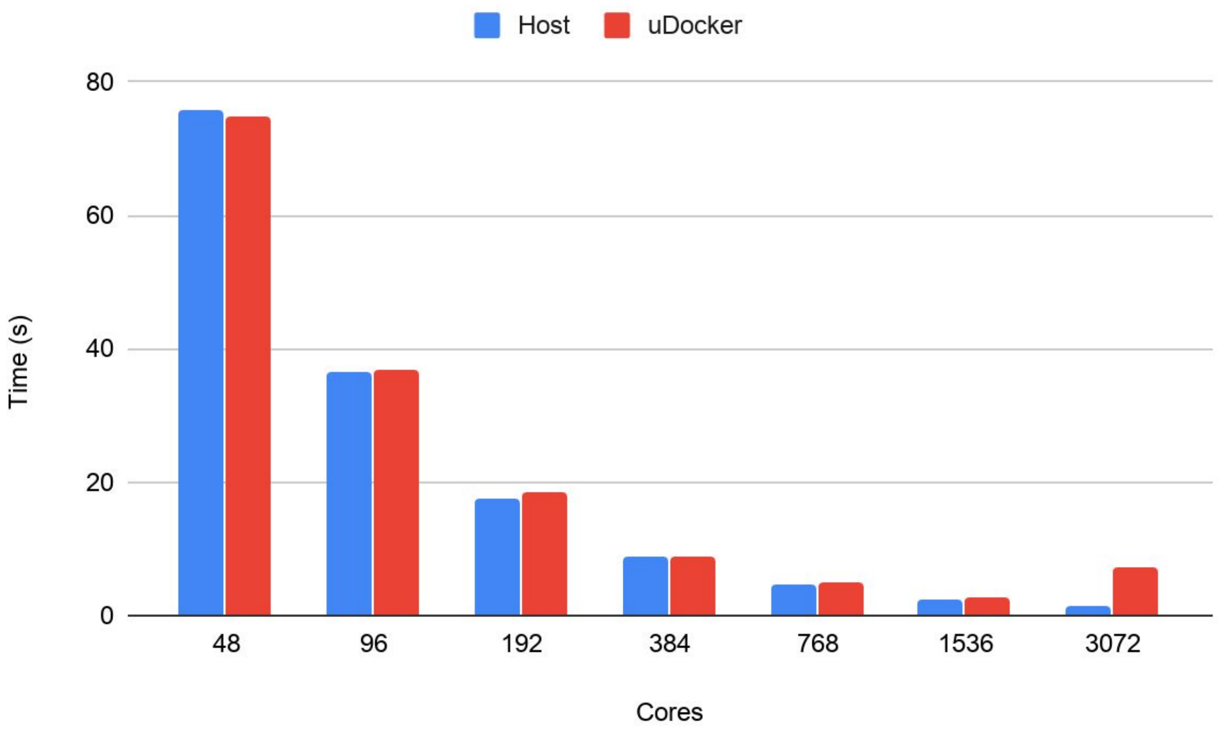
Performance uDocker MiniFE

- Intranode performance Intel SKL.
- Marconi-Fusion with OpenMPI.
- Size 256x256x256.
- Max difference 3%.



Performance uDocker MiniFE (2)

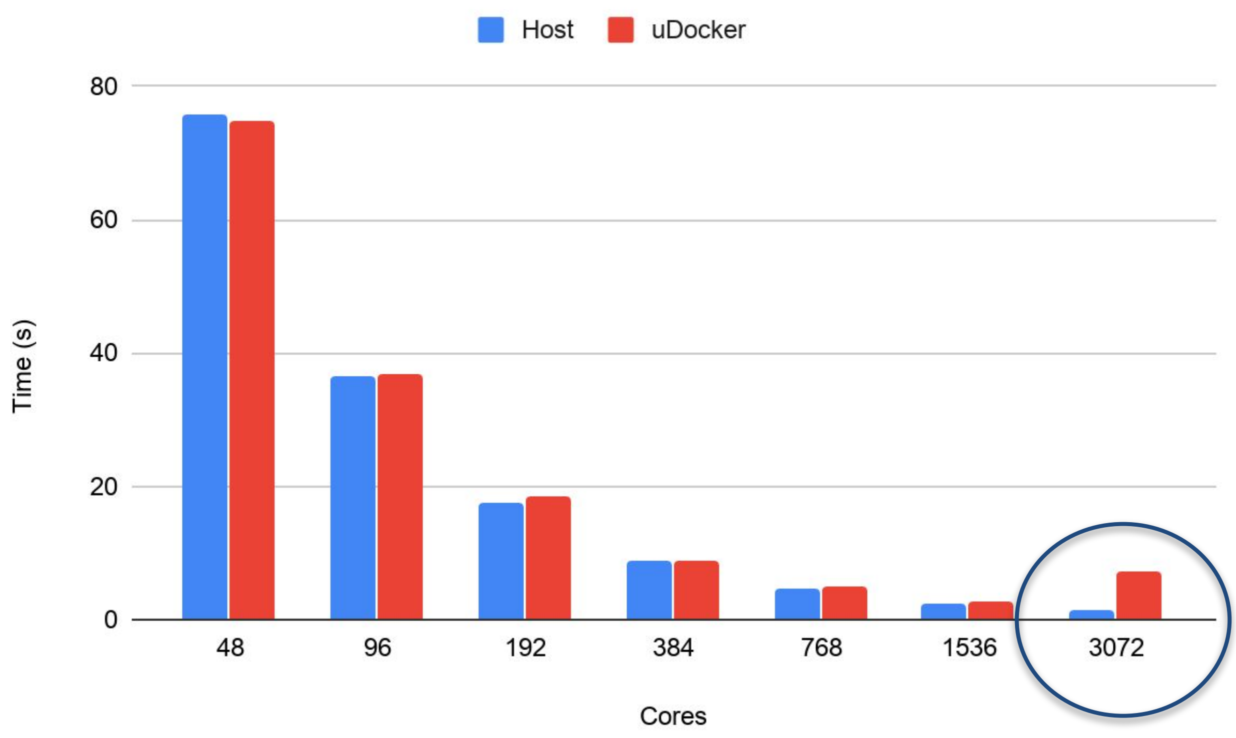
- Internode performance Intel SKL.
- Marconi with OpenMPI.
- Size 512x512x512.
- Max difference 3% until 768.



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

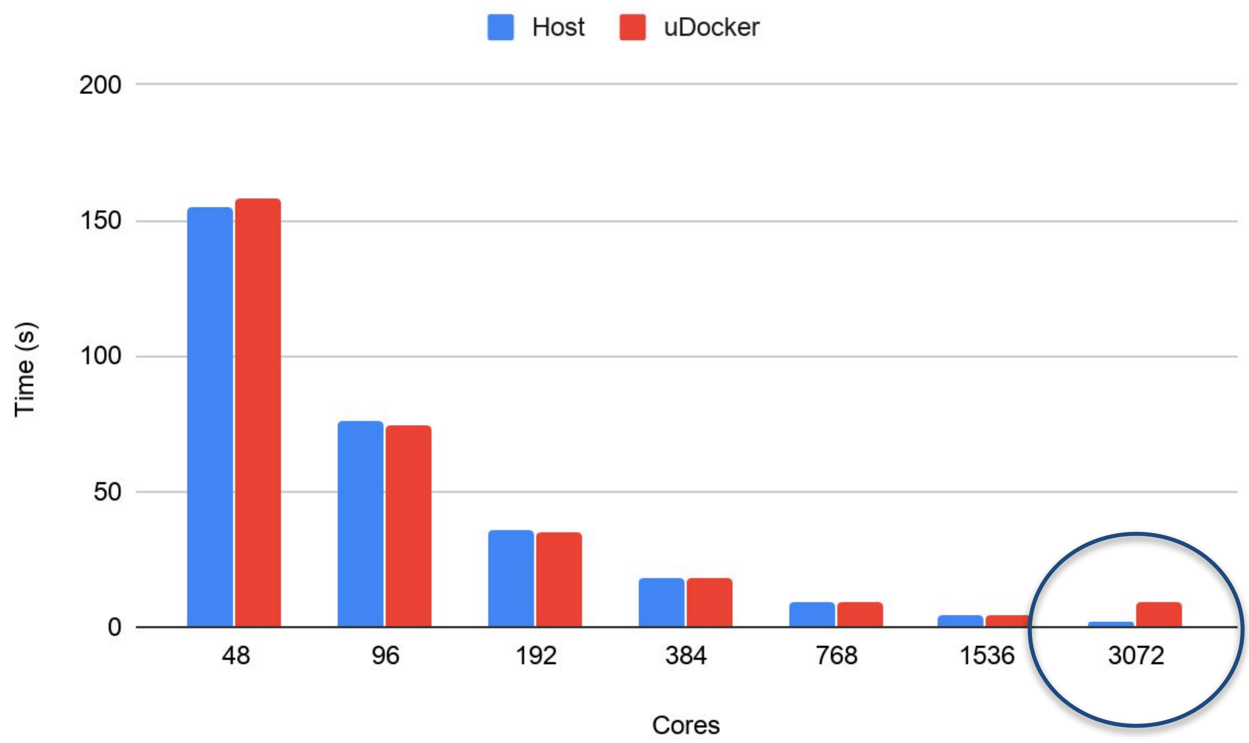
Performance uDocker MiniFE (2)

- Internode performance Intel SKL.
- Marconi with OpenMPI.
- Size 512x512x512.
- Max difference 3% until 768.
- Overhead for small problems.



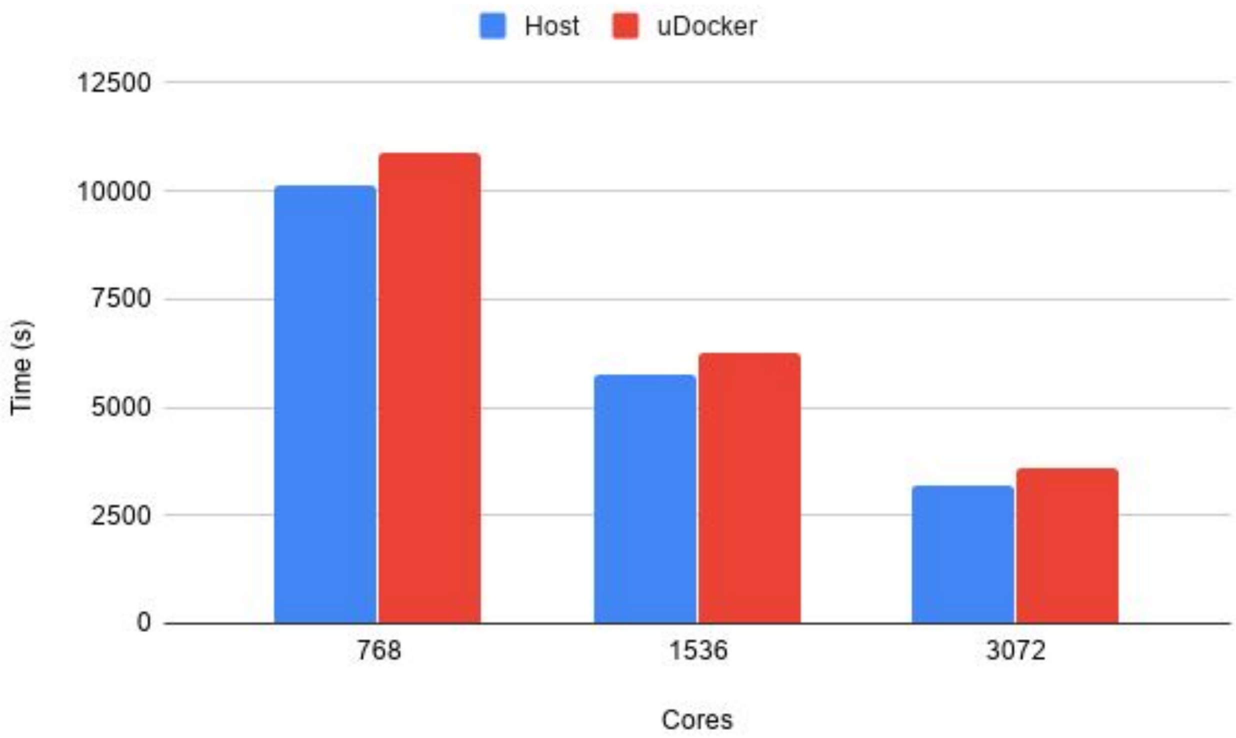
Performance uDocker MiniFE (3)

- Size **1024**x512x512.
- Max difference 3% until 768.
- Max difference 3% until 768.
- Overhead decreases with increasing size problem.



Performance uDocker ASCOT

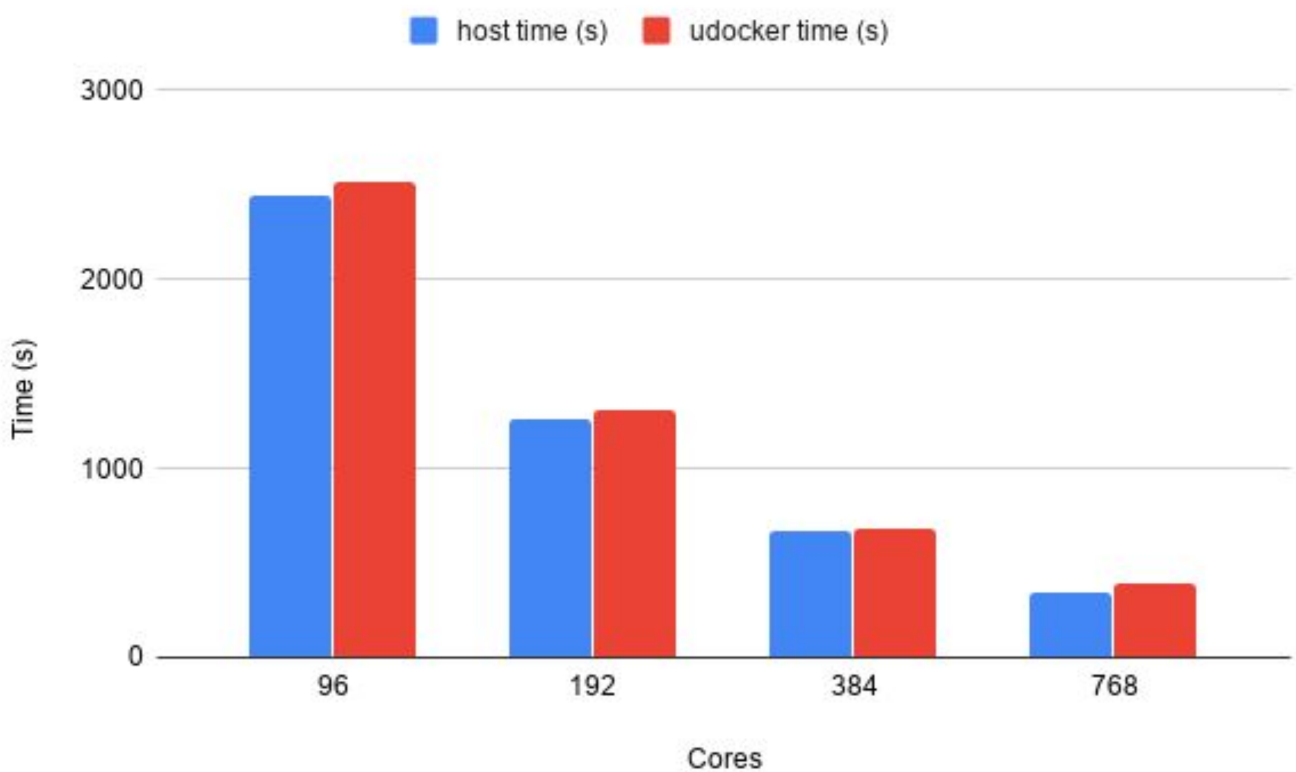
- ASCOT is a Monte Carlo orbit-following code that solves the kinetic equation.
- Max difference 7% until 3072 cores.



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Performance uDocker BIT1

- BIT1 is an electrostatic particle-in-cell (PIC) + Monte Carlo (MC) code.
- Max difference 3.5% until 768 cores.

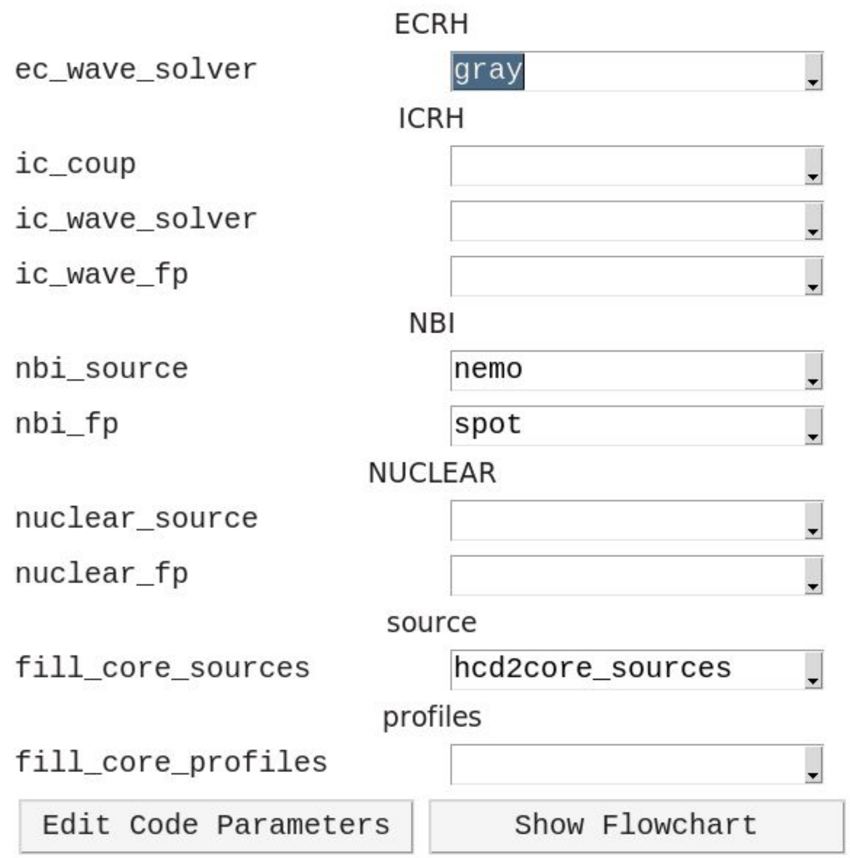


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Workflow case 1
H&CD

The H&CD python workflow

- Heating and current drive (**H&CD)** workflow developed by Mireille.
- The H&CD workflow works solely with **python actors** and **does not use Kepler**.
- Using mainly **NEMO** and **SPOT.**



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Software and data dependence

- SZIP.
- NETCDF & NETCDFF.
- NAG.
- HDF5.
- Intel MPI.
- PyAL.
- IMAS intel (libimas-ifort.so).
- Python actors released for the H&CD workflow.
- Shared files included at ITER cluster /work/imas/shared/heat/nemo/.

- For HPC workflows **we will always depend on host's MPI libraries**.

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

H&CD integration in uDocker

- The H&CD workflow was the first IMAS workflow to be used and implemented inside uDocker.
- H&CD already work with **IDS**, uses the **IMAS library** and involves several codes including **SPOT** which runs using **MPI**.
- Current IMAS Docker image **relies on GCC**.
- However, H&CD depends on an Intel IMAS version not compiled inside the Docker image which needs to be loaded for the execution of the workflow.

Loading the environment inside uDocker: POBAR

- A first approach to manage an image able to **bind the hosting system** inside as well as the environment in the ITER cluster.
- POBAR is a tool that **captures the environment**, binds directories and loads environment variables once the session has started.
- POBAR was tested at ITER cluster and **managed to run the H&CD inside uDocker**.
- Proof-of-concept, showed the limitations of the approach regarding **reproducibility**.

```
#!/bin/bash

# Fill with the configuration to be binded inside uDocker
CONFIG=$HOME/hcd_last/config_hpc.sh

module purge
env_before=$(env)
source $CONFIG
env_after=$(env)
eval $(python3 pobar.py $env_before $env_after)
```

```
[gutiera@hpc-login02 pobar]$ ./pobar.sh
H&CD actors taken from /home/ITER/gutiera/public/imas_actors

 ********************************************************************************
 *                                                                              *
 *               STARTING 425279da-b031-3e6c-9af9-987cc44dc04c                  *
 *                                                                              *
 ********************************************************************************
 executing: bash
Selection fulfills all actor selection rules
-- Open input and output file --
---------------------------------------------
---- Enter time loop of the H&CD wrapper ----
---------------------------------------------
Step = 1/18
Time =  5.00 s
dt   = 20.00 s
  Get core_profiles
  Get equilibrium
  Get ec_launchers
  Get ic_antennas
  Get nbi
  Get distribution_sources
  Get waves
  Get wall
  Get distributions
Execute H&CD workflow for current time slice
-- Step 1: Source codes and Wave solvers
--NEMO--
 NORMAL MODE
 ===============================================================
 START OF NEMO
 # PROCESSORS USED FOR THE CALCULATION =  1
 - READ INPUT VARIABLES
 --> SOURCES TREATED AS RECTANGULAR
 --> USE ADAS CROSS SECTIONS
 Wall not detected --> use SOL radius instead
 ----------------------------
  Number of ions =  8
 ----------------------------
```

Singularity and H&CD workflow

- To focus on the reproducibility and the containerization of IMAS another approach was to build a new image with **Singularity** based on **Intel IMAS.**
- The image was built building IMAS libraries **binding the host Intel compiler** to the compilation process.
- Even though the IMAS libraries where compiled and that code could be compiled, there were some **issues with the image**.
- Moreover, Singularity is considered “good friends” with Docker and a built Docker image can be later moved to Singularity.

Building the H&CD image

- New approach: **reproduce an existing and working setup** inside the Docker image and then move it to a reference supercomputer for testing.
- The **Gateway structure has been reproduced inside uDocker** and some files were copied inside and configured accordingly.
- This was done analysing the **dependencies** and **requirements** (ldd, environment, module, environment, etc).
- After setting it up **an image with H&CD could be released and working**.
- The H&CD image has been tested in **Marconi and it is currently working** with **IMAS 3.28.1**.

```
imas@r000u06l01:~/hcd_last$ python -c "import sys; sys.path.append('interface'); sys.path.append('workflow'); from hcd_wrapper import hcd_wrapper; hcd_wrapper('/home/imas/hcd_last/run_configurations/run_1019_150628/')"
Selection fulfills all actor selection rules
-- Open input and output file --
---------------------------------------------
---- Enter time loop of the H&CD wrapper ----
---------------------------------------------
Step = 1/18
Time =  5.00 s
dt   = 20.00 s
  Get core_profiles
  Get equilibrium
  Get ec_launchers
  Get nbi
  Get wall
  Get distribution_sources
  Get distributions
  Get waves
Execute H&CD workflow for current time slice
-- Step 1: Source codes and Wave solvers
--NEMO--
 NORMAL MODE
 ================================================================
 START OF NEMO
 # PROCESSORS USED FOR THE CALCULATION =  1
 - READ INPUT VARIABLES
 --> SOURCES TREATED AS RECTANGULAR
 --> USE ADAS CROSS SECTIONS
 Wall not detected --> use SOL radius instead
 ----------------------------
  Number of ions =  8
 ----------------------------
 Followed ion:
   - Mass number   A =   2
   - Charge number Z =   1
 ----------------------------
 - CALCULATE BEAM DEPOSITION
   Number of running PINIS  = 32
   Total injected power     =   33.000 MW
 OTHER NEMO CALLS
```

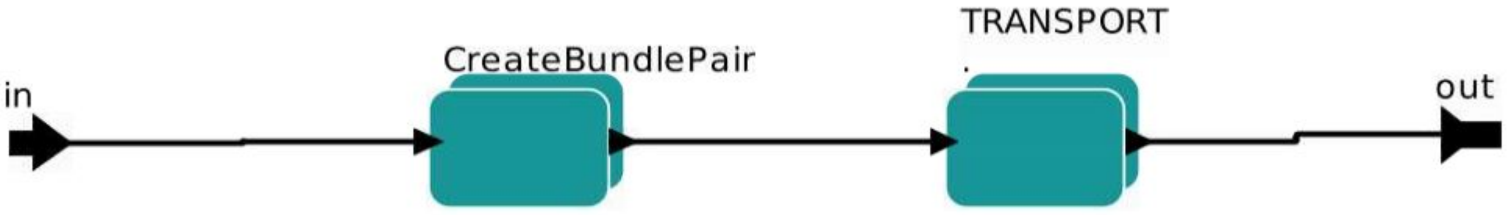
Demo H&CD

Workflow case 2
TGLF

fusion
group

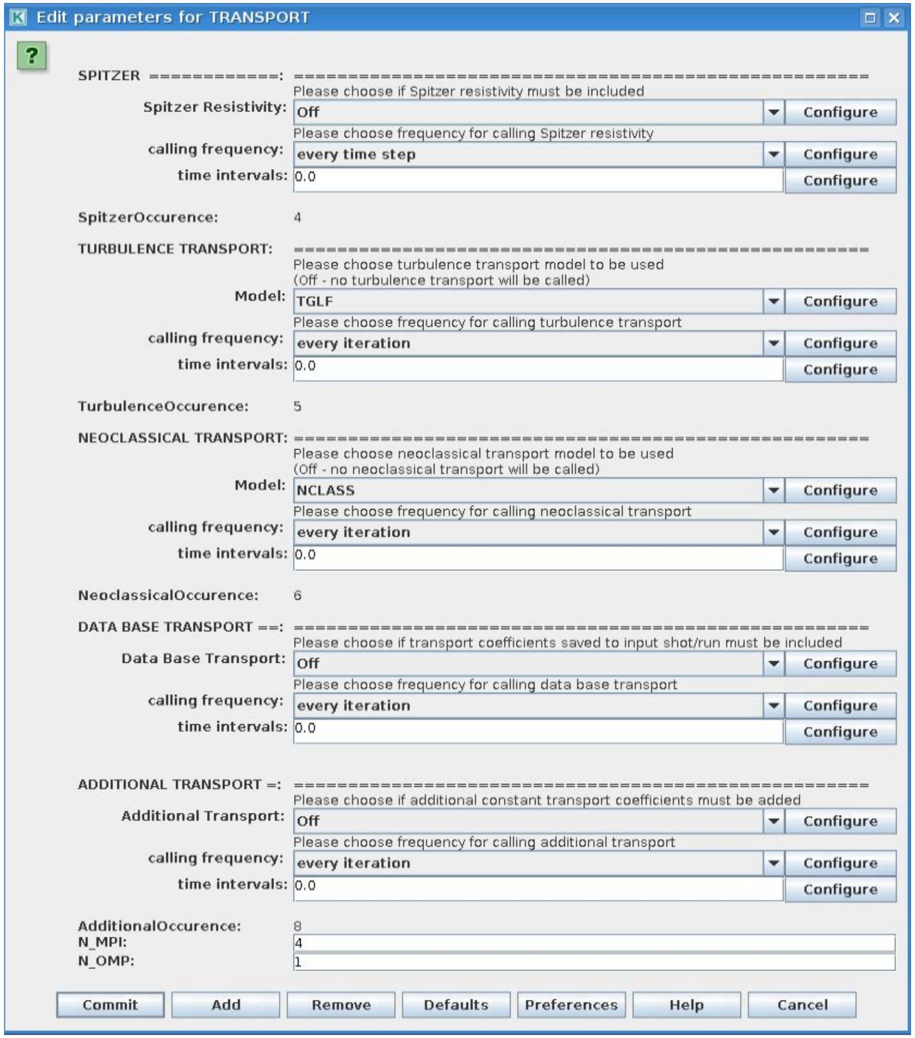
TGLF

- TGLF is not a released workflow, instead and **ad-hoc minimal Kepler workflow with an HPC code**.
- It is intended for the **evaluation of the containerized capabilities**.
- This work is **still under development** even though it is in an advanced stage.



Dependencies

- Dressed Kepler.
- Intel IMAS.
- Intel MPI.
- NAG.
- Lib Interpos.
- MKL.
- Lib AMNS.



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Current status

- The workflow is correctly loading the libraries and starts the execution in **Marconi**.
- Able to load actors from **Dressed Kepler** (of special interest for ETS work).
- Running **Intel MPI from Kepler actor** and opening correctly the MPI_COMM_WORLD.
- However, currently there is a SIGSEGV that needs to be debugged (work in progress..).

```
      [run] KEPLER DIR: /pfs/work/g2agutie/my_imas_keplers/TGLF2/kepler
      [run] 2020-10-19 21:17:09:674 [empty_core_sources] RELOAD: 1 [ms] : /pfs/work/g2agutie/my_imas_keplers/TGLF2/kepler/imas/lib64/libempty_core_
sources_def.so
      [run] 2020-10-19 21:17:09:688 [CHANGEOCC] RELOAD: 1 [ms] : /pfs/work/g2agutie/my_imas_keplers/TGLF2/kepler/imas/lib64/libCHANGEOCC_def.so
      [run] 2020-10-19 21:17:09:700 [TRANSPORT_COMBINER] RELOAD: 42 [ms] : /pfs/work/g2agutie/my_imas_keplers/TGLF2/kepler/imas/lib64/libTRANSPORT_
COMBINER_def.so
      [run] 2020-10-19 21:17:09:751 [TCIIGNORE] RELOAD: 1 [ms] : /pfs/work/g2agutie/my_imas_keplers/TGLF2/kepler/imas/lib64/libTCIIGNORE_def.so
      [run] 2020-10-19 21:17:09:805 [TCITGLF] RELOAD: 4 [ms] : /pfs/work/g2agutie/my_imas_keplers/TGLF2/kepler/imas/lib64/libTCITGLF_def.so
      [run] 2020-10-19 21:17:09:817 [TCIEDWM] RELOAD: 1 [ms] : /pfs/work/g2agutie/my_imas_keplers/TGLF2/kepler/imas/lib64/libTCIEDWM_def.so
      [run] 2020-10-19 21:17:09:826 [TCIWEILAND] RELOAD: 1 [ms] : /pfs/work/g2agutie/my_imas_keplers/TGLF2/kepler/imas/lib64/libTCIWEILAND_def.so
      [run] 2020-10-19 21:17:09:835 [TCIGLF23] RELOAD: 2 [ms] : /pfs/work/g2agutie/my_imas_keplers/TGLF2/kepler/imas/lib64/libTCIGLF23_def.so
      [run] 2020-10-19 21:17:09:845 [TCIQLK] RELOAD: 445 [ms] : /pfs/work/g2agutie/my_imas_keplers/TGLF2/kepler/imas/lib64/libTCIQLK_def.so
      [run] 2020-10-19 21:17:10:301 [EMPTY_TRANSPORT] RELOAD: 1 [ms] : /pfs/work/g2agutie/my_imas_keplers/TGLF2/kepler/imas/lib64/libEMPTY_TRANSPOR
T_def.so
      [run] 2020-10-19 21:17:10:312 [spitzer_resistivity] RELOAD: 7 [ms] : /pfs/work/g2agutie/my_imas_keplers/TGLF2/kepler/imas/lib64/libspitzer_re
sistivity_def.so
      [run] 2020-10-19 21:17:10:330 [TCIIGNORE_NEOCLASSICAL] RELOAD: 1 [ms] : /pfs/work/g2agutie/my_imas_keplers/TGLF2/kepler/imas/lib64/libTCIIGNO
RE_NEOCLASSICAL_def.so
      [run] 2020-10-19 21:17:10:341 [TCINEO] RELOAD: 3 [ms] : /pfs/work/g2agutie/my_imas_keplers/TGLF2/kepler/imas/lib64/libTCINEO_def.so
      [run] 2020-10-19 21:17:10:353 [TCINCLASS] RELOAD: 2 [ms] : /pfs/work/g2agutie/my_imas_keplers/TGLF2/kepler/imas/lib64/libTCINCLASS_def.so
      [run] 2020-10-19 21:17:10:365 [database_transport] RELOAD: 6 [ms] : /pfs/work/g2agutie/my_imas_keplers/TGLF2/kepler/imas/lib64/libdatabase_tr
ansport_def.so
      [run] 2020-10-19 21:17:10:381 [TCIANALYTICAL] RELOAD: 1 [ms] : /pfs/work/g2agutie/my_imas_keplers/TGLF2/kepler/imas/lib64/libTCIANALYTICAL_de
f.so
      [run] cp -r /pfs/work/g2agutie/my_imas_keplers/TGLF2/kepler/imas/src/org/iter/imas/TCITGLF/* .;mpirun -perhost 1 -l -np 4   ./bin/TCITGLF.exe
      [run] [0]  myrank           0
      [run] [2]  myrank           2
      [run] [3]  myrank           3
      [run] [1]  myrank           1
      [run] [2]   use_bper            F
      [run] [2]   use_bpar            F
      [run] [2]   adiabati_electrons  F
      [run] [2]   fastions (jintrac)  F
      [run] [2]   dump_flag           F
      [run] [3]   use_bper            F
      [run] [3]   use_bpar            F
      [run] [3]   adiabati_electrons  F
      [run] [3]   fastions (jintrac)  F
      [run] [3]   dump_flag           F
      [run] [3]   alpha_p               1.00000000000000
      [run] [3]   alpha_e               1.00000000000000
      [run] [3]   alpha_mach           0.000000000000000E+000
      [run] [2]   alpha_p               1.00000000000000
      [run] [2]   alpha_e               1.00000000000000
      [run] [2]   alpha_mach           0.000000000000000E+000
      [run] [2]   alpha_zf              1.00000000000000
      [run] [2]   alpha_quench         0.000000000000000E+000
      [run] [3]   alpha_zf              1.00000000000000
      [run] [3]   alpha_quench         0.000000000000000E+000
      [run] [3]   sat_rule                        0
      [run] [3]   xnu_factor            1.00000000000000
      [run] [3]   epsil                0.000000000000000E+000
      [run] [3]   rhomax               0.800000000000000
```

Thoughts & Future work

Small size & easy-to-use & general-purpose

- We **cannot have** a small image, easy-to-use and general purpose.
- Trying to use a general-purpose image to be binded **breaks the essential philosophy of containers** which are closed pieces.
- Packing all the possible dependencies, workflows and IMAS versions in one file would create a **huge image**.
- Our approach: **being easy to use is much more important than being general-purpose**.
- There are a limited number of HPC IMAS workflows. Idea: **work with releases**.
  - Work with **stable versions** of workflows and codes.
  - **Release these workflows** so they can be moved in an image and used by the community for their **large runs**.

Future work

- Debug and analyse the issue with TGLF TCI execution.
- **Hands-on session** on this work on a general Code Camp.
- Expand test to **other supercomputers**.
- Multi-node execution: **MPI-Kepler-workflow** instead of Kepler-MPI-Workflow.
- Next workflow case: **ETS**?
  - It is the workflow with **highest interest** and also the **highest complexity**.
  - **Dressed Kepler has already been included in the image** which is a good advance in ETS direction.
  - Dmitriy has already provided instructions and an ETS6 setup.

Acknowledgements

Special thanks for their help and contribution to:

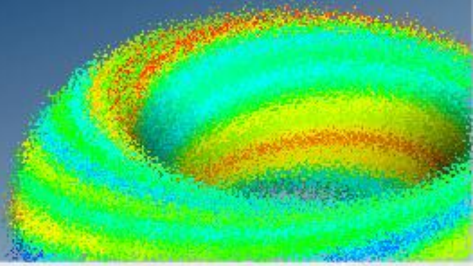
- Tomek Żok.
- Dmitriy Yadykin.
- Mireille Schneider.
- Michal Owsiak.
- Marcin Plociennik.

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Fusion Group

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

EXCELENCIA SEVERO OCHOA



Thank you

fusion.bsc.es

@Fusion_BSC

fusion@bsc.es

albert.gutierrez@bsc.es