MAX-PLANCK-INSTITUT
FÜR PLASMAPHYSIK

ETS (MUSCLE 3 implementation)

Meeting # 15 of the SC on the EUROfusion-IO development of IMAS 2024-03

David Coster, Dmytro Yadykin, Rui Coelho, Thomas Johnson, Pär Strand

EUROfusion

This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

Building a new European Transport Simulator (ETS) using Persistent Actor Framework (PAF) based on MUSCLE3

- Background:
  - **New paradigm based on independent communicating programs**
    - **Each program can be serial, multi-threaded, MPI or hybrid**
  - **Fortran, C++ codes can be prepared as a subroutine receiving IDS inputs and returning IDS outputs that is “iWrap”ped**
    - **Each code (or actor) is linked to the muscle3 library**
  - **A YAML file then describes the data flow between the actors**
    - **Logic of the workflow resides in the YAML file supported by specialised actors**

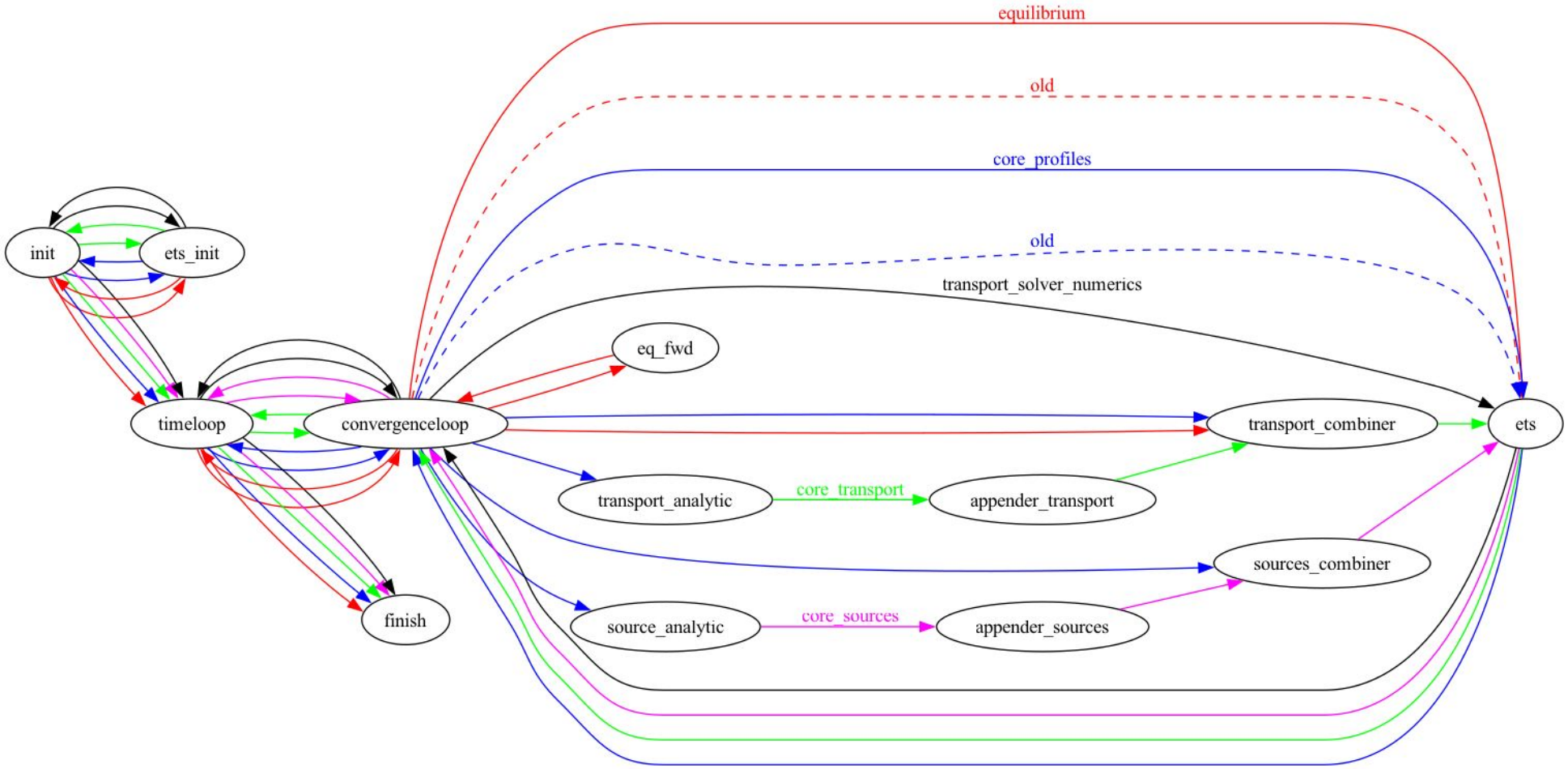
Simple version of the workflow

Simple version of the workflow that shows the key components

Uses

- Analytical sources
- Analytical transport
- Unchanging equilibrium

More complete version has the same basic logic, but a lot more actors



Most recent work

- UQ applied to ETS-PAF workflows

- Inclusion of the HCD workflow in an ETS-PAF workflow
  - Successful demonstration using ECRH
    - **Successful comparison between Gray and Torbeam**

- Inclusion of W in the simulation
  - Sources from ionization and recombination included to determine the charge state balance of each W charge state
  - Busy looking at the conservation of the summed over charge state W density

Actors in ETS6 (Kepler version, IMAS)

- **Equilibrium**: chease, helena, gkmhd, eqinput, database_equilibrium, database_plasma_boundary
- **Transport:** spitzer_resistivity, edwm, tglf, qlk, weiland, glf, bgb, mmm, nclass, neo, database_transport, analytical, neutral_transport, transport_combiner
- **Transport:** spitzer_resistivity, edwm, tglf, qlk, weiland, glf, bgb, mmm, nclass, neo, database_transport, analytical, neutral_transport, transport_combiner
- **Edge:** penn, cec, solpsz1
- **Solver:** transport_solver
- **Service:** ets_init, update_core_profiles, changeocc, empty_transport, empty_core_sources, empy_waves, empty_distribution_sources, empty_distributions, combine_core_profiles, database_profiles, update_numerics, check_convergence, fill summary

dimitriy.yadykin@chalmers.se

ETSpy

- Object oriented ETS fully developed in Python and using iWrap actors.
  - **All actors using *iWrap* e.g. CHEASE, TCI suite, GRAY,…**
  - **Built to be agile and fast.**
  - ***Parameter* and *Plasma* bundle concepts inherited from ETS-Kepler → easy global data propagator between all actors.**
  - ***All workflow components* derive from same *abstract class* → trivial to integrate a new actor in the workflow**
    - **Each of the TCI transport modules took <5min to fully integrate in the workflow (ready to use).**
    - **HCD integration also trivial (granularity/modularity/clarity makes integration easy)**
- Development/benchmarking in parallel to ETS-PAF using progressively more physics.
  - ***Analytic, NCLASS, Bohm-GyroBohm, TGLF, GRAY.***
- Work is ongoing to consolidate/expand physics code and sub-workflows (HCD, MHD, fuelling) portfolio and collaboration with TSVV11/ITER already discussed.

rcoelho@ipfn.ist.utl.pt

Some key needs

- A better model for encapsulation
  - The so far implemented “encapsulation” is *ad hoc* and implemented by the workflow designer
  - Will need something better when we incorporate the ETS workflow into the UQ workflow

- A faster serialization/deserialization method released (not just demonstrated)
  - 95.6% of the cpu time of the most expensive module (which accounted for 35.8% of the total time) was spent in serialization/deserialization

- A way for mapping requirements of the simulation (which actors, which options, …) onto the YMMSL files and the command line for running muscle3

Collaborations in place

- Collaboration between the various ETS developers
- Within TSVV11 (benchmarking, new physics)
- With ITER