GWDG
Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen

Proposal of a New Open Source GUI for OpenFOAM on HPC Systems

Patrick Höhn (patrick.hoehn@gwdg.de), Julian Kunkel



30.06.2026

Agenda

- Introduction
- Approach
- Snakeflow
- Wrap-Up

OpenFOAM Related activities at NHR-Nord@Göttingen / GWDG

- Porting of code between different OpenFOAM-flavours, e.g.:
  - CFDEMcoupling from OpenFOAM 6 to foam-extend-4.0
  - preCICE OpenFOAM-adapter to foam-extend
- Custom development of code, e.g.:
  - development of new force models in CFDEMcoupling
  - implementation of additional functionality in CFDEMcoupling
- Porting of code between OpenFOAM versions, e.g. from foam-extend-4.1 to foam-extend-5.0
- Custom container creation: e.g.: for reproducible research results
- Available Modules on the cluster:
  - OpenFOAM.org v6, v7, v8, v10, v11
  - OpenFOAM.com v2306, v2312, v2406, v2412
  - foam-extend v4.1, v5

Motivation

- Step learning curve when beginning with OpenFOAM
- Many users not familiar with usual terminal based workflows in OpenFOAM
- Easier editing of meshes
- Competition with commercial products

Existing GUIs

- Open Source:
  - baram
  - insightCAE
  - FreeCAD CFDof and CFD addons
  - SplashFOAM
  - HELYX-OS
- Commercial:
  - FEATool Multiphysics
  - SimFlow
  - SimWorks
  - Manitumflow
  - Splash-CFD
  - HELYX

Challenges

- Maintainence of existing code
- Adjustment to new OF versions
- Economic funding model
- Advanced users using CLI interfaces

SplashFOAM

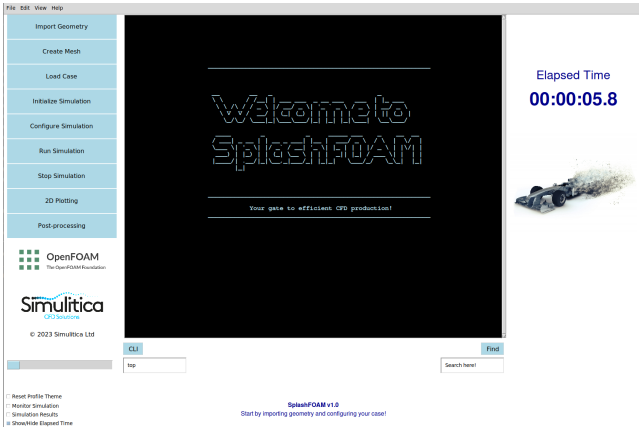


Figure: SplashFOAM (Source: https://github.com/cfddose/Splash)

Workflows

- We already have Allrun scripts, don't we?
- Other domains, e.g. biosciences, use workflow systems
- What is the benefit of a workflow system?
- What distinguishes a workflow system from a "pipeline"?
- Possibility for parametric studies (e.g. doi:10.1016/j.ces.2021.116807)
- Ability for better reproducibility with curated workflows
- Can be represented as a Directed Acyclic Graph

Snakemake

- Very popular workflow system in biosciences (**>1e6** downloads since 2015, **>1300** citations, **>7** citations per week since 2021)
- Easily exentable by plugins for executors and storage systems
- Worfklow files written in simple domain specific language based on python
- Focus on reproducibility
- Rule-based design
- Workflow graph generated based on given in- and outputs of rules
- Already proven as usuable with OpenFOAM (e.g. doi:10.1016/j.ces.2021.116807)

Allrun Scripts

```
1  #!/bin/sh
2  # Source tutorial run functions
3  . $WM_PROJECT_DIR/bin/tools/RunFunctions
4
5  runApplication blockMesh
6  runApplication snappyHexMesh
7  \rm -rf constant/polyMesh
8  \mv 3/polyMesh constant
9  \rm -rf 1 2 3
10 runApplication simpleFoam
```

- Motorbike tutorial
- Based on foam-extend-4.1
- Similar structure compared to scientific workflows

Corresponding Snakefile

```
import foamlib

result = directory(str(foamlib.FoamCase(".").control_dict['endTime']))

rule all:
    input: result, "results.png"

rule blockMesh:
    input: "constant/polyMesh/blockMeshDict"
    output: "constant/polyMesh/boundary"
    shell:
        """
        blockMesh || true
        """
```

Corresponding Snakefile

```
rule snappyHex:
    input: "constant/polyMesh/boundary",
           "system/snappyHexMeshDict"
    output: "constant/polyMesh/faceZones"
    shell:
        """
        snappyHexMesh || true
        rm -rf constant/polyMesh
        mv 3/polyMesh constant
        rm -rf 1 2 3
        """

rule singleCore:
    input: "constant/polyMesh/faceZones" if exists("system/snappyHexMeshDict") else "constant/polyMesh/boundary"
    output:
        result,
        "results.foam"
    params:
        application=str(foamlib.FoamCase(".").control_dict['application'])
    shell:
        """
        {params.application} || true
        touch results.foam
        """
```

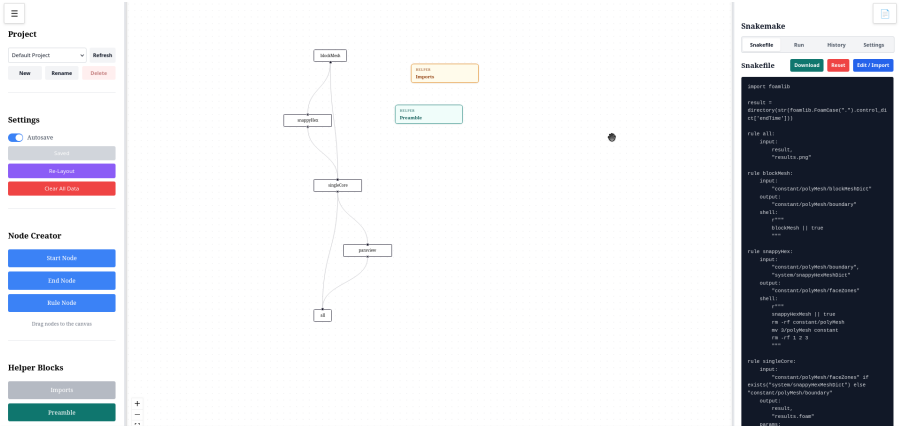
Corresponding Snakefile

```
rule paraview:
    input: "results.foam"
    output: "results.png"
    shell:
        """
        paraview
        """
```

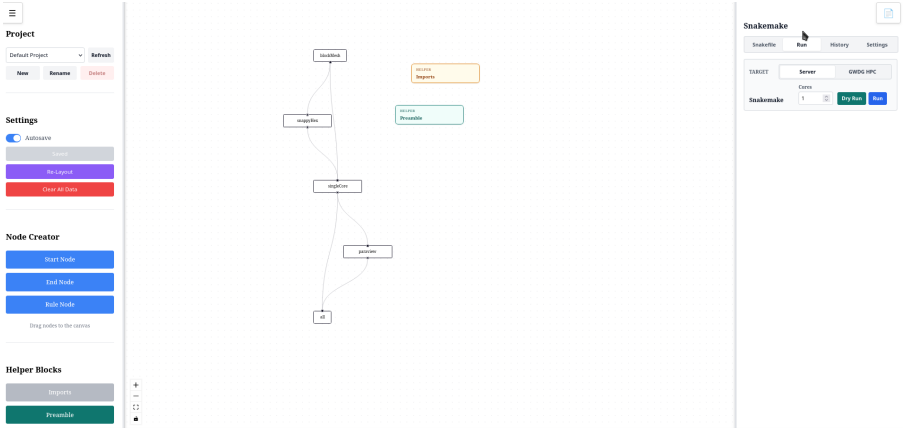
Introducing Snakeflow

- GUI to visualize generated Snakemake workflows
- Drag and drop creation fo Snakefiles
- Includes data transfer from and to cluster
- Inspired both by SplashFOAM and Ansys RSM
- Enables local and remote execution on cluster
- First working prototype with more features to be implemented

Snakeflow GUI



Snakeflow GUI



Future Plans

- Integration of in-situ visualization
- Integration of more OF versions
- Implementation of further functionality of Snakemake
- Script to automatically convert Allrun script to Snakefile
- NLP Interface to setup cases

Wrap-Up

- Prototype implementation working
- More block templates to be implemented
- Planned integration of in-situ visualization based on trame and catalyst
- Planned to be maintained for major OpenFOAM versions
- Easy reproducibility based on the container functionality of Snakemake