

Performance Evaluation of OpenFOAM on Juelich Supercomputing Facilities (JURECA, JUWELS and JUSUF)

^a Abouzar Ghasemi,^a Paul Gibbon^a and Steven B. Beale^b

^b Juelich Supercomputing Center, Forschungszentrum Juelich GmbH

Institute of Energy and Climate Research (IEK-14), Forschungszentrum Juelich GmbH

The Forschungszentrum Juelich is utilizing OpenFOAM to perform calculations for electrochemical devices such as electrolyzers and fuel cells. For some time now, improving the performance of the OpenFOAM software suite on HPC facilities has posed a challenge for users wanting to scale up problem sizes to several tens of million grid cells. To keep time-to-solution within reasonable limits, this requires scalability to many thousands of compute cores. Herein, we investigate the performance and scalability of the OpenFOAM-v1912 for the well-known motorbike benchmark test case, including a strong scaling study of the SimpleFoam solver at the Juelich supercomputing facilities (JUWELS, JURECA, JUSUF). Later we will consider in-house electrochemical models, with specialized solvers and domain decomposition techniques. JURECA was built in 2015 and utilizes a Haswell processor with 24 cores per node, JUWELS was built in 2018 based on a Skylake processor and with 48 cores per node, and JUSUF was built in 2020, based on an AMD EPYC 7742 with 128 cores per node. Scalability limits of up to 3000 and 1000 compute cores respectively is found for cases with and without writing data output to disk. It is shown that the impact of the output can be quite severe when using the default output generation option, but that this can be substantially mitigated by making use of the ADIOS library. By analysing the most time-consuming MPI function calls (collective and point-to-point), as well as the performance of the different MPI implementations, OPENMPI4.02, INTEL MPI2018/2019 and PARASTATION-MPI5.4, further potential optimizations are identified. Finally, we discuss the newly added library “PETSc4FOAM” in OpenFoam in order to make use of external sparse linear solvers such as PETSc/Hypre.