

Monitoring mesoscale convection simulations with nekRS using JuMonC at scale

Christian Witzler & Mathis Bode

Jülich Supercomputing Centre(JSC), Forschungszentrum Jülich GmbH, Germany, [c.witzler, m.bode]@fz-juelich.de

Motivation

- Many scientific challenges (e.g., Fig. 1) need large simulations which increases the computation complexity
- Simulations at Exascale size require new workflows and data usage patterns
 - Interactive in-situ monitoring and control

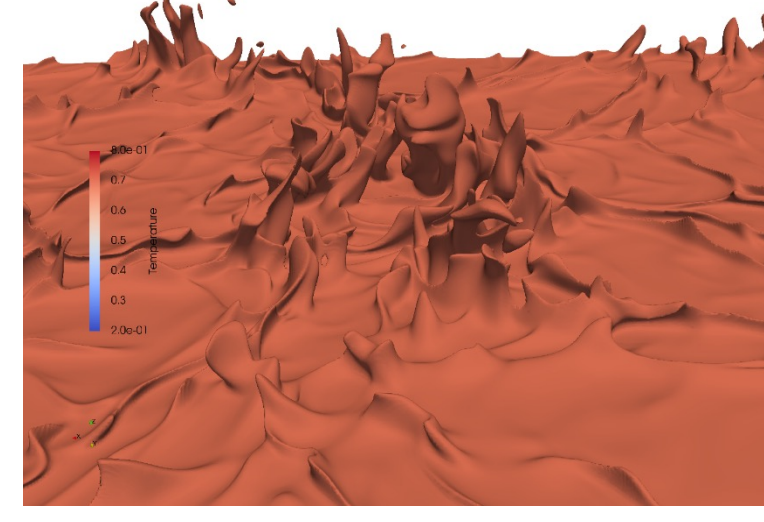


Fig. 1: temperature contour for mesoscale convection application

Towards Exascale

- The Exascale computer JUPITER hosted at JSC is expected to be in operation in 2024
- Simulation codes need to be prepared for GPU-heavy workflows with efficient data usage

JUelich MONitoring and Control (JuMonC)

JuMonC allows easy and user controlled interactive access to monitor data and enables simulation specific interactions.

- JuMonC offers a REST-API that can be accessed in multiple ways, such as with http requests (Fig.2)
 - Some requests can be answered quickly by the plugins
 - Others require time for data collection using the task queue
 - If data is needed from a different node, the request will be dispatched through JuMonC's MPI communicator
- A plugin structure is in place to allow flexible expansion with specific additional monitoring and simulation steering; an example is the nekRS plugin
- The on demand access by users is enabled through the REST-API utilizing the JSON format for data transport and secured by access tokens with an optional SSL encrypted connection
- A data base for data permanency is implemented
- Scheduling tasks is possible to create a monitoring history

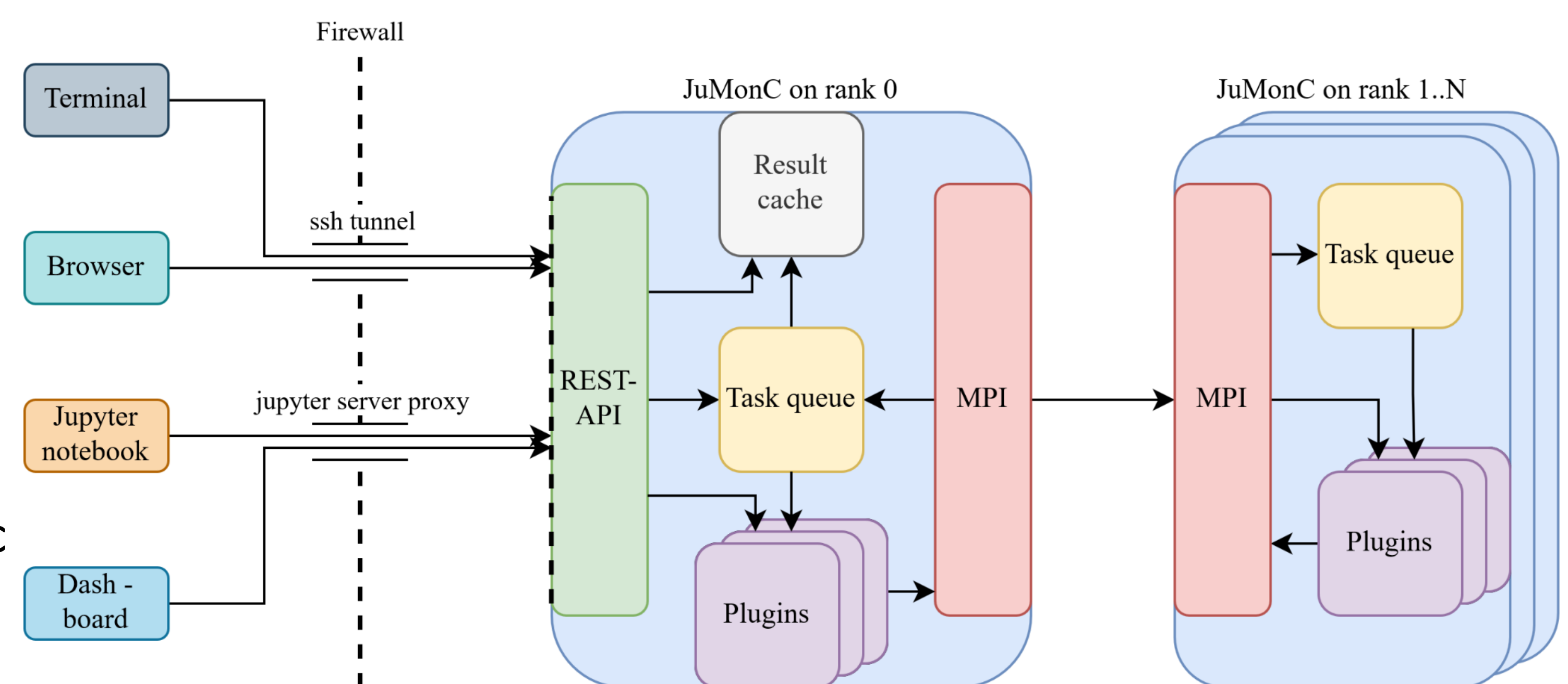


Fig. 2: Communication overview for JuMonC: access from the left through the REST-API with internal JuMonC data flow

Installation
using pip



OpenAPI
documentation



Results: Monitoring and Physics

The results are obtained by using nekRS, a high-order spectral code for GPUs on the example of mesoscale convection at high Rayleigh numbers (Ra) and low Prandtl numbers (Pr) at scale (Fig. 6).

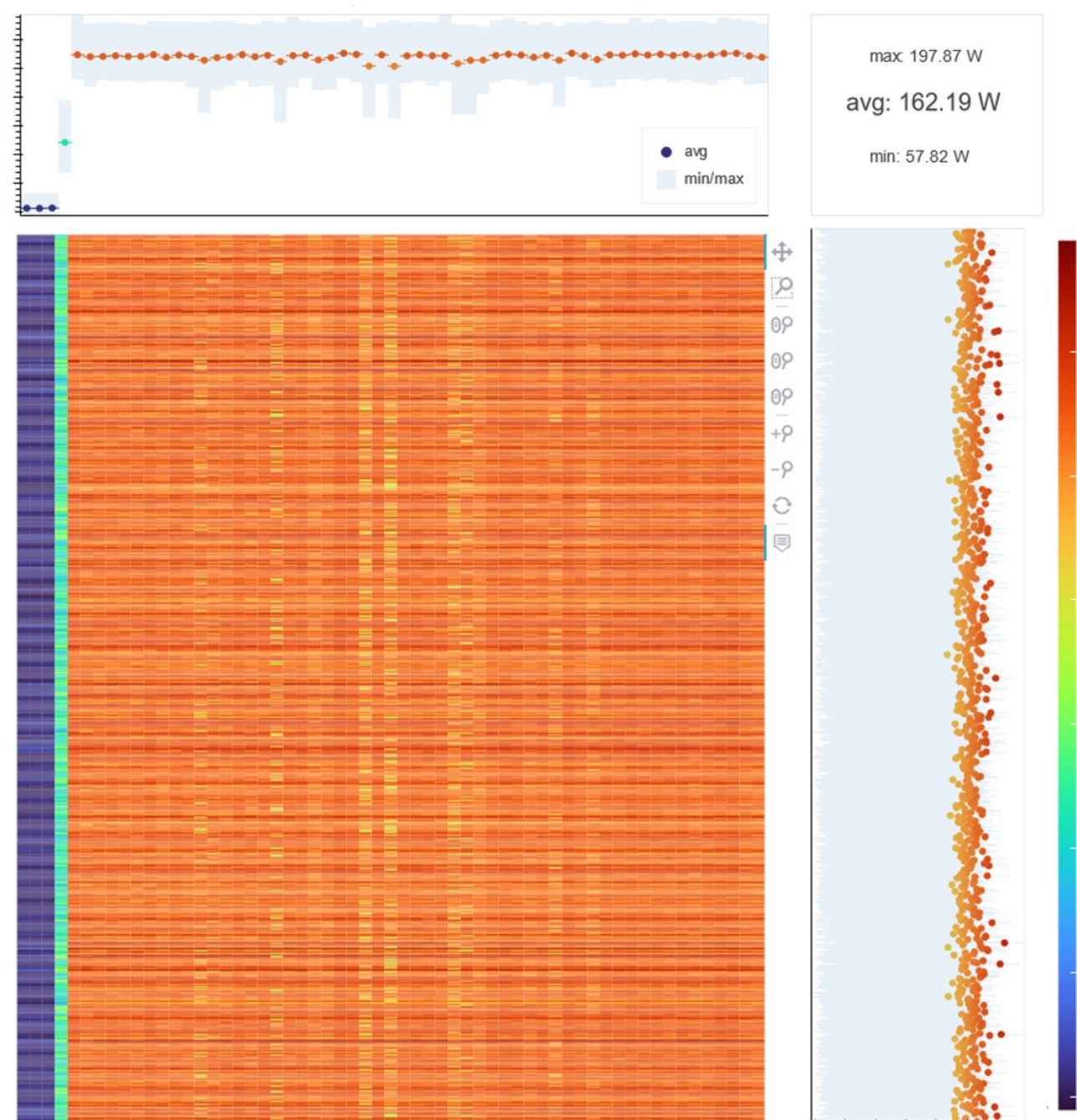


Fig. 3: Example information for each node for different times

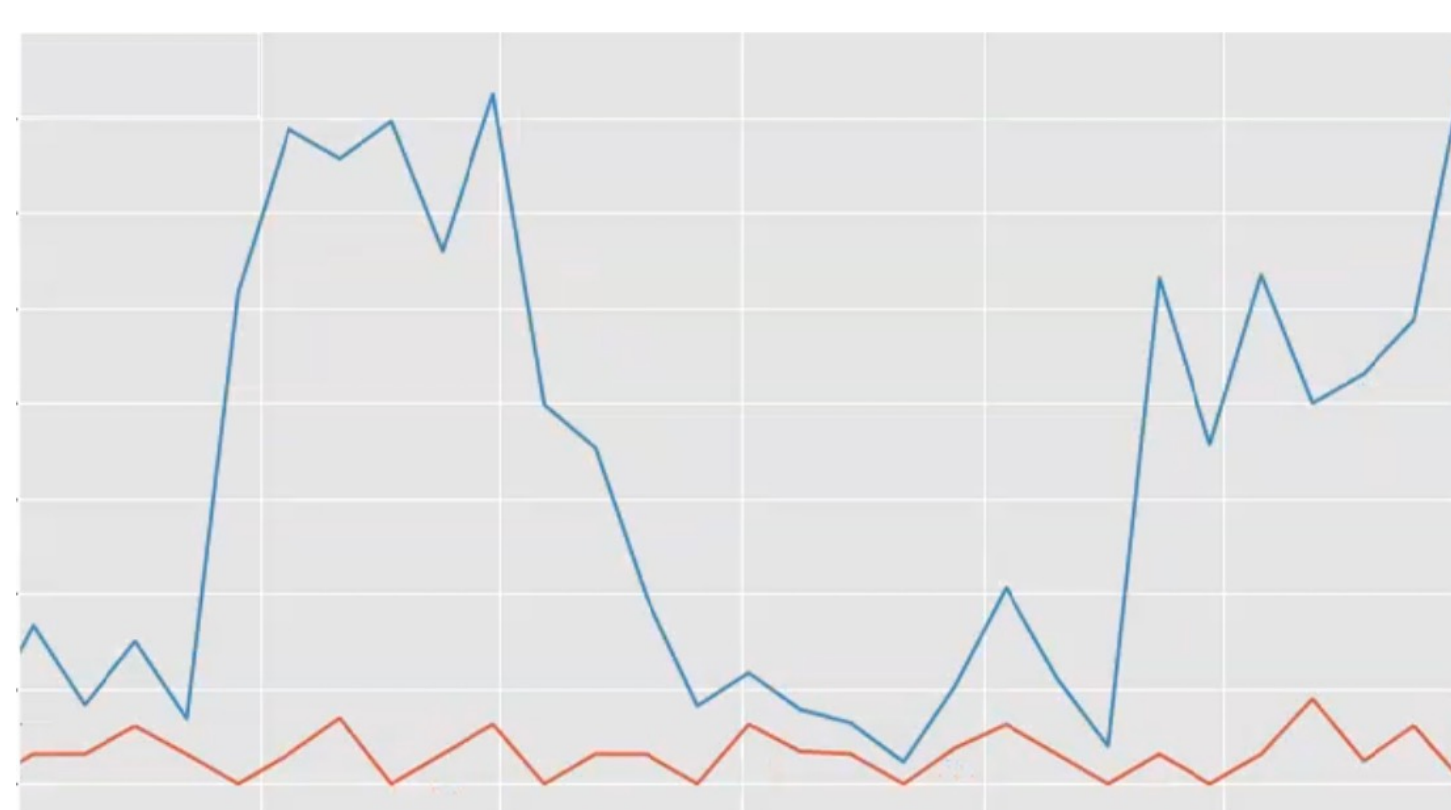


Fig. 4: Example dashboard using JuMonC's data for a self-updating network usage graph

- In-situ monitoring of simulations helps with the failure identification and prevention (Fig. 3 and 4)
- JuMonC is setup to have minimal impact on the simulation performance

JuMonC code and
documentation (incl.
example notebooks)



- Results for $Pr=0.7$ and $Ra=10^7$ (Fig. 5)
- Reynolds number of 867 and Nusselt number of 15.49 (Fig. 7) are within errors with results on CPUs [1]
- nekRS takes 0.3s per timestep (significantly smaller time-to-solution than CPU codes)

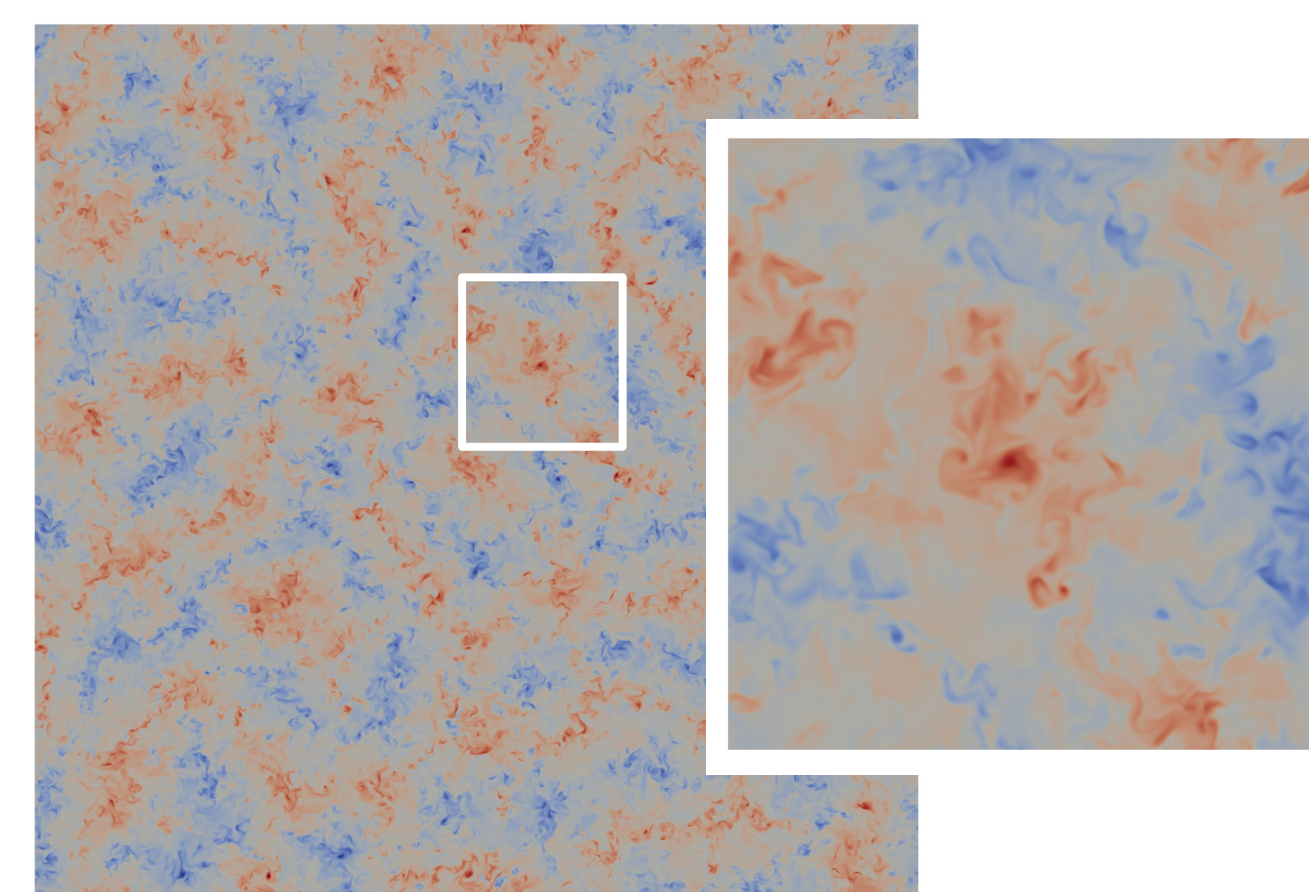


Fig. 5: Temperature for $z=0.5$ slice with zoom to a superstructure

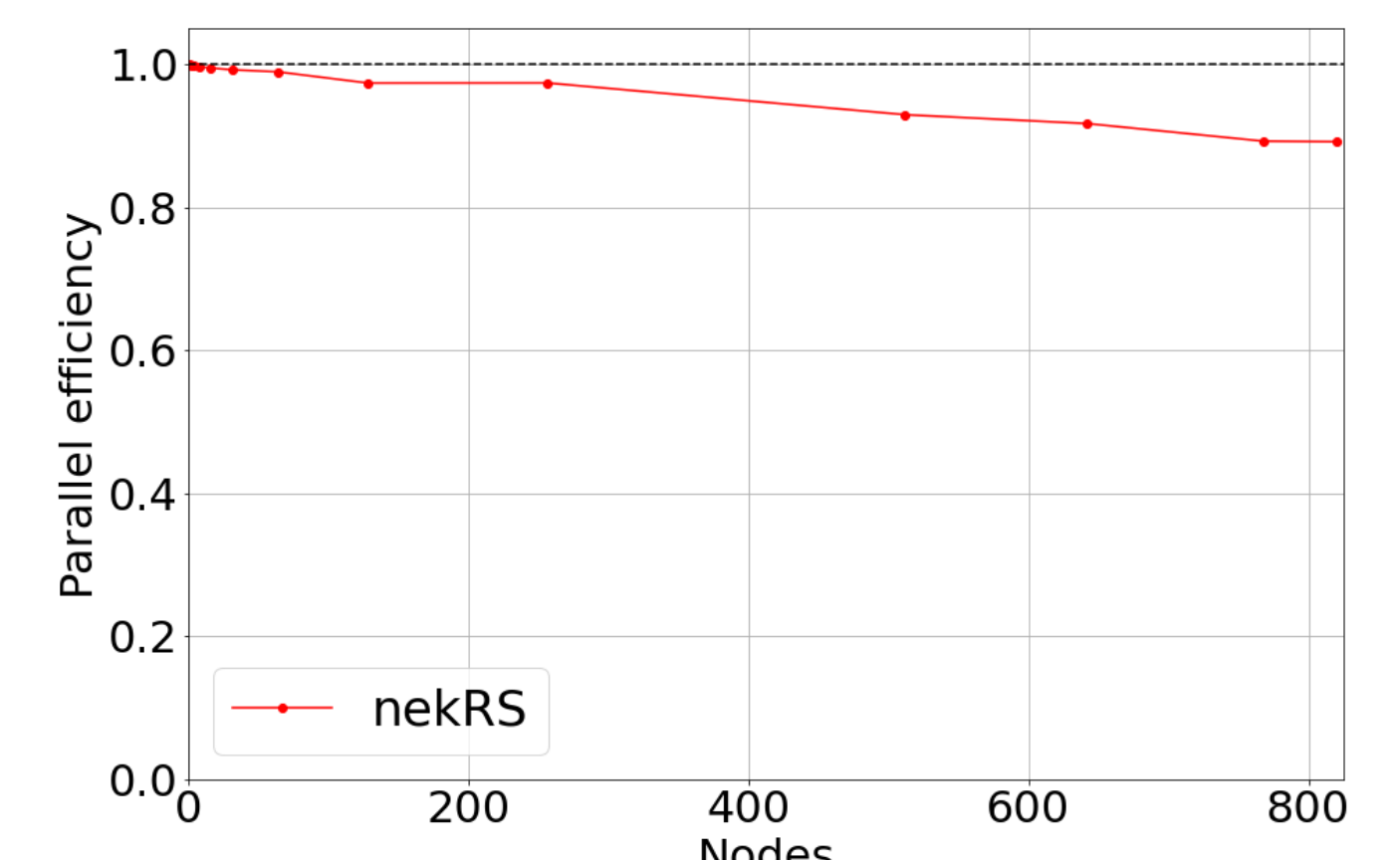


Fig. 6: Weak scaling for nekRS on JUWELS BOOSTER

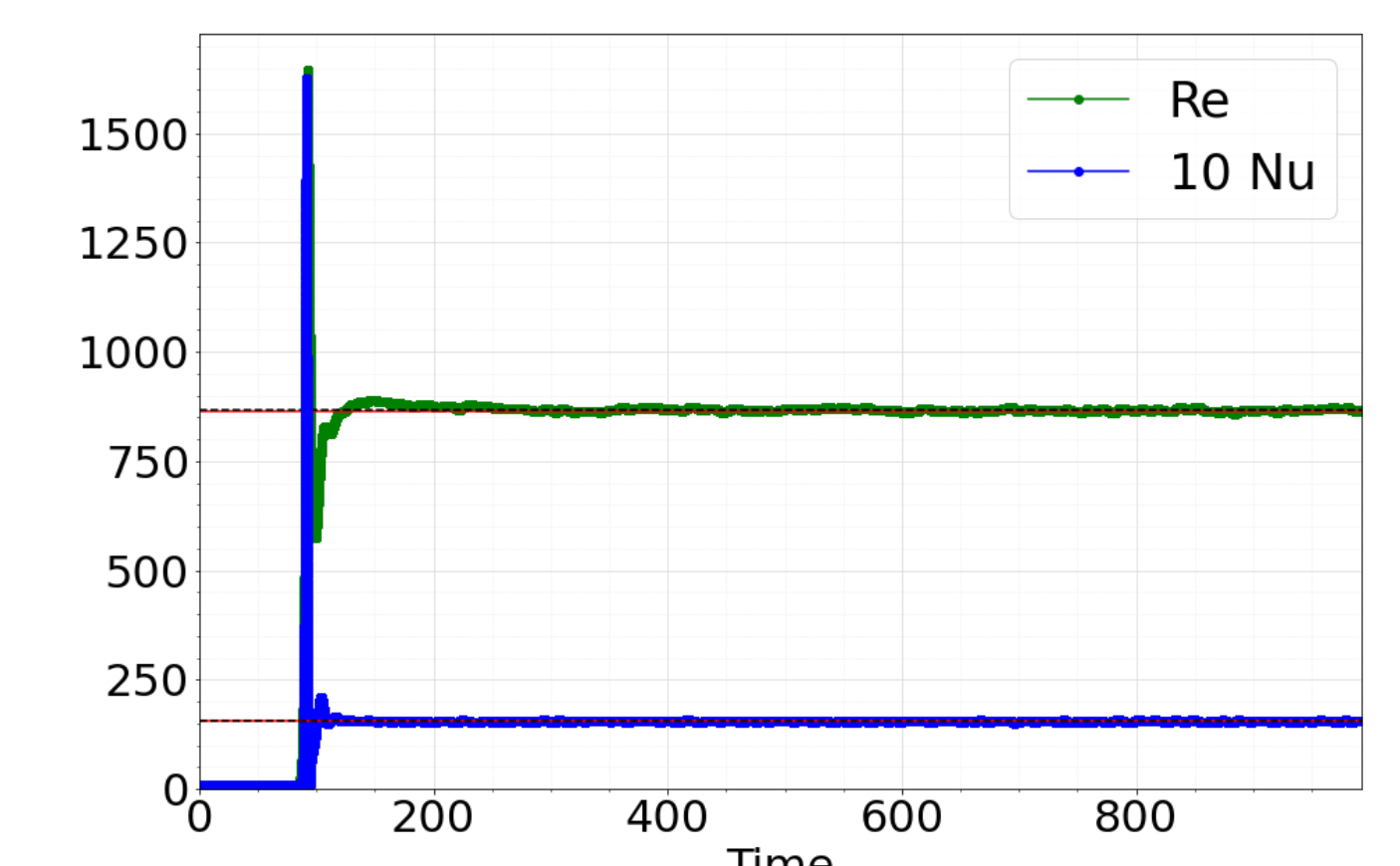


Fig. 7: Reynolds and Nusselt number during the simulation, average as dashed line and red line from [1]

Conclusions

- JuMonC supports Exascale-ready workflows and data usage at scale
- JuMonC does not measurably impact performance of nekRS
- Plugins allow to interact with simulations using a common REST-API interface