

# Performance Profiling of the Atmospheric Model CESM CAMchem

Tracy Kiszler, Theresa.Kiszler@lass-potsdam.de

## What is this CESM?

The Community Earth System Model [1] is a fully-coupled, global climate model that provides computer simulations of the Earth's past, present, and future climate states. It includes several components as shown in Fig. 1. It is possible to add chemical mechanisms to the atmosphere component (CAMchem). In the project group "air quality" this is used to conduct research on how specific chemical species (especially ozone) are transported in the atmosphere. The research on this topic has been extended to developing a "tagged" mechanism which is used for source attribution [2]. This helps to develop policies and strategies combating atmospheric pollution by anthropogenic emissions.

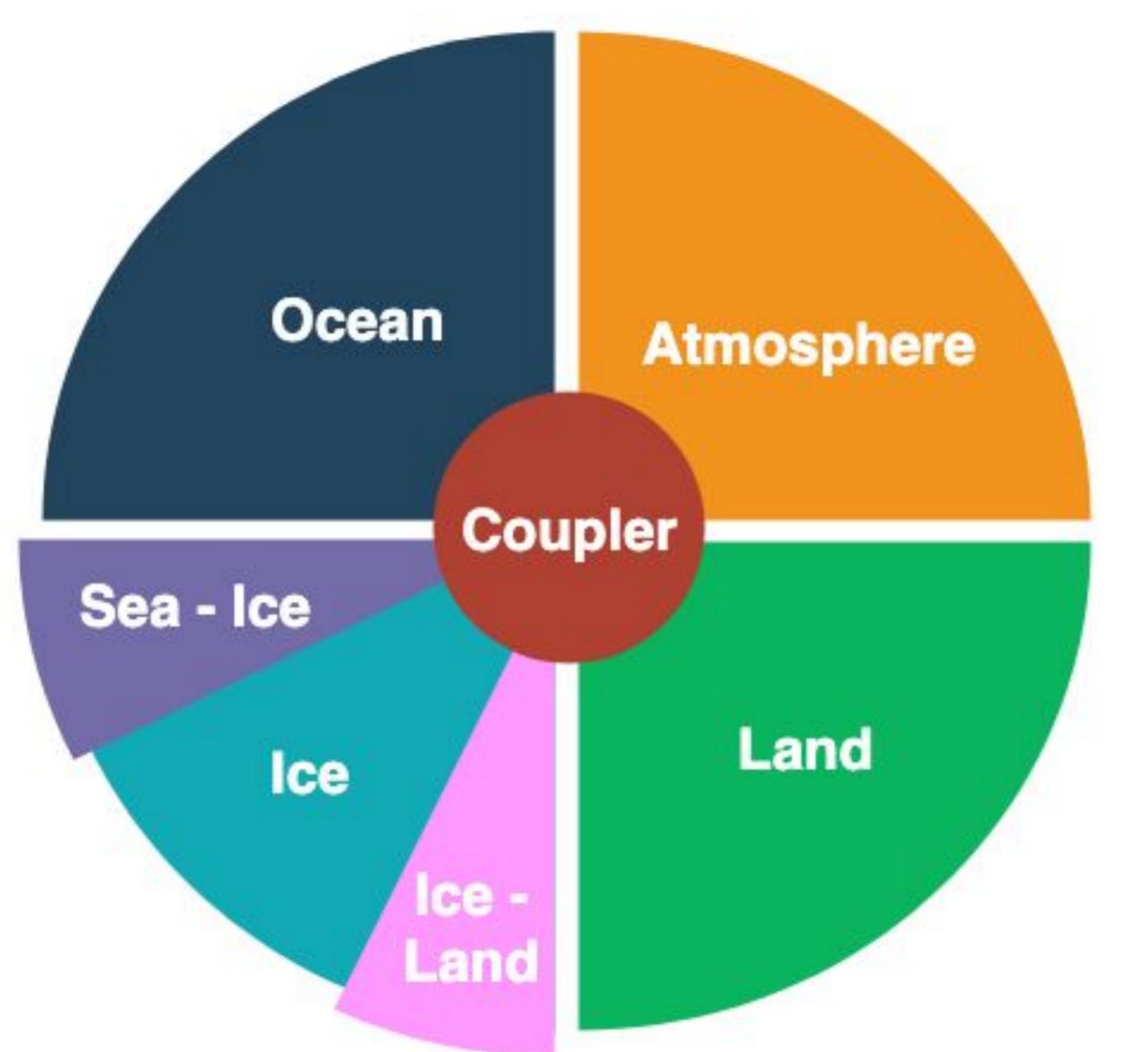


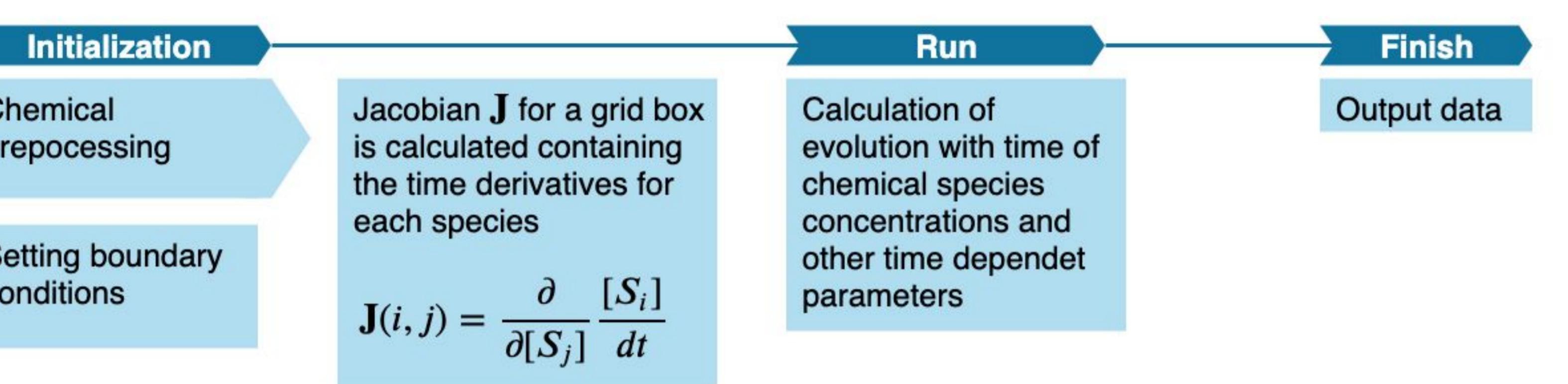
Figure 1: Components of the CESM 1.2.2. The coupler acts as connection between the different components.

### An example of a tagged species:

O<sub>3</sub> traffic would be ozone which was either emitted by traffic or is a secondary product of another chemical species which was emitted by traffic.

## Computational Aspects

There are some basic steps that CESM CAMchem goes through to compute the results for a given setup.



$$J(i, j) = \frac{\partial}{\partial[S_j]} \frac{[S_i]}{dt}$$

## What is the Problem?

Introducing the expanded chemistry mechanism leads to an increase of chemical species which again increases the size of the Jacobian matrix. Due to the nature of the derivations the majority of additional entries are zero, wherefore the computational complexity was only expected to increase slightly. It was noticed though, that the **model run time increased significantly** and by that uses more resources.

To evaluate the running time further, different setups on the IASS cluster were run. All setups use the same chemical mechanism but the "tagged" runs have additional species which use the tags. The runs were conducted on the IASS cluster (see Figure 2) and the simulated time is 5 and 10 days. Each combination of 5/10 days and tagged/not tagged was run on 1 to 8 nodes (28-224 cores).

## Technical Details

- OS is Linux (CentOS)
- 9 Nodes
- 28 Cores per node
- Language of CESM 1.2.2 FORTRAN
- Compiler GNU
- Parallelized using MPI and openMP
- Job scheduler Slurm
- Networking Communication standard InfiniBand Interconnect
- Filesystem based on GlusterFS
- Processors on Compute nodes: Intel Xeon E3-2695v3 14C/28T 2.30 GHz

## Run Setup

Chemical mechanism: MOZART  
Number of chemical species in tagged mechanism: 299  
...in normal mechanism: 103  
Tagged species: NOx  
Grid: 1.9° x 2.5°, Greenland pole  
Simulation time: 5 and 10 days  
Cores used: 28, 56, 84, 112, 140, 168, 196, 224

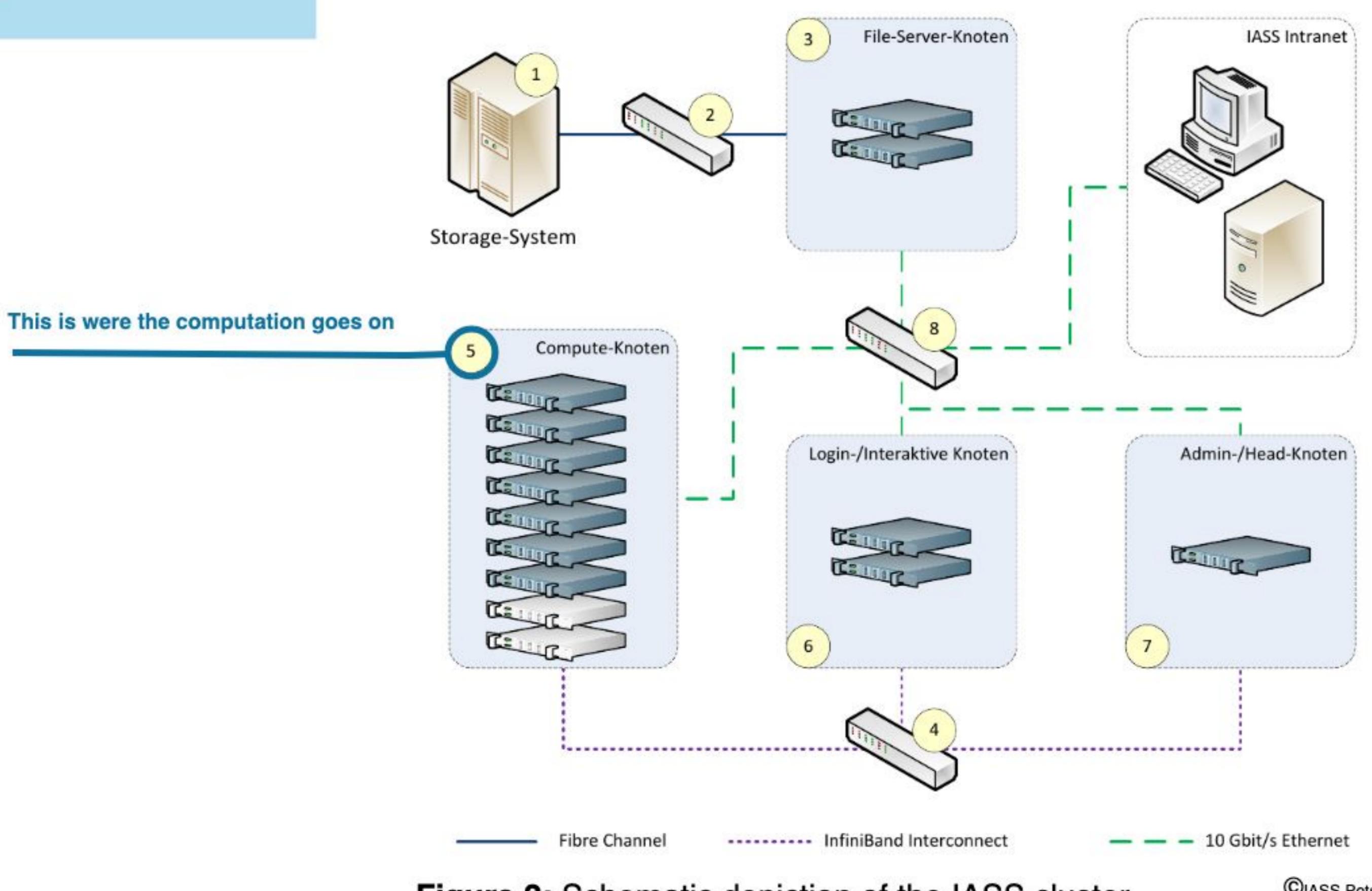


Figure 2: Schematic depiction of the IASS cluster.

## Measuring the Performance

To measure the basic performance the total duration of the different runs is used (Figure 3). The total duration decreases with the increasing numbers of cores until it reaches 112 cores where an optimum was found for both tagged and non tagged runs. This was not foreseen and shows us that the load balancing on our cluster is very likely not optimal.

The total duration is made up of the initialization, run and finalize time. The initialization duration only increases slightly (<1 min) for the tagged version and the duration of the finishing step does not show any tendency at all. Therefore the issues with the run time stems solely from the computation.

It is possible that the communication overhead and inefficient parallelization lead to the stagnation of the run time although more cores are used.

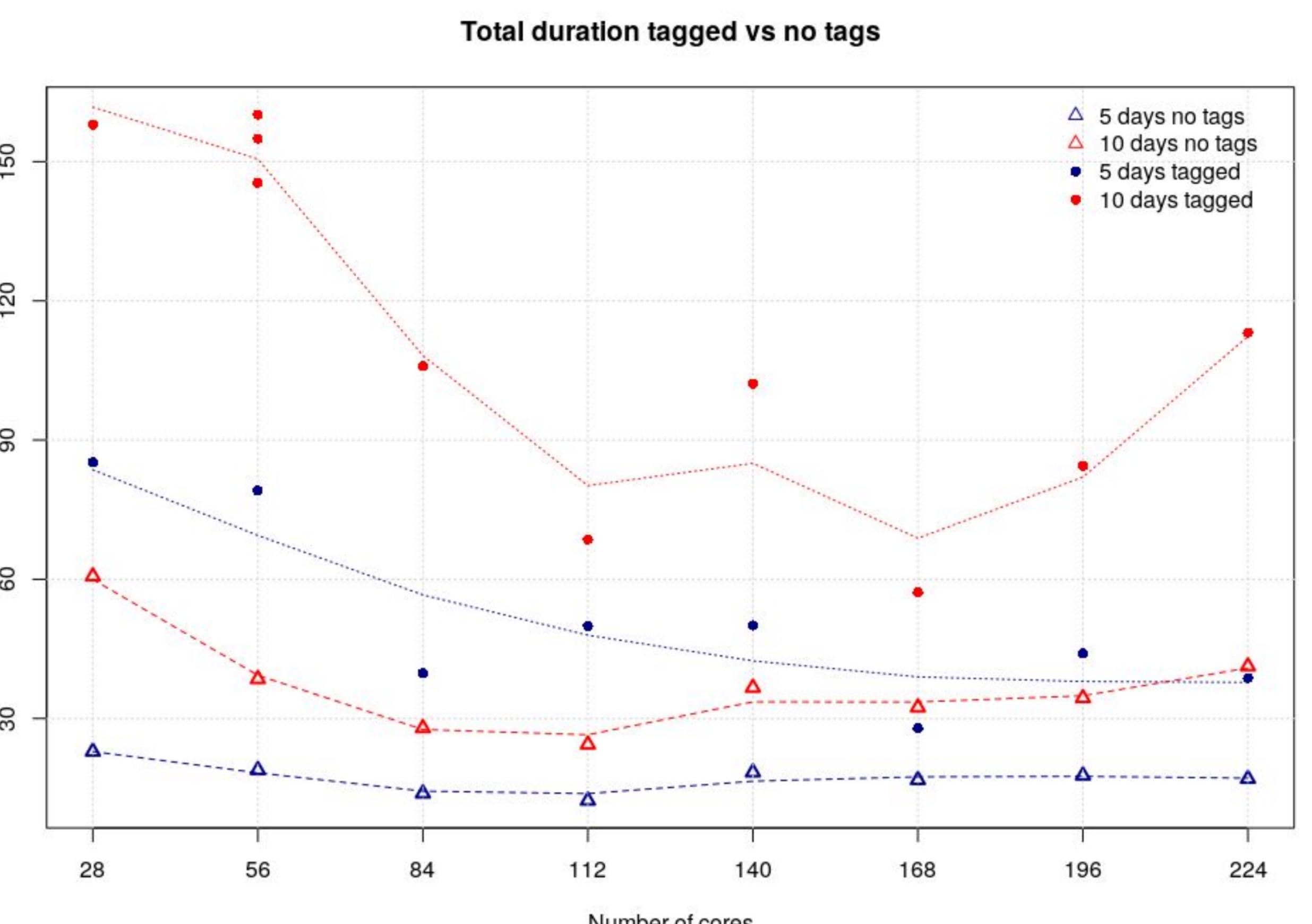


Figure 3: Total duration given in minutes for runs with and without a tagged chemical mechanism (dots and triangles) depending on the number of cores used for each simulation. Red marks the 10 days runs and blue the 5 days runs.

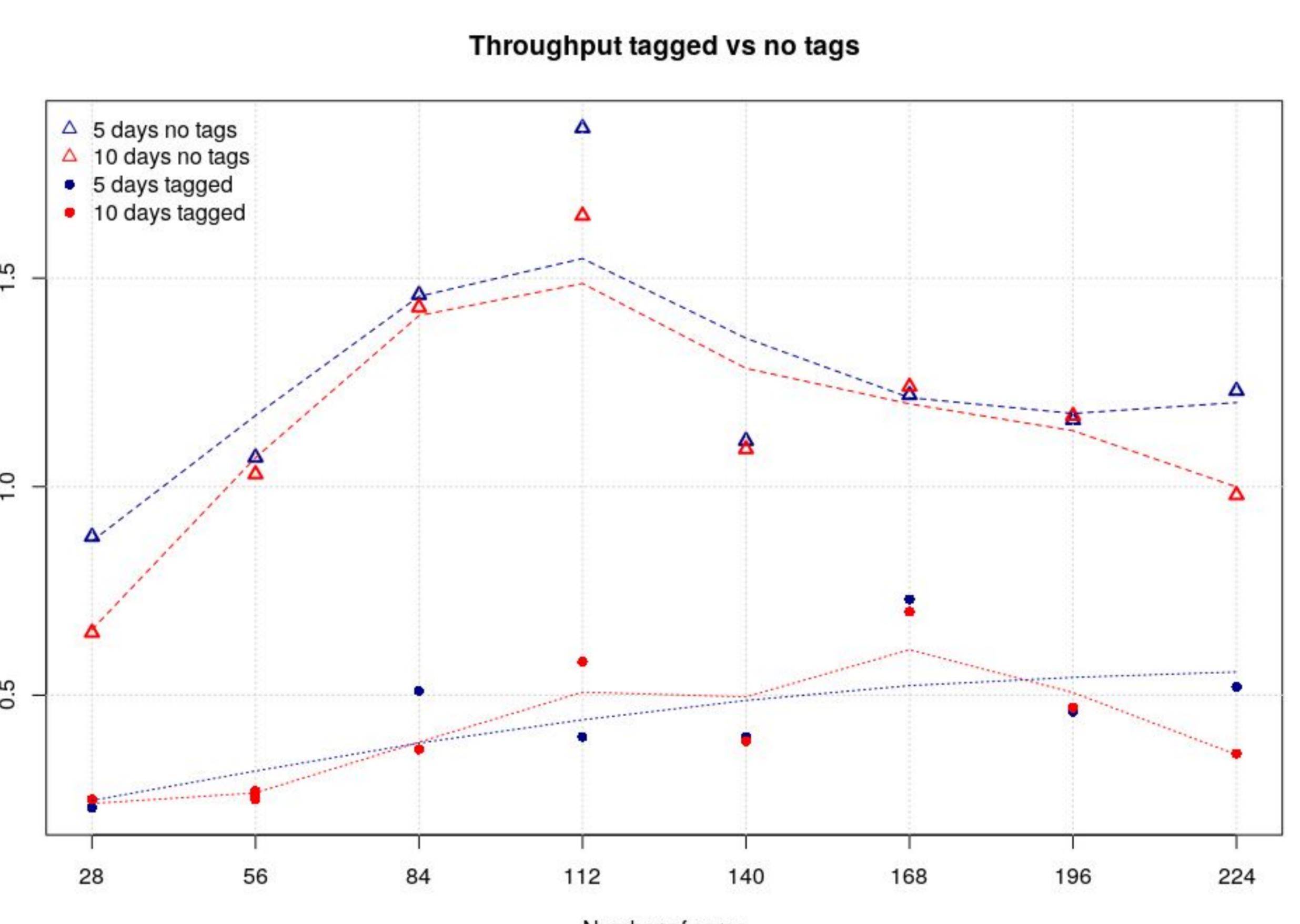


Figure 4: Throughput given as "simulated years per wall day" for runs with and without a tagged chemical mechanism (dots and triangles) depending on the number of cores used for each simulation. Red marks the 10 days runs and blue the 5 days runs.

## Where does the extra time come from?

As the CESM is a global model, it consists of several layers of functions which call each other. Therefore it was necessary to look at the lowest layer, to find the reason why the tagged mechanisms took longer. Several timers are run by default and are used here to extract the functions and their duration.

To be able to compare the non tagged and the tagged runs following parameter  $\Delta$  is used:

$$\Delta = \frac{T_{tagged} - T_{not}}{T_{not}}$$

with  $T_{tagged}$  and  $T_{not}$  the maximum wallclock time for a single timer. From this it follows that a value greater than 1 for  $\Delta$  means that the tagged mechanism took longer for that specific timer section.

Using  $\Delta$  we found some code sections which consumed far more time in the tagged setup than in the non tagged setup. In Figure 5 some of these sections are depicted. Each section has a timer (e.g. "phys\_run2") which is nested in a parent section (e.g. "phys\_run2" is run as part of "CAM\_run2"). From this it was possible to see that some sections of the dynamic and physics computations take longer in the tagged runs than in the non tagged runs.

Another interesting observation was the increase of the Land and Ocean driver time. Due to the low computation time needed for these drivers though, they can be neglected.

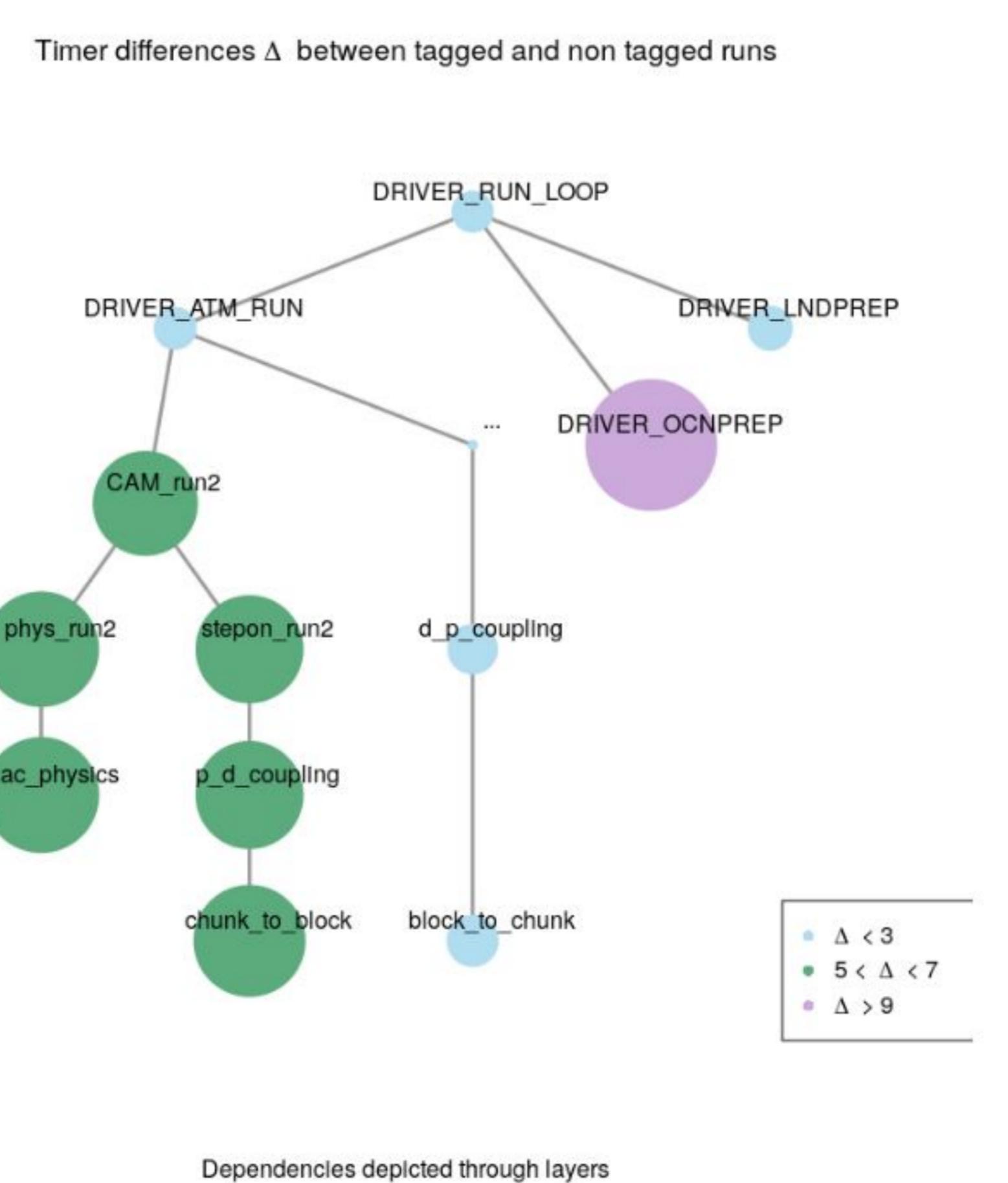


Figure 5: Several timers are shown here with the size and color depending on the  $\Delta$  value calculated for each timer. The layers signify the dependencies and the colors show by how much the wallclock time has increased. The timer names are written in the vertices.

## Increase of Cost

Another metric used in the CESM 1.2.2 model timing is the **cost** (see fig. 6). It is given as pe-hrs (processor element-hours) per simulated year. The cost of the runs gives additional insight on the efficiency and is calculated with following formula [3]:

$$\text{wall-clock hours} \times \text{nodes used} \times \text{cores per node} \times \text{queue factor}$$

For the CESM it is known that it does not scale linearly and that the cost increases with the number of processors used [4]. It is expected though, that adding processors should increase the throughput while the cost increases. This is not the case for our setup and shows that there is potential to optimize the configuration of the CESM runs on the IASS cluster.

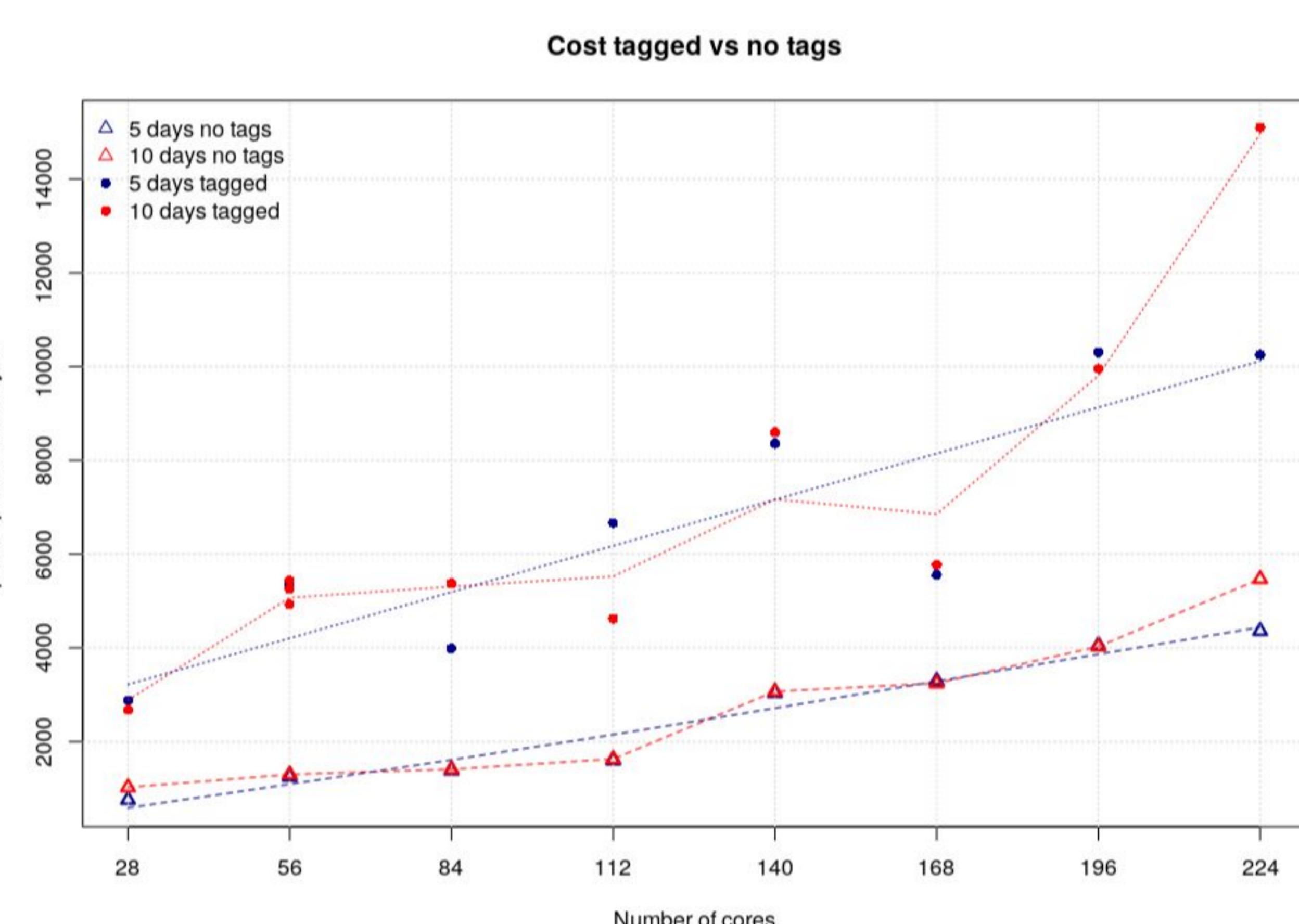


Figure 6: Cost for CESM runs with and without tags (dots and triangles) for 10 and 5 days simulation time (red and blue). The dashed lines represent a smoothed spline function for each set of values.

## Wrapping it up

The given problem was the question why the tagged CESM CAMchem model runs took considerably longer than the non tagged runs. During the exploration of possible answers another issue was discovered, which was the optimum for the CESM runs when running on 4 nodes.

By evaluating several metrics including the throughput, simulation duration and cost it was possible to gain some insight on possible inefficiencies in the configuration of the CESM CAMchem on the IASS cluster. This does not only concern the tagged runs but also the normal runs as they too have an optimum around 4 cores.

The objective of finding the reason why the tagged runs take longer has not been fully answered but there are some functions that contribute more to this than the rest. These include physical and dynamical calculations as well as conversion between outputs and inputs.

## Ideas for further steps

1. Communication processes must be evaluated
2. Different run setups can be tested
3. Optimizing Parallelization
4. Further profiling by instrumentalizing the code and looking at specific results

## How can we use these results?

1. We can dig into the details and try to fully understand why the tagged mechanism is slower than the standard one
2. We can work on improving the efficiency of CESM runs on the IASS cluster
3. We can avoid using unnecessary resources

Increase sustainability

## Sources:

- [1] CESM website: <http://www.cesm.ucar.edu/>
- [2] Butler, Tim & Lupascu, Aura & Coates, Jane & Zhu, Shuai. (2018). TOAST 1.0: Tropospheric ozone attribution of sources with tagging for CESM 1.2.2. Geoscientific Model Development. 11. 2825-2840. 10.5194/gmd-11-2825-2018.
- [3] Computation & Information Systems Lab. (2019). Managing Allocations and Charges, NCAR/UCAR, <https://www2.cisl.ucar.edu/user-support/managing-allocations-and-charges#core-hours>
- [4] CESM Software Engineering Group, NCAR. CESM User's Guide (CESM 1.2 Release Series User's Guide). p. 42-45