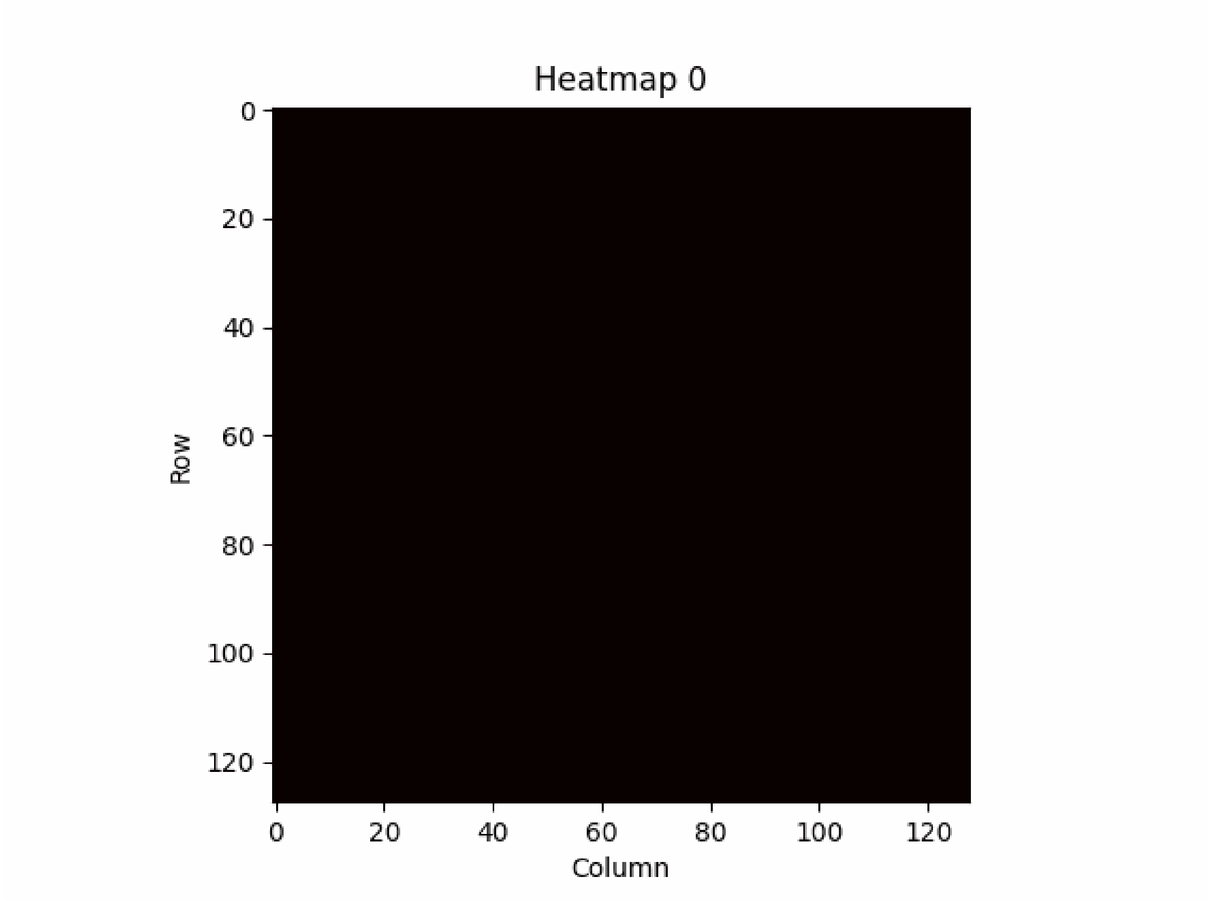
HPC03ex04 - Florian Schrittwieser, Davit Melkonyan, Simon Pavicic

4.1:

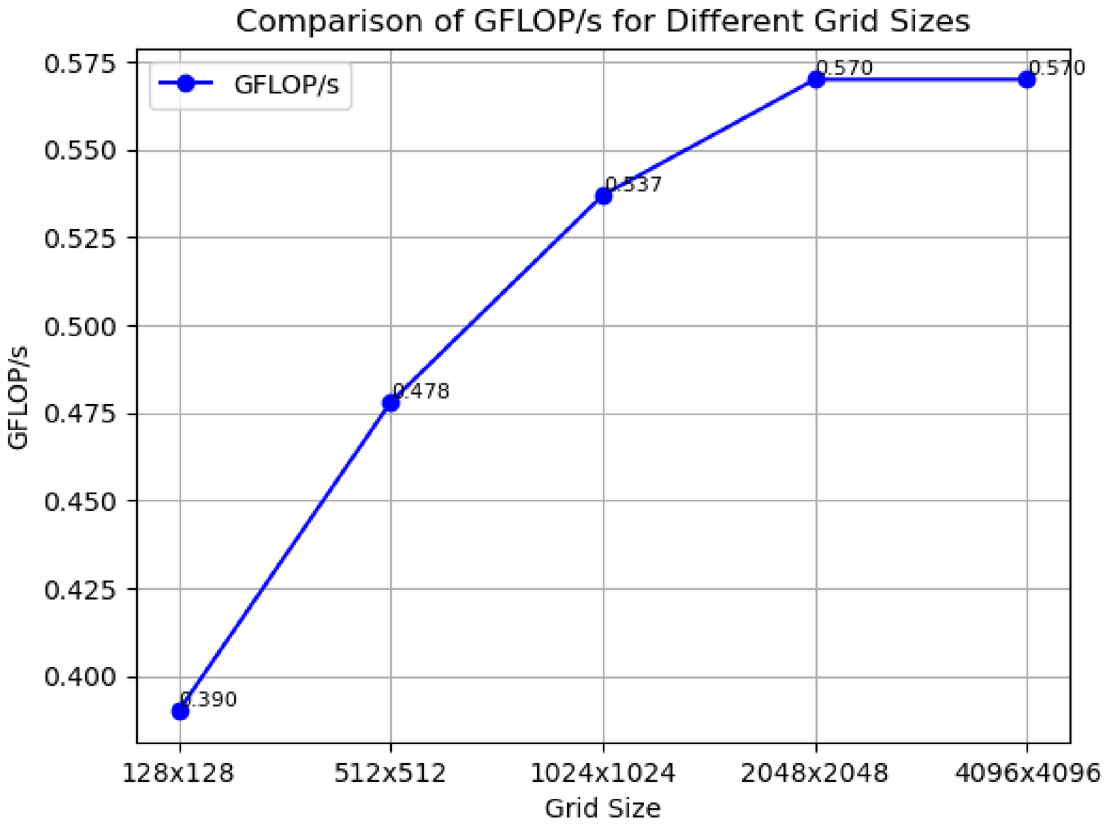
The paper "Score-P – A Joint Performance Measurement Run-Time Infrastructure for Periscope, Scalasca, TAU, and Vampir" provides an overview of the Score-P Suite, a performance measurement toolkit for high-performance computing. The authors emphasize the redundancies that arise when using multiple specialized tools instead of a common infrastructure, especially when it comes to data acquisition and data collection. The proposed Score-P framework aims to address these redundancies and provide a simple installation. The framework bundles many different libraries and tools for performance measurement and tools such as OpenMP or MPI to automatically link and utilize them. It also compares performance against previous and constituent frameworks.
Avoiding code duplication through a common framework seems to be a very good idea, as long as the framework is flexible enough to incorporate new specific features. We accept the premise of the of the paper that a standardized data format and instrumentation will result in less redundancy and better interaction. As always with the introduction of new standards, there is a risk of creating yet another framework.

4.2:



4.3:

Grid size	Time/iteration	Flops total	GFLOP/s
128x128	0.000270	11468800	0.425
512x512	0.003814	183500800	0.481
1024x1024	0.013497	734003200	0.544
2048x2048	0.051446	2936012800	0.571
4096x4096	0.206253	11744051200	0.569



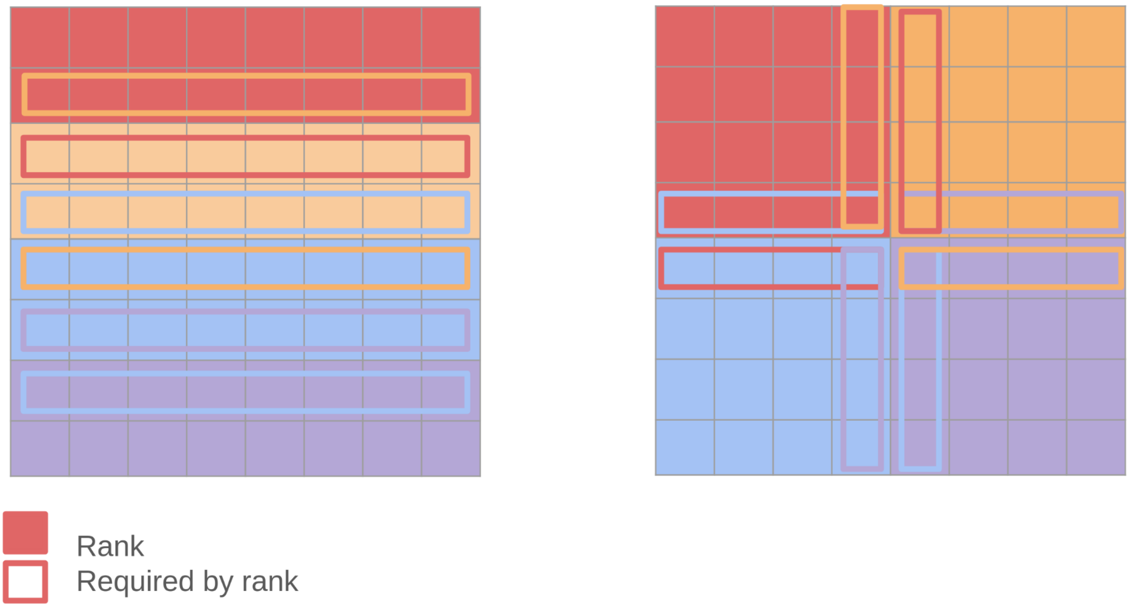
For grid sizes that are within the cache limits, the task is bounded by the computational resources of the system, since the grid sizes below 2048x2048 did indeed fit in the cache (1024^2*8/1024^2 = 8MB), no stagnation has been measured for those grids. The Grid of size 4096x4096 translates to 128MB (16 times larger), hence the GFLOP/s didn’t improve due to the various Cache misses. Therefore, the problem becomes memory-bounded for 2048x2048 and 4096x4096 grids.

4.4:

In order to parallelize this application using message passing the sequential function execution approach needs to be adopted.

1. The problem can be decomposed by splitting the task into subtasks. This means splitting the grid into different parts so that calculations using multiple ranks are possible. However, this requires the ability of the ranks to communicate with each other.
2. A 1D or 2D approach can be used for partitioning. Either we split the grid into 4 equal rectangles with one dimension like N/4 x N or N x N/4 (1D approach) or into 4 equal rectangles of the same aspect ratio as the original N/2 x N/2 (2D approach). The better partitioning for a given problem is the one that requires fewer messages to be

sent between ranks overall (volume-surface rule). Let's look at the two options 1D and 2D. The following figure shows the two approaches. All completely filled grid points are calculated with a corresponding rank. The unfilled outlines represent the points that the same-colored rank requires for its next iteration and that are calculated by a neighbor. Once these areas have been calculated, they are exchanged with the corresponding neighbor.



Using this example of 1D and 2D partitioning with **4 ranks (formulas only work for 4)** as before, we would need the following number of points for each method:

1D: #exchanged_points = 2 * N * #partitions - 2 * N = 2 * N * (#partitions - 1)
2D: #exchanged_points = N/2 * 2 * #partitions = N * #partitions

This means, for our use case (4 ranks) the 2D partitioning approach will have less points to exchange.

This means

3. We can leverage overlap between computation and communication as follows: Let's define Volume as points that are not required by any other rank than the one they are assigned to and surface as the points, that need to be send to other ranks for calculations. To minimize delays due to sending and calculating, we would suggest shifted calculations and sending. This would mean, that while one rank 1 calculates the surface first, the other neighbour rank 2 is calculating it's volume points. After both are finished, rank 1 can exchange the surface points and rank 2 can use them to calculate the missing points.

4.5: willingness to present

4.1 -> YES
4.2 -> YES
4.3 -> YES
4.4 -> YES