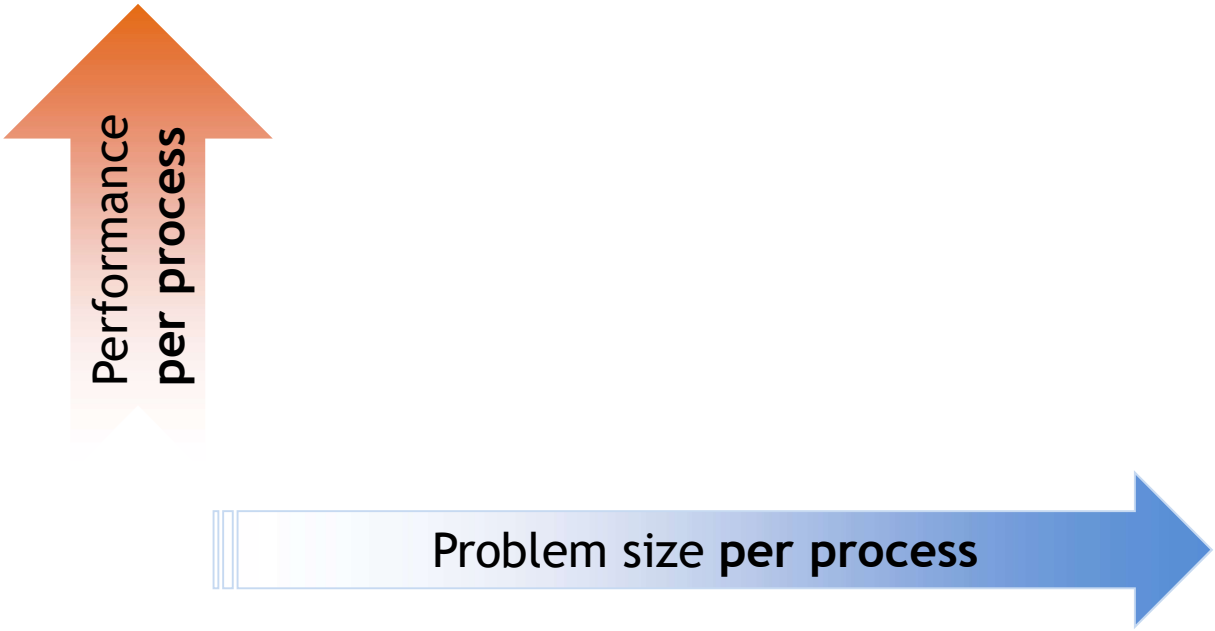
Performance

How to measure and compare using a performance map

What is the Performance Map?

- Characterization of the application performance
- A performance measure per processing unit is plotted over the problem size per processing unit
- This is done for several data series, where each data series represents the run with a fixed count of processing units with varying problem sizes

Axes of the Performance Map



Why Choosing this Plot for Characterization?

- We are interested in the serial performance of the application as basis of everything else, especially a high sustained performance
- Serial performance is usually observed to depend on the problem size (due to caches, overheads, bank conflicts, vectorization, ...)
- Thus, baseline of all performance is the serial execution, which is characterized by a plot of performance over problem size

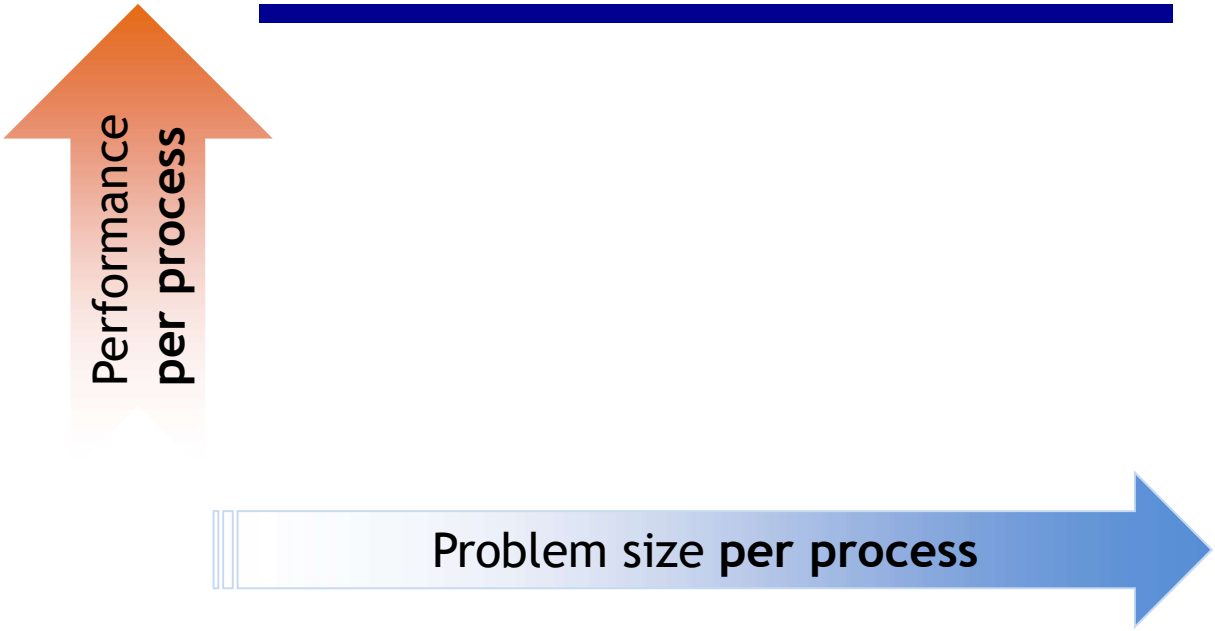
Why Choosing this Plot for Characterization?

- With parallel execution, we can at most get as good as serial on each processing unit, thus ideally we are replicating the serial performance on all processing units.
- Thus, if we plot runs at different processing unit counts with a performance per processor, we immediately see, how far we deviate from that.

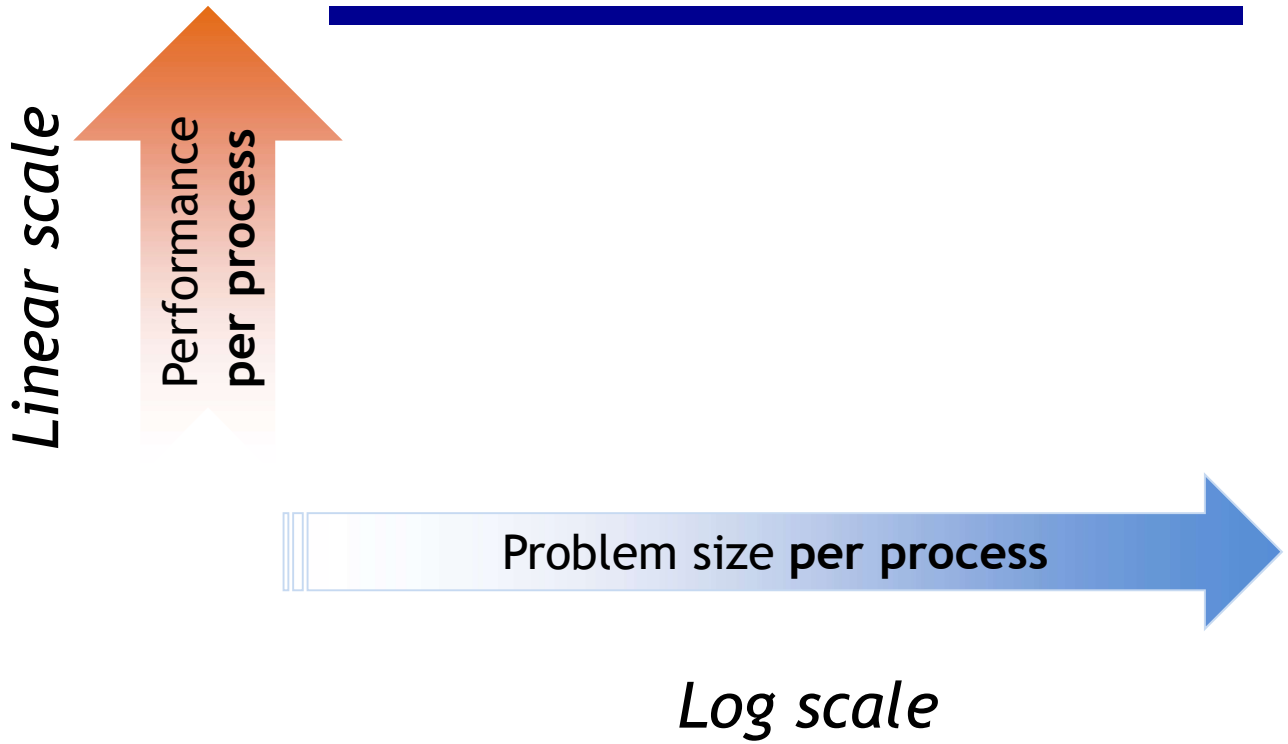
Why Choosing this Plot for Characterization?

- Furthermore, by using a processing unit specific viewpoint there arise no problems by large scale runs, no matter how high the degree of parallelism is. It will always fit into the single plot and be comparable to runs with different processing unit counts.
- This viewpoint emphasizes, that it is the performance of the processing unit that matters, it does not neglect serial performance

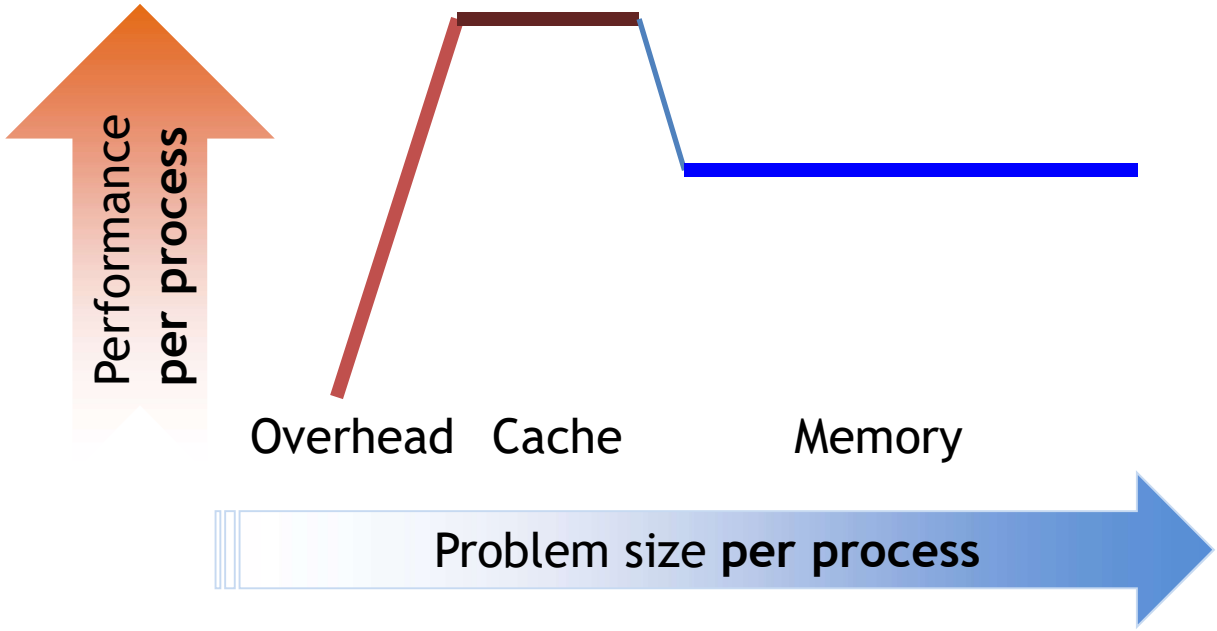
Ideal Performance



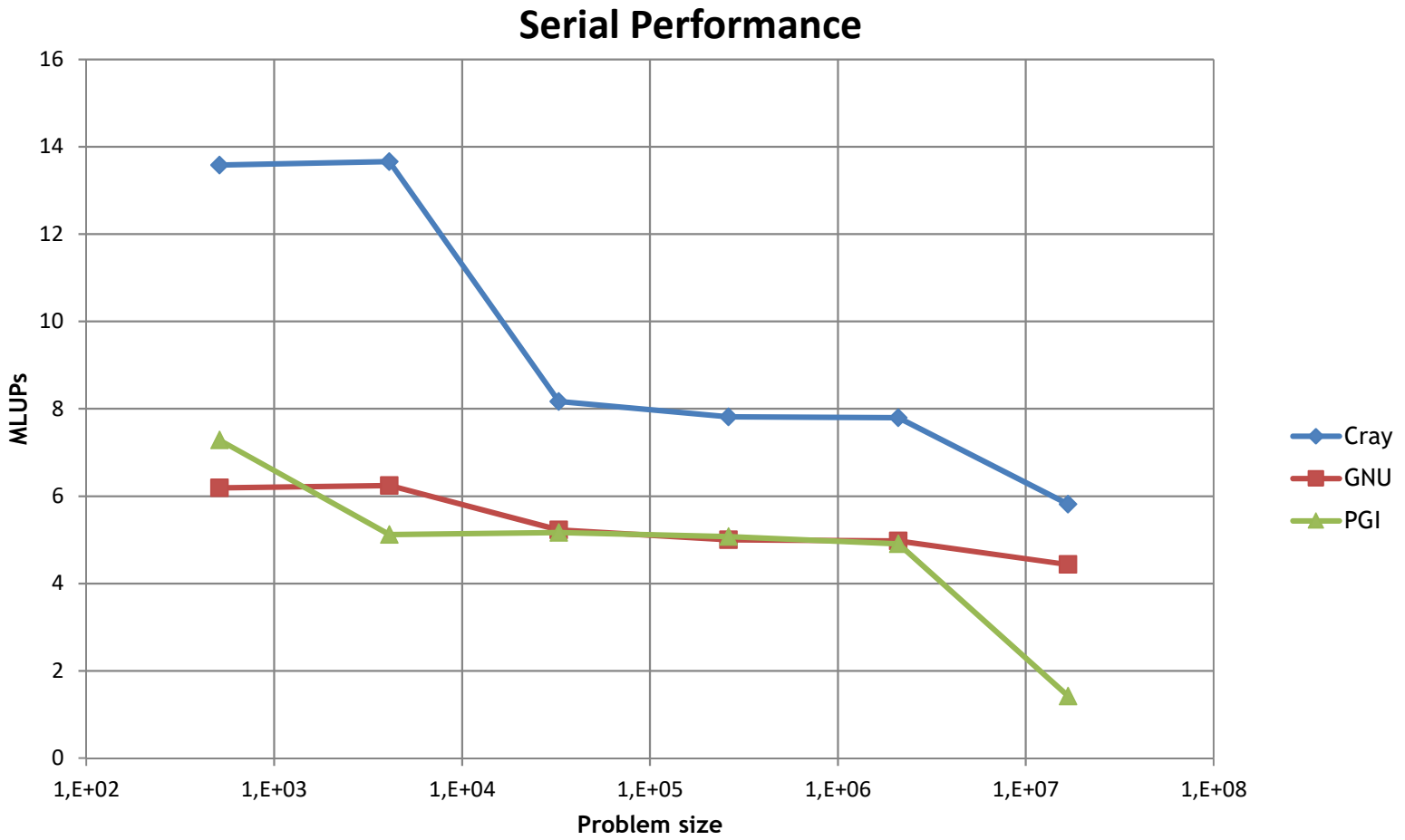
Ideal Performance



Expected Serial Behavior



Actual Performance of Musubi on Hermit

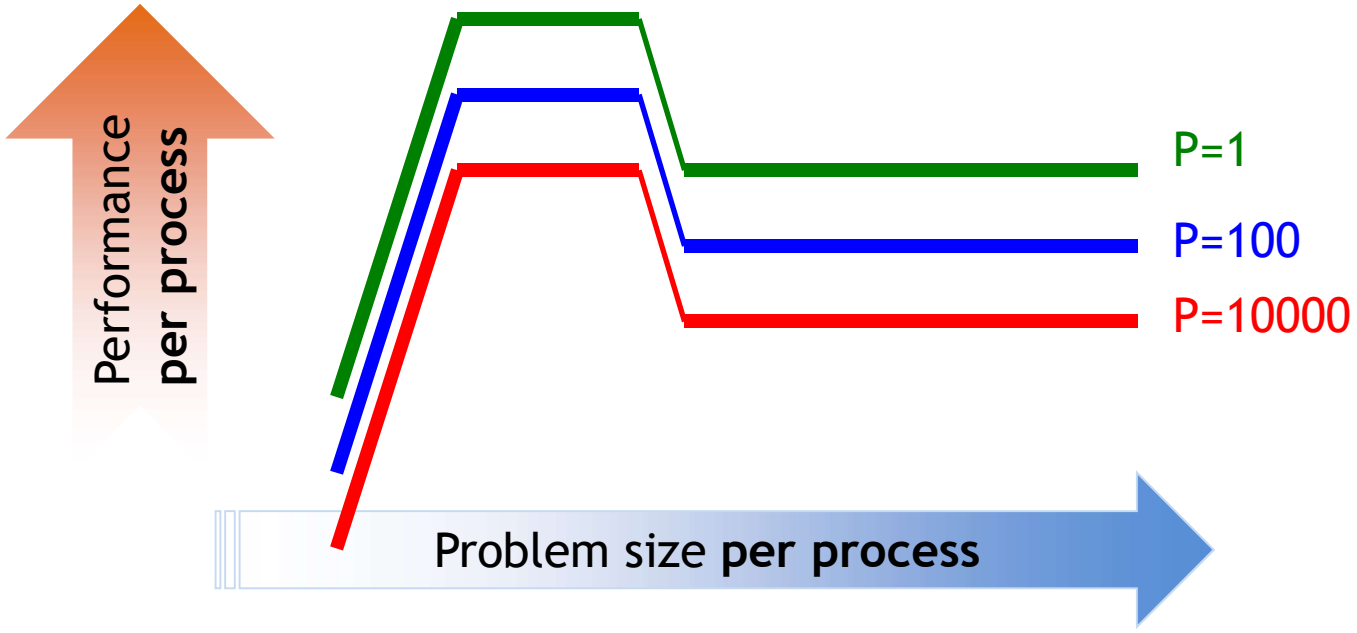


What to Expect from Parallel Execution?

- Parallelism introduces several overheads (communication, process administration, possible O(p) algorithms)

- The expectation therefore is, that the performance for larger number processes decreases

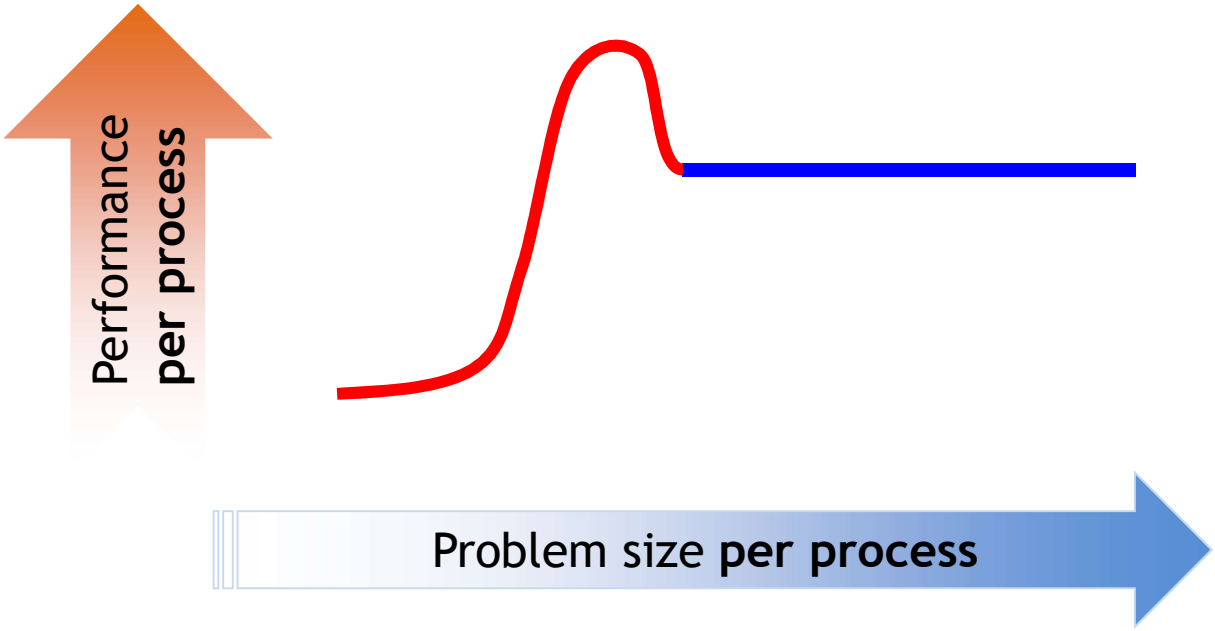
Decreased Performance Due to Parallelism



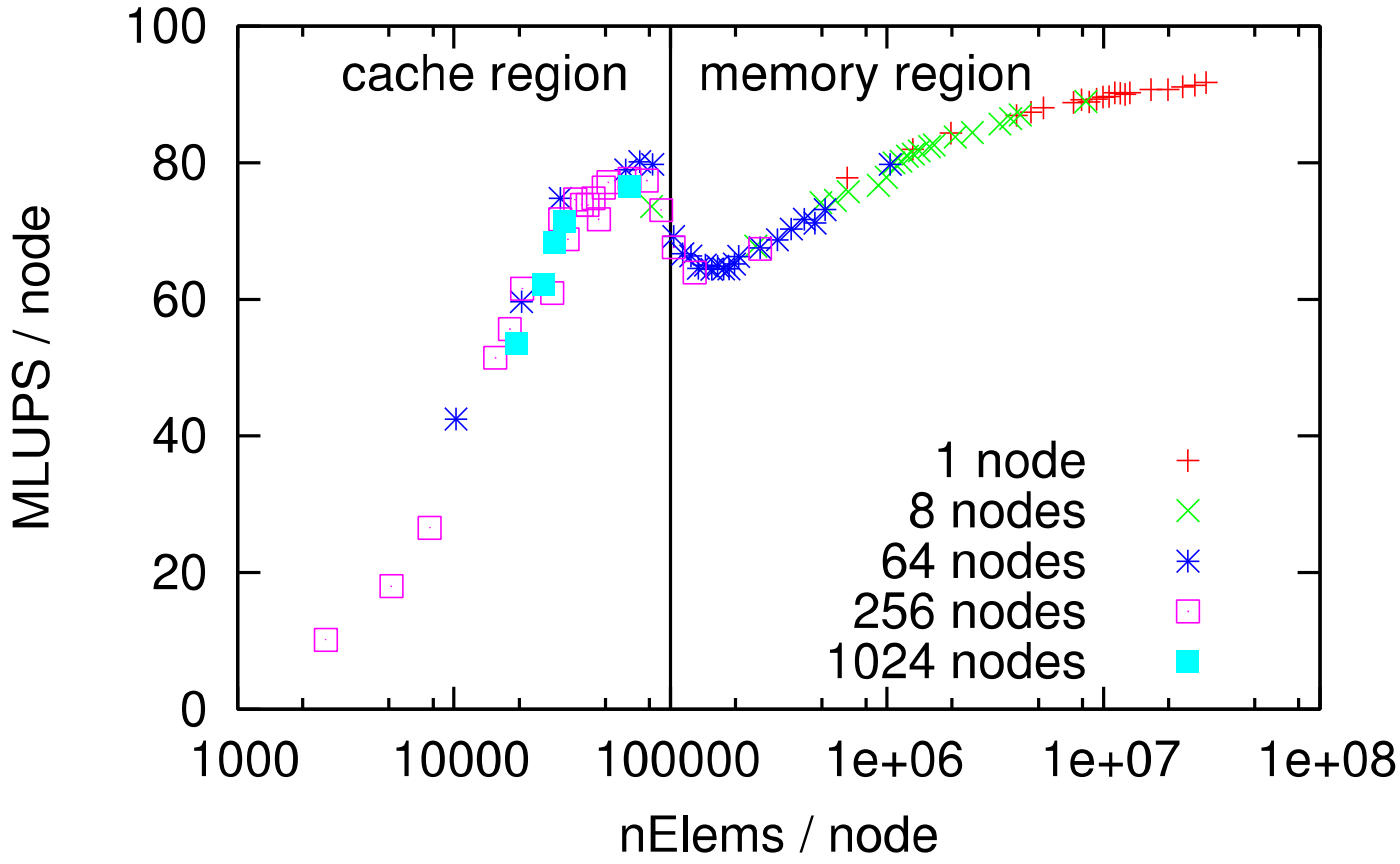
Communication to Computation in Meshes

- For Mesh-Based simulations we see an additional effect, as the communication overhead compared to the computation typically can be estimated by 1/N, where N is the problem size of the partitions.
- This is for small problems, there a larger overhead than for big problems is encountered and thus less Performance achieved
- Hopefully, the drawback gets negligible beyond the cache region.

Increased Overheads for Small Problems



Actual Parallel Performance of Musubi



What We Can See

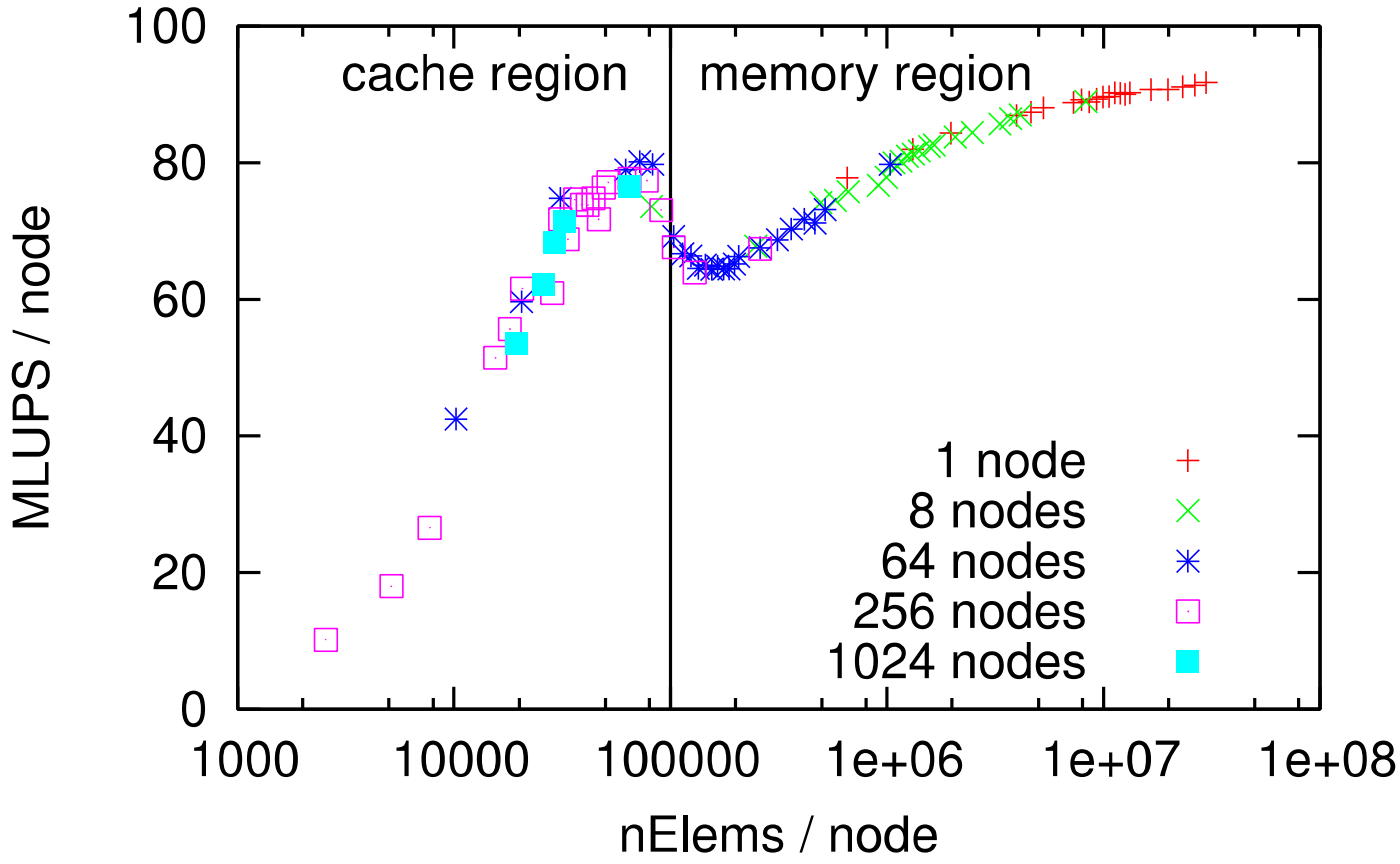
- Performance per node on different node counts nearly coincides → nearly perfect weak scaling

- Problems with relative large overheads get relatively big (performance also increases in the memory region for growing problem sizes).

Scaling

- Efficiency of weak scaling can be directly derived from the vertical distance between points of different process counts
- Efficiency of strong scaling can be seen, under the assumption of ideal weak scaling, by moving from the right end to the left. (Thus, it is given by the steepness of the curve). With a not perfect weak scaling you need to jump from node count to node count appropriately.

Regime of Fairly Reasonable Scaling

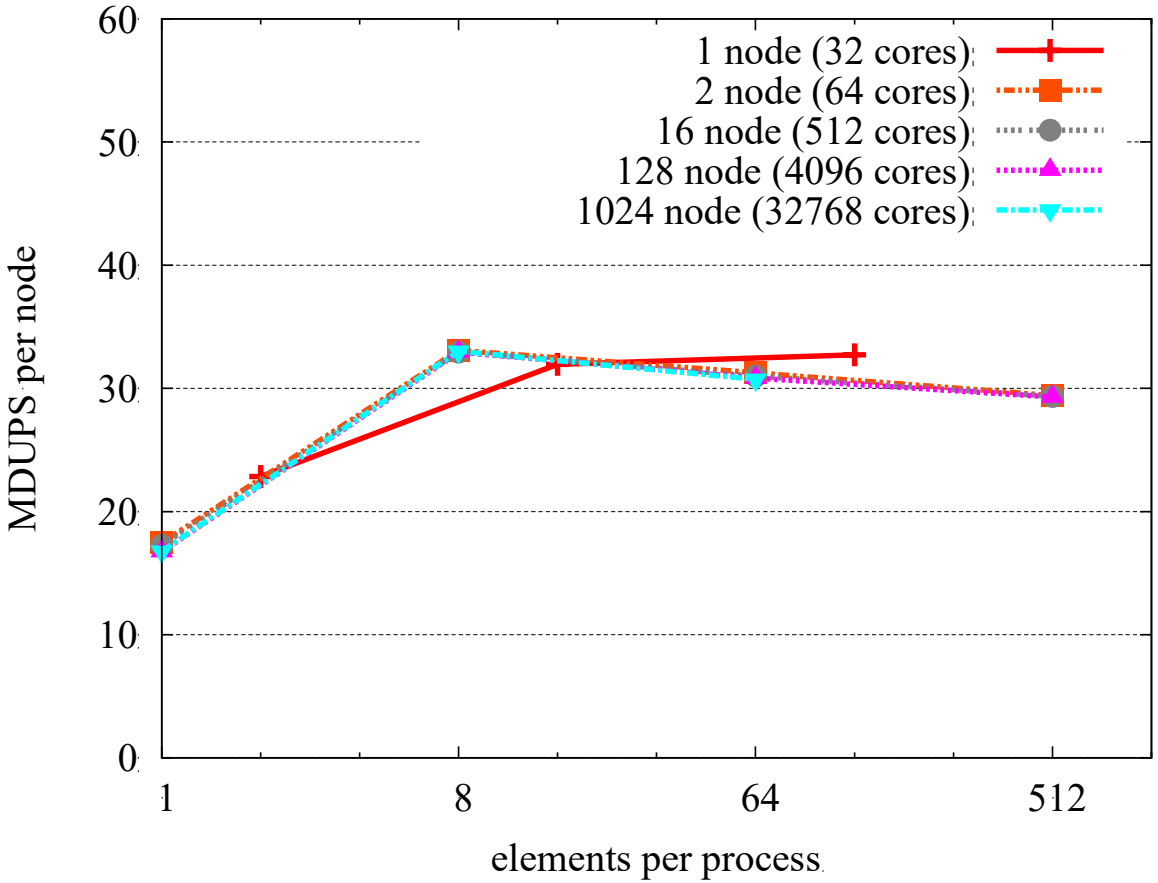


Scaling Up a Given Problem

- There is a largest problem, that fits on a single processing unit
- There is a smallest problem, beyond which performance drops drastically

- The relation of those two tells you approximately, what kind of speed-up can possibly be gained by increasing parallelism

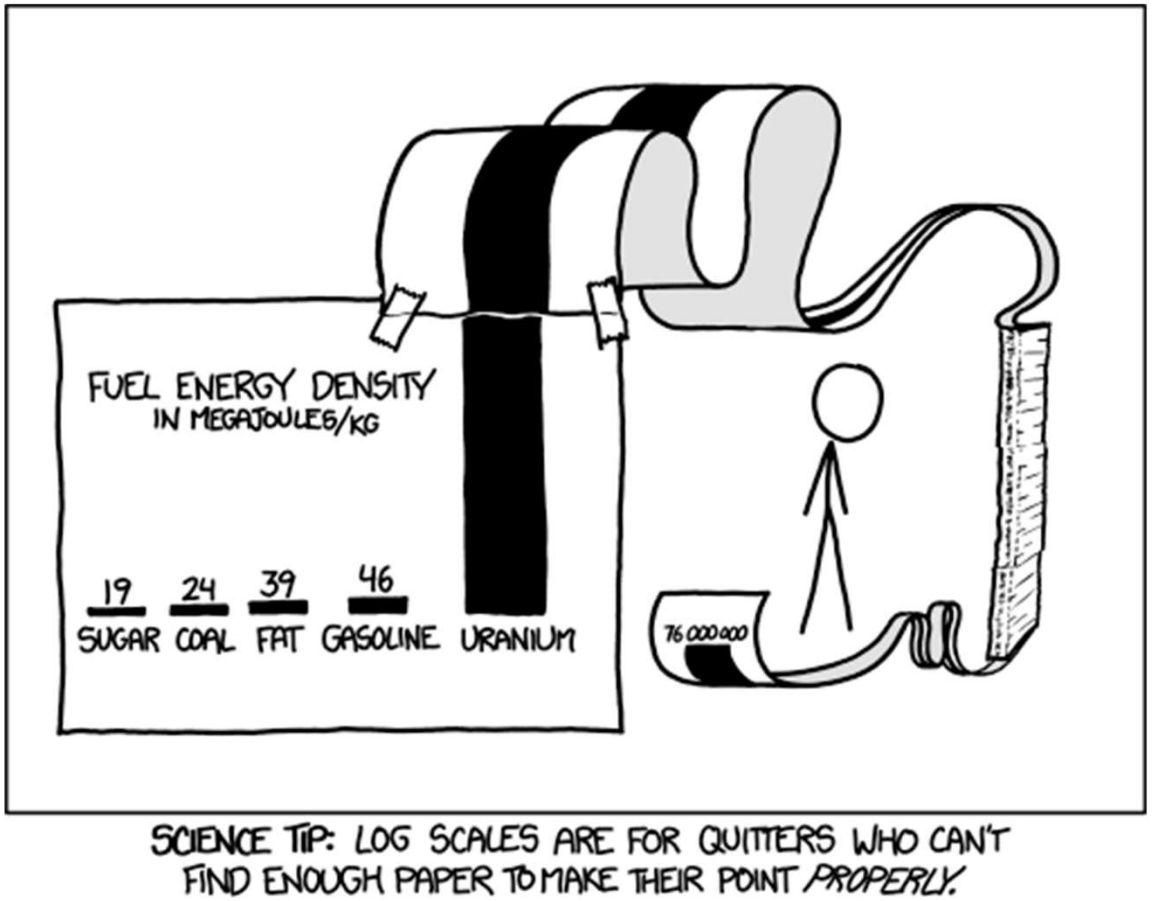
Ateles Performance 16th Order Scheme



Summary

- The performance map neatly characterizes the performance of typical HPC applications.
- While scaling plots can hide bad serial performance, the performance map allows you to make this visible.
- The performance map allows an explanation of scaling behaviors.
- It concentrates the view on the processing unit and does not obscure the actual performance even for large numbers of processors.

Thanks!


FUEL ENERGY DENSITY
IN MEGAJOULES/KG
19
SUGAR
24
COAL
39
FAT
46
GASOLINE
URANIUM
76,000,000
SCIENCE TIP: LOG SCALES ARE FOR QUITTERS WHO CAN'T FIND ENOUGH PAPER TO MAKE THEIR POINT PROPERLY.