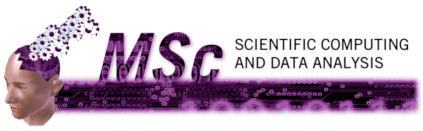

Remarks on a good report

Core I - High Performance Computing

Tobias Weinzierl

January 13, 2020

Below are some remarks what a perfect report could look like. This is a collection of lessons learned from submissions, supplemented by some rationale behind the marking scheme. All remarks are indicative, i.e. you might get really good marks without tickboxing all of the arguments.

1. Profile your code properly:
   - Identify the workhorse function(s) and split it up further, i.e. identify which parts of the code consume the majority of the runtime.
   - Provide data for everything you claim! A simple gprof run or some timings might do the job, but scientific statements require evidence. A statement alike "consumes the majority of the time" is not enough.
   - When you benchmark your code, test it for various input sets. What is the complexity of the code? What does this mean for the relative cost of the individual loops/functions?
   - What do all your insights imply for your upscaling/tuning activities?
2. Set up an upscaling model:
   - Apply both weak and strong upscaling laws.
   - Try to calibrate the free parameters in the law (fraction f, t_0, ...) to your measurements. Be careful w.r.t. the complexity of the code.
   - A roofline model might be good as well. Again, consider the complexity.
   - Provide some plots and limit studies (what happens if core/node counts or vector register widths approach certain values?).
3. Contextualise:
   - Clarify which parallelisation techniques you will (have to) use. Supplement with Flynn's taxonomy. Simple namedropping of terms is not enough here. Point out why certain characteristics (see profiling) fit or do not fit to certain parallelisation concepts from the taxonomy.
   - What does all of this mean for your upscaling models, i.e. which upscaling/performance model is the right one?
   - What are reasonable values when you plot the speedup? For example, does it make sense to upscaling strong scaling for the OpenMP part up to 1,000 processing units?
   - What is the best-case performance improvement you can expect from the different approaches?