EE4C07 - Advanced Computing Systems

Lab report grading scheme 2018-2019
Delft University of Technology

1 Learning Objectives

After completing this course, students will be able to ...

1. Analyze the computational characteristics of an application (e.g. computational parallelism, memory access patterns). This helps students to understand the limitations of running specific algorithms on specific computer hardware.
2. Identify the components of modern multicore architectures (processing elements, memory system, interconnect network) and understand how an algorithm uses these resources. This helps students to estimate how algorithms will perform on specific architectures.
3. Determine the potential performance of implementing an application on various available multicore architectures. This helps students to select an optimal architecture for their application.
4. Use profiling tools to identify bottlenecks in running applications on a specific architecture. This helps students to optimize their implementation, or to tailor their application to a specific architecture.
5. Implement algorithms using various advanced computational technologies (e.g. OpenMP, CUDA/OpenCL, and SIMD extensions). This helps students improve the performance of their software applications.

2 Rubric

The rubric in Table 1 will be used to grade your lab work and lab reports. The rubric is a set of *indicators* for a specific grade. It should *not* be interpreted as a checklist or some legal document along which strictly the grades are given. Lab reports will receive personal feedback in the PDF. If we diverge from the rubric, it will be explained why there.

3 Terminology

What is meant by ...

- Profiling? : Profile the sequential version and improved versions of your application to identify major portions of the run-time (you can use `gprof`, some other tool or manual profiling by inserting timers, as long as you describe your exact method).

- Estimation? : Based on the profile, show the potential speedup if you improve specific high scoring parts of your profile. Use strong and weak scaling laws, and the properties of the hardware platform you are going to run your code on.

Table 1: Rubric for Lab Assignments Advanced Computing Systems (EE4C07) 2018-2019

Description	**Learning objective / (Weight)**	**Unsatisfactory (0)**	**Sufficient (6)**	**Good (8)**	**Excellent (10)**
Profiling the application.	4 / (20%)	Profiling has been performed but the analysis of the application profile has selected less relevant parts to improve, or profiling has not been performed at all.	Profiling has been performed and analysis thereof revealed a major bottleneck in the application with recommendations for improvements.	Profiling has been performed, and analysis thereof reveals one major bottleneck with recommendations for improvement but also reveals some minor bottlenecks.	Profiling has been performed, and analysis thereof reveals at least one major bottleneck and some minor bottlenecks all for which recommendations for improvement are given.
Estimating the potential of improving and porting.	3 / (10%)	The estimation does not exist, uses wrong figures, has been performed incorrectly or does not contain any motivation or explanation.	The correct steps to perform the estimation have been taken, but there are some minor mistakes.	The estimation has been performed correctly.	The estimation has been performed correctly and there is an analysis with regard to the practical feasibility.
Functionality and design of the improved application.	5 / (30%)	The implementation does not function according to specification, critical parts are not functional or contain bugs. There is no obvious structure in the implementation, the implementation appears messy.	The critical parts of the implementation are functional, but cases exist where small bugs hinder minor parts of the specified functionality. There is some structure in the implementation, but some parts are still a bit messy.	The implementation is functional according to specification. The implementation is well structured.	The implementation is functional according to specification, and additional functionality related to the course has been added. The implementation is well structured and the structure has been documented inside or outside of the code.
Attempted optimization	5 / (10%)	The requested optimizations have not all been attempted, or only the most trivial ones have been attempted. The discussion about the optimizations contains many erroneous statements.	The requested optimizations have been attempted. Only a brief discussion is present. In this discussion, some insight is shown but it is not always entirely correct.	The requested optimizations have been attempted and they are discussed in such a way that proper insight about them is shown.	Additionally some in-depth optimizations that go beyond the immediate goals of the course or lab are included.
Quality of the report	Reperesents your progress in all learning goals. / (30%)	The report is unstructured, contains many language errors, and is unclear in describing the analysis, performed work, results and conclusions.	The report is reasonably structured, does not contain many language errors, is mostly, but not always, clear about analysis, performed work, results and conclusions.	The report is well structured, contains almost no language errors, is clear about analysis, performed work, results and conclusions.	The report is of very high quality, in language, structure and clarity, contains all the aforementioned parts and shows insights that go somewhat beyond the contents of the course.