M U N I

HABILITATION THESIS REVIEWER'S REPORT

Masaryk University	
Applicant	RNDr. Jiří Filipovič, Ph.D.
Habilitation thesis	Software Performance Optimization in Scientific Computing
Reviewer	Anton Wijs, PhD
Reviewer's home unit, institution	Parallel Software Development group, Software Engineering & Technology, Faculty of Mathematics and Computer Science, Eindhoven University of Technology

This habilitation thesis presents an impressive body of work on the topic of Software Performance Optimization, with a considerable scope. The applicant has acted as first author in the vast majority of the included papers, and as principal investigator in six of those papers, most of them addressing the autotuning part of the work. In the first part, there are contributions on the algorithmic level, with a particular application in computational biology, and various GPU accelerations of methods from computational chemistry and cryo-electron microscopy, addressing molecular docking, computation of dRMSD, movie alignment and 3D refinement. This resulted in a number of high performance tools.

Besides this, there is considerable work on kernel fusion and autotuning for CPU, GPU and Xeon Phi. The work on kernel fusion addresses both data-independent and data-dependent kernel fusion, with the former involving both serial and parallel fusion. Regarding data-dependent kernel fusion, kernels are targeted without cross-thread block dependencies, as those dependencies would require global synchronisation barriers, which are only provided in-between kernel launches. Furthermore, the fusion of kernels that perform map, reduce, or nested combinations thereof are discussed. The contribution here is that the first-order function executed by the map or reduce is allowed to be parallel. The compiler developed by the applicant and co-authors is thoroughly experimentally evaluated. Finally, the Kernel Tuning Toolkit is presented, to automate inter-kernel optimization and dynamic autotuning. Addressing inter-kernel optimization is novel, as comparable tools completely focus on the autotuning of individual kernels. Dynamic autotuning is supported, allowing a piece of software to tune itself when executed on different hardware. In addition, it uses a portable model involving performance counters, which allows the computation of a model on a particular GPU, and using the gained information to speed up the autotuning of the software when running on a different GPU or when processing different input.

On a more critical note, whereas the individual chapters provide impressive results, any connections between these chapters are not made explicit. This makes the thesis more a collection of published papers than a single body of work with one clear research vision. This is not per se bad, but I would have enjoyed reading about how the methods of the various chapters possibly relate to each other, and reading more about the vision of the applicant for the future. My questions below therefore address these points, and some others, in detail.

Reviewer's questions for the habilitation thesis defence (number of questions up to the reviewer)

1. The thesis contains contributions on various approaches to software performance optimization: changing the mathematical formulation of the involved model, optimizing source code, and (possibly automatically) accelerating implementations on GPUs. Taking a step back, can you explain how to go about improving the performance of a piece of scientific computing software in general? Which of these techniques should be tried first, and why? In other words, how and in which order should one try to apply these methods?
2. A specific question related to a comment made on page 13, last paragraph: You mention that you mainly focussed on the design of the surgical simulators. Can you elaborate on why specifically this was focussed on? As you mention, the mathematical model is computationally demanding, but is it also particularly amenable to parallelisation? If so, why?
3. A comment: Figure 2.5 seems to be missing.
4. On page 30, paragraph 2, a remark is made regarding Table 4.3 that often the performance is close to the theoretical peak. However, for the K20, E5-2650 and 5110P this seems not to be the case at all. Can you elaborate on that, and give a reason for these numbers?
5. On page 31, it is mentioned that we can see in Table 4.4 that the performance penalty of dynamic tuning is smaller than the performance penalty obtained when code is used that was tuned offline for different hardware. It is not clear to me how we can see that. Can you explain how to do this?
6. I have two specific questions about the paper “Optimizing CUDA code by kernel fusion: application on BLAS”, page 3941. First of all, it is remarked that data exchanged between f and g cannot be placed in registers. Does this not entirely depend on how the threads are mapped on the data for f and g individually? Furthermore, intra-warp sharing of register data can also play a role here to use the output of f as input for g within warps. Can you comment on that?
7. Secondly, it is remarked on the same page that functions with different nesting depths are not fused, as this yields redundant execution of functions with lower nesting depth. I do not see why that is the case. Are the functions not executed in sequence then?
8. The second half of the thesis addresses automatic kernel fusion and autotuning, while the first half is about improving the performance of particular computations from various fields. I wonder if the techniques developed in the second half are applicable on the computations of the first half, and whether this has been tried. Would these techniques result in further improvements, or are they not (completely) applicable, and why (not)?
9. High performance computing still requires expert knowledge about targeted hardware platforms. Do you see this changing in the future, and how? Will your research contribute to that, and to what extent? If not, or not completely, are there fundamental limitations, or is it just a matter of time before that has changed?
10. Maybe as a continuation of the previous question(s), with the prominent rise of AI, how will, in your view, AI affect High Performance Computing? Will it play a role in your future research, and if so, how?
11. Finally, what is your longterm vision for your research? Do you have a point on the horizon, or moonshot challenge, to strive towards, and if so, what is it?

Conclusion

The habilitation thesis entitled Software Performance Optimization in Scientific Computing by Jiří Filipovič **fulfils** requirements expected of a habilitation thesis in the field of Computer Science.

Date: Signature:

30/03/2024