External Examiner's Report

Candidate: Mgr. Michal Brabec
Title: Procedural code integration in streaming environments
Supervisor: RNDr. David Bednárek, Ph.D.

Summary

This thesis is an extensive study of performance issues in procedural code in `C#`. This work brings new insights, through a design of novel intermediate code (Hybrid Flow Graph). The original procedural code is transformed into this graph-based language. Then the intermediate form is optimized by the graph algorithms to be better suited for the streaming environments. Finally, the intermediate form is transformed into a streaming application.

Main contributions and strengths

The main contributions and strengths of this thesis include:

1. The thesis presents a complete unique framework for transformation of a procedural code to a streaming application.

2. The experimental results show that this simple idea is surprisingly effective.

3. The systematic approach, the coverage of a wide spectrum of relevant algorithms.

4. The careful and thorough experimental work.

Weaknesses,Questions, Problems, Comments

Of course, there is also always a lot what can be added and improved in the contents itself. In my opinion. conceptual issues include:

- The code optimization is aimed at maximal utilization of ALU (vector) units, but the memory subsystem including a cache hierarchy is not taken in account. All dynamic programming problems are memory-intensive problem, the overall performance depends mainly on the memory bandwidth not on the maximal performance of ALU units

- The thesis does not provide a thorough survey of related (similar and different) projects in the field. In particular, if the relatively old (and deprecated) Brook project is described, **why new ones related to GPU programming (CUDA, OpenCL, OpenACC) are missing?**

- The text is based on candidate's scientific papers (7 conference papers, 1 paper published in high impact journal) and it is sometimes difficult to follow:

  - Some terms e.g., ParallaX compiler is not (even briefly) defined before the first use.
  - Second example: in Section 3.4 vectorization of code is discussed and suddenly GPU and blocked Levenshtein distance algorithm are mentioned.
  - Third example: I think that nobody is able to understand a transformation of the code in Listing 4.1 to the graph in Fig. 4.1 without reading Sections 4, 5, and 6.

- There is only one case study for matrix-based dynamic programming class of algorithm and it is only aimed at efficient implementation of blocked Levenshtein distance algorithm.

- Some decision are not properly discussed. For example, why the Bobox system is chosen? I understand that it is a product of candidate's supervisor, but exhaustive analysis about alternatives, their advantages/drawbacks/limitations should be done.

- Section 8.2: „The compiler currently supports four target platforms.“ **Which ones?**

- I have a lot of comments to measurement and its evaluation

  - Usually, the dedicated Linux-based servers are used for measurements to eliminate the influence of other processes.
  - The performance (not time per elementary operation) is usually computed.
  - Why logarithm scale is used? It is binary or decadic logarithm?
  - If I assume that the binary logarithm is used, then in some graphs the ratio between upper bound and lower bound of performance is more than 4. **How you explain such difference?**

  - **What it exactly means datasize for filter applications?**

- The input for your algorithm is a program in Common intermediate language (CIL). CIL is a stack-based, object-oriented assembly language. CIL and Java bytecode are very similar, so **how difficult will be modification of your program to accept also Java bytecode ?**

- Some minor or formal issues:
  - 20 virtual threads ⇒ HT logical cores ?
  - Sometimes wrong forms of some shortcuts appear (e.g., Gpu, Sql).
  - The code in Listing 2 is wrong – if one of streams reaches the end, the remaining items from the second stream should be copied to the output.
  - Presented listings contain just few comments and some of them are on multiple pages.
  - There are two parts with the same name – Appendix A.

Final Statement

In spite of the criticism above, I think that the author of this thesis has proved to have an ability to perform scientific research and to achieve scientific results. The subject of the thesis is actual, it seems that it reached the intended goal, it contains original results and presents a reasonable starting point for further research in the field. Based on this, I believe that the author is a good candidate for the PhD. degree and **I am recommending the thesis for the defense**.

doc. Ing. Ivan Šimeček, Ph.D.
Department of Computer Systems
Faculty of Information Technology
Czech Technical University in Prague