Theory and Practice of Vectorial Extension for Stream Processing

2023-2024

LCIS Laboratory (Valence – France) & VERIMAG Laboratory (Grenoble – France)
Advisors: Laure Gonnord (Full Professor, Grenoble-INP ÉSISAR/LCIS)
laure.gonnord@lcis.grenoble-inp.fr
Bruno Ferres (Associate Professor, UGA/VERIMAG)
bruno.ferres@univ-grenoble-alpes.fr

Keywords: RISC-V; vector machine; compilation; operational semantics

Context

Stream processing is at the heart of many applications, such as network traffic monitoring, data analysis or bioinformatics. In each of these domains, basic computation blocks can easily be defined, but composing them efficiently remains a complex task, when addressing workloads with terabytes of data. A solution to manage those computation-intensive applications is to exploit the parallelism potential of the algorithms, which can be done through *vectorial programs* [2], *i.e.*, programs where instructions are applied to vectored data structure, instead of single elements. To execute vectorial, we need vectorial machines. Vectorial machines can be built using *Single-Instruction Multiple Data* architectures, which support applying the same instruction to multiple data. Vectorial extensions have been proposed for several families of standard CPUs (SSE/AVX for Intel x86, SVE for arm, . . .). Several projects have shown that vectorial programs can be used for efficient computations, not only in domains with operates with *"regular data"* [1], but also in stream processing [3, 4]. However, for stream processing, solutions are mainly written by hand (using C with intrinsics), and strongly depend on the chosen backend. The approach stills lacks a proper formalisation of the expressivity of vectorial backends, so that to compare between them and/or prove the correctness of future compilation schemes. Operational sematics of languages [5] provides us theoritical tooling to reason on such backends.

This internship will be part in a national research project, which aims at developping a programming language fitted to write stream processing applications, with a specific compilator to generate efficient assembly codes for vectorial machines. One of the first steps of the work would be to capture the expressivity of such vectorial machines (Intel AVX, and the more recent free proposal, RISC-V "V" [6]).
This internship is hence a preliminary step towards a RISC-V "V" backend, where we want to assert that the existing vectorial extension for RISC-V includes every instruction needed to compile stream processing applications.

Goals of the Internship

The goal of this internship is to setup a framework to assert that the RISC-V "V" extension is suitable for the compilation of stream processing application.
It is thus expected that the applicate will:

- study the existing literature to define (or reuse) an operational semantics of the vectorial extension of RISC-V ISA;
- implement several examples of stream processing applications using this semantics, to demonstrate the suitability;

- if needed, propose extensions of the ISA (in the operational semantics) to fit the needs of stream processing.

This internship will rely on the expertise of both VERIMAG and LCIS laboratory, in the domain of compilation and computer architectures as well as streaming languages.

Application

The applicant should be a L3/M1 student in Computer Science, with a strong knowledge in compilation, and an interest in computer architectures. Language semantics is a plus, nevertheless the applicant should not be afraid of mathematical-like formalism. Knowledge of the application domain (*i.e.*, stream processing) is not required, as examples will be provided.

To apply, send an email to laure.gonnord@univ-grenoble-alpes.fr and bruno.ferres@univ-grenoble-alpes.fr, with your resume, a short covering letter, as well as any document that may support your application.

Location

The internship may take place either in Grenoble or Valence, based on the candidate's preferences. Visits to the other location are planned during the internship.

First location : LCIS laboratory, located in the campus of Valence:

Laboratoire LCIS,
50 rue Barthelemy de Laffemas,
26902, Valence

Second location : VERIMAG laboratory, located in the campus of Grenoble:

Laboratoire VERIMAG, Bâtiment IMAG,
150 place du Torrent,
38401, Saint-Martin-d'Hères

References

[1] T. Grosser, H. Zheng, R. Aloor, A. Simbürger, A. Größlinger, and L.-N. Pouchet. Polly-Polyhedral optimization in LLVM. In *IMPACT*, 2011.

[2] L. Lamport. Multiple Byte Processing with Full-Word Instructions. *Commun. ACM*, 1975.

[3] G. Langdale and D. Lemire. Parsing gigabytes of JSON per second. *The VLDB Journal*, 2019.

[4] F. M. M. Gienieczko and C. Paperman. Supporting Descendants in SIMD-Accelerated JSONPath. In *ASPLOS*. ACM, 2023. https://github.com/V0ldek/rsonpath.

[5] H. R. Nielson and F. Nielson. *Semantics with Applications: An Appetizer (Undergraduate Topics in Computer Science)*. Springer-Verlag, Berlin, Heidelberg, 2007.

[6] RISC-V "V" Vector Extension. https://github.com/riscv/riscv-v-spec/releases/tag/v1.0.