[PhD] An intermediate representation for vectorial program

Charles Paperman (charles.paperman@inria.fr)
Laure Gonnord (laure.gonnord@esisar.grenoble-inp.fr)

2024

Efficient Data Processing

Streaming data processing is a crucial approach that focuses on traversing data to extract pertinent information. Applications ranges from network packet manipulation to analysing DNA. Modern data-processing tools heavily depend on efficient implementations that harness hardware acceleration to achieve high performance. This acceleration can sometimes be achieved through automatic compilation, but frequently demands expert developers to craft optimizations by hand.

One critical facet of this optimization process involves SIMD optimization, where data is packed into chunks and processed with minimal branching in the code, often using bitvector operations. These optimizations are at the core of numerous well-known software applications, such as regular expression matching in tools like ripgrep, JSON parsing in libraries like SimdJSON, and even fundamental operations like string encoding and decoding (unicode parsing). Developing these optimizations requires a broad skill set and is hard to achieve.

Developping an intermediate representation for Vectorial Program (VIR)

During this PhD, we will explore the design and implementation of an intermediate representation whose aim is to model carefully the vectorial ability of hardware. It will be based on knowledge from two fields: synchronious programming (*à la Lustre*) and automata theory (through vectorial circuits).

Initially we will focus on an intermediate representation with a very limited scope and gradually add more operators while keeping the semantic well understood. Compilation of VIR toward common CPU architecture will be studied as well as well as more experimentation compilation within the RISCV ecosystem.

Tasks

- Play with concrete vectorial instruction with various hardware (x86 AVX variant, ARM SVE, handcrafted efficient SIMD code)
- Design VIR with its formal semantic and a toy compilation scheme
- Connect VIR with the Vectoid experimental language in collaboration with another PhD student
- Expand VIR to cross-plateform compilation procedure and its interplay with MLIR.
- Study VIR within the RISC V ecosystem and notably with RISC V Vectorial extension.
- Formal modelisation of VIR through Automata theory

Location

This PhD will be co-advise by Laure Gonnord (Université Grenoble-Alpes) and Charles Paperman (Université de Lille).

Candidate profile

The candidate should ideally be familiar with formal approaches in programming language design, notably compilation and logic. From the practical point of view, a basic experience in software programming and usage of collaborative tools such that git. This PhD strongly relies on the fact that practical implementation should have strong theoretical foundations and that further refinements of the theory should get inspiration from the practical side. We expect the candidate to agree with this philosophy.

Bibliography

1. Vectorial languages and linear temporal logic.
2. An algebraic approach to vectorial programs
3. Efficient compilation of array iterators for Lustre
4. Multi-task implementation of multi-periodic synchronous programs