Utilizing Parallel Workers: LLVM's Vectorization Plan

Jonas Fritsch
Technical University of Munich
jonas.fritsch@tum.de

Abstract

Your Abstract.

1 Introduction

1.

2 Background

1. (SIMD explanation necessary?)
2. Vectorization explanation (with Code-Examples?)
   a. Inner-Loop
   b. Outer-Loop
   c. Superword Level Parallelism [6] (maybe already [2])
3. Vectorization Drawbacks (reuse code-examples from above)
   - possible performance loss
   - larger code size
   - (potential security vulnerablities [5])
4. LLVM short introduction

3 LLVM's Vectorization Plan

1. Cost-Model (related work? [8])

4 Related Work

1. Auto-Vectorization in GCC [4, 10]
2. Comparison with GCC [3, 7]
3. (LLM-based Vectorization [9], also if included maybe better fitting in "Future Work")

5 Summary and Future Work

1. VPlan Summary and Future Roadmap [11]
2. Comparison/Outlook auto-vectorization vs. manual vectorization [1, 3]

References

[1] Neil Adit and Adrian Sampson. 2022. Performance Left on the Table: An Evaluation of Compiler Autovectorization for RISC-V. *IEEE Micro* 42, 5 (2022), 41–48. https://doi.org/10.1109/MM.2022.3184867

[2] Yishen Chen, Charith Mendis, and Saman Amarasinghe. 2022. All you need is superword-level parallelism: systematic control-flow vectorization with SLP. In *Proceedings of the 43rd ACM SIGPLAN International Conference on Programming Language Design and Implementation (PLDI 2022)*. Association for Computing Machinery, New York, NY, USA, 301–315. https://doi.org/10.1145/3519939.3523701

[3] Jing Ge Feng, Ye Ping He, Qiu Ming Tao, and Fazli Wahid. 2021. Evaluation of Compilers' Capability of Automatic Vectorization Based on Source Code Analysis. *Sci. Program.* 2021 (Nov. 2021), 15. https://doi.org/10.1155/2021/3264624

[4] Jakub Jelínek. 2023. Vectorization optimization in GCC. https://developers.redhat.com/articles/2023/12/08/vectorization-optimization-gcc Accessed: 2024-11-04.

[5] Sayinath Karuppanan and Samira Mirbagher Ajorpaz. 2023. An Attack on The Speculative Vectorization: Leakage from Higher Dimensional Speculation. *arXiv e-prints*, Article arXiv:2302.01131 (Feb. 2023), 15 pages. https://doi.org/10.48550/arXiv.2302.01131 arXiv:cs.CR/2302.01131

[6] Samuel Larsen and Saman Amarasinghe. 2000. Exploiting superword level parallelism with multimedia instruction sets. In *Proceedings of the ACM SIGPLAN 2000 Conference on Programming Language Design and Implementation (PLDI '00)*. Association for Computing Machinery, New York, NY, USA, 145–156. https://doi.org/10.1145/349299.349320

[7] Klara Modin. 2024. A comparison of auto-vectorization performance between GCC and LLVM for the RISC-V vector extension. https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-354873

[8] Angela Pohl, Biagio Cosenza, and Ben Juurlink. 2020. Vectorization cost modeling for NEON, AVX and SVE. *Performance Evaluation* 140-141 (2020), 102106. https://doi.org/10.1016/j.peva.2020.102106

[9] Jubi Taneja, Avery Laird, Cong Yan, Madan Musuvathi, and Shuvendu K. Lahiri. 2024. LLM-Vectorizer: LLM-based Verified Loop Vectorizer. arXiv:cs.SE/2406.04693 https://arxiv.org/abs/2406.04693

[10] GCC Team. 2023. Auto-vectorization in GCC. https://gcc.gnu.org/projects/tree-ssa/vectorization.html Accessed: 2024-11-04.

[11] LLVM Team. 2024. Vectorization Plan. https://llvm.org/docs/VectorizationPlan.html Accessed: 2024-11-04.