Topic: Basic Block Cost Heuristic for SLP

Group #20: Sam Gonzalez, Kyle Astroth, Carson Hoffman, Xiangyu Qin

---

Project Proposal:

From our research, Clang's heuristic for deciding when to perform vectorization uses hand-picked costs assigned to each individual instruction. This instruction cost based heuristic correlates poorly with actual performance gains [1], and more sophisticated machine learning heuristics have been shown to outperform it in loop vectorization [2]. Our team proposes extending the work in NeuroVectorizer [2] by implementing a machine learning heuristic for SLP that examines the context of neighboring instructions in a basic block. Finally, we will measure the performance of our heuristic empirically by evaluating it against clang's individual instruction cost heuristic by compiling a testsuite of benchmarks and comparing their runtime results.

Proof of Concept:

Clang's cost model for SLP outperforms GCC in figure 1, however it misses the opportunity to perform SLP in figure 2. We suspect this is because Clang estimates SLP cost at an instruction level instead of using a heuristic that examines the context of the whole basic block.

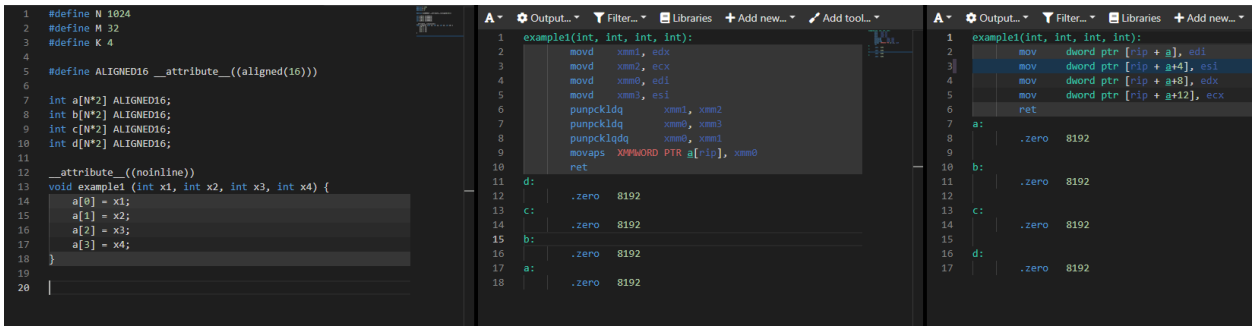

Fig. 1

Source code on the left, GCC assembly in the middle and Clang assembly on the right. Clang makes the right choice in not performing SLP. From testing the assembly on the right is 26% faster than the assembly in the middle.

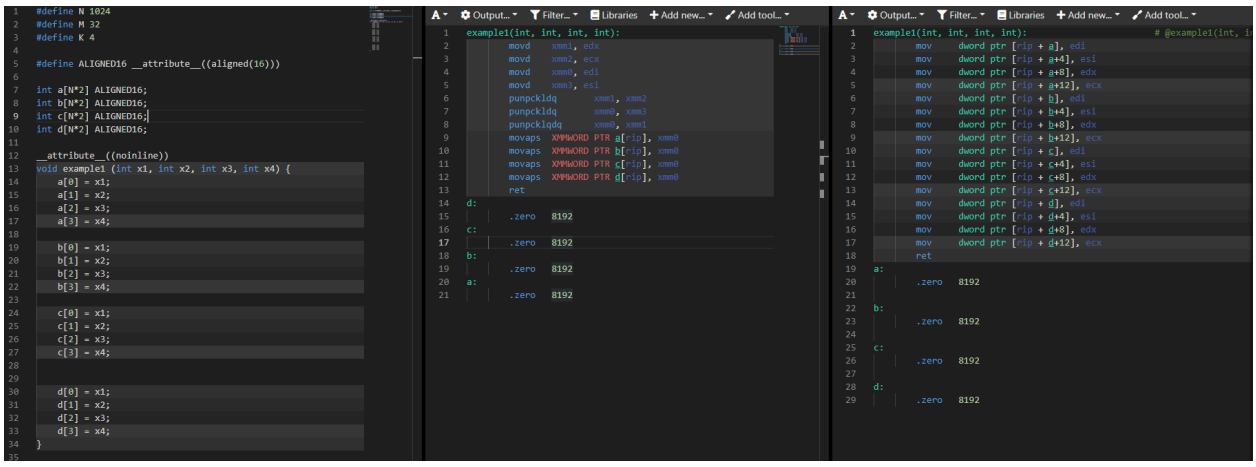

Fig. 2

Source code on the left, GCC assembly in the middle and Clang assembly on the right. Clang makes the wrong choice in not performing SLP. From testing the assembly on the right is 240% slower than the assembly in the middle.

References:

1. Vectorization Cost Modeling for NEON, AVX and SVE⋆
2. NeuroVectorizer: End-to-End Vectorization with Deep Reinforcement Learning