**SuperNode-SLP: Vectorizing Chains of ADD/Subs**

Vasileios Porpodas\(^1\), Rodrigo C. O. Rocha\(^2\), Evgueni Brevnov\(^1\), Luís F. W. Góes\(^3\), and Timothy Mattson\(^1\)

Intel Corporation\(^1\), University of Edinburgh\(^2\), PUC Minas\(^3\)

Intel Corporation

http://vporpo.me

---

**SLP** Auto-vectorization converts straight-line code into vector code. **Super-Node SLP (SN-SLP)** is an improvement on SLP trunk, optimized for expressions that include a commutative operation (such as addition) and its corresponding inverse element (subtraction).

**SN-SLP** uses the algebraic properties of commutative operators and their inverse elements to enable aggressive operand reordering across groups of instructions, which we refer to as Super-Nodes. Super-Nodes extend the Multi-Nodes of “Look-Ahead SLP”, presented in EuroLLVM’18. They form chains of both commutative operations and their inverse operations and allow for legal operand reordering across them.

---

**SLP vs LV**

The Loop Vectorizer (LV) is vectorizing across iterations. SLP is vectorizing across instructions.

---

**SLP Can Fail on ADD/SUB Chains**

SLP cannot reorder operands across chains of ADD/SUB (or MUL/DIV).

SN-SLP forms a “Super-Node” and reorders across them.

---

**SN-SLP Can Reorder Internal Nodes of the Super-Node**

When operands cannot be reordered, we can try reordering the internal nodes of the Super-Node.

---

**Performance**

Target: Intel® Core\textsuperscript{TM} i5-6440HQ CPU

Compiler flags: -O3 -fast-math -arch=native -mtune=native, CPU2006 O3: Trunk LLVM with all vectorizers disabled.

**LSLP**: Trunk SLP + MultiNodes (patches under review).

**SN-SLP**: Trunk SLP + SuperNodes (patches coming soon).

---

**We are Upstreaming it!**

[SLP] Patches for commutative instructions are out. They are adding support for Multi-Nodes, i.e., chains of commutative operations.

Super-Node patches coming soon!

---

**Legality**

Reordering operands (e.g., D and B) is legal if they have the same APO. Even if the APO is not the same, we may still reorder the successors.

---

**Conclusion**

SN-SLP improves SLP on code with ADD/SUB MUL/DIV chains.

It forms Super-Nodes of commutative operations and their inverse elements.

It performs legal operand reordering, guided by the Look-Ahead heuristic.

Please check out our CGO’19 paper:
