FACEBOOK

A fast algorithm for global code motion of congruent instructions

Aditya Kumar

@_hiraditya_

Global scheduling in SSA

- Middle End Optimization
- Generalization of `GVNHoist` and `GVNSink` with improved cost-model

Using modern data structures

- Augmented SSA
- DJ Graph
- Fast liveness analysis in SSA

Alpha and Beta Nodes [Augmented SSA]

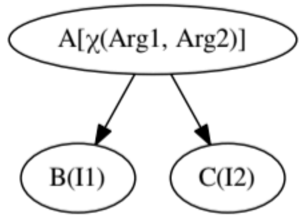

Figure 1. I1, I2 are instructions in B, and C respectively, Arg1 = {B, I1, V} and Arg2 = {C, I2, V}, V is the value number for both I1, and I2

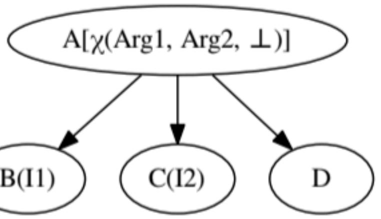

Figure 2. I1, I2 are instructions in B, and C respectively, Arg1 = {B, I1, V} and Arg2 = {C, I2, V}, V is the value number for both I1, and I2. A has missing entry in χ so V is not anticipable

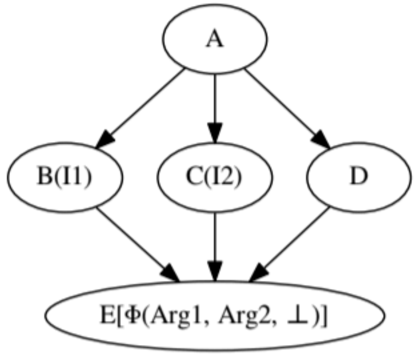

Figure 3. I1, I2 are instructions in B, and C respectively, Arg1 = {B, I1, V} and Arg2 = {C, I2, V}, V is the value number for both I1, and I2. A has missing entry in Φ so V is not available

Cost Model

- Reduces live range of virtual registers
    - Reduces register pressure
- Hoist followed by sink

Performance Improvements

Spec2006 (interesting benchmarks)	Ratio (higher is better)
403.gcc	1.03
462.libquantum	1.03
464.h264ref	1.02
433.milc	1.15
470.lbm	1.07

References

- Global code motion of congruent computations
  - https://reviews.llvm.org/D32140
- llvm/lib/Transforms/Scalar/GVNHoist.cpp
- llvm/lib/Transforms/Scalar/GVNSink.cpp