A fast algorithm for global code motion of congruent instructions

Aditya Kumar
Global scheduling in SSA

• Middle End Optimization
• Generalization of GVNSHoist 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.** N1, N2 are instructions in B, and C respectively, Arg1 = \{B, N1, V\} and Arg2 = \{C, N2, V\}, V is the value number for both N1, and N2

**Figure 2.** N1, N2 are instructions in B, and C respectively, Arg1 = \{B, N1, V\} and Arg2 = \{C, N2, V\}, V is the value number for both N1, and N2. A has missing entry in $\Phi$ so V is not anticipable

**Figure 3.** N1, N2 are instructions in B, and C respectively, Arg1 = \{B, N1, V\} and Arg2 = \{C, N2, V\}, V is the value number for both N1, and N2. A has missing entry in $\Phi$ so V is not available
Cost Model

• Reduces live range of virtual registers
  • Reduces register pressure
• Hoist followed by sink
## Performance Improvements

<table>
<thead>
<tr>
<th>Spec2006 (interesting benchmarks)</th>
<th>Ratio (higher is better)</th>
</tr>
</thead>
<tbody>
<tr>
<td>403.gcc</td>
<td>1.03</td>
</tr>
<tr>
<td>462.libquantum</td>
<td>1.03</td>
</tr>
<tr>
<td>464.h264ref</td>
<td>1.02</td>
</tr>
<tr>
<td>433.milc</td>
<td>1.15</td>
</tr>
<tr>
<td>470.lbm</td>
<td>1.07</td>
</tr>
</tbody>
</table>
References

• Global code motion of congruent computations
  • [https://reviews.llvm.org/D32140](https://reviews.llvm.org/D32140)
• [llvm/lib/Transforms/Scalar/GVNHoist.cpp](https://llvm.org/lib/Transforms/Scalar/GVNHoist.cpp)
• [llvm/lib/Transforms/Scalar/GVNSink.cpp](https://llvm.org/lib/Transforms/Scalar/GVNSink.cpp)