Project 2 Report
ECE 566 Spring 2022
Shawn Salekin

Q1

1. **CSE_Dead**: this is run separately as an initial pass.. For every instruction, we check whether it is a terminating instruction or it may contain side effects. Since it is a trivial check, we only remove the instructions whose uses are empty.
2. **CSE_Simplify**: In this pass, we try to simplify instruction and replace the uses with the calculated results.
3. **CSE_RLoad**: In this pass, we check if two consecutive loads are literal matches, similar to the check suggested in CSE_Basic. If they turn out to be the same, we remove the second one and update their use. For this optimization, we consider only the instructions that are in the same basic block.
4. **CSE_Store2Load**: Similar to CSE_RLoad, we remove a Load that comes after a non-volatile store and update its use. (**CSE_RStore** implementation is buggy so its commented out). For this optimization, we consider only the instructions that are in the same basic block.
5. **CSE_Basic**: We iterate over all the functions in a module. Then we get the first basic block of that function and build a DominatorTree based on that. Starting at the root node, we go over each child of the tree and perform CSE between the root and that child node. We repeat the same for each node recursively. Since we ran into some implementation issues, we are not performing CSE within the same basic block.

Q2

```
=================================
Instructions
Category                    CSE    M2RCSE
adpcm......................417.......242
arm........................732.......403
basicmath..................572.......343
bh........................3197......2049
bitcount...................643.......436
crc32......................142........83
dijkstra...................319.......228
em3d......................1198.......662
fft........................731.......446
hanoi.......................91........51
hello........................4.........2
kmp........................537.......375
l2lat.......................88........57
patricia..................1047.......699
qsort......................144.......100
```

```
sha........................626.......411
smatrix....................291.......227
sql.....................171424....109423
susan...................12232......7514
=================================
Loads
Category                   CSE    M2RCSE
adpcm......................121........15
arm........................216........48
basicmath..................153........24
bh.........................818.......195
bitcount...................155........51
crc32.......................34.........8
dijkstra....................92........47
em3d.......................398.......107
fft........................206........38
hanoi.......................25.........6
hello................(missing).(missing)
kmp........................153........58
l2lat.......................19.........5
patricia...................354.......134
qsort.......................35........13
sha........................179........42
smatrix.....................73........34
sql......................54785.....16393
susan.....................4189......1030
=================================
Stores
Category                   CSE    M2RCSE
adpcm.......................81.........7
arm........................116........18
basicmath..................100........12
bh.........................494.......142
bitcount....................98........18
crc32.......................29.........4
dijkstra....................51........24
em3d.......................192........43
fft........................102........24
hanoi.......................16.........4
hello........................1.(missing)
kmp.........................71........20
l2lat.......................15.........1
patricia...................108........30
qsort.......................16.........4
sha.........................99........28
```

```
smatrix.....................31........10
sql.....................21894......5842
susan....................1438.......157
==================================
CSEDead
Category                       CSE    M2RCSE
adpcm.......................1.........2
arm.........................0.........0
basicmath...................2.........1
bh.........................32.........1
bitcount....................1.........1
crc32.......................0.........0
dijkstra....................0.........0
em3d........................1.........3
fft.........................1.........0
hanoi.......................0.........0
hello.......................0.........0
kmp.........................0.........0
l2lat.......................3.........0
patricia....................1.........0
qsort.......................1.........1
sha.........................0.........0
smatrix.....................1.........0
sql.......................272.......188
susan.......................3.........0
==================================
CSEElim
Category                       CSE    M2RCSE
adpcm.......................0.........3
arm.........................5.........9
basicmath...................0.........2
bh..........................5........13
bitcount....................0.........5
crc32.......................0.........0
dijkstra....................0.........5
em3d........................0.........8
fft.........................0.........2
hanoi.......................0.........1
hello.......................0.........0
kmp.........................0.........6
l2lat.......................0.........1
patricia....................4........32
qsort.......................0.........1
sha.........................1........12
smatrix.....................0.........4
```

```
sql........................223......2637
susan........................9.......245
=================================
CSEStElim
Category                   CSE    M2RCSE
adpcm........................0.........0
arm..........................0.........0
basicmath....................0.........0
bh...........................0.........0
bitcount.....................0.........0
crc32........................0.........0
dijkstra.....................0.........0
em3d.........................0.........0
fft..........................0.........0
hanoi........................0.........0
hello........................0.........0
kmp..........................0.........0
l2lat........................0.........0
patricia.....................0.........0
qsort........................0.........0
sha..........................0.........0
smatrix......................0.........0
sql..........................0.........0
susan........................0.........0
=================================
CSESimplify
Category                   CSE    M2RCSE
adpcm........................0.........2
arm.........................19........21
basicmath....................6.........6
bh...........................0.........1
bitcount.....................1.........2
crc32........................0.........0
dijkstra.....................0.........0
em3d........................13........14
fft..........................0.........7
hanoi........................1.........1
hello........................0.........0
kmp..........................2.........2
l2lat........................0.........0
patricia.....................3.........7
qsort........................0.........1
sha..........................2.........2
smatrix......................0.........0
sql........................624.......778
```

```
susan.......................2.......16
================================
CSELdElim
Category                  CSE    M2RCSE
adpcm.......................1........0
arm........................31........1
basicmath..................14........1
bh.........................74........2
bitcount...................20........0
crc32.......................3........0
dijkstra....................3........2
em3d.......................21.......10
fft........................10........0
hanoi.......................4........0
hello.......................0........0
kmp........................20........0
l2lat.......................6........3
patricia...................24........3
qsort.......................3........0
sha........................32........0
smatrix....................24........5
sql......................4160......191
susan everything...........384.......25
================================
CSEStore2Load
Category                  CSE    M2RCSE
adpcm.......................0........0
arm.........................0........0
basicmath...................0........0
bh..........................0........0
bitcount....................0........0
crc32.......................0........0
dijkstra....................0........0
em3d........................0........0
fft.........................0........0
hanoi.......................0........0
hello.......................0........0
kmp.........................0........0
l2lat.......................0........0
patricia....................0........0
qsort.......................0........0
sha.........................0........0
smatrix.....................0........0
sql.........................0........0
susan.......................0........0
```

```
==================================
root@bbfa020f7cae:/ece566/build# /ece566/wolfbench/timing.py
Category                   .CSE   .M2RCSE
adpcm....................1.45......1.54
arm.......................0.0.......0.0
basicmath................0.09......0.07
bh........................1.8......0.78
bitcount.................0.26.......0.1
crc32....................0.08......0.08
dijkstra.................0.08......0.07
em3d......................0.5......0.34
fft......................0.06......0.04
hanoi....................2.86.......0.0
kmp......................0.14......0.11
l2lat....................0.04......0.03
patricia.................0.08......0.08
qsort....................0.04......0.04
sha......................0.02......0.01
smatrix..................3.86......3.68
sql......................0.01.......0.0
susan....................0.77......0.35
```

Q3

We can observe a significant reduction in load/store instructions in the run where Mem2Reg is run. Due to Load/Store reductions, the timing also goes down noticeably, especially in susan benchmark.

For CSE_RLoad, running memory 2 register promotion first reduces a lof of the opportunities that CSE_RLoad can optimize on. For sqlite, we see our optimization pass reduces 4153, but if we run Mem2Reg first, it takes care of most of that 4,000 load instructions and leaves only 134

CSEDead
It is actually worse to Mem2Reg before CSE_Dead since Mem2Reg may promote some of the dead code, which may mark them as having some use. Our Dead Code Elimination optimization checks for trivially dead code, so as long as there is a use, it won’t be considered. Again, look at SQLite as an example for this.(.272 vs 188 for M2R)

CSEStore2Load
(data collection error)

CSE_Basic

The number of reduced instructions goes as high as 8-10x with Mem2Reg turned on. Again, this is due to the Mem2Reg pass promoting load and stores to other types of operations that CSE_Basic can optimize on.

Q4

---

1. CSE_Dead: most applications do not contain that much dead code, except for sqlite. Since this is trivial dead code elimination, we do not expect a lot of reduction across applications
2. CSE_SImplify: We see a similar number of reductions as CSE_Dead. However, for sqlite there are a decent number of instructions (roughly800) that are simplified
3. CSE_RLoad: matrix, susan and especially sqlite are load heavy because they deal with a lot of data, so it makes sense that our optimization reduces a decent number of redundant loads.
4. CSE_Basic: The number of instructions optimized is proportionate to the size of the program. However, since we had trouble optimizing instructions within the same basic, we missed a lot of opportunities.
5. CSE_RStore: (no meaningful data)
6. CSE_Store2Load: (there is some error in collecting this metric)