Dynamic Cache Pre-Decompression

Project Milestone - November 29 2022

Group Members: Hassan Farooq (hfaroo9), Spenser Fong (sfong5), Timothy Vitkin (tvitkin2)

Progress

We have figured out how to set different compression algorithms provided by gem5 for simulation and modify how threads interact with instruction fetch in the simulator. This has resulted in varying performance results as shown in the experimental results section. We have also determined benchmarks that we will be running as well as statistics we are interested in when looking at the outputs from gem5. Compared to the timeline in our project proposal, we are about a week behind. With that being said, this upcoming week was scheduled as a buffer week and thus we are still confident that we'll be able to complete everything we said we would before the deadline.

Experimental results

For our testing, we ran this command with 2 threads:

```
build/RISCV/gem5.opt --outdir=m5out/riscv/powerful/FFT/
configs/example/se.py --cpu-type=RiscvO3CPU --cpu-clock=3GHz
--cacheline_size=64 --caches --l1i_size=32kB --l1i_assoc=8
--l1d_size=32kB --l1d_assoc=8 --num-cpus=2 --l2cache --l2_size=2MB
--l2_assoc=16 --l3cache --l3_size=6MB --l3_assoc=24
--cmd=benchmarks/riscv/FFT --options="-p2 -m16"
```

We ran using a compressed L3 as well as an uncompressed L3 cache to compare results. The algorithm that we used to compress the cache is the Base-Delta Immediate (BDI) algorithm. L3 cache was the only one we compressed as the majority of papers we had looked at only compressed LLC due to high latency and thus poor performance during the decompression process in higher level caches. As a next step, we may want to investigate what occurs when adding similar compression to other levels of cache and how performance is affected. It may also be interesting to see how various cache sizes and CPU clock speed may affect performance.

Appendix Table 1 shows the results of this run - we have limited it to only show values where we saw a greater than 50% difference between the compressed and uncompressed statistics.

Appendix Image 1 shows the generated schematic from the CPU with no compressed caches while Image 2 shows the generated schematic from the CPU with compressed L3 cache.

From our tests, the most interesting benchmark is the near 50% increase in L1 cache hits, and how core 0 of the CPU has a reduced stall cycles on instruction fetches. However, we are still investigating why core 1 of the CPU has an increased amount of stall cycles on instruction fetches. As expected, we are seeing a very high increase in memory-side hit rate (MSHR) metrics with our design.

Challenges Faced

We wanted to write to a thread state internally in our implementation, but gem5 does not allow writes to a thread state without flushing the entire pipeline, which is a problem for us. Our largest blocker is figuring out how to actually implement a design within gem5. Changing parameters in gem5 requires little code changes, and is easy to do, but we need to make sizable modifications to how saving and restoring threadstate works in the simulator. Since the codebase for the simulator is very large with lots of moving parts, it is hard to know if what we are doing is correct and safe.

Research challenges we're addressing now

We're still looking into dynamic cache pre-decompression and how we can offset some of the latency associated with decompression in compressed caches. Figuring out how to do this with gem5 has been fairly difficult so far due to how large the codebase is and little documentation with compressed caches within gem5. However, we have not changed our main research topic much from our initial project proposal.

Next steps

So far we have only run tests on RISC-V cores, we may want to also explore how compression affects x86 cores. We have also only looked at compression in L3 (LLC) cache and should look into what occurs when adding similar compression to other levels of cache and how performance is affected. Additional benchmarks would also be good to look at - we've only tried running the ones used in MP1 so far (LU, FFT, RADIX) and should look at some of the other benchmarks used by other cache compression papers such as the SPEC CPU 2017 suite. Different compression techniques would also be something we want to look into seeing that so far we have only tried using Base-Delta-Immediate (BDI) compression.

Appendix

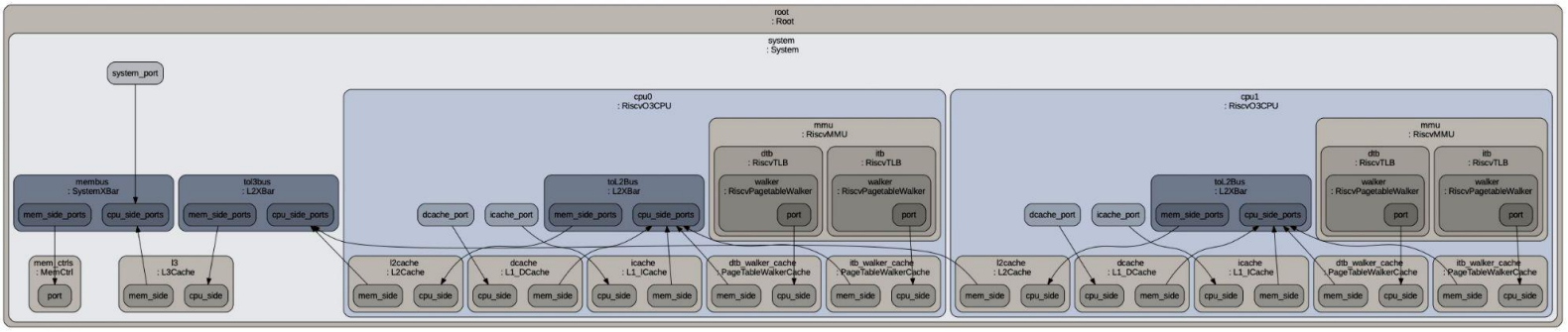
Table 1

Metric	Compressed	Uncompressed	Percent Difference
system.cpu0.fetch.icacheSquashes	3624743	13004	99.641
system.cpu1.fetch.icacheSquashes	1752532	10651	99.392
system.cpu1.icache.demandMshrHits::cpu1.inst	311	77	75.241
system.cpu1.icache.demandMshrHits::total	311	77	75.241
system.cpu1.icache.overallMshrHits::cpu1.inst	311	77	75.241
system.cpu1.icache.overallMshrHits::total	311	77	75.241
system.cpu1.icache.ReadReq.mshrHits::cpu1.inst	311	77	75.241
system.cpu1.icache.ReadReq.mshrHits::total	311	77	75.241
system.cpu0.fetch.icacheStallCycles	14501882	3680342	74.622
system.cpu1.fetch.icacheStallCycles	7001831	1782155	74.547
system.cpu0.icache.demandMshrHits::cpu0.inst	871	255	70.723
system.cpu0.icache.demandMshrHits::total	871	255	70.723
system.cpu0.icache.overallMshrHits::cpu0.inst	871	255	70.723
system.cpu0.icache.overallMshrHits::total	871	255	70.723
system.cpu0.icache.ReadReq.mshrHits::cpu0.inst	871	255	70.723
system.cpu0.icache.ReadReq.mshrHits::total	871	255	70.723
system.cpu0.l2cache.UpgradeReq.hits::cpu0.data	37	17	54.054
system.cpu0.l2cache.UpgradeReq.hits::total	37	17	54.054
system.cpu0.icache.demandMshrMissRate::cpu0.inst	9.50E-05	0.000191	50.262
system.cpu0.icache.demandMshrMissRate::t	9.50E-05	0.000191	50.262

otal			
system.cpu0.icache.overallMshrMissRate::cpu0.inst	9.50E-05	0.000191	50.262
system.cpu0.icache.overallMshrMissRate::total	9.50E-05	0.000191	50.262
system.cpu0.icache.ReadReq.mshrMissRate::cpu0.inst	9.50E-05	0.000191	50.262
system.cpu0.icache.ReadReq.mshrMissRate::total	9.50E-05	0.000191	50.262
system.cpu0.l2cache.InvalidateReq.hits::cpu0.data	2	1	50
system.cpu0.l2cache.InvalidateReq.hits::total	2	1	50
system.cpu1.l2cache.UpgradeReq.hits::cpu1.data	6	3	50
system.cpu1.l2cache.UpgradeReq.hits::total	6	3	50

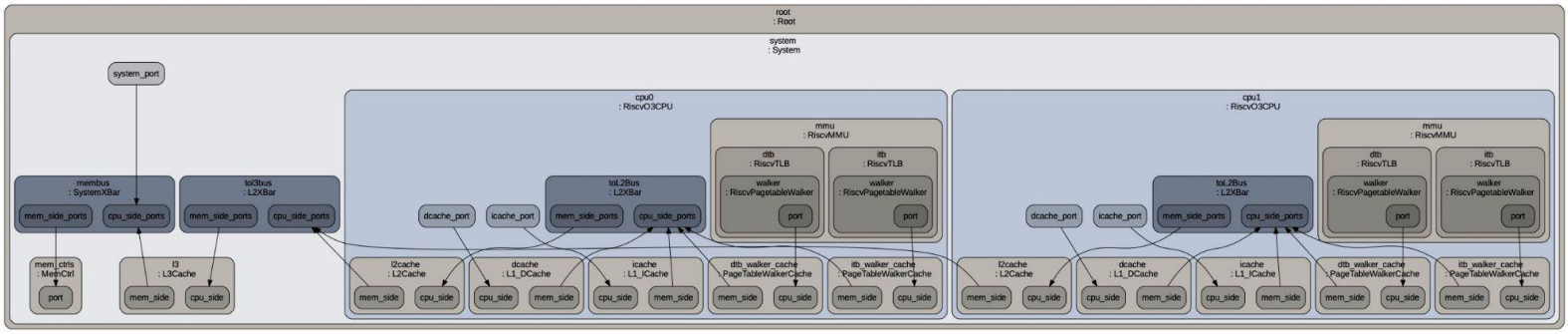
i

Image 1



Schematic of core with no compressed caches

Image 2



Schematic of core with compressed L3 cache