An Overview of CMMC (RISC-V Track)

Yingwei Zheng

Bingzhen Wang

Yifan Wu

Wenqian Yan

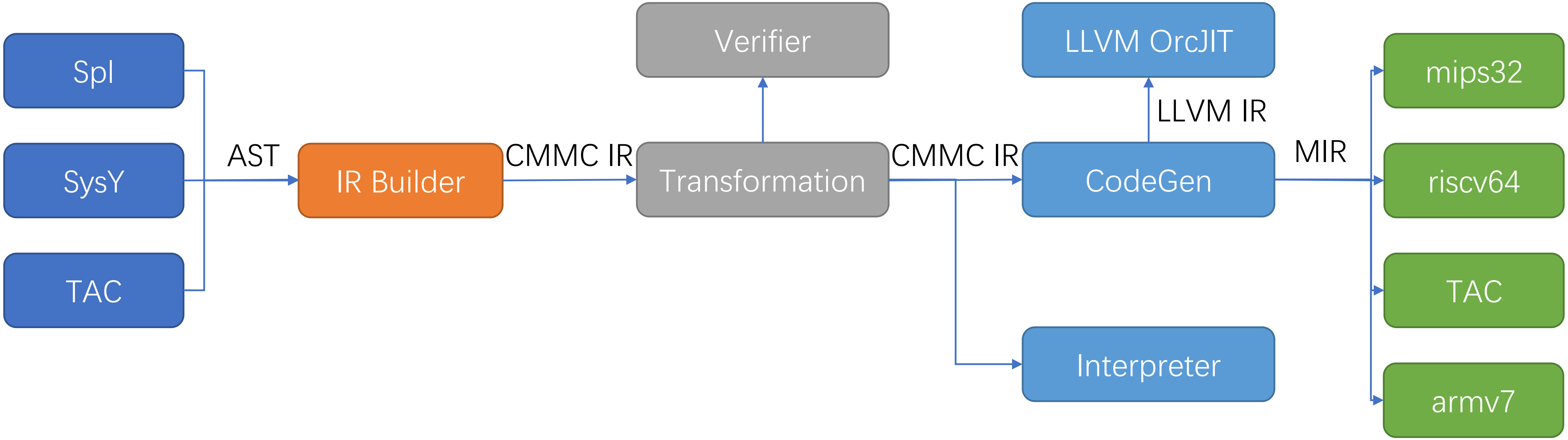
2023/8/22

Team members

- 郑英炜 @dtcxzyw
  - ISCAS PLCT Lab Intern, LLVM Committer (Transform/RISC-V backend)
  - Contribution: Infrastructure/Transform
- 王炳臻 @infiWang
  - ISCAS PLCT Lab Intern, luajit rv64 port contributor
  - Contribution: RISC-V backend
- 邬一帆 @GhostFrankWu
  - CTFer, COMPASS CTF Team Leader
  - Contribution: Regression testing/Fuzzing
- 严文谦 @YanWQ-monad
  - CTFer
  - Contribution: Transform/ARM-v7 backend/CI
- Instructor: 刘烨庞
  - SE
  - CS323 Compilers

Overview

- CMMC (C Minus Minus Compiler)

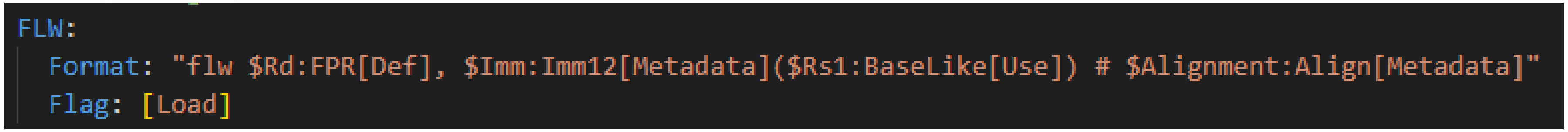


Highlights – Tiny ‘SMT’ solver

- Constraint solver
  - context-sensitive icmp: (a < b) ? (a < b) : (a >= b) → a < b
  - transitive closure: a < b && b < c → a < c
- Integer range analysis
  - signed range [min, max]
    - x is non-negative, srem x, 16 → urem x, 16 → and x, 15
    - x in [min1, max1], y in [min2, max2], max1 < min2 → x < y is true
    - n is non-negative, 0 s< x s< n → x u< n
  - known bits [known zeros, known ones]
    - (lshr x, 31) in [0, 1]
    - x is negative -> (ashr x, y) is negative

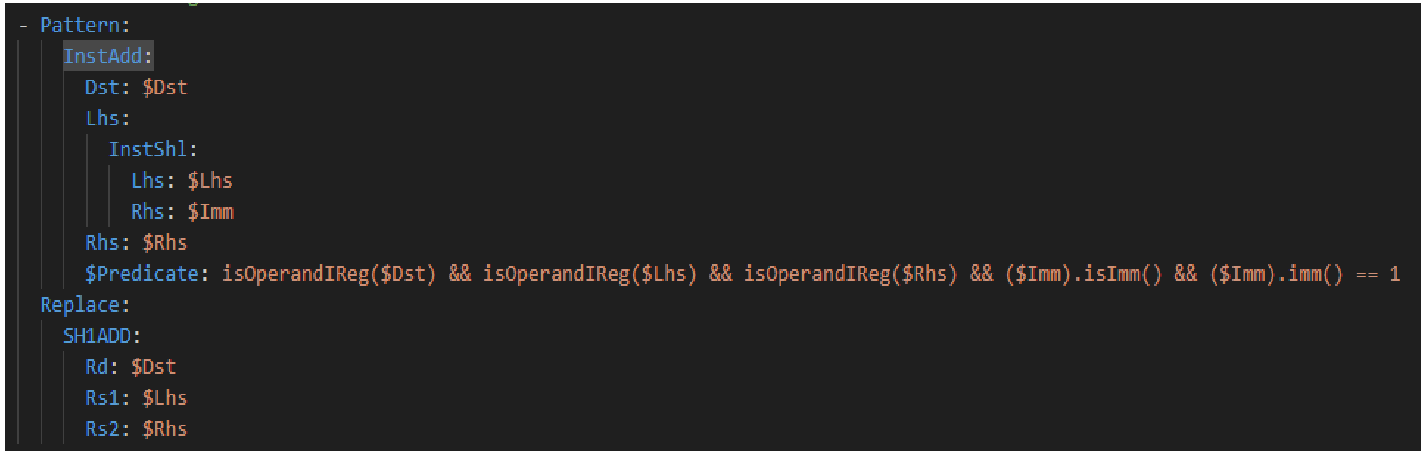
Highlights – Template code generator

- InstInfo

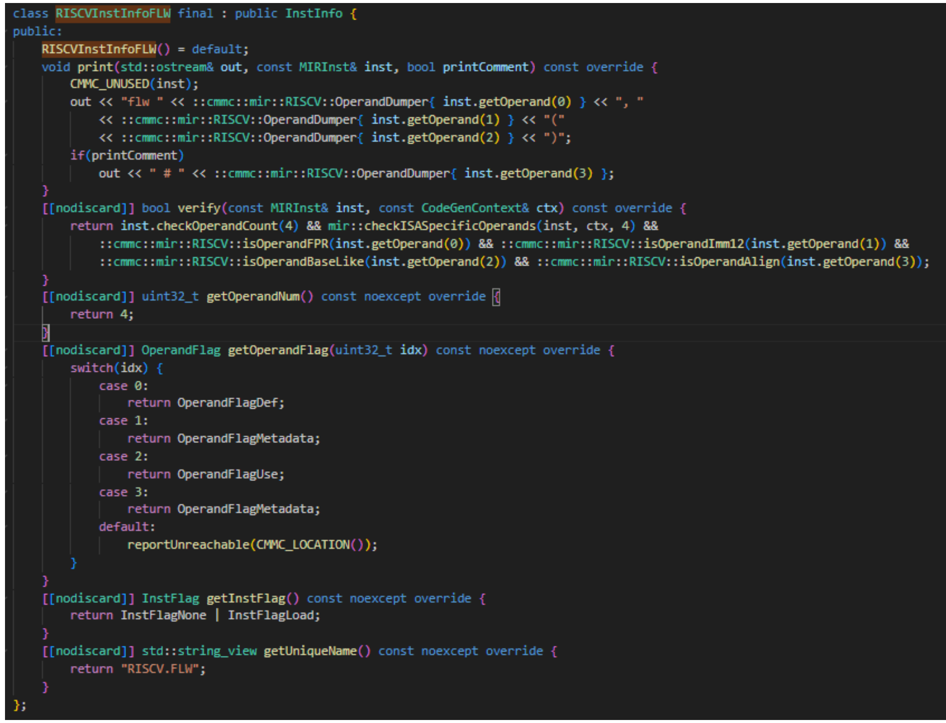


```
FLW:
  Format: "flw $Rd:FPR[Def], $Imm:Imm12[Metadata]($Rs1:BaseLike[Use]) # $Alignment:Align[Metadata]"
  Flag: [Load]
```

- InstSelection

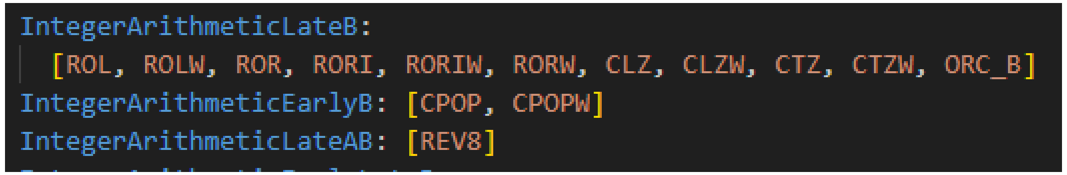


```
- Pattern:
    InstAdd:
      Dst: $Dst
      Lhs:
        InstShl:
          Lhs: $Lhs
          Rhs: $Imm
      Rhs: $Rhs
      $Predicate: isOperandIReg($Dst) && isOperandIReg($Lhs) && isOperandIReg($Rhs) && ($Imm).isImm() && ($Imm).imm() == 1
  Replace:
    SH1ADD:
      Rd: $Dst
      Rs1: $Lhs
      Rs2: $Rhs
```



```
class RISCVInstInfoFLW final : public InstInfo {
public:
    RISCVInstInfoFLW() = default;
    void print(std::ostream& out, const MIRInst& inst, bool printComment) const override {
        CMMC_UNUSED(inst);
        out << "flw " << ::cmmc::mir::RISCV::OperandDumper{ inst.getOperand(0) } << ", "
            << ::cmmc::mir::RISCV::OperandDumper{ inst.getOperand(1) } << "("
            << ::cmmc::mir::RISCV::OperandDumper{ inst.getOperand(2) } << ")";
        if(printComment)
            out << " # " << ::cmmc::mir::RISCV::OperandDumper{ inst.getOperand(3) };
    }
    [[nodiscard]] bool verify(const MIRInst& inst, const CodeGenContext& ctx) const override {
        return inst.checkOperandCount(4) && mir::checkISASpecificOperands(inst, ctx, 4) &&
            ::cmmc::mir::RISCV::isOperandFPR(inst.getOperand(0)) && ::cmmc::mir::RISCV::isOperandImm12(inst.getOperand(1)) &&
            ::cmmc::mir::RISCV::isOperandBaseLike(inst.getOperand(2)) && ::cmmc::mir::RISCV::isOperandAlign(inst.getOperand(3));
    }
    [[nodiscard]] uint32_t getOperandNum() const noexcept override {
        return 4;
    }
    [[nodiscard]] OperandFlag getOperandFlag(uint32_t idx) const noexcept override {
        switch(idx) {
            case 0:
                return OperandFlagDef;
            case 1:
                return OperandFlagMetadata;
            case 2:
                return OperandFlagUse;
            case 3:
                return OperandFlagMetadata;
            default:
                reportUnreachable(CMMC_LOCATION());
        }
    }
    [[nodiscard]] InstFlag getInstFlag() const noexcept override {
        return InstFlagNone | InstFlagLoad;
    }
    [[nodiscard]] std::string_view getUniqueName() const noexcept override {
        return "RISCV.FLW";
    }
};
```

- ScheduleModel



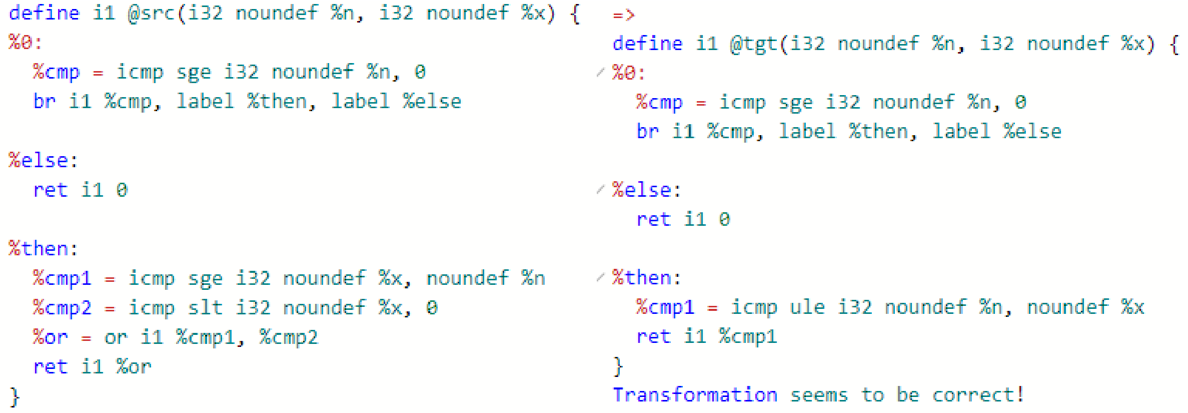
```
IntegerArithmeticLateB:
  [ROL, ROLW, ROR, RORI, RORIW, RORW, CLZ, CLZW, CTZ, CTZW, ORC_B]
IntegerArithmeticEarlyB: [CPOP, CPOPW]
IntegerArithmeticLateAB: [REV8]
```

Highlights – uArch-aware codegen

Optimization	Related feat/comp	Improvement
Top-down postRA schedule/Remove hazards	InOrder/Pipeline	2%
Latency-aware mul/sdiv/srem by constants	Pipeline	1%
Merge sw/lw	LD/ST	5-10%
Short forward branch opt	Pipeline	5-10%
Identical code folding	I-TLB, I-Cache	1%
Tail duplication	BPU	5-20%
Fused address generation	RISC-V Zba Ext	5%
Remove *W insts/sext.w	RISC-V C Ext	1%
Pettis-Hansen block layout	BPU, I-Cache, I-TLB	5-20%
Common base opt	Address generation	5%

Highlights – Compiler testing

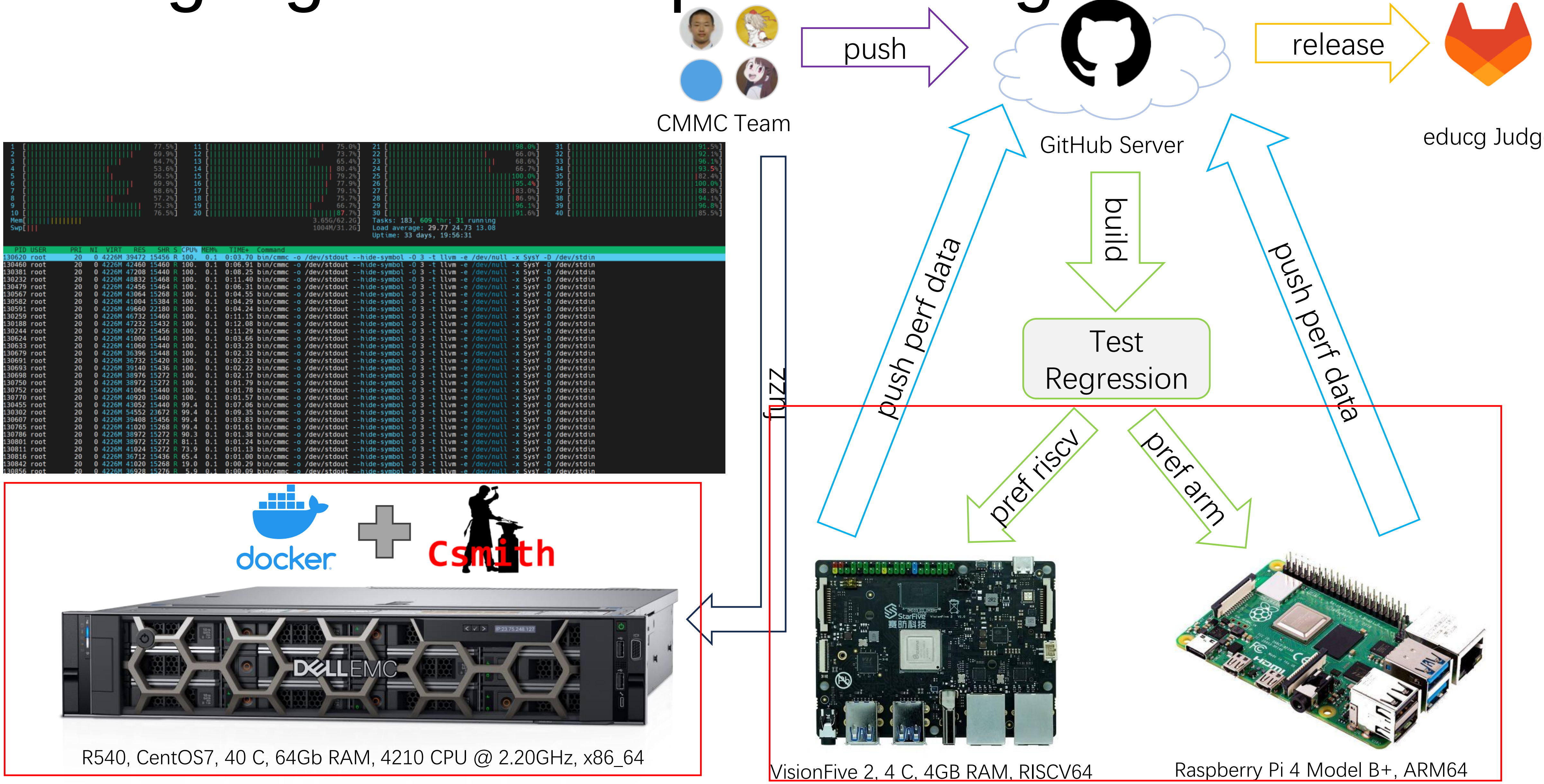
- ~~Functional – SysY2023~~
- Unit testing – IntegerRangeAnalysis
- Fuzzing – csmith: 1M testcases/day on a 40C machine
- Regression testing – 1k+ testcases (self-written/LLVM)
- Differential testing – GCC/LLVM
- Formal verification – Alive2 https://alive2.llvm.org/ce/z/uwy3Fg



```
define i1 @src(i32 noundef %n, i32 noundef %x) {
%0:
  %cmp = icmp sge i32 noundef %n, 0
  br i1 %cmp, label %then, label %else

%else:
  ret i1 0

%then:
  %cmp1 = icmp sge i32 noundef %x, noundef %n
  %cmp2 = icmp slt i32 noundef %x, 0
  %or = or i1 %cmp1, %cmp2
  ret i1 %or
}
```

```
=>
define i1 @tgt(i32 noundef %n, i32 noundef %x) {
%0:
  %cmp = icmp sge i32 noundef %n, 0
  br i1 %cmp, label %then, label %else

%else:
  ret i1 0

%then:
  %cmp1 = icmp ule i32 noundef %n, noundef %x
  ret i1 %cmp1
}
Transformation seems to be correct!
```

Highlights – Compiler testing



CMMC Team

push

GitHub Server

release

educg Judge

build

Test Regression

push perf data

push perf data

fuzz

pref riscv

pref arm

```
  1  [||||||||||||||||||||||||||||     77.5%]   11 [|||||||||||||||||||||||||||||     75.0%]   21 [||||||||||||||||||||||||||||||||98.0%]   31 [||||||||||||||||||||||||||||||||91.5%]
  2  [||||||||||||||||||||||||||       69.9%]   12 [|||||||||||||||||||||||||||       73.7%]   22 [||||||||||||||||||||||          66.0%]   32 [||||||||||||||||||||||||||||||||92.1%]
  3  [|||||||||||||||||||||||          64.7%]   13 [|||||||||||||||||||||||||         65.4%]   23 [|||||||||||||||||||||||         68.6%]   33 [||||||||||||||||||||||||||||||||96.1%]
  4  [||||||||||||||||||||             53.6%]   14 [||||||||||||||||||||||||||||||||  80.4%]   24 [||||||||||||||||||||||          66.7%]   34 [||||||||||||||||||||||||||||||||93.5%]
  5  [|||||||||||||||||||||            56.5%]   15 [||||||||||||||||||||||||||||||||  79.2%]   25 [||||||||||||||||||||||||||||||100.0%]   35 [||||||||||||||||||||||||||||||||82.4%]
  6  [||||||||||||||||||||||||||       69.9%]   16 [|||||||||||||||||||||||||||||||   77.9%]   26 [||||||||||||||||||||||||||||||||95.4%]   36 [||||||||||||||||||||||||||||||100.0%]
  7  [|||||||||||||||||||||||||        68.6%]   17 [|||||||||||||||||||||||||||||||   79.1%]   27 [||||||||||||||||||||||||||||||||83.0%]   37 [||||||||||||||||||||||||||||||||88.8%]
  8  [|||||||||||||||||||||            57.2%]   18 [||||||||||||||||||||||||||||||    75.7%]   28 [||||||||||||||||||||||||||||||||86.9%]   38 [||||||||||||||||||||||||||||||||94.1%]
  9  [||||||||||||||||||||||||||||     75.3%]   19 [|||||||||||||||||||||||||||       66.7%]   29 [||||||||||||||||||||||||||||||||96.1%]   39 [||||||||||||||||||||||||||||||||96.8%]
  10 [||||||||||||||||||||||||||||     76.5%]   20 [||||||||||||||||||||||||||||||||87.7%]   30 [||||||||||||||||||||||||||||||||91.6%]   40 [||||||||||||||||||||||||||||||||85.5%]
  Mem[||||||||||||||                                          3.65G/62.2G]   Tasks: 183, 609 thr; 31 running
  Swp[|||                                                     1004M/31.2G]   Load average: 29.77 24.73 13.08
                                                                             Uptime: 33 days, 19:56:31

   PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
130620 root       20   0 4226M 39472 15456 R 100.  0.1  0:03.70 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130460 root       20   0 4226M 42460 15460 R 100.  0.1  0:06.91 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130381 root       20   0 4226M 47208 15440 R 100.  0.1  0:08.25 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130232 root       20   0 4226M 48832 15468 R 100.  0.1  0:11.40 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130479 root       20   0 4226M 42456 15464 R 100.  0.1  0:06.31 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130567 root       20   0 4226M 43064 15268 R 100.  0.1  0:04.55 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130582 root       20   0 4226M 41004 15384 R 100.  0.1  0:04.29 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130591 root       20   0 4226M 49660 22180 R 100.  0.1  0:04.24 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130259 root       20   0 4226M 46732 15460 R 100.  0.1  0:11.15 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130188 root       20   0 4226M 47232 15432 R 100.  0.1  0:12.08 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130244 root       20   0 4226M 49272 15456 R 100.  0.1  0:11.29 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130624 root       20   0 4226M 41000 15440 R 100.  0.1  0:03.66 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130633 root       20   0 4226M 41060 15440 R 100.  0.1  0:03.23 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130679 root       20   0 4226M 36396 15448 R 100.  0.1  0:02.32 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130691 root       20   0 4226M 36732 15420 R 100.  0.1  0:02.23 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130693 root       20   0 4226M 39140 15436 R 100.  0.1  0:02.22 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130698 root       20   0 4226M 38976 15272 R 100.  0.1  0:02.17 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130750 root       20   0 4226M 38972 15272 R 100.  0.1  0:01.79 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130752 root       20   0 4226M 41064 15440 R 100.  0.1  0:01.78 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130770 root       20   0 4226M 40920 15400 R 100.  0.1  0:01.57 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130455 root       20   0 4226M 43052 15440 R 99.4  0.1  0:07.06 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130302 root       20   0 4226M 54552 23672 R 99.4  0.1  0:09.35 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130607 root       20   0 4226M 39408 15456 R 99.4  0.1  0:03.83 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130765 root       20   0 4226M 41020 15268 R 99.4  0.1  0:01.61 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130786 root       20   0 4226M 38972 15272 R 90.3  0.1  0:01.38 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130801 root       20   0 4226M 38972 15272 R 81.1  0.1  0:01.24 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130811 root       20   0 4226M 41024 15272 R 73.9  0.1  0:01.13 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130816 root       20   0 4226M 36712 15436 R 65.4  0.1  0:01.00 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130842 root       20   0 4226M 41020 15268 R 19.0  0.1  0:00.29 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
130856 root       20   0 4226M 36928 15276 R  5.9  0.1  0:00.09 bin/cmmc -o /dev/stdout --hide-symbol -O 3 -t llvm -e /dev/null -x SysY -D /dev/stdin
```

R540, CentOS7, 40 C, 64Gb RAM, 4210 CPU @ 2.20GHz, x86_64

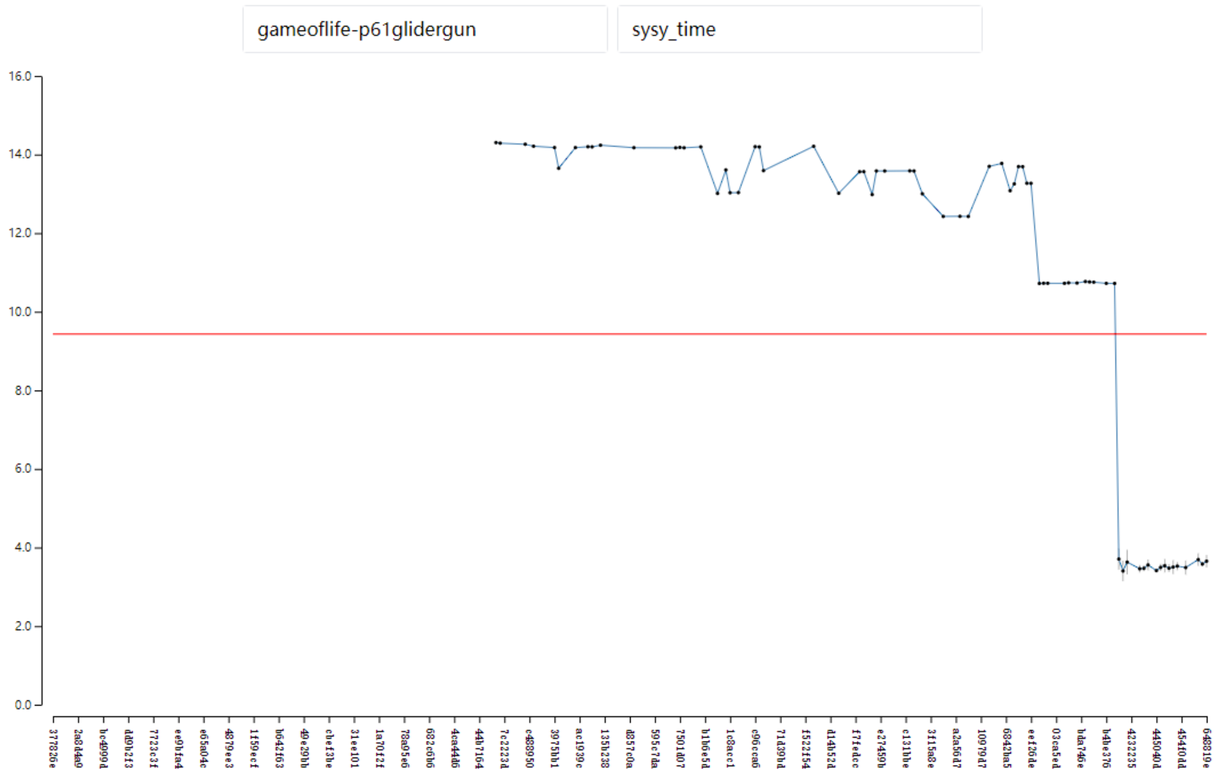
VisionFive 2, 4 C, 4GB RAM, RISCV64

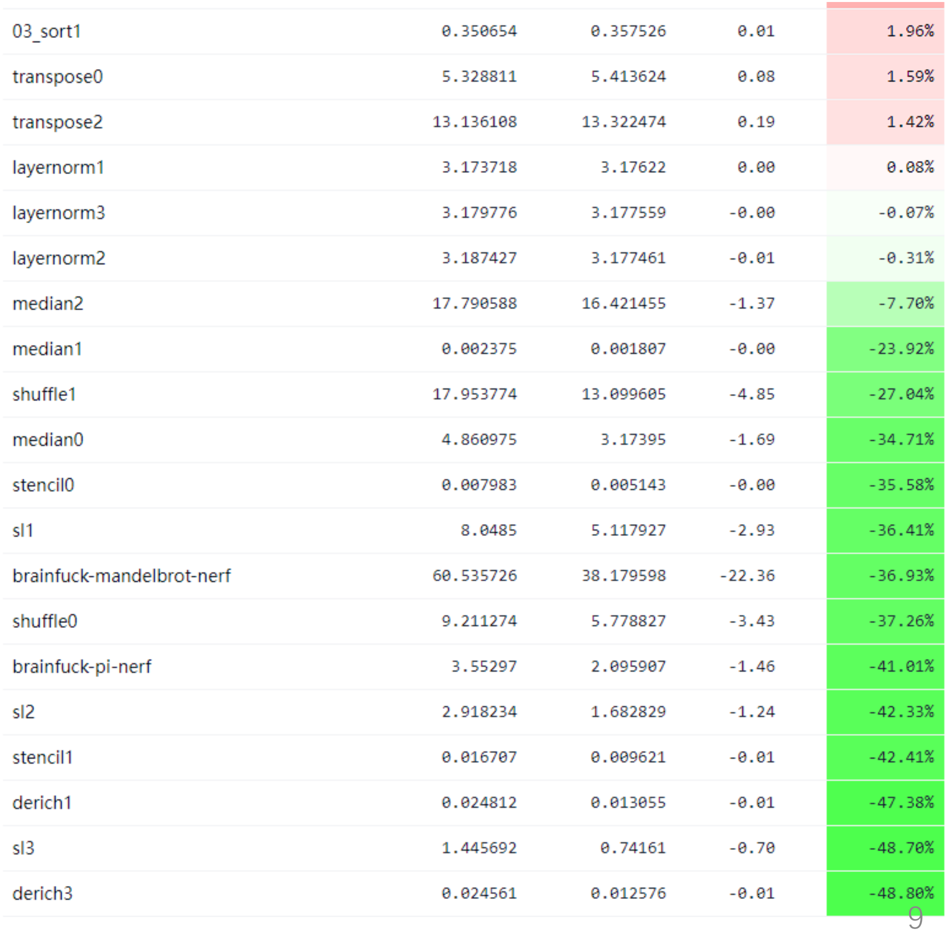
Raspberry Pi 4 Model B+, ARM64

Highlights – Continues perf monitoring

testcases

- https://dtcxzyw.github.io/cmmc/riscv/testcases
- https://dtcxzyw.github.io/cmmc/riscv/summary





03_sort1	0.350654	0.357526	0.01	1.96%
transpose0	5.328811	5.413624	0.08	1.59%
transpose2	13.136108	13.322474	0.19	1.42%
layernorm1	3.173718	3.17622	0.00	0.08%
layernorm3	3.179776	3.177559	-0.00	-0.07%
layernorm2	3.187427	3.177461	-0.01	-0.31%
median2	17.790588	16.421455	-1.37	-7.70%
median1	0.002375	0.001807	-0.00	-23.92%
shuffle1	17.953774	13.099605	-4.85	-27.04%
median0	4.860975	3.17395	-1.69	-34.71%
stencil0	0.007983	0.005143	-0.00	-35.58%
sl1	8.0485	5.117927	-2.93	-36.41%
brainfuck-mandelbrot-nerf	60.535726	38.179598	-22.36	-36.93%
shuffle0	9.211274	5.778827	-3.43	-37.26%
brainfuck-pi-nerf	3.55297	2.095907	-1.46	-41.01%
sl2	2.918234	1.682829	-1.24	-42.33%
stencil1	0.016707	0.009621	-0.01	-42.41%
derich1	0.024812	0.013055	-0.01	-47.38%
sl3	1.445692	0.74161	-0.70	-48.70%
derich3	0.024561	0.012576	-0.01	-48.80%

Summary

- Timeline: 2022/9/1 – 2023/8/21
- Codebase: hand-written ~57kLOC, generated ~49kLOC
- Source available on GitHub: https://github.com/dtcxzyw/cmmc/
- Some related LLVM patches:
  - ⚙ D150286 [RISCV] Fold (select setcc, setcc, setcc) into and/or instructions (llvm.org)
  - ⚙ D150862 [RISCV][CodeGenPrepare] Select the optimal base offset for GEPs with large offset (llvm.org)
  - ⚙ D155412 [ConstraintElim] Add facts implied by MinMaxIntrinsic (llvm.org)
  - ⚙ D156390 [SDAG][RISCV] Avoid expanding is-power-of-2 pattern on riscv32/64 with zbb (llvm.org)