Alibaba Cloud

Worldwide Cloud Services Partner

OpenJDK Work Update

2021-08-24

Alibaba JVM Team

Agenda

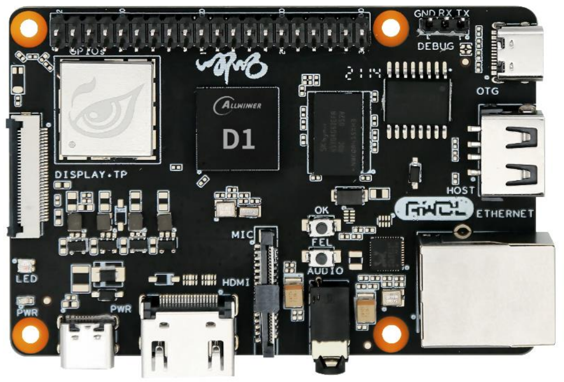
Alibaba Cloud | Worldwide Cloud Services Partner

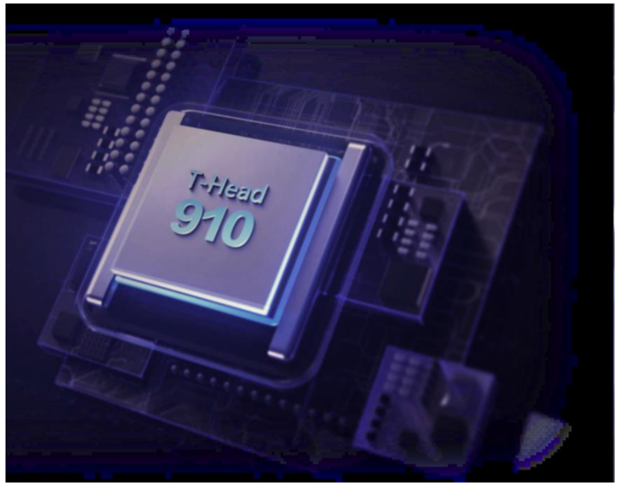
- OpenJDK Work Update
- Cross Architecture Performance Evaluation

OpenJDK Work Update

Alibaba Cloud | Worldwide Cloud Services Partner

- New OpenJDK repo has been created under RVI for community collaboration
  - https://github.com/riscv/riscv-openjdk
  - Sync with upstream OpenJDK sandbox periodically
    - https://github.com/openjdk/jdk-sandbox/tree/riscv-port-branch
- Nightly test build based on QEMU
- FVT/SVT/PVT on C910 hardware
- Trivial test on D1 hardware





Cross Architecture Performance Evaluation

Alibaba Cloud | Worldwide Cloud Services Partner

- Benchmark suites for the performance comparison between different CPU architectures
  - JMH(Java Microbenchmark Harness) based
  - Chosen from OpenJDK upstream and Alibaba internal tests
    - Memory/branch/String/Thread/Serialization
  - Find optimization opportunities for RISC-V

Hardware Configuration

Alibaba Cloud
Worldwide Cloud Services Partner

RISC-V 64 board(RISCV-64)

- Debian 64bit System
- 2 cores, 1.2GHz
- 4G RAM

VS

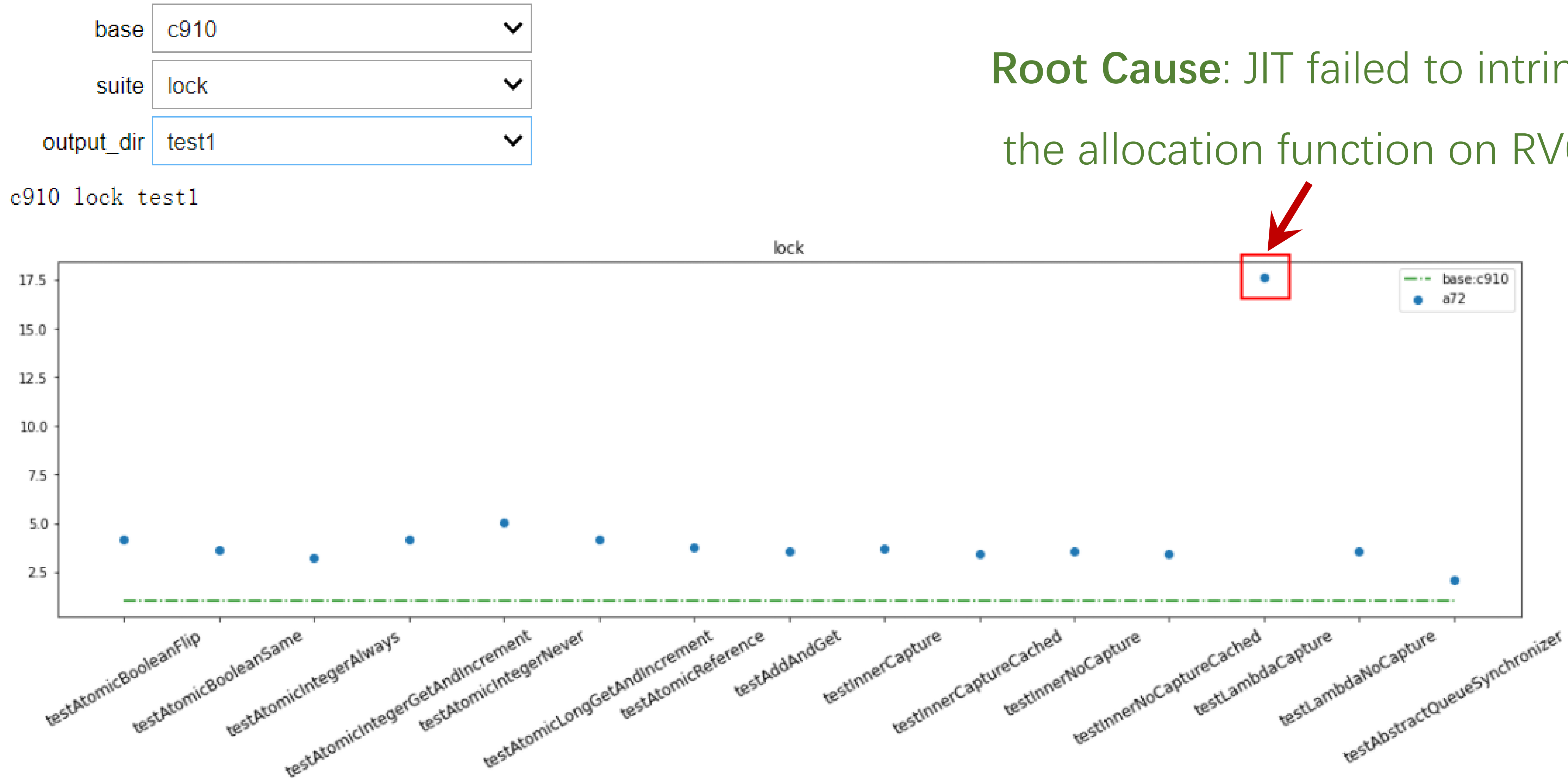
Raspberrypi 4B board(AArch64)

- Ubuntu 19.02 64bit System
- 4 cores, 1.5GHz
- 4G RAM

Case Study I

Alibaba Cloud | Worldwide Cloud Services Partner

Root Cause: JIT failed to intrinsic the allocation function on RV64

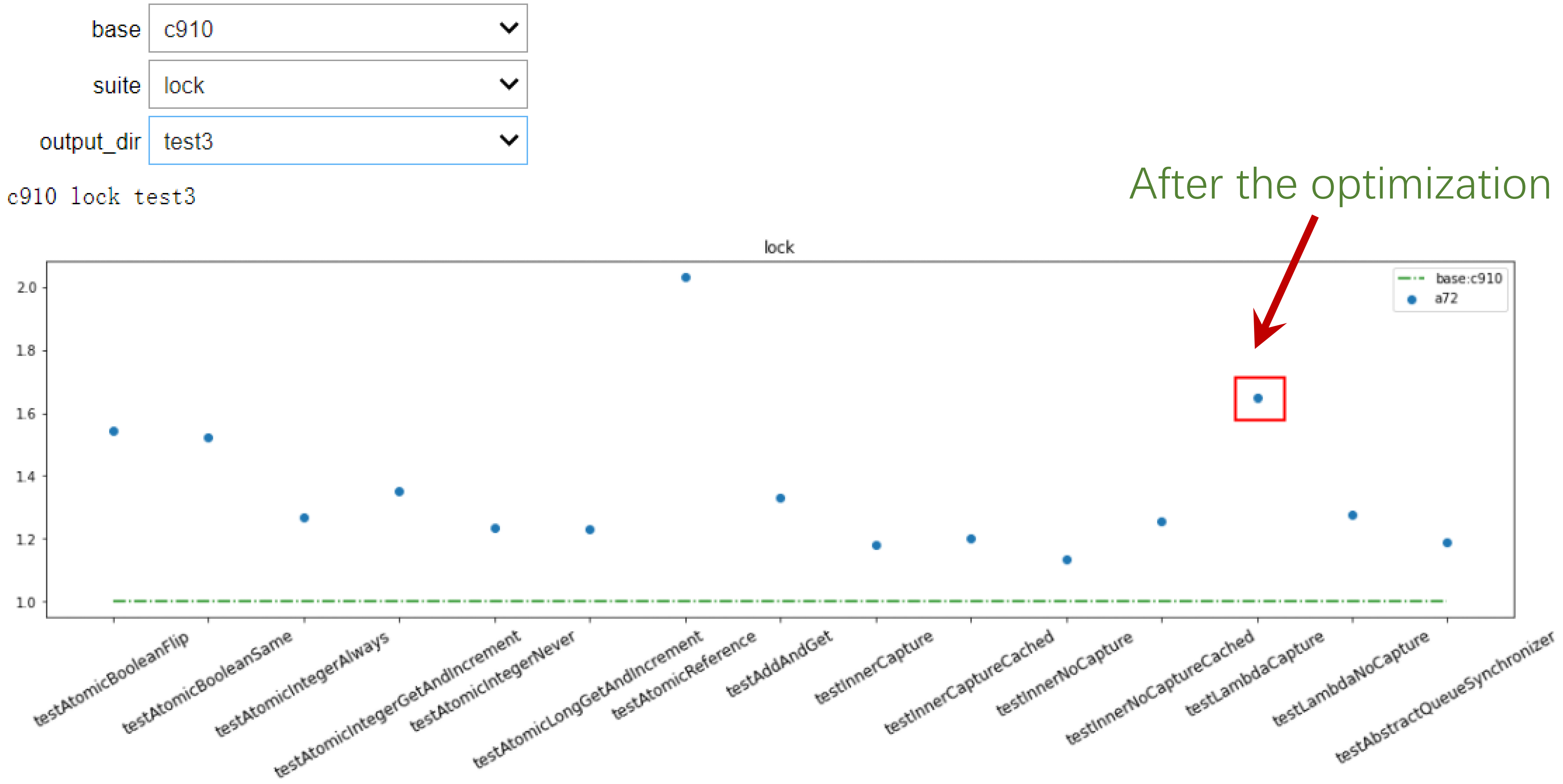


base: c910
suite: lock
output_dir: test1

```
c910 lock test1
```

Case Study I(cont.)

Alibaba Cloud | Worldwide Cloud Services Partner



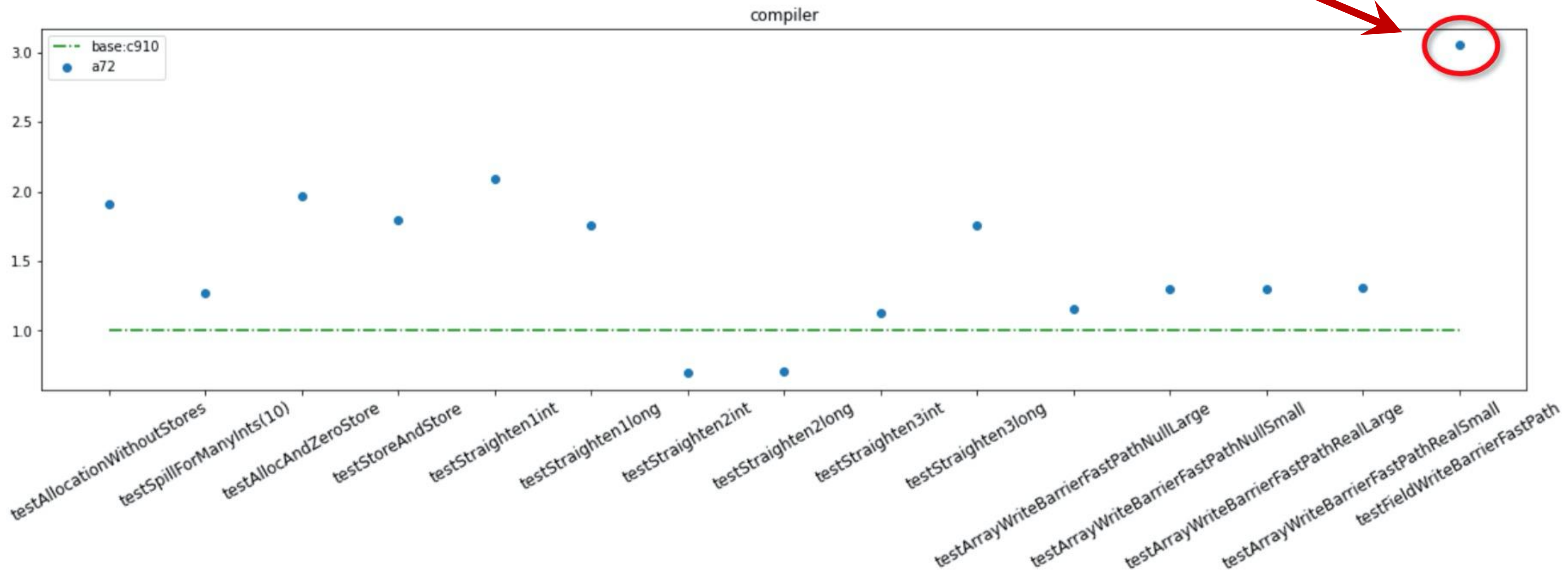
base: c910
suite: lock
output_dir: test3

```
c910 lock test3
```

Case Study II

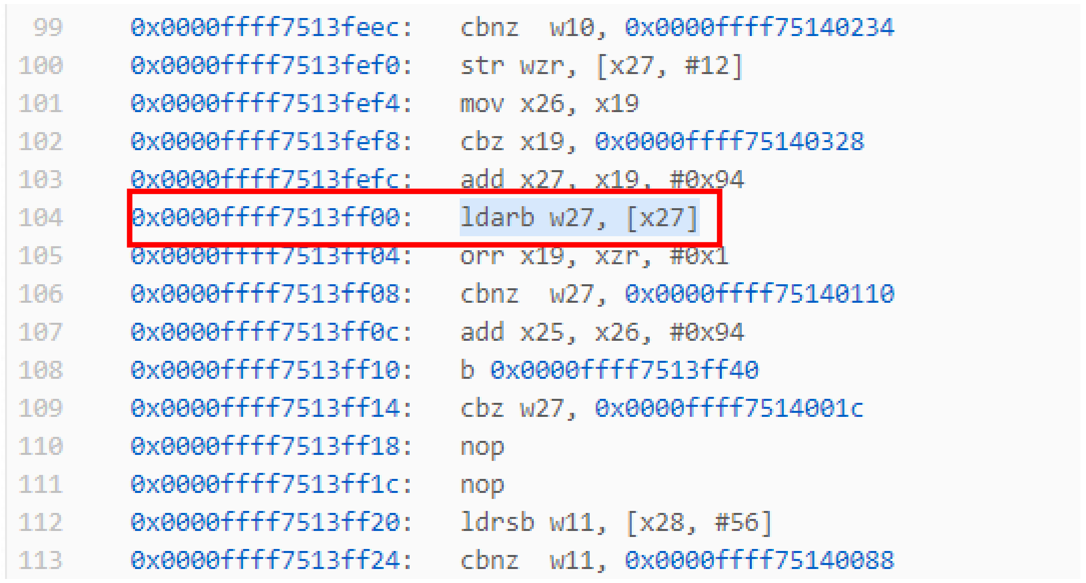
Alibaba Cloud | Worldwide Cloud Services Partner

Significant overhead caused by fence instruction on RV64



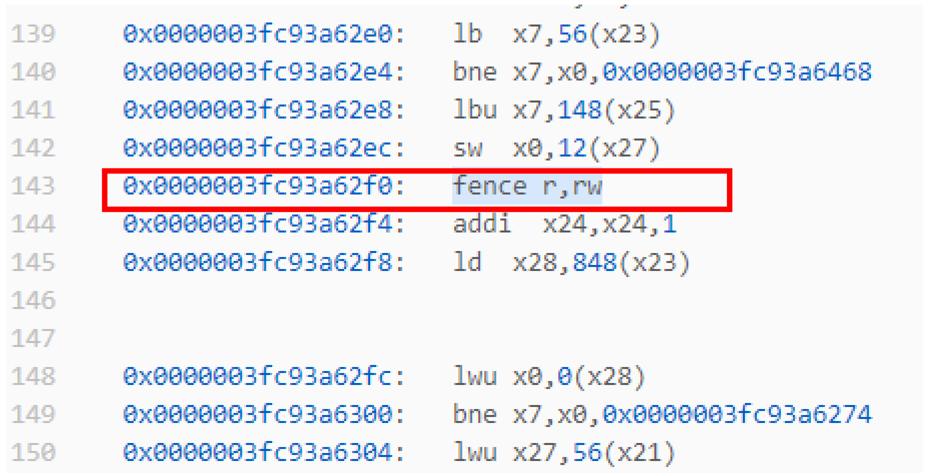
Volatile Access Overhead

ARM A72



```
0x0000ffff7513feec:   cbnz  w10, 0x0000ffff75140234
0x0000ffff7513fef0:   str wzr, [x27, #12]
0x0000ffff7513fef4:   mov x26, x19
0x0000ffff7513fef8:   cbz x19, 0x0000ffff75140328
0x0000ffff7513fefc:   add x27, x19, #0x94
0x0000ffff7513ff00:   ldarb w27, [x27]
0x0000ffff7513ff04:   orr x19, xzr, #0x1
0x0000ffff7513ff08:   cbnz  w27, 0x0000ffff75140110
0x0000ffff7513ff0c:   add x25, x26, #0x94
0x0000ffff7513ff10:   b 0x0000ffff7513ff40
0x0000ffff7513ff14:   cbz w27, 0x0000ffff7514001c
0x0000ffff7513ff18:   nop
0x0000ffff7513ff1c:   nop
0x0000ffff7513ff20:   ldrsb w11, [x28, #56]
0x0000ffff7513ff24:   cbnz  w11, 0x0000ffff75140088
```

RISCV-64

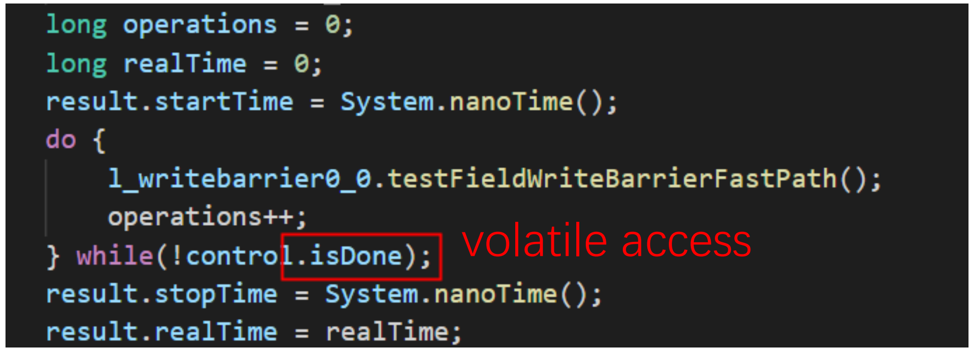


```
0x0000003fc93a62e0:   lb  x7,56(x23)
0x0000003fc93a62e4:   bne x7,x0,0x0000003fc93a6468
0x0000003fc93a62e8:   lbu x7,148(x25)
0x0000003fc93a62ec:   sw  x0,12(x27)
0x0000003fc93a62f0:   fence r,rw
0x0000003fc93a62f4:   addi  x24,x24,1
0x0000003fc93a62f8:   ld  x28,848(x23)


0x0000003fc93a62fc:   lwu x0,0(x28)
0x0000003fc93a6300:   bne x7,x0,0x0000003fc93a6274
0x0000003fc93a6304:   lwu x27,56(x21)
```

- If we remove these barriers, RV64 has similar score to ARM A72
- FENCE in RV64 incurs more penalty than ARM

Java snippet code



```
long operations = 0;
long realTime = 0;
result.startTime = System.nanoTime();
do {
    l_writebarrier0_0.testFieldWriteBarrierFastPath();
    operations++;
} while(!control.isDone());
result.stopTime = System.nanoTime();
result.realTime = realTime;
```

Future Work

- Add more test suites
  - Vector/Crypto/Stream …
- Test on more hardware(D1…)
- Plan to open source and contribute to:
  - https://github.com/riscv/riscv-openjdk

Alibaba Cloud

Worldwide Cloud Services Partner

Thanks