Lab1 O3CPU参数对性能的影响

1. 实验环境与准备

使用docker+Vscode的方式部署gem5，基本流程按照教程文档即可，下面是遇到的一些问题以及解决方案。

1. 容器内拉取时可能出现代理被错误设置，需要关闭代理

```
git config --global --unset http.proxy
git config --global --unset https.proxy
```

2. 进行验证安装正确的测试命令，直接复制存在问题，符号 - 会缺失。

```
cd ~/gem5
build/RISCV/gem5.opt configs/deprecated/example/se.py  -c tests/test-progs/hello/bin/riscv/linu
```

验证安装结果如下，可以看到正确输出了hello world!，即gem5正确安装。

```
root@86bcef818dae:/lab1# cd ~/gem5
root@86bcef818dae:~/gem5# build/RISCV/gem5.opt configs/deprecated/example/se.py  -c tests/test-progs/hello/bin/riscv/linux/hello
gem5 Simulator System.  https://www.gem5.org
gem5 is copyrighted software; use the --copyright option for details.

gem5 version 25.0.0.1
gem5 compiled Oct 16 2025 13:09:23
gem5 started Oct 20 2025 11:38:20
gem5 executing on 86bcef818dae, pid 17435
command line: build/RISCV/gem5.opt configs/deprecated/example/se.py -c tests/test-progs/hello/bin/riscv/linux/hello

warn: The se.py script is deprecated. It will be removed in future releases of  gem5.
Global frequency set at 1000000000000 ticks per second
src/mem/dram_interface.cc:690: warn: DRAM device capacity (8192 Mbytes) does not match the address range assigned (512 Mbytes)
src/arch/riscv/isa.cc:319: info: RVV enabled, VLEN = 256 bits, ELEN = 64 bits
src/base/statistics.hh:279: warn: One of the stats is a legacy stat. Legacy stat is a stat that does not belong to any statistics::Group. Legacy stat is deprecated.
system.remote_gdb: Listening for connections on port 7000
**** REAL SIMULATION ****
src/sim/mem_state.cc:443: info: Increasing stack size by one page.
Hello world!
Exiting @ tick 3306500 because exiting with last active thread context
```

2. 编译程序

```
sudo apt update
sudo apt install -y g++-riscv64-linux-gnu
riscv64-linux-gnu-g++ --static -O2 -o /lab1/daxpy /lab1/daxpy.cpp
chmod +x /lab1/daxpy
```

结果如下，同时可以看到生成了编译好的文件daxpy

```
root@86bcef818dae:~/gem5# cd /lab1
root@86bcef818dae:/lab1# riscv64-linux-gnu-g++ --static -O2 -o /lab1/daxpy /lab1/daxpy.cpp
root@86bcef818dae:/lab1# chmod +x /lab1/daxpy
```

3. 单次运行

在运行 `O3CPU.py` 时默认启动gdb调试，导致进程等待，而无法直接快速生成最终的 `stat.txt` 文件。故针对 `O3CPU.py` 进行一定的修改，默认使其不打开gdb，如果打开了将其关闭并回到原来的状态。

```
if hasattr(system.cpu, "wait_for_remote_gdb"):
    system.cpu.wait_for_remote_gdb = False
if hasattr(system, "workload") and hasattr(system.workload, "wait_for_remote_gdb"):
    system.workload.wait_for_remote_gdb = False

try:
    if hasattr(system, "remote_gdb"):
        delattr(system, "remote_gdb")
except Exception:
    pass

def add_options(parser):
    parser.add_argument("-c", "--cmd", required=True, help="The binary to run.")
    parser.add_argument("--num-rob-entries", type=int, default=192)
    parser.add_argument("--num-iq-entries", type=int, default=64)
    parser.add_argument("--num-phys-int-regs", type=int, default=256)

    # 防止gdb卡住的
    parser.add_argument("--gdb-port", type=int, default=0,
                        help="Enable remote GDB on this port (0 = disable).")
    parser.add_argument("--gdb-wait", action="store_true",
                        help="Wait for GDB to attach before running.")
```

新增两个参数，默认均设置为 `0` 和 `false` ，不使用gdb，运行命令如下：

```
/root/gem5/build/RISCV/gem5.opt \
  --remote-gdb-port=0 --listener-mode=off \
  --outdir=/lab1/m5out_try \
  /lab1/O3CPU.py \
  -c /lab1/daxpy \
  --num-phys-int-regs=256 \
  --num-iq-entries=64 \
  --num-rob-entries=192
```

结果如下：

```
问题  输出  调试控制台  终端  端口 4

root@86bcef818dae:/lab1# /root/gem5/build/RISCV/gem5.opt \
>   --remote-gdb-port=0 --listener-mode=off \
>   --outdir=/lab1/m5out_try \
>   /lab1/O3CPU.py \
>   -c /lab1/daxpy \ …
src/sim/mem_state.cc:443: info: Increasing stack size by one page.
src/sim/mem_state.cc:443: info: Increasing stack size by one page.
src/sim/mem_state.cc:443: info: Increasing stack size by one page.
src/sim/mem_state.cc:443: info: Increasing stack size by one page.
src/sim/mem_state.cc:443: info: Increasing stack size by one page.
src/sim/mem_state.cc:443: info: Increasing stack size by one page.
src/sim/mem_state.cc:443: info: Increasing stack size by one page.
src/sim/mem_state.cc:443: info: Increasing stack size by one page.
src/sim/mem_state.cc:443: info: Increasing stack size by one page.
src/sim/mem_state.cc:443: info: Increasing stack size by one page.
src/sim/mem_state.cc:443: info: Increasing stack size by one page.
src/sim/mem_state.cc:443: info: Increasing stack size by one page.
src/sim/mem_state.cc:443: info: Increasing stack size by one page.
src/sim/mem_state.cc:443: info: Increasing stack size by one page.
src/sim/mem_state.cc:443: info: Increasing stack size by one page.
src/sim/mem_state.cc:443: info: Increasing stack size by one page.
src/sim/mem_state.cc:443: info: Increasing stack size by one page.
src/sim/mem_state.cc:443: info: Increasing stack size by one page.
src/sim/mem_state.cc:443: info: Increasing stack size by one page.
src/sim/mem_state.cc:443: info: Increasing stack size by one page.
225121.084433
Exit @ tick 6642614000 because exiting with last active thread context
root@86bcef818dae:/lab1#
```

关键输出

fullRegistersEvents（381.9 万）数量级远大于 IQFullEvents（4.8 万）和 ROBFullEvents（470）。这说明 Rename 最常因为“没空闲物理寄存器”而被迫阻塞，而 IQ/ROB 满只是次要现象

4. 批次运行

由于三个参数参数给出了3，4，4种，加起来有48种需要运行的，故为了方便，考虑使用批次处理。通过使用ai以及上网查找资料，构建如下的脚本：

```bash
#!/usr/bin/env bash
set -euo pipefail

# gem5 可执行 + 关掉所有 GDB 监听的全局开关
GEM5="/root/gem5/build/RISCV/gem5.opt --remote-gdb-port=0 --listener-mode=off"
CFG=/lab1/O3CPU.py
BIN=/lab1/daxpy

OUTROOT=/lab1/sweep_out
mkdir -p "$OUTROOT"
echo "PR,IQ,ROB,numCycles,committedInsts,IPC,ROBFull,IQFull,RegFull" > "$OUTROOT/summary.csv"

# 想先小跑验证？把数组缩小些再跑
PRS=(64 256 1024)
IQS=(4 16 64 256)
ROBS=(4 16 64 256)

for PR in "${PRS[@]}"; do
  for IQ in "${IQS[@]}"; do
    for ROB in "${ROBS[@]}"; do
      OUTDIR="${OUTROOT}/pr${PR}_iq${IQ}_rob${ROB}"
      echo "==> PR=$PR IQ=$IQ ROB=$ROB"
      # 运行仿真
      eval $GEM5 --outdir="$OUTDIR" "$CFG" -c "$BIN" \
        --num-phys-int-regs=$PR --num-iq-entries=$IQ --num-rob-entries=$ROB

      STATS="$OUTDIR/stats.txt"
      CYC=$(awk '/^system\.cpu\.numCycles/{print $2}'                          "$STATS")
      CI=$(awk  '/^system\.cpu\..*committed.*Insts/{print $2}'                 "$STATS")
      ROBF=$(awk '/^system\.cpu\..*(ROB).*Full.*Events/{print $2}'             "$STATS")
      IQF=$(awk  '/^system\.cpu\..*(IQ).*Full.*Events/{print $2}'              "$STATS")
      REGF=$(awk '/^system\.cpu\..*(Reg|Registers).*Full.*Events/{print $2}'   "$STATS")
      IPC=$(awk -v ci="${CI:-0}" -v cy="${CYC:-1}" 'BEGIN{printf "%.6f", (cy==0?0:ci/cy)}')

      echo "${PR},${IQ},${ROB},${CYC},${CI},${IPC},${ROBF},${IQF},${REGF}" >> "$OUTROOT/summary
    done
  done
done
echo "Done -> $OUTROOT/summary.csv"
```

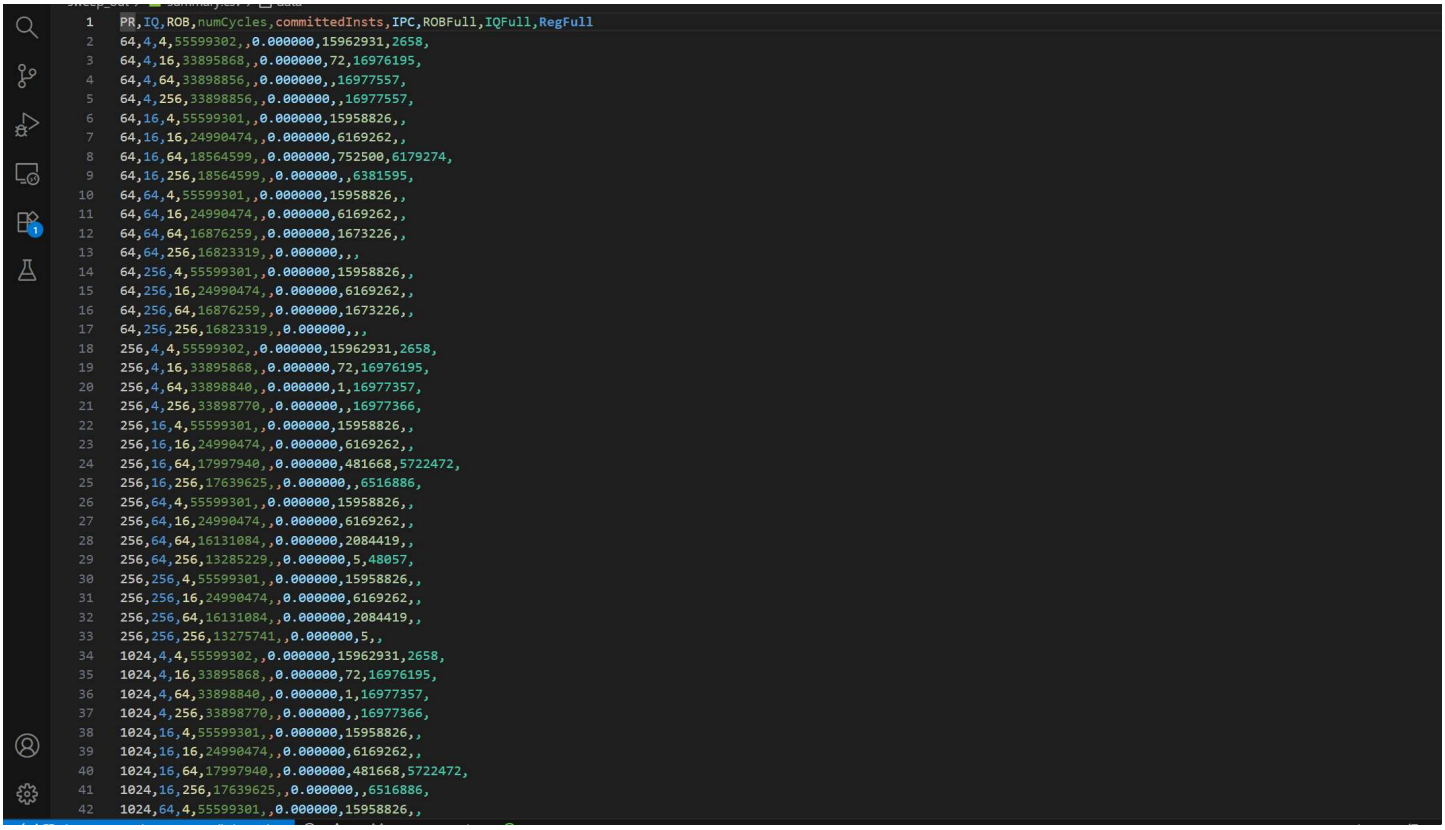
参数名称	说明	取值范围
num-phys-int-regs	物理整数寄存器数目	64, 256, 1024
num-iq-entries	IQ条目数	4, 16, 64, 256
num-rob-entries	ROB条目数	4, 16, 64, 256

注意该脚本需要设置关闭gdb否则会出席那卡住无法进行的情况，最终结果保存在 `sweep/out` 下面，得到每个组合的运行结果，以及关键指标，指标如下

```
numCycles         ← ^system\.cpu\.numCycles
committedInsts    ← ^system\.cpu\..*committed.*Insts
IPC               ← committedInsts / numCycles
ROB/IQ/Reg Full   ← ^system\.cpu\..*(ROB|IQ|Reg).*Full.*Events
```

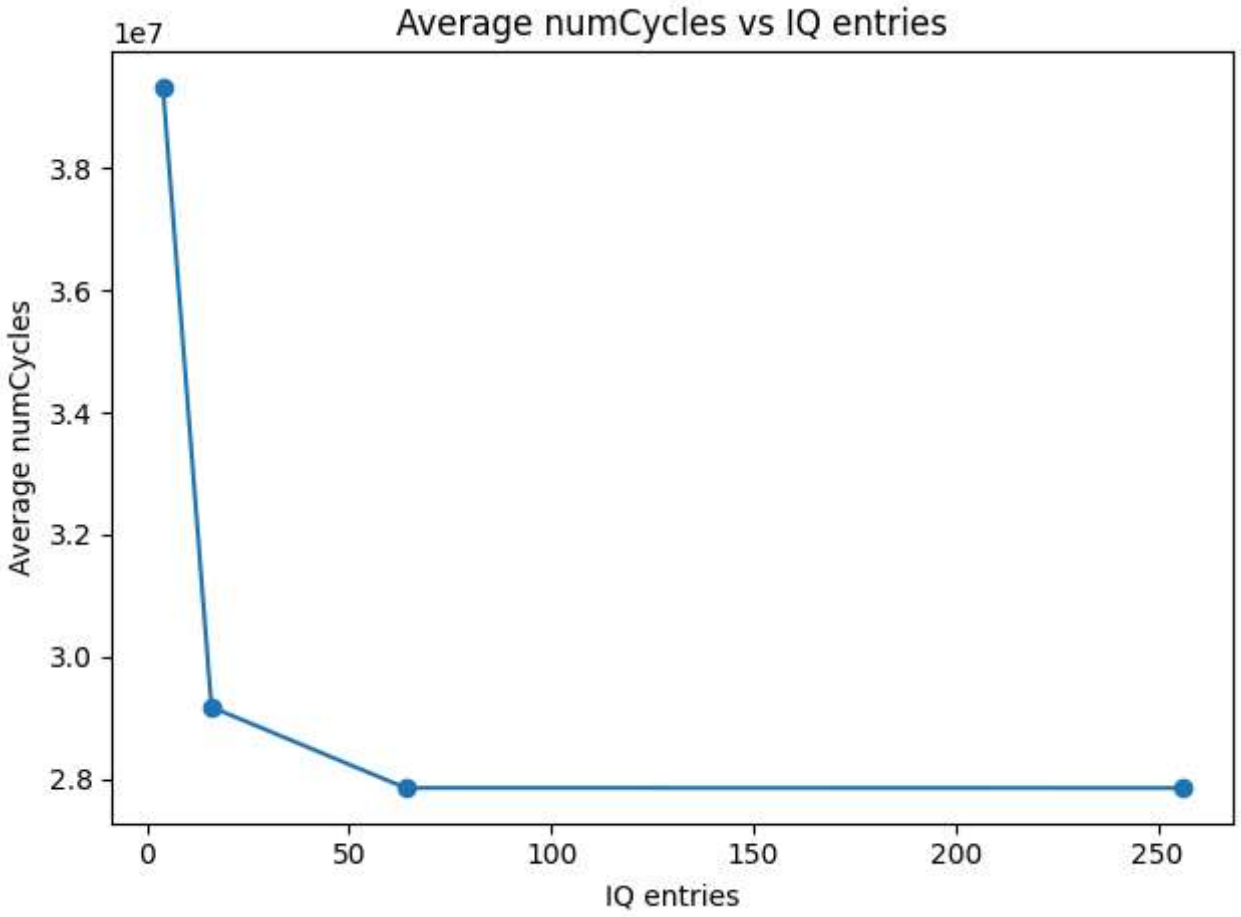
5. 结果分析

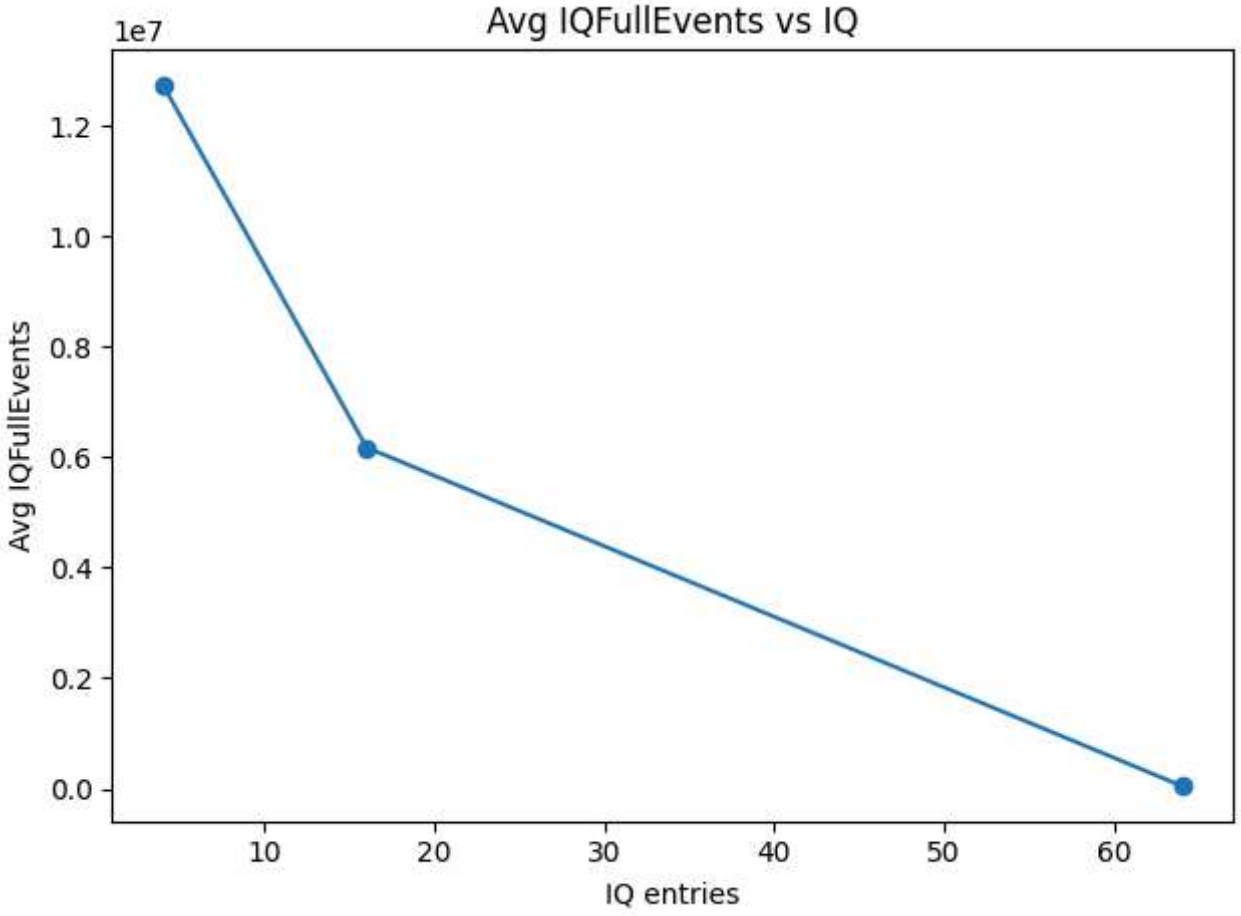
结果如 `sweep_out.csv` 文件所示，部分截图如下：



```
PR,IQ,ROB,numCycles,committedInsts,IPC,ROBFull,IQFull,RegFull
64,4,4,55599302,,0.000000,15962931,2658,
64,4,16,33895868,,0.000000,72,16976195,
64,4,64,33898856,,0.000000,,16977557,
64,4,256,33898856,,0.000000,,16977557,
64,16,4,55599301,,0.000000,15958826,,
64,16,16,24990474,,0.000000,6169262,,
64,16,64,18564599,,0.000000,752500,6179274,
64,16,256,18564599,,0.000000,,6381595,
64,64,4,55599301,,0.000000,15958826,,
64,64,16,24990474,,0.000000,6169262,,
64,64,64,16876259,,0.000000,1673226,,
64,64,256,16823319,,0.000000,,,
64,256,4,55599301,,0.000000,15958826,,
64,256,16,24990474,,0.000000,6169262,,
64,256,64,16876259,,0.000000,1673226,,
64,256,256,16823319,,0.000000,,,
256,4,4,55599302,,0.000000,15962931,2658,
256,4,16,33895868,,0.000000,72,16976195,
256,4,64,33898840,,0.000000,1,16977357,
256,4,256,33898770,,0.000000,,16977366,
256,16,4,55599301,,0.000000,15958826,,
256,16,16,24990474,,0.000000,6169262,,
256,16,64,17997940,,0.000000,481668,5722472,
256,16,256,17639625,,0.000000,,6516886,
256,64,4,55599301,,0.000000,15958826,,
256,64,16,24990474,,0.000000,6169262,,
256,64,64,16131084,,0.000000,2084419,,
256,64,256,13285229,,0.000000,5,48057,
256,256,4,55599301,,0.000000,15958826,,
256,256,16,24990474,,0.000000,6169262,,
256,256,64,16131084,,0.000000,2084419,,
256,256,256,13275741,,0.000000,5,,
1024,4,4,55599302,,0.000000,15962931,2658,
1024,4,16,33895868,,0.000000,72,16976195,
1024,4,64,33898840,,0.000000,1,16977357,
1024,4,256,33898770,,0.000000,,16977366,
1024,16,4,55599301,,0.000000,15958826,,
1024,16,16,24990474,,0.000000,6169262,,
1024,16,64,17997940,,0.000000,481668,5722472,
1024,16,256,17639625,,0.000000,,6516886,
1024,64,4,55599301,,0.000000,15958826,,
```

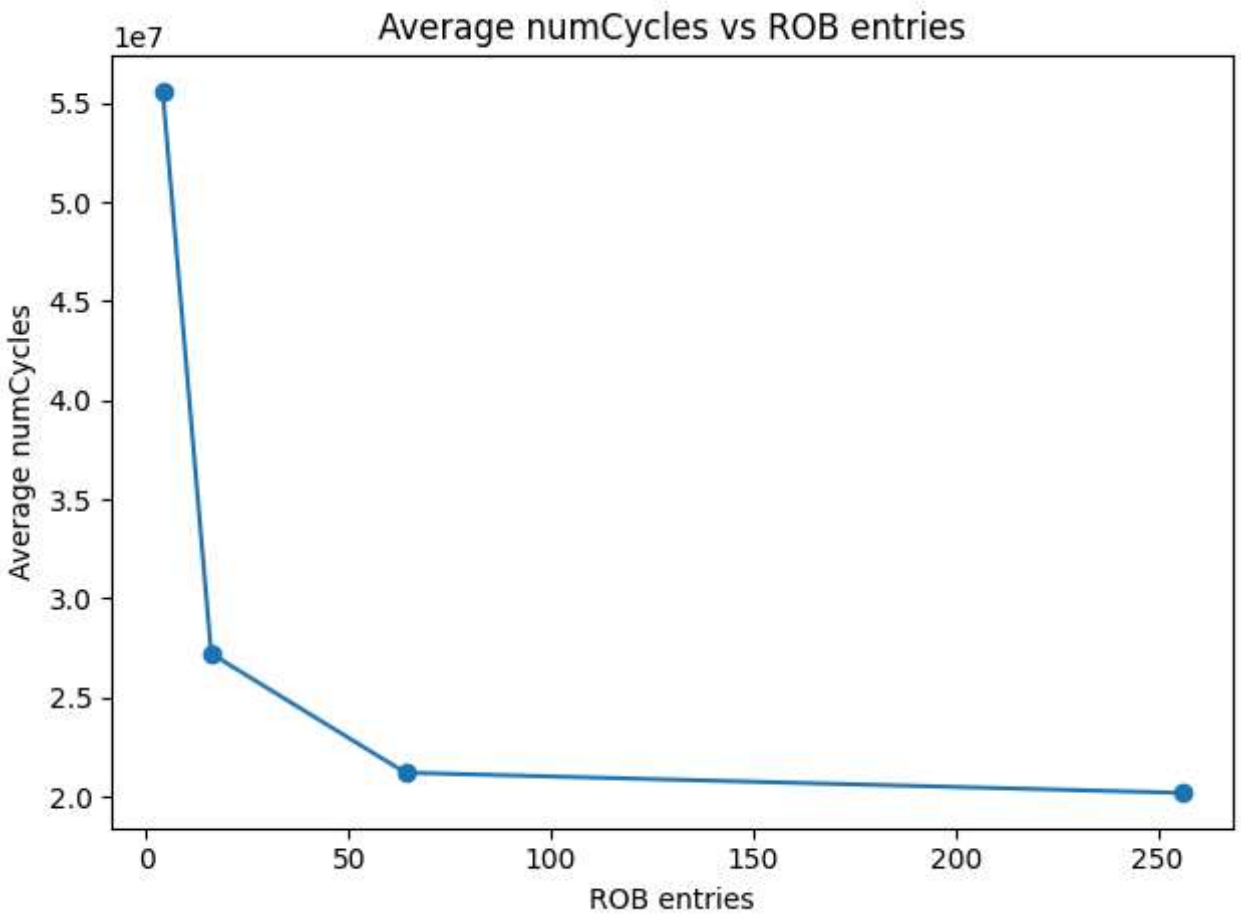
5.1 IQ条目数对总时钟数的影响

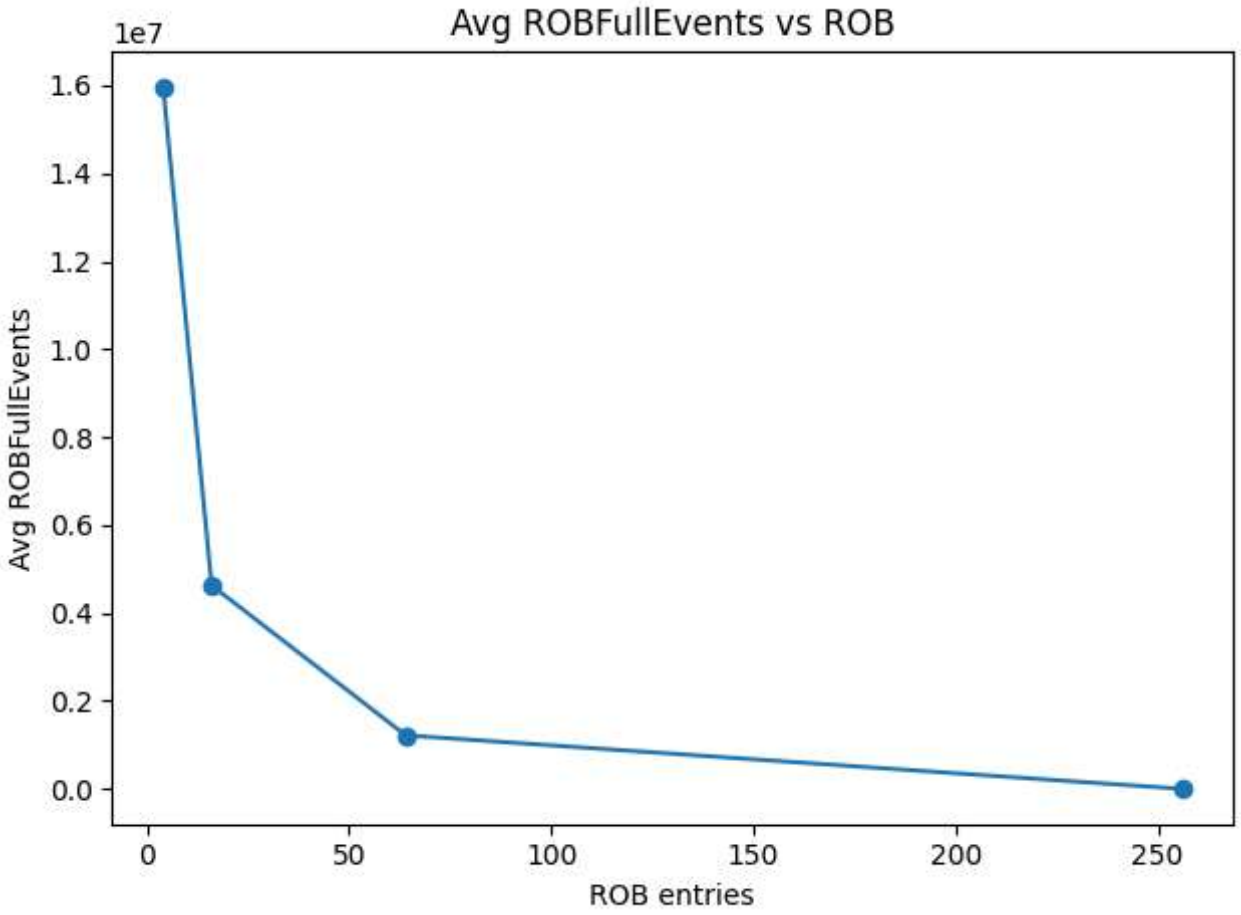




- 观察：在固定 PR、ROB 的条件下，IQ 从较小值提升到中等规模时，numCycles明显下降；继续增大至更大规模后，收益递减甚至平了。
- 解释：增大IQ能容纳更多已准备或待就绪的指令，提高指令级并行度。但是当IQ足够大后，新的瓶颈会变成访存层次时延、数据依赖等，因此整体周期不再显著降低。

5.2 ROB目数对总时钟数的影响



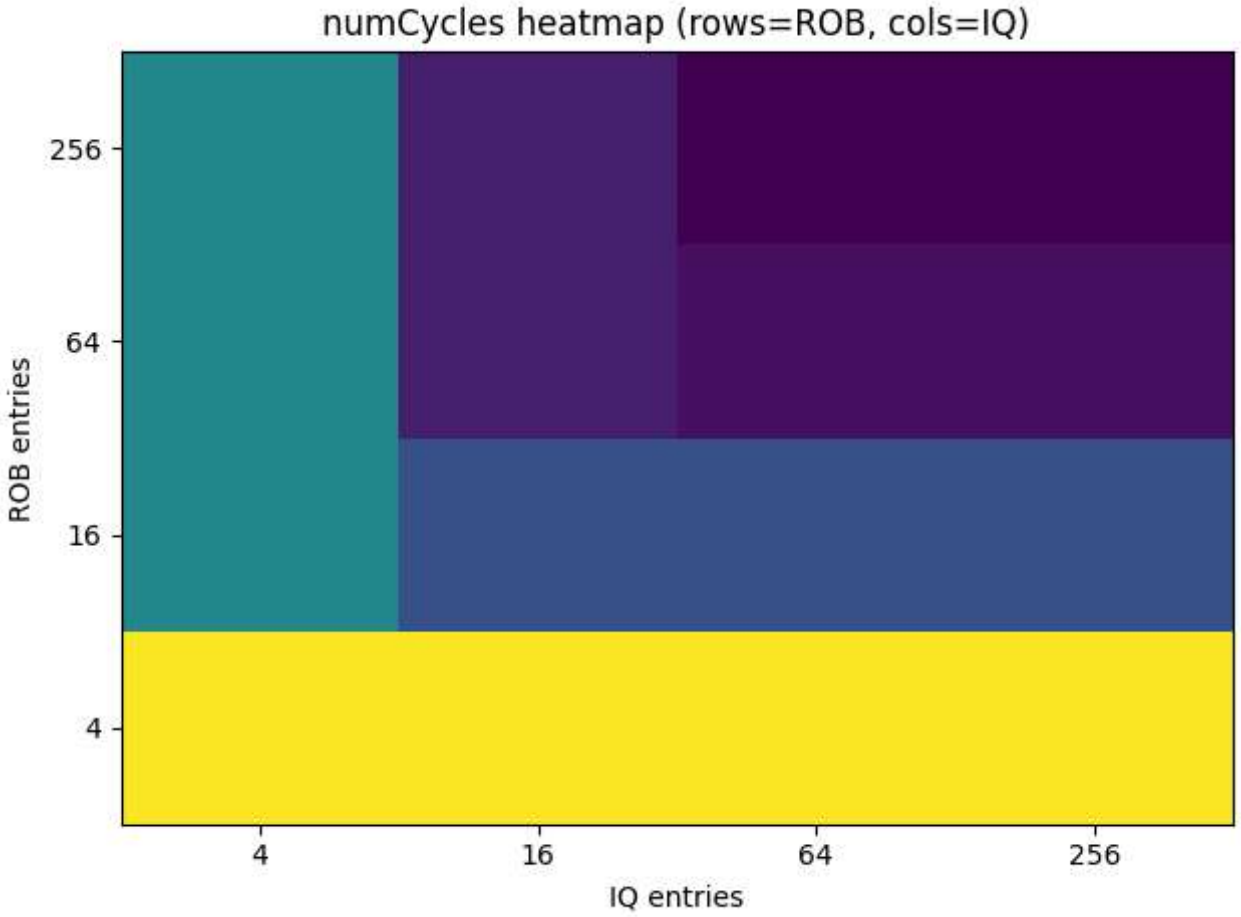


- 观察：ROB扩大也会减少numCycles，但其收益通常弱于IQ，且更早进入递减期。
- 解释：ROB影响的是在序提交窗口的容量与乱序深度；适度增大能减少提交端回压，可是当PR/IQ已经足够时，访存称为主要的依赖，后续继续扩大影响很小。

5.3 总结

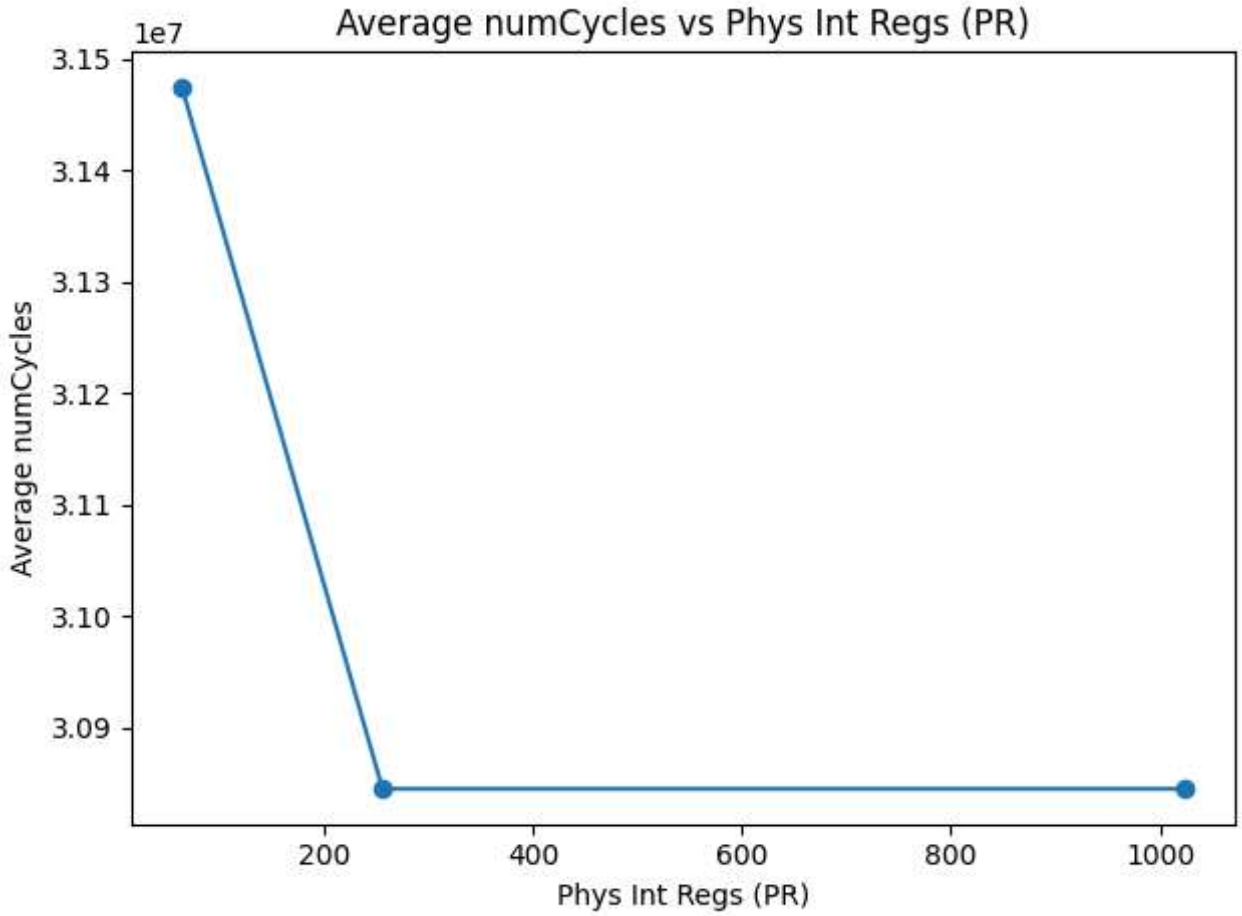
对daxpy这类计算-访存交错负载，二者均存在瓶颈，会更多的受内存层级、寄存器压力或计算约束。

如图可以看到，当二者扩大到一定的数量时，收益下降，具体数据见表格 `pivot_rob_by_iq_cycles.csv`



6. 思考题

1. PR的作用是消除写后读/写后写的假相关，避免因寄存器再利用导致的伪依赖，如果PR不够，会导致
   `system.cpu.rename.fullRegistersEvents` 变高，从而rename速度限制。增大PR会减少numCycles，但是存在瓶颈，因为当 `fullRegistersEvents≈0` 时，继续增加PR，不会再改善，乱序深度由IQ/ROB等决定,同样可以参考图片


Average numCycles vs Phys Int Regs (PR)
1e7
3.15
3.14
3.13
3.12
3.11
3.10
3.09
Average numCycles
200
400
600
800
1000
Phys Int Regs (PR)