COMPUTER ORGANIZATION AND ARCHITECTURE
AUTUMN MID - SEM PROJECT

GPU BOTTLENECK ANALYSIS
GROUP 13

Aim of the Project:
Study the GPGPU Sim-3x Manual and Modify the GPGPU-Sim source code to introduce the counters which capture the state of the warp. State of warp = {waiting, Issued, XALU, XMEM, Other}

State of Warps:
When a kernel executes on an SM, warps of the kernel can be in different states. We classify the warps depending on their state of execution in a given cycle:

• *Waiting-*

The majority of warps are in a waiting state, anticipating an instruction to commit and are primarily waiting for memory values to be returned, making it crucial to manage both memory access frequency and compute capacity to hide memory latency effectively.

To effectively conceal memory access latency, an SM (Streaming Multiprocessor) should concurrently run more warps than the waiting number of warps, considering both memory access patterns and warp compute capabilities.

• *Issued-*

"Issued" warps represent those currently executing instructions in the execution pipeline, and their count reflects the Instructions Per Cycle (IPC) of the Streaming Multiprocessor (SM). A higher number of warps in this state signifies strong SM performance.

A high count of warps in the "Issued" state is indicative of good SM performance and a higher IPC, showcasing efficient execution of instructions within the SM.

• *Excess ALU (Xalu)-*

"Excess ALU (XALU)" warps refer to warps that are prepared to execute arithmetic operations but are currently unable to do so because of resource constraints, typically due to a fixed instruction issuance limit per cycle. XALU represents the surplus warps ready for arithmetic execution.

Warps in the "Excess ALU (XALU)" category are those ready to perform arithmetic operations but are held back due to resource limitations, illustrating the presence of additional computational capacity beyond the current instruction issuance constraints.

• *Excess memory (Xmem)-*

"Excess memory (XMEM)" warps are those that stand ready to dispatch instructions to the Load/Store pipeline but are currently constrained. These warps face limitations when the pipeline experiences stalls due to memory-related bottlenecks or when the pipeline has already reached its maximum instruction issuance capacity. XMEM warps signify additional warps awaiting memory access and potentially increasing pressure on the memory subsystem within the current SM.

Warps categorized as "Excess memory (XMEM)" are prepared to send Load/Store pipeline instructions but are held back due to memory-related bottlenecks or reaching the maximum instruction issuance limit. These warps represent extra demand on the memory subsystem within the current SM.

• *Others-*

"Others" category includes warps waiting on synchronization instructions or warps lacking instructions in the instruction buffer. These warps have unknown requirements since there are no instructions currently available for them.

These five warp states can further be classified into following sub-states:

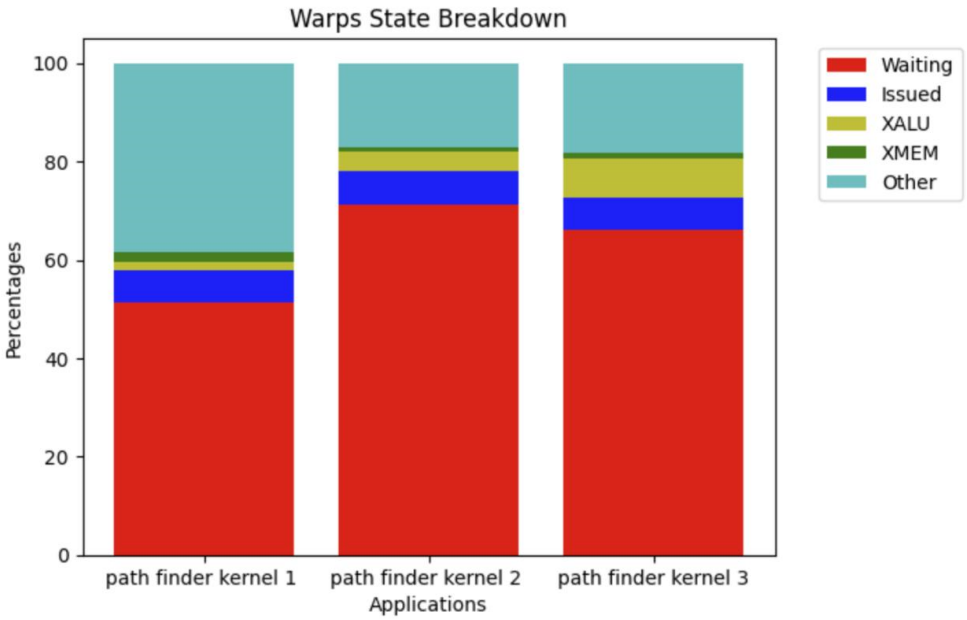
- Waiting
  - Control Hazard
  - Scoreboard
  - Diverged Warp
- Issued
  - Mem
  - ALU
    - SP
    - INT
    - DP
    - SFU
    - Tensor Core
    - SPEC
- Excess ALU
  - SP, INT
  - DP
  - SFU
  - Tensor Core
  - SPEC
- Excess MEM
- Others
  - Empty Instruction Buffer
  - Synchronization

Observations from the state of warps reveal the following patterns:

- Compute-intensive kernels tend to exhibit a notably higher count of warps in the "XALU" state compared to other kernel types. This suggests a strong demand for computational resources in such workloads.
- Memory-intensive and cache-sensitive kernels tend to show a significantly greater number of warps in the "XMEM" state when compared to other categories. This indicates a substantial need for memory resources and suggests a potential sensitivity to cache performance.
- Even in the case of unsaturated kernels (kernels not fully utilizing available resources), there is a clear preference for either compute or memory resources. This is evident from the significant fraction of warps in the "XALU" or "XMEM" states, highlighting the continued demand for these specific resource types within these kernels.

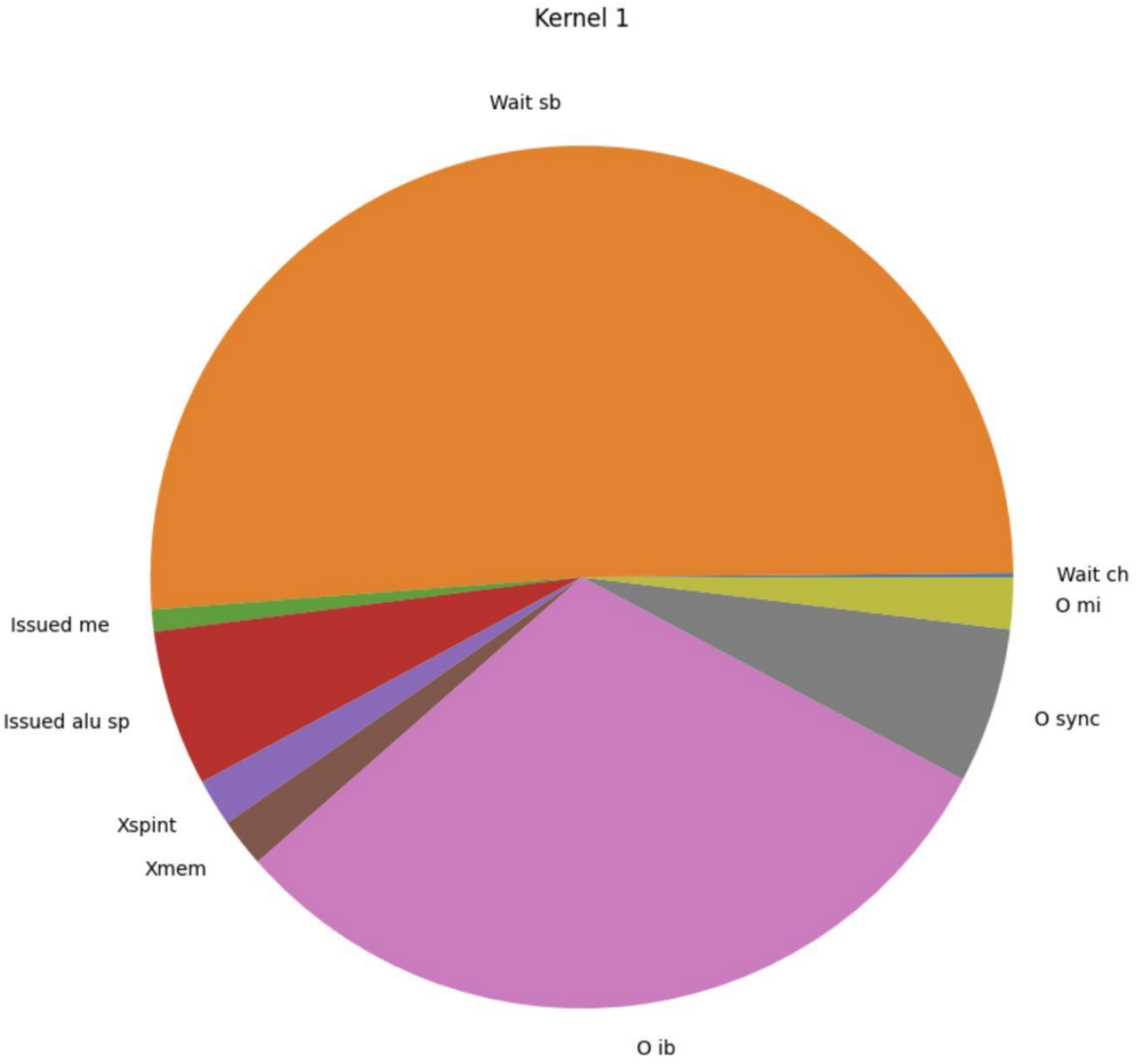
Steps to modify the source code:

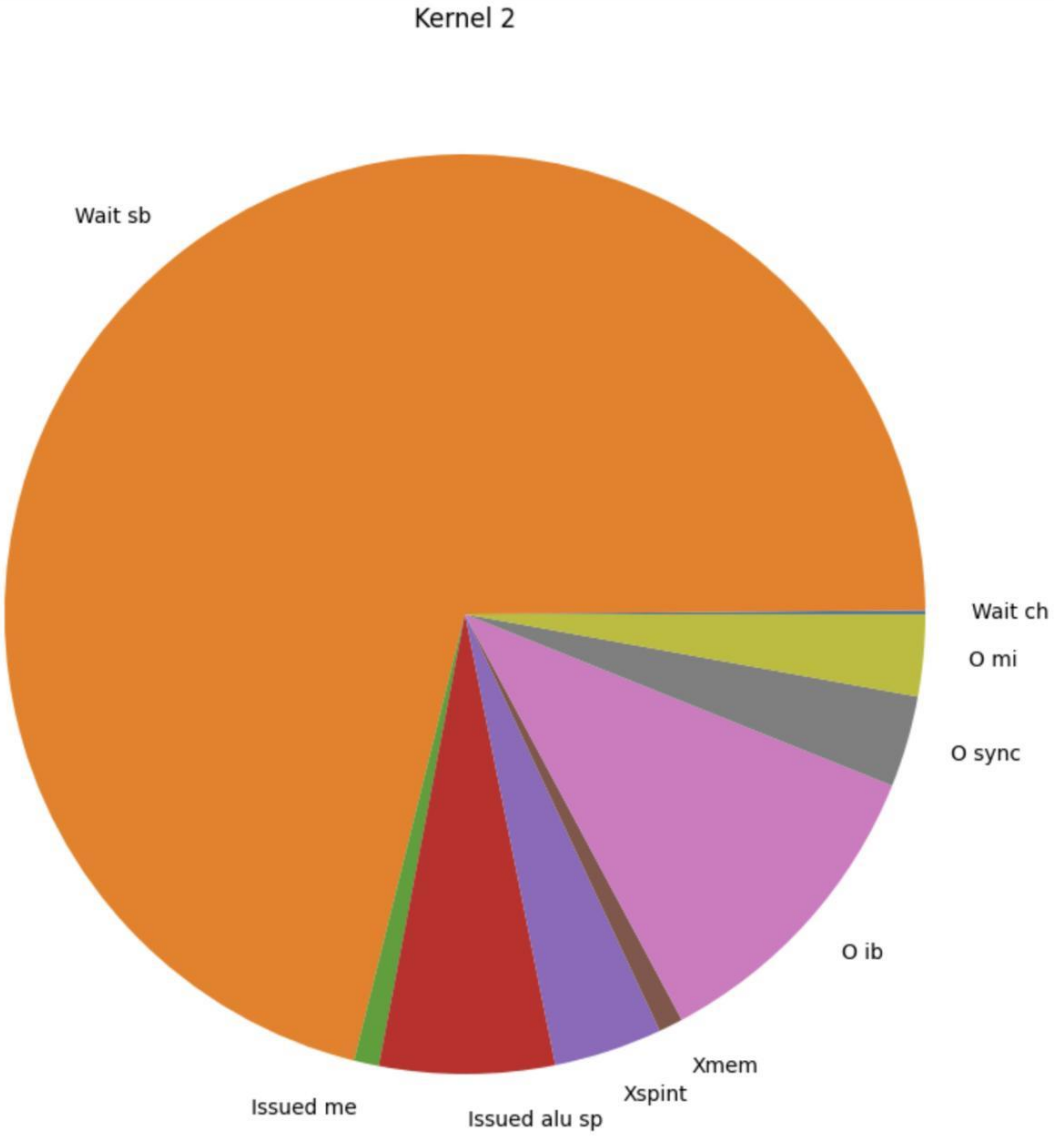
- Simulation cycle count can be obtained by maintaining a counter for the frequency of shader_core_ctx::cycle() function call.
- Based on the conditions specified in the if .. else .. block appropriately maintain and update the counters respectively.
- Following counters have been maintained with their sub-divided counter states
    - Cycle count
    - Waiting
    - XMEM
    - XALU
    - Issued
    - Others
- For finding out the state of prioritized warps that are present after the max_issue_per_warp we have maintained a duplicate while loop ensuring the following conditions.
    - All lines of the code that correspond to the change of state of the scheduler unit class shall be either commented or removed.
    - Ensure that based on the issue state of the warp the further iterations of prioritized warps should handle appropriately.

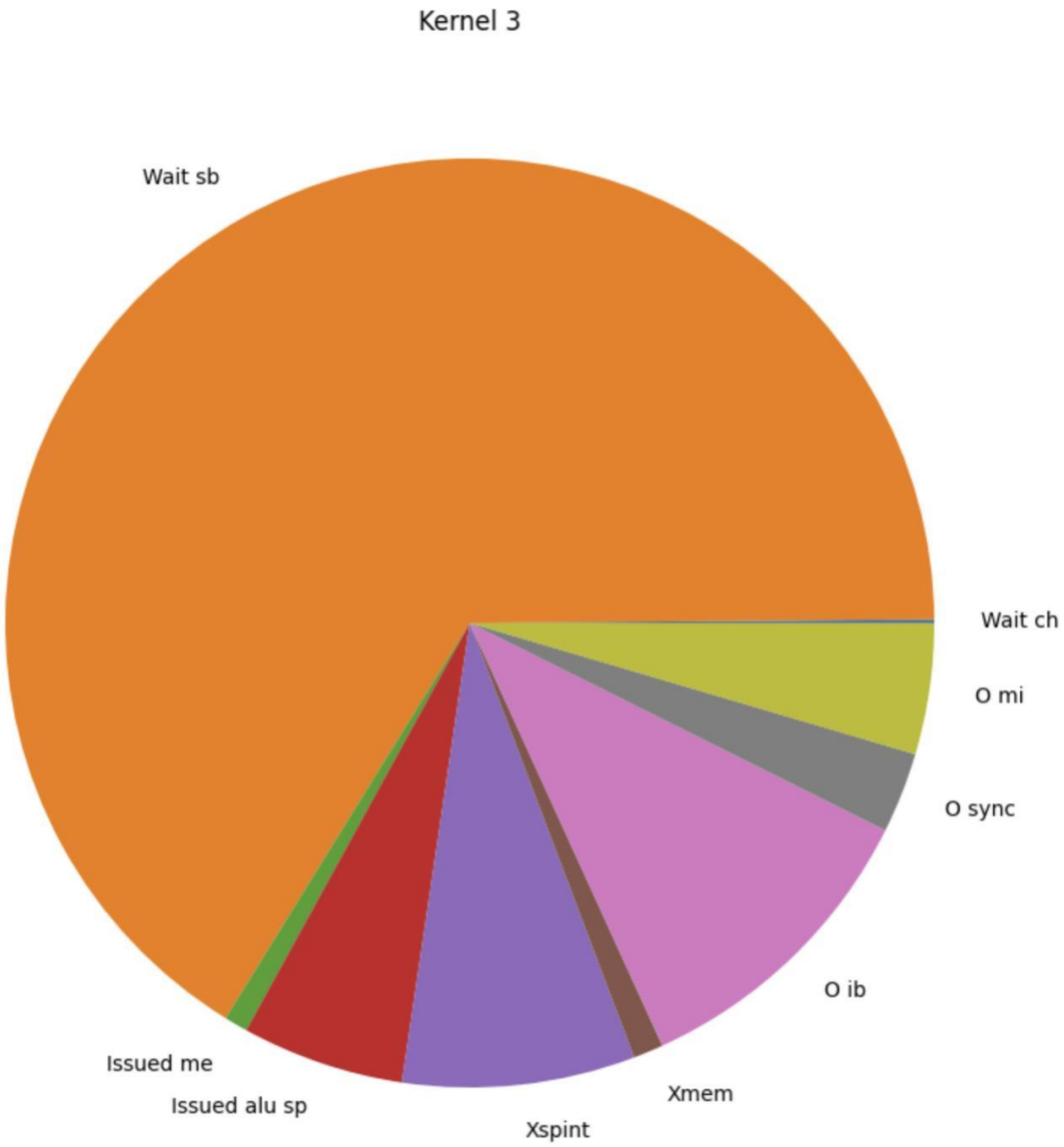


Substates:

- Wait ch – Waiting state due to control hazard
- Wait sb – Waiting state due to scoreboard
- Wait dw – Waiting state due to diverged warp
- Wait mi – Waiting state after maximum instructions have been issued
- Issued me – Issued to memory pipeline
- Issued alu sp - Issued to SP UNIT of ALU pipeline
- Issued alu int - Issued to INT UNIT of ALU pipeline
- Issued alu dp - Issued to DP UNIT of ALU pipeline
- Issued alu sfu - Issued to SFU UNIT of ALU pipeline
- Issued alu tc - Issued to TC UNIT of ALU pipeline
- Issued alu spec - Issued to SPEC UNIT of ALU pipeline
- Xspint – XALU state due to SP, INT UNIT
- Xdp – XALU state due to DP
- Xsfu – XALU state due to DP
- Xtc – XALU state due to DP
- Xspec – XALU state due to SPEC
- Xmem – XMEM state due to memory related bottlenecks
- Xalu mi – XALU state after maximum instructions have been issued
- O ib – Other state due to empty instruction buffer
- O sync – Other state due to barrier
- O mi – Other state after maximum instructions have been issued


Kernel 1
Wait sb
Wait ch
O mi
O sync
O ib
Xmem
Xspint
Issued alu sp
Issued me


Kernel 2
Wait sb
Wait ch
O mi
O sync
O ib
Xmem
Xspint
Issued alu sp
Issued me



DATA:

	wait_ch	wait_sb	wait_dw	wait_mi	issued_mem	issued_alu_sp	issued_alu_int		issued_alu_dp
kernel-1	0.146471	51.05636	0	0	0.82249285	5.85080527		0	0
kernel-2	0.134252	70.99895	0	0	0.87641316	6.143749629		0	0
kernel-3	0.121165	66.06026	0	0	0.81409749	5.649978828		0	0

	issued_alu_dp	issued_alu_sfu	issued_tc	issued_spec	x_spint	x_dp	x_sfu
kernel 1	0	0	0	0	1.774074	0	0
kernel 2	0	0	0	0	3.811523	0	0
kernel 3	0	0	0	0	8.101023	0	0

	x_tc	x_spec	x_mem	xalu_mi	xmem_mi	o_ibe	o_sync	o_mi
kernel 1	0	0	1.826224	0	0	30.75093	5.827627	1.94501
kernel 2	0	0	0.854127	0	0	11.10785	3.206017	2.86712
kernel 3	0	0	1.055804	0	0	10.83348	2.805908	4.558283