GOPTX: FINE-GRAINED GPU KERNEL FUSION BY PTX-LEVEL INSTRUCTION FLOW WEAVING

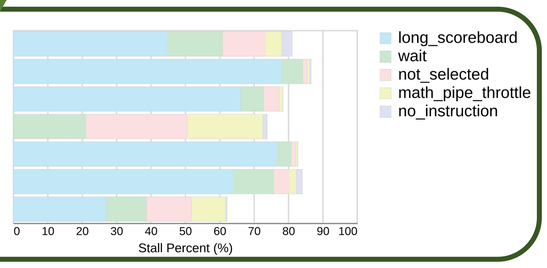
Kan Wu, Zejia Lin, Mengyue Xi, Zhongchun Zheng, Wenxuan Pan, Xianwei Zhang#, Yutong Lu#
Sun Yat-sen University, Guangzhou, China
{wukan3, linzj39, ximy, zhengzhch3, panwx5}@mail2.sysu.edu.cn, {zhangxw79, luyutong}@mail.sysu.edu.cn

Abstract

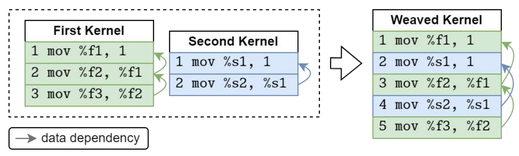
GPUs have been heavily utilized in diverse applications, and numerous approaches, including kernel fusion, have been proposed to boost GPU efficiency through concurrent kernel execution. However, these approaches generally overlook the opportunities to mitigate warp stalls and improve instruction level parallelism (ILP) in inter-kernel resource sharing. To address this issue, we introduce GoPTX, a novel design for kernel fusion that improves ILP through deliberate weaving instructions at the PTX level. GoPTX establishes a merged control flow graph (CFG) from original kernels, enabling to interleaving of instructions that were sequentially executed by default and minimizing pipeline stalls on data hazards. We further propose a latency-aware instruction weaving algorithm for more efficient instruction scheduling and an adaptive code slicing method to enlarge the scheduling space. Experimental evaluation demonstrates that GoPTX achieves an average speedup of 11.2% over the baseline concurrent execution, with a maximum improvement of 23%. The hardware resource utilization statistics show significant enhancements in eligible warps per cycle and resource use.

Insight

The most prevalent stall cause for kernels is the scoreboard, which handles data dependencies and enables out-of-order instruction execution.

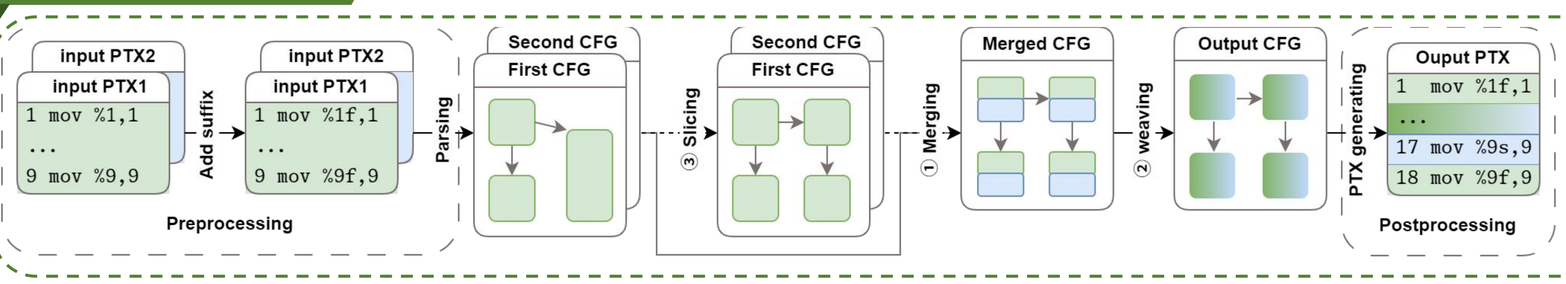


Motivation

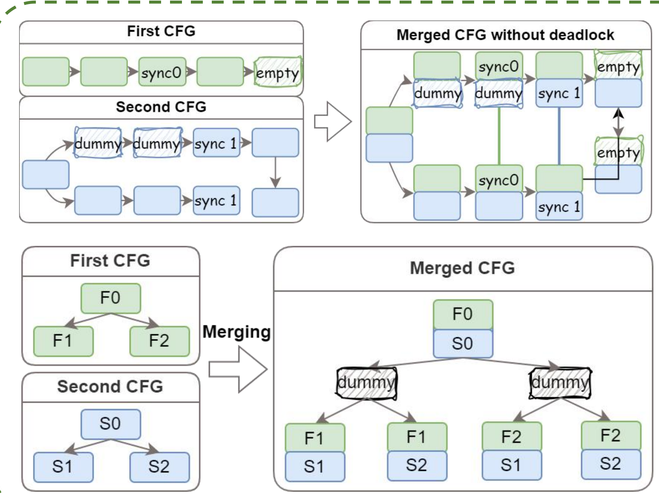


Interleaving independent instructions extends dependency chains, avoiding stalls and hiding latency by filling pipeline bubbles.

Design



Overall workflow of GoPTX, with three key phases of Slicing, Weaving and Merging, and additional phases of Preprocessing and Postprocessing.



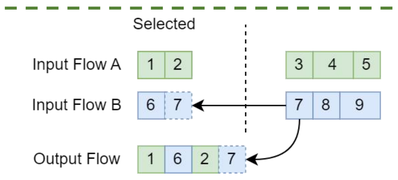
Deadlock solved by inserting dummy blocks before synchronizations, delaying the barriers top revent conflicts.

Merge conditional branches: insert dummy nodes and branching for divergent control flow.

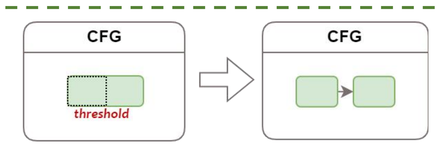
Algorithm 1 Control Flow Graph Merging

```
input: F ≠ ∅, S ≠ ∅ {the first and the second input CFG}
output: M {the merged CFG}
M ← ∅
M.add_node(< 0, 0 >)
Stack ← {< 0, 0 >}
Pushed ← {< 0, 0 >}
while Stack ≠ ∅ do
   < f, s >← Stack.pop()
   M[< f, s >].block ← F[f].block + S[s].block
   for < u, v >∈ get_next_list(< f, s >, F, S) do
      if < u, v >∉ Pushed then
         M.add_node(< u, v >)
         Stack.push(< u, v >)
         Pushed.insert(< u, v >)
      end if
      M.add_edge(< f, s >, < u, v >)
   end for
end while
return M
```

Algorithm above describes CFG merging (CFM) as a traversal procedure working on BB-level.



The weaving process. Instruction 7 is selected because I6's latency is lower than the sum of I1 and I2.



Limit each slice's execution time to the threshold, which is the average of execution cycles of all BBs.

Results

First Kernel

```
__global__ __launch_bounds__(32)
void first(float x) {
  x1 = 1; x2 = 2;
  if (x < 0) {
    x2 = 1; x1 = 2;
  }
}
```

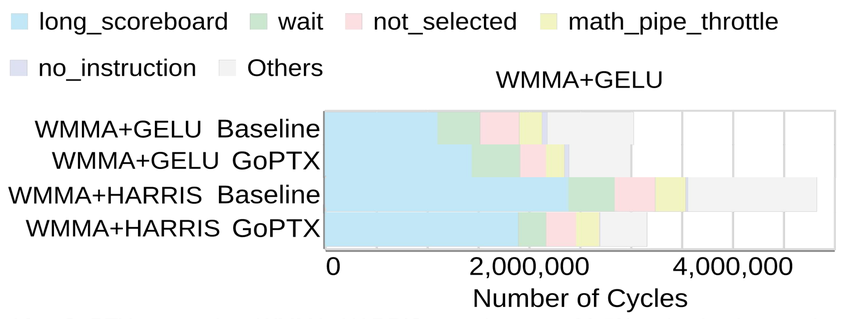
Second Kernel

```
__global__ __launch_bounds__(32)
void second(float y) {
  y1 = 1; y2 = 2;
  if (y < 0) {
    y2 = 1; y1 = 2;
  }
}
```

Desire Kernel

```
__global__ __launch_bounds__(32)
void desire(float x, float y) {
  x1 = 1; y1 = 1;
  x2 = 2; y2 = 2;
  if (x < 0 && y < 0) {
    x2 = 1; y2 = 1;
    x1 = 2; y1 = 2;
  } else if (x < 0) {
    x2 = 1; x1 = 2;
  } else if (y < 0) {
    y2 = 1; y1 = 2;
  }
}
```

Illustrative examples of first and second input kernel with coarse-grained and fine- grained (Desired) fusion.



After GoPTX processing, WMMA+HARRIS experiences a 20.5% reduction in scoreboard stalls, while WMMA+GELU incurs a 30.4% increase (still better than VFuse and HFuse). The overall speedup delivers an average of 11.2% speedup over the baseline, ranging from 23%(STMS+LUD) to-2% (WMMA+GELU).

DAC 2025

SUN YAT-SEN UNIVERSITY