

# Profiling GPU Shaders for Profile-Guided Optimizations

Sebastian Neubauer  
Technische Universität München

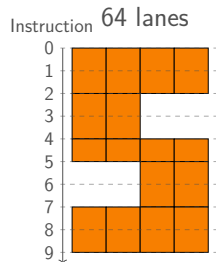
July 25, 2019



# GPUs

## Hardware

- ▶ SIMD-units with 64 lanes
- ▶ Diverging control flow by masking lanes (SIMT)
- ▶ AMD Radeon VII has 240 SIMD units



# Vulkan

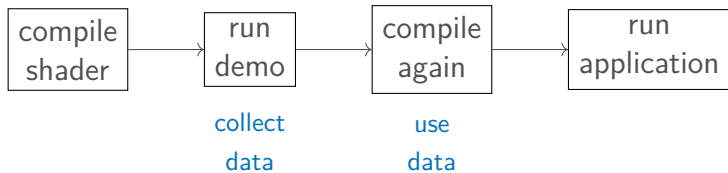
## Software

- ▶ Graphics and compute standard for GPUs
- ▶ Shaders are loaded in SPIR-V
- ▶ Compilation to ISA happens in driver



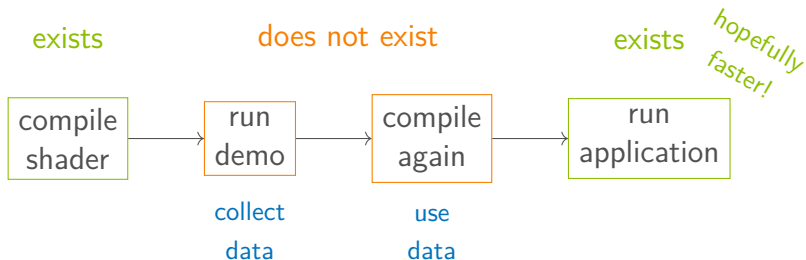
# Workflow

## Profile-Guided Optimization



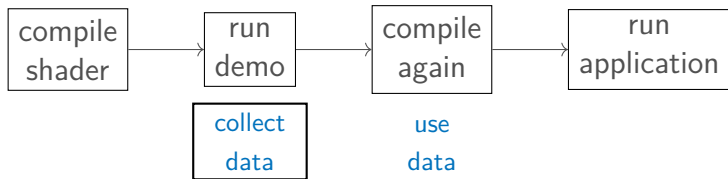
# Current State

## Profile-Guided Optimization



# This Thesis

## Profile-Guided Optimization

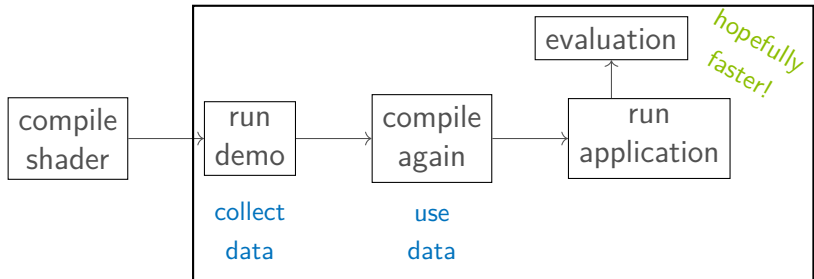


*hopefully  
faster!*

original topic

# This Thesis

## Profile-Guided Optimization

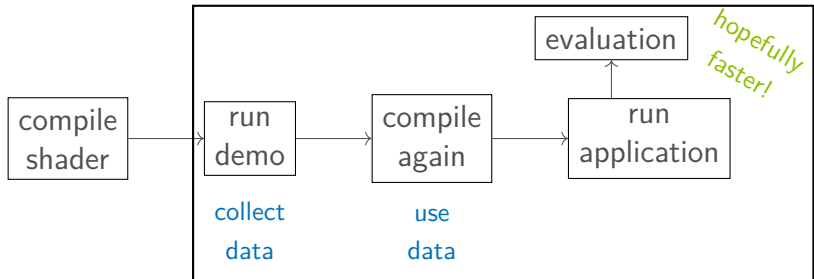


new topic

# This Thesis

## Profile-Guided Optimization

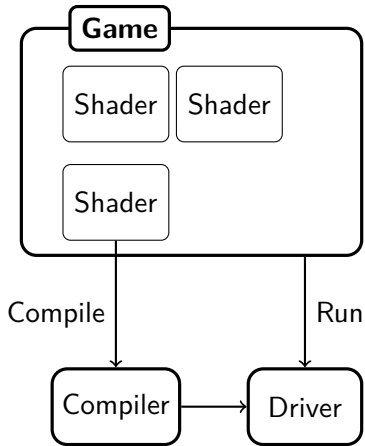
everything works 🎉



new topic

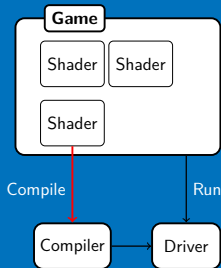


# Basic Block Counting



# Basic Block Counting

## GLSL/SPIR-V

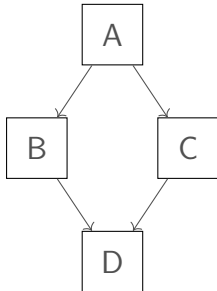
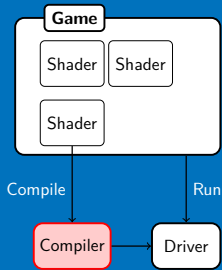


- ▶ GLSL gets precompiled to SPIR-V
- ▶ SPIR-V is passed to driver

```
1 if (inputPos.x < 0.5) {  
2     outColor = vec4(1.0, 0.0, 0.0, 1.0);  
3 } else {  
4     outColor = vec4(0.0, 0.0, 1.0, 1.0);  
5 }
```

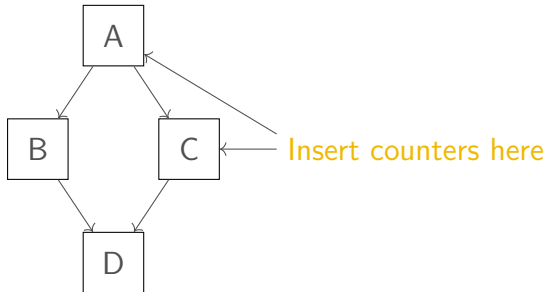
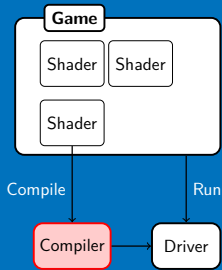
# Basic Block Counting

CFG



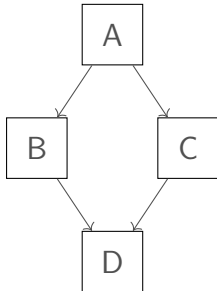
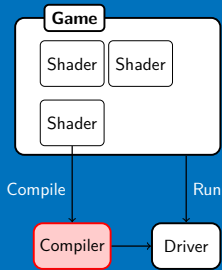
# Basic Block Counting

CFG

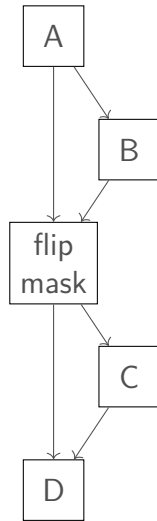


# Basic Block Counting

CFG

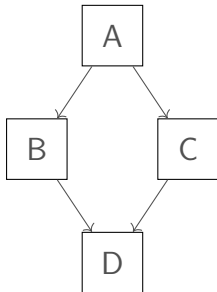
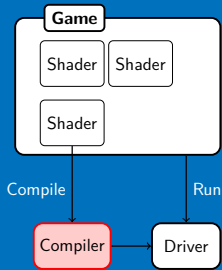


Structurize  
 $\Rightarrow$

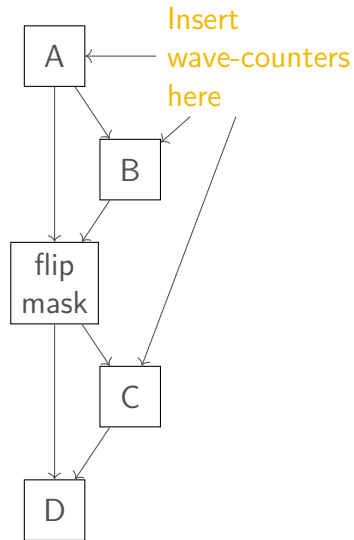


# Basic Block Counting

CFG

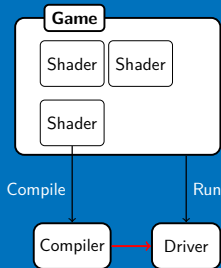


Structurize  
 $\Rightarrow$



# Basic Block Counting

## ELF

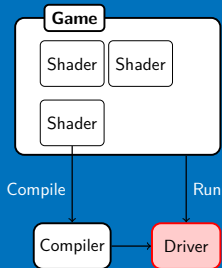


► ELF file contains metadata and sections for counters

```
1  .text
2      <code>
3
4  .rel.text
5      <relocations for counter-pointers>
6
7  __llvm_prf_cnts
8      <zero initialized counters>
9
10 __llvm_prf_data
11     <metadata>
```

# Basic Block Counting

## Save Counter



- ▶ Counters are saved when the pipeline is destroyed and every 10 s
- ▶ Fetch counters from GPU memory
- ▶ Write counters and metadata to file



# Basic Block Counting

Result

- ▶ Declares pixel shader as *hot* and vertex shader as *unlikely*
- ▶ Changes basic block ordering

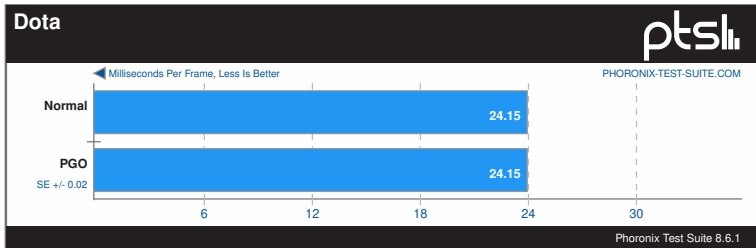


# Basic Block Counting

## Result



- Dota: No change apart from compilation speed on first run



- Ashes:  $(60.007 \pm 0.004)$  ms vs  $(60.009 \pm 0.003)$  ms



# Future Work

- ▶ Find dynamically uniform variables
- ▶ Create some interesting statistics, e.g. unused basic blocks, uniform branches
- ▶ More benchmarks
- ▶ (More optimizations)



Questions?