Profile-Guided Optimizations for GPUs

Sebastian Neubauer
Technische Universität München

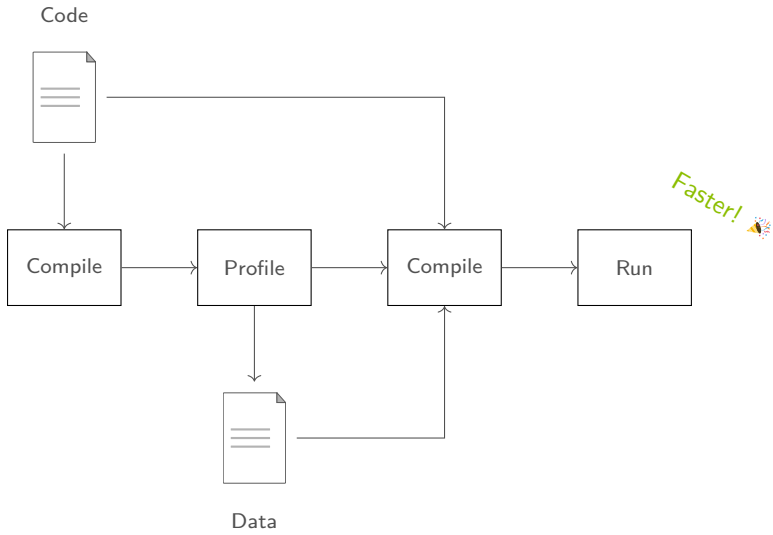
October 10, 2019

Problem

Compiler has to guess what happens at runtime

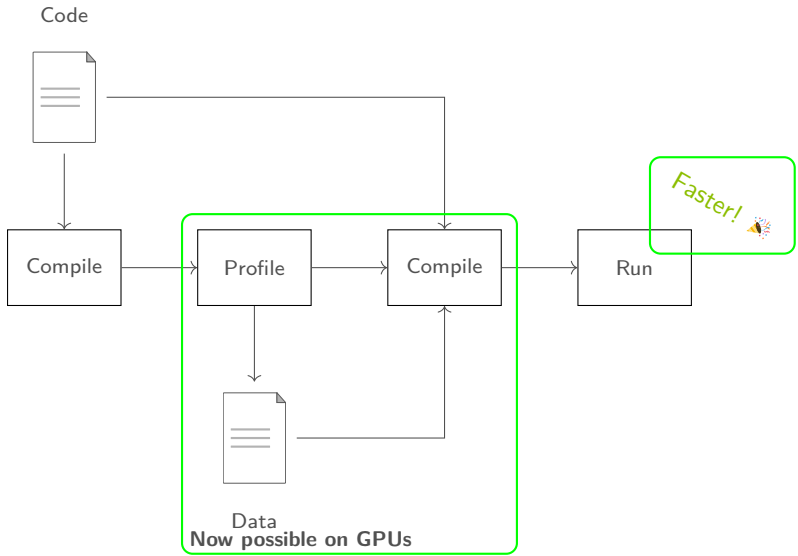
Solution

Profile-Guided Optimizations



Solution

Profile-Guided Optimizations



Solution

Profile-Guided Optimizations

Optimizations

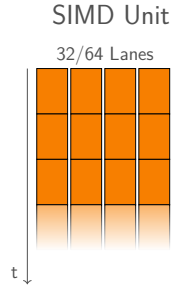
- Linearization
- Register allocation
- “Constant” propagation
- Uniform optimizations

Needed data

- Branch probabilities
- Variable values
- Uniformity of variables

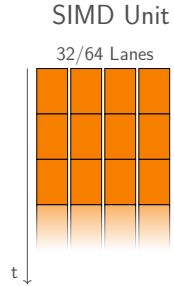
GPUs

Hardware



GPUs

Hardware

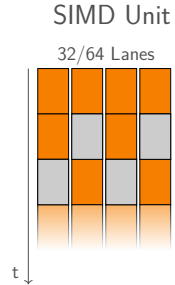



```
layout(location = 0) in float in_value;
layout(location = 1) out ivec3 out_color;
void main() {
    int a;
    if (in_value < 0.5)
        a = 2;
    else
        a = 0;

    int r = a * 3;
    int g = a * int(in_value);
    out_color = ivec3(r, g, 0);
}
```

GPUs

Hardware

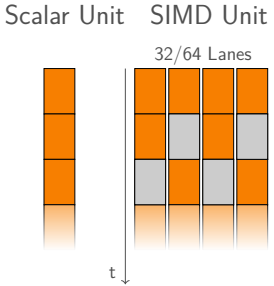



```
layout(location = 0) in float in_value;
layout(location = 1) out ivec3 out_color;
void main() {
    int a;
    if (in_value < 0.5)
        a = 2;
    else
        a = 0;

    int r = a * 3;
    int g = a * int(in_value);
    out_color = ivec3(r, g, 0);
}
```

GPUs

Hardware




```
1  layout(location = 0) in float in_value;
2  layout(location = 1) out ivec3 out_color;
3  void main() {
4      int a;
5      if (in_value < 0.5)
6          a = 2;
7      else
8          a = 0;
9
10     int r = a * 3;
11     int g = a * int(in_value);
12     out_color = ivec3(r, g, 0);
13 }
```

GPUs

Software

```
1 ...
2 ; in_value is in v1
3 ; int a is in v0
4
5 v_mov_b32_e32 v2, 0
6 v_mul_lo_i32 v1, v0, v1
7 v_mul_u32_u24_e32 v0, 3, v0
8 exp mrt0 v0, v1, v2, off
      done vm
9 s_endpgm
```

GPUs

Software

```
...
; in_value is in v1
; int a is in v0

v_mov_b32_e32 v2, 0
v_mul_lo_i32 v1, v0, v1
v_mul_u32_u24_e32 v0, 3, v0
exp mrt0 v0, v1, v2, off
    done vm
s_endpgm
```

```
...
; in_value is in v1
; int a is in s0

v_mul_lo_i32 v0, s0, v1
s_mul_i32 s0, s0, 3
v_mov_b32_e32 v2, 0
v_mov_b32_e32 v1, s0
exp mrt0 v1, v0, v2, off
    done vm
s_endpgm
```

Measure Uniformity

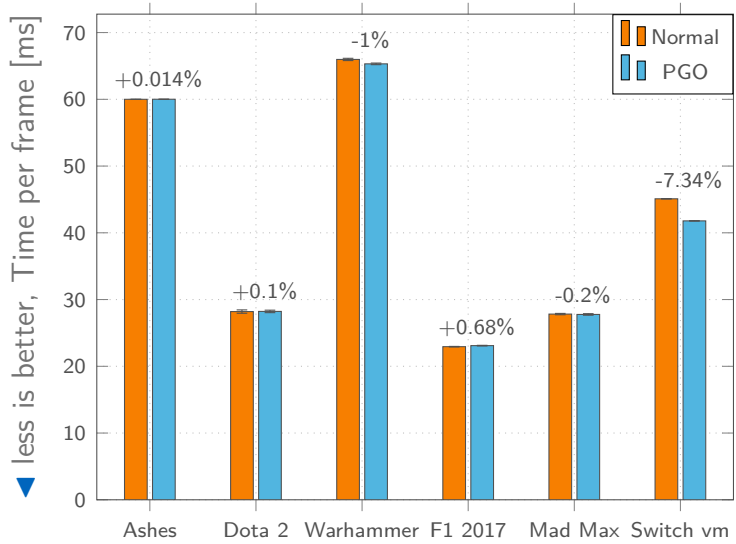
Find out if `x` is uniform

- `first = readfirstlane(x)`
- `cmp_mask = icmp(x != first)`
- `is_non_uniform = cmp_mask != 0`
- `atomic_add(<counter>, is_non_uniform)`

What else?

- Implement PGO for GPUs
- ELF-Loader
- Fix bugs in AMD drivers
- Remove unused shader code
- Statistics about counters, unused code, register usage and uniformity
- Benchmarks of optimizations
- Benchmarks of instrumentation overhead

Performance



Future Work

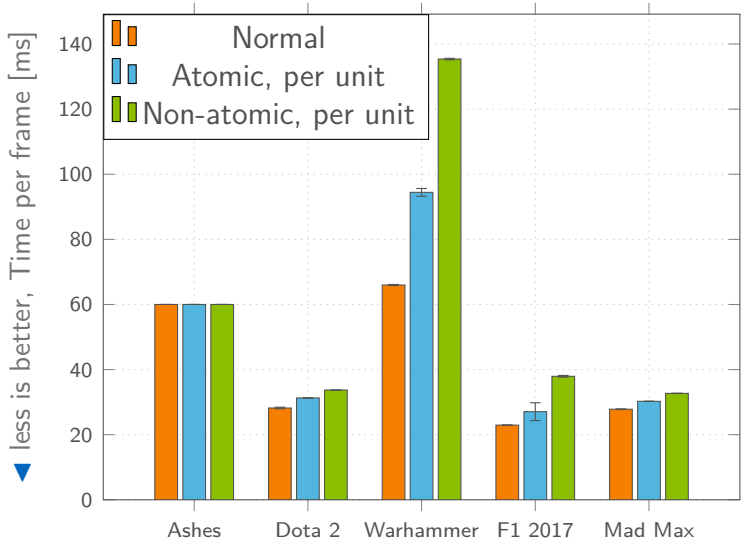
- Implement optimizations
- More benchmarks

Questions?

Performance

Game	Config	Time per frame	Difference
Ashes	Normal	(60.0034 ± 0.0022) ms	
	PGO	(60.0118 ± 0.0019) ms	(0.014 ± 0.005) %
	PGO + removing blocks	(60.006 ± 0.006) ms	(0.004 ± 0.010) %
Dota 2	Normal	(28.20 ± 0.26) ms	
	PGO	(28.22 ± 0.19) ms	(0.1 ± 1.2) %
	PGO + removing blocks	(28.17 ± 0.17) ms	(-0.1 ± 1.1) %
Warhammer	Normal	(65.98 ± 0.18) ms	
	PGO	(65.31 ± 0.13) ms	(-1.0 ± 0.4) %
F1 2017	Normal	(22.94 ± 0.05) ms	
	PGO	(23.10 ± 0.05) ms	(0.68 ± 0.29) %
Mad Max	Normal	(27.82 ± 0.11) ms	
	PGO	(27.77 ± 0.12) ms	(-0.2 ± 0.6) %
	PGO + removing blocks	(27.79 ± 0.09) ms	(-0.1 ± 0.5) %
Switch vm	Normal	(45.10 ± 0.04) ms	
	PGO	(41.79 ± 0.04) ms	(-7.34 ± 0.10) %
	PGO + removing blocks	(35.60 ± 0.04) ms	(-21.06 ± 0.10) %

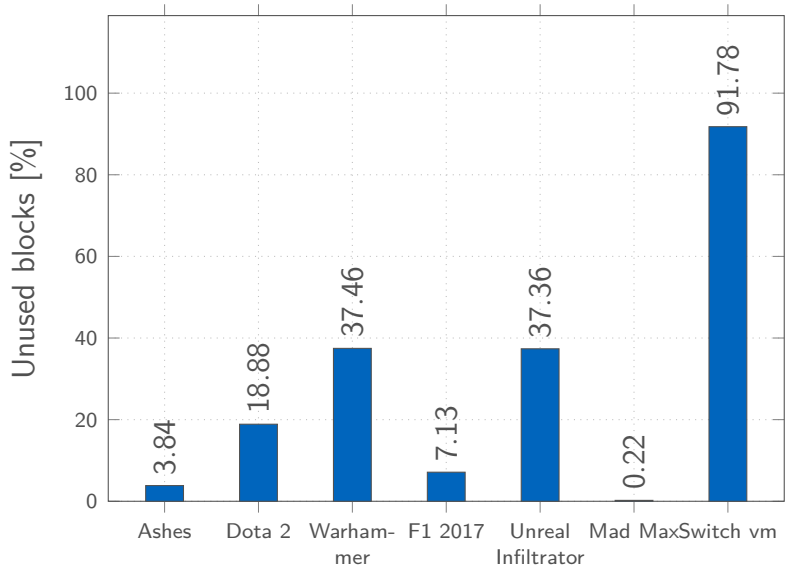
Overhead


Normal
Atomic, per unit
Non-atomic, per unit
less is better, Time per frame [ms]
0
20
40
60
80
100
120
140
Ashes
Dota 2
Warhammer
F1 2017
Mad Max

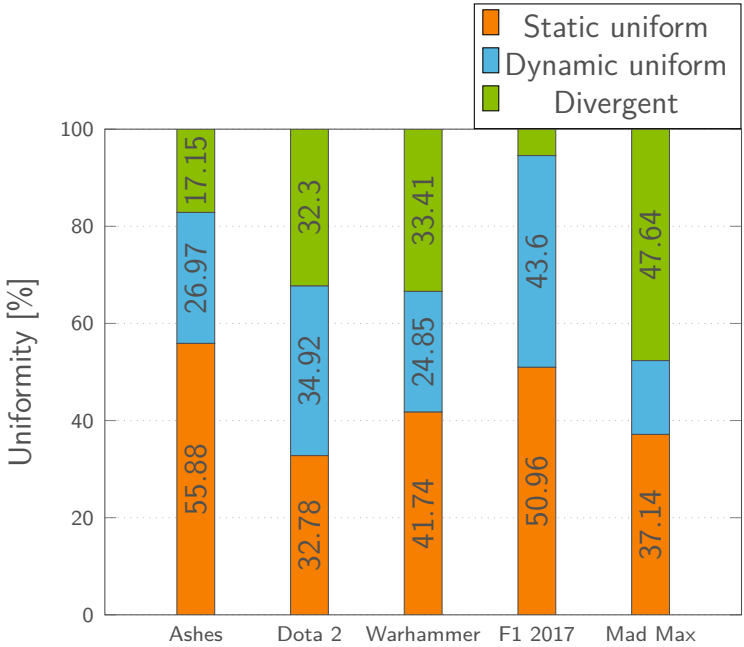
Overhead

Game	Config	Time per frame	Overhead
Ashes	Normal	(60.0034 ± 0.0022) ms	
	Atomic, per unit	(60.010 ± 0.008) ms	(0.010 ± 0.013) %
	Non-atomic, per unit	(60.0080 ± 0.0022) ms	(0.008 ± 0.006) %
Dota 2	Normal	(28.20 ± 0.26) ms	
	Atomic, per unit	(31.28 ± 0.05) ms	(10.9 ± 1.1) %
	Non-atomic, per unit	(33.71 ± 0.06) ms	(19.5 ± 1.1) %
Warhammer	Normal	(65.98 ± 0.18) ms	
	Atomic, per unit	(94.4 ± 1.2) ms	(43.1 ± 1.9) %
	Non-atomic, per unit	(135.39 ± 0.23) ms	(105.2 ± 0.7) %
F1 2017	Normal	(22.94 ± 0.05) ms	
	Atomic, per unit	(27.1 ± 2.8) ms	(18 ± 12) %
	Non-atomic, per unit	(37.94 ± 0.26) ms	(65.4 ± 1.2) %
Mad Max	Normal	(27.82 ± 0.11) ms	
	Atomic, per unit	(30.25 ± 0.06) ms	(8.7 ± 0.5) %
	Non-atomic, per unit	(32.69 ± 0.06) ms	(17.5 ± 0.5) %

Unused Code


Unused blocks [%]
0
20
40
60
80
100
3.84
18.88
37.46
7.13
37.36
0.22
91.78
Ashes
Dota 2
Warham-mer
F1 2017
Unreal Infiltrator
Mad Max
Switch vm

Uniform Branches



Uniform Loads