

Performance Impact of Undefined Behavior Optimizations in C/C++

Lucian Popescu* Răzvan Deaconescu* Nuno
Lopes**

*Facultatea de Automatică și Calculatoare, Universitatea Politehnică din
București

**Instituto Superior Técnico, Universidade de Lisboa

February 3, 2024

Background

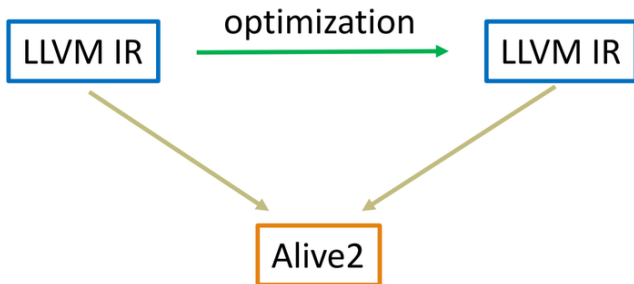
- Undefined behavior is used as a free ticket for optimizations
- We know their impact on microbenchmarks

Past semester

- We created a C/C++ benchmarking suite using Phoronix
- We created flags for controlling undefined behavior optimizations
 - Already existed: -fwrapv, -fstrict-enums, etc
 - Created by us: -fconstrain-shift-value, -mllvm -zero-uninit-loads, etc

This semester (1)

- We ran Alive2 over the suite to discover new flags

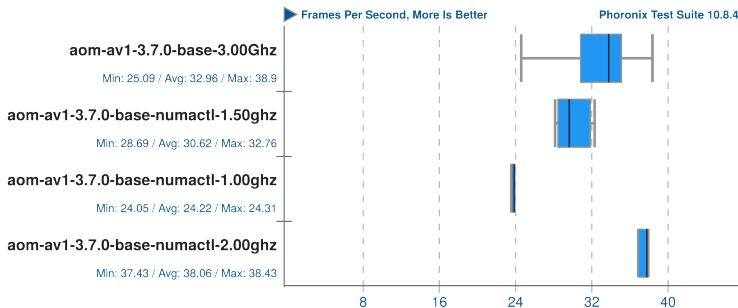


This semester (2)

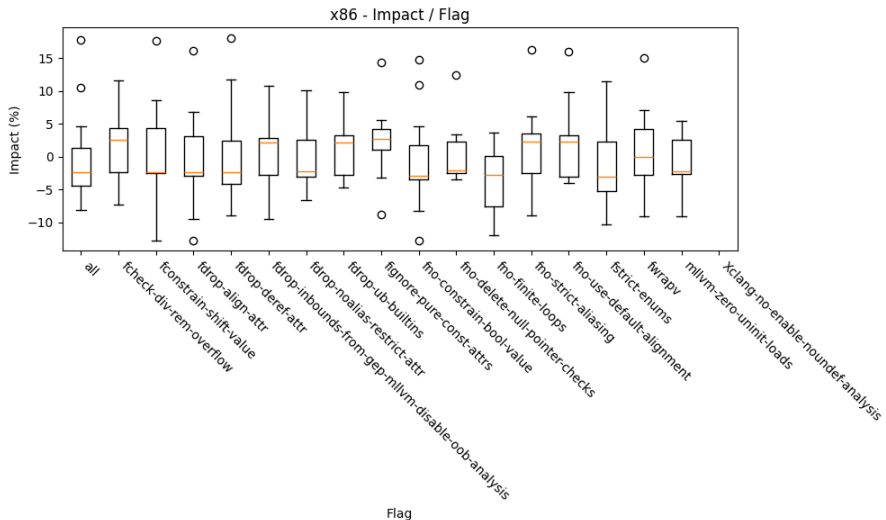
- We ran the suite on ARM as well and stabilized the results

AOM AV1 3.7

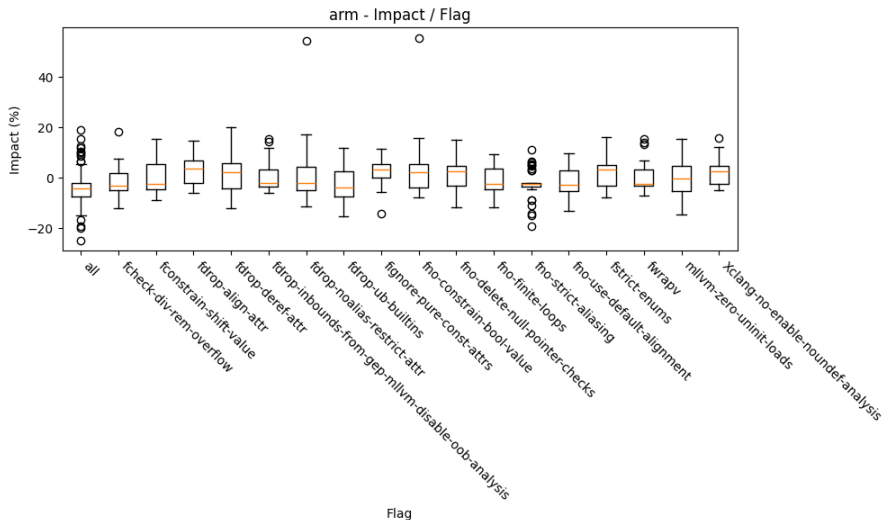
Encoder Mode: Speed 11 Realtime - Input: Bosphorus 4K



This semester (3)



This semester (3)



This semester (4)

- We analyzed a performance drop of 12% in the simdjson benchmark
- Caused by the absence of the align attribute on pointers

This semester (4)

```
-define i1 @_ZN1A3fooEii(ptr align 8 dereferenceable(8) %this, ...
+define i1 @_ZN1A3fooEii(ptr dereferenceable(8) %this, ...
  entry:
-  %0 = load ptr, ptr %this, align 8
    %smax = tail call i32 @llvm.smax.i32(i32 %n, i32 0)
    %wide.trip.count = zext i32 %smax to i64
    br label %while.cond

...

while.body:
  %indvars.iv.next = add nuw nsw i64 %indvars.iv, 1
+  %0 = load ptr, ptr %this, align 8, !tbaa !5
  %arrayidx = getelementptr inbounds i32, ptr %0, i64 %indvars.iv.next
  %1 = load i32, ptr %arrayidx, align 4
  %cmp2 = icmp eq i32 %1, %c
  br i1 %cmp2, label %cleanup, label %while.cond

...
```

Next semester

- We continue to analyze the cause of the performance drops
- We run the suite with -O3, -Os, -Oz
- We run the suite on AMD