

LLVM Advisor

A Unified Visualization Tool for Compiler Artifacts

Miguel Cárdenas | Google Summer of Code 2025

Mentored by Jose M. Monsalve Diaz¹, Kevin Sala² and Johannes Doerfert²

¹Advanced Micro Devices

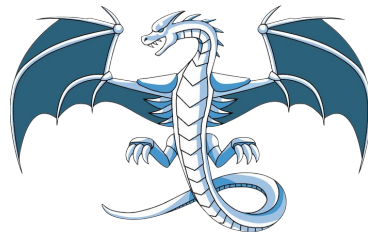
²Lawrence Livermore National Laboratory

2025 LLVM Developers' Meeting
October 28, 2025

Context

Context

- **LLVM reports valuable compilation data**
 - Optimization remarks
 - Profiling data
 - Timing data



Context

- LLVM reports valuable compilation data

- Optimization remarks
- Profiling data
- Timing data

- However...

- Multiple compilation flags
- Different output data formats
- Finding remarks in specific regions

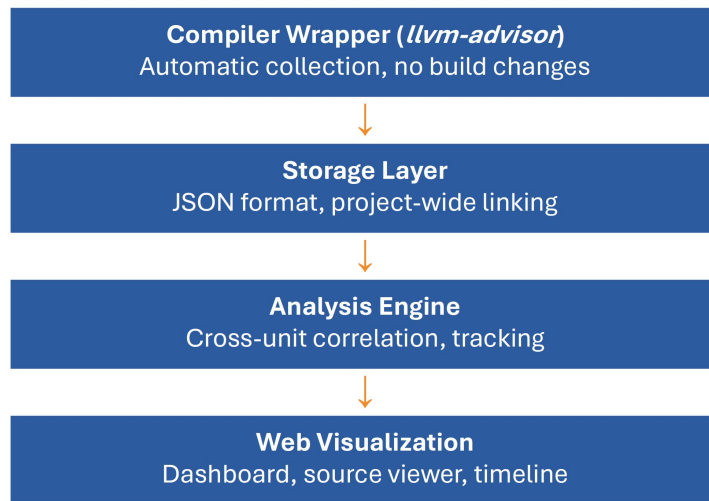


```
remark: in artificial function
'__omp_offloading_803_29744d_main_l9_omp_outlined_omp_outlined',
alloca ('%15') without debug info with static size of 4 bytes
[-Rpass=kernel-info]
test.c:9:5: remark: in artificial function
'__omp_offloading_803_29744d_main_l9_omp_outlined_omp_outlined',
direct call, callee is '@llvm.lifetime.start.p0' [-Rpass=kernel-info]
9 | #pragma omp target teams distribute parallel for map(to: a, b) m
...
| ^
test.c:9:5: remark: in artificial function
'__omp_offloading_803_29744d_main_l9_omp_outlined_omp_outlined',
direct call, callee is '@llvm.lifetime.start.p0' [-Rpass=kernel-info]
test.c:9:5: remark: in artificial function
'__omp_offloading_803_29744d_main_l9_omp_outlined_omp_outlined',
direct call, callee is '@llvm.lifetime.start.p0' [-Rpass=kernel-info]
test.c:9:5: remark: in artificial function
'__omp_offloading_803_29744d_main_l9_omp_outlined_omp_outlined',
direct call, callee is '@llvm.lifetime.start.p0' [-Rpass=kernel-info]
test.c:9:5: remark: in artificial function
'__omp_offloading_803_29744d_main_l9_omp_outlined_omp_outlined',
'load' instruction ('%25') accesses memory in flat address space
[-Rpass=kernel-info]
test.c:9:5: remark: in artificial function
'__omp_offloading_803_29744d_main_l9_omp_outlined_omp_outlined',
direct call, callee is '@_kmpc_for_static_init_4'
[-Rpass=kernel-info]
test.c:11:16: remark: in artificial function
'__omp_offloading_803_29744d_main_l9_omp_outlined_omp_outlined',
"/home/cc/Files/llvm/bin/clang-offload-packager" "-o" "/tmp/test-f38eb3.out"
"-image=file=/tmp/test-sm_75-7a20ba.o, triple=nvptx64-nvidia-cuda, arch=sm-
75, kind=openmp, feature=ptx85"
"/home/cc/Files/llvm/bin/clang-21" "-cc1" "-triple" "x86_64-unknown-linux-g
nu" "-O2" "-emit-obj" "-dumpdir" "a-" "-disable-free" "-clear-ast-before-bac
kend" "-disable-llvm-verifier" "-discard-value-names" "-main-file-name" "tes
t.c" "-mrelocation-model" "pic" "-pic-level" "2" "-pic-is-pie" "-mframe-poin
ter=none" "-fmath-errno" "-ffp-contract=on" "-fno-rounding-math" "-mconstruc
tor-aliases" "-funwind-tables=2" "-target-cpu" "x86-64" "-tune-cpu" "generic"
"-debug-info-kind=constructor" "-dwarf-version=5" "-debugger-tuning=gdb" "-
fdebug-compilation-dir=/home/cc/testing" "-fcoverage-compilation-dir=/home/
cc/testing" "-resource-dir" "/home/cc/Files/llvm/lib/clang/21" "-error-limi
t" "19" "-fmessage-length=76" "-fopenmp" "-fgnuc-version=4.2.1" "-fskip-odr-
check-in-gmf" "-fcolor-diagnostics" "-vectorize-loops" "-vectorize-slp" "-fe
mbed-offload-object=/tmp/test-f38eb3.out" "-offload-targets=nvptx64-nvidia-
cuda" "-faddrsig" "-D_GCC_HAVE_DWARF2_CFI_ASM=1" "-o" "/tmp/test-835b01.o"
"-x" "ir" "/tmp/test-8a78b4.bc"
```

Solution: LLVM Advisor

LLVM Advisor

- **LLVM Advisor** provides a **unified infrastructure**
 - **Gather** and **organize** compilation data
 - **Analyze** and correlate data
 - **Visualize** on a modern web interface



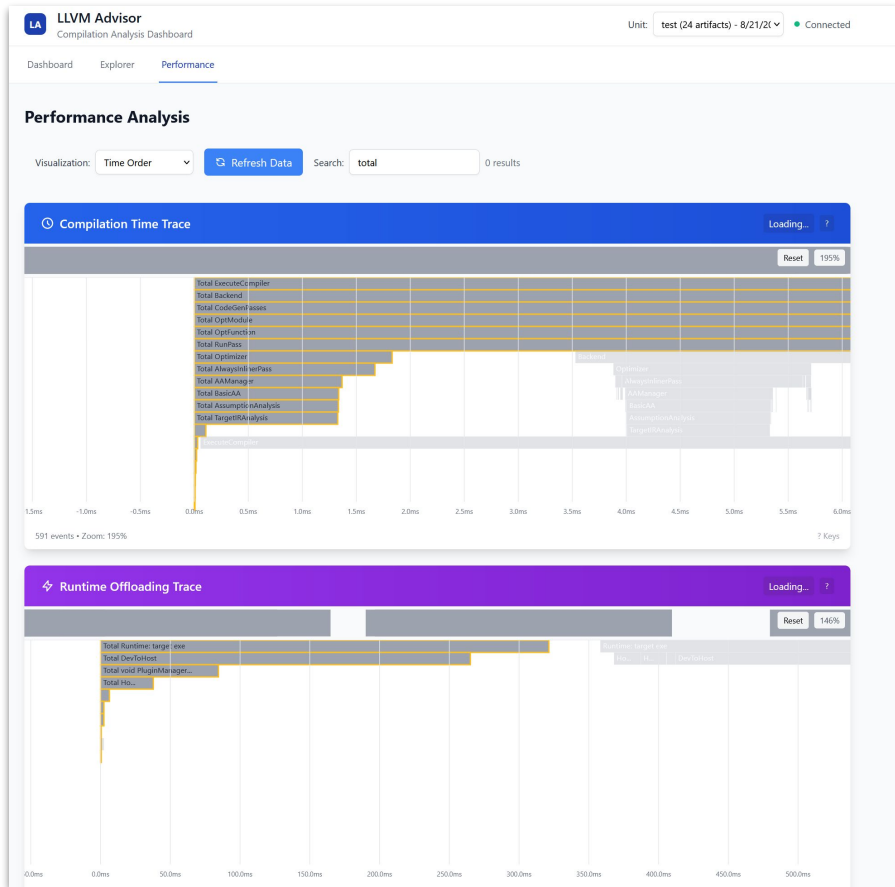
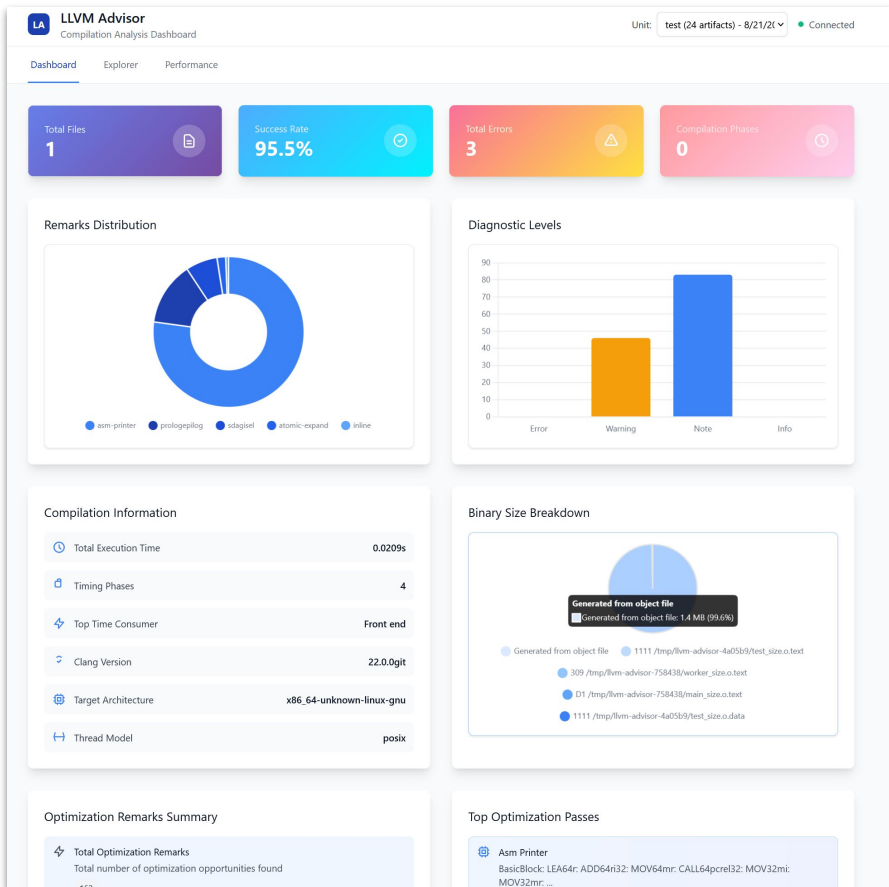
LLVM Advisor

- Compile with data collection

```
$ llvm-advisor clang -O2 -g main.c -o main
```

```
$ tree -a .llvm-advisor
.llvm-advisor/
└─ main/
   └─ main_20250825_143022/ # Timestamped compilation session
      ├── ir/main.ll        # LLVM IR output
      ├── assembly/main.s   # Assembly output
      ├── ast/main.ast      # AST dump
      ├── diagnostics/      # Compiler warnings/errors
      └─ ...                # Additional analysis data
```

LLVM Advisor



Compilation Analysis Dashboard

main_1218 (43 artifacts) - v

- Connected

Explorer

Performance

File: worker.c View: LLVM IR

Diagnostics (6)	Remarks (20)
-----------------	--------------

```
stack bytes in function ''
[prologuepilog]
```

💡 instructions in function
[asm-printer]

```
int sum = 0;
```

```
#pragma omp parallel for reduction(+:sum)
```

💡 FastISel missed terminator
[sdagisel]
Column 5

💡 FastISel missed
[sdagisel]
Column 5

💡 BasicBlock: JMP_1:

Copy Download

```

1 ; ModuleID = '/tmp/workdir-1fc908.bc'
2 source_filename = "worker.c"
3 target datalayout = "e-m:e-p270:32:32-p271:32:32-p272:64:64-i64:64-i128:128-f80:128-n8:16:  

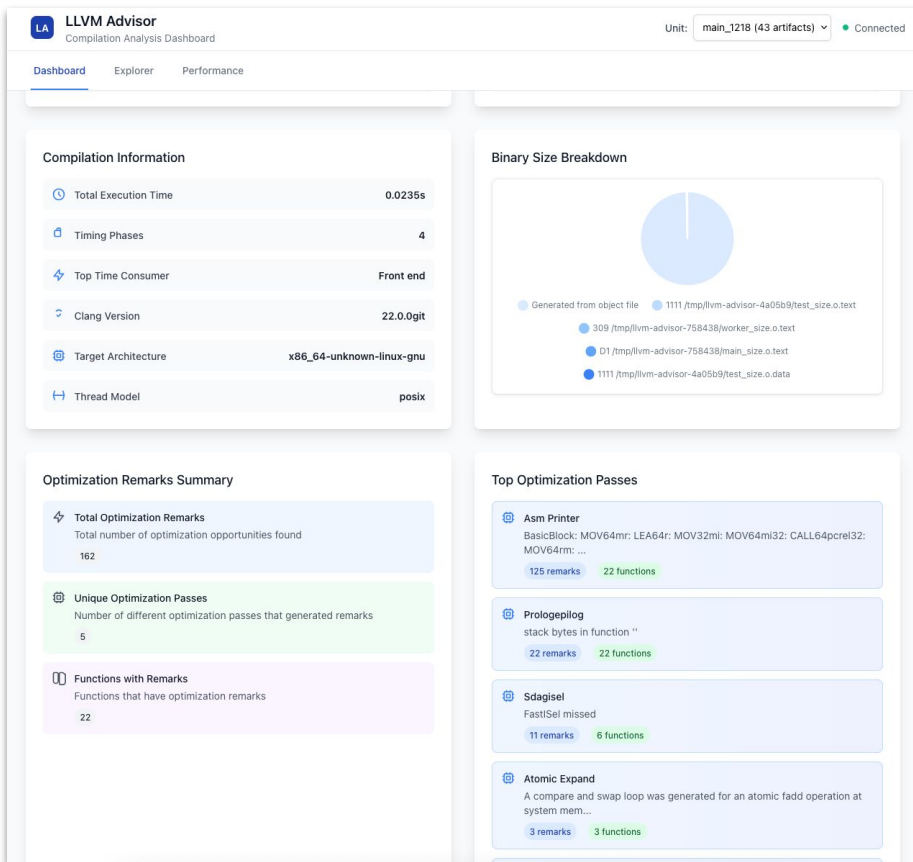
4 target triple = "x86_64-unknown-linux-gnu"
5
6 %struct.idnt_t = type { i32, i32, i32, i32, ptr }
7
8 @0 = private unnamed_addr constant [23 x i8] c";unknown;unknown;0;0;;\00", align 1
9 @1 = private unnamed_addr constant %struct.idnt_t { i32 0, i32 514, i32 0, i32 22, ptr @0 }
10 @gomp_critical_user_.reduction.var = common global [8 x i32] zeroinitializer, align 8
11 @2 = private unnamed_addr constant %struct.idnt_t { i32 0, i32 18, i32 0, i32 22, ptr @0 }
12 @3 = private unnamed_addr constant %struct.idnt_t { i32 0, i32 2, i32 0, i32 22, ptr @0 }
13 @llvm.embedded.object = private constant [2544 x i8] c"\10\FF\10\AD\01\00\00\00\0F\09\00\00  

14 @llvm.compiler.used = appending global [1 x ptr] [ptr @llvm.embedded.object], section ".llv  

15
16 ; Function Attrs: noinline nounwind optnone uwtable
17 define dso_local i32 @parallel_sum(ptr noundef %0, i32 noundef %1) #0 {
18     %3 = alloca ptr, align 8
19     %4 = alloca i32, align 4
20     %5 = alloca i32, align 4
21     store ptr %0, ptr %3, align 8

```

LLVM Advisor



- Who can benefit?
 - Advanced developers
 - Newcomers in LLVM
 - Educators



Questions?

Visit the **poster session** tomorrow!

Try **LLVM Advisor**



<https://llvm-advisor.onrender.com/>

LLVM PR #147451

<https://github.com/llvm/llvm-project/pull/147451>



Miguel Cárdenas

Student at University of Medellin

miguelecsx@gmail.com

@miguelcsx | LLVM, GitHub, LinkedIn