

Why add an IR reader to llvm-debuginfo-analyzer tool

Carlos Alberto Enciso

Reduce the noisiness of comparing the debuginfo in LLVM IR

Reduce the noisiness of comparing the debuginfo in LLVM IR



Sony
Interactive
Entertainment



PlayStation.

```
simplify-cfg.ll
134 = distinct !DILexicalBlock(scope: !26, file: !1,
line: 22, column: 6)
135 = !DILocation(line: 23, column: 26, scope: !34)
136 = !DILocalVariable(name: "x1", scope: !26, file: !1,
line: 20, type: !4)
137 = !DILocation(line: 24, column: 16, scope: !34)
138 = !DILocation(line: 24, column: 26, scope: !34)
139 = !DILocalVariable(name: "x2", scope: !26, file: !1,
line: 20, type: !4)
140 = !DILocation(line: 25, column: 22, scope: !34)
141 = !DILocation(line: 25, column: 17, scope: !34)
142 = !DILocalVariable(name: "w", scope: !26, file: !1,
line: 20, type: !4)
143 = !DILocation(line: 26, column: 14, scope: !26)
144 = !DILocation(line: 26, column: 3, scope: !34)
145 = distinct ![(!45, !32, !46, !47)]
146 = !DILocation(line: 26, column: 20, scope: !26)
147 = !["!llvm.loop.mustprogress"]
148 = !DILocation(line: 28, column: 20, scope: !26)
149 = !DILocation(line: 28, column: 18, scope: !26)
150 = !DILocation(line: 28, column: 30, scope: !26)
151 = !DILocation(line: 28, column: 28, scope: !26)
152 = !DILocation(line: 28, column: 12, scope: !26)
153 = !DILocation(line: 28, column: 7, scope: !26)
154 = !DILocation(line: 29, column: 16, scope: !26)
155 = !DILocation(line: 29, column: 11, scope: !26)
156 = !DILocation(line: 30, column: 16, scope: !26)
157 = !DILocation(line: 30, column: 3, scope: !26)
158 = !DILocation(line: 30, column: 11, scope: !26)
159 = !DILocation(line: 31, column: 1, scope: !26)

slp-vectorizer.ll
134 = distinct !DILexicalBlock(scope: !26, file: !1,
line: 22, column: 6)
135 = !DILocalVariable(name: "x1", scope: !26, file: !1,
line: 20, type: !4)
136 = !DILocation(line: 24, column: 16, scope: !34)
137 = !DILocation(line: 23, column: 26, scope: !34)
138 = !DILocalVariable(name: "x2", scope: !26, file: !1,
line: 20, type: !4)
139 = !DILocation(line: 25, column: 22, scope: !34)
140 = !DILocation(line: 25, column: 17, scope: !34)
141 = !DILocalVariable(name: "w", scope: !26, file: !1,
line: 20, type: !4)
142 = !DILocation(line: 26, column: 14, scope: !26)
143 = !DILocation(line: 26, column: 3, scope: !34)
144 = distinct ![(!44, !32, !45, !46)]
145 = !DILocation(line: 26, column: 20, scope: !26)
146 = !["!llvm.loop.mustprogress"]
147 = !DILocation(line: 28, column: 20, scope: !26)
148 = !DILocation(line: 28, column: 18, scope: !26)
149 = !DILocation(line: 28, column: 30, scope: !26)
150 = !DILocation(line: 28, column: 28, scope: !26)
151 = !DILocation(line: 28, column: 12, scope: !26)
152 = !DILocation(line: 28, column: 7, scope: !26)
153 = !DILocation(line: 29, column: 16, scope: !26)
154 = !DILocation(line: 29, column: 11, scope: !26)
155 = !DILocation(line: 31, column: 1, scope: !26)
```

IR changes: comparison tool

Reference: 'simplify-cfg.ll'
Target: 'slp-vectorizer.ll'

(2) Missing Symbols:

- 20 {Variable} 'x1' -> 'float'
- 20 {Variable} 'x2' -> 'float'

(5) Missing Lines:

- 24 {Line}
- 30 {Line}
- 30 {Line}
- 30 {Line}
- 24 {Line}

Reference: 'simplify-cfg.ll'
Target: 'slp-vectorizer.ll'

Logical View:

{File} 'simplify-cfg.ll'

```
{CompileUnit} 'test.cpp'
{Function} extern not_inlined 'RandF32' -> 'float'
{Variable} 'uRand' -> 'U32'
{Variable} 'fRand' -> 'F32'
{Function} extern not_inlined 'randGauss' -> 'void'
{Parameter} 'work' -> '* float'
{Variable} 'w' -> 'float'
- 20 {Variable} 'x1' -> 'float'
- 20 {Variable} 'x2' -> 'float'
```

IR changes: llvm-debuginfo-analyzer

Why add an IR reader to llvm-debuginfo-analyzer tool



LLVM and debug information

- Different formats, toolchain, tools
- Common problems

Analyzing optimizer IR output

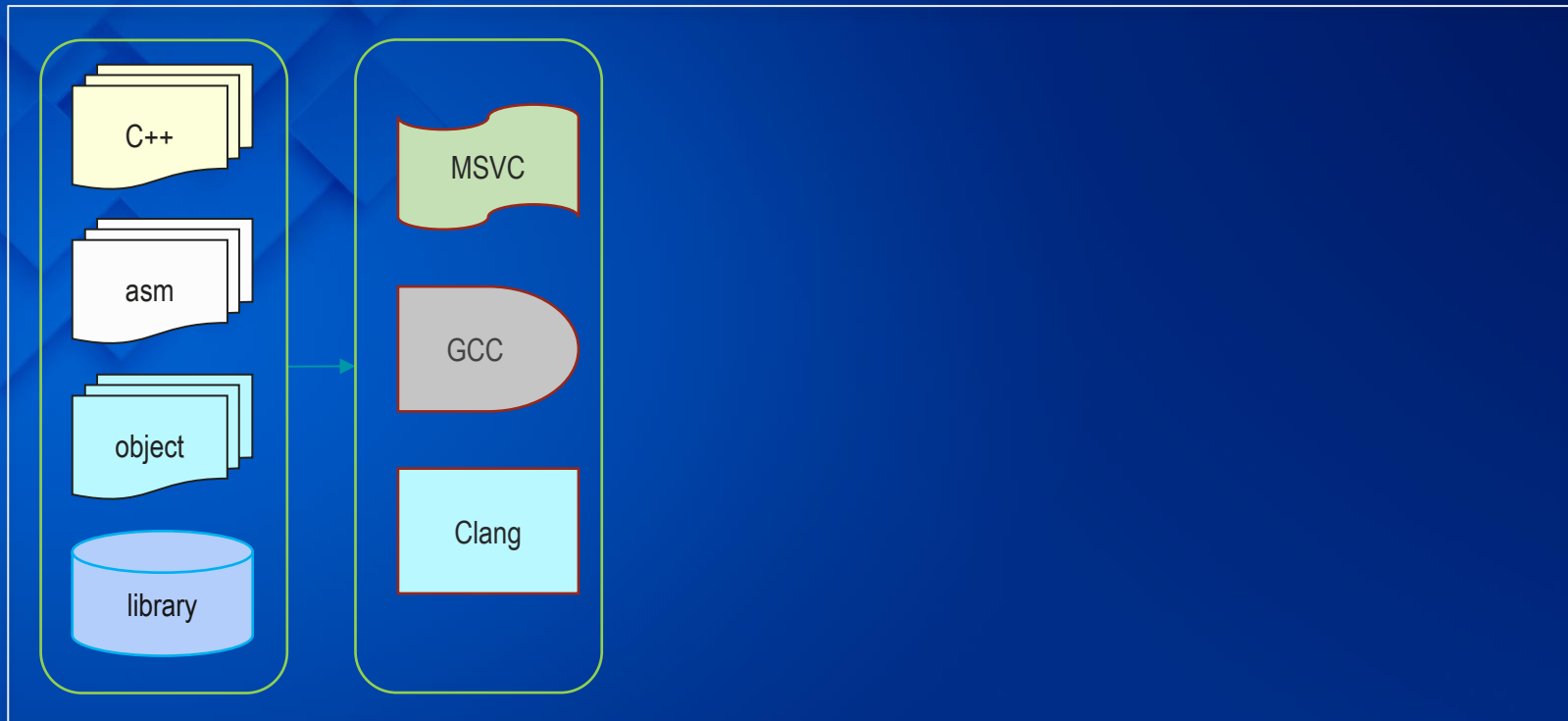
- IR before/after an optimizer pass
- Test case: SLP Vectorizer pass drops debug information

llvm-debuginfo-analyzer

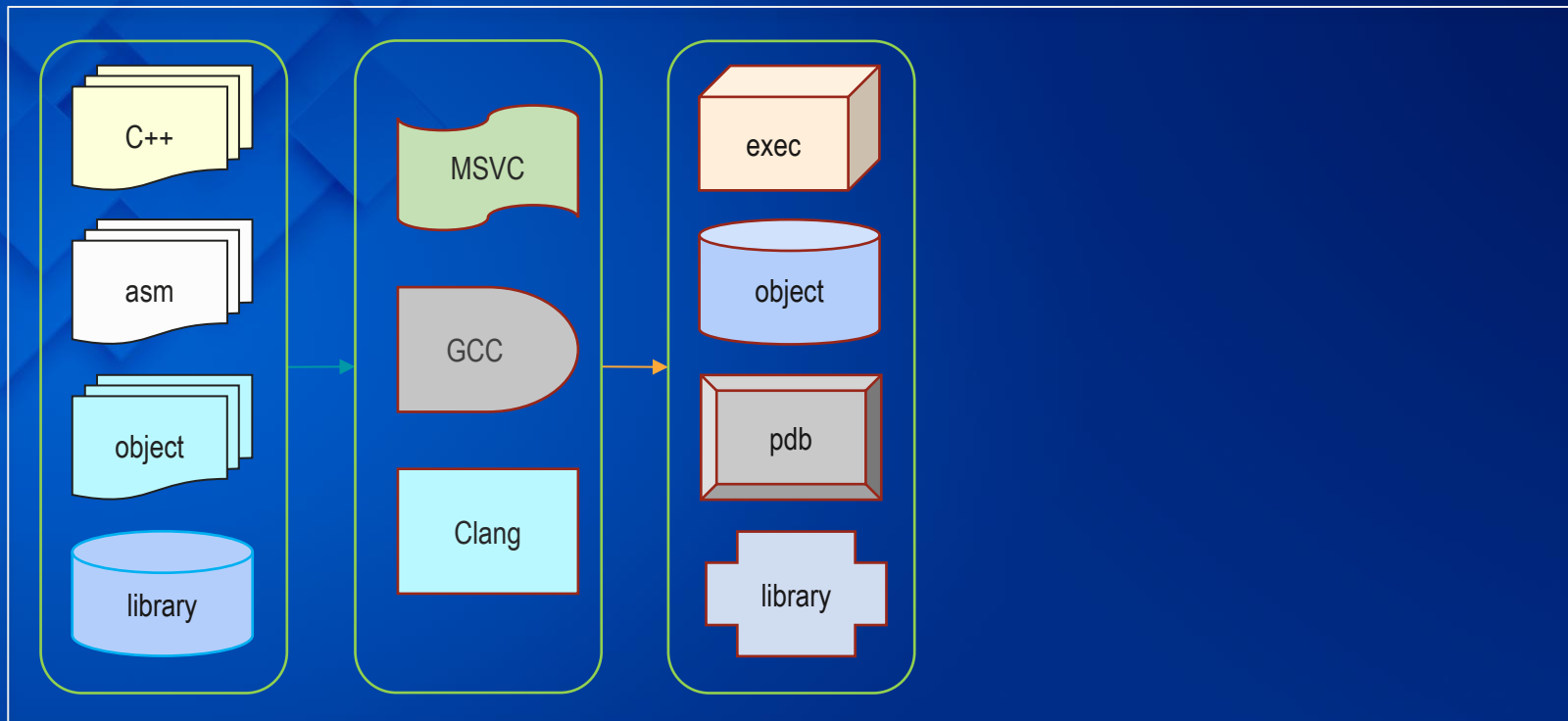
- Basic introduction
- Print logical view
- Compare logical view



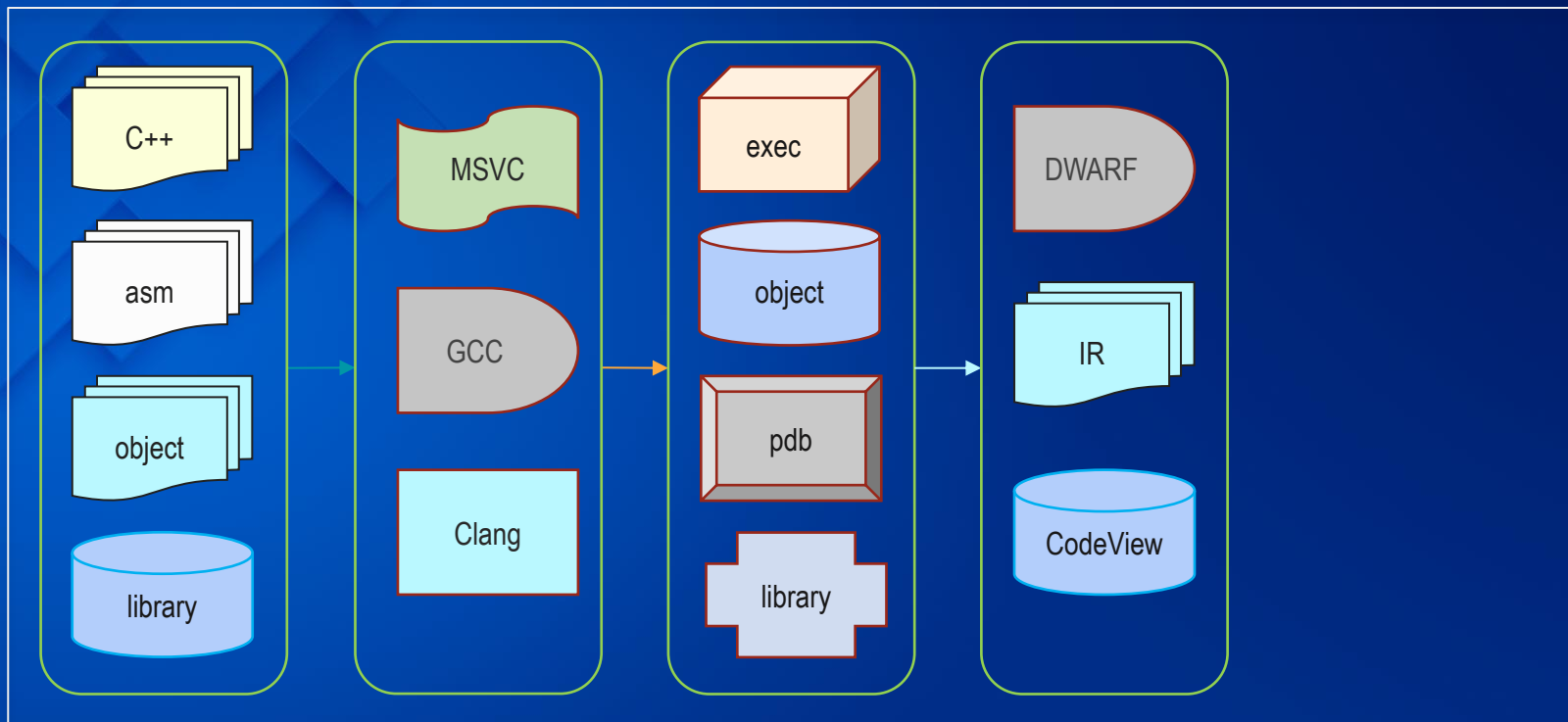
Different inputs



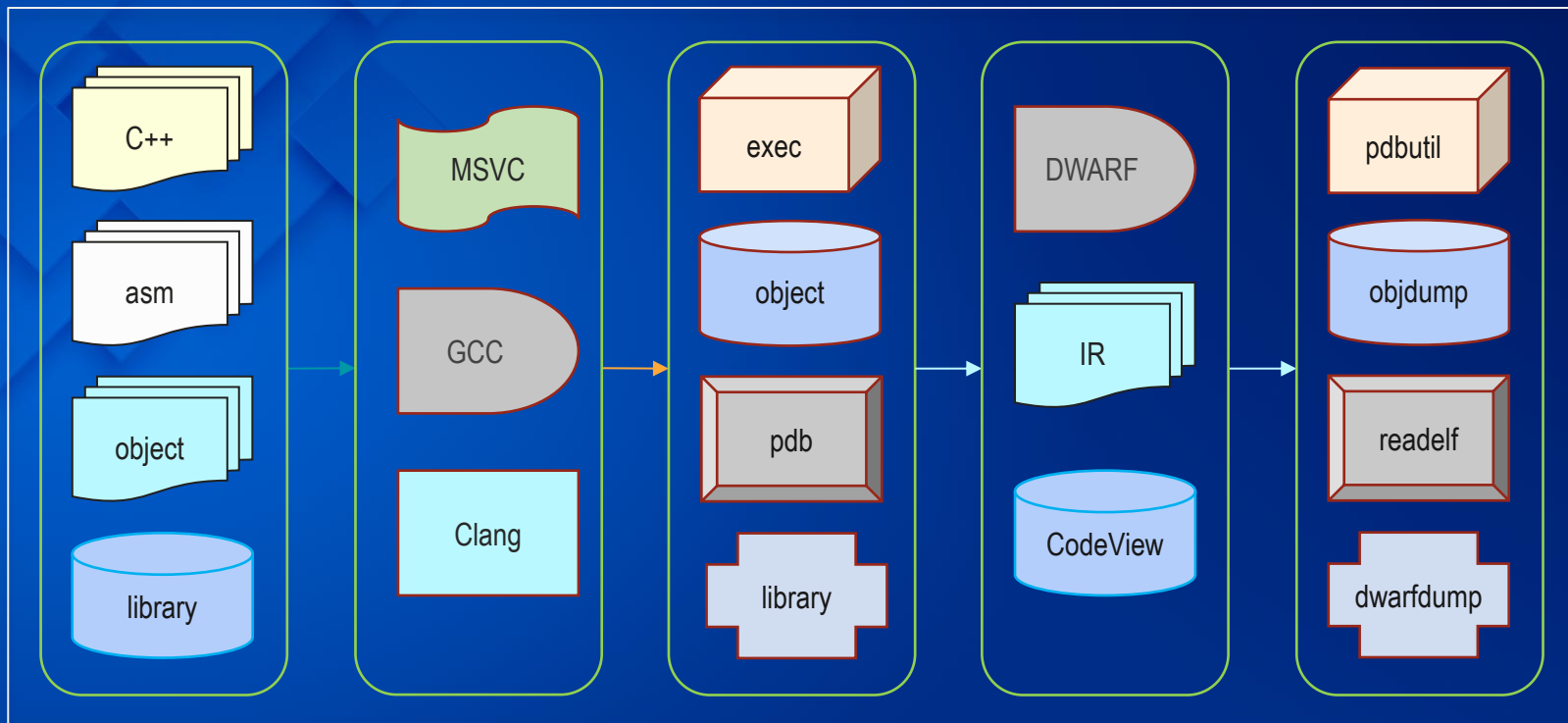
Different toolchains



Different binary formats



Different debug information formats



Different debug information tools

Does the debug information represent the original source

- Which variables are dropped due to optimization
- Why I cannot stop at a particular line
- Which lines are associated to a specific code range
- Size changes due to toolchain features

Does the debug information represent the original source

- Which variables are dropped due to optimization
- Why I cannot stop at a particular line
- Which lines are associated to a specific code range
- Size changes due to toolchain features

Semantic differences in the generated debug information

- By different toolchain versions (same platform)
- Same source code compiled in different platforms

IR before/after an optimizer pass

The IR output is very rich and noisy

- The metadata identifiers changes between passes
- Difficult to see the changes using a comparison tool
- Requires knowledge about the IR and passes
- By bisection, we can pin down which pass causes the change

IR before/after an optimizer pass

The IR output is very rich and noisy

- The metadata identifiers changes between passes
- Difficult to see the changes using a comparison tool
- Requires knowledge about the IR and passes
- By bisection, we can pin down which pass causes the change

The optimizer includes a wide set of debug printing options

- `--print-before`, `--print-before-all`, `--print-after`, `--print-after-all`, etc.

IR before/after an optimizer pass

The IR output is very rich and noisy

- The metadata identifiers changes between passes
- Difficult to see the changes using a comparison tool
- Requires knowledge about the IR and passes
- By bisection, we can pin down which pass causes the change

The optimizer includes a wide set of debug printing options

- `--print-before`, `--print-before-all`, `--print-after`, `--print-after-all`, etc.

Test case: SLP Vectorizer pass drops debug information

- [DebugInfo@O2] <https://github.com/llvm/llvm-project/issues/45507>
- Drops location information for local variables x1 and x2
- Generate IR after Simplify CFG and SLP Vectorizer passes

IR metadata - Simplify CFG and SLP Vectorizer passes



Sony
Interactive
Entertainment



PlayStation.

```
1 // Compile with clang -g -O2.
2 // The SLP Vectorizer pass drops location
3 // information for the local variables:
4 // x1 and x2.
5
6 typedef unsigned int U32;
7 typedef float F32;
8 extern "C" double log(float);
9 extern "C" double sqrt(float);
10
11 extern unsigned RandU32();
12
13 float RandF32() {
14     U32 uRand = RandU32();
15     F32 fRand = ((F32)uRand / 4294967810.0f);
16     return (fRand);
17 }
18
19 void randGauss(float work[2]) {
20     float x1, x2, w;
21
22     do {
23         x1 = 2.f * RandF32() - 1.f;
24         x2 = 2.f * RandF32() - 1.f;
25         w = x1 * x1 + x2 * x2;
26     } while (w >= 1.f);
27
28     w = sqrt((-2.f * log(w)) / w);
29     work[0] = x1 * w;
30     work[1] = x2 * w;
31 }
```

IR metadata - Simplify CFG and SLP Vectorizer passes



Sony
Interactive
Entertainment



PlayStation.

```
1 // Compile with clang -g -O2.
2 // The SLP Vectorizer pass drops location
3 // information for the local variables:
4 // x1 and x2.
5
6 typedef unsigned int U32;
7 typedef float F32;
8 extern "C" double log(float);
9 extern "C" double sqrt(float);
10
11 extern unsigned RandU32();
12
13 float RandF32() {
14     U32 uRand = RandU32();
15     F32 fRand = ((F32)uRand / 4294967810.0f);
16     return (fRand);
17 }
18
19 void randGauss(float work[2]) {
20     float x1, x2, w;
21
22     do {
23         x1 = 2.f * RandF32() - 1.f;
24         x2 = 2.f * RandF32() - 1.f;
25         w = x1 * x1 + x2 * x2;
26     } while (w >= 1.f);
27
28     w = sqrt((-2.f * log(w)) / w);
29     work[0] = x1 * w;
30     work[1] = x2 * w;
31 }
```

```
!35 = !DILocation(line: 23, column: 26, scope: !34)
!36 = !DILocalVariable(name: "x1", scope: !26, file:
!1, line: 20, type: !4)
!37 = !DILocation(line: 24, column: 16, scope: !34)
!38 = !DILocation(line: 24, column: 26, scope: !34)
!39 = !DILocalVariable(name: "x2", scope: !26, file:
!1, line: 20, type: !4)
!40 = !DILocation(line: 25, column: 22, scope: !34)
!41 = !DILocation(line: 25, column: 17, scope: !34)
!42 = !DILocalVariable(name: "w", scope: !26, file:
!1, line: 20, type: !4)
!43 = !DILocation(line: 26, column: 14, scope: !26)
!44 = !DILocation(line: 26, column: 3, scope: !34)
!45 = distinct !{!45, !32, !46, !47}
!46 = !DILocation(line: 26, column: 20, scope: !26)
!47 = !{"llvm.loop.mustprogress"}
!48 = !DILocation(line: 28, column: 20, scope: !26)
!49 = !DILocation(line: 28, column: 18, scope: !26)
!50 = !DILocation(line: 28, column: 30, scope: !26)
!51 = !DILocation(line: 28, column: 28, scope: !26)
!52 = !DILocation(line: 28, column: 12, scope: !26)
!53 = !DILocation(line: 28, column: 7, scope: !26)
!54 = !DILocation(line: 29, column: 16, scope: !26)
!55 = !DILocation(line: 29, column: 11, scope: !26)
!56 = !DILocation(line: 30, column: 16, scope: !26)
!57 = !DILocation(line: 30, column: 3, scope: !26)
!58 = !DILocation(line: 30, column: 11, scope: !26)
!59 = !DILocation(line: 31, column: 1, scope: !26)
```

IR metadata after Simplify CFG

IR metadata - Simplify CFG and SLP Vectorizer passes



```
1 // Compile with clang -g -O2.
2 // The SLP Vectorizer pass drops location
3 // information for the local variables:
4 // x1 and x2.
5
6 typedef unsigned int U32;
7 typedef float F32;
8 extern "C" double log(float);
9 extern "C" double sqrt(float);
10
11 extern unsigned RandU32();
12
13 float RandF32() {
14     U32 uRand = RandU32();
15     F32 fRand = ((F32)uRand / 4294967810.0f);
16     return (fRand);
17 }
18
19 void randGauss(float work[2]) {
20     float x1, x2, w;
21
22     do {
23         x1 = 2.f * RandF32() - 1.f;
24         x2 = 2.f * RandF32() - 1.f;
25         w = x1 * x1 + x2 * x2;
26     } while (w >= 1.f);
27
28     w = sqrt((-2.f * log(w)) / w);
29     work[0] = x1 * w;
30     work[1] = x2 * w;
31 }
```

```
!35 = !DILocation(line: 23, column: 26, scope: !34)
!36 = !DILocalVariable(name: "x1", scope: !26, file:
!1, line: 20, type: !4)
!37 = !DILocation(line: 24, column: 16, scope: !34)
!38 = !DILocation(line: 24, column: 26, scope: !34)
!39 = !DILocalVariable(name: "x2", scope: !26, file:
!1, line: 20, type: !4)
!40 = !DILocation(line: 25, column: 22, scope: !34)
!41 = !DILocation(line: 25, column: 17, scope: !34)
!42 = !DILocalVariable(name: "w", scope: !26, file:
!1, line: 20, type: !4)
!43 = !DILocation(line: 26, column: 14, scope: !26)
!44 = !DILocation(line: 26, column: 3, scope: !34)
!45 = distinct !{!45, !32, !46, !47}
!46 = !DILocation(line: 26, column: 20, scope: !26)
!47 = !{"llvm.loop.mustprogress"}
!48 = !DILocation(line: 28, column: 20, scope: !26)
!49 = !DILocation(line: 28, column: 18, scope: !26)
!50 = !DILocation(line: 28, column: 30, scope: !26)
!51 = !DILocation(line: 28, column: 28, scope: !26)
!52 = !DILocation(line: 28, column: 12, scope: !26)
!53 = !DILocation(line: 28, column: 7, scope: !26)
!54 = !DILocation(line: 29, column: 16, scope: !26)
!55 = !DILocation(line: 29, column: 11, scope: !26)
!56 = !DILocation(line: 30, column: 16, scope: !26)
!57 = !DILocation(line: 30, column: 3, scope: !26)
!58 = !DILocation(line: 30, column: 11, scope: !26)
!59 = !DILocation(line: 31, column: 1, scope: !26)
```

IR metadata after Simplify CFG

```
!35 = !DILocalVariable(name: "x1", scope: !26, file:
!1, line: 20, type: !4)
!36 = !DILocation(line: 24, column: 16, scope: !34)
!37 = !DILocation(line: 23, column: 26, scope: !34)
!38 = !DILocalVariable(name: "x2", scope: !26, file:
!1, line: 20, type: !4)
!39 = !DILocation(line: 25, column: 22, scope: !34)
!40 = !DILocation(line: 25, column: 17, scope: !34)
!41 = !DILocalVariable(name: "w", scope: !26, file:
!1, line: 20, type: !4)
!42 = !DILocation(line: 26, column: 14, scope: !26)
!43 = !DILocation(line: 26, column: 3, scope: !34)
!44 = distinct !{!44, !32, !45, !46}
!45 = !DILocation(line: 26, column: 20, scope: !26)
!46 = !{"llvm.loop.mustprogress"}
!47 = !DILocation(line: 28, column: 20, scope: !26)
!48 = !DILocation(line: 28, column: 18, scope: !26)
!49 = !DILocation(line: 28, column: 30, scope: !26)
!50 = !DILocation(line: 28, column: 28, scope: !26)
!51 = !DILocation(line: 28, column: 12, scope: !26)
!52 = !DILocation(line: 28, column: 7, scope: !26)
!53 = !DILocation(line: 29, column: 16, scope: !26)
!54 = !DILocation(line: 29, column: 11, scope: !26)
!55 = !DILocation(line: 31, column: 1, scope: !26)
```

IR metadata after SLP Vectorizer



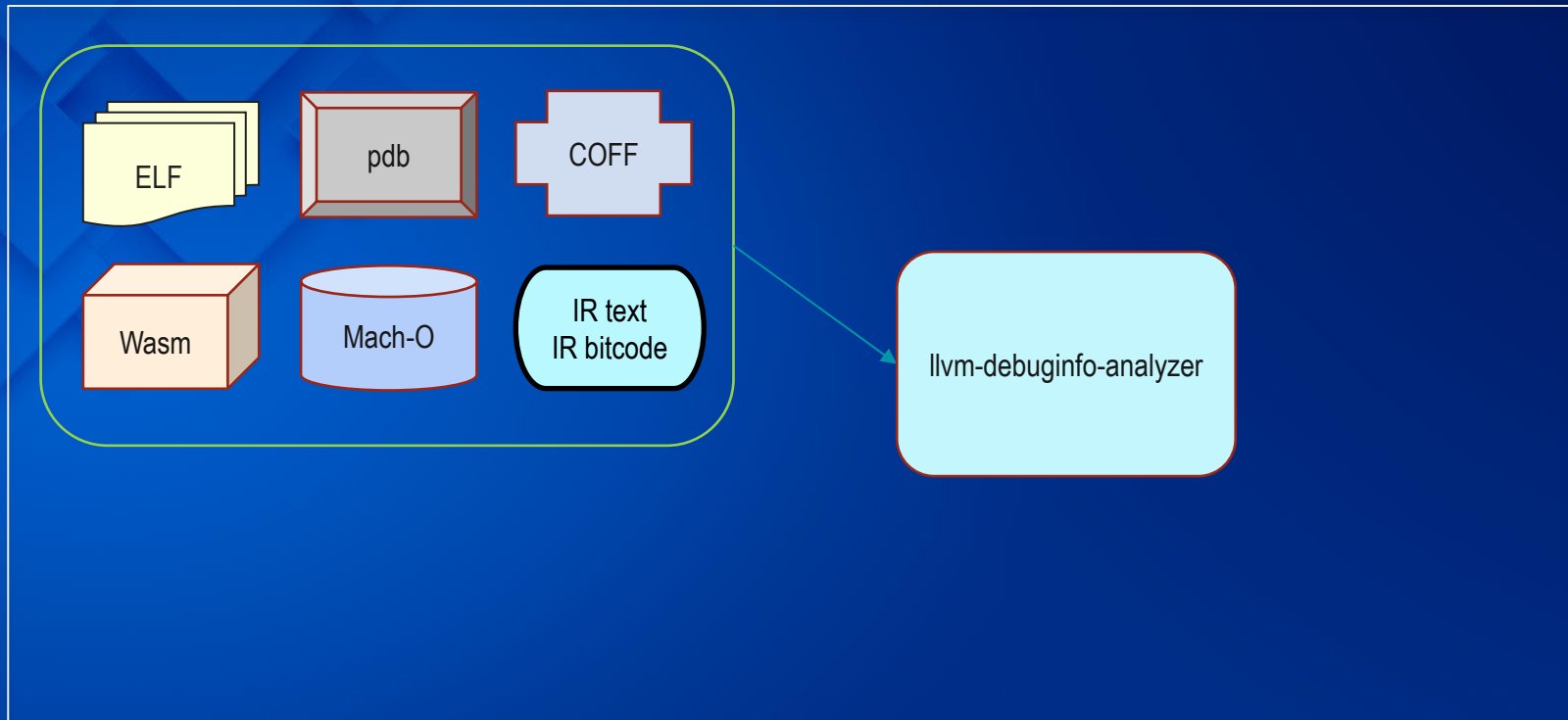
```
simplify-cfg.ll
134 = distinct IDILexicalBlock(scope: 126, file: 11,
line: 22, column: 6)
135 = IDILocation(line: 23, column: 26, scope: 134)
136 = IDILocalVariable(name: "x1", scope: 126, file: 11,
line: 20, type: i4)
137 = IDILocation(line: 24, column: 16, scope: 134)
138 = IDILocation(line: 24, column: 26, scope: 134)
139 = IDILocalVariable(name: "x2", scope: 126, file: 11,
line: 20, type: i4)
140 = IDILocation(line: 25, column: 22, scope: 134)
141 = IDILocation(line: 25, column: 17, scope: 134)
142 = IDILocalVariable(name: "w", scope: 126, file: 11,
line: 20, type: i4)
143 = IDILocation(line: 26, column: 14, scope: 126)
144 = IDILocation(line: 26, column: 3, scope: 134)
145 = distinct !([145, 132, 146, 147])
146 = IDILocation(line: 26, column: 20, scope: 126)
147 = !([!llvm.loop.mustprogress"])
148 = IDILocation(line: 28, column: 20, scope: 126)
149 = IDILocation(line: 28, column: 18, scope: 126)
150 = IDILocation(line: 28, column: 30, scope: 126)
151 = IDILocation(line: 28, column: 28, scope: 126)
152 = IDILocation(line: 28, column: 12, scope: 126)
153 = IDILocation(line: 28, column: 7, scope: 126)
154 = IDILocation(line: 29, column: 16, scope: 126)
155 = IDILocation(line: 29, column: 11, scope: 126)
156 = IDILocation(line: 30, column: 16, scope: 126)
157 = IDILocation(line: 30, column: 3, scope: 126)
158 = IDILocation(line: 30, column: 11, scope: 126)
159 = IDILocation(line: 31, column: 1, scope: 126)

slp-vectorizer.ll
134 = distinct IDILexicalBlock(scope: 126, file: 11,
line: 22, column: 6)
135 = IDILocalVariable(name: "x1", scope: 126, file: 11,
line: 20, type: i4)
136 = IDILocation(line: 24, column: 16, scope: 134)
137 = IDILocation(line: 23, column: 26, scope: 134)
138 = IDILocalVariable(name: "x2", scope: 126, file: 11,
line: 20, type: i4)
139 = IDILocation(line: 25, column: 22, scope: 134)
140 = IDILocation(line: 25, column: 17, scope: 134)
141 = IDILocalVariable(name: "w", scope: 126, file: 11,
line: 20, type: i4)
142 = IDILocation(line: 26, column: 14, scope: 126)
143 = IDILocation(line: 26, column: 3, scope: 134)
144 = distinct !([144, 132, 145, 146])
145 = IDILocation(line: 26, column: 20, scope: 126)
146 = !([!llvm.loop.mustprogress"])
147 = IDILocation(line: 28, column: 20, scope: 126)
148 = IDILocation(line: 28, column: 18, scope: 126)
149 = IDILocation(line: 28, column: 30, scope: 126)
150 = IDILocation(line: 28, column: 28, scope: 126)
151 = IDILocation(line: 28, column: 12, scope: 126)
152 = IDILocation(line: 28, column: 7, scope: 126)
153 = IDILocation(line: 29, column: 16, scope: 126)
154 = IDILocation(line: 29, column: 11, scope: 126)
155 = IDILocation(line: 31, column: 1, scope: 126)
```

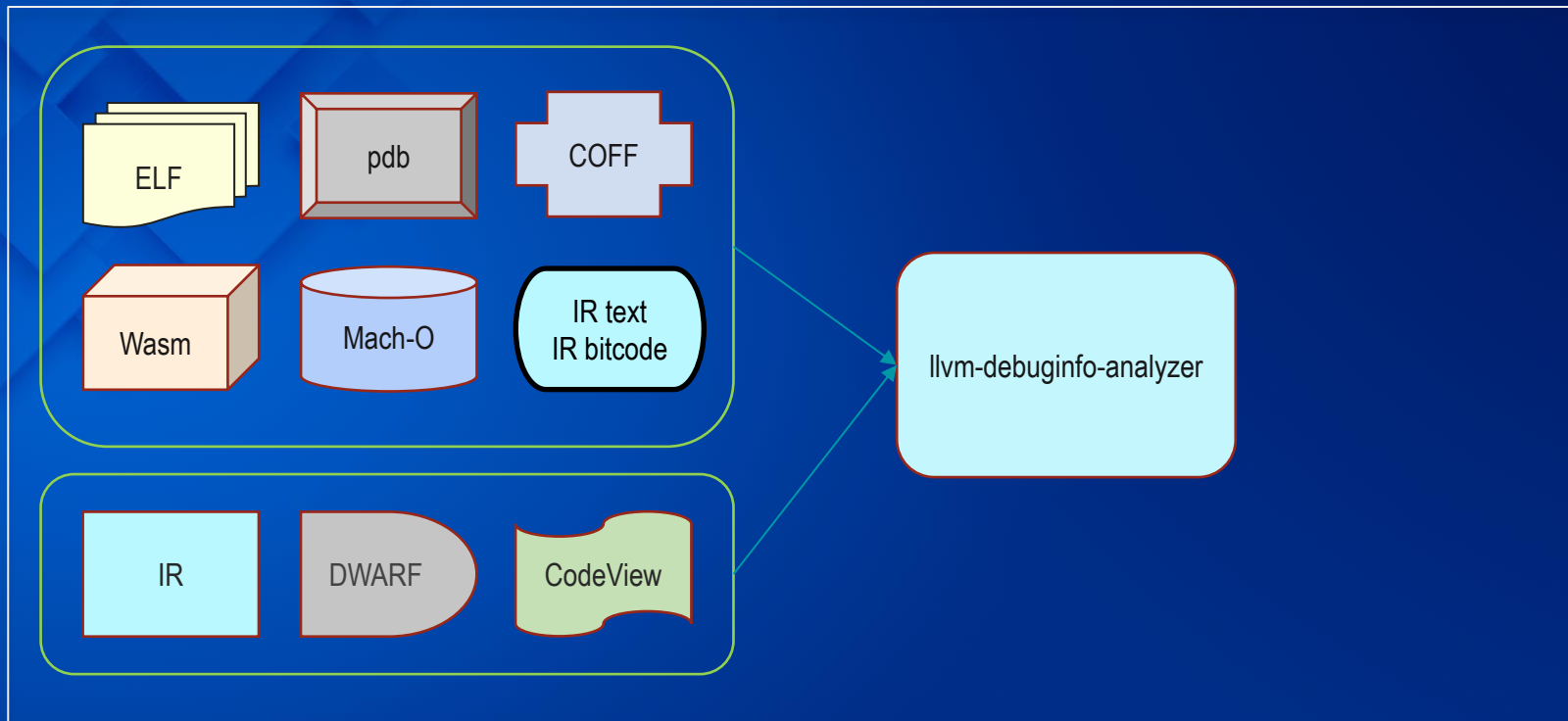
Ln: 105 Col: 1/52 Ch: 1/52 EOL: CRLF Windows-1252 Win

Ln: 131-132 Windows-1252 Win

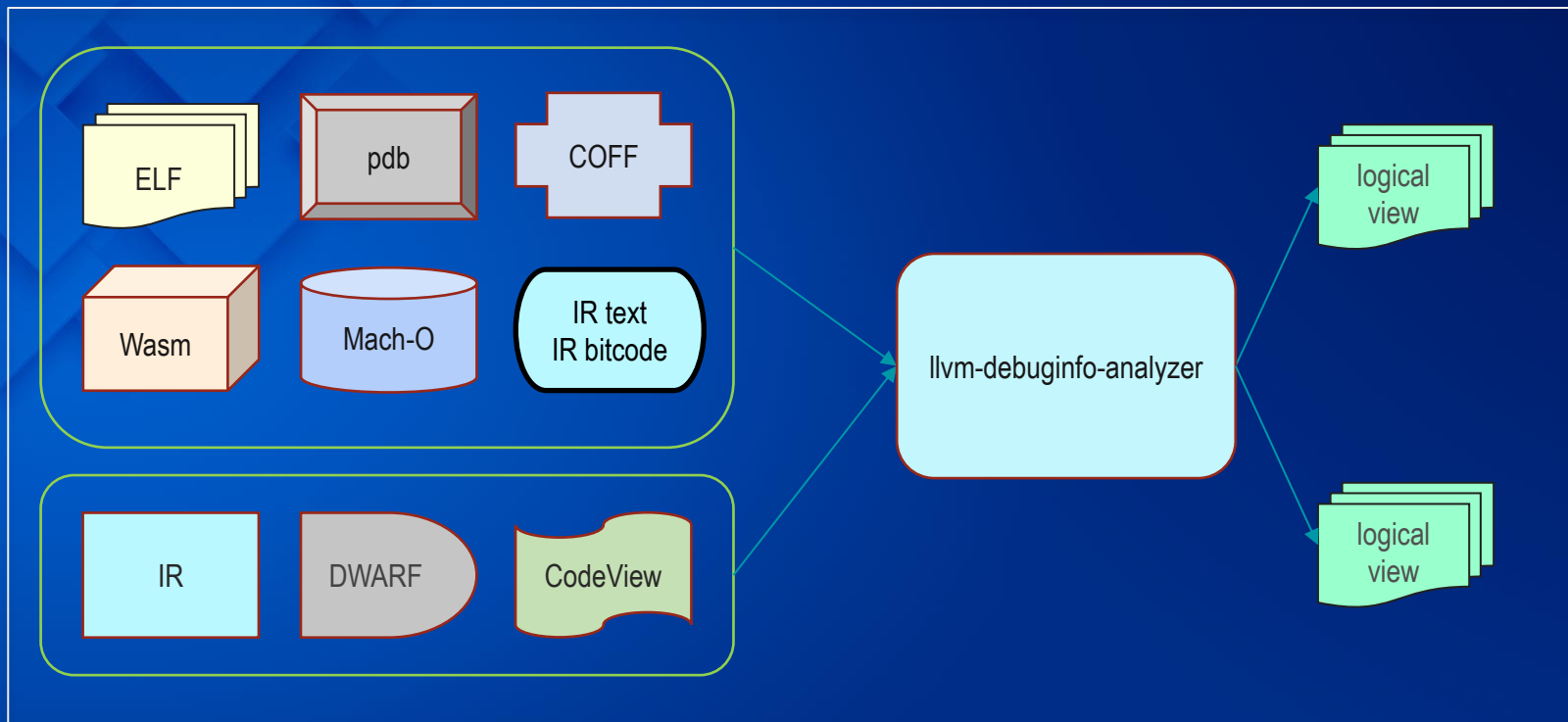
IR changes: comparison tool



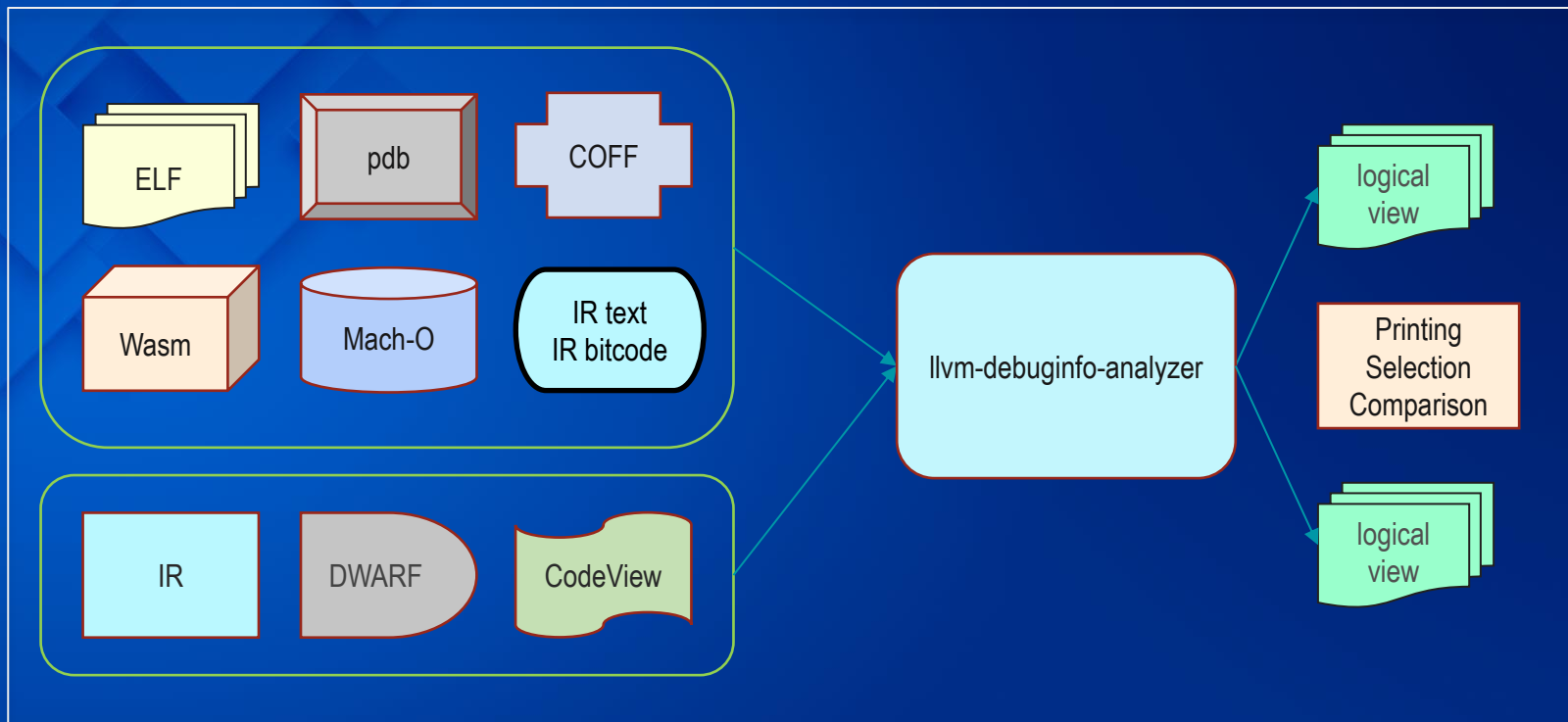
Supported binary file formats



Supported debug information formats



Logical view is a high-level representation of the debug information



Logical views can be printed, selected and compared

```
DW_TAG_compile_unit
  DW_AT_producer    ("clang")
  DW_AT_language    (DW_LANG_C_plus_plus_14)
  DW_AT_name        ("hello-world.cpp")
  DW_AT_low_pc      (0x0000000000000000)
  DW_AT_high_pc     (0x0000000000000014)
  DW_AT_addr_base   (0x00000008)

  DW_TAG_variable
    DW_AT_type      (0x0000002d "const char[13]")
    DW_AT_decl_line (4)
    DW_AT_location  (DW_OP_addrx 0x0)

  DW_TAG_array_type
    DW_AT_type      (0x00000039 "const char")

  DW_TAG_subrange_type
    DW_AT_type      (0x00000042 "__ARRAY_SIZE_TYPE__")
    DW_AT_count     (0x0d)

  DW_TAG_subprogram
    DW_AT_low_pc     (0x0000000000000000)
    DW_AT_high_pc    (0x0000000000000014)
    DW_AT_frame_base (DW_OP_reg6 RBP)
    DW_AT_linkage_name ("_Z3foov")
    DW_AT_name       ("foo")
    DW_AT_decl_line  (3)
    DW_AT_external   (true)
```

DWARF debug information

```
DW_TAG_compile_unit
  DW_AT_producer    ("clang")
  DW_AT_language    (DW_LANG_C_plus_plus_14)
  DW_AT_name        ("hello-world.cpp")
  DW_AT_low_pc      (0x0000000000000000)
  DW_AT_high_pc     (0x0000000000000014)
  DW_AT_addr_base   (0x00000000)

DW_TAG_variable
  DW_AT_type        (0x0000002d "const char[13]")
  DW_AT_decl_line   (4)
  DW_AT_location    (DW_OP_addrx 0x0)

DW_TAG_array_type
  DW_AT_type        (0x00000039 "const char")

DW_TAG_subrange_type
  DW_AT_type        (0x00000042 "__ARRAY_SIZE_TYPE__")
  DW_AT_count       (0x0d)

DW_TAG_subprogram
  DW_AT_low_pc      (0x0000000000000000)
  DW_AT_high_pc     (0x0000000000000014)
  DW_AT_frame_base  (DW_OP_reg6 RBP)
  DW_AT_linkage_name ("_Z3foov")
  DW_AT_name        ("foo")
  DW_AT_decl_line   (3)
  DW_AT_external    (true)
```

DWARF debug information

```
Types (.debug$T)
=====
0x1000 | LF_ARGLIST [size = 8]
0x1001 | LF_PROCEDURE [size = 16]
        return type = 0x0003 (void), # args = 0,
        param list = 0x1000
        calling conv = cdecl, options = None
0x1002 | LF_FUNC_ID [size = 16]
        name = foo, type = 0x1001, parent scope =
<no type>
0x1004 | LF_STRING_ID [size = 24] ID: <no type>,
String: hello-world.cpp

Symbols
=====
Mod 0000 | `.debug$S`:
        0 | S_OBJNAME [size = 64] sig=0, `hello-
        world-clang-cv.o`
        0 | S_COMPILE3 [size = 156]
        machine = intel x86-x64, Ver = clang
        version 21.0.0, language = c++
        frontend = 21.0.0.0 flags = none
        0 | S_GPROC32_ID [size = 44] `foo`
        parent = 0, end = 0, addr = 0000:0000
        type = `0x1002 (foo)`, debug start = 0,
        debug end = 0, flags = noinline | opt debuginfo
        0 | S_FRAMEPROC [size = 32]
        size = 40, padding size = 0 padding = 0
        bytes of callee saved registers = 0,
        exception handler addr = 0000:0000
        local fp reg = RSP, param fp reg = RSP
        flags = safe buffers
        0 | S_PROC_ID_END [size = 4]
        0 | S_BUILDINFO [size = 8] BuildId = `0x1008`
```

CodeView debug information

DWARF & CodeView canonical logical view



```
DW_TAG_compile_unit
DW_AT_producer    ("clang")
DW_AT_language    (DW_LANG_C_plus_plus_14)
DW_AT_name        ("hello-world.cpp")
DW_AT_low_pc      (0x0000000000000000)
DW_AT_high_pc     (0x0000000000000014)
DW_AT_addr_base   (0x00000000)

DW_TAG_variable
DW_AT_type        (0x0000002d "const char[13]")
DW_AT_decl_line   (4)
DW_AT_location    (DW_OP_addrx 0x0)

DW_TAG_array_type
DW_AT_type        (0x00000039 "const char")

DW_TAG_subrange_type
DW_AT_type        (0x00000042 "__ARRAY_SIZE_TYPE__")
DW_AT_count       (0x0d)

DW_TAG_subprogram
DW_AT_low_pc      (0x0000000000000000)
DW_AT_high_pc     (0x0000000000000014)
DW_AT_frame_base  (DW_OP_reg6 RBP)
DW_AT_linkage_name ("__Z3foov")
DW_AT_name        ("foo")
DW_AT_decl_line   (3)
DW_AT_external    (true)
```

DWARF debug information

```
Logical View:
{File} 'hello-world-clang.o'

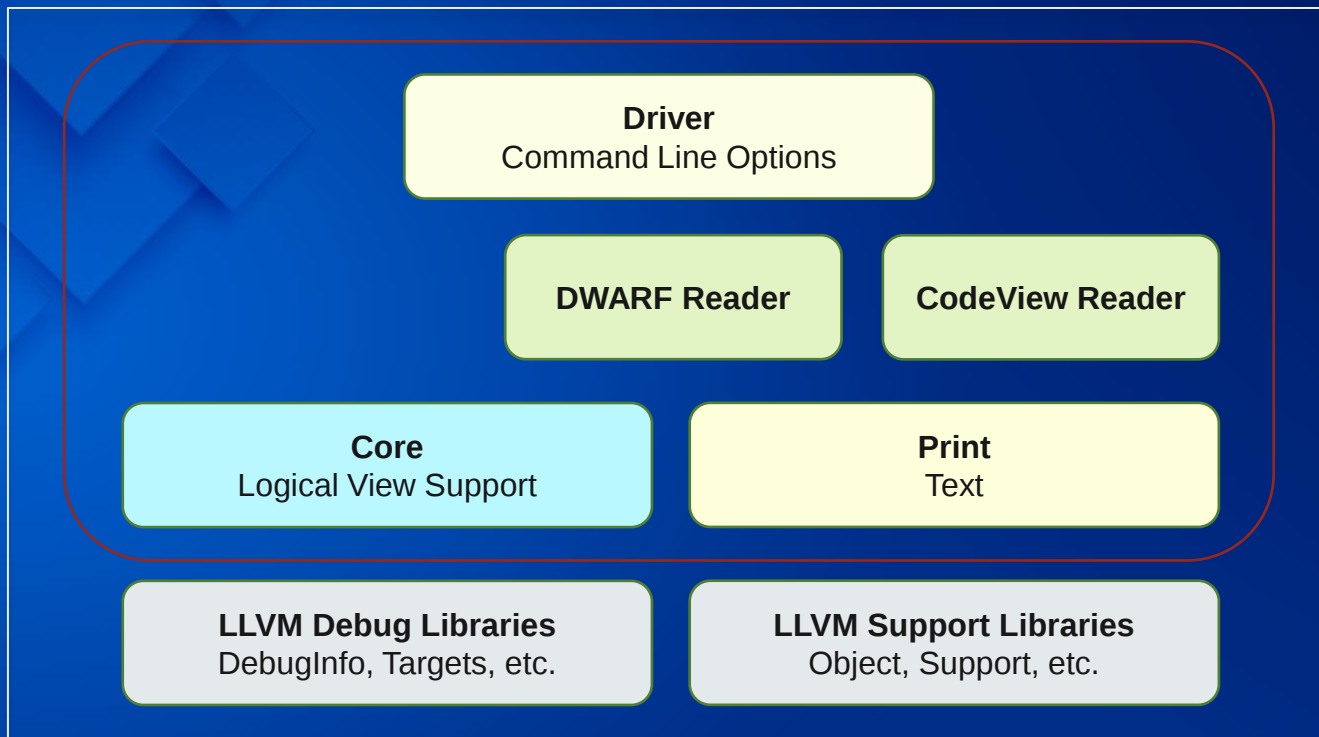
{CompileUnit} 'hello-world.cpp'
{Array} 'const char [13]'
{Function} not_inlined 'foo' -> 'void'
3 {Line}
  {Code} 'pushq          %rbp'
  {Code} 'movq          %rsp, %rbp'
4 {Line}
  {Code} 'leaq          (%rip), %rdi'
  {Code} 'movb          $0x0, %al'
  {Code} 'callq         0x0'
5 {Line}
  {Code} 'popq          %rbp'
  {Code} 'retq'
5 {Line}
4 {Variable} '' -> 'const char [13]'
```

Canonical logical view

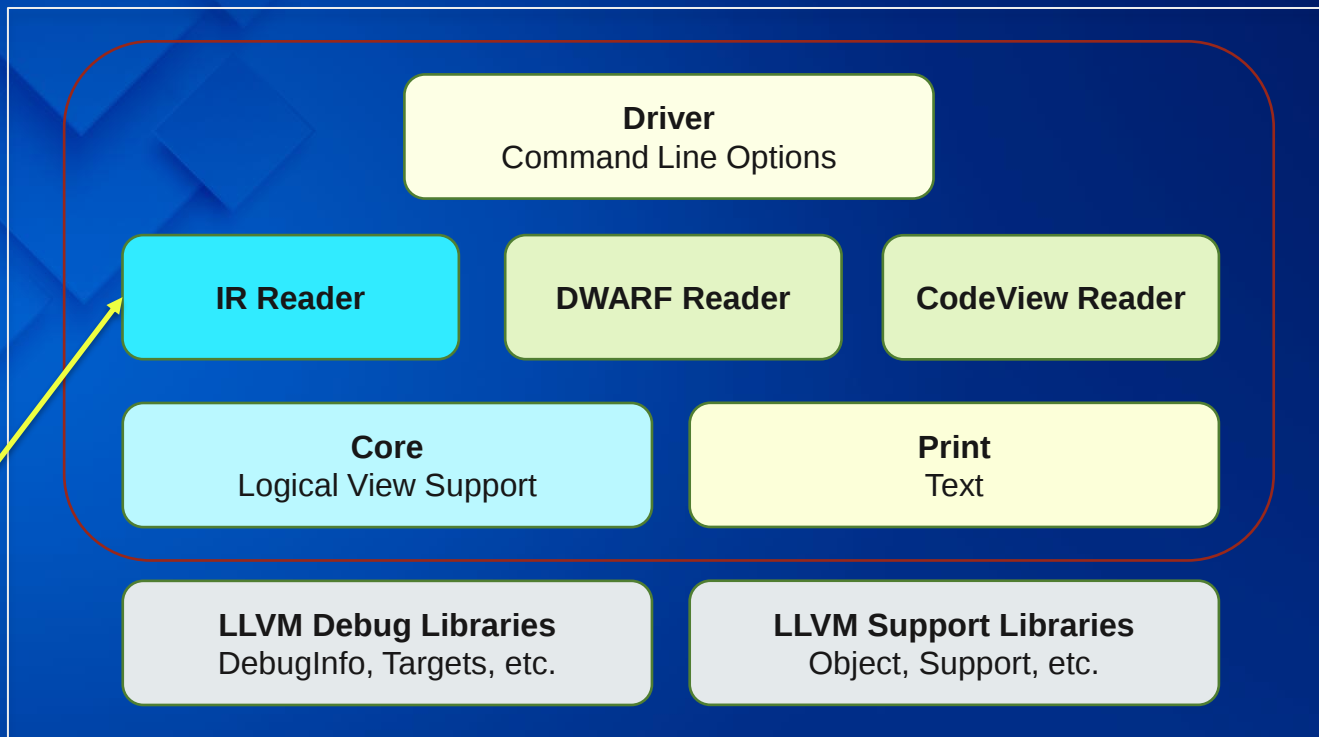
```
Types (.debug$T)
=====
0x1000 | LF_ARGLIST [size = 8]
0x1001 | LF_PROCEDURE [size = 16]
        return type = 0x0003 (void), # args = 0,
        param list = 0x1000
        calling conv = cdecl, options = None
0x1002 | LF_FUNC_ID [size = 16]
        name = foo, type = 0x1001, parent scope =
<no type>
0x1004 | LF_STRING_ID [size = 24] ID: <no type>,
String: hello-world.cpp

Symbols
=====
Mod 0000 | `'.debug$S`:
          0 | S_OBJNAME [size = 64] sig=0, `hello-
          world-clang-cv.o`
          0 | S_COMPILE3 [size = 156]
            machine = intel x86-x64, Ver = clang
            version 21.0.0, language = c++
            frontend = 21.0.0 flags = none
          0 | S_GPROC32_ID [size = 44] `foo`
            parent = 0, end = 0, addr = 0000:0000
            type = `0x1002 (foo)`, debug start = 0,
            debug end = 0, flags = noline | opt debuginfo
          0 | S_FRAMEPROC [size = 32]
            size = 40, padding size = 0 padding = 0
            bytes of callee saved registers = 0,
            exception handler addr = 0000:0000
            local fp reg = RSP, param fp reg = RSP
            flags = safe buffers
          0 | S_PROC_ID_END [size = 4]
          0 | S_BUILDINFO [size = 8] BuildId = `0x1008`
```

CodeView debug information



Current components



IR Reader component

Print logical views



Common options to print logical views when dealing with IR

- `--attribute=level`
- `--print=scopes,types,symbols,lines`

Common options to print logical views when dealing with IR

- `--attribute=level`
- `--print=scopes,types,symbols,lines`

IR tests

- After Simplify CFG pass: `simplify-cfg.ll`
- After SLP Vectorizer pass: `slp-vectorizer.ll`

Common options to print logical views when dealing with IR

- `--attribute=level`
- `--print=scopes,types,symbols,lines`

IR tests

- After Simplify CFG pass: `simplify-cfg.ll`
- After SLP Vectorizer pass: `slp-vectorizer.ll`

llvm-debuginfo-analyzer command line

- `--attribute=level --print=scopes,symbols,types simplify-cfg.ll`
- `--attribute=level --print=scopes,symbols,types slp-vectorizer.ll`

Logical views for Simplify CFG and SLP Vectorizer passes



Sony
Interactive
Entertainment



PlayStation.

```
llvm.dbg.cu = [0]
llvm.module.flags = [15, 16, 17, 18, 19, 110, 111]
llvm.ident = [112]
0 = distinct ID(CompileUnitLanguage: DW_LANG_C_plus_plus_14, file: 11, producer: "clang", isOptimized: false, runtimeVersion: 0, emissionKind: FullDebug, retainedTypes: 12, typeDebugInlining: false, nameTableKind: None)
11 = ID(FileReference: "test.cpp", directory: "")
12 = [13]
13 = ID(DerivedTypeTag: DW_TAG_typedef, name: "F32", file: 11, line: 7, baseType: 14)
14 = ID(BaseType: name: "float", size: 32, encoding: DW_ATE_float)
15 = [32 1, "Dwarf Version", 32 3]
16 = [32 2, "Debug Info Version", 32 3]
112 = [113 "clang"]
113 = distinct ID(Subprogram: name: "RandF32", linkageName: ".Z.RandF32", scope: 11, file: 11, line: 13, type: 114, scopeLine: 13, flags: DIFlagPrototyped, spFlags: DISPFlagDefinition, unit: 0, retainedNodes: 116)
114 = ID(SubroutineType: types: 115)
115 = [16]
116 = [1]
117 = ID(Location: line: 14, column: 15, scope: 113)
118 = ID(LocalVariable: name: "uRand", scope: 113, file: 11, line: 14, type: 119)
119 = ID(DerivedTypeTag: DW_TAG_typedef, name: "U32", file: 11, line: 6, baseType: 120)
120 = ID(BaseType: name: "unsigned int", size: 32, encoding: DW_ATE_unsigned)
121 = ID(Location: line: 0, scope: 113)
122 = ID(Location: line: 15, column: 21, scope: 113)
123 = ID(Location: line: 15, column: 27, scope: 113)
124 = ID(LocalVariable: name: "Rrand", scope: 113, file: 11, line: 15, type: 13)
125 = ID(Location: line: 16, column: 3, scope: 113)
126 = distinct ID(Subprogram: name: "randGauss", linkageName: ".Z.RandGaussPP", scope: 11, file: 11, line: 19, type: 127, scopeLine: 19, flags: DIFlagPrototyped, spFlags: DISPFlagDefinition, unit: 0, retainedNodes: 116)
127 = ID(SubroutineType: types: 128)
128 = [129, 129]
129 = ID(DerivedTypeTag: DW_TAG_pointer_type, baseType: 14, size: 64)
130 = ID(LocalVariable: name: "work", arg: 1, scope: 126, file: 11, line: 19, type: 129)
131 = ID(Location: line: 0, scope: 126)
132 = ID(Location: line: 22, column: 3, scope: 126)
133 = ID(Location: line: 23, column: 16, scope: 134)
134 = distinct ID(LexicalBlock: scope: 126, file: 11, line: 22, column: 6)
135 = ID(Location: line: 23, column: 26, scope: 134)
136 = ID(LocalVariable: name: "x1", scope: 126, file: 11, line: 20, type: 14)
137 = ID(Location: line: 24, column: 16, scope: 134)
138 = ID(Location: line: 24, column: 26, scope: 134)
139 = ID(LocalVariable: name: "x2", scope: 126, file: 11, line: 20, type: 14)
140 = ID(Location: line: 25, column: 22, scope: 134)
141 = ID(Location: line: 25, column: 17, scope: 134)
142 = ID(LocalVariable: name: "w", scope: 126, file: 11, line: 20, type: 14)
143 = ID(Location: line: 26, column: 14, scope: 126)
144 = ID(Location: line: 26, column: 3, scope: 134)
145 = distinct [145, 132, 146, 147]
146 = ID(Location: line: 26, column: 20, scope: 126)
147 = [148 "llvm.loop.mustprogress"]
148 = ID(Location: line: 28, column: 20, scope: 126)
149 = ID(Location: line: 28, column: 18, scope: 126)
150 = ID(Location: line: 28, column: 30, scope: 126)
151 = ID(Location: line: 28, column: 28, scope: 126)
152 = ID(Location: line: 28, column: 12, scope: 126)
153 = ID(Location: line: 28, column: 7, scope: 126)
154 = ID(Location: line: 29, column: 16, scope: 126)
155 = ID(Location: line: 29, column: 11, scope: 126)
156 = ID(Location: line: 30, column: 16, scope: 126)
157 = ID(Location: line: 30, column: 3, scope: 126)
158 = ID(Location: line: 30, column: 11, scope: 126)
159 = ID(Location: line: 31, column: 1, scope: 126)
```

IR metadata after Simplify CFG

Logical views for Simplify CFG and SLP Vectorizer passes



Sony
Interactive
Entertainment



PlayStation.

```
//vm.dbg.cu = 10
//vm.module.flags = {15, 16, 17, 18, 19, 110, 111}
//vm.ident = 112
8 = distinct ID(CompileUnit)(language: DW_LANG_C_plus_plus, file: 11, producer: "clang", isOptimized: false, runtimeVersion: 0,
emissionKind: FullDebug, retainedTypes: 12, optDebugInlining: false, nameTableKind: None)
11 = ID(FileName)(name: "test.cpp", directory: "")
12 = 113
13 = ID(DerivedType)(tag: DW_TAG_typedef, name: "F32", file: 11, line: 7, baseType: 14)
14 = ID(BasicType)(name: "float", size: 32, encoding: DW_ATE_float)
15 = 132 7, "Dwarf Version", 12 5
16 = 132 2, "Debug Info Version", 12 3
112 = 113("clang")
113 = distinct ID(Subprogram)(name: "RandF32", linkageName: ".Z7RandF32", scope: 11, file: 11, line: 13, type: 114, scopeLine: 13, flags:
DIFlagPrototyped, spFlags: DISPFlagDefinition, unit: 0, retainedNodes: 116)
114 = ID(SubroutineType)(types: 115)
115 = 114
116 = 11
117 = ID(Location)(line: 14, column: 15, scope: 113)
118 = ID(LocalVariable)(name: "uRand", scope: 113, file: 11, line: 14, type: 119)
119 = ID(DerivedType)(tag: DW_TAG_typedef, name: "U32", file: 11, line: 6, baseType: 120)
120 = ID(BasicType)(name: "unsigned int", size: 32, encoding: DW_ATE_unsigned)
121 = ID(Location)(line: 0, scope: 113)
122 = ID(Location)(line: 15, column: 21, scope: 113)
123 = ID(Location)(line: 15, column: 27, scope: 113)
124 = ID(LocalVariable)(name: "fRand", scope: 113, file: 11, line: 15, type: 121)
125 = ID(Location)(line: 16, column: 3, scope: 113)
126 = distinct ID(Subprogram)(name: "randGauss", linkageName: ".Z9randGaussPP", scope: 11, file: 11, line: 19, type: 127, scopeLine: 19,
flags: DIFlagPrototyped, spFlags: DISPFlagDefinition, unit: 0, retainedNodes: 116)
127 = ID(SubroutineType)(types: 128)
128 = 126
129 = ID(DerivedType)(tag: DW_TAG_pointer_type, baseType: 14, size: 64)
130 = ID(LocalVariable)(name: "work", arg: 1, scope: 126, file: 11, line: 19, type: 129)
131 = ID(Location)(line: 0, scope: 126)
132 = ID(Location)(line: 22, column: 3, scope: 126)
133 = ID(Location)(line: 23, column: 16, scope: 134)
134 = distinct ID(LexicalBlock)(scope: 126, file: 11, line: 22, column: 6)
135 = ID(Location)(line: 23, column: 26, scope: 134)
136 = ID(LocalVariable)(name: "x1", scope: 126, file: 11, line: 20, type: 14)
137 = ID(Location)(line: 24, column: 16, scope: 134)
138 = ID(Location)(line: 24, column: 26, scope: 134)
139 = ID(LocalVariable)(name: "x2", scope: 126, file: 11, line: 20, type: 14)
140 = ID(Location)(line: 25, column: 22, scope: 134)
141 = ID(Location)(line: 25, column: 17, scope: 134)
142 = ID(LocalVariable)(name: "w", scope: 126, file: 11, line: 20, type: 14)
143 = ID(Location)(line: 26, column: 14, scope: 126)
144 = ID(Location)(line: 26, column: 3, scope: 134)
145 = distinct 145, 132, 146, 147
146 = ID(Location)(line: 26, column: 20, scope: 126)
147 = 113("vm.loop.mustprogress")
148 = ID(Location)(line: 28, column: 20, scope: 126)
149 = ID(Location)(line: 28, column: 18, scope: 126)
150 = ID(Location)(line: 28, column: 30, scope: 126)
151 = ID(Location)(line: 28, column: 28, scope: 126)
152 = ID(Location)(line: 28, column: 12, scope: 126)
153 = ID(Location)(line: 28, column: 7, scope: 126)
154 = ID(Location)(line: 29, column: 16, scope: 126)
155 = ID(Location)(line: 29, column: 11, scope: 126)
156 = ID(Location)(line: 30, column: 16, scope: 126)
157 = ID(Location)(line: 30, column: 3, scope: 126)
158 = ID(Location)(line: 30, column: 11, scope: 126)
159 = ID(Location)(line: 31, column: 1, scope: 126)
```

IR metadata after Simplify CFG

Logical View:

```
[000] {File} 'simplify-cfg.ll'

[001] {CompileUnit} 'test.cpp'
[002] 6 {TypeAlias} 'U32' -> 'unsigned int'
[002] 7 {TypeAlias} 'F32' -> 'float'
[002] 13 {Function} 'RandF32' -> 'float'
[003] 14 {Variable} 'uRand' -> 'U32'
[003] 15 {Variable} 'fRand' -> 'F32'
[003] 14 {Line}
[003] 15 {Line}
[003] 15 {Line}
[003] 16 {Line}
[002] 19 {Function} 'randGauss' -> 'void'
[003] 19 {Parameter} 'work' -> '* float'
[003] 20 {Variable} 'w' -> 'float'
[003] 20 {Variable} 'x1' -> 'float'
[003] 20 {Variable} 'x2' -> 'float'
[003] 28 {Line}
[003] 29 {Line}
[003] 30 {Line}
[003] 30 {Line}
[003] 30 {Line}
[003] 31 {Line}
[003] 23 {Line}
[003] 24 {Line}
[003] 25 {Line}
```

Logical view after Simplify CFG

Logical views for Simplify CFG and SLP Vectorizer passes



Sony
Interactive
Entertainment



PlayStation.

```
%vm_dbg cu = %0
%vm_module.flags = %5,16,17,18,110,111
%vm_ident = %12
0 = distinct IDISubprogramUnit(language: DW_LANG_C_plus_plus,14, file:11, producer: "clang", isOptimized: false, runtimeVersion: 0,
  emscriptenKind: FullDebug, retainedTypes: 12, spsDebugInlining: false, name: %abiName, none)
11 = %0File(%Name: "test.cpp", directory: %
12 = %3
13 = %0DerivedTypeTag: DW_TAG_typeDef, name: "F32", file:11, line: 7, baseType: 14)
14 = %0BasicTypeName: "float", size: 32, encoding: DW_ATE_float)
15 = %32 7, "Default Vector", 02 5)
16 = %332 2, "Debug Info Version", 02 3)
112 = %0"clang"
113 = distinct IDISubprogram(name: "RandF32", linkageName: "ZRandF32", scope:11, file:11, line:13, type:114, scopeLine:13, flags:
  DIFlagPrototyped, spFlags: DISFlagDefinition, unit:10, retainedNodes:116)
114 = %0SubroutineType(types:115)
115 = %14
116 = %1
117 = %0Location(line:14, column:15, scope:113)
118 = %0LocalVariable(name: "uRand", scope:113, file:11, line:14, type:119)
119 = %0DerivedTypeTag: DW_TAG_typeDef, name: "U32", file:11, line:6, baseType:120)
120 = %0BasicTypeName: "unsigned int", size:32, encoding: DW_ATE_unsigned)
121 = %0Location(line:15, scope:113)
122 = %0Location(line:15, column:21, scope:113)
123 = %0Location(line:15, column:27, scope:113)
124 = %0LocalVariable(name: "fRand", scope:113, file:11, line:15, type:13)
125 = %0Location(line:16, column:3, scope:113)
126 = distinct IDISubprogram(name: "randGauss", linkageName: "ZrandGaussPP", scope:11, file:11, line:19, type:127, scopeLine:19,
  flags: DIFlagPrototyped, spFlags: DISFlagDefinition, unit:10, retainedNodes:116)
127 = %0SubroutineType(types:128)
128 = %14,129)
129 = %0DerivedTypeTag: DW_TAG_pointer_type, baseType:14, size:64)
130 = %0LocalVariable(name: "work", arg:1, scope:126, file:11, line:19, type:129)
131 = %0Location(line:19, scope:126)
132 = %0Location(line:22, column:3, scope:126)
133 = %0Location(line:23, column:16, scope:134)
134 = distinct IDISubprogramUnit(language: DW_LANG_C_plus_plus,14, file:11, line:22, column:6)
135 = %0Location(line:23, column:26, scope:134)
136 = %0LocalVariable(name: "x1", scope:126, file:11, line:20, type:14)
137 = %0Location(line:24, column:16, scope:134)
138 = %0Location(line:24, column:26, scope:134)
139 = %0LocalVariable(name: "x2", scope:126, file:11, line:20, type:14)
140 = %0Location(line:25, column:22, scope:134)
141 = %0Location(line:25, column:17, scope:134)
142 = %0LocalVariable(name: "w", scope:126, file:11, line:20, type:14)
143 = %0Location(line:26, column:14, scope:126)
144 = %0Location(line:26, column:3, scope:134)
145 = distinct %145,132,146,147)
146 = %0Location(line:26, column:20, scope:126)
147 = %0"arm.loop.mustprogress")
148 = %0Location(line:28, column:20, scope:126)
149 = %0Location(line:28, column:18, scope:126)
150 = %0Location(line:28, column:30, scope:126)
151 = %0Location(line:28, column:28, scope:126)
152 = %0Location(line:28, column:12, scope:126)
153 = %0Location(line:28, column:7, scope:126)
154 = %0Location(line:29, column:16, scope:126)
155 = %0Location(line:29, column:11, scope:126)
156 = %0Location(line:30, column:16, scope:126)
157 = %0Location(line:30, column:3, scope:126)
158 = %0Location(line:30, column:11, scope:126)
159 = %0Location(line:31, column:1, scope:126)
```

IR metadata after Simplify CFG

```
Logical View:
[000] {File} 'simplify-cfg.ll'

[001] {CompileUnit} 'test.cpp'
[002] 6 {TypeAlias} 'U32' -> 'unsigned int'
[002] 7 {TypeAlias} 'F32' -> 'float'
[002] 13 {Function} 'RandF32' -> 'float'
[003] 14 {Variable} 'uRand' -> 'U32'
[003] 15 {Variable} 'fRand' -> 'F32'
[003] 14 {Line}
[003] 15 {Line}
[003] 15 {Line}
[003] 16 {Line}
[002] 19 {Function} 'randGauss' -> 'void'
[003] 19 {Parameter} 'work' -> '* float'
[003] 20 {Variable} 'w' -> 'float'
[003] 20 {Variable} 'x1' -> 'float'
[003] 20 {Variable} 'x2' -> 'float'
[003] 28 {Line}
[003] 29 {Line}
[003] 30 {Line}
[003] 30 {Line}
[003] 30 {Line}
[003] 31 {Line}
[003] 23 {Line}
[003] 24 {Line}
[003] 25 {Line}
```

Logical view after Simplify CFG

```
Logical View:
[000] {File} 'slp-vectorizer.ll'

[001] {CompileUnit} 'test.cpp'
[002] 6 {TypeAlias} 'U32' -> 'unsigned int'
[002] 7 {TypeAlias} 'F32' -> 'float'
[002] 13 {Function} 'RandF32' -> 'float'
[003] 14 {Variable} 'uRand' -> 'U32'
[003] 15 {Variable} 'fRand' -> 'F32'
[003] 14 {Line}
[003] 15 {Line}
[003] 15 {Line}
[003] 16 {Line}
[002] 19 {Function} 'randGauss' -> 'void'
[003] 19 {Parameter} 'work' -> '* float'
[003] 20 {Variable} 'w' -> 'float'
[003] 28 {Line}
[003] 29 {Line}
[003] 29 {Line}
[003] 29 {Line}
[003] 31 {Line}
[003] 23 {Line}
[003] 23 {Line}
[003] 23 {Line}
[003] 25 {Line}
[003] 25 {Line}
```

Logical view after SLP Vectorizer

Compare logical views



Common options to compare logical views when dealing with IR

- `--report=list` `--report=view`
- `--print=scopes,types,symbols,lines`

Common options to compare logical views when dealing with IR

- `--report=list --report=view`
- `--print=scopes,types,symbols,lines`

IR tests

- After Simplify CFG pass: `simplify-cfg.ll`
- After SLP Vectorizer pass: `slp-vectorizer.ll`

Common options to compare logical views when dealing with IR

- `--report=list --report=view`
- `--print=scopes,types,symbols,lines`

IR tests

- After Simplify CFG pass: `simplify-cfg.ll`
- After SLP Vectorizer pass: `slp-vectorizer.ll`

llvm-debuginfo-analyzer command line

- `--compare=symbols,lines --report=list --print=symbols,lines simplify-cfg.ll slp-vectorizer.ll`
- `--compare=symbols,lines --report=view --print=symbols simplify-cfg.ll slp-vectorizer.ll`

Logical view changes - comparison tool



Sony
Interactive
Entertainment



PlayStation.

simplify-cfg.view

Logical View:
[000] (File) 'simplify_cfg.ll'
[001] (CompileUnit) 'test.cpp'
[002] 6 (TypeAlias) 'U32' -> 'unsigned int'
[002] 7 (TypeAlias) 'F32' -> 'float'
[002] 13 (Function) extern not_inlined
'RandF32' -> 'float'
[003] 14 (Variable) 'uRand' -> 'U32'
[003] 15 (Variable) 'fRand' -> 'F32'
[003] 14 (Line)
[003] 15 (Line)
[003] 15 (Line)
[003] 16 (Line)
[002] 19 (Function) extern not_inlined
'randGauss' -> 'void'
[003] 19 (Parameter) 'work' -> '* float'
[003] 20 (Variable) 'w' -> 'float'
[003] 20 (Variable) 'x1' -> 'float'
[003] 20 (Variable) 'x2' -> 'float'
[003] 22 (Line)
[003] 26 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 29 (Line)
[003] 29 (Line)
[003] 30 (Line)

slp-vectorizer.view

Logical View:
[000] (File) 'slp_vectorizer.ll'
[001] (CompileUnit) 'test.cpp'
[002] 6 (TypeAlias) 'U32' -> 'unsigned int'
[002] 7 (TypeAlias) 'F32' -> 'float'
[002] 13 (Function) extern not_inlined
'RandF32' -> 'float'
[003] 14 (Variable) 'uRand' -> 'U32'
[003] 15 (Variable) 'fRand' -> 'F32'
[003] 14 (Line)
[003] 15 (Line)
[003] 15 (Line)
[003] 16 (Line)
[002] 19 (Function) extern not_inlined
'randGauss' -> 'void'
[003] 19 (Parameter) 'work' -> '* float'
[003] 20 (Variable) 'w' -> 'float'
[003] 22 (Line)
[003] 26 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 29 (Line)
[003] 29 (Line)
[003] 29 (Line)

Ln: 18 Col: 1/49 Ch: 1/49 EOL: LF Windows-1252 Unix
* [003] 20 (Variable) 'x1' -> 'float'
[003] 20 (Variable) 'x2' -> 'float'

IR changes: comparison tool

Logical view changes - built-in compare (report mode)



simplify-cfg.view

Logical View:
[000] (File) 'simplify-cfg.ll'
[001] (CompileUnit) 'test.cpp'
[002] 6 (TypeAlias) 'U32' -> 'unsigned int'
[002] 7 (TypeAlias) 'F32' -> 'float'
[002] 13 (Function) extern not_inlined
'RandF32' -> 'float'
[003] 14 (Variable) 'uRand' -> 'U32'
[003] 15 (Variable) 'fRand' -> 'F32'
[003] 14 (Line)
[003] 15 (Line)
[003] 15 (Line)
[003] 16 (Line)
[002] 19 (Function) extern not_inlined
'randGauss' -> 'void'
[003] 19 (Parameter) 'work' -> '* float'
[003] 20 (Variable) 'w' -> 'float'
[003] 20 (Variable) 'x1' -> 'float'
[003] 20 (Variable) 'x2' -> 'float'
[003] 22 (Line)
[003] 26 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 29 (Line)
[003] 29 (Line)
[003] 30 (Line)

slp-vectorizer.view

Logical View:
[000] (File) 'slp-vectorizer.ll'
[001] (CompileUnit) 'test.cpp'
[002] 6 (TypeAlias) 'U32' -> 'unsigned int'
[002] 7 (TypeAlias) 'F32' -> 'float'
[002] 13 (Function) extern not_inlined
'RandF32' -> 'float'
[003] 14 (Variable) 'uRand' -> 'U32'
[003] 15 (Variable) 'fRand' -> 'F32'
[003] 14 (Line)
[003] 15 (Line)
[003] 15 (Line)
[003] 16 (Line)
[002] 19 (Function) extern not_inlined
'randGauss' -> 'void'
[003] 19 (Parameter) 'work' -> '* float'
[003] 20 (Variable) 'w' -> 'float'
[003] 22 (Line)
[003] 26 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 29 (Line)
[003] 29 (Line)
[003] 29 (Line)

Ln: 18 Col: 1/49 Ch: 1/49 EOL: LF
* [003] 20 (Variable) 'x1' -> 'float'
[003] 20 (Variable) 'x2' -> 'float'

Reference: 'simplify-cfg.ll'
Target: 'slp-vectorizer.ll'

IR changes: comparison tool

IR changes: llvm-debuginfo-analyzer

Logical view changes - built-in compare (report mode)



simplify-cfg.view

Logical View:
[000] (File) 'simplify-cfg.ll'
[001] (CompileUnit) 'test.cpp'
[002] 6 {TypeAlias} 'U32' -> 'unsigned int'
[002] 7 {TypeAlias} 'F32' -> 'float'
[002] 13 (Function) extern not_inlined
'RandF32' -> 'float'
[003] 14 (Variable) 'uRand' -> 'U32'
[003] 15 (Variable) 'fRand' -> 'F32'
[003] 14 (Line)
[003] 15 (Line)
[003] 15 (Line)
[003] 16 (Line)
[002] 19 (Function) extern not_inlined
'randGauss' -> 'void'
[003] 19 (Parameter) 'work' -> '* float'
[003] 20 (Variable) 'w' -> 'float'
[003] 20 (Variable) 'x1' -> 'float'
[003] 20 (Variable) 'x2' -> 'float'
[003] 22 (Line)
[003] 26 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 29 (Line)
[003] 29 (Line)
[003] 30 (Line)

slp-vectorizer.view

Logical View:
[000] (File) 'slp-vectorizer.ll'
[001] (CompileUnit) 'test.cpp'
[002] 6 {TypeAlias} 'U32' -> 'unsigned int'
[002] 7 {TypeAlias} 'F32' -> 'float'
[002] 13 (Function) extern not_inlined
'RandF32' -> 'float'
[003] 14 (Variable) 'uRand' -> 'U32'
[003] 15 (Variable) 'fRand' -> 'F32'
[003] 14 (Line)
[003] 15 (Line)
[003] 15 (Line)
[003] 16 (Line)
[002] 19 (Function) extern not_inlined
'randGauss' -> 'void'
[003] 19 (Parameter) 'work' -> '* float'
[003] 20 (Variable) 'w' -> 'float'
[003] 22 (Line)
[003] 26 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 29 (Line)
[003] 29 (Line)
[003] 29 (Line)

Ln: 18 Col: 1/49 Ch: 1/49 EOL: LF
* [003] 20 (Variable) 'x1' -> 'float'
[003] 20 (Variable) 'x2' -> 'float'

Reference: 'simplify-cfg.ll'
Target: 'slp-vectorizer.ll'

(2) Missing Symbols:

- 20 {Variable} 'x1' -> 'float'
- 20 {Variable} 'x2' -> 'float'

IR changes: comparison tool

IR changes: llvm-debuginfo-analyzer

Logical view changes - built-in compare (report mode)



simplify-cfg.view

Logical View:
[000] (File) 'simplify-cfg.ll'
[001] (CompileUnit) 'test.cpp'
[002] 6 {TypeAlias} 'U32' -> 'unsigned int'
[002] 7 {TypeAlias} 'F32' -> 'float'
[002] 13 (Function) extern not_inlined
'RandF32' -> 'float'
[003] 14 (Variable) 'uRand' -> 'U32'
[003] 15 (Variable) 'fRand' -> 'F32'
[003] 14 (Line)
[003] 15 (Line)
[003] 15 (Line)
[003] 16 (Line)
[002] 19 (Function) extern not_inlined
'randGauss' -> 'void'
[003] 19 (Parameter) 'work' -> '* float'
[003] 20 (Variable) 'w' -> 'float'
[003] 20 (Variable) 'x1' -> 'float'
[003] 20 (Variable) 'x2' -> 'float'
[003] 22 (Line)
[003] 26 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 29 (Line)
[003] 29 (Line)
[003] 30 (Line)

slp-vectorizer.view

Logical View:
[000] (File) 'slp-vectorizer.ll'
[001] (CompileUnit) 'test.cpp'
[002] 6 {TypeAlias} 'U32' -> 'unsigned int'
[002] 7 {TypeAlias} 'F32' -> 'float'
[002] 13 (Function) extern not_inlined
'RandF32' -> 'float'
[003] 14 (Variable) 'uRand' -> 'U32'
[003] 15 (Variable) 'fRand' -> 'F32'
[003] 14 (Line)
[003] 15 (Line)
[003] 15 (Line)
[003] 16 (Line)
[002] 19 (Function) extern not_inlined
'randGauss' -> 'void'
[003] 19 (Parameter) 'work' -> '* float'
[003] 20 (Variable) 'w' -> 'float'
[003] 22 (Line)
[003] 26 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 29 (Line)
[003] 29 (Line)
[003] 29 (Line)

Ln: 18 Col: 1/49 Ch: 1/49 EOL: LF Windows-1252 Unix Line: 17-18 Windows-1252 Unix
* [003] 20 (Variable) 'x1' -> 'float'
[003] 20 (Variable) 'x2' -> 'float'

IR changes: comparison tool

Reference: 'simplify-cfg.ll'
Target: 'slp-vectorizer.ll'

(2) Missing Symbols:

- 20 {Variable} 'x1' -> 'float'
- 20 {Variable} 'x2' -> 'float'

(5) Missing Lines:

- 24 {Line}
- 30 {Line}
- 30 {Line}
- 30 {Line}
- 24 {Line}

IR changes: llvm-debuginfo-analyzer

Logical view changes - built-in compare (view mode)



Sony
Interactive
Entertainment



PlayStation.

simplify-cfg.view

Logical View:
[000] (File) 'simplify-cfg.ll'
[001] (CompileUnit) 'test.cpp'
[002] 6 {TypeAlias} 'U32' -> 'unsigned int'
[002] 7 {TypeAlias} 'F32' -> 'float'
[002] 13 (Function) extern not_inlined
'RandF32' -> 'float'
[003] 14 (Variable) 'uRand' -> 'U32'
[003] 15 (Variable) 'fRand' -> 'F32'
[003] 14 (Line)
[003] 15 (Line)
[003] 15 (Line)
[003] 16 (Line)
[002] 19 (Function) extern not_inlined
'randGauss' -> 'void'
[003] 19 {Parameter} 'work' -> '* float'
[003] 20 {Variable} 'w' -> 'float'
[003] 20 {Variable} 'x1' -> 'float'
[003] 20 {Variable} 'x2' -> 'float'
[003] 22 (Line)
[003] 26 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 29 (Line)
[003] 29 (Line)
[003] 30 (Line)

slp-vectorizer.view

Logical View:
[000] (File) 'slp-vectorizer.ll'
[001] (CompileUnit) 'test.cpp'
[002] 6 {TypeAlias} 'U32' -> 'unsigned int'
[002] 7 {TypeAlias} 'F32' -> 'float'
[002] 13 (Function) extern not_inlined
'RandF32' -> 'float'
[003] 14 (Variable) 'uRand' -> 'U32'
[003] 15 (Variable) 'fRand' -> 'F32'
[003] 14 (Line)
[003] 15 (Line)
[003] 15 (Line)
[003] 16 (Line)
[002] 19 (Function) extern not_inlined
'randGauss' -> 'void'
[003] 19 {Parameter} 'work' -> '* float'
[003] 20 {Variable} 'w' -> 'float'
[003] 22 (Line)
[003] 26 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 29 (Line)
[003] 29 (Line)
[003] 29 (Line)

Ln: 18 Col: 1/49 Ch: 1/49 EOL: LF Windows-1252 Unix
* [003] 20 {Variable} 'x1' -> 'float'
[003] 20 {Variable} 'x2' -> 'float'

Reference: 'simplify-cfg.ll'
Target: 'slp-vectorizer.ll'

(2) Missing Symbols:

- 20 {Variable} 'x1' -> 'float'
- 20 {Variable} 'x2' -> 'float'

(5) Missing Lines:

- 24 {Line}
- 30 {Line}
- 30 {Line}
- 30 {Line}
- 24 {Line}

Reference: 'simplify-cfg.ll'
Target: 'slp-vectorizer.ll'

IR changes: comparison tool

IR changes: llvm-debuginfo-analyzer

Logical view changes - built-in compare (view mode)



The screenshot shows the LLVM-IR built-in compare tool interface. It has two panes: 'simplify-cfg.view' on the left and 'slp-vectorizer.view' on the right. Both panes show a 'Logical View' of the IR. The left pane shows the IR for 'simplify-cfg.ll' with lines 0001 to 0030. The right pane shows the IR for 'slp-vectorizer.ll' with lines 0001 to 0030. The tool is comparing the two IRs and highlighting differences. In the left pane, lines 0020 and 0021 are highlighted in red, indicating a difference. In the right pane, lines 0020 and 0021 are highlighted in yellow, indicating a difference. The status bar at the bottom shows 'Ln: 18 Col: 1/49 Ch: 1/49 EOL: LF' and 'Windows-1252 Unix'.

```
Logical View:
[000] (File) 'simplify-cfg.ll'

[001] (CompileUnit) 'test.cpp'
[002] 6 (TypeAlias) 'U32' -> 'unsigned int'
[002] 7 (TypeAlias) 'F32' -> 'float'
[002] 13 (Function) extern not_inlined
'RandF32' -> 'float'
[003] 14 (Variable) 'uRand' -> 'U32'
[003] 15 (Variable) 'fRand' -> 'F32'
[003] 14 (Line)
[003] 15 (Line)
[003] 15 (Line)
[003] 16 (Line)
[002] 19 (Function) extern not_inlined
'randGauss' -> 'void'
[003] 19 (Parameter) 'work' -> '* float'
[003] 20 (Variable) 'w' -> 'float'
[003] 20 (Variable) 'x1' -> 'float'
[003] 20 (Variable) 'x2' -> 'float'
[003] 22 (Line)
[003] 26 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 29 (Line)
[003] 29 (Line)
[003] 30 (Line)

Logical View:
[000] (File) 'slp-vectorizer.ll'

[001] (CompileUnit) 'test.cpp'
[002] 6 (TypeAlias) 'U32' -> 'unsigned int'
[002] 7 (TypeAlias) 'F32' -> 'float'
[002] 13 (Function) extern not_inlined
'RandF32' -> 'float'
[003] 14 (Variable) 'uRand' -> 'U32'
[003] 15 (Variable) 'fRand' -> 'F32'
[003] 14 (Line)
[003] 15 (Line)
[003] 15 (Line)
[003] 16 (Line)
[002] 19 (Function) extern not_inlined
'randGauss' -> 'void'
[003] 19 (Parameter) 'work' -> '* float'
[003] 20 (Variable) 'w' -> 'float'
[003] 22 (Line)
[003] 26 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 29 (Line)
[003] 29 (Line)
[003] 29 (Line)
```

IR changes: comparison tool

Reference: 'simplify-cfg.ll'
Target: 'slp-vectorizer.ll'

(2) Missing Symbols:

- 20 {Variable} 'x1' -> 'float'
- 20 {Variable} 'x2' -> 'float'

(5) Missing Lines:

- 24 {Line}
- 30 {Line}
- 30 {Line}
- 30 {Line}
- 24 {Line}

Reference: 'simplify-cfg.ll'
Target: 'slp-vectorizer.ll'

Logical View:

{File} 'simplify-cfg.ll'

{CompileUnit} 'test.cpp'

{Function} extern not_inlined 'RandF32' -> 'float'

{Variable} 'uRand' -> 'U32'

{Variable} 'fRand' -> 'F32'

{Function} extern not_inlined 'randGauss' -> 'void'

{Parameter} 'work' -> '* float'

{Variable} 'w' -> 'float'

IR changes: llvdbginfo-analyzer

Logical view changes - built-in compare (view mode)



simplify-cfg.view

Logical View:
[000] (File) 'simplify-cfg.ll'
[001] (CompileUnit) 'test.cpp'
[002] 6 {TypeAlias} 'U32' -> 'unsigned int'
[002] 7 {TypeAlias} 'F32' -> 'float'
[002] 13 {Function} extern not_inlined
'RandF32' -> 'float'
[003] 14 (Variable) 'uRand' -> 'U32'
[003] 15 (Variable) 'fRand' -> 'F32'
[003] 14 (Line)
[003] 15 (Line)
[003] 15 (Line)
[003] 15 (Line)
[003] 16 (Line)
[002] 19 {Function} extern not_inlined
'randGauss' -> 'void'
[003] 19 {Parameter} 'work' -> '* float'
[003] 20 {Variable} 'w' -> 'float'
[003] 20 {Variable} 'x1' -> 'float'
[003] 20 {Variable} 'x2' -> 'float'
[003] 22 (Line)
[003] 26 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 29 (Line)
[003] 29 (Line)
[003] 30 (Line)

slp-vectorizer.view

Logical View:
[000] (File) 'slp-vectorizer.ll'
[001] (CompileUnit) 'test.cpp'
[002] 6 {TypeAlias} 'U32' -> 'unsigned int'
[002] 7 {TypeAlias} 'F32' -> 'float'
[002] 13 {Function} extern not_inlined
'RandF32' -> 'float'
[003] 14 (Variable) 'uRand' -> 'U32'
[003] 15 (Variable) 'fRand' -> 'F32'
[003] 14 (Line)
[003] 15 (Line)
[003] 15 (Line)
[003] 16 (Line)
[002] 19 {Function} extern not_inlined
'randGauss' -> 'void'
[003] 19 {Parameter} 'work' -> '* float'
[003] 20 {Variable} 'w' -> 'float'
[003] 22 (Line)
[003] 26 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 28 (Line)
[003] 29 (Line)
[003] 29 (Line)
[003] 29 (Line)

Ln: 18 Col: 1/49 Ch: 1/49 EOL: LF Windows-1252 Unix Line: 17-18
* [003] 20 {Variable} 'x1' -> 'float'
[003] 20 {Variable} 'x2' -> 'float'

IR changes: comparison tool

Reference: 'simplify-cfg.ll'
Target: 'slp-vectorizer.ll'

(2) Missing Symbols:

- 20 {Variable} 'x1' -> 'float'
- 20 {Variable} 'x2' -> 'float'

(5) Missing Lines:

- 24 {Line}
- 30 {Line}
- 30 {Line}
- 30 {Line}
- 24 {Line}

Reference: 'simplify-cfg.ll'
Target: 'slp-vectorizer.ll'

Logical View:

{File} 'simplify-cfg.ll'

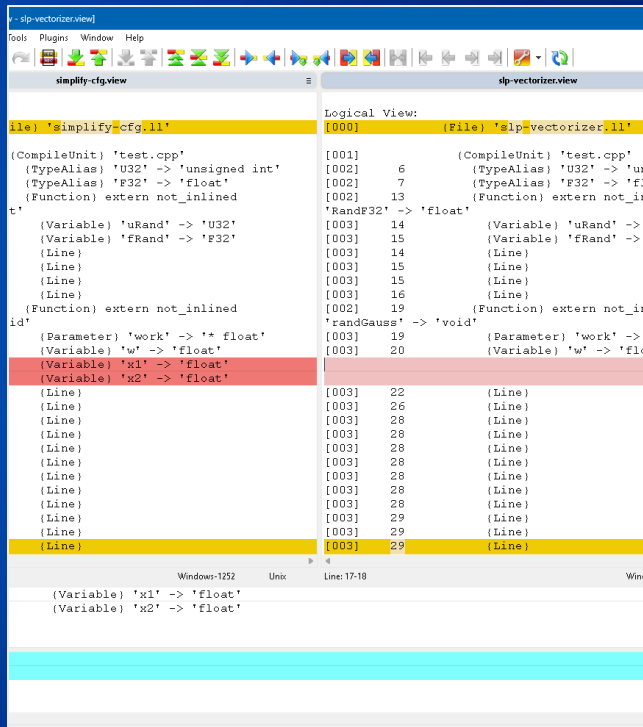
{CompileUnit} 'test.cpp'
{Function} extern not_inlined 'RandF32' -> 'float'
{Variable} 'uRand' -> 'U32'
{Variable} 'fRand' -> 'F32'
{Function} extern not_inlined 'randGauss' -> 'void'
{Parameter} 'work' -> '* float'
{Variable} 'w' -> 'float'
- 20 {Variable} 'x1' -> 'float'
- 20 {Variable} 'x2' -> 'float'

IR changes: llvm-debuginfo-analyzer

Conclusion



PlayStation®



```
Logical View:
{File} 'simplify-cfg.ll'

    {CompileUnit} 'test.cpp'
13     {Function} 'RandF32' -> 'float'
14     {Variable} 'uRand' -> 'U32'
15     {Variable} 'fRand' -> 'F32'
19     {Function} 'randGauss' -> 'void'
19     {Parameter} 'work' -> '* float'
20     {Variable} 'w' -> 'float'
- 20     {Variable} 'x1' -> 'float'
- 20     {Variable} 'x2' -> 'float'
```

IR changes: llvm-debuginfo-analyzer



Thank you!