Optimization Remarks

Update

Francis Visoiu Mistrih
Optimization Remarks

```
18 int main(int argc, const char *argv[]) {
20  4120 stack bytes in function
21 25 instructions in function
22  int result = foo(argc);
23  
24  ``foo`` inlined into main with (cost=65, threshold=225)
25  return 0;
26  }
```

opt-viewer.py
Challenges

- YAML doesn’t scale well
- Discovering remarks
- Processing remarks
• Significant size impact
- Significant size impact

- Slow to parse: \textsc{\textasciitilde 4 minutes} for 27 million remarks
• Significant size impact

• Slow to parse: ~4 minutes for 27 million remarks

• Compile-time impact: 5% slowdown
- Significant size impact

- Slow to parse: \(~4\) minutes for 27 million remarks

- Compile-time impact: 5% slowdown
- Significant size impact 12x smaller
- Slow to parse: ~4 minutes for 27 million remarks
- Compile-time impact: 5% slowdown
- Significant size impact 12x smaller
- Slow to parse: ~4 minutes 27 seconds (8x faster)
- Compile-time impact: 5% slowdown
- Significant size impact: **12x smaller**
- Slow to parse: ~4 minutes 27 seconds (8x faster)
- Compile-time impact: 5% slowdown, 2% slowdown
Extra Benefits

• Well-known format in the community

• Testing and tooling like llvm-bcanalyzer

• Versioning

• Flexible metadata
#include "llvm-c/Remarks.h"

LLVMRemarkParserRef Parser = LLVMRemarkParserCreateYAML(Buf, Size);
LLVMRemarkEntryRef Remark = NULL;
while ((Remark = LLVMRemarkParserGetNext(Parser))) {
    // use Remark
}
#include "llvm-c/Remarks.h"

LLVMRemarkParserRef Parser = LLVMRemarkParserCreateBitstream(Buf, Size);
LLVMRemarkEntryRef Remark = NULL;
while ((Remark = LLVMRemarkParserGetNext(Parser))) {
  // use Remark
}
Remarks Section

- Only in object files
- Not included in the linked binary
- Accessible through the Debug Map
- clang’s remarks section: 0.27GB
Get Involved

- Python bindings for integration with opt-viewer.py
- llvm-remarkutil: convert, merge, extract, etc.
- Faster lookup: build an index
- Remark classes: verbosity, accuracy, etc.
- Tablegen: better remark structure
- Per-remark documentation: possible actions, reasons, etc.
Thanks!