Google

ClangBuiltLinux

What's Next?

Nick Desaulniers

LINUXCONFAU
ONLINE 2021

2020 Recap

2020 Recap

- CI 

  - (0day bot, Linaro TCWG, syzkaller)
- Google Prod Kernel (Android & CrOS earlier)
- ASM goto with outputs
- LTO, PGO, AutoFDO, SCS
- linux-toolchains mailing list
- Integrated assembler work

2021 and beyond

```
Samples: 13M of event 'cycles:pp', Event count (approx.): 10679682356397
  Overhead  Shared Object             Symbol
+    2.24%  clang-12                 [.] clang::SourceManager::getFileIDLocal
+    1.79%  clang-12                 [.] llvm::StringMapImpl::LookupBucketFor
+    1.33%  clang-12                 [.] clang::Lexer::LexTokenInternal
+    1.30%  clang-12                 [.] clang::TokenLexer::Lex
+    0.98%  clang-12                 [.] (anonymous namespace)::IntExprEvaluator::VisitBinaryOperator
+    0.92%  clang-12                 [.] llvm::PMDataManager::findAnalysisPass
+    0.91%  clang-12                 [.] clang::Preprocessor::Lex
+    0.86%  clang-12                 [.] GetFullTypeForDeclarator
+    0.85%  libc-2.31.so             [.] _int_malloc
+    0.77%  [kernel.vmlinux]         [k] clear_page_erms
+    0.74%  clang-12                 [.] CheckICE
+    0.71%  clang-12                 [.] GetDiagInfo
+    0.68%  clang-12                 [.] clang::ASTContext::getDeclAttrs
+    0.65%  clang-12                 [.] clang::SourceManager::isOffsetInFileID
     0.64%  libc-2.31.so             [.] __memmove_avx_unaligned_erms
+    0.63%  clang-12                 [.] clang::Lexer::LexIdentifier
     0.61%  libc-2.31.so             [.] __memcmp_avx2_movbe
     0.59%  libc-2.31.so             [.] malloc
+    0.58%  clang-12                 [.] clang::TokenLexer::ExpandFunctionArguments
     0.44%  clang-12                 [.] llvm::BumpPtrAllocatorImpl<llvm::MallocAllocator, 4096ul, 4096ul, 128ul>::Allocate
     0.43%  clang-12                 [.] clang::Preprocessor::getMacroDefinition
     0.42%  libc-2.31.so             [.] _int_free
     0.42%  clang-12                 [.] clang::ASTContext::getTypeInfo
     0.40%  clang-12                 [.] llvm::FoldingSetBase::FindNodeOrInsertPos
     0.39%  clang-12                 [.] clang::Parser::ParseDeclarationSpecifiers
     0.39%  clang-12                 [.] (anonymous namespace)::CFGBuilder::Visit
     0.38%  clang-12                 [.] clang::IgnoreParensSingleStep
     0.38%  clang-12                 [.] clang::Sema::LookupName
     0.38%  clang-12                 [.] clang::ASTContext::getIntWidth
     0.36%  clang-12                 [.] clang::Parser::ParseCastExpression
     0.35%  [kernel.vmlinux]         [k] filemap_map_pages
     0.35%  clang-12                 [.] AnalyzeImplicitConversions
     0.35%  clang-12                 [.] clang::Preprocessor::ReadMacroCallArgumentList
     0.34%  libc-2.31.so             [.] cfree@GLIBC_2.2.5
     0.34%  libc-2.31.so             [.] __strlen_avx2
     0.34%  [kernel.vmlinux]         [k] unmap_page_range
```

Improving compile times

-flazy-parse -> CFG walk based LLVM IR Gen? -> C IR?

(Chris Lattner alluded to this at CGO 2020)

Macro expansion is killing compile times for both toolchains.

Moar architectures

2017-2020:

- arm
- arm64
- x86
- powerpc
- mips
- riscv

2021?:

- arm
- arm64
- x86
- powerpc
- mips
- riscv
- s390
- Hexagon
- M68k
- CSKY

Post link/re-link optimization

Propeller does whole program basic block layout at link time via basic block sections. We have added support for having each basic block in its own section which allows the linker to do arbitrary reorderings of basic blocks to achieve any desired fine-grain code layout which includes block layout, function splitting and function reordering.

Rust in kernel

I'm not going to _defend_ Rust, I just feel it's potentially interesting if somebody puts in the effort and can show that it has real advantages.

I think that's a big "if", but I think it's also unusually interesting if it actually works, so in that sense I guess I'm rooting for it.

– Linus

- <rust-for-linux@vger.kernel.org>
- https://rust-for-linux.github.io/
- Checkout talk from Linux Plumbers 2020 LLVM MC: https://youtu.be/FFjV9f_Ub9o?t=2060

Clang-tidy driven treewide refactoring

Cool idea by Tom Rix <trix@redhat.com>, has some LLVM patches posted.

Supporting additional distros

- Add distro configs to CI (~10^6000 possible kernel configs)
- Packaging of LLVM
- Documentation
- OpenMandriva 4.2 on track to release with Clang built kernels

And more

- DFSAN
- Thread Locking Annotations
- Further compiler optimizations for kernel
- Streaming kernel development on Twitch
- WG14 Linux kernel representation

Thanks!

For more info:
clangbuiltlinux.github.io