Measuring Kernel Compile Times w/ Clang

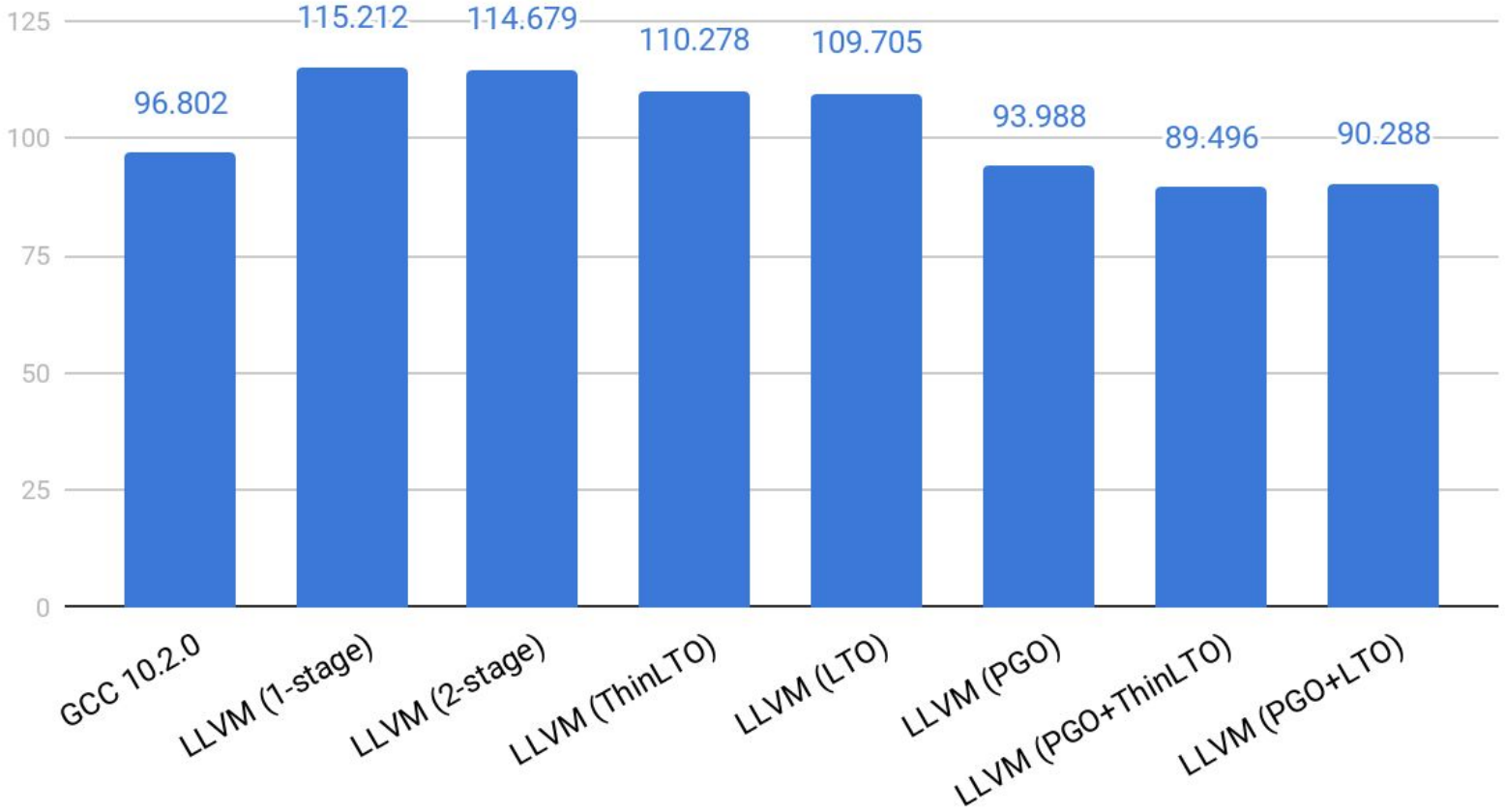
Linux Plumbers Conf 2020 - LLVM MC

Nathan Chancellor, Nathan Huckleberry

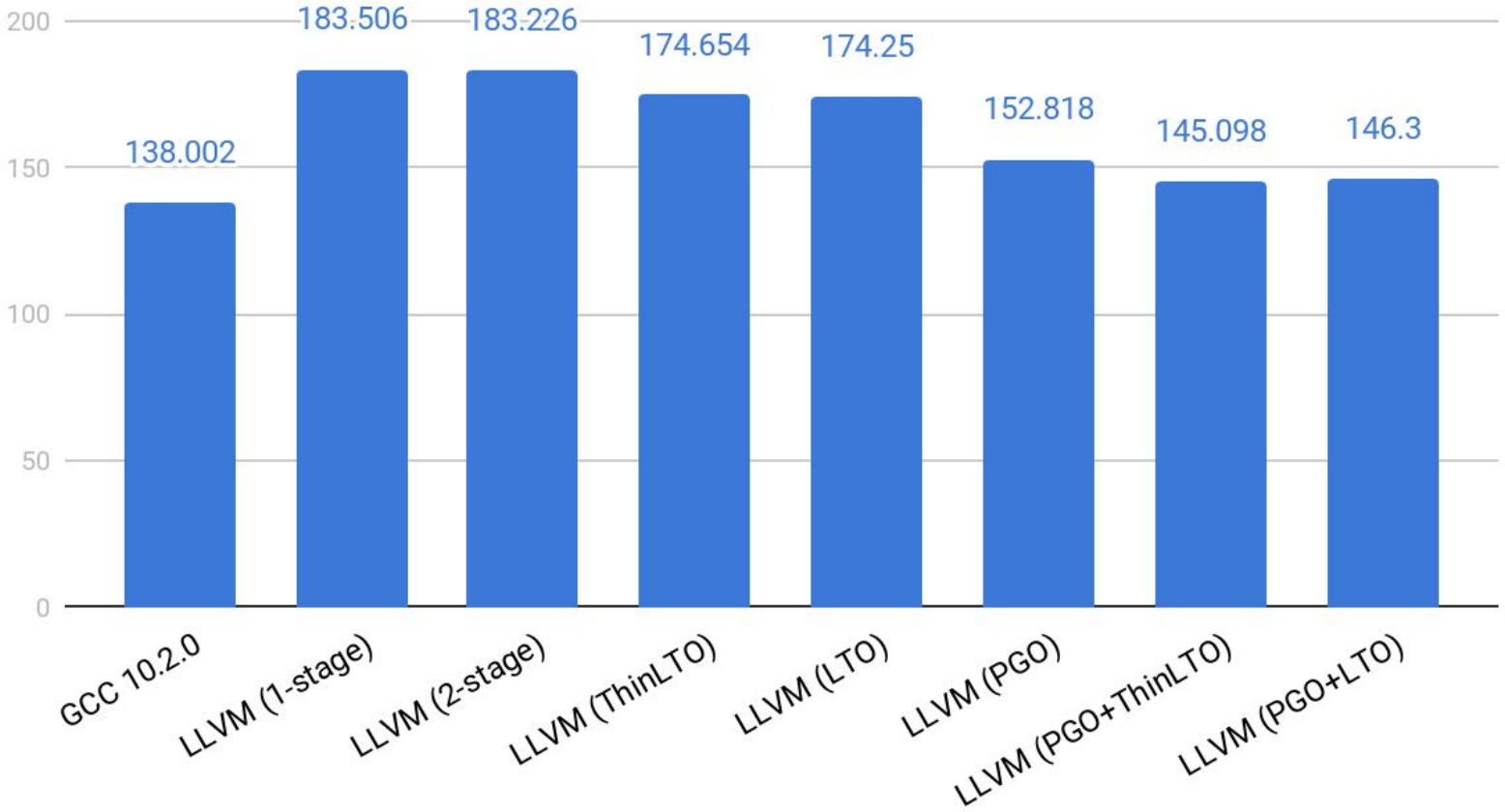
Benchmarking GCC vs. LLVM

- Tests were conducted using hyperfine to build kernels (5.8.1) in a loop with both GCC 10.2.0 and several versions of LLVM at a450654a52874 (11.0.0 prerelease).
- TL;DR: GCC always beats LLVM for arm64 and x86_64, even when LLVM is compiled with LTO and PGO. LLVM needs to be compiled with PGO to build 32-bit ARM faster than GCC.
- Full results available on GitHub along with the testing scripts.

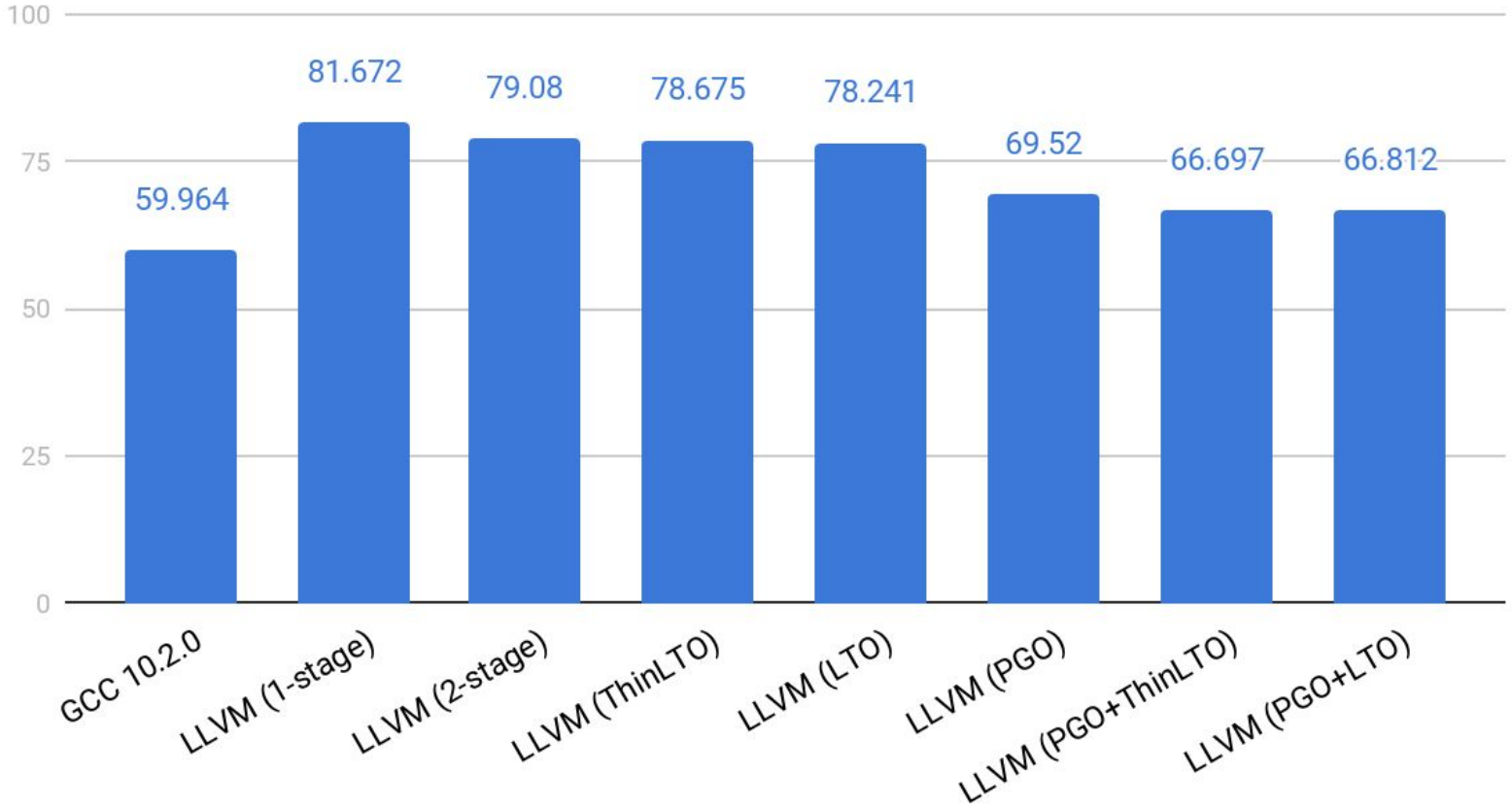
arm multi_v7_defconfig build times



arm64 defconfig build times

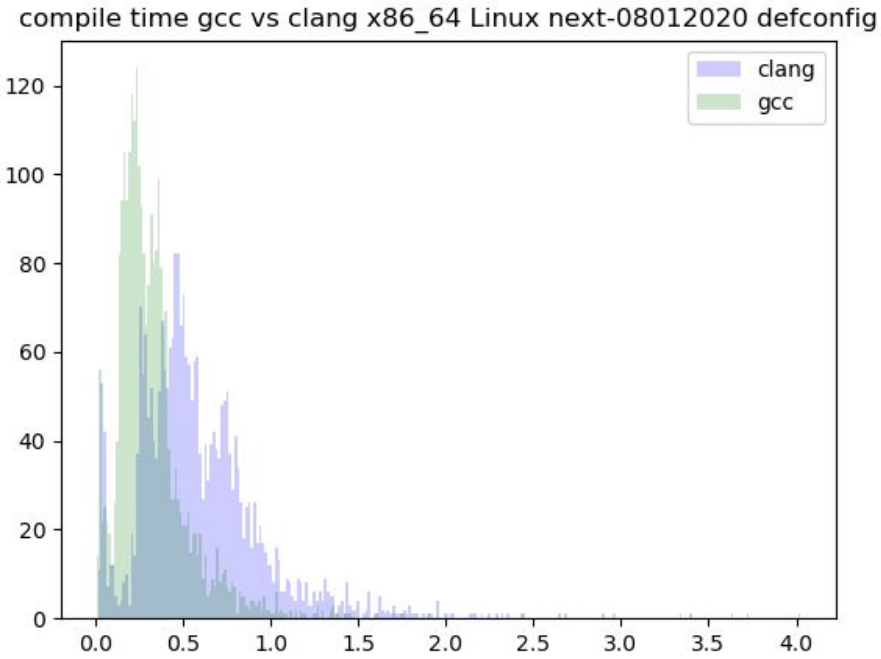


x86_64 defconfig build times



Basic measurements

$ make CC="time <compiler>" &> log_for_post_processing.txt



Profiling

Linux Perf - Sampling based profiling

- Useful for finding functions that are generally slow
- Many of the top functions are known to be slow and are hard to optimize

Perfetto - Event based profiling

- Records time spent in frontend vs backend
- Useful for finding compilation outliers (weird frontend/backend ratio)
- Fixing uncommonly occurring slowdowns gives smaller speedups

Tracing whole builds with perf

```
$ echo 0 | sudo tee /proc/sys/kernel/kptr_restrict \
  /proc/sys/kernel/perf_event_paranoid
$ perf record -e cycles:pp --call-graph lbr make LLVM=1 -j71
$ perf report --no-children --sort=dso,symbol
```

```
Samples: 10M of event 'cycles:ppu', Event count (approx.): 7858027097781
  Overhead  Shared Object              Symbol
+    3.01%  clang-12                  [.] clang::SourceManager::getFileIDLocal
+    2.32%  clang-12                  [.] llvm::StringMapImpl::LookupBucketFor
+    1.47%  clang-12                  [.] clang::Lexer::LexTokenInternal
+    1.47%  clang-12                  [.] clang::TokenLexer::Lex
+    1.11%  clang-12                  [.] GetFullTypeForDeclarator
+    1.10%  clang-12                  [.] clang::Preprocessor::Lex
+    0.93%  libc-2.30.so              [.] _int_malloc
+    0.86%  clang-12                  [.] clang::Lexer::LexIdentifier
+    0.85%  clang-12                  [.] (anonymous namespace)::IntExprEvaluator::VisitBinaryOperator
+    0.81%  clang-12                  [.] CheckICE
+    0.81%  clang-12                  [.] clang::ASTContext::getDeclAttrs
+    0.75%  libc-2.30.so              [.] malloc
+    0.74%  libc-2.30.so              [.] __memcmp_avx2_movbe
+    0.69%  clang-12                  [.] clang::TokenLexer::ExpandFunctionArguments
+    0.69%  libc-2.30.so              [.] __memmove_avx_unaligned_erms
+    0.68%  clang-12                  [.] (anonymous namespace)::CFGBuilder::Visit
     0.64%  [unknown]                 [k] 0xffffffff8d400ae7
+    0.60%  clang-12                  [.] clang::DiagnosticIDs::getDiagnosticSeverity
+    0.59%  clang-12                  [.] clang::APValue::swap
     0.55%  libc-2.30.so              [.] _int_free
+    0.53%  clang-12                  [.] GetDiagInfo
     0.52%  clang-12                  [.] llvm::FoldingSetBase::FindNodeOrInsertPos
+    0.52%  clang-12                  [.] clang::Preprocessor::getMacroDefinition
     0.51%  clang-12                  [.] IgnoreParensSingleStep
     0.47%  clang-12                  [.] clang::Sema::LookupName
     0.47%  clang-12                  [.] clang::Parser::ParseDeclarationSpecifiers
     0.46%  libc-2.30.so              [.] cfree@GLIBC_2.2.5
     0.44%  clang-12                  [.] AnalyzeImplicitConversions
     0.44%  libc-2.30.so              [.] __strlen_avx2
     0.42%  clang-12                  [.] clang::Sema::ActOnFunctionDeclarator
     0.41%  clang-12                  [.] clang::MacroArgs::create
     0.41%  clang-12                  [.] clang::Parser::ParseCastExpression
     0.39%  clang-12                  [.] GetDeclSpecTypeForDeclarator
     0.38%  clang-12                  [.] clang::Preprocessor::ReadMacroCallArgumentList
     0.38%  libc-2.30.so              [.] __memset_avx2_unaligned_erms
     0.35%  clang-12                  [.] clang::StmtVisitorBase<llvm::make_const_ptr, (anonymous namespace)::IntExprEvaluator, bool>::Visit
     0.34%  clang-12                  [.] (anonymous namespace)::DataRecursiveIntBinOpEvaluator::shouldEnqueue
```

Low hanging fruit identified

Opportunities for improvement in LLVM found by compiling the Linux kernel:

- Redundant work inline asm statements (~13% of a build)
  - Fixed in clang-11.
- Expensive calculations tracking macro arguments source locations.
  - Particularly when GNU C statement expressions are passed to macros.
  - TODO: fix

Maybe not surprising it took building the Linux kernel to find these?

Perfetto RFC

Perfetto-instrumented Clang https://reviews.llvm.org/D82994

- Records time spent in frontend vs time spent in backend

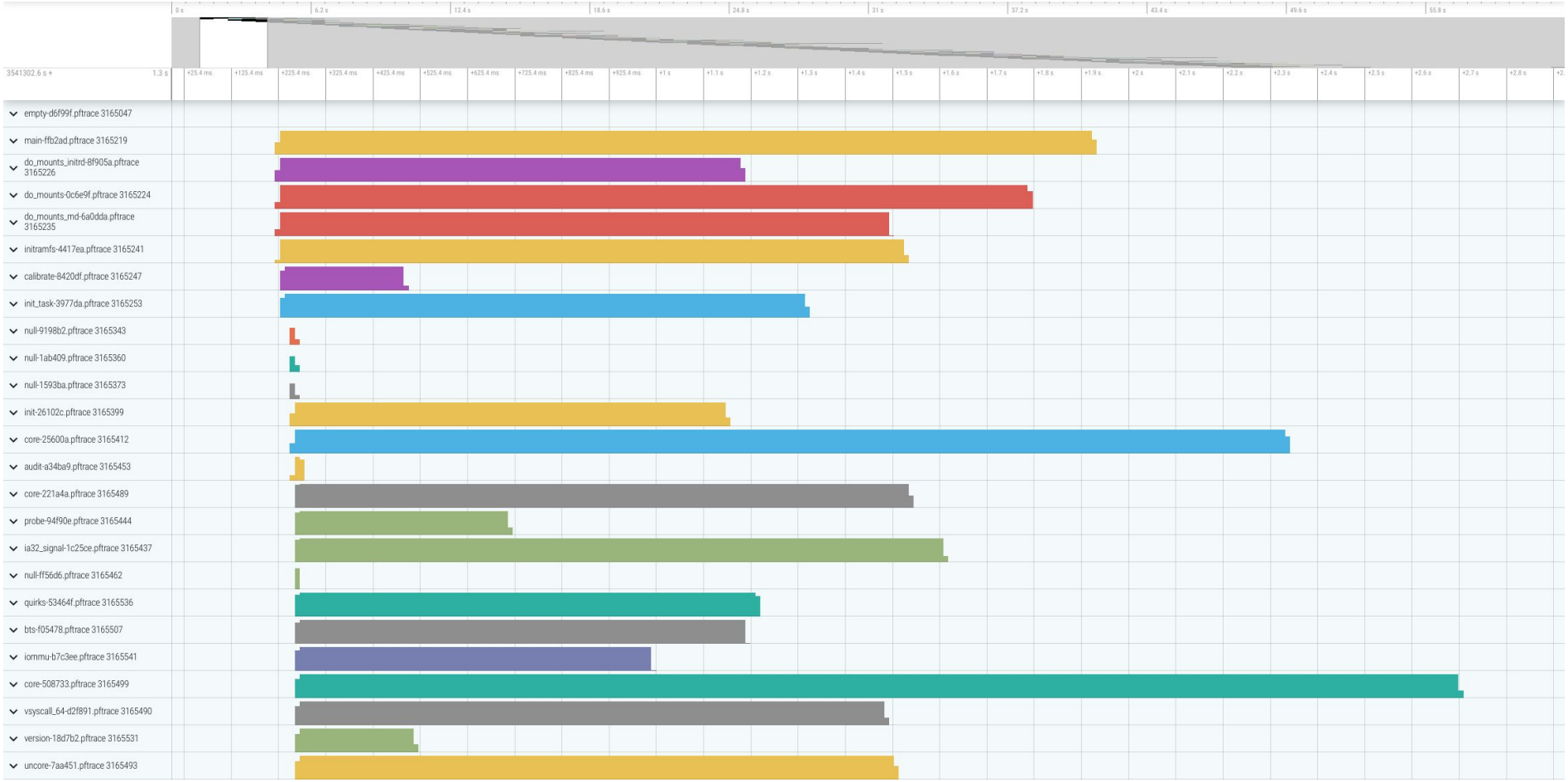
Built-in Query Engine - Useful for finding outliers

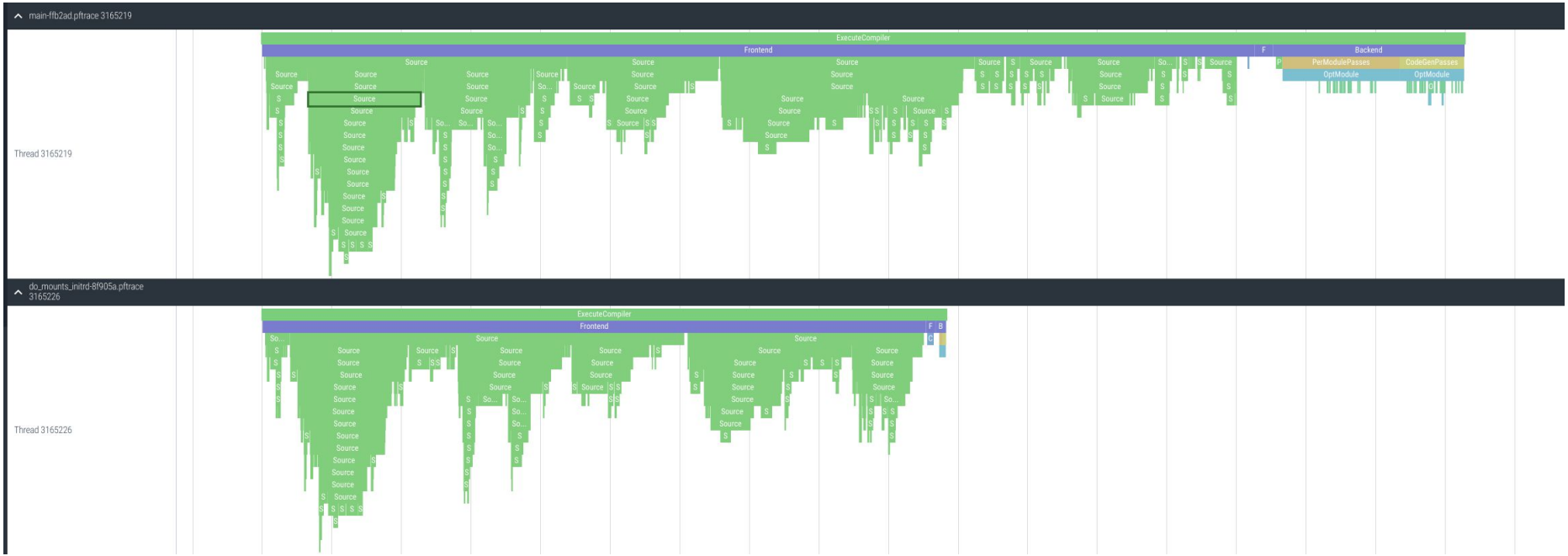
- Million character macro expansions
- Repeated inlining of large functions

Kernel builds:

X86_64: https://ui.perfetto.dev/#!/?s=bbbcbde1daa19cfe7c5746671c7cb6986eb605aa811be6f5cc24d9b853a8c

Arm64: https://ui.perfetto.dev/#!/?s=aba8804ce98f5467fd46a471b7581e6515b9d24c6aed0852f7f99cda7e2c6e2


0 s
6.2 s
12.4 s
18.6 s
24.8 s
31 s
37.2 s
43.4 s
49.6 s
55.8 s
3541302.6 s +
1.3 s
+25.4 ms
+125.4 ms
+225.4 ms
+325.4 ms
+425.4 ms
+525.4 ms
+625.4 ms
+725.4 ms
+825.4 ms
+925.4 ms
+1 s
+1.1 s
+1.2 s
+1.3 s
+1.4 s
+1.5 s
+1.6 s
+1.7 s
+1.8 s
+1.9 s
+2 s
+2.1 s
+2.2 s
+2.3 s
+2.4 s
+2.5 s
+2.6 s
+2.7 s
+2.8 s
empty-d6f99f.pftrace 3165047
main-ffb2ad.pftrace 3165219
do_mounts_initrd-8f905a.pftrace 3165226
do_mounts-0c6e9f.pftrace 3165224
do_mounts_md-6a0dda.pftrace 3165235
initramfs-4417ea.pftrace 3165241
calibrate-8420df.pftrace 3165247
init_task-3977da.pftrace 3165253
null-9198b2.pftrace 3165343
null-1ab409.pftrace 3165360
null-1593ba.pftrace 3165373
init-26102c.pftrace 3165399
core-25600a.pftrace 3165412
audit-a34ba9.pftrace 3165453
core-221a4a.pftrace 3165489
probe-94f90e.pftrace 3165444
ia32_signal-1c25ce.pftrace 3165437
null-ff56d6.pftrace 3165462
quirks-53464f.pftrace 3165536
bts-f05478.pftrace 3165507
iommu-b7c3ee.pftrace 3165541
core-508733.pftrace 3165499
vsyscall_64-d2f891.pftrace 3165490
version-18d7b2.pftrace 3165531
uncore-7aa451.pftrace 3165493


main-ffb2ad.pftrace 3165219
Thread 3165219
ExecuteCompiler
Frontend
Backend
PerModulePasses
CodeGenPasses
OptModule
OptModule
do_mounts_initrd-8f905a.pftrace 3165226
Thread 3165226
ExecuteCompiler
Frontend

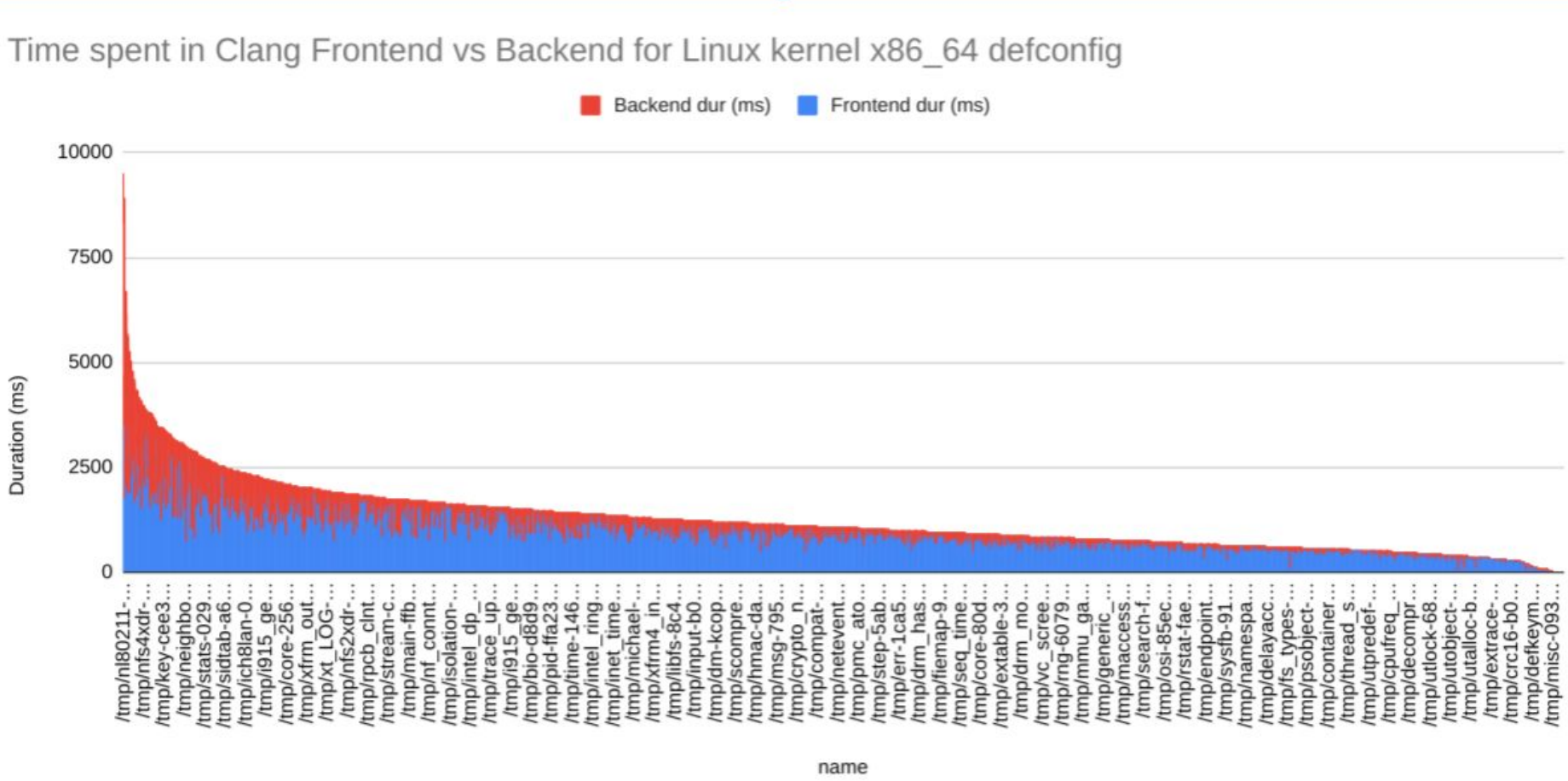
```
SELECT frontend.dur/backend.dur,frontend.dur AS "Frontend dur (ms)",backend.dur AS "Backend dur (ms)",process.name FROM
(SELECT track_id, SUM(dur)/1000.0/1000 AS "dur" FROM slice WHERE name == "Frontend" GROUP BY track_id) AS frontend
INNER JOIN
(SELECT track_id, SUM(dur)/1000.0/1000 AS "dur" FROM slice WHERE name == "Backend" GROUP BY track_id) AS backend
ON Frontend.track_id=Backend.track_id
INNER JOIN thread_track ON
frontend.track_id=thread_track.id
INNER JOIN thread ON
thread.utid=thread_track.utid
INNER JOIN process ON thread.upid=process.upid
ORDER BY frontend.dur/backend.dur desc
```

Query result - 65 ms

SELECT frontend.dur/backend.dur,frontend.dur AS "Frontend dur (ms)",backend.dur AS "Backend dur (ms)",process.name FROM (SELECT track_id, SUM(dur)/1000.0/1000 AS "dur" FROM slice WHERE name == "Frontend" GROUP BY track_id) AS frontend INNER JOIN (SELECT track_id, SUM(dur)/1000.0/1000 AS "dur" FROM slice WHERE name == "Backend" GROUP BY track_id) AS backend ON Frontend.track_id=Backend.track_id INNER JOIN thread_track ON frontend.track_id=thread_track.id INNER JOIN thread ON thread.utid=thread_track.utid INNER JOIN process ON thread.upid=process.upid ORDER BY frontend.dur/backend.dur desc

Copy as .tsv | Close

frontend.dur/backend.dur	Frontend dur (ms)	Backend dur (ms)	name
356.5384799024544	2014.688423	5.65069	/tmp/trace-78820e.pftrace
283.0332050711659	1937.7568370000001	6.846394	/tmp/nfs4trace-3341e7.pftrace
243.14600493584243	1510.548446	6.212516	/tmp/trace-348117.pftrace
224.01471063512844	1574.811767	7.029948	/tmp/nfstrace-81dd02.pftrace
210.9636585883103	1470.273667	6.969322	/tmp/shipped-certs-3541ff.pftrace
184.05446834636786	1305.8430779999999	7.094873	/tmp/trace-48219f.pftrace
167.91226940831882	1130.499578	6.73268	/tmp/agp-e94718.pftrace
164.90163879278444	955.87626	5.796645000000001	/tmp/ipa_data-sdm845-5fd073.pftrace
163.77403241153576	1161.280229	7.090747	/tmp/vc4_trace_points-85d542.pftrace
160.98960844510762	1038.235347	6.449083	/tmp/trace-c51bdb.pftrace
154.57254095011834	1013.947642	6.559688	/tmp/highmem-2b39f4.pftrace
151.75443293265988	977.1329840000001	6.438909	/tmp/int32-865962.pftrace
144.40311529971993	1720.971202	11.917826	/tmp/sysctl-cdc758.pftrace
143.0119915429036	1024.425643	7.163215	/tmp/int16-8e5462.pftrace
141.42046886759474	1256.162365	8.882465	/tmp/ipa_data-sc7180-901c07.pftrace
130.3526701719974	852.1025	6.536900999999999	/tmp/drm_trace_points-3ef945.pftrace
129.86741011310622	1063.132021	8.186288	/tmp/sysctl-00b1da.pftrace
129.1461812288446	967.394654	7.490695	/tmp/xhci-trace-8ffd7c.pftrace
124.34486049837034	744.282203	5.985629	/tmp/lima_trace-17d7a3.pftrace
122.88249869529437	849.775856	6.915353	/tmp/syscall-fa1425.pftrace
121.86906708337571	1647.502217	13.518625	/tmp/netlink-174ec7.pftrace
121.48996163247182	884.334178	7.279072	/tmp/trace-6ecde6.pftrace
118.64660201035662	829.39919	6.990501	/tmp/msm_gpu_tracepoints-5f6689.pftrace
118.40101825840478	1123.896328	9.492286	/tmp/drm_kms_helper_common-c0c5bc.pftrace
118.35776311296632	1857.5731329999999	15.694561	/tmp/nfs4sysctl-9e108f.pftrace
117.91147255136289	879.500023	7.4589859999999994	/tmp/trace-d3ca9b.pftrace
116.54686319447354	1121.269283	9.620759	/tmp/tables-2c5d70.pftrace
115.9611446993357	905.074879	7.804984	/tmp/fallback_table-35272c.pftrace
108.42810974198787	710.830508	6.555777	/tmp/nmi_backtrace-796e21.pftrace
108.15953201888416	594.3559889999999	5.495179	/tmp/utdebug-cbfe70.pftrace
107.939594455953	648.301827	6.006154	/tmp/sysctl-54d500.pftrace
104.16662648070603	544.7761439999999	5.229853	/tmp/evxfevnt-9c1b32.pftrace
104.13642792585917	586.590189	5.6329009999999995	/tmp/rsirq-2c8c23.pftrace
101.56392448656463	446.645741	4.3976809999999995	/tmp/evgpeinit-eb4441.pftrace
100.98225022765743	591.945631	5.861878	/tmp/hwsleep-d9d899.pftrace
100.9710704144336	827.479025	8.195209	/tmp/deflate_syms-654f48.pftrace
98.87999361407934	798.358255	8.074012	/tmp/font_8x16-6a05d1.pftrace
98.3339696962436	767.422588	7.804247	/tmp/rsdumpinfo-06f679.pftrace
97.67702398644553	579.4991269999999	5.932809	/tmp/evsci-b80e2d.pftrace


Time spent in Clang Frontend vs Backend for Linux kernel x86_64 defconfig
Backend dur (ms)
Frontend dur (ms)
Duration (ms)
10000
7500
5000
2500
0
name

Future work?

- Keeping track of major regressions in compile time (something like llvm-compile-time-tracker but specifically for the kernel)
- Analyze traces in depth