Towards the performant P4C

Anton Korobeynikov, Principal Software Engineer, Compiler Development

Access Softek Toolchains, www.softek-toolchains.com

ACCESS SOFTEK, INC

About myself

- Long-term contributor to LLVM
  - First contributions date back to 2005
  - Code owner of MSP430 backend; many contributions to different parts of LLVM
- Some contributions to gcc & derivatives
  - Among primary authors of llvm-gcc 4.2
- Contributions to Swift
  - Mostly automatic differentiation support (Differential Swift)
- Some other open-source projects
  - Sometimes not even compiler-related

Rationale

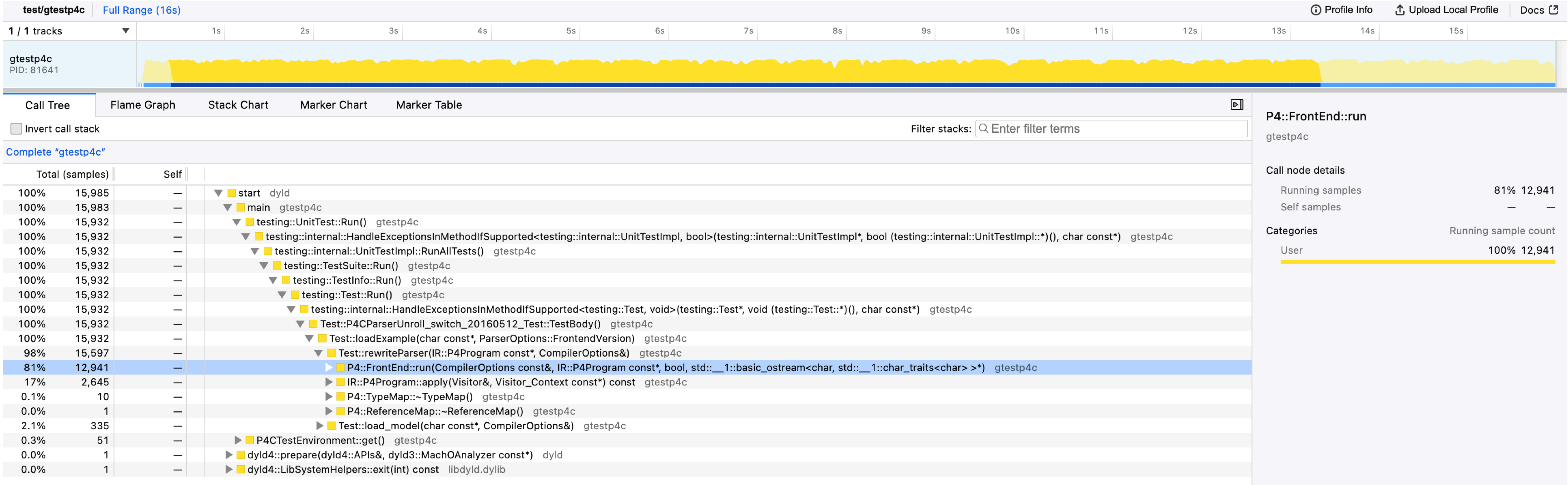
- P4C is a reference compiler for P4 language
  - Usually reference implementations are not required to be fast & efficient
- But there is no other production-ready P4 compiler around…
  - Downstream users rely on P4C
- So we either need to improve P4C
- … or develop something that could be used instead
  - Does not seem to be a viable option today

Baseline

- P4C v1.2.4.8 (released ~January 2024)
- `P4CParserUnroll.switch_20160512` gtest tescase
  - Run via `test/gtestp4c --gtest_filter=P4CParserUnroll.switch_20160512`
  - Source code in `testdata/p4_14_samples/switch_20160512`
  - ~9k lines of P4-14 code
- Benchmarking via Hyperfine (20 runs + 1 warm-up) on Apple M1 Pro Laptop:
  - Time (mean ± σ): 15.193 s ± 0.303 s [User: 15.044 s, System: 0.101 s]
  - Range (min … max): 14.749 s … 16.083 s 20 runs

This looks quite a lot given the input size

Profile



□ Invert call stack

Filter stacks: Enter filter terms

Complete "gtestp4c"

Total (samples)		Self	
100%	15,985	—	▼ start dyld
100%	15,983	—	▼ main gtestp4c
100%	15,932	—	▼ testing::UnitTest::Run() gtestp4c
100%	15,932	—	▼ testing::internal::HandleExceptionsInMethodIfSupported<testing::internal::UnitTestImpl, bool>(testing::internal::UnitTestImpl*, bool (testing::internal::UnitTestImpl::*)(), char const*) gtestp4c
100%	15,932	—	▼ testing::internal::UnitTestImpl::RunAllTests() gtestp4c
100%	15,932	—	▼ testing::TestSuite::Run() gtestp4c
100%	15,932	—	▼ testing::TestInfo::Run() gtestp4c
100%	15,932	—	▼ testing::Test::Run() gtestp4c
100%	15,932	—	▼ testing::internal::HandleExceptionsInMethodIfSupported<testing::Test, void>(testing::Test*, void (testing::Test::*)(), char const*) gtestp4c
100%	15,932	—	▼ Test::P4CParserUnroll_switch_20160512_Test::TestBody() gtestp4c
100%	15,932	—	▼ Test::loadExample(char const*, ParserOptions::FrontendVersion) gtestp4c
98%	15,597	—	▼ Test::rewriteParser(IR::P4Program const*, CompilerOptions&) gtestp4c
81%	12,941	—	▶ P4::FrontEnd::run(CompilerOptions const&, IR::P4Program const*, bool, std::__1::basic_ostream<char, std::__1::char_traits<char> >*) gtestp4c
17%	2,645	—	▶ IR::P4Program::apply(Visitor&, Visitor_Context const*) const gtestp4c
0.1%	10	—	▶ P4::TypeMap::~TypeMap() gtestp4c
0.0%	1	—	▶ P4::ReferenceMap::~ReferenceMap() gtestp4c
2.1%	335	—	▶ Test::load_model(char const*, CompilerOptions&) gtestp4c
0.3%	51	—	▶ P4CTestEnvironment::get() gtestp4c
0.0%	1	—	▶ dyld4::prepare(dyld4::APIs&, dyld3::MachOAnalyzer const*) dyld
0.0%	1	—	▶ dyld4::LibSystemHelpers::exit(int) const libdyld.dylib

P4::FrontEnd::run

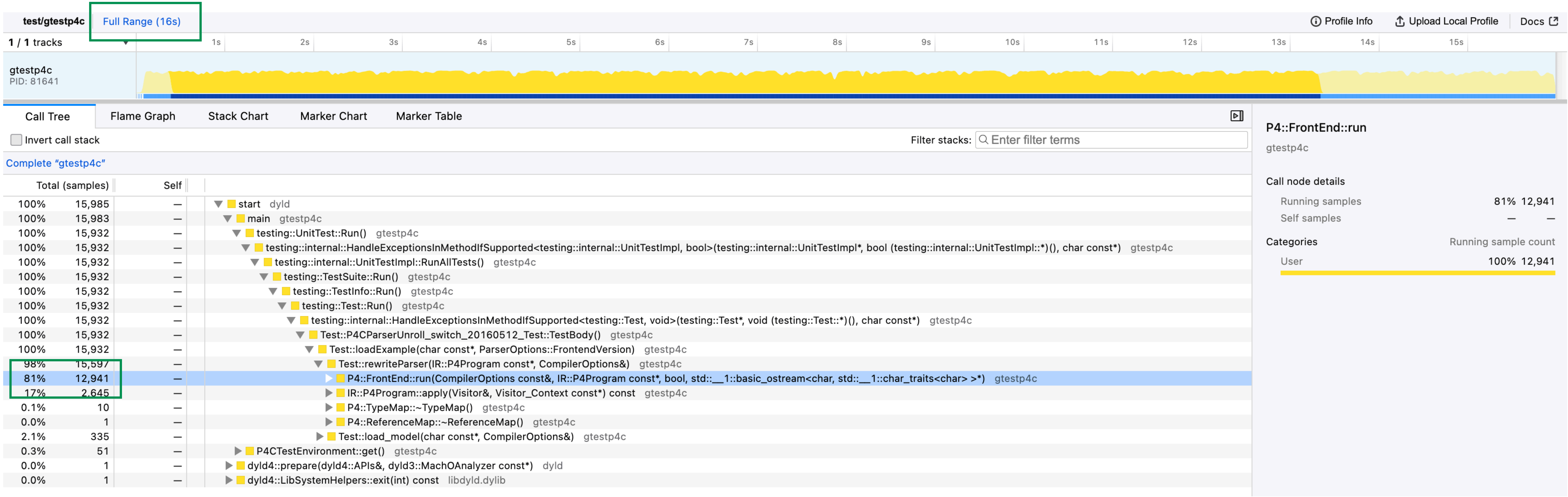
gtestp4c

Call node details

Running samples	81% 12,941
Self samples	— —

Categories	Running sample count
User	100% 12,941

Profile



☐ Invert call stack

Total (samples)		Self	
100%	15,985	—	▼ start dyld
100%	15,983	—	▼ main gtestp4c
100%	15,932	—	▼ testing::UnitTest::Run() gtestp4c
100%	15,932	—	▼ testing::internal::HandleExceptionsInMethodIfSupported<testing::internal::UnitTestImpl, bool>(testing::internal::UnitTestImpl*, bool (testing::internal::UnitTestImpl::*)(), char const*) gtestp4c
100%	15,932	—	▼ testing::internal::UnitTestImpl::RunAllTests() gtestp4c
100%	15,932	—	▼ testing::TestSuite::Run() gtestp4c
100%	15,932	—	▼ testing::TestInfo::Run() gtestp4c
100%	15,932	—	▼ testing::Test::Run() gtestp4c
100%	15,932	—	▼ testing::internal::HandleExceptionsInMethodIfSupported<testing::Test, void>(testing::Test*, void (testing::Test::*)(), char const*) gtestp4c
100%	15,932	—	▼ Test::P4CParserUnroll_switch_20160512_Test::TestBody() gtestp4c
100%	15,932	—	▼ Test::loadExample(char const*, ParserOptions::FrontendVersion) gtestp4c
98%	15,597	—	▼ Test::rewriteParser(IR::P4Program const*, CompilerOptions&) gtestp4c
81%	12,941	—	▶ P4::FrontEnd::run(CompilerOptions const&, IR::P4Program const*, bool, std::__1::basic_ostream<char, std::__1::char_traits<char> >*) gtestp4c
17%	2,645	—	▶ IR::P4Program::apply(Visitor&, Visitor_Context const*) const gtestp4c
0.1%	10	—	▶ P4::TypeMap::~TypeMap() gtestp4c
0.0%	1	—	▶ P4::ReferenceMap::~ReferenceMap() gtestp4c
2.1%	335	—	▶ Test::load_model(char const*, CompilerOptions&) gtestp4c
0.3%	51	—	▶ P4CTestEnvironment::get() gtestp4c
0.0%	1	—	▶ dyld4::prepare(dyld4::APIs&, dyld3::MachOAnalyzer const*) dyld
0.0%	1	—	▶ dyld4::LibSystemHelpers::exit(int) const libdyld.dylib

P4::FrontEnd::run

gtestp4c

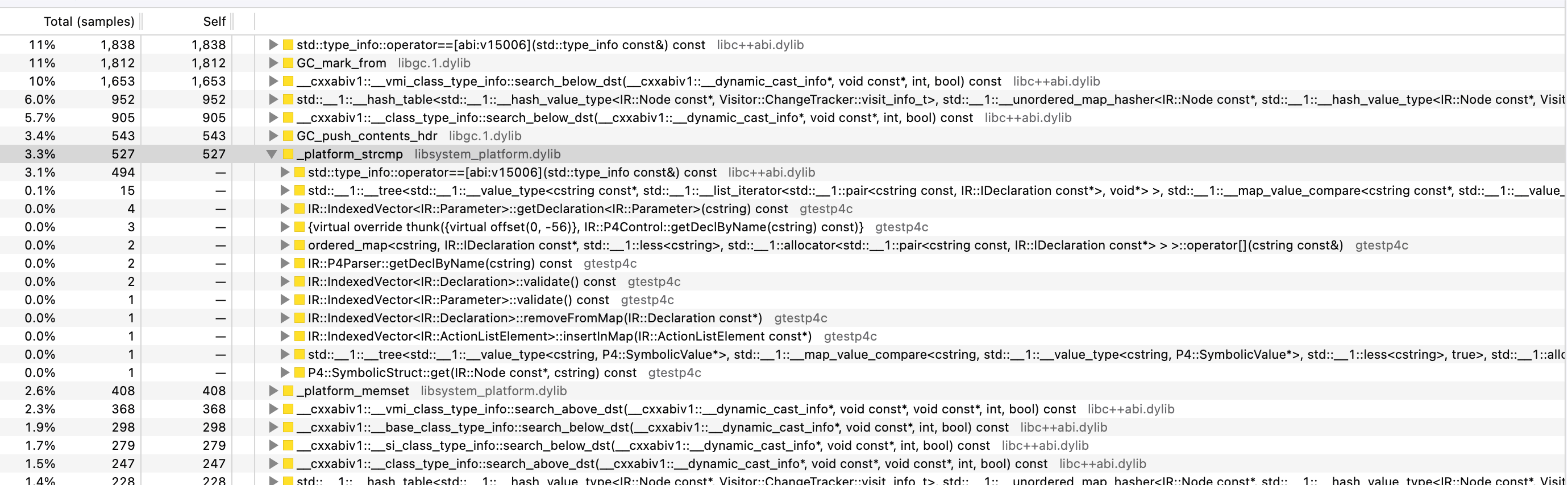
Call node details

Running samples	81% 12,941
Self samples	— —

Categories	Running sample count
User	100% 12,941

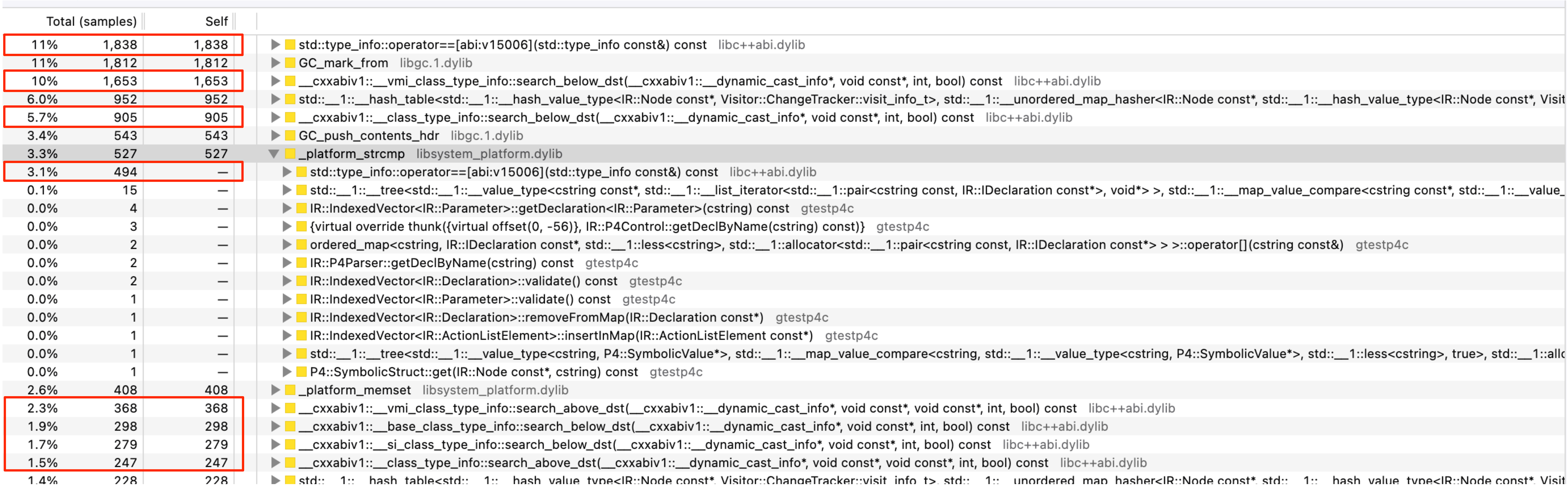
Frontend takes 81% of entire 16s compilation time

Profile with inverted call stack



Total (samples)		Self	
11%	1,838	1,838	▶ std::type_info::operator==[abi:v15006](std::type_info const&) const libc++abi.dylib
11%	1,812	1,812	▶ GC_mark_from libgc.1.dylib
10%	1,653	1,653	▶ __cxxabiv1::__vmi_class_type_info::search_below_dst(__cxxabiv1::__dynamic_cast_info*, void const*, int, bool) const libc++abi.dylib
6.0%	952	952	▶ std::__1::__hash_table<std::__1::__hash_value_type<IR::Node const*, Visitor::ChangeTracker::visit_info_t>, std::__1::__unordered_map_hasher<IR::Node const*, std::__1::__hash_value_type<IR::Node const*, Visit
5.7%	905	905	▶ __cxxabiv1::__class_type_info::search_below_dst(__cxxabiv1::__dynamic_cast_info*, void const*, int, bool) const libc++abi.dylib
3.4%	543	543	▶ GC_push_contents_hdr libgc.1.dylib
3.3%	527	527	▼ _platform_strcmp libsystem_platform.dylib
3.1%	494	–	▶ std::type_info::operator==[abi:v15006](std::type_info const&) const libc++abi.dylib
0.1%	15	–	▶ std::__1::__tree<std::__1::__value_type<cstring const*, std::__1::__list_iterator<std::__1::pair<cstring const, IR::IDeclaration const*>, void*> >, std::__1::__map_value_compare<cstring const*, std::__1::__value_
0.0%	4	–	▶ IR::IndexedVector<IR::Parameter>::getDeclaration<IR::Parameter>(cstring) const gtestp4c
0.0%	3	–	▶ {virtual override thunk({virtual offset(0, -56)}, IR::P4Control::getDeclByName(cstring) const)} gtestp4c
0.0%	2	–	▶ ordered_map<cstring, IR::IDeclaration const*, std::__1::less<cstring>, std::__1::allocator<std::__1::pair<cstring const, IR::IDeclaration const*> > >::operator[](cstring const&) gtestp4c
0.0%	2	–	▶ IR::P4Parser::getDeclByName(cstring) const gtestp4c
0.0%	2	–	▶ IR::IndexedVector<IR::Declaration>::validate() const gtestp4c
0.0%	1	–	▶ IR::IndexedVector<IR::Parameter>::validate() const gtestp4c
0.0%	1	–	▶ IR::IndexedVector<IR::Declaration>::removeFromMap(IR::Declaration const*) gtestp4c
0.0%	1	–	▶ IR::IndexedVector<IR::ActionListElement>::insertInMap(IR::ActionListElement const*) gtestp4c
0.0%	1	–	▶ std::__1::__tree<std::__1::__value_type<cstring, P4::SymbolicValue*>, std::__1::__map_value_compare<cstring, std::__1::__value_type<cstring, P4::SymbolicValue*>, std::__1::less<cstring>, true>, std::__1::allo
0.0%	1	–	▶ P4::SymbolicStruct::get(IR::Node const*, cstring) const gtestp4c
2.6%	408	408	▶ _platform_memset libsystem_platform.dylib
2.3%	368	368	▶ __cxxabiv1::__vmi_class_type_info::search_above_dst(__cxxabiv1::__dynamic_cast_info*, void const*, void const*, int, bool) const libc++abi.dylib
1.9%	298	298	▶ __cxxabiv1::__base_class_type_info::search_below_dst(__cxxabiv1::__dynamic_cast_info*, void const*, int, bool) const libc++abi.dylib
1.7%	279	279	▶ __cxxabiv1::__si_class_type_info::search_below_dst(__cxxabiv1::__dynamic_cast_info*, void const*, int, bool) const libc++abi.dylib
1.5%	247	247	▶ __cxxabiv1::__class_type_info::search_above_dst(__cxxabiv1::__dynamic_cast_info*, void const*, void const*, int, bool) const libc++abi.dylib
1.4%	228	228	▶ std::__1::__hash_table<std::__1::__hash_value_type<IR::Node const*, Visitor::ChangeTracker::visit_info_t>, std::__1::__unordered_map_hasher<IR::Node const*, std::__1::__hash_value_type<IR::Node const*, Visit

Profile with inverted call stack

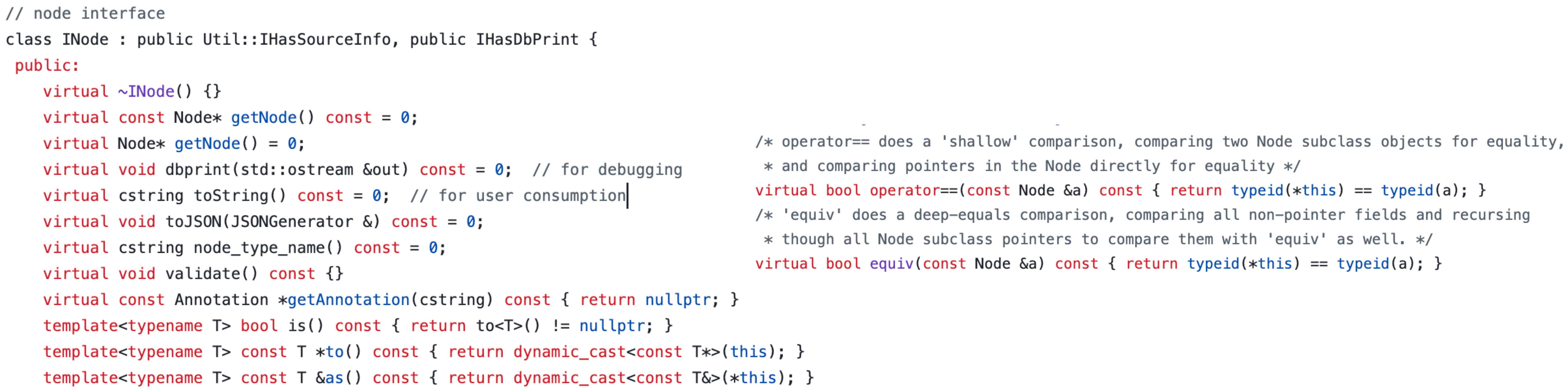


Total (samples)		Self	
11%	1,838	1,838	▶ std::type_info::operator==[abi:v15006](std::type_info const&) const libc++abi.dylib
11%	1,812	1,812	▶ GC_mark_from libgc.1.dylib
10%	1,653	1,653	▶ __cxxabiv1::__vmi_class_type_info::search_below_dst(__cxxabiv1::__dynamic_cast_info*, void const*, int, bool) const libc++abi.dylib
6.0%	952	952	▶ std::__1::__hash_table<std::__1::__hash_value_type<IR::Node const*, Visitor::ChangeTracker::visit_info_t>, std::__1::__unordered_map_hasher<IR::Node const*, std::__1::__hash_value_type<IR::Node const*, Visit
5.7%	905	905	▶ __cxxabiv1::__class_type_info::search_below_dst(__cxxabiv1::__dynamic_cast_info*, void const*, int, bool) const libc++abi.dylib
3.4%	543	543	▶ GC_push_contents_hdr libgc.1.dylib
3.3%	527	527	▼ _platform_strcmp libsystem_platform.dylib
3.1%	494	–	▶ std::type_info::operator==[abi:v15006](std::type_info const&) const libc++abi.dylib
0.1%	15	–	▶ std::__1::__tree<std::__1::__value_type<cstring const*, std::__1::__list_iterator<std::__1::pair<cstring const, IR::IDeclaration const*>, void*> >, std::__1::__map_value_compare<cstring const*, std::__1::__value_
0.0%	4	–	▶ IR::IndexedVector<IR::Parameter>::getDeclaration<IR::Parameter>(cstring) const gtestp4c
0.0%	3	–	▶ {virtual override thunk({virtual offset(0, -56)}, IR::P4Control::getDeclByName(cstring) const)} gtestp4c
0.0%	2	–	▶ ordered_map<cstring, IR::IDeclaration const*, std::__1::less<cstring>, std::__1::allocator<std::__1::pair<cstring const, IR::IDeclaration const*> > >::operator[](cstring const&) gtestp4c
0.0%	2	–	▶ IR::P4Parser::getDeclByName(cstring) const gtestp4c
0.0%	2	–	▶ IR::IndexedVector<IR::Declaration>::validate() const gtestp4c
0.0%	1	–	▶ IR::IndexedVector<IR::Parameter>::validate() const gtestp4c
0.0%	1	–	▶ IR::IndexedVector<IR::Declaration>::removeFromMap(IR::Declaration const*) gtestp4c
0.0%	1	–	▶ IR::IndexedVector<IR::ActionListElement>::insertInMap(IR::ActionListElement const*) gtestp4c
0.0%	1	–	▶ std::__1::__tree<std::__1::__value_type<cstring, P4::SymbolicValue*>, std::__1::__map_value_compare<cstring, std::__1::__value_type<cstring, P4::SymbolicValue*>, std::__1::less<cstring>, true>, std::__1::allc
0.0%	1	–	▶ P4::SymbolicStruct::get(IR::Node const*, cstring) const gtestp4c
2.6%	408	408	▶ _platform_memset libsystem_platform.dylib
2.3%	368	368	▶ __cxxabiv1::__vmi_class_type_info::search_above_dst(__cxxabiv1::__dynamic_cast_info*, void const*, void const*, int, bool) const libc++abi.dylib
1.9%	298	298	▶ __cxxabiv1::__base_class_type_info::search_below_dst(__cxxabiv1::__dynamic_cast_info*, void const*, int, bool) const libc++abi.dylib
1.7%	279	279	▶ __cxxabiv1::__si_class_type_info::search_below_dst(__cxxabiv1::__dynamic_cast_info*, void const*, int, bool) const libc++abi.dylib
1.5%	247	247	▶ __cxxabiv1::__class_type_info::search_above_dst(__cxxabiv1::__dynamic_cast_info*, void const*, void const*, int, bool) const libc++abi.dylib
1.4%	228	228	▶ std::__1::__hash_table<std::__1::__hash_value_type<IR::Node const*, Visitor::ChangeTracker::visit_info_t>, std::__1::__unordered_map_hasher<IR::Node const*, std::__1::__hash_value_type<IR::Node const*, Visit

37.2% of all compilation time is consumed by RTTI (`dynamic_cast / typeid`)!

RTTI in P4C

Where does RTTI usage come from?

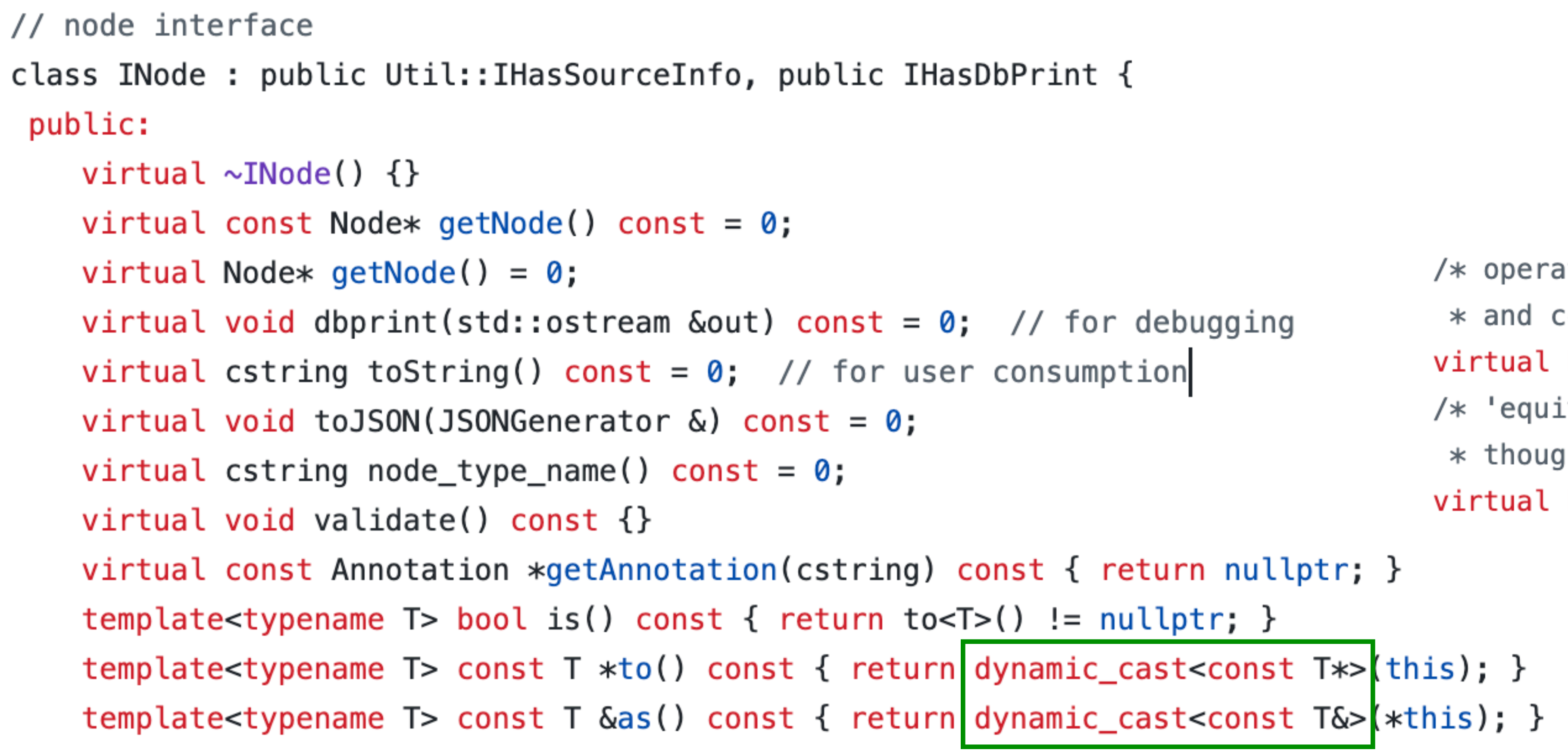


```
// node interface
class INode : public Util::IHasSourceInfo, public IHasDbPrint {
 public:
    virtual ~INode() {}
    virtual const Node* getNode() const = 0;
    virtual Node* getNode() = 0;
    virtual void dbprint(std::ostream &out) const = 0;  // for debugging
    virtual cstring toString() const = 0;  // for user consumption
    virtual void toJSON(JSONGenerator &) const = 0;
    virtual cstring node_type_name() const = 0;
    virtual void validate() const {}
    virtual const Annotation *getAnnotation(cstring) const { return nullptr; }
    template<typename T> bool is() const { return to<T>() != nullptr; }
    template<typename T> const T *to() const { return dynamic_cast<const T*>(this); }
    template<typename T> const T &as() const { return dynamic_cast<const T&>(*this); }
```

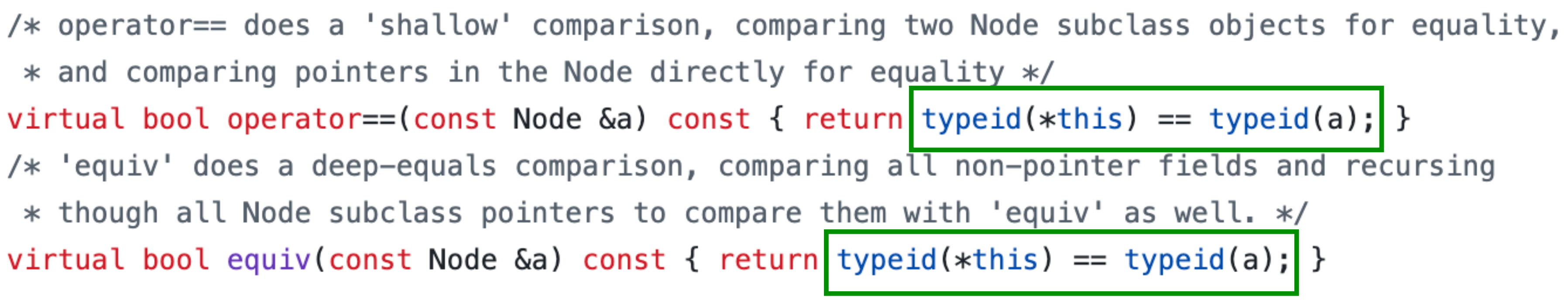
```
/* operator== does a 'shallow' comparison, comparing two Node subclass objects for equality,
 * and comparing pointers in the Node directly for equality */
virtual bool operator==(const Node &a) const { return typeid(*this) == typeid(a); }
/* 'equiv' does a deep-equals comparison, comparing all non-pointer fields and recursing
 * though all Node subclass pointers to compare them with 'equiv' as well. */
virtual bool equiv(const Node &a) const { return typeid(*this) == typeid(a); }
```

RTTI in P4C

Where does RTTI usage come from?



```
// node interface
class INode : public Util::IHasSourceInfo, public IHasDbPrint {
 public:
    virtual ~INode() {}
    virtual const Node* getNode() const = 0;
    virtual Node* getNode() = 0;
    virtual void dbprint(std::ostream &out) const = 0;  // for debugging
    virtual cstring toString() const = 0;  // for user consumption
    virtual void toJSON(JSONGenerator &) const = 0;
    virtual cstring node_type_name() const = 0;
    virtual void validate() const {}
    virtual const Annotation *getAnnotation(cstring) const { return nullptr; }
    template<typename T> bool is() const { return to<T>() != nullptr; }
    template<typename T> const T *to() const { return dynamic_cast<const T*>(this); }
    template<typename T> const T &as() const { return dynamic_cast<const T&>(*this); }
```



```
/* operator== does a 'shallow' comparison, comparing two Node subclass objects for equality,
 * and comparing pointers in the Node directly for equality */
virtual bool operator==(const Node &a) const { return typeid(*this) == typeid(a); }
/* 'equiv' does a deep-equals comparison, comparing all non-pointer fields and recursing
 * though all Node subclass pointers to compare them with 'equiv' as well. */
virtual bool equiv(const Node &a) const { return typeid(*this) == typeid(a); }
```

Downcasting & identity checks

RTTI

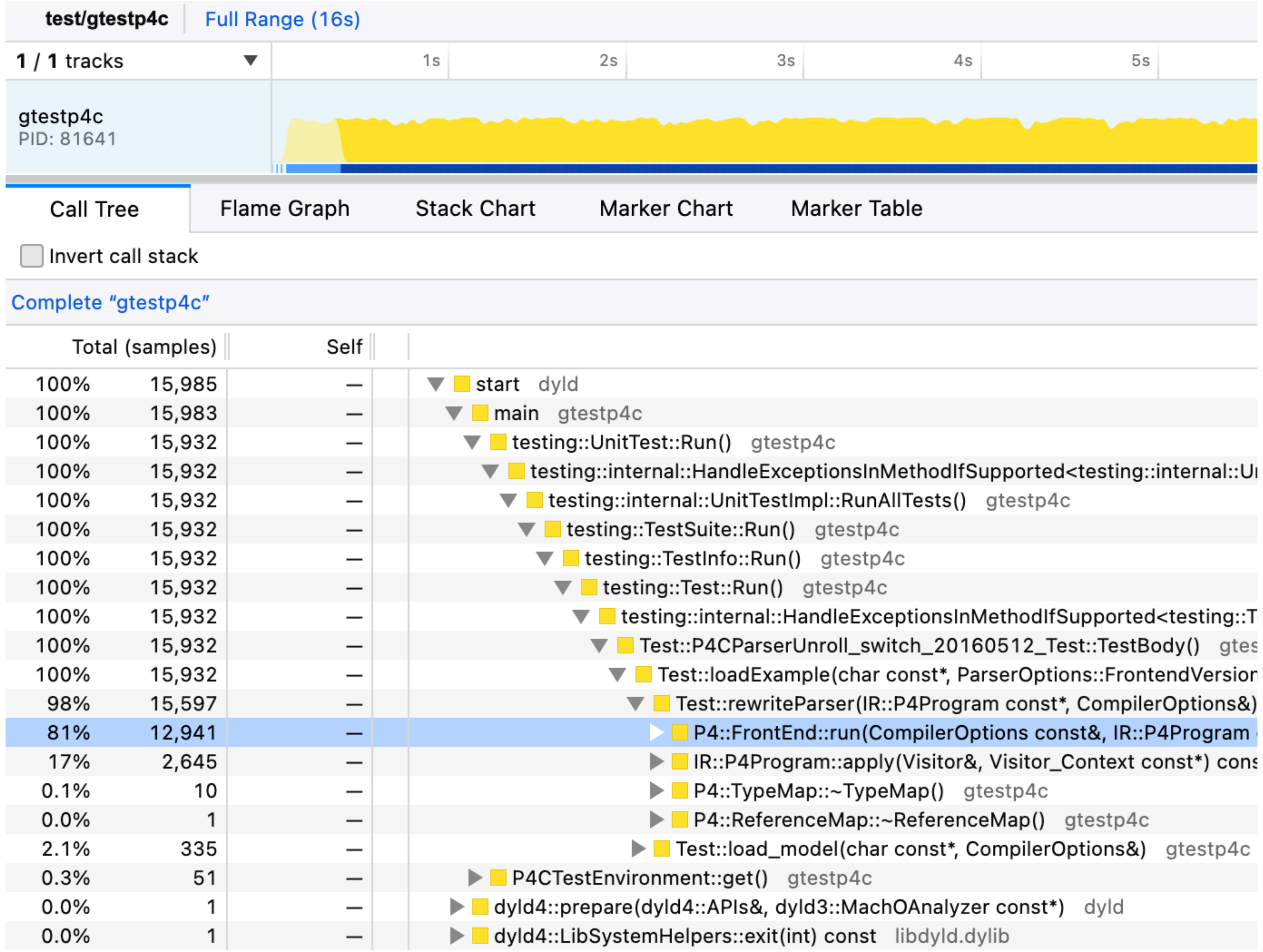
- Generic RTTI is slow:
  - has to deal with arbitrary open class hierarchies,
  - relies on compiler-generated metadata,
  - hard to inline / optimize, etc.
- Many projects implemented their own RTTI for closed / semi-closed class hierarchies
  - LLVM / clang
  - MFC
  - Unreal Engine & other game engines (AWS Lumberyard, …)
- Cannot use the lightweight static LLVM-style RTTI for P4C IR nodes:
  - Multiple inheritance
  - Abstract & virtual base classes
  - Cannot use `static_cast` for downcast, need to know the offset of base class in derived

New P4C RTTI Implementation

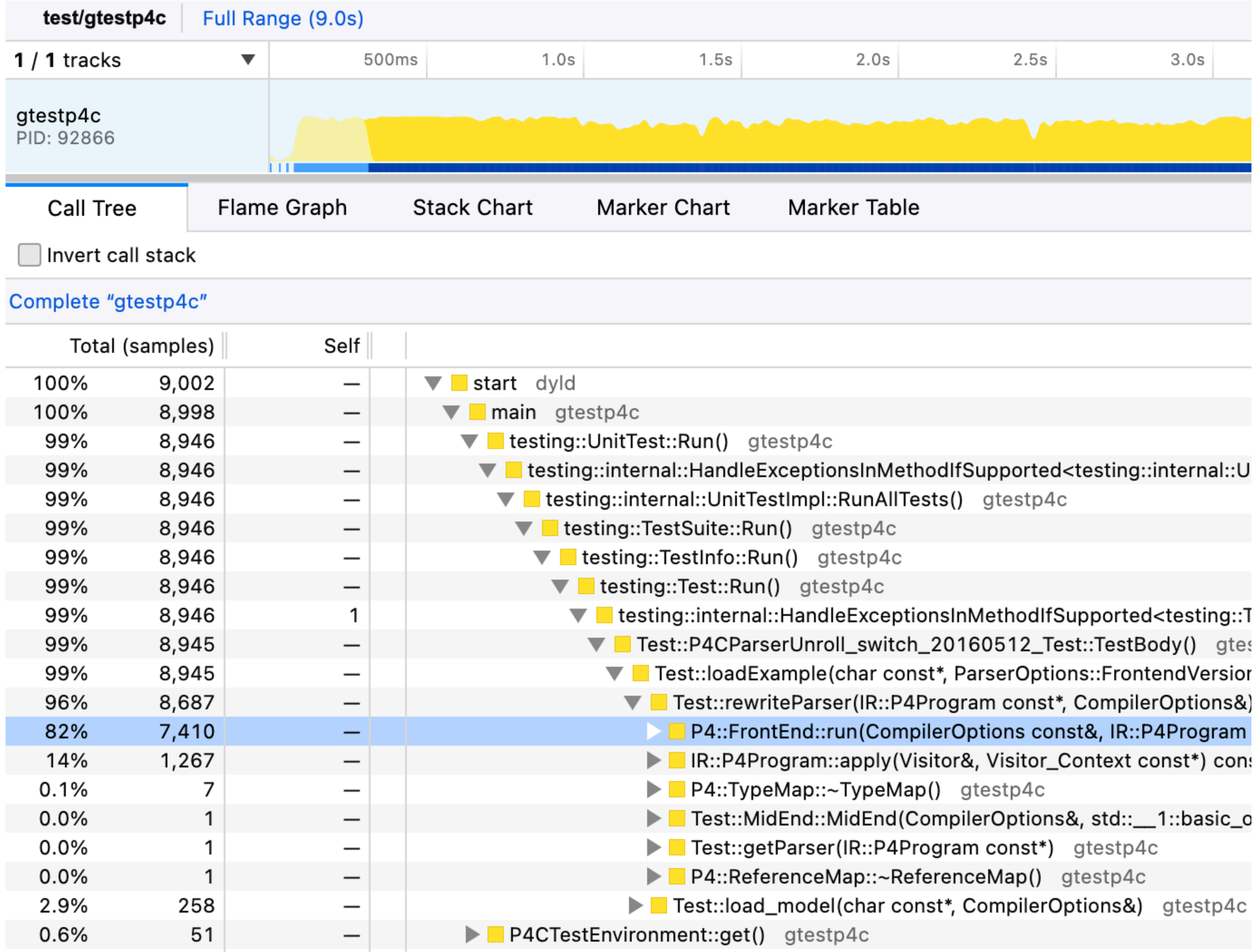
- `typeid` is generated from node type name at compile time
- Supports semi-open-ended class hierarchies
  - Need to derive from the single base class (`RTTI::Base`) that does heavy lifting & actual implementation
- Supports multiple inheritance and virtual base classes:
  - Compiler generates necessary `this` adjustment for us via a virtual function call
- Some boilerplate code hidden behind macros (autogenerated for `Node`)
- Provides `is<T>(), typeId(), to<T>()` class methods
- Downstream code that uses `dynamic_cast` / `typeid` on Node pointers works as usual

Overhead: one virtual call + some easily optimizable code

Results

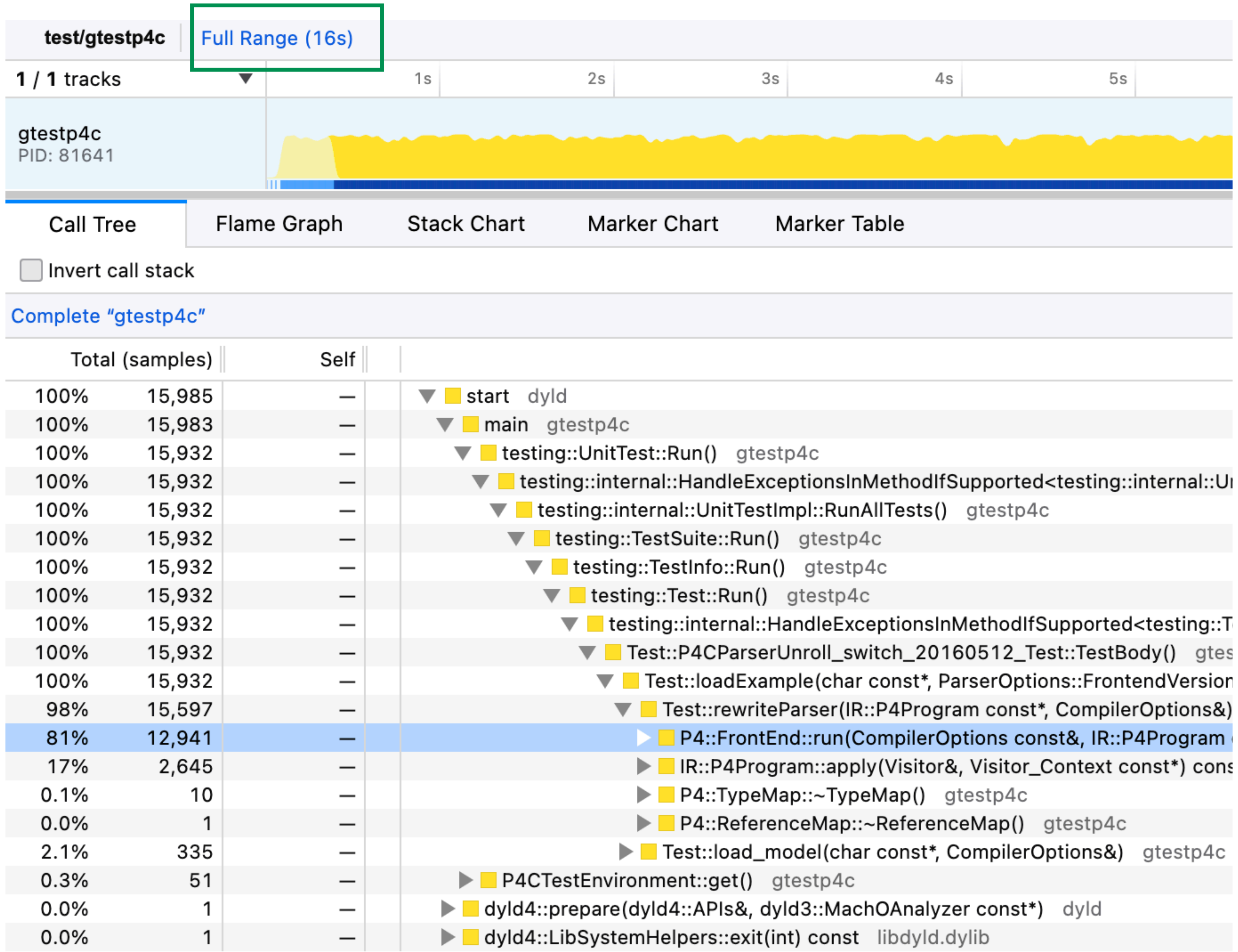


Total (samples)		Self	
100%	15,985	—	start dyld
100%	15,983	—	main gtestp4c
100%	15,932	—	testing::UnitTest::Run() gtestp4c
100%	15,932	—	testing::internal::HandleExceptionsInMethodIfSupported<testing::internal::U
100%	15,932	—	testing::internal::UnitTestImpl::RunAllTests() gtestp4c
100%	15,932	—	testing::TestSuite::Run() gtestp4c
100%	15,932	—	testing::TestInfo::Run() gtestp4c
100%	15,932	—	testing::Test::Run() gtestp4c
100%	15,932	—	testing::internal::HandleExceptionsInMethodIfSupported<testing::T
100%	15,932	—	Test::P4CParserUnroll_switch_20160512_Test::TestBody() gtes
100%	15,932	—	Test::loadExample(char const*, ParserOptions::FrontendVersion
98%	15,597	—	Test::rewriteParser(IR::P4Program const*, CompilerOptions&)
81%	12,941	—	P4::FrontEnd::run(CompilerOptions const&, IR::P4Program
17%	2,645	—	IR::P4Program::apply(Visitor&, Visitor_Context const*) cons
0.1%	10	—	P4::TypeMap::~TypeMap() gtestp4c
0.0%	1	—	P4::ReferenceMap::~ReferenceMap() gtestp4c
2.1%	335	—	Test::load_model(char const*, CompilerOptions&) gtestp4c
0.3%	51	—	P4CTestEnvironment::get() gtestp4c
0.0%	1	—	dyld4::prepare(dyld4::APIs&, dyld3::MachOAnalyzer const*) dyld
0.0%	1	—	dyld4::LibSystemHelpers::exit(int) const libdyld.dylib

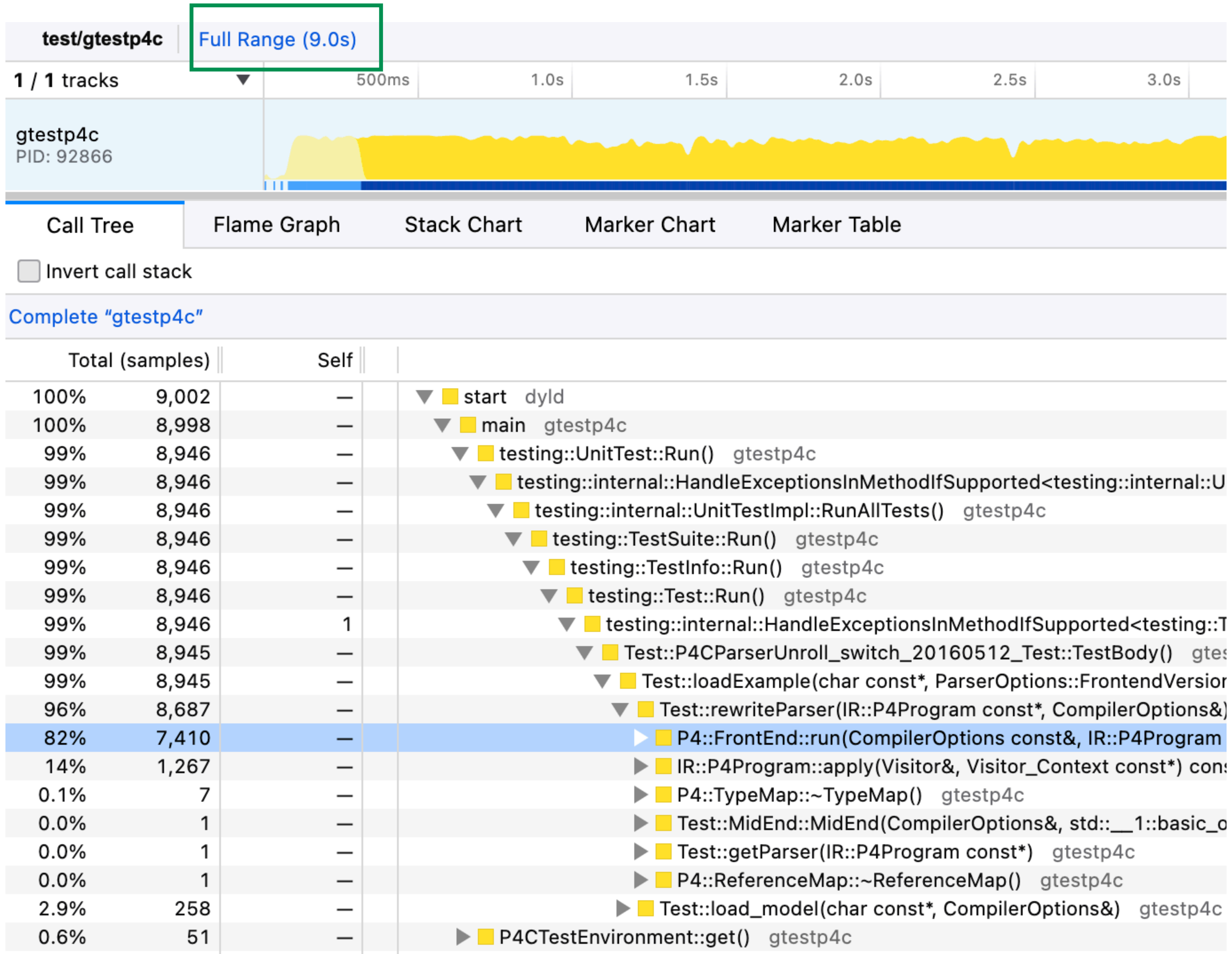


Total (samples)		Self	
100%	9,002	—	start dyld
100%	8,998	—	main gtestp4c
99%	8,946	—	testing::UnitTest::Run() gtestp4c
99%	8,946	—	testing::internal::HandleExceptionsInMethodIfSupported<testing::internal::U
99%	8,946	—	testing::internal::UnitTestImpl::RunAllTests() gtestp4c
99%	8,946	—	testing::TestSuite::Run() gtestp4c
99%	8,946	—	testing::TestInfo::Run() gtestp4c
99%	8,946	—	testing::Test::Run() gtestp4c
99%	8,946	1	testing::internal::HandleExceptionsInMethodIfSupported<testing::T
99%	8,945	—	Test::P4CParserUnroll_switch_20160512_Test::TestBody() gtes
99%	8,945	—	Test::loadExample(char const*, ParserOptions::FrontendVersior
96%	8,687	—	Test::rewriteParser(IR::P4Program const*, CompilerOptions&)
82%	7,410	—	P4::FrontEnd::run(CompilerOptions const&, IR::P4Program
14%	1,267	—	IR::P4Program::apply(Visitor&, Visitor_Context const*) cons
0.1%	7	—	P4::TypeMap::~TypeMap() gtestp4c
0.0%	1	—	Test::MidEnd::MidEnd(CompilerOptions&, std::__1::basic_o
0.0%	1	—	Test::getParser(IR::P4Program const*) gtestp4c
0.0%	1	—	P4::ReferenceMap::~ReferenceMap() gtestp4c
2.9%	258	—	Test::load_model(char const*, CompilerOptions&) gtestp4c
0.6%	51	—	P4CTestEnvironment::get() gtestp4c

Results



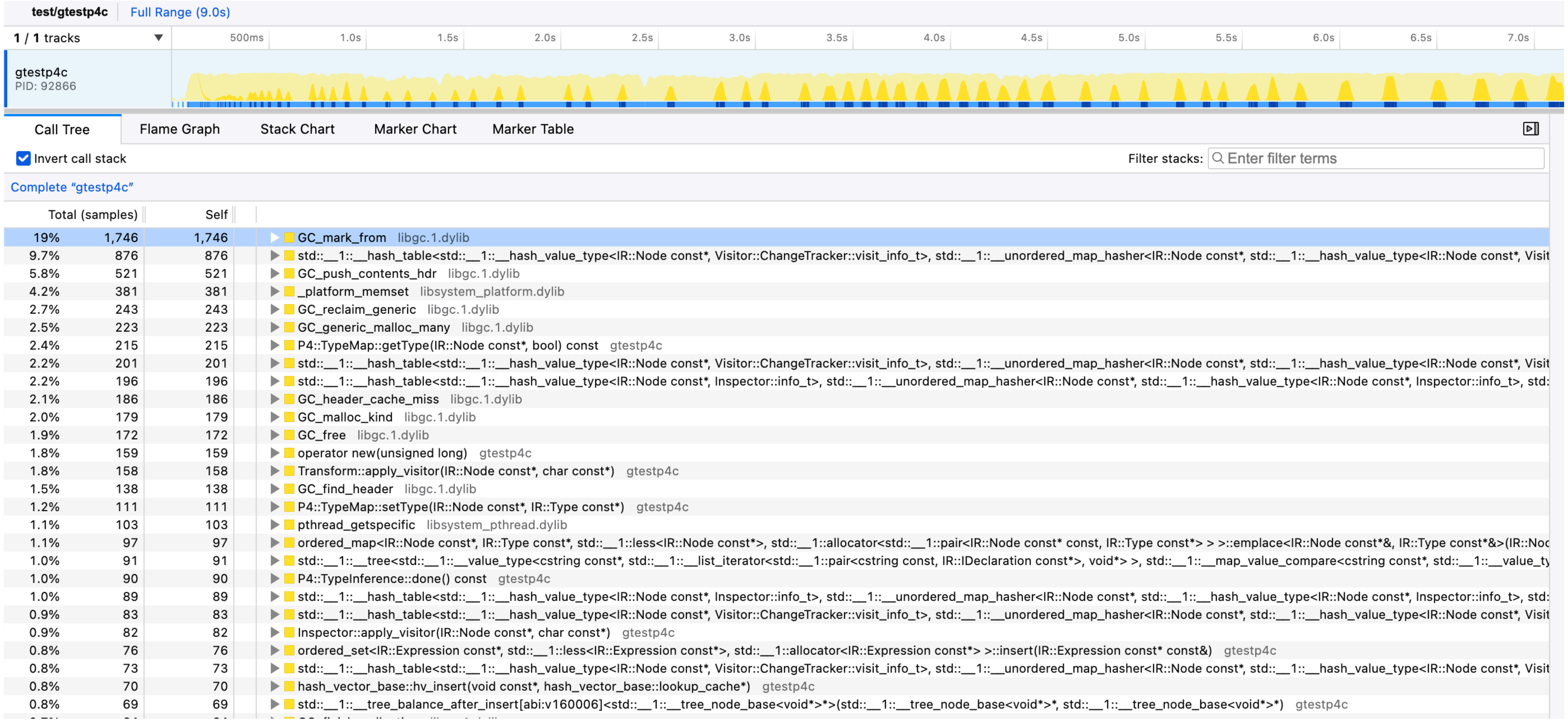
Total (samples)		Self	
100%	15,985	—	▼ start dyld
100%	15,983	—	▼ main gtestp4c
100%	15,932	—	▼ testing::UnitTest::Run() gtestp4c
100%	15,932	—	▼ testing::internal::HandleExceptionsInMethodIfSupported<testing::internal::Ui
100%	15,932	—	▼ testing::internal::UnitTestImpl::RunAllTests() gtestp4c
100%	15,932	—	▼ testing::TestSuite::Run() gtestp4c
100%	15,932	—	▼ testing::TestInfo::Run() gtestp4c
100%	15,932	—	▼ testing::Test::Run() gtestp4c
100%	15,932	—	▼ testing::internal::HandleExceptionsInMethodIfSupported<testing::T
100%	15,932	—	▼ Test::P4CParserUnroll_switch_20160512_Test::TestBody() gtes
100%	15,932	—	▼ Test::loadExample(char const*, ParserOptions::FrontendVersion
98%	15,597	—	▼ Test::rewriteParser(IR::P4Program const*, CompilerOptions&)
81%	12,941	—	▶ P4::FrontEnd::run(CompilerOptions const&, IR::P4Program
17%	2,645	—	▶ IR::P4Program::apply(Visitor&, Visitor_Context const*) cons
0.1%	10	—	▶ P4::TypeMap::~TypeMap() gtestp4c
0.0%	1	—	▶ P4::ReferenceMap::~ReferenceMap() gtestp4c
2.1%	335	—	▶ Test::load_model(char const*, CompilerOptions&) gtestp4c
0.3%	51	—	▶ P4CTestEnvironment::get() gtestp4c
0.0%	1	—	▶ dyld4::prepare(dyld4::APIs&, dyld3::MachOAnalyzer const*) dyld
0.0%	1	—	▶ dyld4::LibSystemHelpers::exit(int) const libdyld.dylib



Total (samples)		Self	
100%	9,002	—	▼ start dyld
100%	8,998	—	▼ main gtestp4c
99%	8,946	—	▼ testing::UnitTest::Run() gtestp4c
99%	8,946	—	▼ testing::internal::HandleExceptionsInMethodIfSupported<testing::internal::U
99%	8,946	—	▼ testing::internal::UnitTestImpl::RunAllTests() gtestp4c
99%	8,946	—	▼ testing::TestSuite::Run() gtestp4c
99%	8,946	—	▼ testing::TestInfo::Run() gtestp4c
99%	8,946	—	▼ testing::Test::Run() gtestp4c
99%	8,946	1	▼ testing::internal::HandleExceptionsInMethodIfSupported<testing::T
99%	8,945	—	▼ Test::P4CParserUnroll_switch_20160512_Test::TestBody() gtes
99%	8,945	—	▼ Test::loadExample(char const*, ParserOptions::FrontendVersior
96%	8,687	—	▼ Test::rewriteParser(IR::P4Program const*, CompilerOptions&)
82%	7,410	—	▶ P4::FrontEnd::run(CompilerOptions const&, IR::P4Program
14%	1,267	—	▶ IR::P4Program::apply(Visitor&, Visitor_Context const*) cons
0.1%	7	—	▶ P4::TypeMap::~TypeMap() gtestp4c
0.0%	1	—	▶ Test::MidEnd::MidEnd(CompilerOptions&, std::__1::basic_o
0.0%	1	—	▶ Test::getParser(IR::P4Program const*) gtestp4c
0.0%	1	—	▶ P4::ReferenceMap::~ReferenceMap() gtestp4c
2.9%	258	—	▶ Test::load_model(char const*, CompilerOptions&) gtestp4c
0.6%	51	—	▶ P4CTestEnvironment::get() gtestp4c

43% reduction of compile time!

Results



☑ Invert call stack

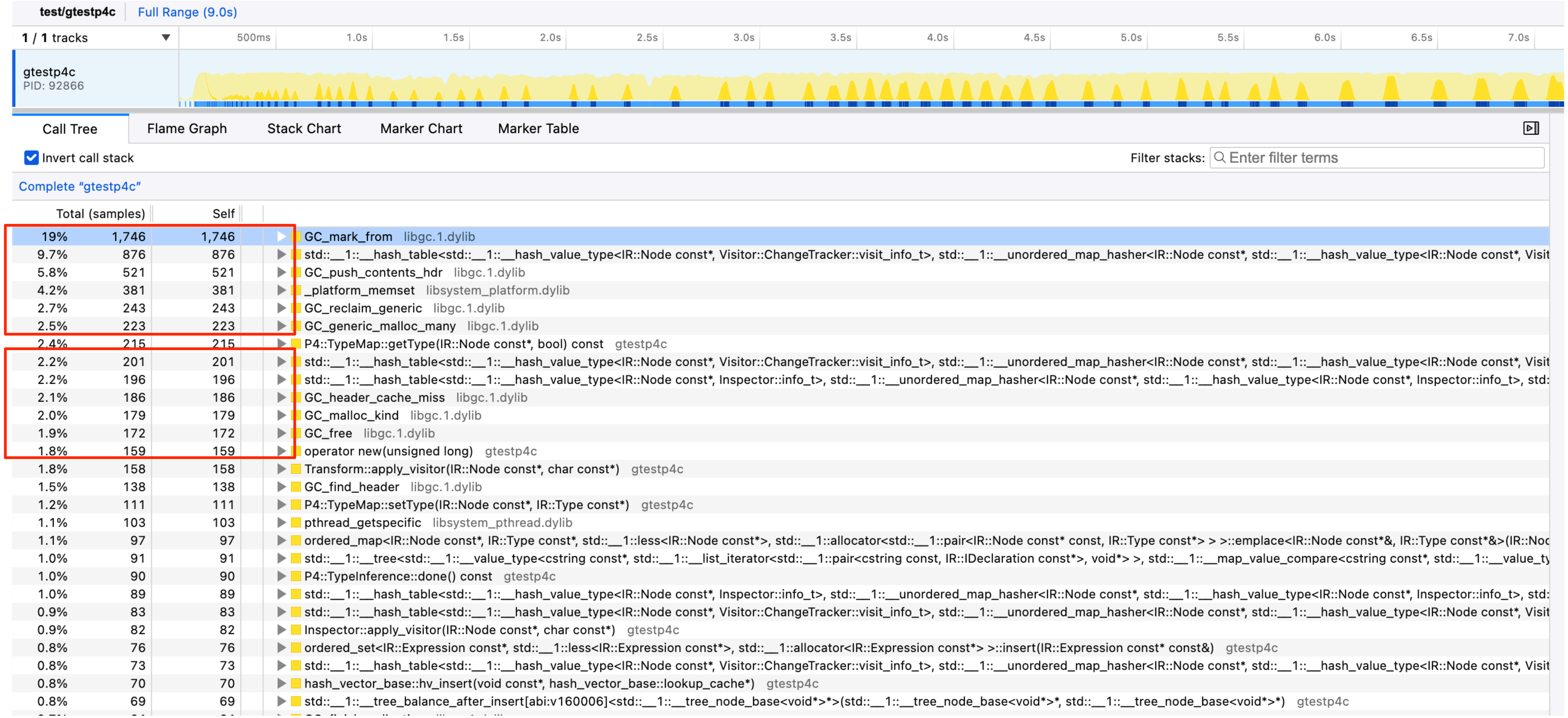
Filter stacks: Enter filter terms

Complete “gtestp4c”

Total (samples)		Self	
19%	1,746	1,746	▶ GC_mark_from libgc.1.dylib
9.7%	876	876	▶ std::__1::__hash_table<std::__1::__hash_value_type<IR::Node const*, Visitor::ChangeTracker::visit_info_t>, std::__1::__unordered_map_hasher<IR::Node const*, std::__1::__hash_value_type<IR::Node const*, Visit
5.8%	521	521	▶ GC_push_contents_hdr libgc.1.dylib
4.2%	381	381	▶ _platform_memset libsystem_platform.dylib
2.7%	243	243	▶ GC_reclaim_generic libgc.1.dylib
2.5%	223	223	▶ GC_generic_malloc_many libgc.1.dylib
2.4%	215	215	▶ P4::TypeMap::getType(IR::Node const*, bool) const gtestp4c
2.2%	201	201	▶ std::__1::__hash_table<std::__1::__hash_value_type<IR::Node const*, Visitor::ChangeTracker::visit_info_t>, std::__1::__unordered_map_hasher<IR::Node const*, std::__1::__hash_value_type<IR::Node const*, Visit
2.2%	196	196	▶ std::__1::__hash_table<std::__1::__hash_value_type<IR::Node const*, Inspector::info_t>, std::__1::__unordered_map_hasher<IR::Node const*, std::__1::__hash_value_type<IR::Node const*, Inspector::info_t>, std:
2.1%	186	186	▶ GC_header_cache_miss libgc.1.dylib
2.0%	179	179	▶ GC_malloc_kind libgc.1.dylib
1.9%	172	172	▶ GC_free libgc.1.dylib
1.8%	159	159	▶ operator new(unsigned long) gtestp4c
1.8%	158	158	▶ Transform::apply_visitor(IR::Node const*, char const*) gtestp4c
1.5%	138	138	▶ GC_find_header libgc.1.dylib
1.2%	111	111	▶ P4::TypeMap::setType(IR::Node const*, IR::Type const*) gtestp4c
1.1%	103	103	▶ pthread_getspecific libsystem_pthread.dylib
1.1%	97	97	▶ ordered_map<IR::Node const*, IR::Type const*, std::__1::less<IR::Node const*>, std::__1::allocator<std::__1::pair<IR::Node const* const, IR::Type const*> > >::emplace<IR::Node const*&, IR::Type const*&>(IR::Noc
1.0%	91	91	▶ std::__1::__tree<std::__1::__value_type<cstring const*, std::__1::__list_iterator<std::__1::pair<cstring const, IR::IDeclaration const*>, void*> >, std::__1::__map_value_compare<cstring const*, std::__1::__value_ty
1.0%	90	90	▶ P4::TypeInference::done() const gtestp4c
1.0%	89	89	▶ std::__1::__hash_table<std::__1::__hash_value_type<IR::Node const*, Inspector::info_t>, std::__1::__unordered_map_hasher<IR::Node const*, std::__1::__hash_value_type<IR::Node const*, Inspector::info_t>, std:
0.9%	83	83	▶ std::__1::__hash_table<std::__1::__hash_value_type<IR::Node const*, Visitor::ChangeTracker::visit_info_t>, std::__1::__unordered_map_hasher<IR::Node const*, std::__1::__hash_value_type<IR::Node const*, Visit
0.9%	82	82	▶ Inspector::apply_visitor(IR::Node const*, char const*) gtestp4c
0.8%	76	76	▶ ordered_set<IR::Expression const*, std::__1::less<IR::Expression const*>, std::__1::allocator<IR::Expression const*> >::insert(IR::Expression const* const&) gtestp4c
0.8%	73	73	▶ std::__1::__hash_table<std::__1::__hash_value_type<IR::Node const*, Visitor::ChangeTracker::visit_info_t>, std::__1::__unordered_map_hasher<IR::Node const*, std::__1::__hash_value_type<IR::Node const*, Visit
0.8%	70	70	▶ hash_vector_base::hv_insert(void const*, hash_vector_base::lookup_cache*) gtestp4c
0.8%	69	69	▶ std::__1::__tree_balance_after_insert[abi:v160006]<std::__1::__tree_node_base<void*>*>(std::__1::__tree_node_base<void*>*, std::__1::__tree_node_base<void*>*) gtestp4c

No traces of RTTI runtime calls (and no custom RTTI either)

Tale of malloc and 3 Visitors



☑ Invert call stack

Total (samples)		Self	
19%	1,746	1,746	GC_mark_from libgc.1.dylib
9.7%	876	876	std::__1::__hash_table<std::__1::__hash_value_type<IR::Node const*, Visitor::ChangeTracker::visit_info_t>, std::__1::__unordered_map_hasher<IR::Node const*, std::__1::__hash_value_type<IR::Node const*, Visit
5.8%	521	521	GC_push_contents_hdr libgc.1.dylib
4.2%	381	381	_platform_memset libsystem_platform.dylib
2.7%	243	243	GC_reclaim_generic libgc.1.dylib
2.5%	223	223	GC_generic_malloc_many libgc.1.dylib
2.4%	215	215	P4::TypeMap::getType(IR::Node const*, bool) const gtestp4c
2.2%	201	201	std::__1::__hash_table<std::__1::__hash_value_type<IR::Node const*, Visitor::ChangeTracker::visit_info_t>, std::__1::__unordered_map_hasher<IR::Node const*, std::__1::__hash_value_type<IR::Node const*, Visit
2.2%	196	196	std::__1::__hash_table<std::__1::__hash_value_type<IR::Node const*, Inspector::info_t>, std::__1::__unordered_map_hasher<IR::Node const*, std::__1::__hash_value_type<IR::Node const*, Inspector::info_t>, std:
2.1%	186	186	GC_header_cache_miss libgc.1.dylib
2.0%	179	179	GC_malloc_kind libgc.1.dylib
1.9%	172	172	GC_free libgc.1.dylib
1.8%	159	159	operator new(unsigned long) gtestp4c
1.8%	158	158	Transform::apply_visitor(IR::Node const*, char const*) gtestp4c
1.5%	138	138	GC_find_header libgc.1.dylib
1.2%	111	111	P4::TypeMap::setType(IR::Node const*, IR::Type const*) gtestp4c
1.1%	103	103	pthread_getspecific libsystem_pthread.dylib
1.1%	97	97	ordered_map<IR::Node const*, IR::Type const*, std::__1::less<IR::Node const*>, std::__1::allocator<std::__1::pair<IR::Node const* const, IR::Type const*> > >::emplace<IR::Node const*&, IR::Type const*&>(IR::Noc
1.0%	91	91	std::__1::__tree<std::__1::__value_type<cstring const*, std::__1::__list_iterator<std::__1::pair<cstring const, IR::IDeclaration const*>, void*> >, std::__1::__map_value_compare<cstring const*, std::__1::__value_ty
1.0%	90	90	P4::TypeInference::done() const gtestp4c
1.0%	89	89	std::__1::__hash_table<std::__1::__hash_value_type<IR::Node const*, Inspector::info_t>, std::__1::__unordered_map_hasher<IR::Node const*, std::__1::__hash_value_type<IR::Node const*, Inspector::info_t>, std:
0.9%	83	83	std::__1::__hash_table<std::__1::__hash_value_type<IR::Node const*, Visitor::ChangeTracker::visit_info_t>, std::__1::__unordered_map_hasher<IR::Node const*, std::__1::__hash_value_type<IR::Node const*, Visit
0.9%	82	82	Inspector::apply_visitor(IR::Node const*, char const*) gtestp4c
0.8%	76	76	ordered_set<IR::Expression const*, std::__1::less<IR::Expression const*>, std::__1::allocator<IR::Expression const*> >::insert(IR::Expression const* const&) gtestp4c
0.8%	73	73	std::__1::__hash_table<std::__1::__hash_value_type<IR::Node const*, Visitor::ChangeTracker::visit_info_t>, std::__1::__unordered_map_hasher<IR::Node const*, std::__1::__hash_value_type<IR::Node const*, Visit
0.8%	70	70	hash_vector_base::hv_insert(void const*, hash_vector_base::lookup_cache*) gtestp4c
0.8%	69	69	std::__1::__tree_balance_after_insert[abi:v160006]<std::__1::__tree_node_base<void*>*>(std::__1::__tree_node_base<void*>*, std::__1::__tree_node_base<void*>*) gtestp4c

More than 15% is Visitor boilerplate

Visitor Boilerplate

- Each visitor maintains internal state in a hash table (aka `visited`)
  - `IR::Node*` => some state (just 2 bools for `Inspector` and `ChangeTracker` for `Modifier / Transform`)
- Total number of `init_apply()` calls here:
  - 117k `Inspector`'s, 13 `Modifier`'s and 86k `Transform`'s
- `std::unordered_map` is not the fastest / best implementation out there
- Huge malloc traffic to create / destroy these hash tables and their contents
  - For each `init_apply()` call: new hash map + corresponding malloc traffic
- GC is expensive:
  - Needs to `memset` allocated / freed memory
  - Slow implementation as compared to other memory allocators
  - Has significant overhead: ~25% runtime improvements with GC off

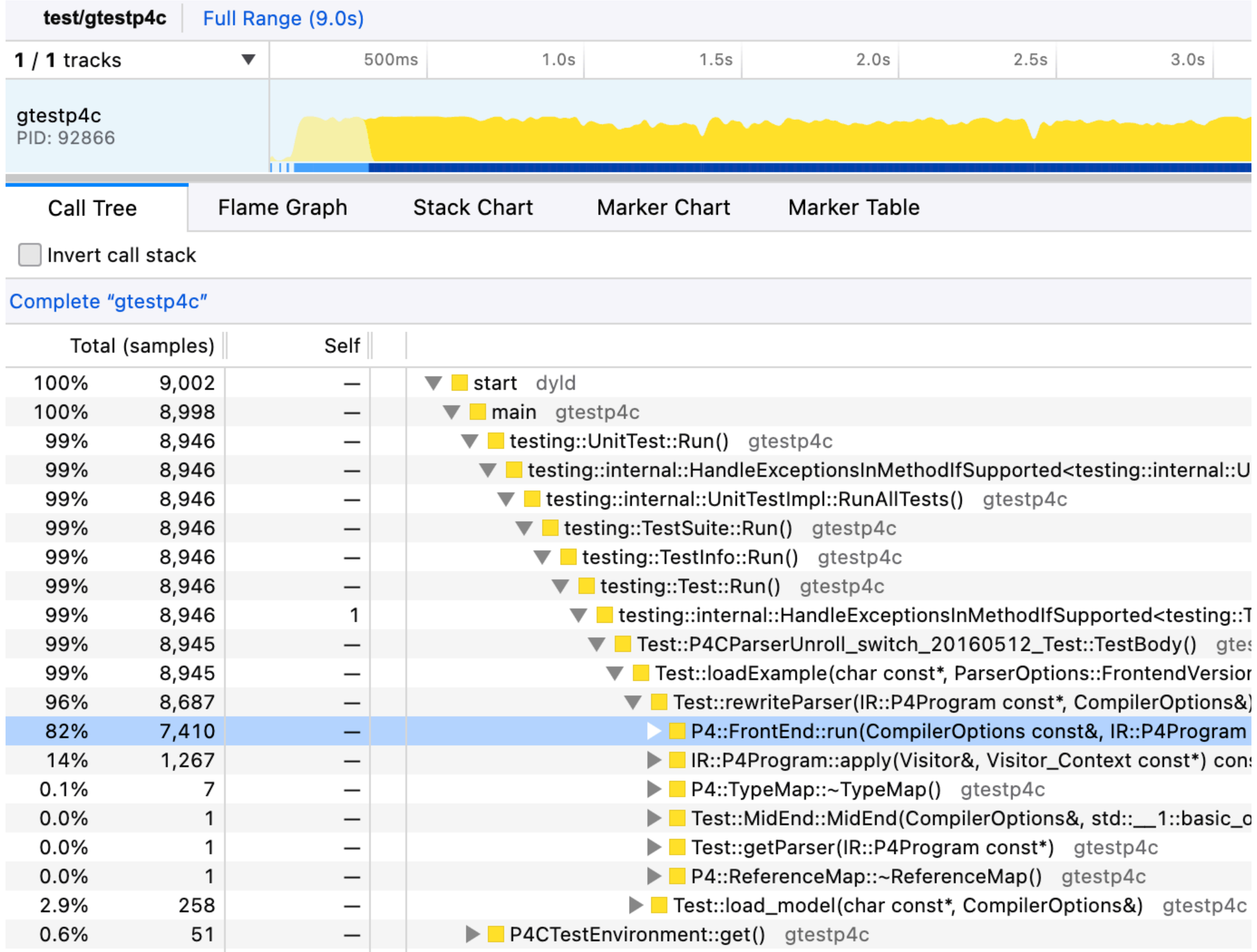
Visitor Boilerplate: Caveats & Observations

- Pointers to map values do escape (`visitCurrentOnce`): code relies on their stability during insertion
  - Prevents drop-in use of not standard-compliant modern maps
- Few places rely on iterator stability during insertion
  - Need to revise the code in order not to do this
- Extra unnecessary lookups (e.g. `count()` + `at()` for the same key)
- In many cases these maps are small (contain a few values),
  - Although some might be pretty big
  - Try to preallocate some slots during map construction to reduce malloc traffic

Visitor Boilerplate: Solution

- Rewrite code so pointers to map values do not escape
  - Store pointer to current node instead
- Use abseil swiss map (`flat_hash_map`) implementation
  - Modern header-only drop-in replacement of `std::unordered_map` with lots of useful tweaks and decent performance.
  - Seems an excellent choice for the purpose.
  - Already available due to protobuf abseil dependency.
- Rewrite the code not to do unnecessary double lookups
- Preallocate 16 map slots by default (single memory allocation for small map)

Results



test/gtestp4c Full Range (9.0s)

1 / 1 tracks ▼ 500ms 1.0s 1.5s 2.0s 2.5s 3.0s

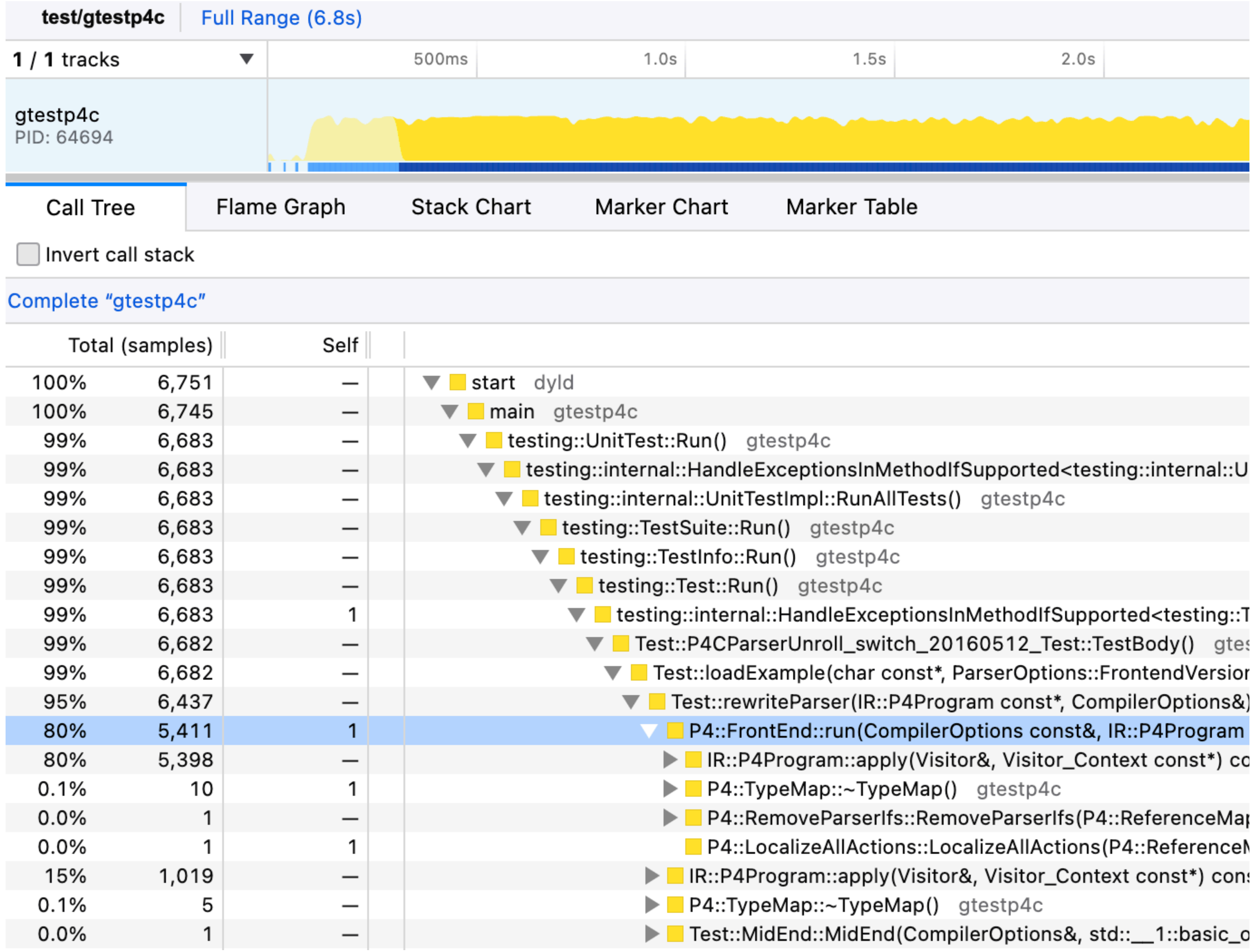
gtestp4c
PID: 92866

Call Tree | Flame Graph | Stack Chart | Marker Chart | Marker Table

□ Invert call stack

Complete "gtestp4c"

Total (samples)		Self	
100%	9,002	—	▼ start dyld
100%	8,998	—	▼ main gtestp4c
99%	8,946	—	▼ testing::UnitTest::Run() gtestp4c
99%	8,946	—	▼ testing::internal::HandleExceptionsInMethodIfSupported<testing::internal::U
99%	8,946	—	▼ testing::internal::UnitTestImpl::RunAllTests() gtestp4c
99%	8,946	—	▼ testing::TestSuite::Run() gtestp4c
99%	8,946	—	▼ testing::TestInfo::Run() gtestp4c
99%	8,946	—	▼ testing::Test::Run() gtestp4c
99%	8,946	1	▼ testing::internal::HandleExceptionsInMethodIfSupported<testing::T
99%	8,945	—	▼ Test::P4CParserUnroll_switch_20160512_Test::TestBody() gtes
99%	8,945	—	▼ Test::loadExample(char const*, ParserOptions::FrontendVersior
96%	8,687	—	▼ Test::rewriteParser(IR::P4Program const*, CompilerOptions&)
82%	7,410	—	▶ P4::FrontEnd::run(CompilerOptions const&, IR::P4Program
14%	1,267	—	▶ IR::P4Program::apply(Visitor&, Visitor_Context const*) con:
0.1%	7	—	▶ P4::TypeMap::~TypeMap() gtestp4c
0.0%	1	—	▶ Test::MidEnd::MidEnd(CompilerOptions&, std::__1::basic_o
0.0%	1	—	▶ Test::getParser(IR::P4Program const*) gtestp4c
0.0%	1	—	▶ P4::ReferenceMap::~ReferenceMap() gtestp4c
2.9%	258	—	▶ Test::load_model(char const*, CompilerOptions&) gtestp4c
0.6%	51	—	▶ P4CTestEnvironment::get() gtestp4c



test/gtestp4c Full Range (6.8s)

1 / 1 tracks ▼ 500ms 1.0s 1.5s 2.0s

gtestp4c
PID: 64694

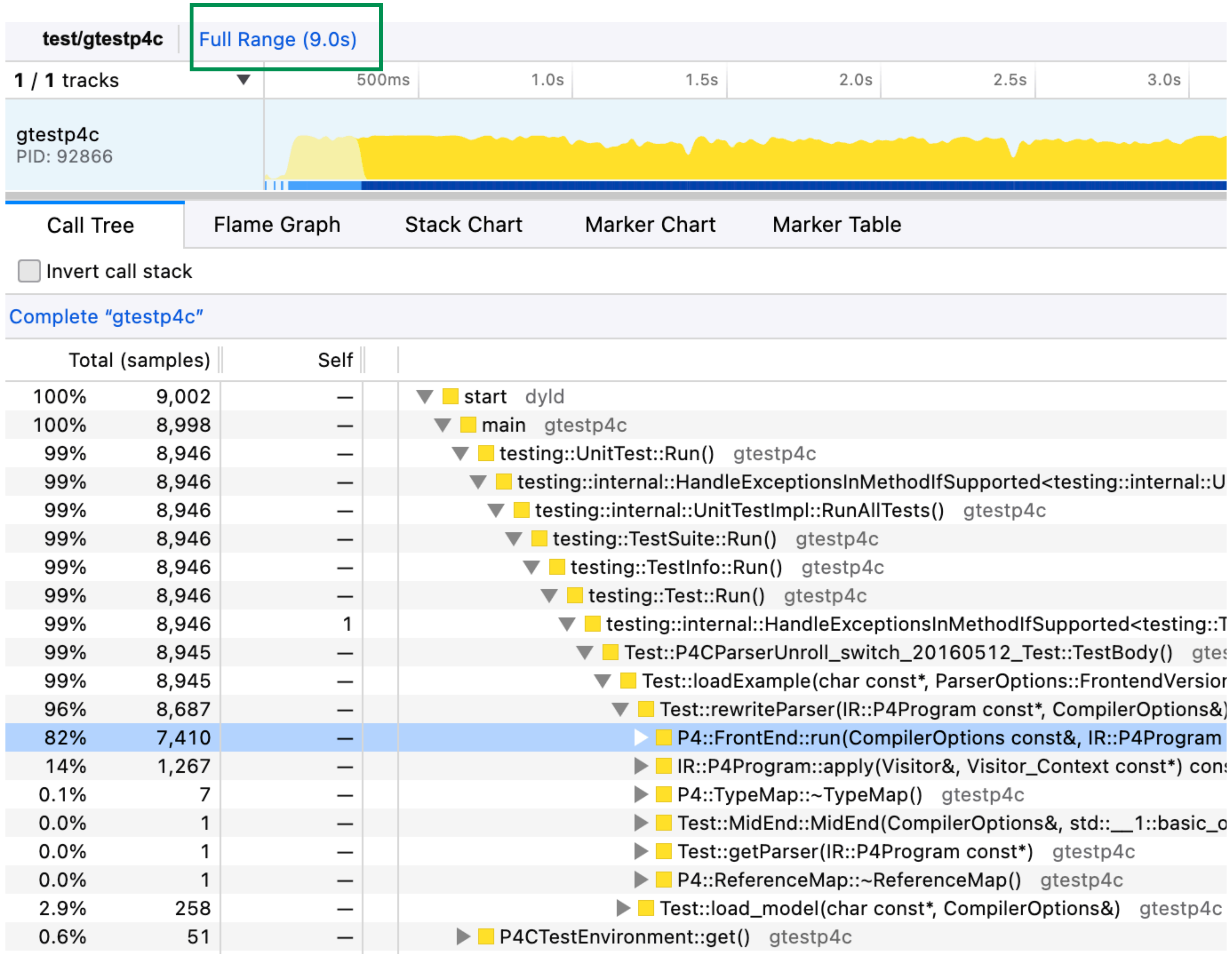
Call Tree | Flame Graph | Stack Chart | Marker Chart | Marker Table

□ Invert call stack

Complete "gtestp4c"

Total (samples)		Self	
100%	6,751	—	▼ start dyld
100%	6,745	—	▼ main gtestp4c
99%	6,683	—	▼ testing::UnitTest::Run() gtestp4c
99%	6,683	—	▼ testing::internal::HandleExceptionsInMethodIfSupported<testing::internal::U
99%	6,683	—	▼ testing::internal::UnitTestImpl::RunAllTests() gtestp4c
99%	6,683	—	▼ testing::TestSuite::Run() gtestp4c
99%	6,683	—	▼ testing::TestInfo::Run() gtestp4c
99%	6,683	—	▼ testing::Test::Run() gtestp4c
99%	6,683	1	▼ testing::internal::HandleExceptionsInMethodIfSupported<testing::T
99%	6,682	—	▼ Test::P4CParserUnroll_switch_20160512_Test::TestBody() gtes
99%	6,682	—	▼ Test::loadExample(char const*, ParserOptions::FrontendVersior
95%	6,437	—	▼ Test::rewriteParser(IR::P4Program const*, CompilerOptions&)
80%	5,411	1	▼ P4::FrontEnd::run(CompilerOptions const&, IR::P4Program
80%	5,398	—	▶ IR::P4Program::apply(Visitor&, Visitor_Context const*) co
0.1%	10	1	▶ P4::TypeMap::~TypeMap() gtestp4c
0.0%	1	—	▶ P4::RemoveParserIfs::RemoveParserIfs(P4::ReferenceMa|
0.0%	1	1	P4::LocalizeAllActions::LocalizeAllActions(P4::Reference\
15%	1,019	—	▶ IR::P4Program::apply(Visitor&, Visitor_Context const*) con:
0.1%	5	—	▶ P4::TypeMap::~TypeMap() gtestp4c
0.0%	1	—	▶ Test::MidEnd::MidEnd(CompilerOptions&, std::__1::basic_o

Results



test/gtestp4c | Full Range (9.0s)

1 / 1 tracks ▼ 500ms 1.0s 1.5s 2.0s 2.5s 3.0s

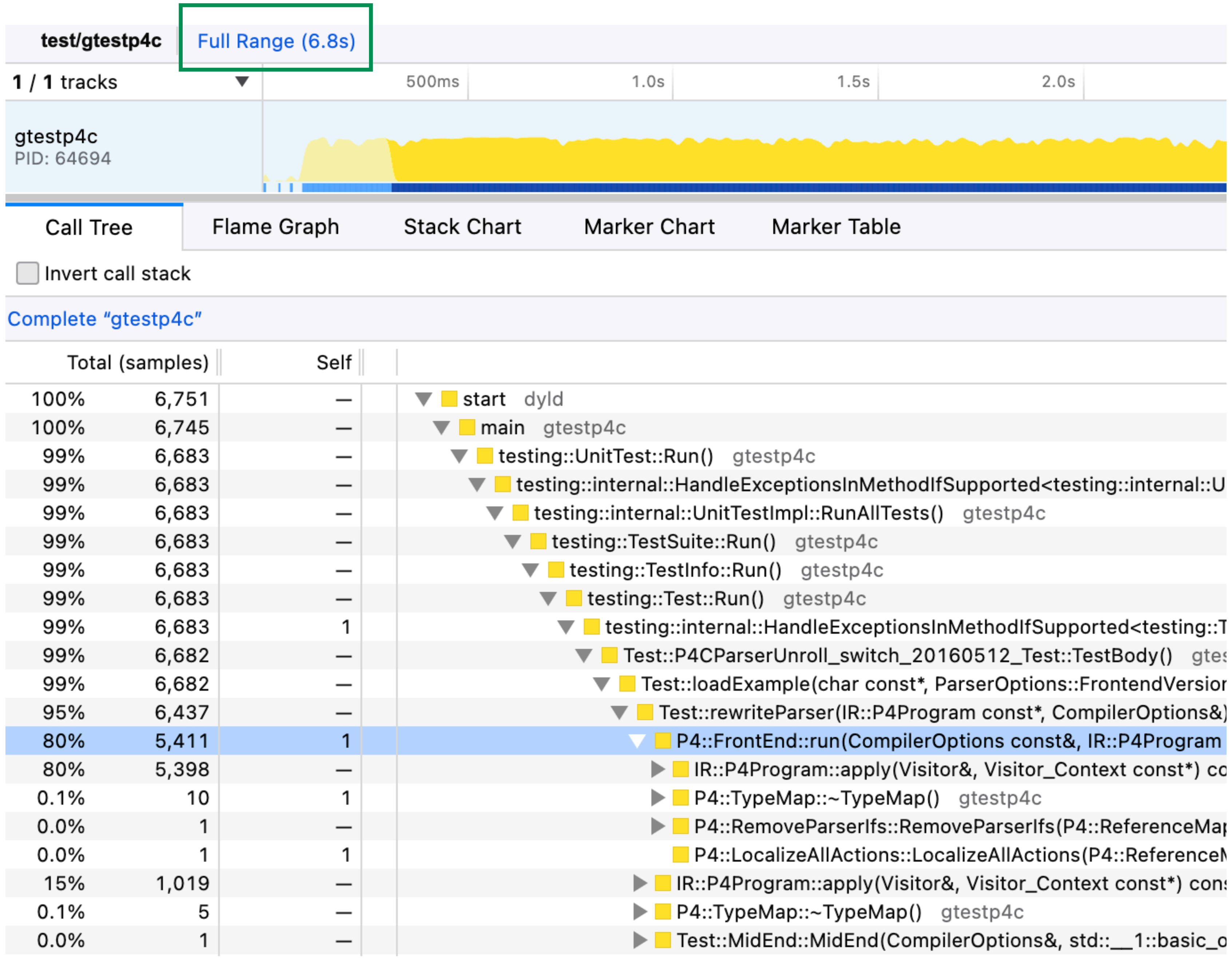
gtestp4c
PID: 92866

Call Tree | Flame Graph | Stack Chart | Marker Chart | Marker Table

□ Invert call stack

Complete "gtestp4c"

Total (samples)		Self	
100%	9,002	—	▼ start dyld
100%	8,998	—	▼ main gtestp4c
99%	8,946	—	▼ testing::UnitTest::Run() gtestp4c
99%	8,946	—	▼ testing::internal::HandleExceptionsInMethodIfSupported<testing::internal::U
99%	8,946	—	▼ testing::internal::UnitTestImpl::RunAllTests() gtestp4c
99%	8,946	—	▼ testing::TestSuite::Run() gtestp4c
99%	8,946	—	▼ testing::TestInfo::Run() gtestp4c
99%	8,946	—	▼ testing::Test::Run() gtestp4c
99%	8,946	1	▼ testing::internal::HandleExceptionsInMethodIfSupported<testing::T
99%	8,945	—	▼ Test::P4CParserUnroll_switch_20160512_Test::TestBody() gtes
99%	8,945	—	▼ Test::loadExample(char const*, ParserOptions::FrontendVersior
96%	8,687	—	▼ Test::rewriteParser(IR::P4Program const*, CompilerOptions&)
82%	7,410	—	▶ P4::FrontEnd::run(CompilerOptions const&, IR::P4Program
14%	1,267	—	▶ IR::P4Program::apply(Visitor&, Visitor_Context const*) con:
0.1%	7	—	▶ P4::TypeMap::~TypeMap() gtestp4c
0.0%	1	—	▶ Test::MidEnd::MidEnd(CompilerOptions&, std::__1::basic_o
0.0%	1	—	▶ Test::getParser(IR::P4Program const*) gtestp4c
0.0%	1	—	▶ P4::ReferenceMap::~ReferenceMap() gtestp4c
2.9%	258	—	▶ Test::load_model(char const*, CompilerOptions&) gtestp4c
0.6%	51	—	▶ P4CTestEnvironment::get() gtestp4c



test/gtestp4c | Full Range (6.8s)

1 / 1 tracks ▼ 500ms 1.0s 1.5s 2.0s

gtestp4c
PID: 64694

Call Tree | Flame Graph | Stack Chart | Marker Chart | Marker Table

□ Invert call stack

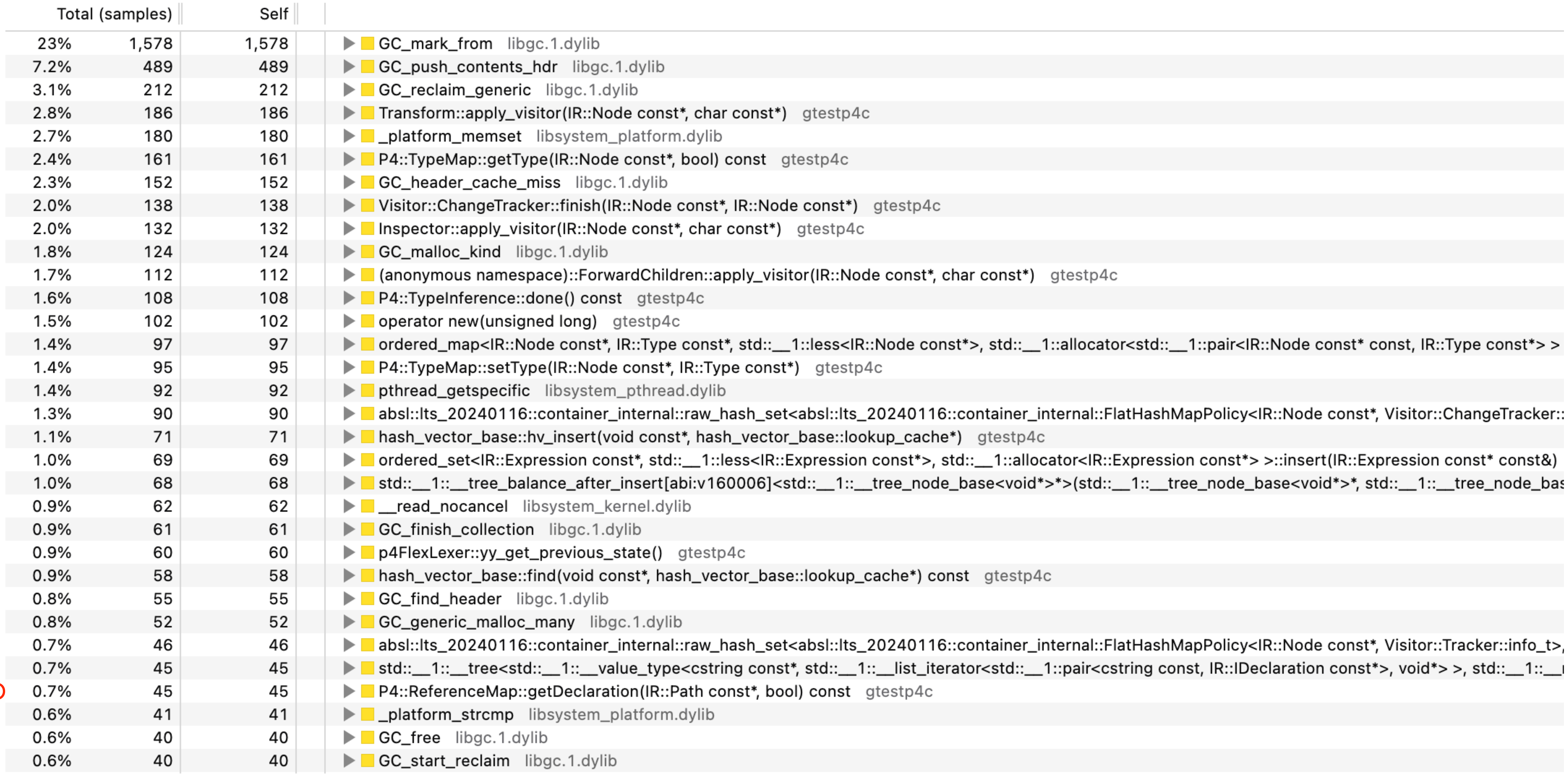
Complete "gtestp4c"

Total (samples)		Self	
100%	6,751	—	▼ start dyld
100%	6,745	—	▼ main gtestp4c
99%	6,683	—	▼ testing::UnitTest::Run() gtestp4c
99%	6,683	—	▼ testing::internal::HandleExceptionsInMethodIfSupported<testing::internal::U
99%	6,683	—	▼ testing::internal::UnitTestImpl::RunAllTests() gtestp4c
99%	6,683	—	▼ testing::TestSuite::Run() gtestp4c
99%	6,683	—	▼ testing::TestInfo::Run() gtestp4c
99%	6,683	—	▼ testing::Test::Run() gtestp4c
99%	6,683	1	▼ testing::internal::HandleExceptionsInMethodIfSupported<testing::T
99%	6,682	—	▼ Test::P4CParserUnroll_switch_20160512_Test::TestBody() gtes
99%	6,682	—	▼ Test::loadExample(char const*, ParserOptions::FrontendVersior
95%	6,437	—	▼ Test::rewriteParser(IR::P4Program const*, CompilerOptions&)
80%	5,411	1	▼ P4::FrontEnd::run(CompilerOptions const&, IR::P4Program
80%	5,398	—	▶ IR::P4Program::apply(Visitor&, Visitor_Context const*) co
0.1%	10	1	▶ P4::TypeMap::~TypeMap() gtestp4c
0.0%	1	—	▶ P4::RemoveParserIfs::RemoveParserIfs(P4::ReferenceMap
0.0%	1	1	P4::LocalizeAllActions::LocalizeAllActions(P4::ReferenceM
15%	1,019	—	▶ IR::P4Program::apply(Visitor&, Visitor_Context const*) con:
0.1%	5	—	▶ P4::TypeMap::~TypeMap() gtestp4c
0.0%	1	—	▶ Test::MidEnd::MidEnd(CompilerOptions&, std::__1::basic_o

25% reduction of compile time!

ACCESS SOFTEK, INC

Further analysis



	Total (samples)		Self	
	23%	1,578	1,578	▶ 🟨 GC_mark_from libgc.1.dylib
	7.2%	489	489	▶ 🟨 GC_push_contents_hdr libgc.1.dylib
	3.1%	212	212	▶ 🟨 GC_reclaim_generic libgc.1.dylib
	2.8%	186	186	▶ 🟨 Transform::apply_visitor(IR::Node const*, char const*) gtestp4c
	2.7%	180	180	▶ 🟨 _platform_memset libsystem_platform.dylib
TypeMap	2.4%	161	161	▶ 🟨 P4::TypeMap::getType(IR::Node const*, bool) const gtestp4c
	2.3%	152	152	▶ 🟨 GC_header_cache_miss libgc.1.dylib
	2.0%	138	138	▶ 🟨 Visitor::ChangeTracker::finish(IR::Node const*, IR::Node const*) gtestp4c
	2.0%	132	132	▶ 🟨 Inspector::apply_visitor(IR::Node const*, char const*) gtestp4c
	1.8%	124	124	▶ 🟨 GC_malloc_kind libgc.1.dylib
	1.7%	112	112	▶ 🟨 (anonymous namespace)::ForwardChildren::apply_visitor(IR::Node const*, char const*) gtestp4c
	1.6%	108	108	▶ 🟨 P4::TypeInference::done() const gtestp4c
	1.5%	102	102	▶ 🟨 operator new(unsigned long) gtestp4c
TypeMap	1.4%	97	97	▶ 🟨 ordered_map<IR::Node const*, IR::Type const*, std::__1::less<IR::Node const*>, std::__1::allocator<std::__1::pair<IR::Node const* const, IR::Type const*> >
TypeMap	1.4%	95	95	▶ 🟨 P4::TypeMap::setType(IR::Node const*, IR::Type const*) gtestp4c
	1.4%	92	92	▶ 🟨 pthread_getspecific libsystem_pthread.dylib
	1.3%	90	90	▶ 🟨 absl::lts_20240116::container_internal::raw_hash_set<absl::lts_20240116::container_internal::FlatHashMapPolicy<IR::Node const*, Visitor::ChangeTracker::
	1.1%	71	71	▶ 🟨 hash_vector_base::hv_insert(void const*, hash_vector_base::lookup_cache*) gtestp4c
TypeMap	1.0%	69	69	▶ 🟨 ordered_set<IR::Expression const*, std::__1::less<IR::Expression const*>, std::__1::allocator<IR::Expression const*> >::insert(IR::Expression const* const&)
TypeMap	1.0%	68	68	▶ 🟨 std::__1::__tree_balance_after_insert[abi:v160006]<std::__1::__tree_node_base<void*>*>(std::__1::__tree_node_base<void*>*, std::__1::__tree_node_bas
	0.9%	62	62	▶ 🟨 __read_nocancel libsystem_kernel.dylib
	0.9%	61	61	▶ 🟨 GC_finish_collection libgc.1.dylib
	0.9%	60	60	▶ 🟨 p4FlexLexer::yy_get_previous_state() gtestp4c
	0.9%	58	58	▶ 🟨 hash_vector_base::find(void const*, hash_vector_base::lookup_cache*) const gtestp4c
	0.8%	55	55	▶ 🟨 GC_find_header libgc.1.dylib
	0.8%	52	52	▶ 🟨 GC_generic_malloc_many libgc.1.dylib
	0.7%	46	46	▶ 🟨 absl::lts_20240116::container_internal::raw_hash_set<absl::lts_20240116::container_internal::FlatHashMapPolicy<IR::Node const*, Visitor::Tracker::info_t>,
TypeMap	0.7%	45	45	▶ 🟨 std::__1::__tree<std::__1::__value_type<cstring const*, std::__1::__list_iterator<std::__1::pair<cstring const, IR::IDeclaration const*>, void*> >, std::__1::__
ReferenceMap	0.7%	45	45	▶ 🟨 P4::ReferenceMap::getDeclaration(IR::Path const*, bool) const gtestp4c
	0.6%	41	41	▶ 🟨 _platform_strcmp libsystem_platform.dylib
	0.6%	40	40	▶ 🟨 GC_free libgc.1.dylib
	0.6%	40	40	▶ 🟨 GC_start_reclaim libgc.1.dylib

Expensive IR modifications

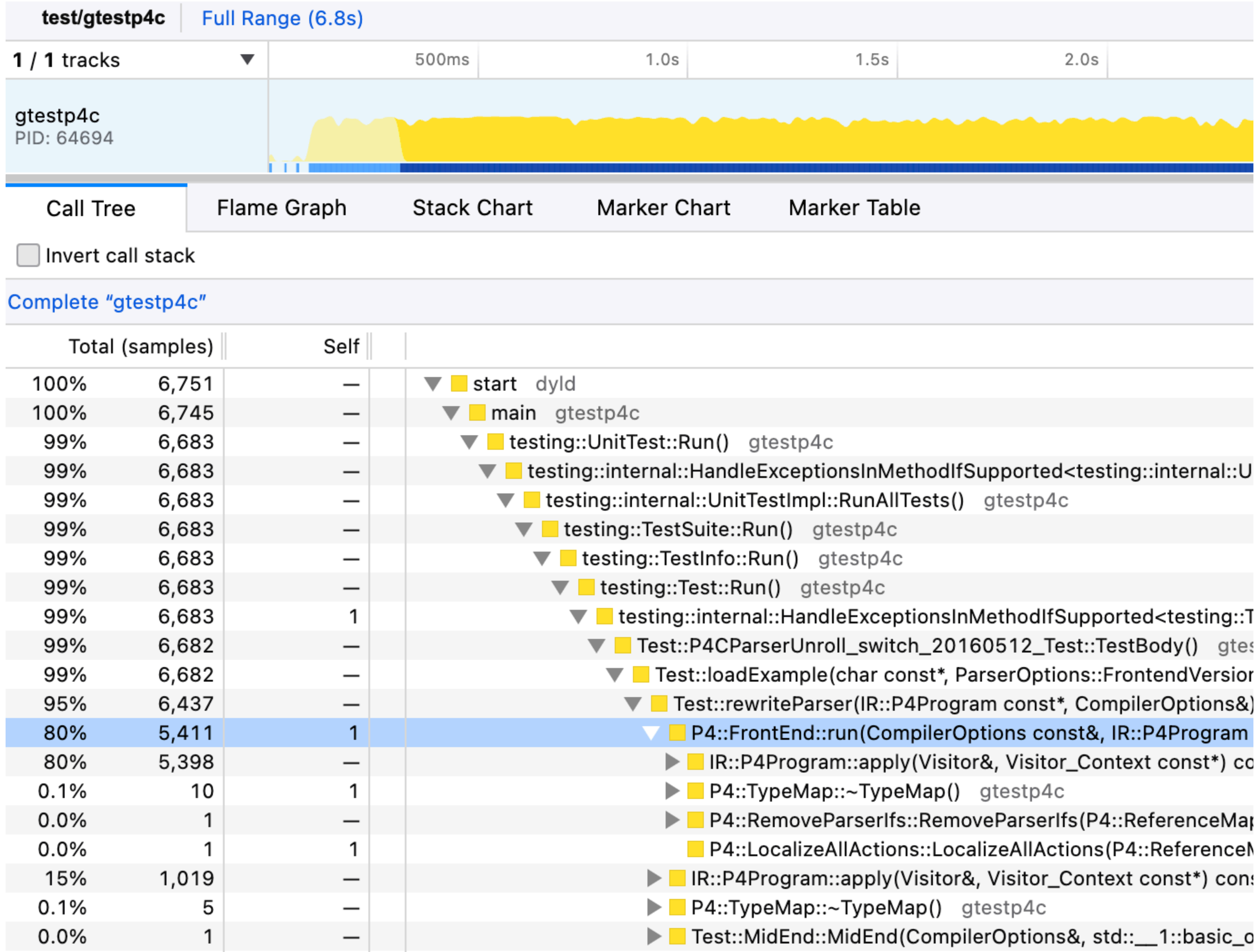
- Both `ReferenceMap` and `TypeMap` are recalculated from scratch after every (!) IR modification
  - Ignore this for a moment and take a look under the hood: bunch of `ordered_map`’s
- `ordered_map` is routinely used in P4C codebase
  - even when there is no iteration done at all
- `ordered_map` is terribly expensive:
  - It’s essentially `std::map<Key*, std::list_iterator> + std::list<std::pair<Key, Value>>`
  - Huge memory overhead (at least 8 pointers per entry!)
  - Slow lookup time
  - Huge malloc traffic
- There are some double lookups performed as well

As no iteration is done, let’s simply switch to abseil maps

Malloc traffic & GC

- GC is overly conservative
  - Needs to scan whole memory
  - Cannot use compiler annotations for pointer locations like in managed languages
  - Needs to `memset(0, &data, sizeof(data))` on allocation / deallocation
- GC is expensive: at least 25% of runtime overhead
- GC is unpredictable:
  - Leads to memory usage spikes
  - Leads to 20-30% of execution time differences on small code changes / allocation differences
- Poor coding practices: lots of code simply leak objects with clear runtime for no reason
- `PassManager` owns passes:
  - Extends the lifetime of pass internal state (even if pass is finished)
  - Could result in OOM due to large peak memory consumption

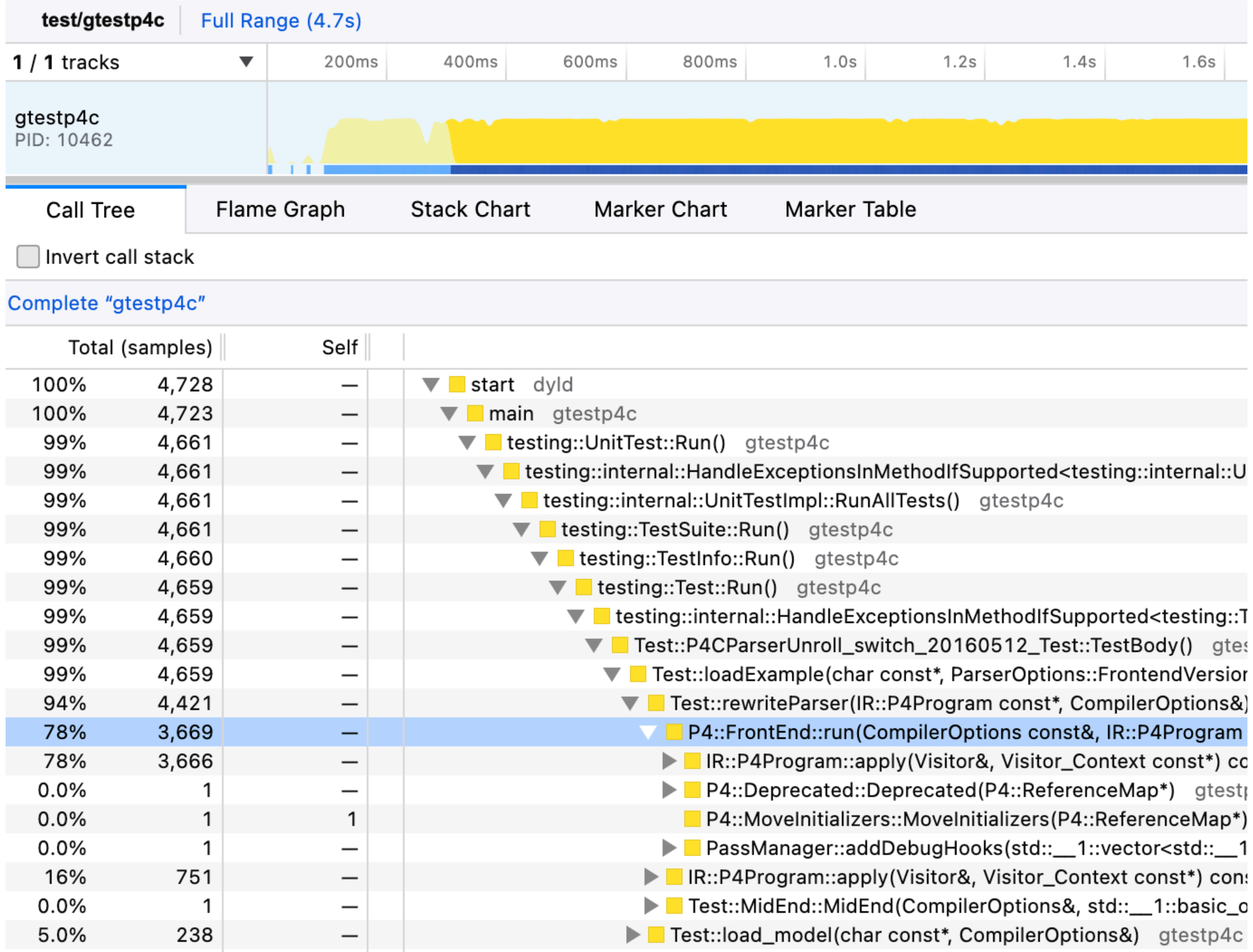
Results (maps + use-def malloc traffic)



☐ Invert call stack

Complete “gtestp4c”

Total (samples)		Self	
100%	6,751	—	▼ start dyld
100%	6,745	—	▼ main gtestp4c
99%	6,683	—	▼ testing::UnitTest::Run() gtestp4c
99%	6,683	—	▼ testing::internal::HandleExceptionsInMethodIfSupported<testing::internal::U
99%	6,683	—	▼ testing::internal::UnitTestImpl::RunAllTests() gtestp4c
99%	6,683	—	▼ testing::TestSuite::Run() gtestp4c
99%	6,683	—	▼ testing::TestInfo::Run() gtestp4c
99%	6,683	—	▼ testing::Test::Run() gtestp4c
99%	6,683	1	▼ testing::internal::HandleExceptionsInMethodIfSupported<testing::T
99%	6,682	—	▼ Test::P4CParserUnroll_switch_20160512_Test::TestBody() gtes
99%	6,682	—	▼ Test::loadExample(char const*, ParserOptions::FrontendVersior
95%	6,437	—	▼ Test::rewriteParser(IR::P4Program const*, CompilerOptions&)
80%	5,411	1	▼ P4::FrontEnd::run(CompilerOptions const&, IR::P4Program
80%	5,398	—	▶ IR::P4Program::apply(Visitor&, Visitor_Context const*) co
0.1%	10	1	▶ P4::TypeMap::~TypeMap() gtestp4c
0.0%	1	—	▶ P4::RemoveParserIfs::RemoveParserIfs(P4::ReferenceMa
0.0%	1	1	P4::LocalizeAllActions::LocalizeAllActions(P4::Reference
15%	1,019	—	▶ IR::P4Program::apply(Visitor&, Visitor_Context const*) cons
0.1%	5	—	▶ P4::TypeMap::~TypeMap() gtestp4c
0.0%	1	—	▶ Test::MidEnd::MidEnd(CompilerOptions&, std::__1::basic_o



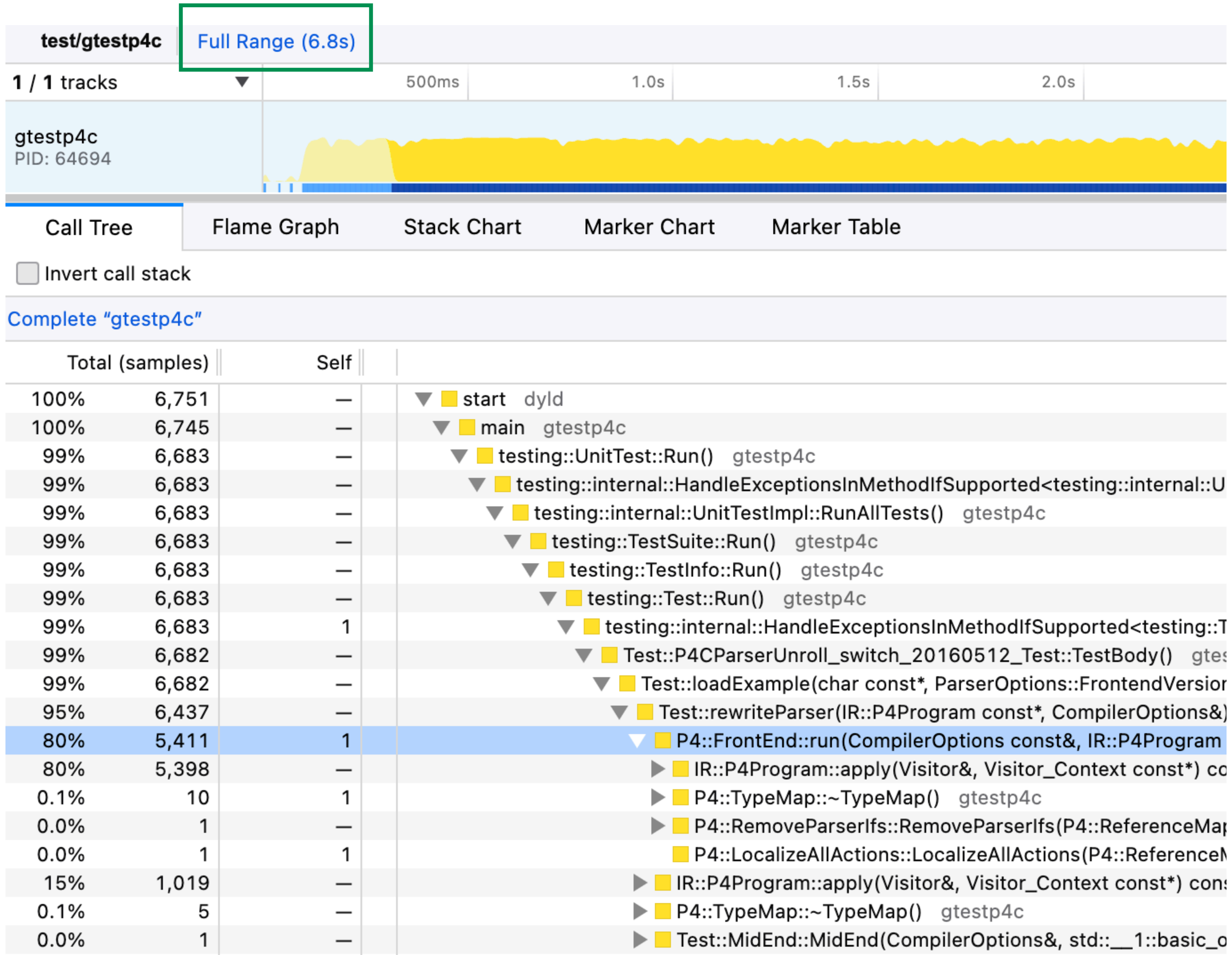
☐ Invert call stack

Complete “gtestp4c”

Total (samples)		Self	
100%	4,728	—	▼ start dyld
100%	4,723	—	▼ main gtestp4c
99%	4,661	—	▼ testing::UnitTest::Run() gtestp4c
99%	4,661	—	▼ testing::internal::HandleExceptionsInMethodIfSupported<testing::internal::U
99%	4,661	—	▼ testing::internal::UnitTestImpl::RunAllTests() gtestp4c
99%	4,661	—	▼ testing::TestSuite::Run() gtestp4c
99%	4,660	—	▼ testing::TestInfo::Run() gtestp4c
99%	4,659	—	▼ testing::Test::Run() gtestp4c
99%	4,659	—	▼ testing::internal::HandleExceptionsInMethodIfSupported<testing::T
99%	4,659	—	▼ Test::P4CParserUnroll_switch_20160512_Test::TestBody() gte
99%	4,659	—	▼ Test::loadExample(char const*, ParserOptions::FrontendVersior
94%	4,421	—	▼ Test::rewriteParser(IR::P4Program const*, CompilerOptions&)
78%	3,669	—	▼ P4::FrontEnd::run(CompilerOptions const&, IR::P4Program
78%	3,666	—	▶ IR::P4Program::apply(Visitor&, Visitor_Context const*) co
0.0%	1	—	▶ P4::Deprecated::Deprecated(P4::ReferenceMap*) gtest
0.0%	1	1	P4::MoveInitializers::MoveInitializers(P4::ReferenceMap*)
0.0%	1	—	▶ PassManager::addDebugHooks(std::__1::vector<std::__1
16%	751	—	▶ IR::P4Program::apply(Visitor&, Visitor_Context const*) cons
0.0%	1	—	▶ Test::MidEnd::MidEnd(CompilerOptions&, std::__1::basic_o
5.0%	238	—	▶ Test::load_model(char const*, CompilerOptions&) gtestp4c

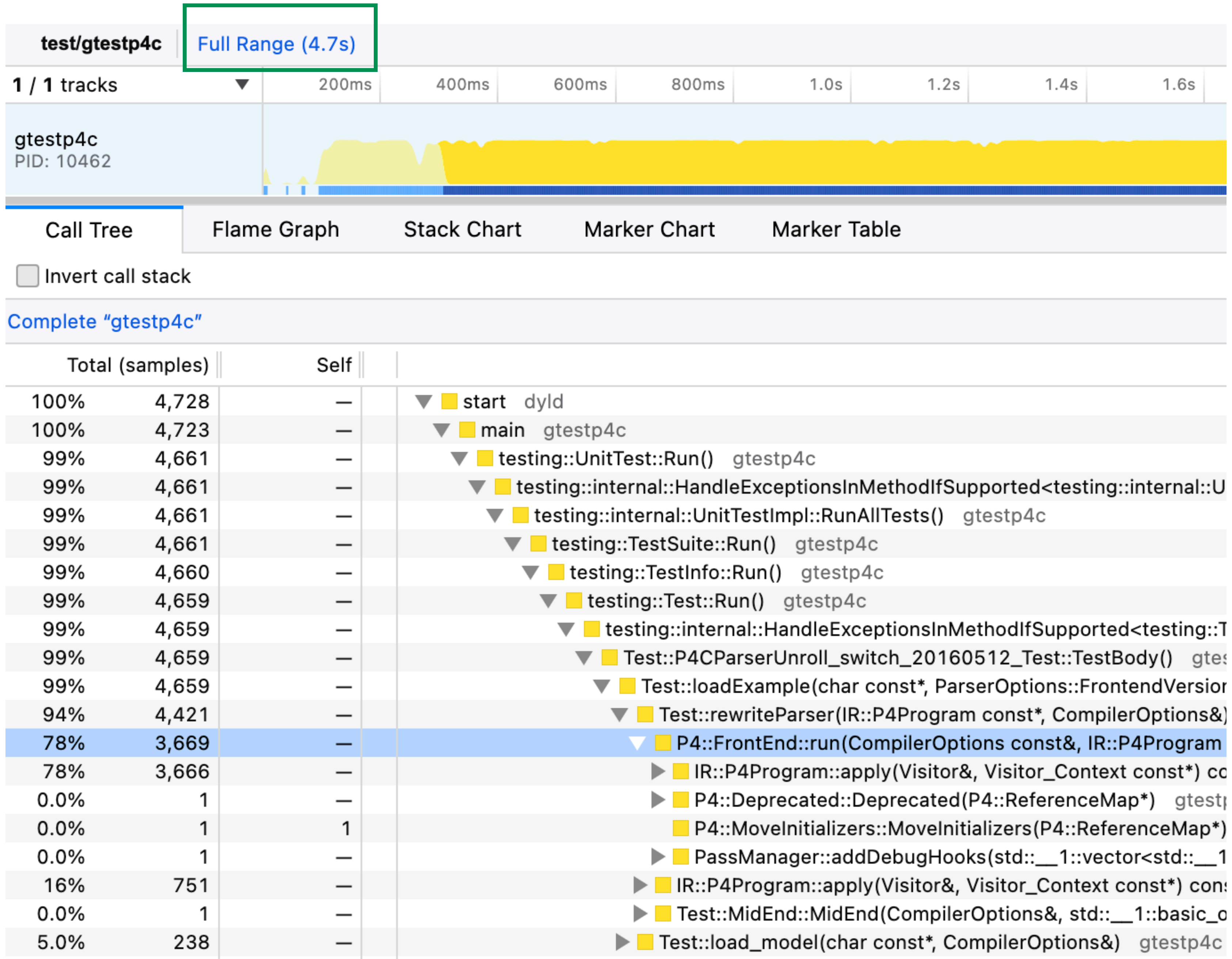
ACCESS SOFTEK, INC

Results (maps + use-def malloc traffic)



Complete “gtestp4c”

Total (samples)		Self	
100%	6,751	—	▼ start dyld
100%	6,745	—	▼ main gtestp4c
99%	6,683	—	▼ testing::UnitTest::Run() gtestp4c
99%	6,683	—	▼ testing::internal::HandleExceptionsInMethodIfSupported<testing::internal::U
99%	6,683	—	▼ testing::internal::UnitTestImpl::RunAllTests() gtestp4c
99%	6,683	—	▼ testing::TestSuite::Run() gtestp4c
99%	6,683	—	▼ testing::TestInfo::Run() gtestp4c
99%	6,683	—	▼ testing::Test::Run() gtestp4c
99%	6,683	1	▼ testing::internal::HandleExceptionsInMethodIfSupported<testing::T
99%	6,682	—	▼ Test::P4CParserUnroll_switch_20160512_Test::TestBody() gtes
99%	6,682	—	▼ Test::loadExample(char const*, ParserOptions::FrontendVersior
95%	6,437	—	▼ Test::rewriteParser(IR::P4Program const*, CompilerOptions&)
80%	5,411	1	▽ P4::FrontEnd::run(CompilerOptions const&, IR::P4Program
80%	5,398	—	▶ IR::P4Program::apply(Visitor&, Visitor_Context const*) co
0.1%	10	1	▶ P4::TypeMap::~TypeMap() gtestp4c
0.0%	1	—	▶ P4::RemoveParserIfs::RemoveParserIfs(P4::ReferenceMa
0.0%	1	1	P4::LocalizeAllActions::LocalizeAllActions(P4::Reference
15%	1,019	—	▶ IR::P4Program::apply(Visitor&, Visitor_Context const*) con:
0.1%	5	—	▶ P4::TypeMap::TypeMap() gtestp4c
0.0%	1	—	▶ Test::MidEnd::MidEnd(CompilerOptions&, std::__1::basic_o



Complete “gtestp4c”

Total (samples)		Self	
100%	4,728	—	▼ start dyld
100%	4,723	—	▼ main gtestp4c
99%	4,661	—	▼ testing::UnitTest::Run() gtestp4c
99%	4,661	—	▼ testing::internal::HandleExceptionsInMethodIfSupported<testing::internal::U
99%	4,661	—	▼ testing::internal::UnitTestImpl::RunAllTests() gtestp4c
99%	4,661	—	▼ testing::TestSuite::Run() gtestp4c
99%	4,660	—	▼ testing::TestInfo::Run() gtestp4c
99%	4,659	—	▼ testing::Test::Run() gtestp4c
99%	4,659	—	▼ testing::internal::HandleExceptionsInMethodIfSupported<testing::T
99%	4,659	—	▼ Test::P4CParserUnroll_switch_20160512_Test::TestBody() gtes
99%	4,659	—	▼ Test::loadExample(char const*, ParserOptions::FrontendVersior
94%	4,421	—	▼ Test::rewriteParser(IR::P4Program const*, CompilerOptions&)
78%	3,669	—	▽ P4::FrontEnd::run(CompilerOptions const&, IR::P4Program
78%	3,666	—	▶ IR::P4Program::apply(Visitor&, Visitor_Context const*) cc
0.0%	1	—	▶ P4::Deprecated::Deprecated(P4::ReferenceMap*) gtest
0.0%	1	1	P4::MoveInitializers::MoveInitializers(P4::ReferenceMap*)
0.0%	1	—	▶ PassManager::addDebugHooks(std::__1::vector<std::__1
16%	751	—	▶ IR::P4Program::apply(Visitor&, Visitor_Context const*) cons
0.0%	1	—	▶ Test::MidEnd::MidEnd(CompilerOptions&, std::__1::basic_o
5.0%	238	—	▶ Test::load_model(char const*, CompilerOptions&) gtestp4c

Another 40% reduction….

ACCESS SOFTEK, INC

ReferenceMap / TypeMap rants

- Both `ReferenceMap` and `TypeMap` are recalculated from scratch after every (!) IR modification
  - `TypeMap` involves whole-program type inference / type checking
  - `ReferenceMap` involves whole-program name / declaration resolution
- Often recalculated before every pass execution
  - Even if we'd only need couple of declarations / types
  - Standard pass combo: `ResolveReferences + TypeChecking + Pass`
- Could be recalculated multiple times during pass execution
  - `Inliner` does this after every successful function inlining
  - Lots of `PassRepeated` cases
  - Some type checking is done at every `MethodInstance::resolve()` call
  - …

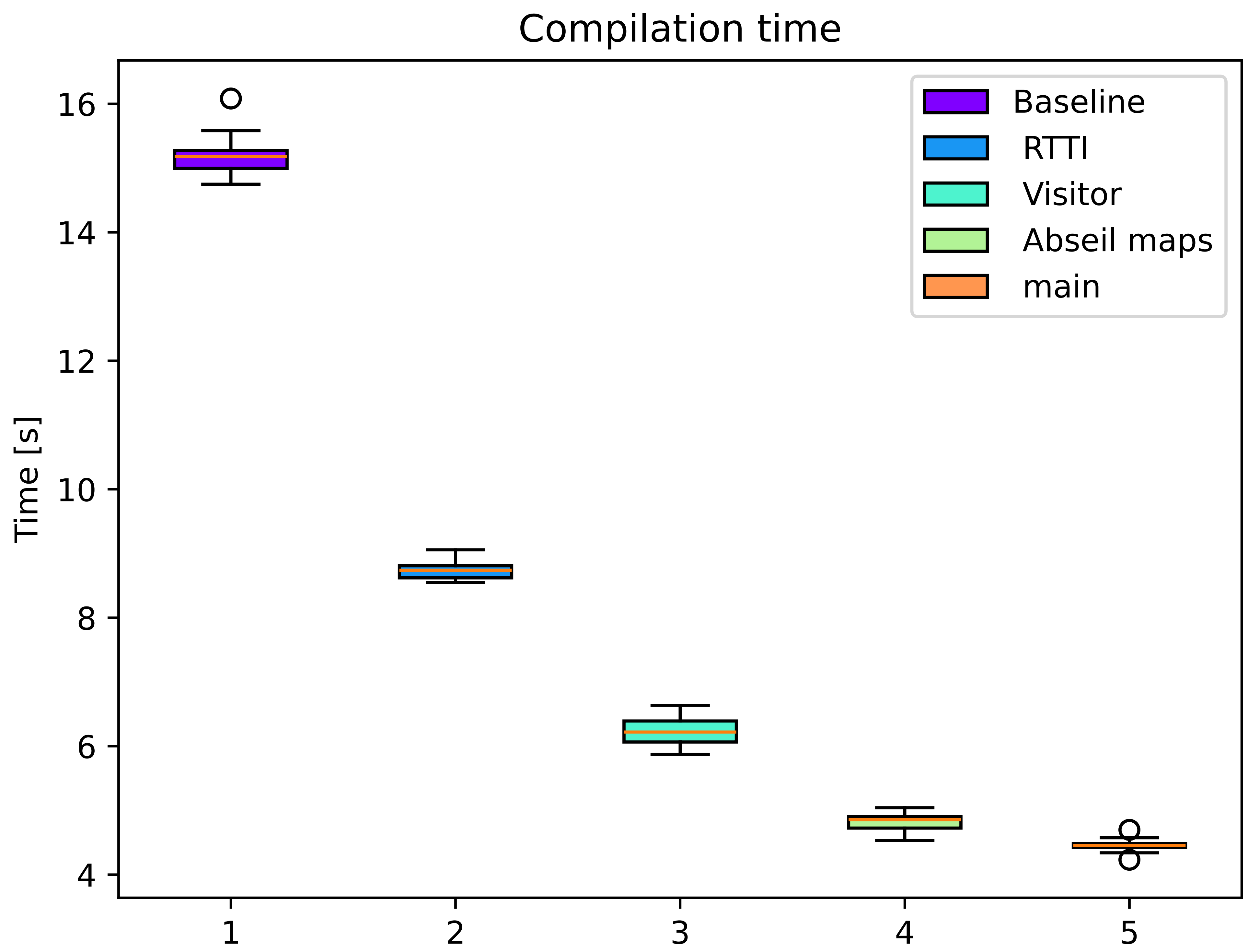
ReferenceMap elimination

- Use `ResolutionContext` pass mixin
  - Performs declaration resolution on fly & caches results
  - Requires more accurate context tracking and context inheritance
  - Resolves within current context only: cannot be used to query declarations in the context of callee from the caller
- Ported almost the whole frontend, except few places where significant refactoring would be required
- Midend is still there as-is except passes shared with frontend

More changes & improvements

- Improved some common classes internals (e.g. `IndexedVector<T>`)
- Improved `cstring` cache to reduce number of lookups
- Added `string_map<Value>` – same as `ordered_map<cstring, Value>`, but done properly
- `TypeChecking` / `TypeInference` improvements: `TypeChecking` is a proper `Inspector` now
  - Do not `clone()` everything just to immediately drop it
- Improve def-use memory consumption even more (both transient and peak)

Results: before vs now



Overall 3.5x improvement

check-p4 times

- Running `ninja check-p4` in 10 threads (not apples-to-apples though):

- Before:

```
p4    = 1170.63 sec*proc (1216 tests)
Total Test time (real) = 117.43 sec
```

- After:

```
p4    = 730.66 sec*proc (1248 tests)
Total Test time (real) =  73.39 sec
```

Results: large downstream app

- 43k lines of real P4 code (5x times larger than `switch_20160512` app)
- Compile time before (P4C v1.2.4.8): 396.45 seconds
- Compile time after (P4C v1.2.4.15): 56.9 seconds
- Overall **6.97x** improvement!
- Still pretty slow and more speedup is desired!

Lessons learned & ToDo

- IR is immutable
  - Lots of overheads here and there
  - P4C just allocates memory and does `clone()` majority of the time
- Reduce memory allocations & overheads as much as possible:
  - Switch to reference counting?
  - Try to allocate lots of things inline (aka “trailing objects”)
  - Allocate IR nodes from some arena / pool
  - Eliminate ReferenceMap & TypeMap entirely
- Reduce Visitor overhead:
  - Do not do unnecessary `clone`() in Transform
  - Track visited nodes somehow better?
- Maybe some other IR?
  - MLIR FTW?

Thank You

Immutable IR design rants

- Small change requires cloning of the whole IR subtree
- No sane way to update side structures on IR change
- No parent links: need to establish use-user relationship on-fly
  - Requires context lookup
  - Or even subtree walk
- No IR ownership
  - Just some objects allocated from global heap
- Low-level access to IR
  - One can create IR nodes anywhere