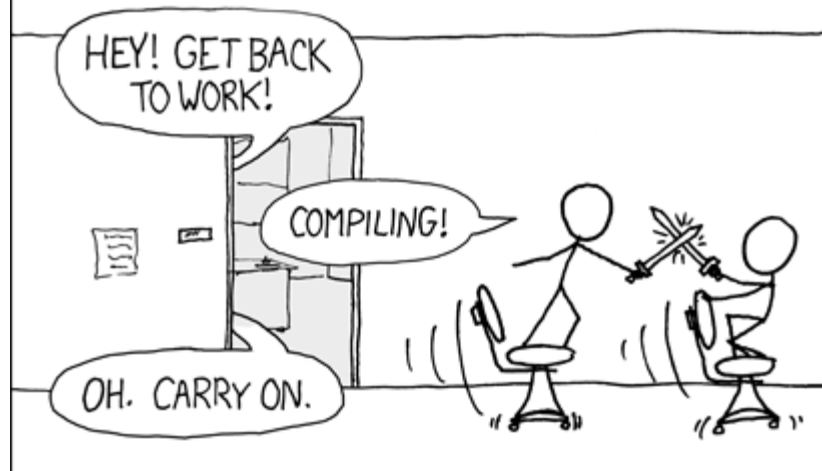


Run Scalac, Run

Zainab Ali

THE #1 PROGRAMMER EXCUSE
FOR LEGITIMATELY SLACKING OFF:
"MY CODE'S COMPILING."

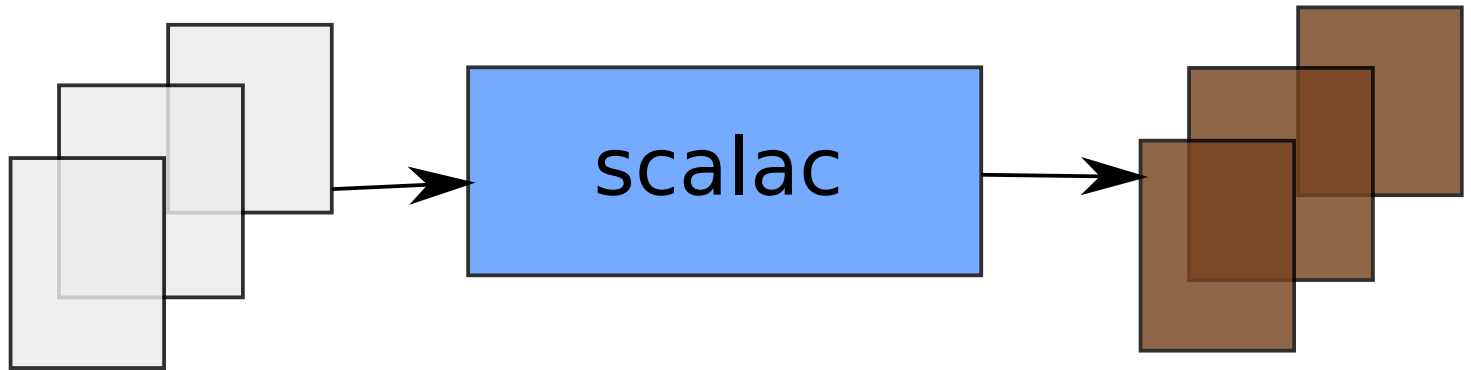


Services	11.754s	23.508s	35.262s	47.016s	58.770s
[-] bloop	.58.766s : compile root (transitively)
[+] bloop	.58.768s : background io work after compiling root (transitively)
[+] bloop	.175.754ms : computing bundle models
bloop	.97μs : compile models
[+] bloop	.4.298ms : computing bundle ml
[+] bloop	.6.545ms : computing bundle services
bloop	.40μs : compile ml
bloop	.36μs : compile services
[+] bloop	.176.224ms : computing bundle root
[-] bloop	.58.349s : compile root
bloop	.13.678ms : enabling plugin caching
bloop	.1.152ms : wait on populating products
bloop	.8.612ms : creating zinc inputs
[-] bloop	.56.356s : zinc endpoint
[+] bloop	.19.427ms : detecting initial changes
bloop	.54.313s : scalac

What is scalac?

.scala

.class



39.511s 36 spans

bloop x36 39.511s

4 minutes ago

44.770s 36 spans

bloop x36 44.770s

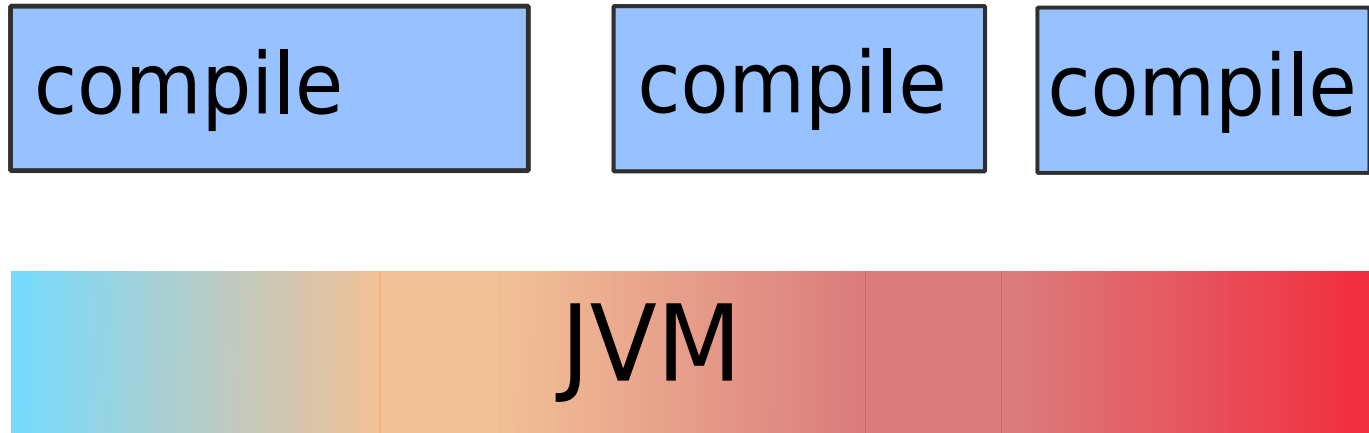
5 minutes ago

58.770s 37 spans

bloop x37 58.768s

6 minutes ago

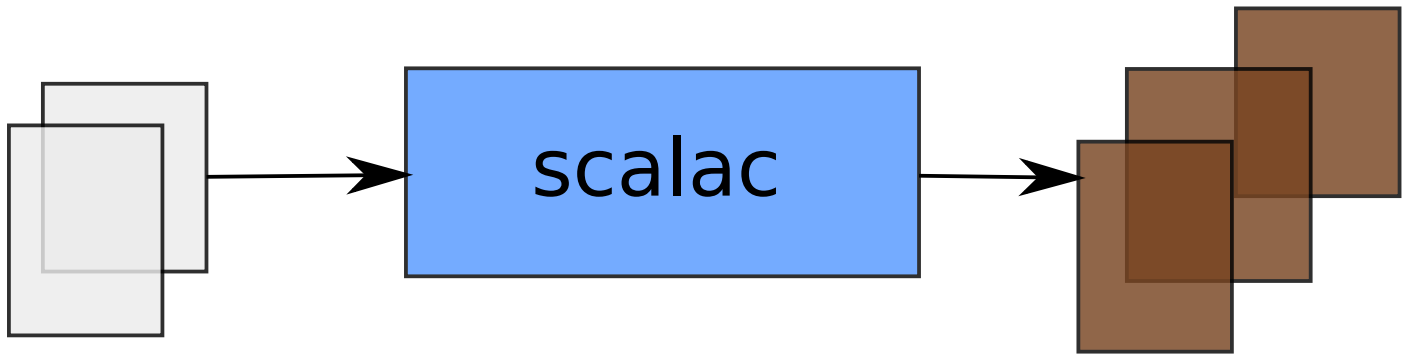
JIT Compilation



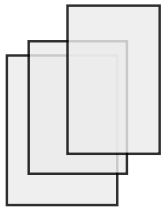
Incremental compile

changed

.class



unchanged .class



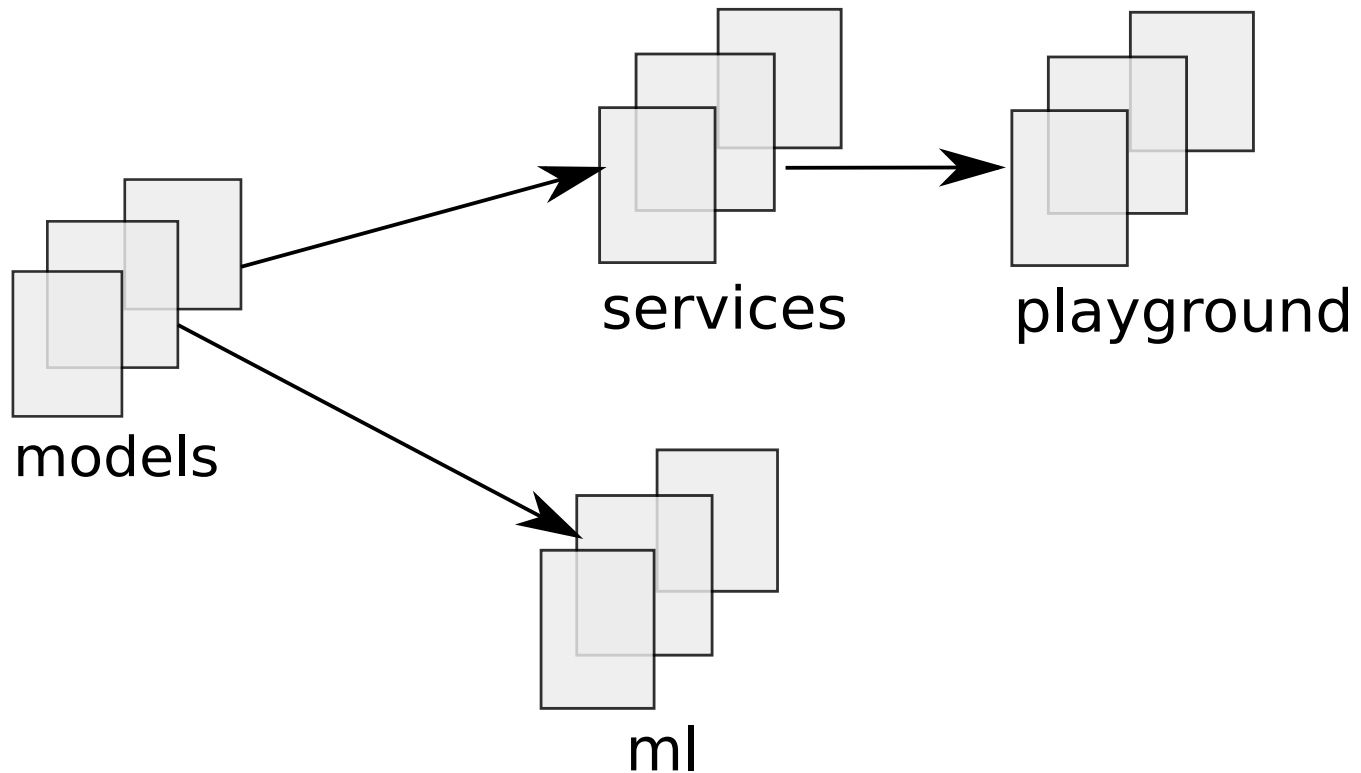
Speeding things up

do less work

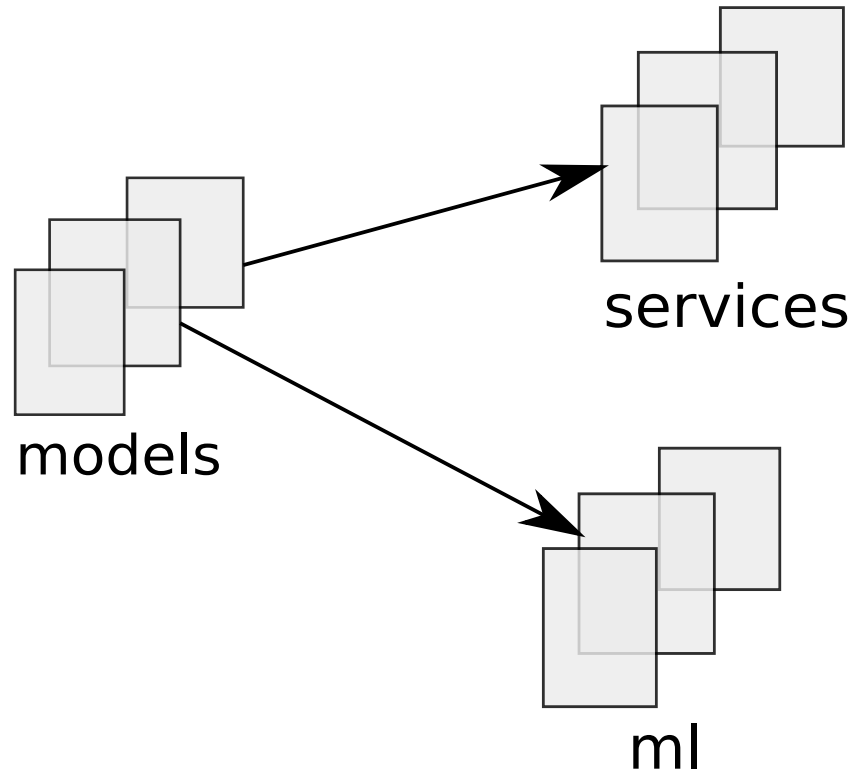
do it in parallel

do it better

Delete your code



Split up your codebase



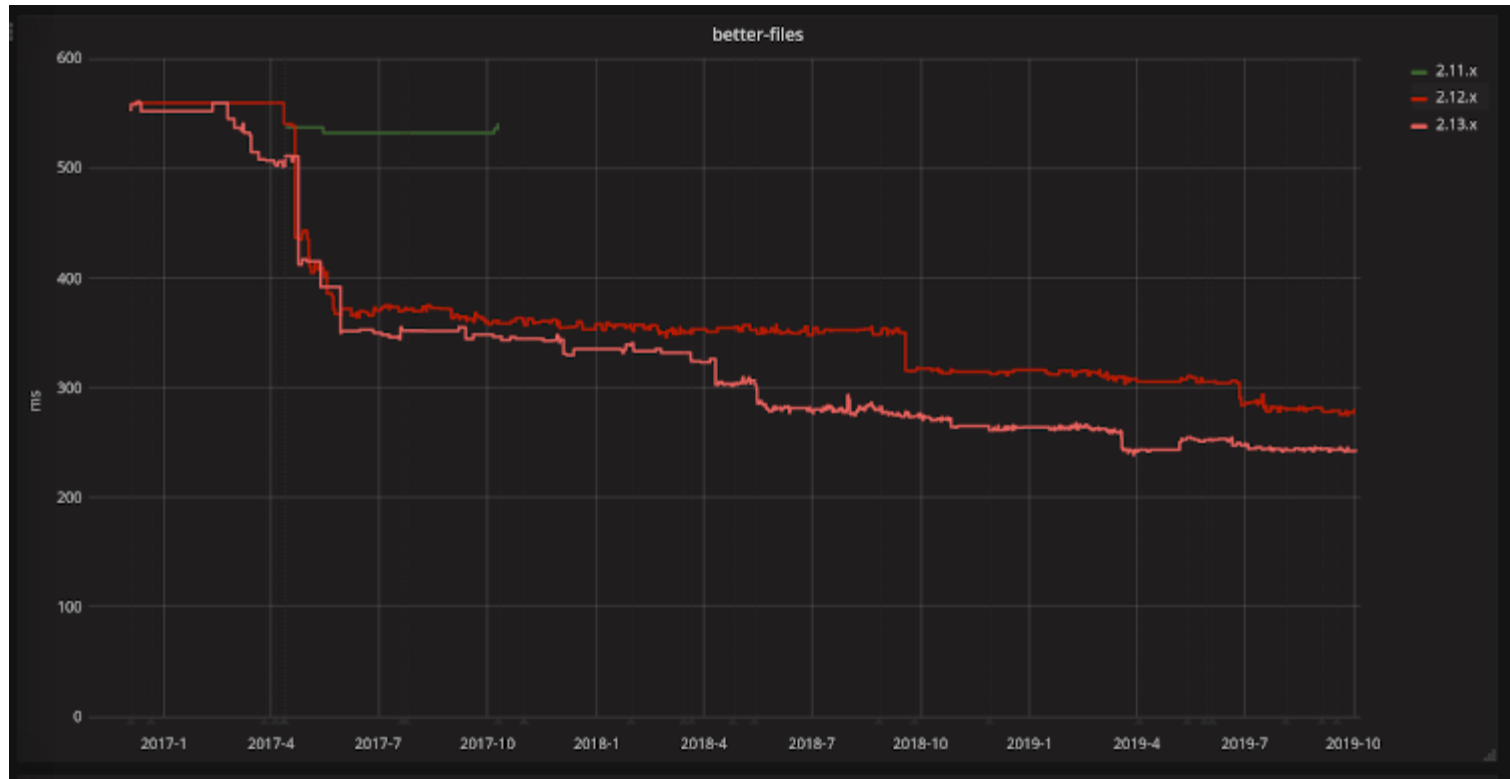
Split up your codebase

```
lazy val models = (project in file("models"))  
  
lazy val services = (project in file("services"))  
  .dependsOn(models)  
  
lazy val ml = (project in file("ml"))  
  .dependsOn(models)  
  
lazy val root = (project in file("."))  
  .aggregate(models, services, ml)
```

Services

	4.766s	9.533s	14.299s	19.065s	23.832s
[-] bloop	.23.831s : compile root (transitively) . . .				
[+] bloop	.23.087s : background io work after compiling root (transitively) . .				
[+] bloop	.25.019ms : computing bundle models . . .				
[+] bloop	.11.610s : compile models . . .				
[+] bloop	. . . 24.397ms : computing bundle ml .				
[+] bloop	. . . 28.869ms : computing bundle services .				
[+] bloop	. . . 12.584s : compile ml .				
[+] bloop	. . . 12.546s : compile services .				
[+] bloop 8.75				
bloop 10µs				

Upgrade Scala



<https://scala-ci.typesafe.com/grafana/dashboard/db/scala-benchmark>

Tweak the JVM

GraalVM™

So far...

	Time (s)	Improvement (s)
Cold JVM	59	
Warm	40	19
Code deletion	31	9
Multi project	23	8
Upgrades	18	4

The Scala compiler

```
scalacOptions += "-Ystatistics"
```

```
info] #checked applications : 10000
info] *** Cumulative timers for phases
info] #total compile time : 1 spans, ()10250.434ms
info] parser : 1 spans, ()45.902ms (0.4%)
info] namer : 1 spans, ()118.95ms (1.2%)
info] packageobjects : 1 spans, ()0.522ms (0.0%)
info] typer : 1 spans, ()3699.459ms (36.1%)
info] superaccessors : 1 spans, ()92.642ms (0.9%)
info] extmethods : 1 spans, ()14.194ms (0.1%)
info] pickler : 1 spans, ()122.533ms (1.2%)
info] xsbt-api : 1 spans, ()763.787ms (7.5%)
info] xsbt-dependency : 1 spans, ()339.265ms (3.3%)
info] refchecks : 1 spans, ()446.784ms (4.4%)
info] patmat : 1 spans, ()416.136ms (4.1%)
info] uncurry : 1 spans, ()305.374ms (3.0%)
info] fields : 1 spans, ()85.616ms (0.8%)
info] tailcalls : 1 spans, ()73.555ms (0.7%)
info] specialize : 1 spans, ()243.949ms (2.4%)
info] explicitouter : 1 spans, ()114.723ms (1.1%)
info] erasure : 1 spans, ()1183.17ms (11.5%)
info] posterasure : 1 spans, ()103.693ms (1.0%)
info] lambdalift : 1 spans, ()148.897ms (1.5%)
info] constructors : 1 spans, ()171.511ms (1.7%)
info] flatten : 1 spans, ()63.857ms (0.6%)
info] mixin : 1 spans, ()104.421ms (1.0%)
info] cleanup : 1 spans, ()37.811ms (0.4%)
info] delambdafy : 1 spans, ()55.084ms (0.5%)
info] jvm : 1 spans, ()1429.737ms (13.9%)
info] xsbt-analyzer : 1 spans, ()15.098ms (0.1%)
info] Done compiling.
```


The typer : elaboration

- resolve types
- resolve terms
- implicit searches

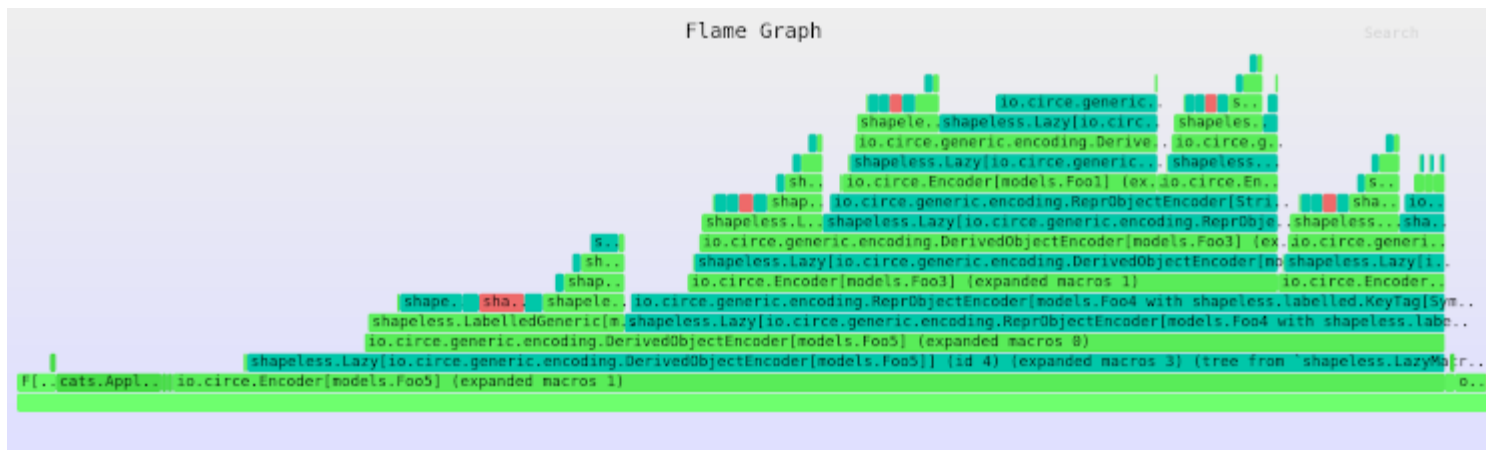
```
import even.more.implicit._

val num: Int = 42
val bar = num.magic

implicit def magicConversion(val num: Int): Foo = ...
implicit def anotherMagicConversion(val num: Int): Bar = ...
```

Profiling the typer

```
addCompilerPlugin("ch.epfl.scala" %% "scalac-profiling" % "1.0.0")
```



Implicit derivation

```
import io.circe.generic.auto._

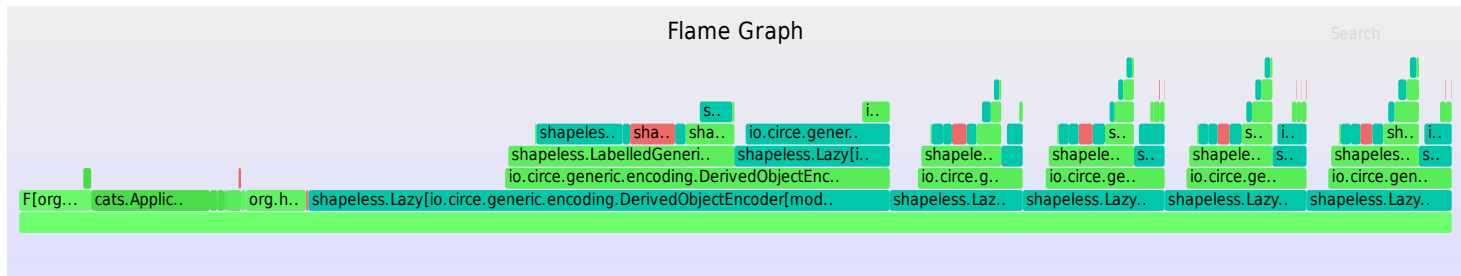
def routes: HttpRoutes[F] = {
  HttpRoutes.of[F] {
    case GET -> Root / "hello" / name =>
      Ok(foo) // automatic generic derivation
  }
}
```

Implicit derivation

```
import io.circe._, io.circe.generic.semiauto._

final case class Foo5(name: Foo4, fre: Foo3, sdf: Foo3)

object Foo5 {
  implicit val foo5Encoder: Encoder[Foo5] = deriveEncoder[Foo5]
}
```



There is hope!

- Scalac is simple
- Scalac can be fast
- Simple fixes have large gains

Try it yourself

- [London Scala User Group workshop](#)
- [Bloop performance guide](#)
- [Speeding up compilation times with scalac-profiling](#)
- [ScalaClean](#)
- [GraalVM](#)

Thank you!

