Master Thesis: Rust code generation (Part of CDA project)

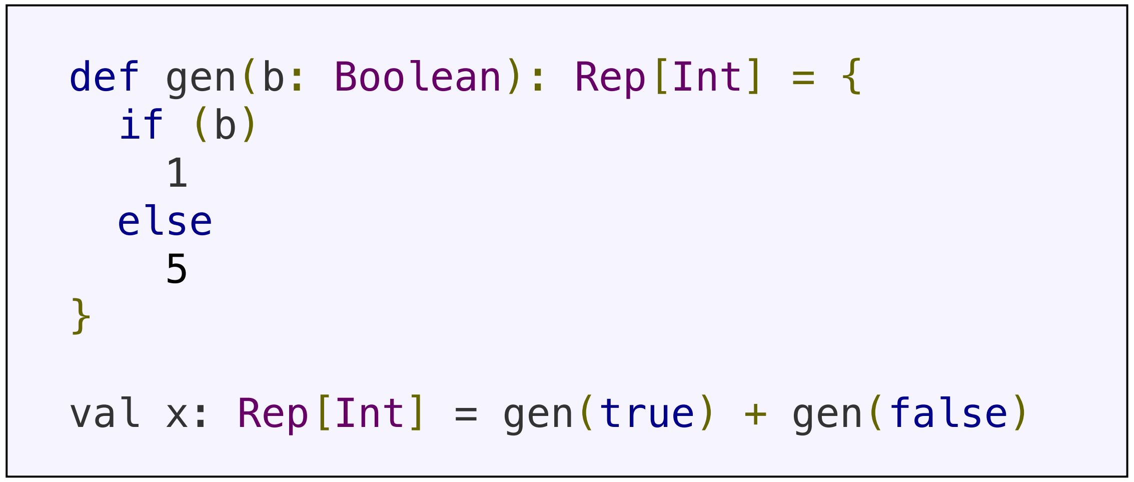
Klas Segeljakt

Problem: Spark/Flink run on the JVM => 10x - 1000x slower than compiled code

Motivation: Need better performance to support new programming models (ML tensors, graph analytics)

Idea: Generate Rust code with *Lightweight Modular Staging (LMS)*

- Rep[T] : Overload expression T to generate code

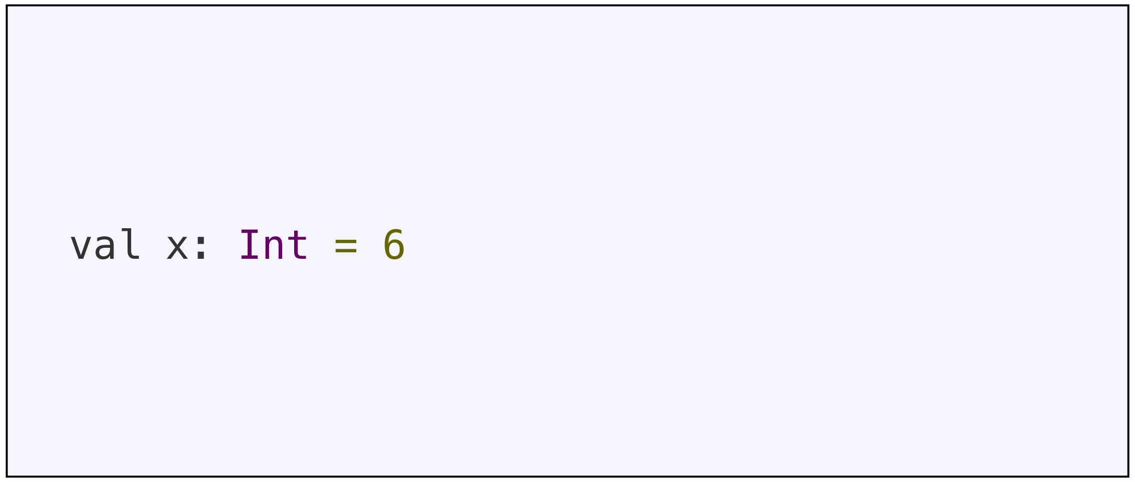


```
def gen(b: Boolean): Rep[Int] = {
  if (b)
    1
  else
    5
}

val x: Rep[Int] = gen(true) + gen(false)
```

Original Scala

Generate



```
val x: Int = 6
```

Generated Scala, (Will be Rust)