

Scala v. Clojure

David Pollak

QCon Beijing 2015

About @dpp



- Founded Lift, Scala since 2006
- Wrote Commercial Spreadsheets (3)
- Brick Alloy: We Solve Hard Problems

Clojure & Scala



- My Opinion and Perspective
抛砖引玉
- Much is Likely Factually Incorrect
- I'm a Stupid Bastard — 笨蛋

Scala is my "Go To" Language



- "The food is terrible"
- "And such small portions, too."

History of a Language Slut



- NextStep (ObjC & C++) '90-'96
- Java '96-'01
- C# '02-'03
- Ruby '04-'06
- Scala '06-Now
- Clojure '13-Now

Overview



• Clojure



• Scala



Philosophy



- ◉ Scala
 - ◉ Better than Java
 - ◉ Better Theoretical Underpinnings
 - ◉ "We're smarter"
- ◉ Clojure
 - ◉ Gluing together bits of excellence
 - ◉ Well considered
 - ◉ Rhymin' & Stealin'

Opinionation

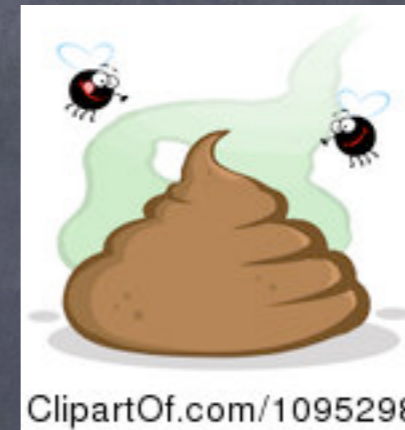


- Scala
 - OO or FP or...
 - Mutability, Immutability... eh...
 - But Types!!!
- Clojure
 - State goes here!
 - Smaller, simpler is better

Tooling



- Scala: Mostly Crap
 - SBT - mule vomit
 - Eclipse - 8 years of not working
 - IntelliJ - Pretty good with 2GB
- Clojure: Nice for a niche language
 - Lein - Easy to use, works
 - REPL - Yep
 - Vim & Emacs & Cursive - All very nice



Documentation



- Scala
 - Excellent core library docs
 - Akka well documented
 - Lift: Sucks rubber donkey lungs
- Clojure
 - Core: Very Lift-like documentation
 - Horrible to navigate
 - But nice READMEs on most GitHub projects



雞看不見雞背脊

Stability



- Scala

- Version Fragility
- Language & Library Instability
- Compiler & Library Bugs



- Clojure

- Libraries "Just Work"
- Stable



ScalaC



- Unbelievably Slow:
21K LoC 138 seconds
- Batch Development
- Horrid type inference

Implicit



- An interesting idea
- In all 00 languages
- Complete failure

Clojure Gripes



- One error at a time compiler
- Stack Traces!!!! When compiling
- Lame Clojure/ClojureScript co-projects – Getting better

Other Thoughts



- Transducers v. Scala 2.8+ Collections
- Protocols v. Type Classes
- Scala doesn't support `java.util.*`

Ecosystem



- TypeSafe v. Cognitect
- Akka v. core.async
- Lift v. pieces-parts

Maturity



- Clojure owns this
 - Technically
 - Community-wise

Static vs. Dynamic



- Dynamic OO is Hell (for me)
- FP Dynamic:
Smaller footprint of errors
- Still love Scala for refactoring large codebases
- Easier to "see" complex data in Scala
- Easier to navigate complex data in Clojure

Data Predictability



- How predictable is the shape of your data?
- That's how well you can "reason" about your code with a type system

Greenspun's 10th



- Any sufficiently complicated C or Fortran program contains an ad hoc, informally-specified, bug-ridden, slow implementation of half of Common Lisp.

@dpp's
11th



- And any sufficiently complicated Lisp program contains an ad hoc, informally-specified, bug-ridden, slow implementation of a type system

Perspective



- Clojure is about Data
- Scala is about Types
- Java is about Objects

How Many Semantics?



- Clojure: S-expressions
- Scala: Classes and Types
- Java: Objects and Annotations and XML++ and ...

Future (My Prediction)



- Neither Scala nor Clojure will replace Java!
- Excellent "Systems" Languages: Spark, Storm, etc.
- Excellent for focused pieces of a project

EOF



- Thanks!