Scala

A Critical Analysis

Motivation

OpenSource Software - Constantly Changing Landscape

In The Real World - Software Adoption
Language Adoption!?

The Rise of Functional Programming

Aims

To Develop an Argument - Is Scala viable for current and future development?

A Critical Analysis:

- Discover Key Scala Problem Areas.
- Identify Key Advantages and Disadvantages in Adopting Scala Within An Area.
- Compare Results With Other Popular Languages - Java, Python, Haskell etc.
- In-Depth Assessment of Individual Core Language Features and Scala Ecosystem
- Consider Other Arguments For and Against the Language.

An Introduction

A Multi-Paradigm Programming Language -

Both Object Oriented and Functional

- “Scala is a pure object-oriented language in the sense that every value is an object. ”
- “Scala is also a functional language in the sense that every function is a value.”

Fully Interoperable With Java:

- Designed with Java’s shortcomings in mind.
- Standard backend is the Java Virtual Machine.
- \+ An additional compiler Scala.js that compiles Scala code to equivalent JavaScript.

All In One Single, Concise and Elegant Language...


```java
public static void quickSort(int[] array, int left, int right) {
    if (right <= left) {
        return;
    }
    int pivot = array[right];
    int p = left;
    int i = left;
    while (i < right) {
        if (array[i] < pivot) {
            if (p != i) {
                int tmp = array[p];
                array[p] = array[i];
                array[i] = tmp;
            }
            p += 1;
        }
    i += 1;
    }
    array[right] = array[p];
    array[p] = pivot;
    quickSort(array, left, p - 1);
    quickSort(array, p + 1, right);
}
```


```scala
def quickSort(list: List[Int]): List[Int] = list match {
  case Nil => Nil
  case head :: tail => quickSort(tail.filter(_ < head)) :::
       head :: quickSort(tail.filter(_ >= head))
}

val nums: List[Int] = List(11,3,50,10,3,7)
println(quickSort(nums))
```

.sc Evaluation

```
nums: List[Int] = List(11, 3, 50, 10, 3, 7)
List(3, 3, 7, 10, 11, 50)
```

Scala's Type System

- Scala is Statically Typed
- Scala Offers Local Type Inference
- Polymorphic Methods and Functions are Parameterizable by Type

```
def listOfDuplicates[A](x: A, length: Int): List[A] = {
  if (length < 1)
    Nil
  else
    x :: listOfDuplicates(x, length - 1)
}

println(listOfDuplicates(3,  length = 4))
println(listOfDuplicates[Double](3,  length = 4))
println(listOfDuplicates("La",  length = 4))
```

.sc Evaluation

```
List(3, 3, 3, 3)
List(3.0, 3.0, 3.0, 3.0)
List(La, La, La, La)
```

Scala In The Real World - A Study

Detailed at the Official Twitter Developer Conference 2010:

- Twitter famously uses/used Ruby on Rails.
- However RoR is a dynamically typed language.
- Therefore the adoption of Scala for distributed/long-running processes.

Preliminary Results -
Initial Focus Areas

We've discovered some initial focus areas -

- Concurrency
- Big Data

Also looked at some individual core language feautures and the benefits that they provide.

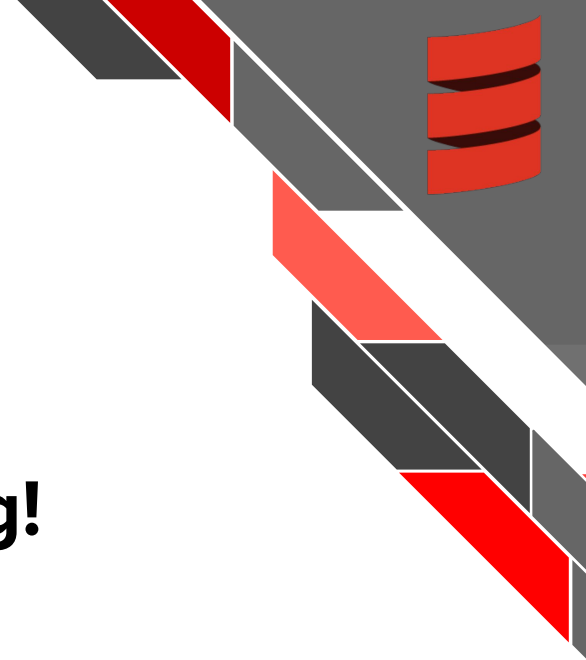
Further Work

Further Analysis of Scala... - “*Being a little vague?*”

Some Specifics -

- Concurrency: Benchmark Bitonic Sorting Algorithm - $O(n \log^{2n})$ Comparisons

- Scala For Machine Learning - Discovering Apache Spark



Thank you for listening!