CS846 Paper Review Form - Winter 2012

Reviewer: Kaheer Suleman

Paper Title: An Overview of the Scala Programming Language
Authors: Odersky, M., Altherr, P., Cremet, V., Dragos, I., Dubochet, G., Emir, B., McDirmid, S., Micheloud, S., Mihaylov, N., Schinz, M., Stenman, E., Spoon, L., Zengerhttp

Author(s):

1) Is the paper technically correct?
 [x] Yes
 [ ] Mostly (minor flaws, but mostly solid)
 [ ] No

2) Originality
 [ ] Very good (very novel, trailblazing work)
 [x] Good
 [ ] Marginal (very incremental)
 [ ] Poor (little or nothing that is new)

3) Technical Depth
 [ ] Very good (comparable to best conference papers)
 [x] Good (comparable to typical conference papers)
 [ ] Marginal depth
 [ ] Little or no depth

4) Impact/Significance
 [ ] Very significant
 [ ] Significant
 [x] Marginal significance.
 [ ] Little or no significance.

5) Presentation
 [ ] Very well written
 [x] Generally well written
 [ ] Readable
 [ ] Needs considerable work
 [ ] Unacceptably bad

6) Overall Rating
 [ ] Strong accept (award quality)
 [x] Accept (high quality - would argue for acceptance)
 [ ] Weak Accept (borderline, but lean towards acceptance)
 [ ] Weak Reject (not sure why this paper was published)

7) Summary of the paper's main contribution and rationale
   for your recommendation. (1-2 paragraphs)
The authors provide an indepth summary of the high level programming language Scala. Scala , was created to allow for greater support for component based software engineering similar to the manner in which hardware is engineered. The language brings together both functional and object oriented programming. In order to increase the adoption of the language , Scala is compatible with both Java and C# and has a number of similarities with both of these object oriented languages. The paper describes in detail each of the components of the language. In Scala all values
are objects and all operations are method calls. This includes non reference primitives such as integers and floats. Scala further differs from Java in that there are two "null" classes scala.Nothing and scala.Null. All operations in Scala are method invocations. These include mathematical operations such as +. In scala a + b is equivalent to the method call a.+(b). Scala has greater support for generics than Java. In scala all objects (including functions which are objects in Scala) can have type parameters. Furhtermore, constraints can be set on these type definitions. The paper provides an example of constraining a type to one that can be ordered. Scala has more support for multiple inheritance than Java. Here Objects can have a single superclass but multiple "traits". Unlike interfaces which only specify method calls. Traits are allowed to have variables and method implementations. Finally, Scala supports pattern matching for method calls. Here the input is matched against a set of provided patterns and different operations can then
be executed. An example of evaluation of mathematical operations is provided in the paper. The pattern matching support allows for greater ease when dealing with XML strucured documents.

The above paper is well written and consise. It provides great detail on the capabilites of the Scala programming language. Furhtermore they authors provide many examples which helped in understanding the content. However, there were parts (especially in the discussion of variance and position) that I found difficult to understand and an example showing what each position with arrows would have been helpful. In terms of the significance of this paper, I dont see how it is that siginifcant. The language borrows ideas from multiple languages and seems to put them together in a interesting manner but there are other languages that serve a similar purpose as was seen in the related work section. I feel that the authors did not do a good job in explaining the differences and why their choices were better.

8) List 1-3 strengths of the paper. (1-2 sentences each,
identified as S1, S2, S3.)
Technical Depth: A large number of details were provided in the paper on the langauge. Each affordance in the language was described in great detail allowing for a greater understanding of the language.

Examples provides: The paper provided many examples that helped in understanding the topics presented. Examples were provided of erroroneous code as well
as correct examples to show where things can go wrong.

Well Organized: I found the paper to be well organized. Each sub section flowed nicely into the other which made reading the paper much easier.

9) List 1-3 weaknesses of the paper (1-2 sentences each,
identified as W1, W2, W3.)

Difficult Areas: A few parts of the paper were confusing. In particular I found the section on variance to be difficult to understand. However this might be due to my inexperience in the study of programming languages and am not too familiar with the jargon.

Comparison: I would have liked to see a comparison of the performance of the language against other languages that are designed for similar purposes. This however may be the objective of another study.