

Julia: high performance programing the easy way

Tuesday, August 28, 2018 1:00 PM (5 hours)

Besides offering fresh ideas and new programming concepts, Julia was mainly created to solve the two language problem.

The two language problem describes the common pattern of prototyping algorithms in an easy to use high level language and then reimplementing it in a fast language like C - doubling development costs and making updates and further development more complicated.

This also has led to a split in scientific computing: you work mainly in a scripting language, while all the performance critical libraries are unapproachable black boxes written in a more difficult language. For most users this is okay, but for developpers it makes growing the ecosystem more difficult and it's not as easy to engage users into contributing back to the core library.

Julia solves this with a sophisticated compiler model which manages to combine the usability of dynamic scripting languages with the performance of low level languages.

In this workshop I will introduce the basic mechanisms of how Julia works and will teach some fun programming examples showing how to use Julia's type system, meta programing and how to make any Julia program run as fast as highly optimized C - all while being at least as readable as python code!

I will also show some more advanced examples which will explain how Julia can offer completely new possibilities for library developpers, by having high performance libraries written in a dynamic language.

One of those examples is how to seamlessly move your code to the GPU and do e.g. automatic differentiation on the GPU and CPU alike without loosing any performance.

Summary

Presenter: DANISCH, Simon

Session Classification: Tutorials