

julia - Eine dynamische Sprache optimiert für die wissenschaftliche Programmierung

Igor Kröner

Alfred Wegener Institut, Helmholtz-Zentrum für Polar- und Meeresforschung, Potsdam, Germany (igor.kroener@awi.de)

In der angewandten Programmierung erfreuen sich high-level, dynamische Sprachen, wie zum Beispiel Matlab oder Python, immer grösserer Beliebtheit, da sich mit relativ geringem Aufwand sich bereits einige Routinen schreiben und viele Probleme lösen lassen. Diese Nutzerfreundlichkeit ist jedoch nicht selten mit deutlichen Performance-Einschränkungen erkauft. Für rechenintensive Aufgaben wird daher meistens noch immer auf low-level, statische Sprachen zurückgegriffen (z.B. C, Fortran). Diese sind zwar deutlich schneller, jedoch auch deutlich aufwendiger zu lernen und programmieren.

Die Entwickler von **julia** haben sich diesen Spagat, zwischen Spezialisierung und Abstraktion, zwischen den Bedürfnissen des Programmierers und denen der Maschine, zur Herausforderung gemacht. Sie wollten eine open-source Sprache erschaffen, welche leicht zu lernen ist und dennoch die individuellen Vorteile bisher existierender Sprachen wie C, Ruby, Lisp, Matlab, Python oder R vereint.

Natürlich erlaubt sich die Frage, inwiefern es lohnt eine neue Sprache zu lernen. In diesem Workshop ähnlichen Format möchte ich eine Einführung in die Welt von julia geben. Folgende Fragestellungen sollen dabei im Fokus stehen:

1. Wie lässt es sich als programmierender Wissenschaftler mit julia arbeiten?
2. Wie lassen sich bereits bestehende Programme, Funktionen oder Pakete anderer Sprachen unkompliziert in julia Quellcode integrieren?
3. Was gilt es zu beachten, wenn mit Rücksicht auf die Ausführung optimiert in julia programmiert werden soll?

Eine Teilnahme ist auch ohne Computer möglich. Einige Beispiele können jedoch auf einem eigenen Notebook unmittelbar nachvollzogen werden.