Veille Technologique RICM5

06/10/2017

Rémi SAVARY

Julia, qu'est-ce que c'est?

- Langage de programmation de haut niveau

- Auteurs: Jeff Bezanson, Stefan Karpinski, Viral B. Shah, Alan Edelman

- Première version: 23 août 2009

- Dernière version: 19 juin 2017 (Version 0.6.0)

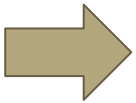
Une communauté active

Sur GitHub:

Motivations

Les objectifs de Julia sont:

- d’avoir les performances du langage C
- de pouvoir être utilisé comme Python
- d’avoir le dynamisme de Ruby
- de faire facilement des statistiques comme R
- d’avoir les utilisations mathématiques de Matlab
- d’être facile à apprendre



Langage de haut niveau, dynamique, performant, adapté pour le calcul scientifique comme pour la programmation “générale”

Principales fonctionnalités

- Très bonnes performances
- Distribution multiple
- Typage dynamique
- Permet le parallélisme, le calcul distribué
- Possède un shell intégré
- Gestionnaire de paquets intégré
- Un très grand nombre de librairies disponibles
- Prise en charge d’unicode
- Sous licence MIT (open source)

Performances

	Fortran gcc 5.1.1	Julia 0.4.0	Python 3.4.3	R 3.2.2	Matlab R2015b	Octave 4.0.0	Mathematica 10.2.0	JavaScript V8 3.28.71.19	Go go1.5	LuaJIT gsl-shell 2.3.1
fib	0.70	2.11	77.76	533.52	26.89	9324.35	118.53	3.36	1.86	1.71
parse_int	5.05	1.45	17.02	45.73	802.52	9581.44	15.02	6.06	1.20	5.77
quicksort	1.31	1.15	32.89	264.54	4.92	1866.01	43.23	2.70	1.29	2.03
mandel	0.81	0.79	15.32	53.16	7.58	451.81	5.13	0.66	1.11	0.67
pi_sum	1.00	1.00	21.99	9.56	1.00	299.31	1.69	1.01	1.00	1.00
rand_mat_stat	1.45	1.66	17.93	14.56	14.52	30.93	5.95	2.30	2.96	3.27
rand_mat_mul	3.48	1.02	1.14	1.57	1.12	1.12	1.30	15.07	1.42	1.16

Temps d’exécution par rapport à C (performances de C = 1)

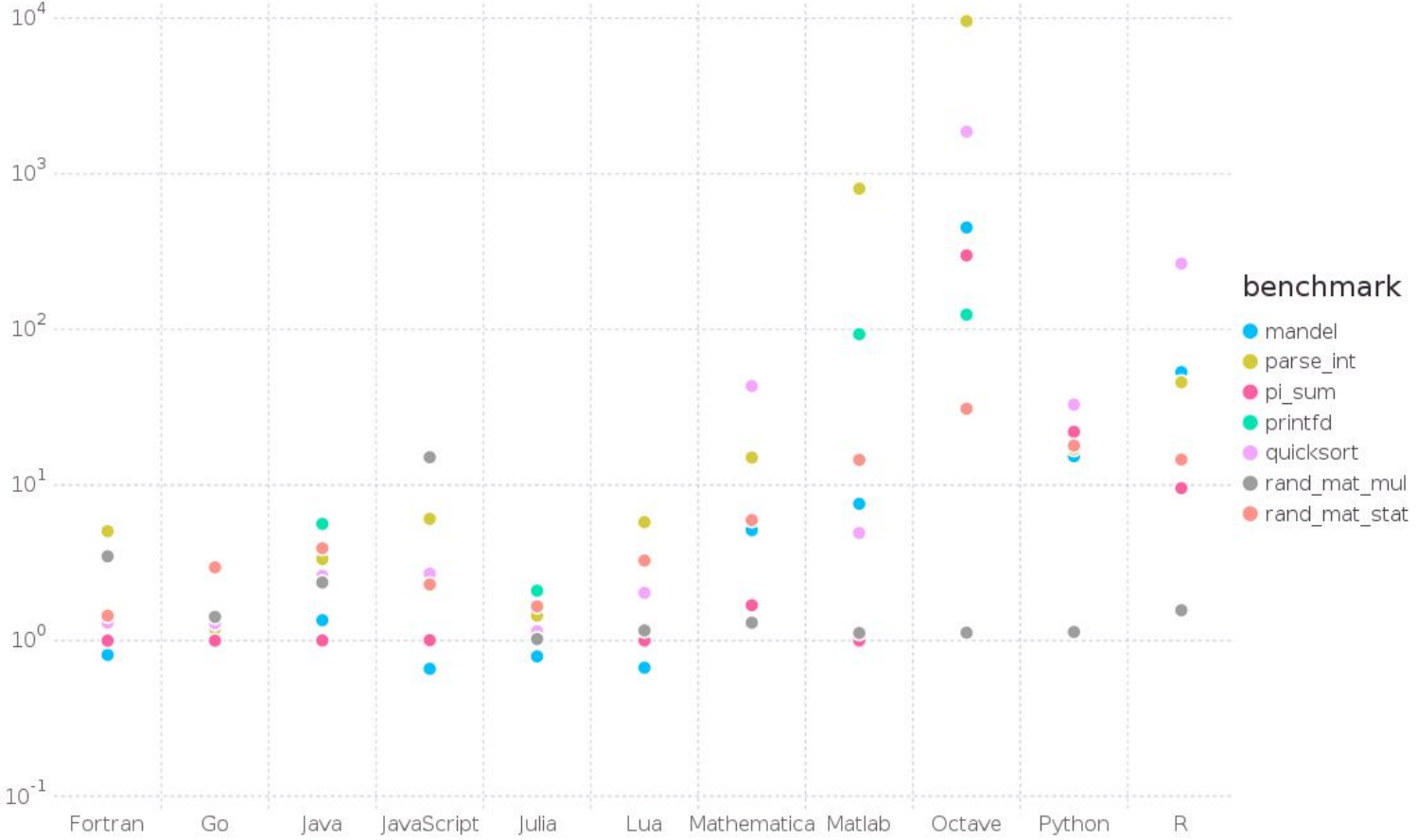


Figure: benchmark times relative to C (smaller is better, C performance = 1.0).

Place à la démonstration!