

 alazar 20 февраля 2012 в 18:10

Реклама

Почему мы создали Джулию, новый ЯП для вычислений

Автор оригинала: Viral Shah, Jeff Bezanson, Stefan Karpinski, Alan Edelman

Программирование, Julia, Matlab

[Перевод](#)

Если вкратце, потому что мы жадные.

Мы продвинутые пользователи Matlab. Некоторые из нас хакеры Lisp. Некоторые питонисты, другие рубисты, есть ещё Perl-хакеры. Среди нас есть такие, кто использовал Mathematica раньше, чем у него начали расти волосы на лице. Есть и такие, у кого до сих пор не выросли. Мы построили больше графиков на R, чем способен любой здравомыслящий человек. C — это язык, который мы бы взяли на необитаемый остров.

Мы любим все эти языки; они прекрасны и могучи. Для той работы, которую мы делаем — научные вычисления, машинное обучение, дата-майнинг, крупномасштабная линейная алгебра, распределённые и параллельные вычисления — каждый идеально подходит в определённом аспекте, но ужасен в других. Каждый из них — это компромисс.

Мы жадные: мы хотим больше.

Мы хотим язык с открытыми исходниками, под свободной лицензией. Мы хотим скорость C и динамизм Ruby. Мы хотим язык с равнозначностью кода и данных, с настоящими макросами как в Lisp, но с очевидными, знакомыми математическими нотациями, как в Matlab. Мы хотим удобное средство для обобщённого программирования как Python, простое для статистики как R, естественную обработку строк как Perl, мощную линейную алгебру как Matlab, хорошую склейку программ как shell. Что-то абсолютно простое в изучении, но при этом делающее счастливыми большинство серьёзных хакеров. Мы хотим это интерактивным и мы хотим это скомпилированным.

(Мы упомянули, что оно должно быть быстрым, как C?)

Пока мы продолжаем требовать, добавим ещё что-нибудь, обеспечивающее распределённую производительность Hadoop — без килобайтов boilerplate-кода Java и XML; без необходимости пробираться через гигабайты логов на сотнях машин, чтобы отловить баги. Мы хотим мощь без оболочек непроходимой сложности. Мы хотим писать простые скалярные циклы, которые компилируются в компактный машинный код, используя только регистры на одном CPU. Мы хотим написать A^*B и запустить тысячу процессов на тысяче машин, вместе вычисляющих огромную матрицу.

Мы не хотим упоминать типы данных без необходимости. Но если нам нужны полиморфные функции, мы хотим использовать обобщённое программирование для записи алгоритма один раз и применения его к бесконечной решётке типов; мы хотим использовать множественную диспетчеризацию для эффективного выбора лучшего метода для всех аргументов функции, из десятков описаний метода, обеспечивая общую функциональность среди радикально различных типов. Несмотря на всю эту мощь, мы хотим, чтобы язык был простым и ясным.

Мы ведь не слишком многого просим, верно?

Даже хотя мы осознаём свою непозволительную жадность, мы по-прежнему хотим всё это. Около двух с половиной лет назад мы начали создавать язык своей мечты. Он ещё не закончен, но пришло время для релиза 1.0 — язык программирования, который мы создали, называется Джулия. Она уже удовлетворяет 90% наших грубых запросов, и теперь ей нужно ещё больше грубых запросов от других людей, чтобы развиваться дальше. Так что, если вы жадный, безрассудный, требовательный программист, вы должны попробовать.

	C++ (GCC) 4.2.1*	Julia 7183d97a	NumPy 1.5.1	Matlab R2011a	Octave 3.4	R 2.9.0	JavaScript V8 3.6.6.11
fib	0.200	1.97	30.74	1360.47	2463.97	334.94	1.48
parse_int	0.242	1.22	16.49	827.13	6871.04	1082.67	2.14
quicksort	0.416	1.25	61.73	135.51	3357.78	971.06	6.66
mandel	0.249	7.15	31.98	66.27	870.55	253.10	6.03
pi_sum	53.524	0.74	18.77	1.09	356.08	269.19	0.74
rand_mat_stat	7.347	3.94	40.98	12.20	57.39	32.39	8.30
rand_mat_mul	227.331	1.01	1.20	0.78	1.69	2.65	291.83

Микробенчмарк на MacBook Pro с 2,53 ГГц Intel Core 2 Duo CPU и 8 ГБ DDR3 RAM 1066 МГц, для C++ указано абсолютное время выполнения в миллисекундах, остальные относительно к C++ (меньше — лучше)

Исходники всех бенчмарков

Пример кода Julia для множества Мандельброта и статистики случайных матриц

```
function mandel(z)
    c = z
    maxiter = 80
    for n = 1:maxiter
        if abs(z) > 2
            return n-1
        end
        z = z^2 + c
    end
    return maxiter
end

function randmatstat(t)
    n = 5
    v = zeros(t)
    w = zeros(t)
    for i = 1:t
        a = randn(n,n)
        b = randn(n,n)
        c = randn(n,n)
        d = randn(n,n)
        P = [a b c d]
        Q = [a b; c d]
        v[i] = trace((P.*P)^4)
        w[i] = trace((Q.*Q)^4)
    end
    std(v)/mean(v), std(w)/mean(w)
end
```

Теги: язык программирования, Matlab, обработка данных, Julia

Хабы: Программирование, Julia, Matlab

↑ +45 ↓ 73 8,5k 19 Поделиться



872,7

Карма

205,0

Рейтинг

Анатолий Ализар @alazar

хакер, редактор

Telegram

ПОХОЖИЕ ПУБЛИКАЦИИ

23 февраля 2021 в 12:23

Языку программирования Python исполнилось 30 лет

↑ 18 👁 5,5k 📖 20 💬 6

22 февраля 2021 в 07:30

Рейтинг языков программирования 2021: доля Python падает, а TypeScript обошел C++, в лидерах JavaScript, Java, C#

↑ 40 👁 24,2k 📖 47 💬 118

24 октября 2013 в 19:18

Язык программирования и база данных Q: в энтерпрайсе синтаксис роли не играет

↑ 12 👁 7,7k 📖 33 💬 10

ВОПРОСЫ И ОТВЕТЫ

Matlab строки и данные?

Обработка данных • Средний • 1 ответ

Matlab не может нормально запускаться. Как исправить?

MATLAB • Простой • 0 ответов

Как построить график функции с синусом, логарифмом и интегралом в Matlab?

Математика • Средний • 1 ответ

Как создать датасет из траектории движения в 3Д?

MATLAB • Средний • 0 ответов

Как изменить название приложения App Disgner Matlab?

MATLAB • Простой • 0 ответов

Больше вопросов на Хабр Q&A

Реклама

Комментарии 19

 **KiriKiri** 20 февраля 2012 в 18:44 🗨 📖 ↑ +2 ↓

Перевод как-то не очень законченным кажется. Но спасибо за то, что нашли и рассказали об этом новом языке.

 **Goryn** 20 февраля 2012 в 19:05 🗨 📖 ↑ +6 ↓

Не взлетит

 XenJ 20 февраля 2012 в 19:41 # 📖 ↺ ↻


↑ 0 ↓

Для любителей R и Matlab может и взлетит, но C или хотя бы Ruby не заменит, конечно. Но не для этого он создавался.

 KOLANICH 21 февраля 2012 в 15:28 # 📖 ↺ ↻

↑ +4 ↓


не нравятся мне все эти end

 moborb 20 февраля 2012 в 19:18 # 📖

↑ +4 ↓

А как же benchmark для Fortran?

Этот самый Formula Translator создавался для математических вычислений и, насколько я слышал, делает это еще быстрее, чем C/C++.

 Dair_Targ 20 февраля 2012 в 19:20 # 📖

↑ 0 ↓

Синтаксис один-в-один похож на Фортран.

 shisoik 20 февраля 2012 в 22:01 # 📖 ↺ ↻


↑ 0 ↓

Подозреваю, что это не случайно. Так будет проще продвигать среди целевой аудитории.

 Dair_Targ 21 февраля 2012 в 02:37 # 📖 ↺ ↻


↑ +9 ↓

«Программист на Фортране может на любом языке писать на Фортране»

 Beholder 20 февраля 2012 в 23:44 # 📖 ↺ ↻

↑ 0 ↓

А мне на первый взгляд больше напомнило Lua.

 slovak 20 февраля 2012 в 19:20 # 📖

↑ +2 ↓

Matlab... Matlab...

Кто бы реализовал Simulink под Octave?))

 aen 20 февраля 2012 в 19:41 # 📖 ↺ ↻

↑ 0 ↓

А разве нет аналогов для Scilab?

 Derailed 20 февраля 2012 в 20:02 # 📖

↑ 0 ↓

По C++ также результаты по верхним 4-м функциям вызывают сомнения. Как же нужно извернуться, чтобы так медленно их написать? ИМХО, довольно противоречивая табличка у Вас по быстродействию. Судя по таблице — чем же лучше Ваш язык по сравнению с остальными?

 webwarrior 20 февраля 2012 в 23:03 # 📖 ↺ ↻

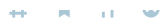
↑ +4 ↓

Ализар как всегда упустил важный комментарий к таблице:

**C++ numbers are absolute benchmark times in milliseconds;
other timings are relative to C++ (smaller is better).**



Derailed 20 февраля 2012 в 23:11



0

Так бы сразу, и таким же крупным шрифтом — иначе и не заметишь. А если бы оригинальный английский текст оставили — было бы даже понятнее.



catlion 20 февраля 2012 в 21:45



↑ +4 ↓

> C is our desert island programming language.

Более по-русски это звучит «С — это язык, который мы бы взяли на необитаемый остров».



nerudo 20 февраля 2012 в 22:33



↑ +2 ↓

Какие-то невероятные результаты — отличие на порядки то в одну, то в другую сторону. За счет разных алгоритмов, такое, понятно дело быть может, но лишь из-за различных языков... Лично мне сомнительно.



TheShock 20 февраля 2012 в 23:55



↑ +1 ↓

У одного языка оптимизируется хвостовая рекурсия, у второго — значительно более быстрая работа с массивами



TheShock 20 февраля 2012 в 23:58



↑ +2 ↓

Интересно вот что. Согласно тестам автора JS по скорости очень близок к Си++ (5 раз разница — это копейки) кроме последнего — rand_mat_mul, который ВСЕ языки выполняют очень быстро кроме него. Почему? Вообще похоже, что по ошибке поставили время, а не фактор замедления ($291/227=1,28$)



dmitriid 21 февраля 2012 в 12:35



↑ +2 ↓

github.com/JuliaLang/julia/tree/master/test/perf

Тут исходники всех бенчмарков

Только полноправные пользователи могут оставлять комментарии. Войдите, пожалуйста.

САМОЕ ЧИТАЕМОЕ

Сутки

Неделя

Месяц

Судьба предателя, угнавшего новейший МиГ-25 в Японию

↑ +69 👁 382k 📖 93 💬 1818

Расширен список госуслуг, которые можно получить не по прописке

↑ +15 👁 31,3k 📖 2 💬 50

Торговая сеть DNS распродала все карты РТХ 3060 по ошибочному названию продаж из-за технической ошибки

торговая сеть DNS распродала все карты к 17:00 до официального начала продаж «из-за технической ошибки»

 +18  21,3k  5  91

Серебряная пуля для кремлевского демона

 +85  18,5k  61  75

Куда же плывут облака? Подбиваем итоги опроса и даём аналитику

Мегапост

Ваш аккаунт	Разделы	Информация	Услуги
Войти	Публикации	Устройство сайта	Реклама
Регистрация	Новости	Для авторов	Тарифы
	Хабы	Для компаний	Контент
	Компании	Документы	Семинары
	Пользователи	Соглашение	Мегaproекты
	Песочница	Конфиденциальность	Мерч