

基于 julia 编程语言的前沿论文最新进展

2018.11.07 方建勇

提示：采用手机 safari 微软翻译技术

1. 时尚的通量建模

作者: [michael innes](#), [elliot saba](#), [keno fischer](#), [dhairyagandd](#), [marco concetto rudosso](#), [neethu mariya joy](#), [tejan karmali](#), [avik singh](#), [viral shah](#)

摘要: 近年来, 机器学习作为一门学科, 人们的兴趣达到了不可思议的高涨, 这在很大程度上是由于新理论的完美风暴、优越的工具、对其能力的重新兴趣。在本文中, 我们提出了一个名为 flux 的框架, 该框架展示了在 julia 编程语言的基础上进一步完善机器学习的核心理念, 如何能够产生一个简单、易于修改、性能。我们详细介绍了 flux 作为可微编程框架的基本原理, 并举例说明了在 flux 中实现的模型, 以显示许多语言和框架级别的功能, 这些功能有助于提高其易用性和高生产力, 显示内部编译器技术, 用于实现在 flux 的核心的加速和性能, 最后给出通量适合的更大的生态系统的概述。少

2018 年 10 月 31 日提交;最初宣布 2018 年 11 月。

2. julia 的非均衡速快速傅立叶变换 (nfft) 接口

作者: [michael schmiscke](#)

摘要:此报告描述了新添加到 `nfft3` 库的 **julia** 接口。我们解释了多维 `nfft` 算法和接口的基础知识。此外, 我们还详细介绍了不同的参数以及如何正确调整它们。

2018 年 10 月 23 日提交;最初宣布 2018 年 10 月。

3. 自动全面编译朱莉娅程序和 ml 模型到云 tpu

作者:[基诺·菲舍尔](#),[埃利奥特·萨巴](#)

摘要: google 的云 tpu 是一种很有前途的用于机器学习工作负载的新硬件体系结构。近年来, 他们推动了谷歌许多具有里程碑意义的机器学习成就。谷歌目前已经在他们的云平台上提供了 tpu 供一般使用, 截至最近, 它还被进一步开放, 允许非 tensorflow 前端使用。我们描述了一种方法和实现, 通过这个新的 api 和 google xla 编译器将 **julia** 程序的适当部分卸载到 tpu。我们的方法能够将 vgg19 模型的正向传递完全融合到一个 tpu 可执行文件中, 以便卸载到设备中。我们的方法很好地结合了 **julia** 代码上现有的基于编译器的自动微分技术, 因此我们还能够自动获得 vgg19 向后传递, 并同样将其卸载到 tpu。针对使用编译器的 tpu, 我们能够评估 vgg19 转发在 0.23 s 中的 100 张图像上的传递, 这些图像与 cpu 上原始模型所需的 52.4。我们的实现不到 1000 行的 **julia**, 没有对核心 **julia** 编译器或任何其他 **julia** 包进行特定的 tpu 更改。少

2018 年 10 月 23 日提交;最初宣布 2018 年 10 月。

4. ensmallen: 一个灵活的 c++ 库, 用于高效的函数优化

作者: [shikhar bhardwaj](#), [ryan r.curtin](#), [marcus edel](#), [yannis mentekidis](#), [conrad sanderson](#)

摘要: 我们提出了一个快速灵活的 c++ 库, 用于任意用户提供的函数的数学优化, 可应用于许多机器学习问题。支持多种类型的优化, 包括可微、可分离、约束和分类目标函数。该库提供了许多预先构建的优化器 (包括许多 sgd 和准牛顿优化器的变体), 以及用于实现新的优化器和目标函数的灵活框架。新优化器的实现只需要一种方法, 新的目标函数通常需要一个或两个 c++ 函数。这有助于快速实现和原型设计新的机器学习算法。由于使用了 c++ 模板元编程, ensmallen 能够支持提供快速运行时的编译器优化。实证比较表明, ensmallen 能够超越其他优化框架 (如 julia 和 s 密), 有时利润很大。该库在 bsd 许可证下分发, 可在生产环境中使用。少

2018 年 10 月 22 日提交;最初宣布 2018 年 10 月。

5. gpu 广播内核的动态自动分化

作者: [jarrett revles](#), [tim besard](#), [valentin churavy](#), [bjorn de sutter](#), [juan pablo vielma](#)

摘要: 我们展示了如何在更大的反向模式计算中使用前模自动微分 (ad), 以 gpu 友好的方式动态区分广播操作。我们的技术充分利用了广播 jacobian 固有的稀疏结构, 与纯粹的反模式方法不同, 这种 "混合模式" 方法不需要在广播操作的子图上进行反向传递, 从而无需进行几次用户创作的广播操作中的反向特定模式可编程性限制。最值得注意的是, 尽管存在依赖于数据的控制流, 但这种方法允许在原始代码中进行广播融合。我们讨论了一个实验, 在这个实验中, 我们的技术的 **julia** 实现优于纯反向模式 tensorflow 和 **julia** 实现, 通过 hm-lstm 单元更新计算中的广播操作进行区分.少

2018年10月24日提交;v1于2018年10月18日提交;**最初宣布** 2018年10月。

6. 不要展开伴随: 区分 ssa-for 形程序

作者:[michael innes](#)

摘要: 本文提出了基于源代码转换的反模式算法微分 (ad), 特别是现代编译器使用的静态单分配 (ssa) 形式。该方法可以支持控制流、嵌套、突变、递归、数据结构、高阶函数和其他语言构造, 并将输出提供给现有编译器以生成高效的差异化代码。我们的实现是朱莉娅语言的一个新的 ad 工具, 称为 zygote, 它提供了高

级动态语义, 同时透明地编译引擎盖下的伴随代码。我们讨论了这种方法对 `ad` 工具的可用性和性能的好处。少

2018年10月22日提交;v1于2018年10月18日提交;**最初宣布** 2018年10月。

7. 调整计算持久性同源包的性能

作者 :[alan hylton](#), [gregory Henselman-Petrusek](#), [janche sang](#), [robert short](#)

摘要: 近年来, 持久的同源性已成为一种有吸引力的数据分析方法。它从点云数据中捕获拓扑特征, 如连接的组件、孔和空隙, 并总结这些功能在过滤序列中出现和消失的方式。在这个项目中, 我们专注于提高 `eirene` 的性能, 这是一个持久同源的计算包。艾琳是一个 5000 行的开源软件库, 采用动态编程语言 `julia` 实现。我们使用 `julia` 分析工具来识别性能瓶颈, 并开发新的方法来管理这些瓶颈, 包括在多核核心硬件上并行化一些耗时的功能。实证结果表明, 性能有了很大的提高。少

2018年9月8日提交;**最初宣布** 2018年9月。

8. `aligator.jl`—环比不变一代朱莉娅套餐

作者 :[andandas humenberger](#), [maximian jaroschek](#), [laura kovács](#)

摘要: 我们描述了 `aligator.jl` 软件包, 用于自动生成具有嵌套条件的扩展 p 可解循环的丰富类的所有多项式不变量。`aligator.jl` 是用编程语言 **julia** 编写的, 是开源的。将程序循环转换为代数递归系统, 实现从符号计算到解决递归的技术, 推导出循环变量的闭形式解, 并通过变量推导出多项式不变性的理想基于 `gröbner` 计算的消除。少

2018 年 8 月 16 日提交;最初宣布 2018 年 8 月。

9. `d4m.jl` 中的数据库操作

作者:[lauren milechin](#), [vijay Gadepally](#), [jeremy Gadepally](#)

摘要: 数据分析管道中的每个步骤都很重要, 包括数据库接收和查询。`d4m-acumulo` 数据库连接器使分析人员能够使用 `matlab (r)/gnu 八度法` 快速、轻松地从 `apache 累积库罗` 进行访问和查询。`d4m.jl` 是 `d4m` 的 **julia** 实现, 它为 **julia** 社区提供了最初 `d4m` 实现的大部分功能。在本工作中, 我们扩展了 `d4m.jl`, 以包括 `matlab (r)/gnu octave` 实现提供的许多相同的数据库功能。在这里, 我们将描述 `d4m.jl` 数据库连接器, 演示如何使用它, 并显示它具有类似或更好的性能, 在 `matlab (r)/gnu 八度法` 中的原始实现。少

2018 年 8 月 13 日提交;最初宣布 2018 年 8 月。

10. 语言相关性与编程语言

作者:陈家豪

摘要: 几十年来, 编程语言的使用可以结对和减少。我们研究统计计算中常见的拆分应用程序-组合模式, 并考虑它在 matlab 和 apl 等语言中的调用或实现与 r/dplyr 有何不同。拼写上的差异说明了语言相对论的概念如何以类似于人类语言的方式应用于编程语言。最后, 我们讨论了 julia 如何通过成为高性能但通用的动态语言, 允许其用户表达不同的抽象来满足个人偏好。少

2018 年 8 月 12 日提交;最初宣布 2018 年 8 月。

11. 朱莉娅快速灵活的功能调度

作者:杰夫·贝赞森,杰克·博莱夫斯基,陈嘉豪

摘要: 技术计算是编程语言需要解决的一个具有挑战性的应用领域。该地区的专门语言数量异常多 (如 matlab、r), 以及常见软件堆栈的复杂性 (通常涉及多种语言和自定义代码生成器) 就证明了这一点。我们认为, 这最终是由于该领域的关键特征: 高度复杂的运算符、对广泛的代码专业化以实现性能的需求, 以及对允许进行高效实验的高级编程样式的渴望。julia 语言试图通过允许程序员使用动态多调度而不是参数类型来表示复杂的多态行为, 从而为这种编程提供一个更有效的结构。这种范式所允许的扩展

和重用形式已被证明对技术计算很有价值。我们报告了这种方法如何使领域专家能够表达有用的抽象, 同时为高级技术代码提供了实现更好性能的自然路径。少

2018 年 8 月 9 日提交;最初宣布 2018 年 8 月。

12. 数据存储: 可复制数据科学的可重复数据设置

作者:[林登·怀特](#),[罗伯托·托涅里](#), [刘伟](#),[穆罕默德·本纳蒙](#)

摘要: 我们提出了一个 `julia` 包, 用于静态数据集的重现性处理, 以提高数据和计算科学中使用的脚本的可重复性。它用于自动化运行软件的数据设置部分, 伴随着纸张复制结果。此步骤通常是手动完成的, 它会花费时间并允许混淆。此功能对于需要数据才能正常工作的其他包(例如基于训练的机器学习模型)也很有用。`dat` 参照 `jl` 通过自动管理依赖关系简化了研究软件的扩展, 并使运行其他作者的代码变得更加容易, 从而提高了数据科学研究的重现性。少

2018 年 8 月 3 日提交;最初宣布 2018 年 8 月。

13. 计算机视觉和机器学习中几个问题的选定算法微分工具的基准

作者:[filipš rajer](#), [zuzana kkelova](#), [andrew fitzgibbon](#)

摘要: 算法微分 (ad) 允许精确计算导数只给目标函数的一个实现。虽然有许多 ad 工具可用, 但正确和有效地实现 ad 方法并不简单。现有的工具往往太不一样, 无法使用通用的测试套件。本文比较了 15 种计算导数的方法, 包括 11 种自动微分工具, 实现各种方法, 并以各种语言 (c++、f #、matlab、**julia** 和 python) 编写, 这两种符号化差异化工具, 有限的差异, 和手工派生的计算。我们从计算机视觉和机器学习三个客观功能来研究。这些目标在很大程度上很简单, 也就是说不涉及迭代循环, 并且条件语句封装在函数中, 如 `{\tt abs}` 或 `{\tt logsumexp}`。然而, 有效地处理这种 "简单" 的目标函数对于算法微分的成功是很重要的, 因为计算机视觉和机器学习中的许多问题都是这种形式的。当然, 我们的结果取决于程序员的技能, 以及对工具的熟悉程度。然而, 我们认为, 本文提出了一个重要的数据点: 一个熟练的程序员花了大约一个星期的时间来每个工具产生的时间, 我们提出的。我们已将我们的实现作为开源提供, 以允许社区复制和更新这些基准。少

2018 年 7 月 26 日提交;最初宣布 2018 年 7 月。

14. tiramisu 的技术报告: 一种从 dsl 编译器中隐藏硬件复杂性的三层抽象

作者: 利雅得巴格达迪, [jessica ray](#), [malek ben romdhane](#), [Emanuele del sozzo](#), [patricia suriana](#), [shoaibkamil](#), [saman amarasinghe](#)

摘要: 高性能 dsl 开发人员努力利用现代硬件。dsl 编译器必须先构建自己的复杂中端, 然后才能针对公共后端 (如 llvm), 该后端仅使用 simd 指令处理单个指令流。我们介绍 tiramisu, 这是一种常见的中端, 它可以为现代处理器和加速器 (如多核、gpu、fpga 和分布式集群) 生成高效代码。tiramisu 引入了一种新的三级 ir, 它将算法、算法的执行方式以及中间数据的存储位置分开。这种分离简化了优化, 并使从同一算法定位多个硬件架构变得更容易。因此, dsl 编译器可以大大降低复杂性, 而不会损失性能, 同时立即针对多个硬件或硬件组合, 如同时同时使用 cpu 和 gpu 的分布式节点。我们通过为 halide 和 julia 编译器创建一个新的中间端来评估 tiramisu。我们展示了 tiramisu 扩展了 halide 和 julia 的许多新功能, 包括能够: 表示新算法 (如递归筛选器和非矩形迭代空间), 执行新的复杂循环巢转换 (如波前并行化、环路转换和环路融合), 并为更多架构 (如分布式集群、多核、gpu 和 fpga 的组合) 生成高效代码。最后, 我们证明 tiramisu 可以生成非常有效的代码, 与高度优化的英特尔 mkl gemm (广义矩阵乘法) 实现相匹配, 我们还显示了在 halide 中达到 4x 和在 julia 达到 16x 的速度: 由提拉米苏启用的优化。少

2018 年 5 月 28 日提交;v1 于 2018 年 2 月 28 日提交;**最初宣布** 2018 年 3 月。

15. 通过佩塔斯卡尔的贝叶斯推理对可见宇宙进行编目

作者: [jeffrey regier](#), [kiran Pamnany](#), [keno fischer](#), [andreas noack](#), [maximian lam](#), [jarrett revles](#), [steve howard](#), [ryan giordano](#), [david schlegel](#), [jon mcauliffe](#), [rolin thomas](#), [prabhat](#)

摘要: 从广域成像测量中获得的天文目录是了解宇宙的重要工具。我们使用 celeste 构建了一个来自 55 tb 成像数据的天文目录, 这是一种完全用高生产率编程语言 **julia** 编写的贝叶斯变分推理代码。celeste 在 cori 二期超级计算机的 650, 000 个英特尔至强融核内核上使用了 130 多万个线程, 达到了 1.54 dp plops 定制的峰值速率。celeste 能够共同优化 188m 恒星和星系的参数, 在 8192 颗恒星和星系中加载和处理 178 tb 在 14.6 分钟内为了实现这一点, celeste 利用多个级别 (群集、节点和线程) 的并行性, 并通过 cori 的 burst 缓冲区加速 ii/o。**julia** 的本机性能使 celeste 能够使用高级构造, 而无需使用手写或生成的低级代码 (csc + +/fortran), 同时实现 petascale 性能。少

2018 年 1 月 30 日提交;最初宣布 2018 年 1 月。

16. zoopt: 用于无导数优化的工具箱

作者: [刘玉仁](#), [胡一琪](#), [洪谦](#), [杨宇](#), [赵谦](#)

摘要: 最近在无导数优化方面取得的进展使得可以有效地逼近复杂函数的全局最优解, 例如具有许多局部优化函数、非微函数和非连续函数的函数。本文介绍了 <https://github.com/eyounx/ZOOpt> () 工具箱, 该工具箱提供高

效的无派生求解器, 易于使用。zoopt 提供了用于单线程优化的 python 包, 以及在 python 的 **julia** 语言帮助下提供的轻量级分布式版本。zoopt 工具箱特别关注机器学习中的优化问题, 解决高维、嘈杂和大规模的问题。在现实世界的机器学习任务中, 工具箱正朝着随时可用的工具进行维护。少

2018 年 2 月 6 日提交;v1 于 2017 年 12 月 31 日提交;**最初宣布** 2018 年 1 月。

17. 有效的可扩展编程: 释放 gpu 朱莉娅

作者:[tim besard](#), [christophe foket](#), [bjorn de sutter](#)

摘要: gpu 和其他加速器是加速计算密集型、可并行化应用程序的常用设备。然而, 对这些设备进行编程是一项艰巨的任务。编写高效的设备代码具有挑战性, 通常是用低级编程语言完成的。高级语言很少得到支持, 或者与高级语言生态系统的其他部分不集成。为了克服这一问题, 我们建议使用编译器基础结构, 以便有效地将对新硬件或环境的支持添加到现有编程语言中。我们通过在 **julia** 编程语言中添加对 nvidia gpu 的支持来评估我们的方法。通过与现有编译器集成, 我们大大降低了实现和维护新编译器的成本, 并促进了现有应用程序代码的重用。此外, 使用高级 **julia** 编程语言可以为 gpu 编程提供新的动态方法。这大大提高了程

序员的工作效率, 同时保持了类似于官方 `nvidia cuda` 工具包的应用程序性能。少

2017 年 12 月 8 日提交;最初宣布 2017 年 12 月。

18. 同人运动连续性. jl: 在朱莉娅同体延续的包

作者:[paul bre 搜查](#), [sascha timme](#)

摘要: 我们提出了朱莉娅包 `homotopy` 认识. jl, 它提供了一个算法框架, 通过数值同伦延续求解多项式系统。我们介绍了包的基本功能, 并在一个示例中演示了该软件。我们鼓励我们选择 `julia`, 以及它的功能如何使我们能够在可用性、模块化和性能方面改进现有的软件包。此外, 我们还将 `homotopy` 合务必. jl 的性能与现有的贝尔蒂尼和 `phcpack` 包进行了比较。少

2018 年 5 月 30 日提交;v1 于 2017 年 11 月 28 日提交;最初宣布 2017 年 11 月。

19. 链状复合物的几何计算: 朱莉娅封装的设计与特点

作者:[francesco furiani](#), [giulio martella](#), [alberto paoluzzi](#)

摘要: 带有链复合物的几何计算允许计算由空间分解为单元复合体所产生的整个线性空间链和 (共同) 边界算子。空间分解使用 `lar` (线性代数表示) 存储和处理, 即使用稀疏整数数组, 并允许使

用非常通用类型的单元格，甚至是非凸和内部孔。本文讨论了该方法的特点和优点，并描述了一个软件包的目标和实现，旨在提供简单和有效的计算支持几何计算的任何类型的网格，使用线性具有稀疏矩阵的代数工具。图书馆是用朱莉娅语编写的，朱莉娅是一种用于科学计算的高效、并行的新型语言。该软件，这是被移植到混合架构（cpu + gpu）上一代，尚未开发。少

2017年11月6日提交;v1于2017年10月21日提交;最初宣布2017年10月。

20. 静态图形挑战: 子图同构

作者: [siddharth samsi](#), [vijay Gadepally](#), [michael hurley](#), [michael jones](#), [edwardkao](#), [sanjeev mohindra](#), [paul monticciolo](#), [albertreuther](#), [steven smith](#), [williamsong](#), [diane staheli](#), [jeremy kepner](#)

摘要: 图形分析系统的兴起产生了测量和比较这些系统能力的方法的需要。图形分析存在独特的可伸缩性困难。几十年来，机器学习、高性能计算和可视化分析社区一直在与这些困难作斗争，并开发了应对这些社区前进挑战的方法。拟议的子图同构图挑战利用了机器学习、高性能计算和可视化分析中以前的挑战，创造了一个反映许多真实世界图形分析处理系统的图形挑战。子图同构图挑战是一个整体规范，具有多个集成内核，可以一起运行，也可以独立运行。每个内核在数学上都有很好的定义，可以在任何编程环境中实现。子图同构适用于以顶点为中心的实现和基于

阵列的实现（例如，使用 graphblas.org 标准）。计算非常简单，可以根据简单的计算硬件模型进行性能预测。周围的内核为每个内核提供了上下文，允许对每个内核的输入和输出进行严格定义。此外，由于所提出的图形挑战在问题大小和硬件上都是可扩展的，因此可以用来测量和定量比较当今和未来的各种系统。实现了 `c++`、`python`、`python` 中的串行实现、`熊猫`、`matlab`、`octave` 和 `julia`，并测量了它们的单线程性能。规格、数据和软件可在 challenge.org 上公开查阅。少

2017 年 8 月 22 日提交;最初宣布 2017 年 8 月。

21. 西奥海: 向光明行军的理论

作者:[mark a. stalzer](#), [chao ju](#)

摘要: 在辐射偶极子天线的远场中，有足够的信息可以重新发现麦克斯韦方程和光的波动方程，包括光速 C 。西奥西是一个朱莉娅程序，大约一秒钟就能做到这一点，关键的洞察是，理论的紧凑性推动了搜索。该程序是科学方法的计算体现：观察、考虑候选理论和验证。少

2017 年 8 月 14 日提交;最初宣布 2017 年 8 月。

22. 增强器: 用于机器学习的图像增强库

作者:[marcus d. Stocker](#), [christof stocker](#), [anderas holzinger](#)

文摘: 基于现有观测的人工数据生成（称为数据扩容）是机器学习中用于提高模型精度、通用性和控制超拟合的一种技术。奥古斯丁是一个软件包，提供了 python 和朱莉娅版本，它提供了一个高级 api，用于扩展图像数据使用随机的，基于管道的方法，有效地允许在运行时分发增强型图像。增强器提供了大多数标准增强实践的方法以及几个高级功能（如标签保留、随机弹性扭曲），并为机器学习中使用的典型增强任务提供了许多辅助功能。

少

2017 年 8 月 11 日提交;最初宣布 2017 年 8 月。

23. d4m 3.0: 扩展的数据库和语言功能

作者: [lauren milechin](#), [vijay Gadepally](#), [Siddharth samsi](#), [jeremy Gadepally](#), [亚历山大 chen](#), [dylan hagchison](#)

摘要: d4m 工具是为满足当今的许多数据需求而开发的。数百名研究人员使用此工具对非结构化数据执行复杂的分析。在过去几年中, d4m 工具箱不断发展, 以支持与包括科学和技术数据库在内的各种新数据库引擎的连接。d4m-gragulo 提供了在 apache 累积数据库中进行图形分析的能力。最后, 现在还提供了使用 julia 编程语言的实现。在本文中, 我们介绍了 d4m 工具箱的一些最新新增功能以及即将发布的 d4m 3.0 版本。我们通过基准测试和缩放结果表明, 我们可以使用 d4m-s 银连接器实现快速的

科学和数据数据库接收, 使用 graphulo 可以在可以限制内存的比例上启用图形算法, 并且 d4m 的 **Graphulo** 实现了或超过现有 matlab (r) 实施的性能。少

2017 年 8 月 9 日提交;最初宣布 2017 年 8 月。

24. 朱莉娅的准区域算法实现与仿真

作者:[tyler m. masthay](#) , [saverio perugini](#)

摘要: 我们在 **julia** 编程语言中提出了一个完整的并行算法——一种并行求解微分方程的集成技术——对于一个完全通用的、一阶的、初始值的问题。我们的实现接受粗积分器和细分积分器作为功能参数。在实验中, 我们使用欧拉的方法和另一种 runge–kutta 集成技术作为积分器。我们还提出了一个模拟算法的目的, 为教育学的目的。少

2017 年 6 月 26 日提交;最初宣布 2017 年 6 月。

25. 试论用紧凑性对句子进行的枚举

作者:[马克·斯托尔泽](#)

摘要: 提出了一个朱莉娅元程序, 发现紧凑的理论从数据, 如果存在。它写了朱莉娅的候选理论, 然后验证: 抛出坏的理论 and 保持

好的理论。压缩度是用一个度量来度量的: 例如时空导数的数量。

该算法适用于各种组合问题, 紧凑度有助于缩小搜索空间。少

2017 年 6 月 15 日提交;最初宣布 2017 年 6 月。

26. nemo\ hecke: 朱莉娅编程语言的计算机代数和数字理论包

作者:[克劳斯·菲克](#),[威廉·哈特](#),[汤米·霍夫曼](#),[弗雷德里克·约翰森](#)

文摘: 我们介绍了两个新的软件包, nemo 和 hecke, 用朱莉娅编程语言编写的计算机代数和数论。我们证明了高性能泛型算法可以在 julia 中实现, 而不需要诉诸低级 c 实现。对于专门的算法, 我们使用 julia 高效的本机 c 接口来包装现有的 csc++ 库, 如弗林特、arb、antic 和奇异。我们给出了如何使用 hecke 和 nemo 的例子, 并讨论了我们为提供高性能基本算法而实现的一些算法。少

2017 年 5 月 17 日提交;最初宣布 2017 年 5 月。

27.d4m 3. 0

作者:[lauren milechin](#),[亚历山大 chen](#) , [vijay Gadepally](#), [dylan hagchison](#), [Siddharth samsi](#), [jeremy Gadepally](#)

摘要: 数百名研究人员使用 d4m 工具对非结构化数据执行复杂的分析。在过去几年中, d4m 工具箱已经发展到支持与各种数据库引擎的连接、apacacacacarculo 数据库中的图形分析以及使

用 **Accumulo** 编程语言的实现。在本文中, 我们介绍了 d4m 工具箱的一些最新新增功能以及即将发布的 d4m 3.0 版本。少

2017 年 1 月 18 日提交;最初宣布 2017 年 2 月。

28. 帮助朱莉娅理解你的 matlab–octave 代码

作者:[vardan Andriasyan](#), [yauhen yakimovich](#), [artur yakimovich](#)

摘要: matlabxioctave 中的科学遗留代码与研究工作流程的现代化不兼容, 在整个学术界是非常丰富的。用 matlabxyoctave 编写的非矢量化代码的性能是一个主要的负担。一种用于技术计算的新编程语言 **julia** 有望解决这些问题。尽管 **julia** 语法类似于 matlab–octave, 但将代码移植到 **julia** 对研究人员来说可能会很麻烦。在这里, 我们提出 matlabcompt. jl–一个库, 旨在简化转换您的 matlab/octave 代码给朱莉娅。我们使用一个简单的图像分析用例表明, matlab/octave 代码可以很容易地移植到高性能朱莉娅使用 matlabcomt. jl。少

2017 年 1 月 6 日提交;最初宣布 2017 年 1 月。

29. 用于在 julia 实现高效多尺度接触计算的加速卷积

作者:[亚历山大·阿米尼](#), [伯瑟德·霍恩](#), [艾伦·埃德尔曼](#)

摘要: 长期以来, 卷积一直被认为是应用数学、物理和工程的基础。他们的数学优雅允许常见的任务, 如数值微分计算有效的大型数据集。在实时应用 (如机器视觉) 中, 卷积的高效计算对于人工智能至关重要, 因为在这种情况下, 卷积必须以每秒几十到数百千字节的速度连续有效地计算。在本文中, 我们探讨了卷积如何在基本机器视觉应用中的应用。我们提出了一个高性能计算语言 `julia` 加速 `n` 维卷积包, 并演示了它在解决机器视觉接触时间问题方面的有效性。根据综合生成的视频对结果进行测量, 并根据其来自地面真相的平均平方误差进行定量评估。我们实现了计算时间的数量级以上, 并为可比较的机器视觉应用分配了内存。所有代码都打包并集成到正式的 `julia` 包管理器中, 以便在各种其他方案中使用。少

2016 年 12 月 28 日提交;最初宣布 2016 年 12 月。

30. hpat: 易于使用的脚本的高性能分析

作者: [ehsan totoni](#), [todd a.anderson](#), [tatiana shpeisman](#)

摘要: 大数据分析要求在大规模集群上同时实现高程序员工作效率和高性能。但是, 当前的大数据分析框架 (例如 `apache spark`) 具有令人望而却步的运行时开销, 因为它们是基于库的。我们引入了一种新的自动并行化编译器方法, 该方法利用数据分析域的特征 (如 `map/减少` 并行模式), 与以前的自动并行化方法不同,

具有鲁棒性。使用此方法, 我们构建高性能分析工具包 (hpat), 它自动并行并行高级脚本 (**julia**) 程序, 生成高效的 mpicl ++ 代码, 并提供弹性。此外, 它还为脚本程序提供自动优化, 例如阵列操作的融合。因此, hpat 比 cori 超级计算机上的 spark 快 36 倍至 2033x, 在 amazon aws 上的速度快 20 倍至 256 倍。少

2017 年 4 月 10 日提交;v1 于 2016 年 11 月 15 日提交;**最初宣布** 2016 年 11 月。

31. 通过可缩放贝叶斯推理学习可见宇宙的天文目录

作者 : [jeffrey regier](#), [kiran Pamnany](#), [ryan giordano](#), [rolin thomas](#), [david schlegel](#), [jon mcauliffe](#), [prabhat](#)

文摘: celeste 是一个输入获得最先进科学成果的目录的程序。到目前为止, celeste 已扩展到最多数百兆字节的天文图像: 贝叶斯后部推理是众所周知的要求计算。在本文中, 我们报告了一个可扩展的并行版本的 celeste, 适合从现代大型天文数据集学习目录。我们的算法创新包括贝叶斯后验推理的快速数值优化例程和将天文优化问题分解为子问题的统计高效方案。我们的可扩展实现完全是用 **julia** 编写的, julia 是一种新的高级动态编程语言, 专为科学和数值计算而设计。我们使用 **julia** 的高级结构来实现共享和分布式内存并行性, 并在 nersc cori 超级计算机上演示有效的负载平衡和高效扩展, 最多可扩展 8192 至多直到。少

2016 年 11 月 10 日提交;最初宣布 2016 年 11 月。

32. 使用 gpu 生成分形艺术

作者: [威尔, 你在吧梅菲尔德](#) [贾斯汀 c. eiland](#), [taylor。j. huyra](#), [matt。c. paulsen](#), [bryant。m. wyatt](#)

摘要: 分形图像生成算法表现出极高的并行性。利用通用图形处理单元 (gpu) 编程来实现 **julia** 函数集的逃逸时间算法, 并行方法生成具有视觉吸引力的分形图像的速度远远快于传统方法。利用这种计算方法实现了极高的速度, 可以实时生成和显示图像。对该算法的顺序实现和并行实现进行了比较。作者创建的一个应用程序演示了如何提高速度来创建分形的动态成像, 用户可以在其中探索与给定函数的 mandelbrot 集合相对应的参数值的路径。给出了通过以交互方式体验分形和快速、全面地采样参数空间的能力所获得的艺术和数学见解的例子。少

2016 年 11 月 8 日提交;最初宣布 2016 年 11 月。

33. 在嘈杂的环境中实现可靠的基准测试

作者: [陈家浩](#), [贾勒特狂欢](#)

摘要: 我们提出了一种基准策略, 该策略在存在计时器错误、os 抖动和其他环境波动的情况下是稳健的, 并且对计时测量生成的高度不理想的统计数据不敏感。我们构建了一个模型, 解释了这

些强烈的非理想统计数据是如何从环境波动中产生的, 同时也证明了我们提出的战略是合理的。我们在基准工具朱莉娅软件包中实施了这一策略, 将其用于开发 **julia** 语言及其生态系统的生产持续集成 (ci) 管道。少

2016 年 8 月 15 日提交;最初宣布 2016 年 8 月。

34. 动态分布式维数据模型的朱莉娅实现

作者: [亚历山大·陈](#), [alan edelman](#), [jeremy kepner](#), [vijayKepner](#), [dylan hagchison](#)

摘要: **julia** 是一种用于编写易于实现和高性能运行的数据分析程序的新语言。同样, 动态分布式维度数据模型 (d4m) 旨在阐明数据分析操作, 同时保持强大的性能。d4m 通过关联数组上可组合的统一数据模型来实现这些目标。在这项工作中, 我们在 **julia** 中介绍了 d4m 的实现, 并描述了它如何支持和促进数据分析。与最初的 **matlab** 实现相比, 几个实验展示了我们新的 **julia** 版本中的可扩展性能。少

2016 年 8 月 13 日提交;最初宣布 2016 年 8 月。

35. 朱莉娅的前向模式自动区分

作者: [jarrett revles](#), [miles luin](#), [theodore papamarkou](#)

摘要: 我们提供向前 diff, 一个 **julia** 包向前模式自动区分 (ad) 的特点是性能竞争的低级语言, 如 c++。与最近在其他流行的高级语言 (如 python 和 matlab) 中开发的 ad 工具不同, 返众 diff 利用实时 (jit) 编译来透明地重新编译不了解 ad 的用户代码, 从而实现对高阶的高效支持使用自定义数字类型 (包括复数) 进行区分和区分。对于梯度和雅可比计算, 转发器 diff 提供了矢量转发模式的变体, 避免了昂贵的堆分配, 并比传统的矢量模式更好地利用内存带宽。在数值实验中, 我们证明了对于非小尺寸, 向前 diff 的梯度计算可以比基于 python 的自动图封装的反向模式实现更快。我们还说明了如何在 jump 中有效地使用转发器。根据我们的使用统计数据, github 上的 41 个独特存储库依赖于返格 diff, 用户来自不同领域, 如天文学、优化、有限元分析和统计。本文档是一个扩展摘要, 已被接受在 ad2016 第七届算法差异化国际会议上发表。少

2016 年 7 月 26 日提交;最初宣布 2016 年 7 月。

36. 朱莉娅的广义采样

作者: [robert dahl jacobson](#), [morten nielsen](#), [morten grud rasmussen](#)

摘要: 广义采样是从样本中获取不同基和帧信号重建的数值稳定框架。在本文中, 我们将介绍一个精心记录的工具箱, 用于在 **julia** 中执行广义采样。**julia** 是一种以性能为重点的技术计算新语言,

非常适合处理广义采样中经常遇到的大尺寸问题。工具箱为傅立叶基和小波的设置提供了专门的解决方案。将工具箱的性能与 matlab 中现有的广义采样实现进行了比较。少

2016 年 11 月 23 日提交;v1 于 2016 年 7 月 14 日提交;**最初宣布** 2016 年 7 月。

37.jinv--一个灵活的 julia 包 pde 参数估计

作者:[lars ruthotto](#), [elan treister](#), [eldad haber](#)

摘要: 从噪声和间接测量中估计偏微分方程 (pde) 的参数通常需要求解不恰当的逆问题。这些所谓的参数估计或逆介质问题出现在各种应用中, 如地球物理、医学成像和无损检测。它们的解在计算上是非常强烈的, 因为在参数的重建足够准确之前, 需要多次求解底层的偏微分方程。通常情况下, 当更多的测量可用时, 计算需求会显著增长, 这将对反演算法构成严峻挑战, 因为测量设备变得更加强大。本文提出了一种灵活的框架和开源软件 jinv, 它提供了并行算法来解决具有多种测量值的参数估计问题。jinv 是用表达式编程语言 julia 编写的, 它是便携式的, 易于理解和扩展, 跨平台测试, 并有很好的文档记录。它提供了新的并行化方案, 利用许多参数估计问题的固有结构, 可用于解决多物理场反演问题, 并通过地球物理成像驱动数值实验进行了验证。少

2016 年 12 月 14 日提交;v1 于 2016 年 6 月 23 日提交;最初宣布 2016 年 6 月。

38. 朱莉娅的高级 gpu 编程

作者:tim besard, pieter verstraete, bjorn de sutter

摘要: gpu 是加速科学计算的常用设备。但是, 由于 gpu 代码通常是用低级语言编写的, 因此它打破了科学程序员流行的高级语言的抽象。为了克服这一问题, 我们在高级朱莉娅编程语言中提出了一个 cuda gpu 编程框架。此框架为 gpu 执行编译 julia 源代码, 并使用现代代码生成技术处理必要的低级交互, 以避免运行时开销。在对包含图像处理领域的跟踪变换的案例研究中评估框架及其 api 时, 我们发现对性能的影响很小, 同时大大提高了程序员的工作效率。julia 语言的元编程功能被证明是实现这一目标的无价的。我们的框架显著提高了 gpu 的可用性, 使各种程序员都可以访问 gpu。它是可用的免费和开源软件授权下麻省理工学院许可证。少

2016 年 4 月 12 日提交;最初宣布 2016 年 4 月。

39. pagerank 管道基准: 关于大数据平台整体系统基准的建议

作者 :patrick dreher, chansup byun, chris hill, vijayGadepally, bradley kuszmaul, jeremy Gadepally

摘要: 大数据系统的兴起产生了对衡量和比较这些系统能力的基准的需求。大数据基准带来了独特的可扩展性挑战。几十年来, 超级计算界一直在努力应对这些挑战, 并开发了创建严格可扩展基准的方法 (例如, hpc 挑战)。拟议的 pagerank 管道基准利用超级计算基准方法来创建一个可扩展的基准, 反映许多真实世界的大数据处理系统。pagerank 管道基准基于现有的可扩展基准 (graph500、sort 和 paragrak), 创建一个具有多个集成内核的整体基准, 这些内核可以一起运行或独立运行。每个内核在数学上都有很好的定义, 可以在任何编程环境中实现。年龄排名的线性代数性质使其非常适合使用 graphblas 标准来实现。计算非常简单, 可以根据简单的计算硬件模型进行性能预测。周围的内核为每个内核提供了上下文, 允许对每个内核的输入和输出进行严格定义。此外, 由于提出的 paragrak 管道基准在问题大小和硬件上都具有可扩展性, 因此它可用于测量和定量比较当今和未来的各种系统。实现了 c++、python、python 中的串行实现、熊猫、matlab、octave 和 **julia**, 并测量了它们的单线程性能。少

2016 年 6 月 3 日提交;v1 于 2016 年 3 月 6 日提交;**最初宣布** 2016 年 3 月。

40. 以重心为中心的细化图谱

作者: [奥利弗·克尼勒](#)

摘要: 给定一个有限的简单图 g , 让 g' 成为它的以巴以为中心的细化: 它是一个图, 其中的顶点是 g 的完整子图, 其中两个这样的子图连接在另一个子图中。如果 $L(0)=0 < L(1) \leq L(2) \leq \dots \leq L(n)$ 是 g 的 laplacian 的特征值, 将光谱函数 $f(x)$ 定义为间隔 $[0, 1]$ 上的函数 $f(x) = L([nx])$, 其中 $[r]$ 是给出最大整数小于或等于 r 的平面函数。图 g' 已知与欧拉特征 $\chi(g') = \chi(g)$ 和暗淡 $(g) \leq \text{暗淡}(g)$ 同质为 g 。让 $g(m)$ 成为 $G=G(0)$ 以重心为中心的一系列改进。我们证明, 对于任何有限的简单图 g , 连续细化的光谱函数 $f(g(m))$ 在紧凑型子集 $(0, 1)$ 上均匀地收敛到 m 到无穷大, 并以指数速度向通用限特征值分布函数 f 。分别取决于小块数, 分别取决于 g 最大完整子图的维度 d , 而不是起始图 g 上的尺寸 d 。在 d 何 ≤ 1 的情况下, 我们处理没有三角形的图形, 极限分布是平滑函数 $f(x) = 4 \sin^2(\pi x^2)$ 。这与二次映射 $t(z) = 4z - z^2$ 的 **julia** 集合有关, 因为 Laplacians 满足这样的重新规范化递归, 它具有一维的 **julia** 集 $[0, 4]$ 和 f 满足 $T(F(k/n)) = F(2k/n)$ 。在 d 供需 1 壳中的光谱密度则是杏仁分布, 是 **朱莉娅** 集上的平衡测量。在较高的维度中, 极限函数 f 仍然是未知的, f' 似乎有一个离散或奇异的分量。少

2015 年 8 月 9 日提交;最初宣布 2015 年 8 月。

41. 一步随机过程仿真软件包

作者: [e. g. eferina](#), [a. v. korkova](#), [m.n. gevorkyan](#), [d. s. kulyabov](#), [i. a. sevastyanov](#)

摘要: 背景。假定在数学模型中引入随机使其更加充分。但几乎没有协调（取决于系统结构）的方法随机引入确定性模型。作者改进了进一步过程类的随机模型构建方法，并通过种群动力学模型进行了说明。选择种群动态进行研究是因为它的确定性模型得到了充分的探索，可以将结果与已经知道的结果进行比较。目的。为了尽可能地优化模型的创建，一些常规操作应该是自动化的。在这种情况下，模型方程的绘制过程可以在计算机代数系统中进行算法化和实现。在此基础上，可以获得一套数值实验程序。方法。计算机代数系统公理用于分析计算的实现。为了进行数值实验，使用了 fortran 和 **julia** 语言。将随机微分方程的 **rung-kutta** 方法作为数值方法。结果。构建了建立随机一步过程模型的程序组合。捕食者-猎物种群动态系统说明了它的应用。结论。计算机代数系统对于数学模型设计和分析中的快速原型设计非常方便。少

2015 年 3 月 25 日提交;最初宣布 2015 年 3 月。

42. 朱莉娅：一种新的数值计算方法

作者: [杰夫·贝赞森](#), [alan edelman](#), [stefan karpinski](#), [viral b. shah](#)

摘要: 朱莉娅将经常遥远的文化联系在一起,结合了计算机科学和计算科学不同领域的专业知识,创造了一种新的数值计算方法。

朱莉娅的设计既简单又快捷。朱莉娅质疑那些被数值计算实践者普遍认为是 "自然法则" 的概念: 1. 高级动态程序必须是缓慢的。2. 必须用一种语言进行原型, 然后用另一种语言重写速度或部署, 并且 3. 系统中有部分是供程序员使用的, 而其他部分最好是不被触及, 因为它们是由专家构建的。我们介绍了朱莉娅编程语言及其设计——专业化和抽象之间的舞蹈。专业化允许自定义处理。多元调度是计算机科学的一种技术, 它针对合适的环境选择合适的算法。抽象, 什么是好的计算真正的, 认识到什么仍然是相同的差异被剥离后。数学中的抽象是通过计算机科学、通用编程的另一种技术作为代码捕获的。julia 证明, 一个人可以在不牺牲人的方便的情况下具有机器性能。少

2015 年 7 月 19 日提交;v1 于 2014 年 11 月 6 日提交;最初宣布 2014 年 11 月。

43. 使用多调度的阵列运算符: 一种动态语言阵列实现的设计方法

作者:杰夫·贝赞森,陈嘉豪, stefan karpinski, viral shah, alan edelman

摘要: 数组是一种丰富而基本的数据类型, 它们往往被内置到一种语言中, 可以是在编译器中, 也可以是在大型低级库中。而是在用户级别定义此功能, 为语言设计器未设想的应用程序域提供了更大的灵活性。只有少数语言 (如 c++ 和 haskell) 提供了定义 n-维数组, 但这些系统依赖于编译时抽象, 牺牲了一定的灵活性。

相反, 动态语言使用户可以很容易地定义他们可能想要的任何行为, 但可能会牺牲性能。作为 **julia** 语言项目的一部分, 我们开发了一种在灵活性和编译时分析之间进行权衡的方法。我们使用的核心抽象是多调度。我们已经开始相信, 虽然多调度在大多数类型的编程中并不是特别流行, 但技术计算是其杀手级的应用。通过使用多方法签名表示数组索引等关键函数, 可以获得令人惊讶的行为范围, 其方式既相对容易编写, 又易于进行编译器分析。这些方法提供的关注的紧凑分解使用户定义的类型更容易与标准库中的类型保持一致。少

2014 年 7 月 14 日提交;最初宣布 2014 年 7 月。

44. 新的朱莉娅和曼德尔布罗特设置在 **jungck** 石川迭特雷斯

作者:su 曼·乔希, [yashwant singh chauhan](#) 博士, [ashish negi](#) 博士

文摘: 分形的产生和多项式动力学的研究是当今新兴的、有趣的研究领域之一。本文介绍了 $n \geq 2$ 多项式 $z^n - z + c = 0$ 的动力学, 并应用 shigke 石川迭代生成了新的相对高级曼德尔布罗特集和相对高级朱莉娅集。为了通过 **jungck** 类型的迭代方案来求解这个函数, 我们以 $sz = tz$ 的形式编写它, 其中函数 t, s 被定义为 $tz = z^n + c$ 和 $sz = z$ 。在文献中, 将石川的 **jungck** 迭

代应用于多项式, 只得到了数学解释, 但在本文中, 我们生成了相对曼德尔布罗特集合和相对**朱莉娅**集合。少

2014 年 4 月 3 日提交;最初宣布 2014 年 4 月。

45. java 字节码的非终止性分析

作者:[Étienne payet](#), [fred mesnard](#) , [fausto spoto](#)

摘要: 我们引入了一个完全自动化的静态分析, 它以顺序 java 字节码程序 p 为输入, 并试图证明 p 的存在无限执行。该技术包括将 p 编译为约束逻辑程序 p_clp 和证明 p_clp 的非终止;当 p 由精确编译为约束的指令组成时, p_clp 的非终止就意味着 p 的不终止。我们的方法可以处理方法调用;据我们所知, 这是 java 字节码第一个能够证明存在无限递归的静态方法。我们已经在**朱莉娅**分析仪中实现了我们的技术。我们已经将**julia**在113个程序中的结果与 `aprove` 和 `invel` 提供的程序进行了比较, `aprove` 和 `invel` 提供了这些程序, 这是唯一可自由使用的非终止分析仪, 与我们所知的相当。只有**朱莉娅**能检测到由于无限递归而导致的非终止。少

2014 年 1 月 21 日提交;最初宣布 2014 年 1 月。

46. 用向量语言组合有限元矩阵的一种有效方法

作者:[françois cuvelier](#), [caroline juhet](#), [gilles Scarella](#)

摘要: 最近提出了建立有限元矩阵的二维和三维高效 matlab 码。在本文中, 我们提出了简单, 紧凑和有效的矢量化算法, 这些都是这些代码的变种, 在任意维度, 没有使用任何较低的语言。它们可以很容易地在许多矢量语言中实现 (例如, matlab、octave、python、scilab、r、**julia**、c++ 和 stl...)。这些技术的原理是一般的, 我们提出了它的组装在任意维的几个有限元矩阵, 在 p1 有限元的情况下。我们还提供了算法的扩展到一个 pde 系统的情况。然后给出了高阶分段多项式的扩展。我们用数字方法比较了 matlab、octave 和 python 中这些算法的性能, 以及 freefem ++ 和 c. 示例中的自由 fem ++ 和编译语言中的性能, 表明, 与人们通常认为的不同, 这些算法的性能并不比 c 的性能差得多: 在最好的情况下在最坏的情况下, 选定的矢量语言在标量和矢量情况下分别比 c 慢 2.3/3.5 和 2.9/4.1。我们还给出了数值结果, 说明了与标准算法和其他最近算法相比, 这些算法的计算成本。少

2015 年 6 月 19 日提交;v1 于 2014 年 1 月 14 日提交;**最初宣布** 2014 年 1 月。