

# 基于 Julia 语言的并行计算方法初探

巩庆奎<sup>1,2</sup> 张常有<sup>1</sup> 张先轶<sup>1</sup> 张云泉<sup>1</sup>

(中国科学院软件研究所并行软件与计算科学实验室 北京 100190)<sup>1</sup>

(石家庄铁道大学信息科学技术学院 石家庄 050043)<sup>2</sup>

**摘要** Julia 语言是一种在 MIT 许可证下免费开发的开发中脚本语言(beta 0.2.0),目标是降低并行程序的编程难度。基于 Julia 现有语法机制,逐步增强 Julia 语法特性,结合公交线路的平均走行时间统计案例,研究 Julia 并行编程框架和程序逐步精化的方法。Julia 程序支持本地多核心/多 CPU 并行计算。为充分发挥实验平台的计算潜能,尝试了提高 Julia 程序计算性能的策略。对案例程序的实验分析表明,Julia 并行程序在管理计算核心方面耗费了一定的工作时间,但随着问题规模的增大,其影响可逐渐忽略,从而可获得接近线性的加速比。

**关键词** Julia 语言,计算规模,并行程序,程序精化,性能分析

**中图分类号** TP338.8 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.1.009

## Primary Investigation into Parallel Computing in Julia Language

GONG Qing-kui<sup>1,2</sup> ZHANG Chang-you<sup>1</sup> ZHANG Xian-yi<sup>1</sup> ZHANG Yun-quan<sup>1</sup>

(State Key Laboratory of Computing Science, Institute of Software Chinese Academy of Sciences, Beijing 100190, China)<sup>1</sup>

(School of Information Science and Technology, Shijiazhuang Tiedao University, Shijiazhuang 050043, China)<sup>2</sup>

**Abstract** Julia language is a free developing scripting language under the MIT license. Its goal is to ease the difficulty of parallel programming. Based on the language mechanisms of Julia, we constructed a use case of computing the average running-time between every two bus stops. And then, we exemplified the Julia programming framework and the code refining steps. Julia language supports both multi-cores/CPU's parallel programming modes. To full use all the computing resources, we developed some experiments on new policies about how to improve the computing performance. Experiments show that managing processors in parallel computing model consume working time, but with the increasing of problem size, this impact can be gradually ignored, and gaining nearly linear speedup.

**Keywords** Julia language, Computing scale, Parallel program, Program optimization, Performance analysis

Julia 是一种正在开发中的动态高性能的高级编程语言,适合科学和数值计算,性能可与传统静态类型语言媲美。在 MIT 许可证<sup>[1]</sup>下,它可免费下载使用,当前发行版本为 v 0.1.2。通过使用 LLVM(Low Level Virtual Machine),Julia 具有可选的类型声明、重载、高性能、类型推断,以及即时(Just-In-Time, JIT)编译等特性。Julia 采用多编程范式,同时具备指令式、函数式和面向对象编程的特征。它的语法类似 MATLAB,保留了 MATLAB 的简易性和简洁的高等数值计算特性,但适用范围更广,比如可用于数值计算和科学计算。Julia 在数学编程语言的基础上,参考了部分流行的动态语言: Lisp, Perl, Python, Lua 和 Ruby。

## 1 Julia 语言对并行计算的支持

与传统动态语言相比,Julia 语言最大的特点是:(1)核心语言影响较小;(2)标准库用 Julia 语言实现,包括整数运算在内的基础运算;(3)完善的类型机制,方便构造对象和类型声明;(4)基于参数类型进行函数重载;(5)参数类型不同,自动

生成高效、专用的代码;(6)性能高,接近静态编译语言(如 C 语言)<sup>[2]</sup>。

Julia 语言基于消息传递机制为多处理器或多核环境提供并行计算环境,程序能够运行于分布式存储的集群。不同于 MPI 的消息传递机制<sup>[3]</sup>,Julia 采用“单边”通信机制。程序员只需在程序中显式控制多个处理器,调用高层操作即可实现协同计算。

Julia 并行编程以远程引用(remote references)和远程调用(remote calls)两个原语为基础。远程引用是一个对象,用于从任何一个处理器指向存储在特定处理器中的对象。远程调用是一个处理器(主调处理器)向另一个处理器(被调处理器)发出请求,提供约定的参数,调用特定的函数。远程调用的返回值是指向其运行结果的远程引用。远程调用操作将立即返回,主调处理器继续处理后续的操作语句。被调处理器异步执行被调用函数。根据计算流程需要,主调处理器可以调用 wait 函数暂时阻塞,等待被调处理器完成计算,实现多处理器的计算同步。主调处理器调用 fetch 函数,取回被调处

到稿日期:2013-12-26 返修日期:2014-03-15 本文受国家自然科学基金项目(61379048,61100073)资助。

巩庆奎(1986—),男,硕士生,主要研究方向为分布式计算;张常有(1970—),男,博士,教授,主要研究方向为并行与分布式计算,E-mail:changyou@iscas.ac.cn;张先轶(1983—)男,博士,主要研究方向为并行计算;张云泉(1973—),博士,主要研究方向为并行计算。



挖掘语言特性,能够实现程序对计算资源的自动感知,并精心设计程序结构,实现数据初始化、数据划分、任务分解等功能。

Julia 语言提供了函数 `nprocs()` 得到 Julia 当前运行环境中可用的处理单元个数。用 `np=nprocs()` 感知系统环境的处理单元个数,则可将 `np` 变量用作数据划分和任务分配的参数,由程序自身适应不同运行环境下的计算资源数量而自动调优。

鉴于 Julia 数组的支持暂未能达到 C 语言中的灵活度,比如不能把带下标的子数组看作一个独立数组(因为是指针),本文 Julia 程序中子数组的名字与处理单元个数相关联,在远程调用中需要严格对应。充分利用 Julia 程序的脚本语言特性,程序可先生成程序文件,并在后面程序执行过程中的适当位置包含进来(include 指令),实现程序对处理单元个数的自适应效果,包括数据的自动划分和远程调用中的参数匹配等。

分布式集群上对大数组的划分和数据传输都是时间消耗的重要来源,且需要更多的存储空间,使计算规模受到限制。如果把数据文件存于共享目录,各处理器按照特定规则读取文件中的指定数据段,则可提升效率。我们测试了 4 核 CPU 并行读取数据文件情况下的程序执行效率,效率对比数据如表 2 所列。

表 2 不同读文件方式下的程序执行效率对比

时间(s)	主控读入	分布读
数据准备	1.212	1.096
分配	6.258	0.97, 1.12, 1.14, 1.11
合成结果	1.56	1.58
合计	9.03	≈3.8

(上接第 37 页)

[11] Zhou S N, Zheng X H, Wang J W, et al. UTOPIA: a load sharing facility for large, heterogeneous distributed computer systems [J]. *Software: Practice and Experience*, 1993, 23(12): 1305-1336

[12] Mu'alem A W, Feitelson D G. Utilization, predictability, workloads, and user runtime estimates in scheduling the IBM SP2 with backfilling [J]. *IEEE Transactions on Parallel and Distributed Systems*, 2001, 12(6): 529-543

[13] Sherwani J, Ali N, Lotia N, et al. Libra: a computational economy-based job scheduling system for clusters [J]. *Software: Practice and Experience*, 2004, 34(6): 573

[14] Shan Z G, Lin C. Modeling and performance evaluation of hierarchical job scheduling on the grids [C] // *Proceedings of the 2007 International Conference on Grid and Cooperative Computing*, 2007, Urumchi, IEEE Computer Society, 2007: 296-303

[15] Naik V K, Setia S K, Squillante M S. Performance analysis of job scheduling policies in parallel supercomputing environments [C] // *Proceedings of the 1993 ACM/IEEE conference on Supercomputing*, 1993. New York: ACM, 1993: 824-833

[16] Moschakis I A, Karatzas H D. Evaluation of gang scheduling performance and cost in a cloud computing system [J]. *The Journal of Supercomputing*, 2012, 59(2): 975

[17] Abawajy J H. An efficient adaptive scheduling policy for high-performance computing [J]. *Future Generation Computer Systems*, 2009, 25(3): 364

**结束语** 本文通过公交车次站间走行时间统计案例,尝试了 Julia 并行程序的精化过程,并对 Julia 在平台自适应、规模动态扩展等方面进行了探讨。Julia 作为脚本语言,其优秀之处在于其编程的容易性。通过为特定领域的应用程序开发高层算法库,隐式支持并行算法实现,能进一步提高其易用性。Julia 语言提供了丰富的高效的函数库,比如通常情况下使用 Julia 自带的 `mean()` 函数对数组的指定维度求平均值,性能明显高于自编程序。Julia 能够方便地与其他语言接口(比如 C 语言和 Fortran 语言),或调用外部程序,使其容易适应平台,充分利用平台的计算能力。如果用 CUDA/C 制作动态库,通过 Julia 程序调用,可利用节点上的 GPU 协处理器,加速应用程序中的关键算法。

## 参考文献

- [1] [http://en.wikipedia.org/wiki/MIT\\_License](http://en.wikipedia.org/wiki/MIT_License)
- [2] <http://julialang.org/>
- [3] Mathematics and Computer Science Division Argonne National Laboratory. MPICH User's Guide (Version 3.0.4) [OL]. [2013-10]. <http://www.mpich.org/documentation/guides/>. Apr. 24, 2013
- [4] 李润梅,刘建忠,朱凤华. 平行公交系统中的计算实验问题研究 [J]. *自动化学报*, 2013, 39(7): 1011-1017
- [5] <http://bus.17u.com/bus/beijing/>
- [6] 张常有,张先轶. Julia 语言与并行计算 [R]. 第 6 届 R 语言大会主题报告. 北京: 人民大学
- [18] Platform Computing Corporation. Running Jobs with Platform LSF [OL]. <http://www-03.ibm.com/systems/services/platformmcomputing>
- [19] Palma J N. Performance evaluation of interconnection networks using simulation-tools and case studies [D]. Bilbao: University of the Basque Country, 2009
- [20] Sur S, Koop M J, Chai L, et al. Performance analysis and evaluation of Mellanox ConnectX InfiniBand architecture with multi-core platforms [C] // *Proceedings of the 15th Annual IEEE Symposium on High-Performance Interconnects*, 2007. Stanford: IEEE, 2007: 125-134
- [21] Mellanox. Mellanox InfiniBand Training [OL]. <http://www.mellanox.com/>
- [22] 超级计算机 TOP500 排名 [OL]. <http://www.top500.org/>
- [23] 林闯. 随机 Petri 网和系统性能评价 (第 2 版) [M]. 北京: 清华大学出版社, 2005
- [24] 陈永然. 面向高性能计算的性能评价模型技术研究 [D]. 长沙: 国防科学技术大学, 2007
- [25] 王翠萍. LSF 系统中作业调度的研究与优化 [D]. 西安: 西安电子科技大学, 2009
- [26] 曹宗雁. 高性能计算集群运行时环境的配置优化 [J]. *科研信息化技术与应用*, 2011, 2(6): 52-61
- [27] 林闯. 随机 Petri 网模型的精化设计 [J]. *软件学报*, 2000, 11(1): 104-109