

基于 MPI 和 CUDA 的蛋白质定量软件的设计和分析

詹 科^{1,2} 王 靖¹ 袁 良^{1,2} 张云泉¹

(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)¹

(中国科学院研究生院 北京 100049)²

摘 要 介绍了蛋白质定量软件的设计及测试结果,分析结果表明:P-QuantWiz 软件在曙光 6000 的 Intel 集群部分,测试规模达到 2048 核,在 256 核加速比仍在增加。相对 P-QuantWiz 软件在单 CPU 运行的结果,PG-QuantWiz 软件在单 GPU 运行的加速比为 8.1,在 16 个 GPU 上运行的加速比为 14.18,并行效率为 89%。

关键词 MPI, CUDA, 蛋白质定量

中图法分类号 TP301 **文献标识码** A

Design and Analysis of Protein Quantification Software Based on MPI and CUDA

ZHAN Ke^{1,2} WANG Jing¹ YUAN Liang^{1,2} ZHANG Yun-quan¹

(Laboratory of Parallel Software and Computational Science, ISCAS, Beijing 100190, China)¹

(Graduate University, The Chinese Academy of Sciences, Beijing 100049, China)²

Abstract We introduced the design and the testing results of protein quantification software. The analytical results indicate that the testing size reaches 2048 cores, when P-QuantWiz runs on the clusters of DAWNING 6000, and the speed-up ratio still increases when we use 256 cores. Compared with the results of P-QuantWiz which is ran on one CPU, the speed-up ratio of PG-QuantWiz on one single GPU is 8.1. Compared with the results which is ran on one GPU, the speed-up ratio of PG-QuantWiz on 16 GPU is 14.18, and the parallel efficiency is 89%.

Keywords MPI, CUDA, Protein quantification

1 背景

随着质谱技术和生物实验手段的飞速发展,蛋白质定量的精度越来越高,一次定量实验所产生的质谱数据量也日益增大,定量性能已成为制约定量蛋白质组学向前发展的一个瓶颈。对于超大规模的定量实验数据,利用现有定量软件分析处理需要长达数天甚至数月之久,传统定量分析算法和软件已经不能满足大规模质谱数据分析处理的要求。基于此,设计并实现高性能定量算法和软件对于定量蛋白质组学的研究具有意义重大^[1]。

基于质谱的非标记定量方法是最近几年随着蛋白质分离技术和质谱技术的发展而发展起来的,这种方法省去了同位素标记等繁杂的化学处理过程,只需要分析处理质谱原始数据,适用于大规模质谱数据分析。

本文基于两种并行编程技术 MPI 和 CUDA 对蛋白质定量软件进行优化。其中 MPI 是消息传递函数库的标准规范,由 MPI 论坛开发^[2]。

CUDA 由 NVIDIA 公司于 2007 年 6 月推出,是一种将 GPU 作为数据并行计算设备的软硬件体系^[3]。CUDA 代码包括在 CPU 上运行的串行部分和在 GPU 上运行的并行部

分,运行在 GPU 上的程序被称为内核函数。当对大量数据进行处理步骤基本相同时,适合使用 GPU 进行计算;需要使用复杂数据结构的计算不适合使用 GPU 进行计算。

基于串行版本的蛋白质定量软件利用 MPI 和 CUDA 编程技术实现了软件的并行化。测试结果表明,P-QuantWiz 和 PG-QuantWiz 均有良好的并行性。

本文第 2 节介绍软件实现方法;第 3 节介绍测试环境和测试过程;第 4 节是测试结果和对测试结果进行的分析。

2 蛋白质定量软件简介

2.1 CS-QuantWiz 软件

CS-QuantWiz 是蛋白质定量软件的串行版本^[4]。非标记定量实际上就是对多个质谱数据进行分析处理。在定量之前,对质谱数据进行对齐。质谱对齐实质上就是一种映射关系,即根据某一质谱文件中鉴定到的肽段 P,能够预测肽段 P 在其它质谱文件中的鉴定位置。色谱峰平滑就是用某一种函数来拟合平滑色谱峰,以便峰的识别。峰识别就是根据平滑后的色谱峰图形,找出色谱峰的波峰和波谷。峰下面积可以衡量肽段的量,根据峰下面积可以计算肽段在多个质谱文件中的比值。其软件流程如图 1 所示。

到稿日期:2012-10-19 返修日期:2012-12-12 本文受中国科学院知识创新工程重大项目课题(KGCX1-YW-13)资助。

詹 科(1980-),男,博士生,主要研究方向为并行计算;王 靖(1982-),男,硕士,主要研究方向为生物信息学及并行软件;袁 良(1984-),男,博士生,主要研究方向为并行计算模型、并行算法设计;张云泉(1973-),男,研究员,博士生导师,主要研究方向为高性能计算、并行数值软件、并行计算模型、并行数据挖掘,E-mail:zyq@mail.rdcps.ac.cn(通信作者)。

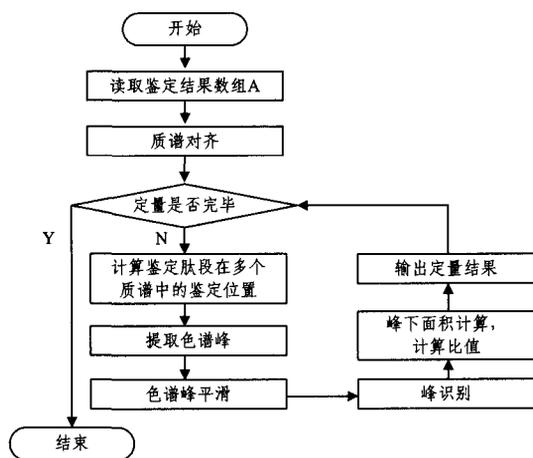


图1 CS-QuantWiz 软件流程

2.2 P-QuantWiz 软件

P-QuantWiz 软件是 CS-QuantWiz 软件的 MPI 并行版本^[5,6]。非标记定量软件流程主要是一个循环过程。其基本思想是通过数据分解,将鉴定结果按鉴定时间划分到每个处理器之上,进行定量分析后返回主节点,输出结果。数据分解,将鉴定结果按负载划分到每个处理器之上,就是 P-QuantWiz 并行化最基本的思想。

其中软件流程如图 2 所示。

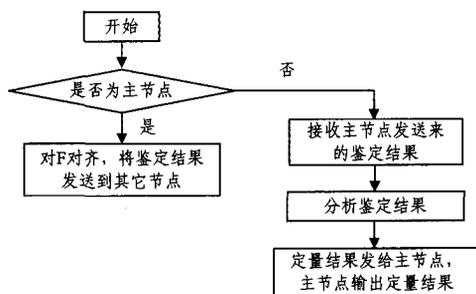


图2 P-QuantWiz 软件流程

2.3 PG-QuantWiz 软件

本节基于上一节的 MPI 版本设计 MPI+CUDA 并行代码,经过分析,非标记量化的流程主要是一个循环过程。采用的并行方案是将鉴定结果按负载进行划分。由 CPU 的 MPI 主进程将多个鉴定结果传送给 GPU,由 GPU 完成定量计算。最后将结果返回到 CPU。算法如下:

输入:多个质谱数据集 F,鉴定结果 I

输出:定量结果

1. 判断节点是否是主节点,如果不为主节点,转 3。
2. 将 F 对齐,把鉴定结果 I 按鉴定时间排序并按负载静态划分发送到其它节点,转 4。
3. 接收主节点发送来的鉴定结果。
4. 从主机内存拷贝数据到 GPU 显存。
5. 调用 RemoveBaseLineKernel 对数据进行去噪处理。
6. 从 GPU 显存拷贝数据到主机内存。
7. 继续定量分析的其他步骤。
8. 将定量结果发送到主节点,主节点输出定量结果。

3 测试环境和过程

曙光 6000 集群 Intel 部分采用的 CPU 为 Intel(R) Xeon (R) CPU X5650 @ 2.67GHz 2 路 6 核,操作系统为 Linux 2.6.32.12,gcc 版本为 4.3.4,MPI 为 intel openmpi 1.4.4。

龙芯部分采用的 CPU 为 ICT Loongson-3 V0.5 包含 FPU V0.1 2 路 4 核,操作系统为 Linux 2.6.36,gcc 版本为 4.4.5,mpi 为 MPICH2 version 1.2.1p1。

超龙一号 X86+GPU 部分共包括 8 个计算节点、1 个存储节点。每个计算节点由 10 个刀片组成。CPU 类型为 X5650,主频为 2.67GHz。GPU 为 Nvidia Fermi C2050,3GB GDDR5 内存。存储节点 CPU 为 2×Intel Nehalem X5650,主频 2.66GHz。存储总容量为 48TB。操作系统为 SUSE Linux Enterprise Server 11 SP1,内核版本为 2.6.32.12-0.7,MPI 通信库为 OpenMPI 1.4.4,数学函数库为 Intel Math Kernel 2012。超龙一号龙芯部分共两个机柜,每个机柜 32 个 HPP 节点,每个 HPP 节点由 4 个计算模块和 1 个管理模块组成。操作系统为 Debian lenny/sid,内核版本为 2.6.27.6。MPI 通信库为 Hybrid-MPI (基于 Mvapich-1.1),数学函数库为龙芯优化版 GotoBLAS2。

测试数据为中国科学院系统生物学重点实验室提供的五样本的小鼠蛋白质 Orbitrap 质谱数据以及相应鉴定数据。

P-QuantWiz 在 Intel 集群部分,进行了从 64 到 2048 处理器核的规模测试;在龙芯集群部分,进行了从 8 到 1600 核的测试;在超龙一号进行了节点内和跨节点测试。

PG-QuantWiz 在超龙一号进行了单 GPU 和多 GPU 的测试。

4 测试结果及分析

4.1 P-QuantWiz 测试结果

Intel 集群和龙芯集群上,P-QuantWiz 软件的主要测试结果如图 3 和图 4 所示。

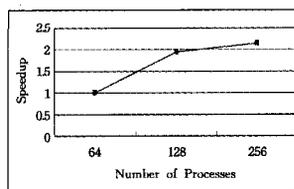


图3 在曙光 6000(Intel CPU) 的加速比

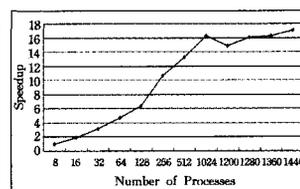


图4 在曙光 6000(龙芯 CPU) 的加速比

从图 3 可以看出,在曙光 6000 的 Intel 集群部分,测试规模达到 2048 核,相对于 64 核,在 256 核加速比仍在增加。

如图 4 所示,在龙芯集群部分,测试规模达到 1600 核,相对于 8 核,在 1440 核加速比仍在增加。在规模增加初期,由于所划分的数据各部分之间相对独立,不存在数据依赖问题,因此加速比呈较好的增大趋势。

在图 3 处理器数为 256 和图 4 处理器数为 1024 的时候,加速比有明显的下降,原因是处理器数目增加,处理器之间的通信开销增大,影响了软件的加速。

P-QuantWiz 在超龙一号的测试结果如图 5 和图 6 所示。

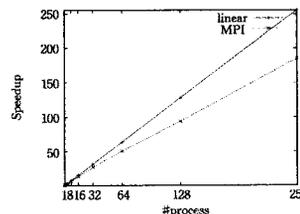
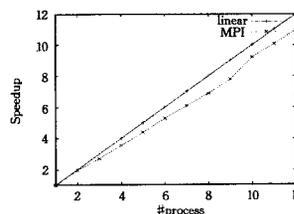


图5 X86 平台节点内测试结果 图6 X86 平台跨节点测试结果

(下转第 54 页)

[3] Alfaro J, Boulahia-Cuppens N, Cuppens F. Complete Analysis of Configuration Rules to Guarantee Reliable Network Security Policies[J]. Int'l J. Information Security, 2008, 7(2): 103-122

[4] Hari A, Suri S, Parulkar G M. Detecting and resolving packet filter conflicts[C]//INFOCOM. 2000(3):1203-1212

[5] Fu Z, Wu S F, Huang H, et al. IPSec/VPN security policy: Correctness, conflict detection, and resolution[C]//Proceedings of Policy2001 Workshop. January 2001

[6] Golnabi K, Min R K, Khan L, et al. Analysis of firewall policy rules using data mining techniques[C]//IEEE/IFIP Network Operations and Management Symposium (NOMS 2006). April 2006

[7] Hu H, Ahn G J, Kulkarni K. Detecting and resolving firewall policy anomalies[J]. IEEE Transactions on Dependable and Secure Computing, 2012, 9(3): 318-331

[8] Abedin M, Nessa S, Khan L, et al. Detection and resolution of anomalies in firewall policy rules[C]//DBSEC'06 Proceedings of

the 20th IFIP WG 11.3 Working Conference on Data and Applications Security. 2006;15-29

[9] 田大新, 刘衍珩. 数据包过滤规则的快速匹配算法和冲突检测[J]. 计算机研究与发展, 2005, 42(7): 1128-1134

[10] Al-Shaer E, Hamed H, Boutaba R, et al. Conflict Classification and Analysis of Distributed Firewall Policies[J]. IEEE Journal on Selected Areas in Communications, 2005, 23: 2069-2084

[11] Mell P, Scarfone K, Romanosky S. A Complete Guide to the Common Vulnerability Scoring System Version 2.0[K]. Published by FIRST—Forum of Incident Response and Security Teams. June 2007

[12] 张永铮, 方滨兴, 迟悦, 等. 用于评估网络信息系统的风险传播模型[J]. 软件学报, 2007, 18(1): 137-145

[13] Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters[J]. Communication of the ACM-50th anniversary issue, 2008, 51(1): 107-113

[14] 李建江, 崔健, 王鹏, 等. MapReduce 并行编程模型研究综述[J]. 电子学报, 2011, 39(11): 2635-2642

(上接第 37 页)

如图 5 所示, 在一个节点内运行多个进程, 随着进程数增加, 加速比几乎线性增加。在进程数为 12 时, 加速比为 10.9, 并行效率为 90.8%。

如图 6 所示, 在多个节点中, 使用多进程测试。进程数从 1, 2, 4 一直倍增到 256。在 32 进程时, 加速比为 28.9, 并行效率为 90.3%。在进程数达到 256 时, 加速比为 184.6, 并行效率为 72.1%。综合图 5 和图 6, 软件在超龙一号 X86 部分单节点和多节点运行时, 都有很好的加速效果。

对比图 4 和图 6 的加速比, 二者的加速效果不同, 可能原因是在图 4 中, 蛋白质定量程序运行在曙光 6000 龙芯部分, 而图 6 中, 程序运行在超龙一号 X86 平台, 二者的 CPU 频率不同, 其他的硬件配置也不同, 所以程序运行在这两个平台加速比不同。

如图 7 所示, 在多个节点中, 使用多进程测试。进程数从 1, 2, 4, 8 到 16。在 16 进程时, 加速比为 14.36, 并行效率为 89.7%。软件在龙芯部分得到很好的加速效果, 并行效率达到与 X86 CPU 相同乃至更高水平。

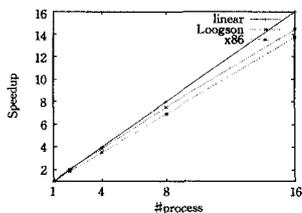


图 7 龙芯平台和 X86 CPU 测试结果对比

4.2 PG-QuantWiz 测试结果

在 X86 部分, MPI+CUDA 版软件 PG-QuantWiz 主要的测试结果如图 8 所示。

PG-QuantWiz 在单 GPU 运行时间为 498.19 秒, 结合 P-QuantWiz 运行在超龙一号 X86 部分单 CPU 的情况, MPI+

CUDA 版软件 PG-QuantWiz 在单 GPU 上运行的加速比为 8.1。16 GPU 时间为 35.11 秒, 16 个 GPU 对单 GPU 的加速比为 14.18, 并行效率为 89%。这表明 MPI+CUDA 版软件 PG-QuantWiz 也有很好的加速效果。

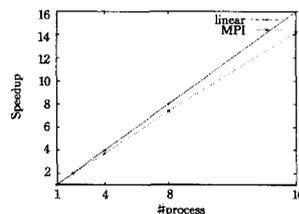


图 8 PG-QuantWiz 测试结果

但是因为测试数据规模有限, 在较大的并行环境下, 其可扩展性受到影响。因此需要对其进行进一步的优化, 包括进一步加大数据测试规模, 并根据负载均衡进行数据动态划分。

参考文献

[1] 胡泽林. 基于质谱的高性能蛋白质非标记定量软件设计与性能优化[D]. 2009

[2] MPI: A Message-Passing Interface Standard [OL]. <http://www.mpi-forum.org/docs/mpi-11-html/mpi-report.html>

[3] NVIDIA corporation. NVIDIA CUDA C Programming Guide version 4.2[Z]

[4] 费辉. 蛋白质定量和分子动力学模拟若干算法的 GPU 并行设计与实现[D]. 2011

[5] Wang Jing, Zhang Yun-quan, Zhang Xian-yi, et al. QuantWiz: A Parallel Software Package for LC-MS-based Label-free Protein Quantification[C]// Accepted by 2009 International Workshop on Parallel Algorithm and Parallel Software. 2009; 683-687

[6] 费辉, 张云泉, 王可, 等. 基于 GPU 的分子动力学模拟并行化及实现[J]. 计算机科学, 2011, 38(9): 275-278