桌面网格计算平台在多序列比对算法中的运用①

李金龙，贾晓雯

(浙江传媒学院 电子信息学院，杭州 310018)

摘 要：提出了一种多序列比对 ClustalW 算法并行化处理的新方法 ParaClustalW，该方法使用桌面网格计算平台作为高性能编程环境和运行平台。分析了多序列比对算法在桌面网格平台上的任务划分方式、并行化策略和实现技术。ParaClustalW 策略考虑到序列的数目与序列的长度等因素，实现任务划分的均衡性。经实验证明，ParaClustalW 方法具有较好的计算速度和加速比，在同等机器配置下，优于当前的一些计算方法，对多序列比对算法的研究具有参考价值。

关键词：桌面网格计算；计算模型；序列比对；生物信息学；并行计算

Application of Desktop Grid Computing Platform to Multiple Sequence Alignment

LI Jin-Long, JIA Xiao-Wen

(College of Electronic and Information, Zhejiang University of Media and Communications, Hangzhou 310018, China)

Abstract: A new parallel processing approach for ClustalW was presented and described in this paper. This approach uses Desktop Grid Computing Platform, which is a promising idea for high performance computing as its programming environment and running platform. On Desktop Grid computing platform, task partition, parallelism strategy and implementation technology of clustalW algorithm program were discussed in detail. ParaClustalW policy is load balances which was strongly supported for its consideration of alignment length and number factors. In order to demonstrate the effectiveness of this approach, a serial of simulation experiments were also done. The results obtained from performance analysis show that it gained a quick working time and good speedup. Under the same machine configuration, our approach is better than other solutions and it is a feasible and valuable approach for Multiple Sequence Alignment in bioinformatics.

Keywords: desktop grid computing; computing model; sequence alignment; bioinformatics; parallel computing

1 引言

近年来桌面网格(Desktop Grid)被广泛应用，通过运行计算量大的、分布式的应用，来充分共享大规模因特网上的计算资源，如 OurGrid[1], ShareGrid[2]等都是典型桌面网格的例子。

在生物多序列比对中， ClustalW[3]算法是最常用的一种多序列比对程序。它是一种渐进的比对方法，ClustalW 算法对计算的性能要求很高，计算量的大小都由所选择的序列的长短来决定，由于受算法复杂度的限制，几乎不可能在单机上实现大量的序列的多序列比对，所以采用网络计算平台来解决多序列比对算法并行化问题成为研究热点[4]。

本文把桌面网格计算平台运用到生物多序列比对计算领域，通过构建一个桌面网格计算平台，提出了适合该计算平台的多序列比对 ClustalW 算法的并行化方法，通过实验与性能分析，验证了该方法的高效性。

2 多序列比对的工作平台

根据前面的思想，为了实现多序列比对的并行化 ClustalW 算法，先根据桌面网格计算平台的原理及目前国际上著名的桌面网格项目的经验，构建了 ClustalW 算法的桌面网格平台。整个桌面网格平台的

① 收稿时间:2010-07-07;收到修改稿时间:2010-08-05

网络体系结构如图 1 所示，该桌面网格把所有节点按照功能划分为服务器端节点，调度节点，工作机节点和后台数据服务器，形成一个层次网络体系结构。服务器端可以是一个节点，调度节点部分由多个节点组成，服务器端负责整个平台的节点管理、对平台和应用进行监控和处理并行计算任务的提交。调度节点作为平台的调度器，负责对计算子任务进行分发和调度。大规模因特网环境下的志愿机构成了工作机节点，是子任务的具体执行者，数据服务器存储整个平台的代码和数据。

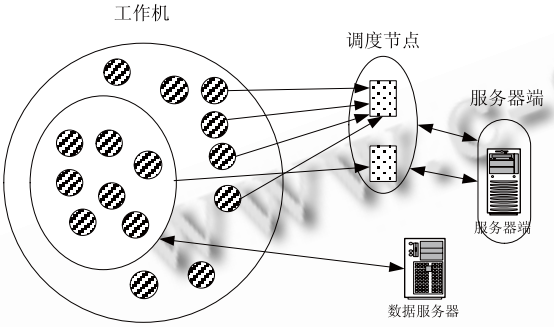

图 1　ClustalW 算法在桌面网格上的体系结构

为了使各个工作机节点能够并行的工作，该桌面网格采用主-从(Master-Worker)的方式对并行应用进行编程。首先用户根据应用编写好串行的子任务代码，把子任务所需的数据文件提交给数据服务器保存，并向服务器端节点提交计算任务；然后开始编写主程序并将其运行，在平台工作时，通过调度节点的任务分发机制，把各个子任务分派给工作机节点。用户主程序控制整个并行算法的运行，当所有的子任务计算结果计算完后，主程序从数据服务器中获取子任务的计算结果进行判断汇总而完成整个并行应用的计算。有了该桌面网格平台，多序列比对的 ClustalW 算法并行处理成为可能。

3　多序列比对的任务划分

3.1 多序列比对算法 ClustalW

多序列比对是在双序列比对的基础上发展起来的，由于比对本身是个 NP 难度问题，在研究多序列比对算法的问题上没有最优解，人们研究多序列比对算法多集中在从各种角度寻求一种近似最优处理，同时使算法在时空复杂度和比对效果之间取得一个更好的平衡，包括动态规划算法 、渐进比对算法等、迭代比对算法[5]等。

ClustalW 算法是目前使用最广泛的多序列比对程序。ClustalW 是一种渐进比对方法，它的基本思想也是基于假设需要比对的序列之间是有进化关系的。研究表明 ClustalW 算法基本步骤包括 PA、NJ 和 PW 三个阶段。经过计算统计，ClustalW 算法的三个计算阶段在计算序列条数不同时各自所占时间比例也不同，ClustalW 各阶段计算时间分布如图 2 所示，由图可见，算法的 PW 阶段占用较大比例的计算时间。

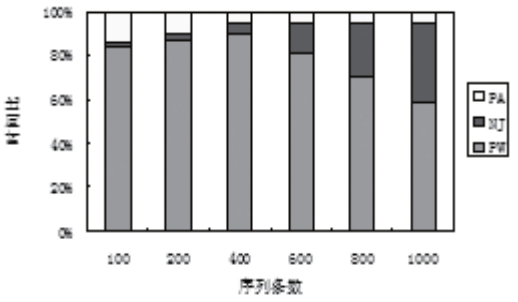

图 2　Clustalw 各阶段计算时间分布

本文之所以选择了 ClustalW 作为多序列比对并行化的试验程序，因为和遗传算法与蚁群算法比起来，这种利用桌面网格来实现多序列比对计算具有一定的新颖性与现实意义，而从前面第 2 节中提出的桌面网格要求应用为主-从(Master-Worker)模式，即两个子任务之间不需要发生通信，而 ClustalW 算法在 PW 阶段的是一种两层循环，每一次内部循环将处理一次两个序列的双序列比对，而每一次双序列比对之间都是相互独立的，把每次或若干次双序列比对作为一个子任务，所以 ClustalW 算法并行化可基于 PW 阶段进行。

3.2 ClustalW 算法的任务划分

由于 ClustalW 算法中多序列比对需要把比对目标序列进行全部的两两比对，因此，如果有 N 条序列需要进行多序列比对比对，则在 PW 阶段，序列 1 需要和其他 N-1 个序列进行双序列比对，序列 2 需要和其它 N-2 个序列比对，依此类推，一直到序列 N-1，只需要和序列 N 进行一次比对，这样一次多序列比对总

共需要进行次双序列比对。以 12 条序列的比对为例，则 ClustalW 算法需要进行的双序列比对情况如图 3 左图所示。黑框表示所在行和列的两个序列之间需要进行双序列比对，白框则表示这两个序列之间不需要进行双序列比对。由图 3 可知，12 条序列需进行的双序列比对的次数等于黑框的数目，即为 66。

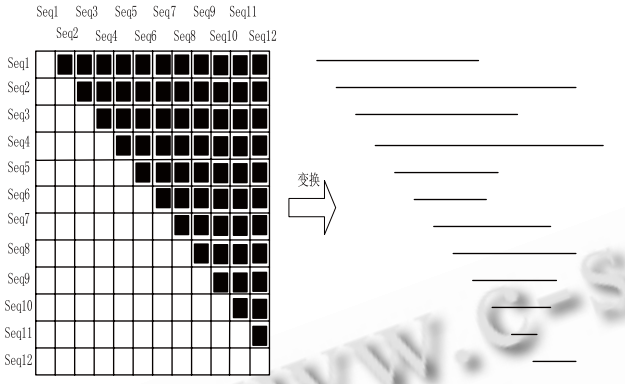

图 3 多序列比对的任务矩阵变换

图 4 显示了 ClustalW 串行的算法输入序列数目 N 和计算时间 T 之间的关系曲线；如图 5 为多序列比对输入序列长度 L 与计算时间 T 之间的关系曲线。由图分析可知，多序列比对的总的计算量和输入序列的数目基本呈指数关系(约 X1.8)，与序列的长度也基本是指数关系(约 X2)。在对 ClustalW 算法进行并行化时，首先需要根据序列的长度 L 与序列的数目 N 进行任务划分，将串行执行的任务划分为多个可并行执行的计算子任务。结合 ClustalW 算法的特点，目前已有两种并行任务划分策略：平均划分策略和基于时间预测的划分策略。

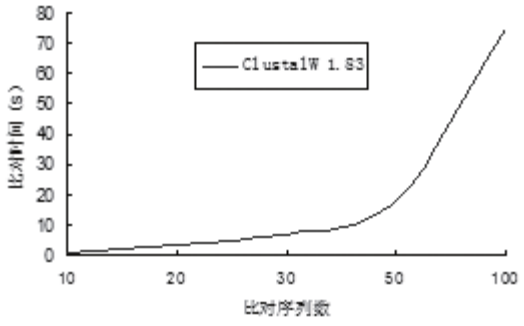

图 4 序列数目 N 和计算时间 T 的关系

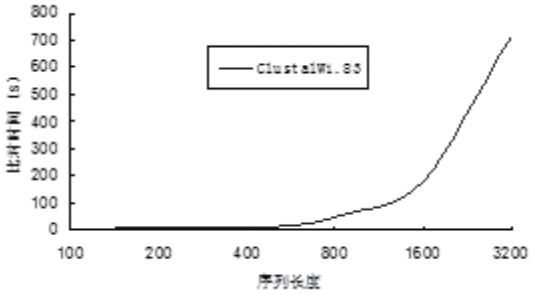

图 5 序列长度 L 与计算时间 T 的关系

在平均划分策略中，假设计算任务划分刻度为 K，即将总的计算任务划分为 K 个子任务，用平均划分策略，使划分后每个子任务进行的双序列比对的数目基本相同，这样每个子任务需要进行次双序列比对。平均划分策略简单易实现。在基于时间预测的划分策略中，ClustalW 算法进行多序列比对，考虑比对序列长度 L 与计算时间之间的关系，根据 L 的长度划分子任务，实现任务的自适应并行与任务负载均衡。

3.3 并行化方法 ParaClustalW

为了在前面的桌面网格平台上实现 ClustalW 算法并行化，本文既要考虑平均划分策略的方便简单，也要结合基于时间预测的负载均衡性，把这种策略称为 ParaClustalW。ParaClustalW 任务划分策略的思想是：在任务划分之前，需要根据时间复杂度经验公式(图 4 与图 5)，对每一次双序列比对的比对时间进行估计，再根据估计时间将其映射成长度不等的整数型线段，如图 3 右边所示。图 3 中为 ParaClustalW 策略的任务矩阵变换图，每一次双序列比对不再变换成长度为 1 的线段，而是以基准比对量为函数，变换成不等长的线段。

ParaClustalW 策略的任务划分过程具体描述如下：

(1) 读入 N 个序列，并统计每一个序列的长度，并保存起来；

(2) 根据时间经验估计公式以及比对序列的长度，估计出每一个双序列比对的计算时间，并保存起来；

(3) 根据预设的基准时间参数 Tk(由时间复杂度公式决定)，计算出每一个双序列比对的比对时间刻度，并对计算结果取整后记录起来；

(4) 根据时间刻度进行子任务划分,一个或若干个

计算量差不多的双序列比对构成子任务。

此任务划分 ParaClustalW 方法特点充分考虑到由于序列长度的不同导致的任务划分的不均衡，桌面网格下如果子任务之间的计算量相差较大，会出现某一个子任务计算时间过长，会将正常运行的子任务误认为是计算超时，并进行容错处理，这样容易导致系统计算资源浪费，甚至子任务计算失败。

4　实验与性能分析

4.1 实验环境

实验环境包括桌面网格平台的环境和 ClustalW 算法的任务划分的方式。实验的平台环境采用内部网络环境来模拟大规模的因特网，如表 1 和图 6 所示。在任务划分的方式采用 ParaClustalW 划分策略，设定划分后每个子任务需完成 10 次双序列比对，这样划分后可以得到 123 子任务。试验测试序列为 SARS 冠状病毒序列，序列数据全部来源于 GenBank 数据库。这些序列的数据源来自不同的国家和地区。

表 1　ClustalW 算法的硬件环境

节点类型	主频(HZ)	内存(M)	节点数	操作系统	Java 环境
服务器端	P4 2.8G	1000	1	Windows 2003	J2sdk1.4.2
调度节点	Celeron 2.9G	1000	4	Windows 2003	J2sdk1.4.2
数据服务器	P4 2.8G	2000	1	Reahat Linux	J2sdk1.4.2
工作机	Celeron 2.9G	1000	40	Windows XP	J2sdk1.4.2

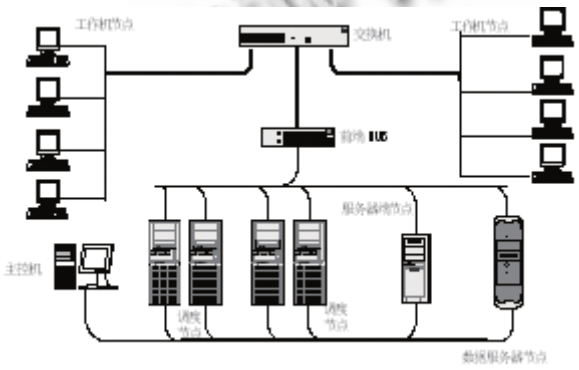

图 6　ParaClustalW 算法实验的工作平台环境

4.2 结果与性能分析

在多序列比对的 ParaClustalW 算法并行化时，根据选取好的 50 条 SARS 冠状病毒序列，用户先设置好序列文件，序列文件作为子程序的数据，包括数据文件(**.seq)和参数文件(**.parm)。其中前者为子程序要进行比对的序列对象数据，包括 N(N>=2)条基因数据；而后者则包含一些比对过程中所需的简单参数。在桌面网格的工作机节点个数 n(n=8、16、24、32、40)的情况下进行了测试。当任务提交后，平台的调度节点将各个序列分配在工作机节点上进行比对。当所有的子任务完成后，主程序进行处理，最后得到比对结果。

经测试，在 PW 阶段需进行总共 1225 次双序列比对。试验中性能最好的一台 PC 完成一个子任务需要大约 30 分钟，而性能较差的 PC 则需要约 54 分钟，计算后得出运行一个子程序平均需要大约 42 分钟。在桌面网格工作机节点数不同的情况下，计算的加速比如图 7 所示。

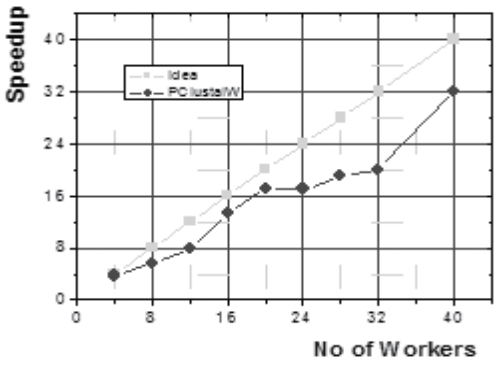

图 7　ParaClustalW 策略并行处理的加速比

由图 7 分析，计算加速比随着工作机节点数的增加而增加，在工作机节点数为 8 时，计算加速比为 5.7，在工作机节点数增加到 40 时，计算加速比为 30.6。加速比的变化基本正比于工作机节点数的变化，加速比曲线出现的一些波动是由工作机节点和网络状态的不稳定造成的，从规模来看，当工作机数量达到很多的时候，加速度提高比很明显，基本接近于理想值，这也证明了桌面网格平台在时间长、规模大的情况下，有时性能也会优于集群或网格计算模型。该实验也验

(下转第 209 页)

```
FROM tbl_xml_data ;
SQL>SELECT count(*)
FROM tbl_xml_data
WHERE existsNode
(f02,'/书[书名="Oracle 实用教程"]') = 1;
```

（3）返回 XML 文档片断

```
SQL>SELECT extract(f02,'/书')
    FROM tbl_xml_data ;
```

4 总结

关系数据库仍然是目前主流的数据库存储技术，使用关系数据库存取 XML 数据实用性强，可以实现用户对 XML 数据的透明管理，同时提高了数据交换的效率。本文根据 XML 标准化思路，结合关系数据库的优势，提出了基于关系数据库管理 XML 文件的数据存储方案，通过使用 Oracle XML DB 技术构建了 XML 数据存取系统，大大简化了建立数据仓库的数据的获取，转化和处理过程，同时实现了 XML 数据的并发访问，提高了数据的完整性和安全性。随着这两种技术的相互借鉴和结合，数据的标准化工作和数据挖掘技术也会更加日臻完善。

参考文献

1 陈弦,陈松乔.基于数据仓库的通用 ETL 工具的设计与实现.计算机应用研究,2004,21(8):214－216.

2 张丽华.基于 PB 数据管道的异构数据库转换系统设计与实现.计算机系统应用,2006,15(11):73－76.

3 罗林球,孟琦,李晓,苏国平,张澄澈.异构数据库迁移的设计和实现.计算机应用研究,2006,23(12):233－238,

4 周红波,孙宇达,王继霞,王瑞,王志宝.基于 XML 的数据交换及其参照完整性研究.计算机工程与设计,2006,27(14):2611－2613.

5 顾天竺,沈洁,陈晓红.基于 XML 的异构数据集成模式的研究.计算机应用研究,2007,24(4):94－96.

6 文必龙,王守信.一个基于 XML Schema 的数据交换模型.大庆石油学院学报,2004,28(2):65－68,

7 屈正庚.利用 SQLXML 创建 XML 查询的方法.商洛学院学报,2006,(4):34－37.

8 江枫.Oracle XML DB 的发展历程.程序员,2007,(12):67－69.

9 仇丽青,赵庆祯,基于 XML 的数据仓库系统.计算机系统应用,2004,13(2):12－14.

10 兰小机,鲁小娟.基于 Oracle XML DB 的 XML 文档存储技术的研究.测绘科学,2008,33(5):201－203.

(上接第 172 页)

证了本文提出的 ParaClustalW 算法的并行处理具有很好的性能。

5 结论与展望

本文把桌面网格平台运用到生物信息学多序列比对领域，分析了 ClustalW 算法的任务划分策略，描述了 ParaClustalW 并行化策略，模拟实验表明，该方法具有较好的性能，对多序列比对算法的研究具有一定的参考价值。如何更好的根据桌面网格平台的特点，优化 ClustalW 算法的并行划分策略，实现自适应并行与负载均衡，充分提高因特网上工作机的空闲处理器周期是本文的下一步工作。

参考文献

1 Andrade N, Costa L, Germoglio G, Cirne W. Peer-to-peer grid computing with the ourgrid community. Proc. of the SBRC 2005-IV Salao de Ferramentas. 2005.

2 Anglano C, Canonico M, Guazzone M, Botta M, Rabellino S, Arena S, Girardi G. Peer-to-Peer Desktop Grids in the Real World: The ShareGrid Project. Proc. of 8th IEEE International Symp. on Cluster Computing and the Grid, 2008. CCGRID'08. 2008: 621－626.

3 Thompson JD, Higgins DG, Gibson TJ. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Research. 1994(22):4673－4680.

4 Zhu MJ, Hu GW, Zheng QL, et al. Multiple sequence alignment using minimum spanning tree. Proc. of 2005 International Conference on Machine Learning and Cybernetics, 2005(ICMLC 2005). Guangzhou, IEEE Computer Society, 2005: 3352－3356.

5 Edgar RC. Muscle: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res, 2004 (23): 1792－1797.