高技术通讯 2009 年 第 19 卷 第 10 期:1014 ~ 1019

doi:10.3772/j.issn.1002-0470.2009.10.004

生物计算网格中的在线调度技术研究①

刘文懋② 张伟哲 张宏莉 方滨兴

(哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)

摘 要 研究了在异构网格环境下的生物应用集成,定义了服务的提供者、部署者和使用者三种用户角色,设计了网格环境下的服务和资源整合机制,重点实现了用户管理以及作业调度控制等功能。根据计算资源的异构特点,设计了多种启发式调度算法。考虑到生物计算应用的不同类型,提出了自适应调度算法,该算法根据应用的特点动态选择启发式调度算法。实验表明,非阻塞调度优于阻塞调度方式;自适应调度算法比静态的在线调度算法有更好的性能,而在异构的网络中,带宽优先调度算法的性能比其他静态调度算法性能更好。

关键词 生物信息学, 计算网格, 在线调度, BioLab

0 引 言

自 20 世纪 80 年代末以来,基因组测序数据迅猛增长,生物信息学(bioinformatics)[1]逐渐成为一门独立的新兴学科。适应这一发展趋势,复用现有的生物软件,加之引入新的生物软件,建立一个通用的生物信息计算平台,对于生物学研究有重要的作用和深远的影响。网格计算(grid computing)[2]能将游离的闲散计算资源集成为一个统一的计算平台,这一技术的出现,为生物信息学的大规模集成应用提供了可能。

国际上已完成了多项研究生物网格的项目,例如英国的 myGrid、欧洲的 EuroGrid、美国的 NCGrid 等。此外,Xu[3]等及 Gong[4]等着重研究了网格在生物计算方面的应用,如网格架构的建立和总体系统的设计(网格中的资源和作业的映射是一个难点),Braun[5]等研究了映射的分类,Ghafoor[6]等研究了异构网格系统的资源管理,Kim[7]等研究了在异构环境下有限时限制的作业调度,Attiya[8]等着重研究了异构环境下的调度可靠性问题,Muthucumaru[9]及 Ibarra 等[10]研究了在异构网络环境下的作业调度算法,Casanov[11]研究了在异构环境下的参数扫描算法,Amos[12]等、Yoon[13]等、Christoph[14]等研究了在线算法和在线调度的一些原理,胡玲玲[15]等研究了在线调度算法中的 QoS 需求。还有一些研究的是敏感的调度算法,如 Marchal[16]研究了网格环境下带宽最优化问题,Jones[17]研究了带宽敏感的机群的任务分配,Koukis[18]等研究了在带宽敏感的对称多处理器机群中的调度策略。但是,上述研究缺乏对生物计算的整个服务生命周期的整合及角色整合和面向终端用户的作业调度的研究。同时,在异构网格环境下,以前的调度主要以注重吞吐量的批处理调度为主,为用户敏感的在线调度研究较少。

本文研究了异构网格环境下的生物应用集成,设计了一种基于在线调度的生物信息学网格计算系统——BioLab。BioLab 拥有一个 7 × 24 小时不间断的门户站点,能够同时接受 100 个并发实时作业,提供无缝集成生物计算服务,提供统一可编程的接口,基本上建成了基于网格的生物信息系统与应用环境,可用于进行面向分布式网格环境的资源性能评价和任务调度研究。

1 系统设计

1.1 角色管理

从功能上分,BioLab 系统的用户角色有三种:服务的发布者、服务的部署者和服务的使用者。服务发布是指提供服务的源代码或可执行程序;服务部

① 863 计划(2006AA02Z334,2009AA01Z437),国家自然科学基金(60703014),973 计划(G2005CB321806),高等学校博士学科点专项科研基金(20070213044)和中国博士后科学基金(20070410263)资助项目。

② 男,1983 年生,博士生;研究方向:网格调度,信息安全,智能交通;联系人,E-mail: liuwenmao@pact518.hit.edu.cn

(收稿日期:2008-10-30)

署是指将发布的服务部署在相应的资源上,供终端用户使用;服务的使用是指寻找适合自己的服务,定位服务的资源,在资源上运行服务并获得结果。

从权限上分,BioLab 系统的用户有两类:普通用户和管理维护者。普通用户使用系统,发布、注册或使用服务;管理维护者负责控制用户权限、管理用户资料等,维护配置系统的运行参数,备份系统数据,维护系统的正常运行。

1.2 服务描述

本文中的服务皆指 Web 服务,即任何在 Internet 上使用标准化的可扩展标记语言(XML)消息机制。

服务有诸多属性,所有属性都可归为两类。一类是服务的固有属性,如服务版本号、服务描述、服务发布者首页及服务的特征码,这些属性可区分不同的服务,我们称之为服务的共性。另一类是服务的资源属性,服务可以被部署在异构的资源上,提供不同的应用。资源有不同种,如硬件资源、软件资源和数据库资源等[3]。在不同资源上的服务之间也需要被区分出来。故服务不可避免地带上了资源的属性,如资源地址、服务在资源中的存放位置等。这些特征并不是服务固有的,一个服务部署在不同资源上表现出的这些属性是不尽相同的。我们称之为服务的特性。

服务的共性是服务发布者赋予的,不可变动,而服务的特性是服务部署者提供的,不同资源上的服务特性不同。

1.3 服务发布

生物信息服务种类数量繁多,重用已有的生物计算程序,可以最小程度地降低系统本身的开发成本,同时增加潜在的生物计算用户的数量。但生物计算软件间有很大的差异,不能直接使用统一的模式使用这些软件。服务的发布和部署需解决如何使系统按照统一格式来发布服务的问题。

文献[19]中研究的是生物信息服务的内在任务逻辑形式,文中指出生物信息任务可以抽象成 6 类:单个、顺序、并行、迭代、条件和整合。在 BioLab 中,任务可以通过作业流的形式进行批提交,也可以通过标准的 Web Service API 进行任务的提交,故上述各种任务理论上都可以进行控制。作业流为用户事先定义的若干作业的集合,根据时间顺序排列,以文件形式提交,系统解析后依次处理。

BioLab 根据服务的共性,定义了服务共性描述文件,以 XML 格式规定了共性属性。服务的发布者在编写完服务软件的代码之后,提交服务代码及共性描述文件,在 BioLab 上发布服务。

1.4 服务部署

当服务发布之后,服务部署者可从服务生产商的官方站点下载服务程序,部署到自己的服务器上,将自己服务器上的服务部署信息登记到 BioLab 上,供终端用户使用。

部署者首先需要安装生物计算的服务软件,将该服务加入 Globus Container 中,然后将服务信息存入在 Ganglia 的数据库中,最后生成服务的特性文件,提供服务的特性部署信息,部署信息包括服务的特性,以标准的 XML 格式发布。

1.5 服务检测

服务的检测是指检测部署在资源上的服务的状态,主要分为硬件资源检测和服务资源检测。

硬件资源监测指系统定时监测机群中的节点的增删情况,并更新节点的硬件信息。

服务资源监测指系统定时启动监测程序,监测资源列表中的服务是否存在或是否可用。

如遇到不可用资源,系统会以逐步递增的时间间隔继续检测该资源,如间隔超过一定阈值则将其废弃。

2 作业调度和状态检测

2.1 网格模型

本文研究的网格环境有多个非专用机群组成,每一个机群包括若干个主机,这些机群和主机共同构成了一个巨大的计算资源。这些主机和机群的信息可以通过 Ganglia 软件获得。

网格中的作业调度器独占一台服务器,并不包含任何计算资源,调度器根据调度结果将作业分发到相应机群的主机上。

在设计算法时,我们需要考虑到以下基本方面的问题:

(1) 各个计算节点之间的资源被共享,不同应用程序的能力动态变化(包括延时和服务质量 QoS)。

(2) 网络中各个计算节点是异构的,对于同样的应用程序性能可能不同

(3) 虽然程序彼此独立,但是它们之间可能会共享远程大文件,大文件的传输会增加作业总完成时间。

2.2 调度算法模型

在异构的网格环境下的机群任务调度是一个

NP 完全问题,至今没有最优的算法能够解决这一难题。所以一般会采用启发式算法进行调度,其中最常用的有批处理调度算法和在线调度算法。

批处理调度算法的思想是当一个作业到达调度器时,它并非立刻被调度,而是被放置到一个作业集合中,调度器每隔一段时间检查一次该作业集合。这些相互独立的作业的集合在调度过程中被称为元作业。而在线调度算法的思想是当作业一旦到达调度器,就被调度到一个资源上。

由于 BioLab 是终端用户敏感的具体应用,故系统采用注重任务响应时间而非系统吞吐量的在线处理调度算法。在线算法框架(除随机算法)如下:

```
OnlineSchedule(hosts)
{
    chosenHost = null;
    for(i = 1; i < hosts.length; i ++ )
    {
        if(! isAvailable (hosts[i]))
            continue;
        if(compareHost(hosts[i], chosenHost))
            chosenHost = hosts[i];
    }
    return chosenHost;
}
```

Hosts 是每次可调度的计算资源集合,OnlineSchedule 返回每次调度结果。具体的在线调度子算法可在函数 compareHost 中定义,这些算法包括最快处理器速度优先调度算法(max cpu speed)、最大处理器总量优先调度算法(max cpu total)、最大空闲内存容量优先调度算法(max mem free)、最大带宽优先调度算法(max band)、随机调度算法(random),其中随机算法只需随机选择一个可用资源即可,无需进行比较操作。

上述算法中,最快处理器速度优先调度算法适合调度处理器密集型的作业;最大处理器总量优先调度算法考虑了处理器速度和数量,适合处理并行化较好的作业;最大空闲内存容量优先调度算法适合处理内存消耗大的作业;最大带宽优先调度算法适合处理输入输出文件较大作业或网络异构的资源。

2.3 自适应在线调度

在实际应用中,上述启发式算法在不同的环境下有不同的性能,例如处理器优先算法在高速网络中性能高于其他算法,而带宽优先算法在广域网中的性能提高明显,同时,生物应用也十分繁多,有处理器密集型、内存密集型、IO 密集型和带宽密集型等类型。所以采用单一的调度策略很难适应复杂的现有复杂网络和生物应用。

我们提出了自适应的在线调度算法(adaptive),它可为生物应用标记的不同属性,选择最适合该属性的调度算法,以获得最适合的资源。例如,当前作业是内存密集型的,那么它将采用最大空闲内存容量优先调度算法,这种调度,能够启发式地最小化作业的运行时间。启发式调度算法如下:

```
AdaptiveSchedule(job, hosts)
{
if(job.type == JobType.CPU _ SPEED)
    return MaxCPUSpeedSchedule(job, hosts)
else if(job.type == JobType.CPU _ TOTAL)
    return MaxCPUTotalSchedule(job, hosts)
else if(job.type == JobType.FREE _ MEM)
    return MaxFreeMemorySchedule(job, hosts)
else if(job.type == JobType.BAND)
    return MaxBandSchedule(job, hosts)
else
    return null;
}
```

3 实验

3.1 硬件实验条件

在实验中,我们采用真实的实验环境研究作业调度,这与以往的模拟和仿真[20]有很大的区别:实验时间是真实作业运行的时间,所费时间远大于模拟和仿真的运行时间,但是实验数据更加真实可信。

我们进行的实验是真实系统运行一个作业流,从运行的情况中获得实验数据。实验环境中,有 4 台服务器,操作系统为 RH AS4,处理器、内存和磁盘空间相异。实验的生物应用是 EMBOSS 软件包中的 getorf、seqret 及基因序列分析软件 hmmsearch 和 hmmpfam。

3.2 阻塞调度和非阻塞调度的关系

给定自然数 $N > 0$,假设实验环境有 3 台计算资源 A、B 和 C,调度器在调度之前会依次检查每个资源,如果调度器检查到某个资源已有大于 N 个作业,则将其从可用资源列表中删除,检查完毕之后生成一个可用列表,调度器在该可用资源列表中继续使用某一个在线调度算法进行调度,提交作业,这种

调度策略我们称之为非阻塞调度。相反,如果调度器没有检查资源上是否存在大于 N 个作业的过程,那么我们称之为阻塞调度。

我们设计了 4 个有 40 个作业的作业流序列,作业的到达满足泊松分布。这 4 个作业流分别满足 P(20)、P(40)、P(80)和 P(20),前 3 个调度策略为非阻塞,第 4 个调度策略为阻塞。

我们对最快处理器速度优先调度算法、最大空闲内存容量优先调度算法和随机调度算法进行了比较,实验结果如图 1 所示。

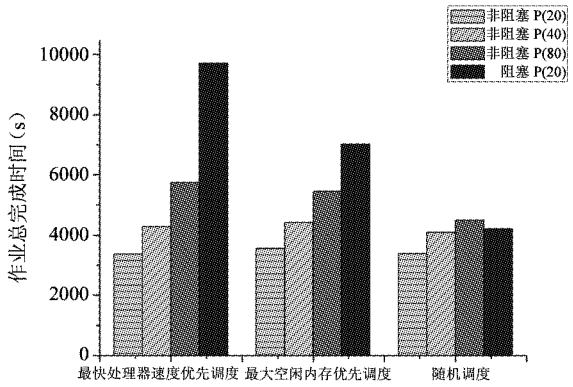

图 1　非阻塞和阻塞调度对比

从图中可见,在平均到达时间 20s 的条件下,在较稳定的调度算法中,例如最快处理器速度优先调度算法,非阻塞的调度策略比阻塞的调度策略快 2.87 倍,次稳定的最大内存调度算法,非阻塞调度比阻塞调度快 1.97 倍,而较均匀的随机调度算法,非阻塞调度比阻塞调度快 1.24 倍。

以上数据可证实,在阻塞的策略下,稳定的调度算法会导致更长的作业运行时间,而随机的调度算法,会有比较短的作业运行时间。

随着平均到达时间的增长,作业的总运行时间也会增加,但即使从 20s 增加到 80s,非阻塞调度的效果也好于阻塞调度。

3.3　作业平均到达时间和作业总运行时间的关系

由上可知,作业平均到达时间增加,可能会造成作业的总完成时间的增加。我们设计了如下 3 组实验进行验证。这 3 组实验的作业流分别满足泊松分布P(20)、P(40)和 P(80),调度策略均为非阻塞。

实验对最快处理器速度优先调度算法、最大处理器总量优先调度算法、最大空闲内存优先调度算法、最大带宽优先调度算法、随机调度算法进行了比较,结果如图 2 所示。

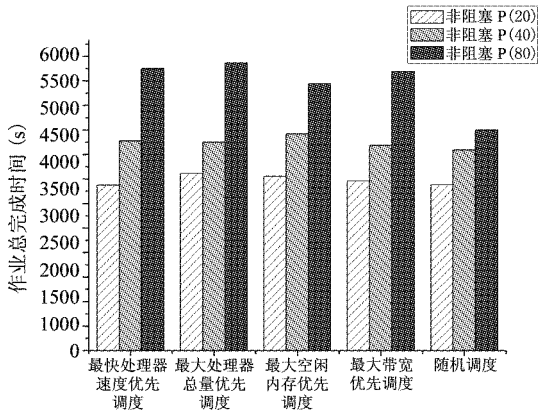

图 2　作业平均到达时间和作业总运行时间的关系

将第二组的结果除以第一组,再将第三组的结果除以第二组,我们得到如表 1 所示的结果。

表 1　作业平均到达时间对作业总完成时间的影响

	P(40)/P(20)	P(80)/P(40)
最快处理器速度优先调度	1.269356274	1.34330451
最大处理器总量优先调度	1.174177495	1.380974806
最大空闲内存优先调度	1.244300591	1.232074191
最大带宽优先调度	1.209537572	1.361051374
随机调度	1.20892435	1.0989978

可以看出,随着任务平均达到时间的增加,作业总运行时间会增加,但两者并非成正比关系,而是一种缓慢增加的趋势。因为作业总的完成时间不仅涉及到任务平均到达时间,还涉及到作业的调度时间和执行时间。因为作业的执行时间比较稳定,所以,任务平均到达时间的增加造成总完成时间增长的趋势会变缓。

3.4　在线调度算法的比较

本文对我们提出的各种在线调度算法进行了比较。我们选用最快处理器速度优先调度算法、最大处理器总量优先调度算法、最大空闲内存容量优先调度算法、随机调度算法和自适应调度算法。

为研究异构网格环境下的调度算法性能,我们在真实实验数据基础上进行模拟。模拟环境有处理器速度、处理器总量、内存容量和带宽占优服务器各两台。共有 4 个作业流,均匀随机出现处理器速度、处理器总量、内存容量和带宽密集的作业,满足泊松分布 P(10)、P(20)、P(30)和 P(40)。因非阻塞调度的性能较优,实验均为非阻塞调度,对比结果如图 3 所示。

可见,自适应的调度算法在所有的作业平均到

达时间下都有最小的作业总完成时间，比最差的随机算法性能高 12.16%；随机算法在随着作业平均到达时间增加时，性能下降最快，因为在系统不繁忙时，不同的调度算法能够启发式地将作业调度到相对较优的资源上，而随机算法没有考虑作业和资源的特性，所以性能下降。而带宽优先的算法的性能最接近自适应的算法，因为在异构的不同机群调度中，网络延迟是非常重要的因素，而生物计算中有很多应用产生数百兆的输出文件，这样的应用如采用带宽优先算法，能够最大程度地降低作业的总完成时间。

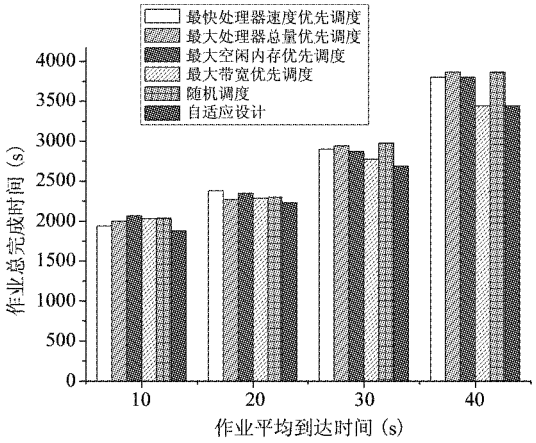

图 3 在线调度算法对比

4 结 论

针对生物计算服务的特点，BioLab 定义了服务的共性和特殊属性，属性的区分使得系统能够在一个通用的平台上描述所有的异构服务，同时，在用户层面建立了服务提供者-服务部署者-服务使用者三层结构，根据用户模型建立了相应的功能模块。分层的角色设计，使系统能够集成更多的用户和资源。根据服务的属性对系统的功能进行划分，目的是建立一个统一的服务发布、部署和运行的平台。

在算法研究中，我们采用终端敏感的在线调度算法。实验表明，在较大的作业平均到达时间条件下，非阻塞的调度方式的作业总完成时间比阻塞的调度方式小；当作业平均到达时间增加时，作业总运行时间也会增加，但速度会变缓；带宽优先的调度算法在网络异构的环境性能最优。我们提出的自适应的调度算法综合考虑了不同资源的优势和不同应用的特点，且动态选用不同的启发式调度算法，实验表明，这种算法的性能比单一的启发式在线调度算法好。

参考文献

[1] Wikipedia. Bioinformatics. http://en.wikipedia.org/wiki/Bioinformatics: Wikimedia Foundation, 2008

[2] Foster I. The grid: a new infrastructure for 21st century science. *Physics Today*, 2002, 55: 42-47

[3] Xu G S, Luo Y, Yu H S, et al. An approach to SOA-based bioinformatics grid. In: Proceedings of the IEEE Asia-Pacific Conference on Services Computing (APSCC 2006), Guangzhou, China, 2006. 323-328

[4] Gong X J, Nakamura K, Yu H, et al. BAAQ: an infrastructure for application integration and knowledge discovery in bioinformatics. *IEEE Transactions on Information Technology in Biomedicine*, 2007, 11(4): 428-434

[5] Braun T D, Siegel H J, Beck N, et al. A taxonomy for describing matching and scheduling heuristics for mixed-machine heterogeneous computing systems. In: Proceedings of the 17th IEEE Symposium on Reliable Distributed Systems, Washington D.C., USA, 1998. 330

[6] Ghafoor A, Yang J. A distributed heterogeneous supercomputing management system. *IEEE Computer Society*, 1993, 26(6): 78-86

[7] Kim J K, Shivle S, Siegel H J, et al. Dynamic mapping in a heterogeneous environment with tasks having priorities and multiple deadlines. In: Proceedings of the 17th International Symposium on Parallel and Distributed Processing, Washington D.C., USA, 2003. 98

[8] Attiya G, Hamam Y. Reliability oriented task allocation in heterogeneous distributed computing systems. In: Proceedings of the 9th International Symposium on Computers and Communications, Washington D.C., USA, 2004. 68-73

[9] Maheswaran M. Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing systems. In: Proceedings of the 8th Heterogeneous Computing Workshop, San Juan, Puerto Rico, 1999. 30

[10] Ibarra O H, Kim C E. Heuristic Algorithms for scheduling independent tasks on nonidentical processors. *Journal of the ACM (JACM)*, 1977, 24(2): 1-4

[11] Casanov H, Zagorodnov D, Francine B, et al. Heuristics for scheduling parameter sweep applications in grid environments. In: Proceedings of the 9th Heterogeneous Computing Workshop, Cancun, Mexico, 2000. 349

[12] Amos F, Gerhard W. Online Algorithms: The State of the Art. New York: Springer, 1998. 436

[13] Yoon H J, Lee D Y. Online scheduling of integrated single-wafer processing tools with temporal constraints. *Semiconductor Manufacturing*, 2005, 18(3): 390-398

[14] Christoph S, Herbert W, Marco P, et al. Online scheduling and placement of real-time tasks to partially reconfigurable devices. In: Proceedings of the 24th IEEE International Real-Time Systems Symposium, Los Alamitos, CA, USA, 2003. 224-225

[15] 胡玲玲,杨寿保,张然美等.网格中效用驱动的多维 QoS 在线调度机制.华中科技大学学报,2007,35(1):57-58

[16] Marchal L. Optimal Bandwidth Sharing in Grid Environments. In: Proceedings of the15th IEEE International Symposium on High Performance Distributed Computing, Paris, France, 2006. 144-155

[17] Jones W M. Characterization of bandwidth-aware meta-schedulers for co-allocating jobs across multiple clusters. *The Journal of Supercomputing*, 2005, 34(2): 135-163

[18] Koukis E, Koziris N. Memory bandwidth aware scheduling for SMP cluster nodes. In: Proceedings of the 13th Euromicro Conference on Parallel, Distributed and Network-Based Processin, Los Alamitos, CA, USA, 2005. 187-196

[19] Xu G S, Luo Y, Yu H S, et al. Study on Bioinformatics Grid Application and its Supporting Environment. In: Proceedings of the 5th International Conference on Grid and Cooperative Computing Workshops (GCCW 2006), Washington D.C., USA, 2006. 375-380

[20] 查礼,徐志伟,林国璋等.基于 Simgrid 的网格任务调度模拟.计算机工程与应用,2003,34(14):90-92

BioLab: a bioinformatics grid computing system based on online scheduling

Liu Wenmao, Zhang Weizhe, Zhang Hongli, Fang Binxing

(Department of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001)

Abstract

In this paper bioinformatics application integration in the heterogeneous grid environment is studied, and the BioLab, a bioinformatics grid computing system based on online scheduling, is designed. Three roles are defined in the system: service providers, service distributers and service users. The integrated mechanism for services and resources is designed, and the modules such as the user management, the job scheduling and the job management are implemented. Several heuristic scheduling algorithms are designed according to the heterogeneous features of computing resources. The adaptive scheduling algorithm, which chooses heuristic scheduling algorithms dynamically based on alterative application characteristics, is proposed after considering different types of biological applications. The experiments show that in the large average job arrival time, unblocked scheduling is better than blocked scheduling, and the adaptive scheduling algorithm outperforms the static scheduling algorithms, while in the heterogeneous grid the bandwidth-first scheduling algorithm outperforms the other static scheduling algorithms.

Key words: bioinformatics, computing grid, online scheduling, BioLab