并行数字地形分析的容错与负载均衡研究

赵 菁[1],窦万峰[1,2]

(1. 南京师范大学 计算机科学与技术学院,江苏 南京 210097)
(2. 江苏省信息安全保密技术工程研究中心,江苏 南京 210097)

[摘要] 并行数字地形分析是将数字地形分析与并行计算相结合的分析方法. 总结了并行数字地形分析的关键技术,介绍了目前并行计算中常用的并行计算平台,并提出了适用于数字地形分析的并行计算平台,概括了数字地形分析中现有的并行化技术,重点介绍了当前的研究热点:容错技术和负载均衡策略,并提出了存在的问题,作为将来可能的研究方向.

[关键词] 并行计算,数字地形分析,并行化,容错机制,负载均衡
[中图分类号] TP302.8 [文献标志码] A [文章编号] 1672-1292(2011)03-0065-08

Research on Fault Tolerance and Load Balance in Parallel Analysis of Digital Terrain

Zhao Jing[1], Dou Wanfeng[1,2]

(1. School of Computer Science and Technology, Nanjing Normal University, Nanjing 210097, China)
(2. Jiangsu Research Center of Information Security and Privacy Technology, Nanjing 210097, China)

Abstract: Parallel digital terrain analysis is a method of combining digital terrain analysis and parallel computing. This paper summarizes the key technologies of parallel analysis of digital terrain. Firstly, typical parallel computing platforms are introduced in this paper and the suitable one for parallel analysis of digital terrain is then proposed. Secondly, this paper sums up the parallelization methods of parallel analysis of digital terrain. Finally, the fault-tolerant mechanism and load balance in parallel computing are focused in this paper and some problems are put forward so as to make them as possible research directions in parallel digital terrain analysis.

Key words: parallel computing, digital terrain analysis, parallelization, fault tolerant mechanism, load balance

随着计算机技术的飞速发展,几乎所有学科都走上定量化和精确化的道路,从而产生了一系列计算性的学科分支,即科学计算[1]. 科学计算的主要特点是数据量巨大,使用传统计算机已不能满足计算需求,因此导致了并行计算的出现.

数字地形分析是科学计算中的一个重要分支,且作为地理信息科学和技术发展的重要研究内容,已经深入各行各业. 随着各类新型传感器的出现,数字高程模型(Digital Elevation Model,DEM)数据呈几何级数增长,传统的计算机处理技术已不能有效处理 DEM 数据,所以将数字地形分析与并行计算融合是提高数据处理性能的有效途径.

传统的高性能计算一般采用基于向量机或并行处理器等超级计算机硬件. 随着网络技术的迅速发展,还出现了集群、分布式、网格、普适计算等新型并行计算架构[2]. 这些技术的出现使得并行计算具有高速、低投入等特点,已日渐成为科学计算中的主流技术. 目前很多学者正致力于将数字地形分析与并行计算相结合,但对于数字地形并行分析中的容错机制与负载均衡问题的研究还有待加强.

本文以数字地形分析为对象,介绍了现阶段并行计算的现状,概述了并行计算平台的搭建,总结了数字地形分析算法的并行化,重点讨论了并行计算中的容错机制、负载均衡策略以及二者的发展趋势.

收稿日期:2011-06-01.
基金项目:国家"863"基金(2011AA120304).
通讯联系人:窦万峰,博士后,教授,研究方向:分布协同软件工程和计算机支持的协同工作(cscw). E-mail: douwanfeng@njnu.edu.cn

1 并行计算的平台架构

1.1 并行计算的硬件基础

大规模科学计算需要依赖高性能的并行计算平台. 典型的并行计算机按照指令和数据可以分为单指令多数据并行计算机(Single-Instruction Multiple-Data,SIMD)和多指令多数据并行计算机(Multiple-Instructions Multiple-Data,MIMD),典型代表有阵列处理机和向量处理机;按照存储方式可以分为共享存储多处理机和分布式存储多处理机,典型的代表有集群系统.

虽然大规模并行处理机(Massively Parallel Processing,MPP)能够有效地解决对海量数据的处理,但造价较高,且随着个人计算机的普及和计算机硬件技术的发展,集群系统因其造价低廉、易扩展的特性逐渐成为并行计算平台的主流硬件结构. 集群系统由一组独立的计算机组成,这些计算机可以是工作站或者个人计算机. 这组计算机之间通过网络互连,如高速的以太网、局域网等. 多核计算机的出现使得集群系统的性能得到了进一步的提升,现阶段使用较多的集群系统是采用多核计算机通过局域网相连的结构.

1.2 并行计算的软件基础

并行计算平台架构的另一个重要组成部分就是软件基础. 软件基础主要有操作系统和支持并行计算的软件程序. 目前支持并行计算的软件程序按照并行编程模型可以分为基于数据并行和基于消息传递两种. 基于数据并行的典型代表是HPF(High Performance Fortran),基于消息传递的典型代表是MPI(Message Passing Interface)和PVM(Parallel Virtual Machine Computing). 基于数据并行和基于消息传递的软件都可以适用于基于共享存储的计算机,而后者同时也适合于基于分布式存储的计算机. 相比而言,基于消息传递的并行计算平台比基于数据并行的并行计算平台灵活性、可扩展性以及可控性更好,所以在实际应用中使用得较多.

1.3 适用于数字地形分析的并行计算架构

数字地形分析的程序架构简单,但DEM数据量庞大,所以进行并行计算时,节点间的通信量较大,需要功能强大的通信机制做支持,故采用基于消息传递的软件平台较为合适. 由于MPI与C和Fortran语言的结合,提供了强大的消息传递机制,并为用户提供了方便的库函数调用,故很多科学计算的平台都采用了MPI+集群系统的模式,如文献[4~8]都对基于MPI的集群系统下的编程进行了深入的研究.

随着多核处理器体系结构的普及,越来越多的编程者开始使用MPI+OpenMP的混合编程模式. OpenMP是用于共享存储并行系统中的多线程程序设计的一套指导性注释(Compiler Directive),可以同时实现节点间和节点内的两级并行化. 但对于大数据量的并行计算,节点间的频繁通信可能会降低并行的效率,所以需要合适的并行化算法以及合理的任务拆分.

2 基于数字地形分析的并行化

数字地形分析(Digital Terrain Analysis,DTA)是在DEM上进行地形属性计算和特征提取的数字信息处理技术[9]. 目前,数字地形分析的并行化主要集中在对DEM数据的构建和数字地形分析算法的并行化.

2.1 数字高程模型构建的并行化

最常见的DEM数据组织方式有规则格网结构、不规则三角网结构和等高(值)线结构. 规则格网和不规则三角网是目前DEM的主要数据模型. 尽管各种DEM构建技术均作了不同方面的改进,但DEM数据庞大,采用传统的单机处理仍然不是最理想的方法,采用并行计算的方法可以有效地解决这个问题. DEM构建的并行计算的研究主要有内插生成网格DEM、等高线的并行计算、三角网的并行生成.

在规则网格DEM构建并行化方面,众多学者提出了不同的算法,主要分为整体内插、部分分块和逐点内插法. 针对网格计算环境下分布式DEM构建并行计算,Wang[10]使用Globus Toolkit,通过基于四叉树的域分割算法和静态任务管理算法,实现了在网格计算环境下并行空间内插算法的开发和研究. 对于三角网的生成,学者们利用并行计算,提出了分层次递进的结果融合的策略[11]、用增量构建的方法将凸壳边界区域进行划分[12]以及基于GPU的三角剖分的并行算法[13].

2.2　数字地形分析算法的并行化

DEM 地形分析方法包括坡面地形因子提取、特征地形要素提取、地形统计分析以及基于 DEM 地学模型分析等 4 个方面. 数字地形分析并行计算具有数据密集和计算密集的特征,所以需要从数据并行、任务并行、算法并行等方面来考虑.

在可视性分析方面,Carsten [14] 提出在异构地形上寻找最短路径的新型可扩展的并行算法,对 DEM 数据进行静态划分,采用动态数据广播的方式进行数据的分发,对 A^* 算法进行并行化,使用全局索引数组记录已访问和未访问节点,从而降低通信时间,提高并行的效率. Kidner [15] 提出了一种基于数据并行的并行可视性分析算法,各个节点采用相同的同时算法,最后主节点负责结果的融合.

在流域网络提取方面,则主要是对 D8 算法进行了并行化处理 [16] ,或对 DEM 数据进行分割采用 SIMD/SPMD 的并行架构来提高流域网络提取算法的效率 [17,18] ,以及利用 NVDIA CUDA 技术和多线程编程技术提出基于 CUDA 的并行流域网络提取算法 [19] .

3　并行计算中的容错机制

随着计算机硬件技术的发展以及计算能力的增强,高性能计算的效率得到了显著的提高. 但伴随计算机系统的规模不断扩大,计算的可靠性问题显得日益突出. 根据最新的 Top500 排名,目前已经有 3 台超级计算机系统的处理器/处理器核突破了 20 万个 [20] . 而与此同时,大规模并行系统的平均无故障时间(Mean Time Between Failures,MTBF) 下降到了若干小时的量级 [21] . 因此,在通过硬件技术不断提高并行计算加速比的同时,采取相应的容错机制来保障计算的可靠性显得非常必要.

计算机故障大致分为两类: 硬件故障和软件故障. 硬件故障是由于机器老化、突然断电等突发状况引起的,又可以划分为永久故障和暂时故障,可通过相应的检测技术进行恢复. 软件故障是由于软件设计本身的缺陷而引起的错误,这种故障是无法绝对避免的.

针对计算机故障,主要有两种方法来解决: 一种是避免故障的发生,这在现实中很难做到; 另一种就是进行容错,即在计算机硬件或软件发生故障的情况下,计算机系统能够检测出故障并采取相应的措施进行恢复,不影响计算机的正常工作.

容错最基本的方法是冗余技术,冗余的思想最早是由 Neumann JV [22] 提出的. 冗余包括硬件冗余、软件冗余、时间冗余和信息冗余. 本文主要讨论时间冗余和软件冗余. 在时间冗余技术中,目前使用得最多的是检查点技术. 在软件冗余中,主要使用“多样性”的冗余来解决软件本身出现的故障 [23] ,例如多版本技术和恢复块技术等.

3.1　检查点技术

大规模科学计算的计算量巨大,运算时间较长,若出现故障,重启整个计算,将带来很大的资源耗损和时间浪费. 检查点机制就提供了这样一种方法: 在程序的关键位置保存当前程序的状态及数据信息至检查点,在发生故障时,将程序回卷到上一个检查点处,对上一个检查点到故障处的一段程序进行重新运行,也即复算. 这样可以大大缩短容错处理的时间.

检查点技术一般分为系统检查点(System Level Checkpoint,SLC) 和用户级检查点(Application Level Checkpoint,ALC) . 系统检查点是系统自动在一定的时间间隔将系统的状态保存到坚固存储器上,而用户级检查点则允许用户在程序中自定义检查点. 对比两者,ALC 比 SLC 更具灵活性,并且能够更好地降低系统的开销,是当前研究的热点.

3.1.1　检查点的设置

在对用户级检查点的研究中,如何降低保存检查点的系统开销是研究重点. 降低检查点的系统开销的关键技术在于对并行程序中的变量进行分析.

富弘毅等人 [20] 提出一种基于扩展数据流分析的应用级检查点机制,提出了基于 OpenMP 并行控制流图,据此对并行程序中的变量进行扩展分析得到活跃变量集合,并区分程序中的共享变量和私有变量,不同性质的变量采用不同的进程来保存.

Yang 进一步讨论了活跃变量的确定 [24,25] ,通过变量在进程内的定值-引用链以及进程间的通信定值-引用链来确定变量是否为活跃变量,并将变量分为全局活跃变量、部分活跃变量以及死变量,从而找出每

次检查点需要保存的变量的最小集合,有效减少了检查点的存储开销. 文献[24]在此基础上还架构了一个预编译系统来自动生成检查点.

3.1.2 检查点的同步

在并行计算中,进程之间存在频繁的通信,出现故障时,需要所有相关进程都要恢复到上一个正确状态,就需要在检查点设置时进行同步. 一般使用的方法有阻塞式和非阻塞式消息机制. 阻塞式方法属于同步通信,在大规模计算中等待所有节点进行同步将耗费很多时间,不利于提高系统的并行效率. 而非阻塞式属于异步通信,更加灵活.

Shang 等人[26]提出了一种基于数字流的集群计算容错机制,采用 Agent 来维护节点服务器列表、记录系统的工作状态以及负责系统的负载均衡,引入 Coordinator 来发送检查点建立请求,采用阻塞式消息机制来保证检查点的一致性.

3.1.3 检查点的存储

对于大规模的科学计算,检查点的数据量也很惊人. 检查点的存储方式分为集中式和分布式. 检查点集中存储,读取方便,但易造成检查点 I/O 的瓶颈. 检查点分布式存储,解决了检查点大量 I/O 带来的阻塞问题,但是远程访问检查点则会有更多的通信开销.

周恩强等人提出了将检查点分布式存储[27],建立分布式虚拟检查点文件系统,可以解决检查点集中存储带来的 I/O 问题,同时优化分布存储时检查点的读取.

3.1.4 检查点技术与并行计算

检查点技术在发生硬件故障以及系统故障时能够有效进行解决,它涉及到指令的复制或者程序的回卷,需要保存累加器、状态寄存器等涉及到编译级的变量. 对于大规模并行计算,如数字地形并行分析,检查点技术存在以下问题:(1)同步所有节点间的检查点,数据量过大,存在 I/O 瓶颈;(2)目前大多数并行程序采用了 MPI 编程模式,而 MPI 编程的特点是主进程通过 if 语句来实现任务分发到各个并行节点上,若出现故障进行程序回卷,在编程实现上存在困难;(3)检查点技术采用的是时间冗余的思想,对于数字地形分析这样的大规模科学计算来说,用户需求的是比较迅速的结果呈现,检查点技术无法很好地满足,故杜云飞等人提出了基于复算的容错[28].

3.2 软件冗余

在数字地形分析中计算错误的发生,需要借鉴软件冗余的方法来解决.

传统的软件冗余是使用多版本程序设计和可恢复模块技术[29]. Elmendor[30]提出了多版本程序技术并由 Avizienis[31]等人进行了完善. 多版本程序设计是指对同一任务采用相同的算法、不同的开发模式和开发人员,各个版本的程序并行运行,采用投票表决的方式进行错误检测,适合实时故障. 该方法对于单一结果的程序比较有效,但对有多种结果的程序不能准确判断. 可恢复模块是指对同一任务采用不同的算法,多使用串行方式[29]. 通过与不变式进行比较,适合有多种结果的任务. 由于在发现错误后才调用另一个算法进行再次计算,因此在时间上存在缺陷. 这两种冗余策略不同之处在于检错机制上,Manic[32]等人在传统投票机制的基础上提出了“模糊选举”的方法,从而使多版本技术更好地适用于结果不唯一的计算. 对于并行数字地形分析,往往结果存在一定的误差区间,所以“模糊选举法”是较为适合的错误检测策略.

刘心松等人[33]提出由具有多版本冗余节点组成的分布式系统,并且将分布式冗余系统与传统的冗余系统相比较,论证了分布式冗余系统的优越性,同时验证了三冗余策略已经足以满足容错需要. 但在并行计算中,可靠性是以资源和时间为代价,仅从理论上进行分析还不够. 对于不同的硬件条件,不同的冗余副本数目对性能影响不同,需要具体地从资源约束和时间约束上进行比较论证.

到目前为止,大量的以硬件冗余技术为主的容错方案,对于大规模计算来说,实施简单,但成本较高,且这些方案只关心了冗余机制的可用性,缺少对冗余机制性能的探讨. 曹小华等人[34]为了解决这个问题,将容错问题与负载均衡问题相结合,提出以往的冗余节点往往是被分配任务,而论文将冗余节点也看作主节点,以主动的方式请求任务的分配,从而提高了冗余系统的性能.

目前对于软件冗余的研究主要集中在冗余策略上,表现为如何降低冗余度以及更有效地检测错误. 在并行计算中,软件本身的故障并非主要,更需要考虑的是由于多个进程对资源的竞争而导致的计算错误,所以对冗余副本个数的分析以及冗余容错与负载均衡相结合构成的两级调度显得尤为重要.

4　并行计算中的负载均衡调度

计算机软硬件的发展使得集群系统变得更加普及,但这些系统潜在性能的实际利用通常仅为 1% ~ 10% [35]. 在这类系统中,负载的不均衡常导致系统的并行效率低下. 并行计算中的均衡调度策略的优化,能够显著地提高并行计算的加速比.

并行计算中的负载均衡策略可以分为静态和动态两类[35],静态负载均衡策略是指在任务调度前将任务划分好,而动态负载均衡策略则是在运行中实时进行任务分配.

4.1　静态调度

由于并行计算是将原有的串行算法进行并行化,需要分析系统各任务间的关系、原有算法语句中更细粒度的语句段或之间的关系,很多文献针对该问题提出了任务依赖图[37-40]. 任务之间的依赖关系是由控制依赖和数据依赖引起的[37]. 在任务依赖图中,采用带权节点来表述任务,节点权重表示该项任务的计算量,采用带权重的弧来连接有依赖关系的节点,弧权重表示节点间的通信量. 郭龙等人[37]给出了具有数据依赖关系的任务依赖图的构造方法,并给出了相关的算法描述,但是没有考虑依赖图中的权重问题.

张爱清等人[38]在任务依赖图的基础上,提出了新的截弧优先策略的顺逆交替迭代调度算法,对原有的任务依赖图中的节点先进行最晚完成时间的计算完成一次逆序迭代,再对节点进行最早完成时间计算完成顺序迭代,从而得到最优的任务调度方式.

在任务调度均衡中还有一个值得关注的问题就是任务的粒度划分,杜建成等人[39]指出,由于并行计算中各个任务之间会存在通信,如果任务的粒度过小,会导致通信量过大,从而降低并行的效率,并针对此提出了任务的合并策略. 并行任务的粒度是需要针对实际问题来划分的,任务的粒度可以是算法级、语句级,甚至是指令级. 到目前为止,并没有相关的文献结合带权的任务依赖图对任务粒度进行建模量化分析,从而得出最优的粒度划分策略.

4.2　动态调度

动态调度均衡策略相对于静态调度,能更充分利用系统资源,更实时地对系统进行负载均衡. 动态调度策略根据控制位置主要分为分布式、集中式和混合/层次 3 种策略[35]. 分布式策略中最常见的为近邻法,通过邻居节点之间的负载交换来达到负载均衡[36]. 在集中式策略中,有一个主节点负责收集所有节点的负载信息,并根据全局的资源做出相应的均衡策略,该方法中主节点就会成为系统的瓶颈. 混合/层次策略是将分布式策略与集中式策略相结合,效率要比前两者高. 向建军等人[42]对实时集群系统中的节点进行建模,用负载当量来量化节点的负载,根据轮转式任务调度、最少任务分配法来进行系统的负载均衡,不足之处是采用了全局集中式策略.

在众多学者研究实时任务的调度问题时,也有学者将目光转向了容错与均衡调度相结合的问题. 在大规模并行计算中,检查点技术的时间开销包括建立检查点的时间以及故障恢复的时间,降低了并行计算的加速比,因此杜云飞等人[29,41]提出了一种新的容错并行复算算法,在一个进程发生故障时将发生故障的进程的任务进行划分,分发给当前空闲的 n 个进程进行计算,这样就将复算的时间降到原来的 $1/n$. 因此可以考虑将负载均衡和容错调度融合以有效降低容错恢复的时间.

5　结语

本文简单介绍了数字地形分析中的并行化,重点讨论了容错调度和负载均衡,但尚有未尽之处需进一步探讨:

(1) 数字地形分析的并行计算主要集中在算法并行化以及 DEM 数据的拆分策略上,缺少针对数字地形分析并行计算中的负载均衡调度和容错处理机制的研究.

(2) 在故障检测上,现阶段大多数研究都集中在检查点的建立上,而检查点机制虽然对系统故障和硬件故障较有效,却不能发现计算错误,所以将时间冗余和软件冗余的思想结合,通过软件冗余中的投票机制来检测节点中的计算错误,而程序复算的过程可以采用检查点技术中的变量保存的方法来进行.

(3) 对于冗余机制,并行数字分析的大规模科学计算、冗余副本的建立会带来很大的系统开销,那么针对哪些关键节点进行冗余并且采用二冗余还是三冗余,目前没有相关文献进行定量分析. 此外,对于冗

余机制中发生故障后,采用全部重新复算还是从上一个正确位置开始复算,也值得进一步通过定量分析进行探讨.

(4) 在并行计算中,粒度的划分是动态调度的前提,粒度过粗,达不到并行的效果,而粒度过细则会导致频繁通信而降低并行效率.在并行数字地形分析中,粒度的划分涉及到数据(数据粒度)、任务(任务粒度)以及计算环境(结构粒度),所以需要进行相关的粒度建模,并对其进行定量分析,从而优化负载调度.同时,粒度模型对两级调度问题也至关重要.两级调度是指在系统调度的基础上,在发生故障或错误时进行容错调度,也即在发生故障时,可以将原有的粒度进行细化,分配给系统中的空闲节点进行复算.那么原有粒度的选取就决定了容错调度时如何细化粒度.

(5) 随着多核计算机的出现,节点内并行加节点间并行的机制越来越广泛,但针对这种新型架构的容错机制以及负载均衡的研究还比较少.由于节点内的通信较少,故可根据任务依赖图对任务进行分层,制定相关的任务调度规则,并根据负载均衡以及时间效率需求来对各个任务进行调度.

(6) 目前,有学者使用构建任务依赖图的方法来指导算法的并行化,但很多任务依赖图仍是对数据流程图的一种变形.在数字地形分析中,不仅各个地形计算因子紧密相关,还存在大量的数据依赖关系,因此需结合数据依赖关系研究各个任务之间的可并行性,定量地分析任务本身的计算量以及涉及到的数据量,从而实现负载均衡.

[参考文献](References)

[1] 姚震. 并行程序设计模型若干问题研究[D]. 合肥:中国科学技术大学计算机系,2006.
Yao Zhen. Study on parallel programming models [D]. Hefei: Department of Computer, University of Science and Technology of China, 2006. (in Chinese)

[2] 薛勇,万伟,艾建文. 高性能地学计算进展[J]. 世界科技研究与发展,2008,30(3):314-319.
Xue Yong, Wan Wei, Ai Jianwen. High performance geo-computation developments [J]. World SCI-TECH R and D, 2008, 30(3): 314-319. (in Chinese)

[3] 文剑. 并行计算平台的建立及性能分析[D]. 广州:广东工业大学计算机学院,2007.
Wen Jian. The set-up and performance analysis of parallel computing platform [D]. Guangzhou: Institute of Computer, Guangdong University of Technology, 2007. (in Chinese)

[4] 田蕾. 基于集群的并行计算的研究及其在离散元计算中的应用[D]. 北京:中国农业大学信息与电气工程学院,2006.
Tian Lei. Studies of parallel computing based on the cluster and applications in discrete element computation [D]. Beijing: Institute of Information and Electrical Engineering, China Agricultural University, 2006. (in Chinese)

[5] 李俊照,罗家融. 基于Linux集群的并行计算[J]. 计算机测量与控制,2004,12(11):1 064-1 090.
Li Junzhao, Luo Jiarong. Parallel computing based on Linux cluster [J]. Computer Measurement and Control, 2004, 12(11): 1 064-1 090. (in Chinese)

[6] 冯云,周淑秋. MPI + OpenMP混合并行编程模型应用研究[J]. 计算机系统应用,2006(2):86-89.
Feng Yun, Zhou Shuqiu. Research on development of mixed mode MPI + OpenMP applications [J]. Computer Systems and Applications, 2006(2): 86-89. (in Chinese)

[7] 牛志伟,黄红女. Windows平台下集群并行编译环境配置[J]. 计算机技术与发展,2007,17(8):15-18.
Niu Zhiwei, Huang Hongnü. Configuration of parallel compile environment of cluster on Windows platform [J]. Computer Technology and Development, 2007, 17(8): 15-18. (in Chinese)

[8] 李永旭. 基于MPI标准的并行计算平台的设计与实现[D]. 长春:东北师范大学计算机学院,2007.
Li Yongxu. The design and implement of the MPI-Based parallel computing platform [D]. Changchun: Institute of Computer, Northeast Normal University, 2007. (in Chinese)

[9] 周启鸣,刘学军. 数字地形分析[M]. 北京:科学出版社,2006.
Zhou Qiming, Liu Xuejun. Digital Terrain Analysis [M]. Beijing: Science Press, 2006. (in Chinese)

[10] Wang Shaowen, Armstrong M P. A quadtree approach to domain decomposition for spatial interpolation in grid computing enmronments [J]. Parallel Computing, 2003, 29(10): 1 481-1 504.

[11] Chen Minbin, Chuang Tyngruey, Wu Janjan. A parallel divide and conquer scheme for delaunay triangulation [C] // Ninth International Conference on Parallel and Distributed Systems. Taipei, 2002: 571-576.

[12] Lee Sangyoon, Park Chan-Ik, Park Chan-Mo. An improved parallel algorithm for delaunay triangulation on distributed memory parallel computers [C] // Advances in Parallel and Distributed Computing Comference(APDC'97). Shanghai, 1997: 131-138.

[13] Cervenansky M, Toth Z, Starinsky J, et al. Parallel GPU-based data-dependent triangulations [J]. Computers and Graphics, 2010, 34(2): 125-135.

[14] Carsten Maple, Jon Hitchcock. A novel scalable parallel algorithm for finding optimal paths over heterogeneous [C] // Ninth International Conference on Information Visualisation(IV'05). London: IEEE Press, 2005.

[15] Kidner D B, Railings P J, Ware J A. Parallel processing for terrain analysis in GIS: visibility as a case study [J]. Geoinformation, 1997, 1(2): 183-207.

[16] Willis C, Watson D, Tarboton D, et al. Parallel flow-direction and contributing area calculation for hydrology analysis in digital elevation models [C] // The 2009 International Conference on Parallel and Distributed Processing Techniques and Applications. Las Vegas, 2009.

[17] Mower James E. Data-parallel procedures for drainage basin analysis [J]. Computer and Geosciences, 1994, 20(9): 1365-1378.

[18] Clematis A, Coda A, Spagnuolo M. Developing non-local iterative parallel algorithms for GIS on a workstation network [J]. Recent Advances in Parallel Virtual Machine and Message Passing Interface, 1997, 1663: 435-442.

[19] Ortega L, Rueda A. Parallel drainage network computation on CUDA [J]. Compute and Geosciences, 2010, 36(2): 171-178.

[20] 富弘毅,丁滟,宋伟,等. 一种基于扩展数据流分析的 OpenMP 程序应用级检查点 [J]. 计算机学报, 2010, 33(10): 1809-1822.
Fu Hongyi, Ding Yan, Song Wei, et al. An application level checkpointing based on extended data flow analysis for OpenMP programs [J]. Chinese Journal of Computers, 2010, 32(10): 38-53. (in Chinese)

[21] 富弘毅,杨学军. 大规模并行计算机系统硬件故障容错技术综述 [J]. 计算机工程与科学, 2010, 32(10): 38-53.
Fu Hongyi, Yang Xuejun. A survey of the fault-tolerance techniques for large-scale parallel computing systems [J]. Computer Engineering and Science, 2010, 32(10): 38-53. (in Chinese)

[22] J von Neumann. Probabilistic Logic and the Synthesis of Reliable Organisms From Unreliable Components [M]. New Jersy: Princeton University Press, 1956.

[23] 孙鹏一,赵锁军,张文君. 软件容错: 技术与展望 [J]. 计算机工程与科学, 2007, 29(8): 88-93.
Sun Pengyi, Zhao Suojun, Zhang Wenjun. Software fault tolerance: techniques and prospects [J]. Computer Engineering and Science, 2007, 29(8): 88-93. (in Chinese)

[24] Yang Xuejun, Du Yunfei, Wang Panfeng, et al. Fault tolerant parallel algorithm: the parallel recomputing based failure recovery [C] // 16th International Conference on Parallel Architecture and Compilation Techniques. Brasov, 2007: 199-209.

[25] Yang Xuejun, Wang Panfeng, Fu Hongyi, et al. Compiler-assisted application-level checkpoint for MPI programs [C] // Pro of the 28th International Conference on Distributed Computing Systems. Beijing, 2008: 251-259.

[26] Shang Yizi, Wu Baosheng, Li Tiejian, et al. Fault-tolerant technique in the cluster computation of the digital watershed model [J]. TsingHua Science and Technology, 2007, 12(S1): 162-168.

[27] 周恩强,卢宇彤,沈志宇. 一个适合大规模集群并行计算的检查点系统 [J]. 计算机研究与发展, 2005, 42(6): 987-992.
Zhou Enqiang, Lu Yutong, Shen Zhiyu. Implementation of checkpoint system towards large scale parallel computing [J]. Journal of Computer Research and Development, 2005, 42(6): 987-992. (in Chinese)

[28] 杜云飞,王攀峰,富弘毅,等. 矩阵 LU 分解的容错并行算法设计与实现 [J]. 微电子学与计算机, 2008, 25(10): 1-4.
Du Yunfei, Wang Panfeng, Fu Hongyi, et al. Fault-tolerant matrix LU algorithm using parallel recovery [J]. Microelectronics and Computer, 2008, 25(10): 1-4. (in Chinese)

[29] 周笛. 软件容错方法、模型与实现 [J]. 计算机研究与发展, 1987, 24(2): 46-52.
Zhou Di. Software fault-tolerance method, model and implementation [J]. Computer Research and Development, 1987, 24(2): 46-52. (in Chinese)

[30] Elmendorf W R. Fault-tolerant programming [C] // Proc of FTCS-2. Newton, 1972: 79-83.

[31] Avizienis A, Chen L. On the implementation of N-Version programming for software fault tolerance during execution [C] // Proc of IEEE COMPSAC'77. Chicago, 1977: 149-155.

[32] Manic M, Frincke D. Towards the fault tolerant software: fuzzy extension of crisp equivalence voters [C] // The 27th Annual Conference of the IEEE Industrial Electronics Society. Denver, 2001: 84-89.

[33] 刘心松,朱鹰. 容错并行处理系统结构研究 [J]. 计算机应用, 1994(1): 8-11.

Liu Xinsong, Zhu Ying. Architecture research on fault tolerant parallel processing system [J]. Computer Applications, 1994(1): 8-11. (in Chinese)

[34] 曹小华,周勇. 基于主动请求与动态分配负载的CAN容错算法[J]. 华南理工大学学报:自然科学版, 2010, 38(9): 30-38.
Cao Xiaohua, Zhou Yong. The CAN load algorithms based on active request and dynamic distribution [J]. Journal of South China University of Technology: Natural Science Edition, 2010, 38(9): 30-38. (in Chinese)

[35] 杨际祥,谭国真,王荣生. 并行与分布式计算动态负载均衡策略综述[J]. 电子学报, 2010, 38(5): 1 121-1 130.
Yang Jixiang, Tan Guozhen, Wang Rongsheng. A survey of dynamic load balancing strategies for parallel and distributed computing [J]. Chinese Journal of Electronics, 2010, 38(5): 1 121-1 130. (in Chinese)

[36] Cybenko G. Dynamic load balancing for distributed memory multiprocessors [J]. J Par Distr Comp, 1989, 7(2): 279-301.

[37] 郭龙,陈闵中,叶青. 构造串行程序对应的并行任务(DAG)图[J]. 计算机工程与应用, 2007, 43(1): 41-44.
Guo Long, Chen Hongzhong, Ye Qing. Develop direct acyclic graph (DAG) corresponding to serial program [J]. Computer Engineering and Applications, 2007, 43(1): 41-44. (in Chinese)

[38] 张爱清,莫则尧. 有向图并行计算中一种新的结点调度算法[J]. 计算机学报, 2009, 32(11): 2 178-2 186.
Zhang Aiqing, Mo Zeyao. A new scheduling algorithm for digraph-based parallel computing [J]. Chinese Journal of Computers, 2009, 32(11): 2 178-2 186. (in Chinese)

[39] 杜建成,黄皓,陈道蓄,等. 基于最佳并行度的任务依赖图调度[J]. 软件学报, 1999, 10(10): 1 038-1 046.
Du Jiancheng, Huang Hao, Chen Daoxu, et al. Optimum degree of parallelism-based task dependence graph scheduling scheme [J]. Journal of Software, 1999, 10(10): 1 038-1 046. (in Chinese)

[40] 王霜,李心科. 基于LBT的网格依赖任务调度算法[J]. 合肥工业大学学报:自然科学版, 2010, 33(1): 64-67.
Wang Shuang, Li Xinke. Grid task scheduling algorithm based on LBT [J]. Journal of Hefei University of Technology: Natural Science Edition, 2010, 33(1): 64-67. (in Chinese)

[41] 杜云飞,唐玉华,杨学军. 容错并行算法的性能分析[J]. 计算机科学, 2009, 36(9): 248-251.
Du Yunfei, Tang Yuhua, Yang Xuejun. Performance evaluation for fault-tolerant parallel algorithm [J]. Computer Science, 2010, 33(9): 64-67. (in Chinese)

[42] 向建军,白欣,左继章. 一种用于实时集群的多任务负载均衡算法[J]. 计算机工程, 2003, 29(12): 36-38.
Xiang Jianjun, Bai Xin, Zuo Jizhang. A multipletask load balancing algorithm used in real-time cluster system [J]. Computer Engineering, 2003, 29(12): 36-38. (in Chinese)

责任编辑:严海琳