

高性能计算环境下地理计算服务流程并行处理方法

吴佳妮 刘 露 陈 犇 景 宁

(国防科学技术大学电子科学与工程学院 长沙 410073)

摘 要 地理计算是地理信息科学中重要的高级计算模式,是新一代地理信息系统(Geographic Information System, GIS)中不可缺少的组成部分。地理计算过程涉及数据处理、空间分析、过程模拟等多个流程,过程繁复且耗时,计算复杂且运算量大。高性能计算(High Performance Computing, HPC)是解决复杂地理计算问题的有效方法。针对高性能计算环境中地理计算服务流程并行执行问题,提出一种简洁的地理计算服务流程建模方法和基于多线程的并发流程处理策略,设计并实现了高性能计算环境下地理计算服务流程执行引擎原型系统,实现了并发条件下地理计算服务流程的自动执行与负载均衡。通过在真实高性能计算环境中的实验,验证了所提出的方法能够有效解决地理计算服务流程建模、执行优化问题,具有良好的性能、准确性及应用前景。

关键词 地理计算,高性能计算,地理计算服务流程,并行执行

中图分类号 TP311 文献标识码 A

Parallel Implementation Approach for the GeoComputation Service Process in the High Performance Computing Environment

WU Jia-ni LIU Lu CHEN Luo JING Ning

(Department of Electronic Science and Engineering, National University of Defense Technology, Changsha 410073, China)

Abstract GeoComputation is an important advanced computing paradigm in geographic information science. It is an indispensable part of the next generation of GIS (Geographic Information System). GeoComputation involves lots of processes such as data processing, spatial analysis, process simulation, etc. The process of GeoComputation is so complicated that usually consumes large amount of time. HPC (High Performance Computing) provides a powerful tool to solve the complex GeoComputation problems. To cope with the parallel implementation issues of the GeoComputation service process in HPC environment, a modeling method of the GeoComputation service process and a strategy for processing concurrent processes based on multithread were proposed. A prototype system of executing engine for the GeoComputation service process in HPC environment was designed and implemented, which can automatically resolve and dispatch the GeoComputation service process and keep the load balanced in concurrent conditions. The experimental results prove that the proposed method can effectively model GeoComputation service process with good practical performance, which can be widely used in geospatial data application fields.

Keywords GeoComputation, High performance computing, GeoComputation service process, Parallel implementation

1 引言

传统的 GIS 工具主要是为了在单工作节点上执行连续的地理算法而设计的,不能有效地处理海量数据以及复杂的地理计算,尤其在海量空间数据高效处理与分析、过程模拟与空间决策支持等方面存在着技术瓶颈。这些技术瓶颈将严重制约整个地理信息服务的发展,而随着并行计算集群、多核处理器等先进计算设施逐渐成为主流,以云计算系统为代表的新型应用体系架构可以为复杂地理计算提供高性能计算能力。因此,基于新型硬件架构的高性能复杂地理计算将成为

新型 GIS 发展的核心技术,并可为地理空间信息服务提供重要支撑。

在 HPC 环境下,GIS 服务的地位与作用不断提高,将逐渐超越传统的 GIS 工具,成为地理空间信息处理的主流模式^[1]。通常,为了满足用户针对特定复杂领域问题的处理需求,需要将多个不同功能的地理计算服务、资源按照一定的业务流程逻辑组合起来,以实现服务组合。该服务组合的流程既包含服务间的并行执行关系,又包含由于数据流向依赖而产生的顺序执行关系。因此,选用何种流程规范及执行引擎来处理这种复杂的地理计算服务流程,就显得至关重要。

到稿日期:2011-12-03 返修日期:2012-04-03 本文受国家高技术研究发展计划(863 计划)主题项目(2011AA120300),高等学校博士学科点专项科研基金(20104307110017),国家自然科学基金(60902036,61070035)资助。

吴佳妮(1988—),女,硕士生,主要研究方向为地理计算、并行计算、高性能计算,E-mail: tuotuoynini@163.com;刘 露(1982—),男,博士,讲师,CCF 会员,主要研究方向为地理计算、高性能计算、地理空间信息处理;陈 犇(1973—),男,博士,教授,硕士生导师,CCF 会员,主要研究方向为地理信息系统、地理空间信息处理、高性能计算。

从国内外 GIS 服务组合的研究现状看,将传统 Web 服务组合技术直接引入 GIS 服务组合是当前研究和应用的主要策略。George Mason 大学的 GeoBrain^[2] 项目将 GIS 服务组合领入到实用阶段,在服务的自动选择、自动组合方面进行了深入探讨。其在服务组合技术上就采用了传统 Web 服务领域的各种组合技术,包括采用 BPEL(Business Process Execution Language)^[3] 作为地理计算服务流程的描述语言,采用集中式的 BPEL 引擎实现服务流程的执行。此种策略没有考虑地理计算服务的特殊性,不利于持续推进并巩固基于地理计算服务的各种应用^[4]。

本文针对 HPC 环境中的地理计算服务组合与流程处理问题,基于有向无环图(Directed Acyclic Graph, DAG)对地理计算服务流程建模,形成对地理计算服务流程直观、简洁的表述。在实现层面,采用多线程技术处理并发请求,实现流程级别与活动级别的并行执行;基于底层作业管理系统实现负载均衡,设计并实现了 HPC 环境下的地理计算服务流程执行引擎原型系统,为地理计算服务流程的执行和管理提供了一个高效的并行计算平台。最后通过大量测试实验,验证了设计实现的流程执行引擎能有效地提高地理计算流程的执行效率。

2 背景知识

2.1 地理计算

1996 年,里兹大学召开了第一次国际地理计算大会,并正式将地理计算定名为合成新词“GeoComputation”。其定义为:利用不断发展的高性能计算机和计算方法,采用聚合集成等方法,开展各种求解地理问题的研究^[5]。它包括地理信息处理与管理、地理数据挖掘、地理过程建模模拟以及支持这些处理与分析的软件工程和计算体系研究。地理计算是地理信息科学的核心内容之一,主要研究地理信息科学的方法学问题,包括算法、建模和计算体系。

高性能计算,以并行高速计算为特征,以多处理器、多计算机或分布式系统为并行计算机的互联形式。地理计算常常涉及大规模计算问题,因此高性能计算为其提供了计算能力^[6]。以高性能为基础,地理计算中具有全球性或大区域性时空演变特征的地理现象模拟得以真正实现,原有的一些受计算机条件限制而难以模拟的模型得以完善、改进和运行。高性能计算为地理计算的发展提供了重要支撑。

2.2 DAG 定义

本文中地理计算服务流程基于 DAG 构建模型。

定义 1(DAG) $G=(N, E)$ 。其中 $N=\{n_i | i=1, 2, 3, \dots, n\}$, 表示图中 n 个节点的集合; $E=\{\langle n_i, n_j \rangle | n_i, n_j \in N, i < j\}$, 表示图中有向边的集合,其中 n_i 为始点, n_j 为终点^[7]。

定义 2(前驱/后继节点) 一条有向边中的两个节点中,始点为前驱节点,终点为后继节点。

定义 3(入/出口节点) 没有前驱节点的节点为入口节点,没有后继节点的节点为出口节点。

2.3 多线程技术

多线程技术被广泛应用于实现动态、异步、并发程序,尤其适合开发细粒度并行。在并行计算中常采用多线程技术主要是因为其具有开销小、线程间通信方便的特点。由于同一进程中的所有线程共享包括数据、环境变量等资源,因此启动

一个线程所花费的空间远小于一个进程的花费,并且一个线程的数据可以直接为其他线程所用,这样大大方便了线程间的通信。

多线程技术采用互斥锁、状态量、信号量等方法来解决线程的同步问题,防止并行的多个线程同时访问同一数据或资源时所造成的冲突。

2.4 作业管理系统

作业管理系统(Job Management System, JMS)由资源管理系统和作业调度系统两部分组成^[8]。资源管理系统能够屏蔽底层资源的异构性和复杂性,有效地管理资源,提高资源的利用率,并可按照管理员的意愿控制资源的使用方式。作业调度系统通常从资源管理系统得到各个节点上的资源状况和系统的作业信息,之后根据调度策略生成一个作业队列优先级列表。这个列表通知资源管理器何时在哪些节点上运行哪个作业。作业调度系统基于底层资源管理系统的支持,两者配合使用,组成集群作业管理系统。作业管理系统中各子系统的相互关系如图 1 所示。

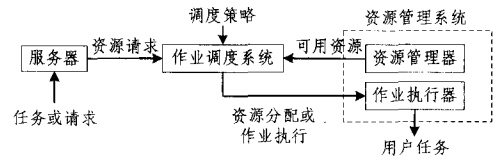


图 1 作业管理系统结构

作业管理系统提供了系统资源状态的搜集和更新、作业提交、调度、执行及控制等功能,能更加有效地利用系统资源、平衡负载,从而提高系统的整体性能。

3 基于 DAG 的地理计算服务流程建模

地理计算服务流程中通常包含空间数据集与地理处理算法两类要素。要素之间通过输入与输出、处理与被处理的关系形成前后依赖。此种结构可以基于 DAG 进行建模。在 DAG 中每个节点代表地理计算流程中的一个独立的地理计算服务;而任一条有向边描述了一种依赖关系,每条边的后继节点对前驱节点存在数据上的依赖,在执行时需要严格按照流程定义的先后顺序执行。而没有边连接的顶点间则不具有依赖关系,即可以并行执行其代表的服务。DAG 入口节点代表了流程的开始,出口节点代表了流程的结束,它们均不作为服务考虑。

在此基础上构建地理计算服务流程模型^[4],给出以下定义。

定义 4(地理计算成分服务, GeoComputation Component Service, GCCS) 地理计算成分服务是地理计算服务流程的参与者,代表流程中一个可以完成特定功能的地理计算子服务项,定义为 $GCCS=\{id, gccs\}$, 简称成分服务。其中 id 为服务标识, $gccs$ 为成分服务实体。

定义 5(数据流, Data Flow, DF) 数据流表达了成分服务之间数据的流量和流向。满足表达式 $DF \subseteq GCCS_s \times GCCS_n \times \Omega$, 其中 Ω 表示成分服务间的数据依赖映射。对于元素 $df=\{GCCS_m, GCCS_n, \omega\} \in DF$, 数据依赖映射 ω 定义为 $GCCS_m$ 到 $GCCS_n$ 的函数, 表征 $GCCS_n$ 的输入数据来源于 $GCCS_m$ 的输出数据, 记作 $GCCS_m \xrightarrow{\omega} GCCS_n$ 。此时 $GCCS_m$ 称为前驱服务(Predecessor Service, PS), $GCCS_n$ 称为后继服务

(Successor Service, SS)。

定义 6(流程活动, Process Activity, PA) 流程活动表示地理计算服务流程中的一个步骤, 定义为三元组 $PA = \{GCCS, PSc, Ss\}$, 简称为活动。其中, $GCCS$ 为地理计算服务流程的成分服务, PSc 为该成分服务的前驱服务个数, Ss 为其后继服务的集合。

定义 7(地理计算服务流程模型, GeoComputation Service Process Model, GSPM) 地理计算服务流程模型用于描述地理计算服务组合的流程逻辑, 由多个活动和数据依赖关系组成。定义为 $GSPM = \{GCCSs, PAs, DFs\}$, 其中:

- 1) $GCCSs = \{GCCS_i | i = 1, 2, 3, \dots, n\}$, 表示地理计算服务流程的 n 个成分服务的集合。
- 2) PAs 为流程活动的集合。
- 3) DFs 为数据流的集合, 用来表征活动之间的数据依赖关系。

定义 8(地理计算服务流程图, GeoComputation Service Process Graph, GSPG) 地理计算服务流程图是地理计算服务流程模型的数学模型, 用 DAG 表示, 记作 $GSPG = \{N, E\}$, 其中, N 是图中节点的集合, E 是图中有向边的集合。节点集合 N 包含两个特殊的节点 n_{start} 、 n_{stop} , 分别为入口节点和出口节点。流程图满足如下条件:

- 1) 对于 $\forall n \in N$, 存在从 n_{start} 到 n_{stop} 且经过 n 节点的有向边路径。
- 2) $E \subseteq N \times N \times \{w\}$ 。 $\forall e \in E$, 有 $e = \{u, v, w\}$, 其中 w 表示权重, $w \in \mathbf{R}^+$, $u, v \in N$, u 为前驱节点, v 为后继节点。
- 3) $\forall e = \{u, v, w\} \in E$, 在 GSPG 中存在从 u 到 v 的有向边路径。

定义 7 给出了地理计算服务流程模型的形式化定义, 本文采用定义 8 中的流程图对定义 7 进行数学描述。两者之间的关系对应为: 活动对应节点; 数据流对应有向边; 成分服务对应流程图中的节点具体内容。

给出一个地理计算服务组合例子进行说明。

图 2 给出的是一个较为典型的地理计算服务流程实例, 其流程简单描述为: 将 3 个覆盖服务 ($GCCS_1, GCCS_2, GCCS_3$) 提供的栅格数据进行边缘检测 ($GCCS_4, GCCS_5, GCCS_6$), 之后将输出图像进行拼接 ($GCCS_7$), 最后再进行构建金字塔服务 ($GCCS_8$), 以提高图像查询和预览速度。将其建模为如图 3 所示的流程图。

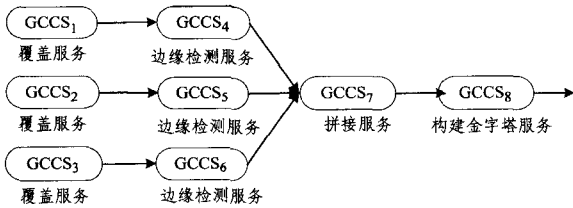


图 2 地理计算服务流程实例

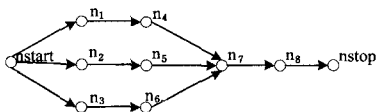


图 3 地理计算服务流程建模实例

图 3 中 GSPG 包含 10 个节点、11 条有向边, 其中 $n_1 - n_8$ 分别代表成分服务 $GCCS_1 - GCCS_8$, n_{start} 、 n_{stop} 代表了流程的

开始和结束。

与 GSPG 对应, 图 3 的流程模型为 $GSPM = \{GCCSs, PAs, DFs\}$ 。其中, $GCCSs = \{GCCS_i | i = 1, 2, \dots, 8\}$; 流程活集合 $DFs = \{GCCS_1 \xrightarrow{0} GCCS_4, GCCS_2 \xrightarrow{0} GCCS_5, GCCS_3 \xrightarrow{0} GCCS_6, GCCS_4 \xrightarrow{0} GCCS_7, GCCS_5 \xrightarrow{0} GCCS_7, GCCS_6 \xrightarrow{0} GCCS_7, GCCS_7 \xrightarrow{0} GCCS_8\}$; 流程活动集合 PAs 中: $PA_1 = \{GCCS_1, 0, GCCS_4\}$, $PA_2 = \{GCCS_2, 0, GCCS_5\}$, $PA_3 = \{GCCS_3, 0, GCCS_6\}$, $PA_4 = \{GCCS_4, 1, GCCS_7\}$, $PA_5 = \{GCCS_5, 1, GCCS_7\}$, $PA_6 = \{GCCS_6, 1, GCCS_7\}$, $PA_7 = \{GCCS_7, 3, GCCS_8\}$, $PA_8 = \{GCCS_8, 1, NULL\}$ 。

4 流程执行引擎设计

本文根据建立的流程模型, 设计实现相应的执行引擎, 其执行流程如图 4 所示。引擎采用多线程技术分别实现流程级与活动级这两种级别的并行执行策略, 同时结合作业管理系统实现了活动在线程级别以及作业级别的调度, 以此保证流程的正确执行以及集群资源的有效利用, 为地理服务流程的高效执行提供了一个并行计算平台。

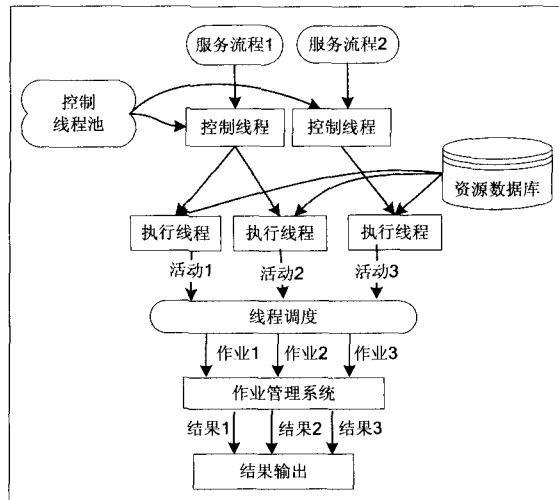


图 4 执行引擎执行流程

流程执行引擎的用户访问接口基于 Thrift 实现, 提供了流程或活动的开始执行、暂停、恢复、删除、状态查询等功能。Thrift 作为一种跨语言服务部署框架, 基于 TCP/IP 通信协议, 允许基于不同系统平台, 以不同语言实现的客户端与服务端进行消息通信, 增强了系统的弹性。

4.1 两级并行机制

引擎为地理计算服务流程提供了两类并行策略^[9]。第一类为流程级别的并行执行策略。引擎启动时创建一个控制线程池, 用于维护和管理引擎的控制线程。控制线程接收用户的服务流程请求, 为流程活动开启执行线程实现活动的具体执行; 之后执行用户的管理请求, 包括状态查询、暂停、恢复、删除以及返回执行结果等操作。如此, 一个用户对应一个控制线程, 保证了多用户同时提交的流程间的并行执行。由此, 用户数目的增加不会影响引擎的正常执行和执行效率。同时某个用户的服务执行出现异常也不会影响到其他服务的执行, 极大地提高了系统的吞吐量和性能。

第二类为活动级别的并行执行策略。流程中一个活动对应一个执行线程, 通过线程调度后各执行线程并发执行各活

动。通过两级并行机制,引擎实现了流程控制与流程执行的解耦,便于实现对流程执行过程的控制;同时保证了流程中无数据依赖的活动间可以并发执行,减少了引擎等待返回结果的时间,极大提高了流程的并发执行效率。

4.2 线程调度机制

引擎通过线程级的调度,保证了流程中具有依赖关系的活动间的正确执行顺序。引擎定义了一个变量数组,数组各元素记录对应活动当前未完成的前驱活动数量,其初始值为活动的前驱服务个数,即 PS_c 。如果数组中某元素不等于 0,意味着元素对应活动的前驱活动仍在运行,活动不能开始执行,执行线程需阻塞等待;当元素为 0 时,满足执行条件,活动开始执行。

活动执行完后,执行线程修改其所有后继活动(S_s)对应的数组元素,并通过广播函数通知被阻塞线程变量数组已被修改。此时被阻塞线程解除阻塞并查看是否满足执行条件。由此,依据地理计算服务流程模型,引擎实现了线程级的调度,确保了具有复杂依赖关系的流程能正确执行。

对于 HPC 环境中的地理计算服务流程,其复杂性还表现在每个服务实体还包含不定长的算法参数,以及关于并行执行环境的配置选项等信息,这些信息由执行线程在活动执行时从资源数据库中查找得到。执行线程获得算法及相应参数后,形成执行命令,以脚本形式提交到作业管理系统形成作业。作业管理系统根据引擎定制的调度策略,管理作业执行。

4.3 作业调度机制

引擎依赖 Torque+Maui 作业管理系统实现作业级别的调度。Torque^[10]是一个开源的作业管理系统,Maui^[11]是一种适用于集群和超级计算机的高级作业调度系统。本文将 Maui 与 Torque 相结合,用 Maui 替换 Torque 的调度功能模块,通过查询服务器和资源管理器获得更新的任务和节点信息。Maui 提供更好的调度策略支持,以指导 Torque 按照定制的调度算法、优先权和预定资源来管理任务,更大地优化了集群作业管理。

考虑到地理计算中常常是大作业与小作业并存,引擎将回填算法定制为调度策略。其示例如图 5 所示,在处理器空闲资源(图中网点部分)少于队列中等待的第一个作业(作业 3)所需资源时,在队列中搜寻下一个作业,直至找到所需资源量符合当前空闲资源并且所需执行时间小于队列第一个作业预留等待时间的作业,提前处理该作业(作业 5)。

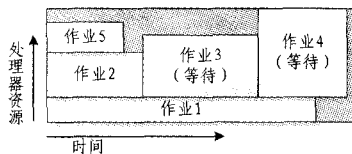


图 5 回填算法

由此,引擎实现了作业级别的调度。基于回填算法可以提高小作业的响应时间,同时不会导致大作业出现饥饿现象,有效提高了集群资源的利用率。

5 实验与结果分析

5.1 实验环境

实验硬件环境为 4 节点 32 核服务器集群系统;管理节点为:2 * Intel Xeon 4CE56202,4GHz,16G 内存,300G 热插拔

SAS 硬盘;计算节点为:2 * Intel Xeon 4CE56202,4GHz,16G 内存,300G 热插拔 SAS 硬盘;存储系统为:NFS,2TB,RAID0;千兆交换机。软件环境为 Linux 操作系统;Thrift 跨语言服务部署框架;Pthread 多线程库;Torque+Maui 作业管理系统。

实验使用数据如表 1 所列。

表 1 实验数据

数据	数据类型	数据大小(MB)	数据规模
1	栅格数据	135	7813 * 6000
2	栅格数据	353	21588 * 16970
3	栅格数据	1045	21654 * 20000
4	栅格数据	2107	30423 * 24000
5	矢量数据	869	209460 条线记录

5.2 正确性验证

本文对遥感影像边缘检测流程进行实验。该服务流程在地理计算中应用广泛,具有代表性,同时其综合了流程中的前后数据依赖关系和并行执行关系,能清楚地验证流程执行引擎的正确性。其流程图如图 6 所示。

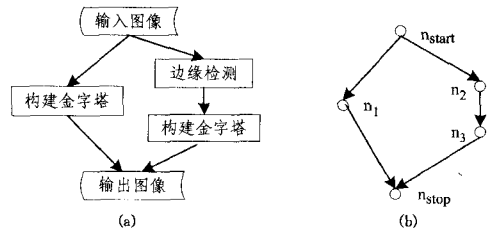


图 6 边缘检测流程

以数据 1 北京地区遥感影像作为原始影像,在 4 节点服务器集群系统上进行实验。经流程执行引擎处理,分别构建了北京地区遥感影像金字塔和该影像边缘检测后金字塔。

实验中,流程执行完后,通过 Torque 可以查看作业的详细执行情况,包括作业运行时间(walltime)、开始和结束时间、使用 cpu 时间(cput)、使用的节点和各节点核数以及内存资源等情况。该流程 3 个作业的运行信息如表 2 所列。

表 2 实验结果

作业	walltime(s)	节点(节点/核)	开始时间
1	5	4/0-4/7	11:03:49
2	3	4/0-4/7	11:03:54
3	18	1/0-1/7	11:03:49

表 2 显示作业 1 与作业 3 同时开始执行,分别占用了节点 4 与节点 1 的 8 个核。5s 后作业 1 执行完成,作业 2 开始执行,并占用作业 1 释放的节点 4 的 8 个核资源。由此可以清楚了解流程的执行过程。流程提交引擎后,活动 1 和活动 3 条件满足,分别作为作业 1 和作业 3 提交 Torque 执行;活动 2 所在线程阻塞等待。当作业 1 执行完后,活动 2 条件满足,执行线程解除阻塞并将活动 2 作为作业 2 提交 Torque,开始执行。

从输出图像以及作业执行状态可以看出,流程执行顺序和结果与期望的一致。同时,本文还对多用户同时访问以及具有更多服务、更复杂数据依赖关系的组合流程进行了实验,查看流程执行的情况,得到的结果均正确,由此验证了执行引擎的正确性。

5.3 性能测试

在 4 节点 32 核服务器集群系统上,将引擎执行地理计算

服务流程的时间与串行执行流程中各活动的的时间进行比对,测试在高性能计算环境下流程执行引擎的性能。本实验中通过命令行提交的方式串行执行各活动,是为了模拟传统 GIS 软件为用户提供地理计算服务的模式。

5.3.1 输入数据对执行效率的影响

仍以图像边缘检测流程进行实验,将不同大小的遥感影像数据集(数据 1 到数据 4)作为输入数据,测试运行时间,比对两种执行方式的效率。实验结果如图 7 所示。

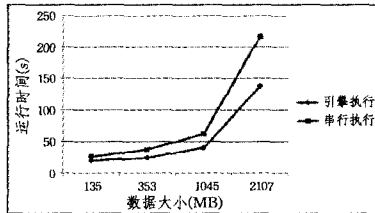


图 7 运行时间比对

从图 7 可以看出,与串行执行相比,引擎对流程执行有很好的优化,运行时间大大减少;并且随着数据规模扩大,执行引擎性能提升更显著。

5.3.2 活动数对执行效率的影响

本实验测试流程中活动数量对引擎执行性能的影响,通过增加不具有数据依赖关系的活动数量,查看引擎执行效率的变化。

测试矢量数据栅格化服务流程,流程如图 8 所示,其中栅格化服务的数量由输入数据集中矢量数据的数量决定。输入数据为数据集 5 中数据大小近似的一组矢量数据,矢量数据栅格化程序为 8 进程的 MPI 程序。实验结果如图 9 所示。

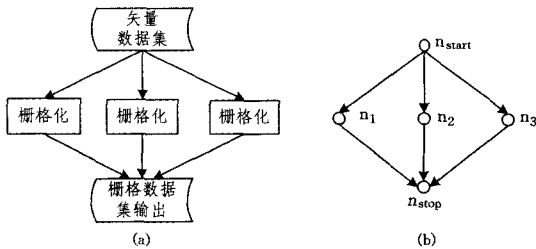


图 8 矢量数据栅格化服务流程

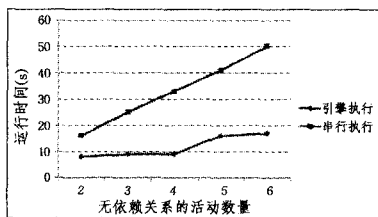


图 9 运行时间比对

从图 9 可以看出,随着无依赖关系的活动数量的增加,串行执行时间近似呈线性增长;而引擎执行时间增长缓慢。与串行执行相比,引擎的执行效率明显提高。

分析引擎运行时间曲线,在活动数量为 2,3,4 时运行时间基本无增长,当达到 5 时执行时间出现一个阶跃,之后随活动数增加又无明显增长,这是因为矢量数据栅格化程序是 8 进程的 MPI 程序,即一个程序运行时需要占用 8 个核的资

源,而实验环境为 4 节点 32 核服务器集群,当活动数量小于 4 时,集群计算资源满足所需,各活动并发执行,运行时间为各活动执行时间的最大值。当活动数量大于 4 时,计算资源不满足同时执行各活动,此时所有作业提交到 Torque 后,由定制的调度策略进行调度,部分作业将在队列中等待资源满足后开始执行。可以预测,在本实验中当活动数量达到 4 的倍数时,运行时间都将有一次明显增长。由此可见,引擎的执行性能是受集群性能约束的。

由此证明,流程执行引擎的并发处理机制有效利用了高性能计算环境中的计算资源,提高了流程的执行效率。

结束语 随着高性能计算的发展,高性能计算环境下的地理计算逐渐成为研究的重点。本文为地理计算服务流程建立模型,形成了对流程直观、简洁的表述方式,方便用户直观地对地理计算服务流程进行描述。在此基础上,设计实现了相应的流程执行引擎,为地理计算服务在高性能计算环境下的执行和管理提供了一个高效的并行计算平台。之后在高性能计算集群上进行测试实验,验证了所提方法的可行性。

下一步工作要完善引擎对故障和外部事件的处理,以及流程实例之间进行消息交换的协调机制,以实现对复杂流程更精确的控制功能。

参考文献

- [1] 贾文珏,李斌,龚健雅. 基于工作流技术的动态 GIS 服务链研究[J]. 武汉大学学报, 2005, 30(11): 982-985
- [2] Di Li-ping. GeoBrain-A Web Services based Geospatial Knowledge Building System[A]//Proceedings of NASA Earth Science Technology Conference, 2004[C]. Palo Alto, CA, U. S. A., June 2004: 22-24
- [3] Web Services Business Process Execution Language(WS-BPEL) [EB/OL]. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel, 2011-11-12
- [4] 左怀玉. 非集中式空间服务管理与组合技术研究[D]. 长沙: 国防科学技术大学, 2008
- [5] Rees P, Turton I. Guest editorial: geocomputation: solving geographical problems with new computing power [J]. Environment and Planning, 1998, 3: 1835-1838
- [6] Clarke K C. Geocomputation's future at the extremes: high performance computing and nanoclients [J]. Parallel Computing, 2003, 29: 1281-1295
- [7] 严蔚敏, 吴伟民. 数据结构(C语言版)[M]. 北京: 清华大学出版社, 1997: 179-190
- [8] 新华社. 面向空间数据处理的集群作业管理系统研究与设计[D]. 武汉: 武汉大学, 2005
- [9] Chen Qi-chang, Wang Li-qiang, Shang Zong-bo. MRGIS: A MapReduce-Enabled High Performance Workflow System for GIS [A]//Proceedings of the 2008 Fourth IEEE International Conference on eScience, 2008[C]. December 2008: 646-651
- [10] Torque [EB/OL]. <http://www.adaptivecomputing.com/products/open-source/torque/>, 2011-09-22
- [11] Maui [EB/OL]. <http://www.adaptivecomputing.com/products/open-source/maui/>, 2011-09-22