doi:10.13582/j.cnki.1672-9102.2014.03.017

基于 WCF 架构的 DEM 数据并行绘制

冯志元,李朝奎,张强

(湖南科技大学 地理空间信息技术国家地方联合工程实验室,地理空间信息湖南省工程实验室,湖南 湘潭 411201)

摘 要:在计算机集群上,采用 Windows 通信基础(WCF)架构,实现大范围 DEM 数据的并行绘制.根据控制节点的任务分配策略对集群中各服务器进行任务分配;服务端采用 WCF 的任务并发模式,对控制节点分配的任务直接分发给服务实例,无需放到消息队列中进行等待,以此提高集群对 DEM 数据的并行绘制效率.通过对不同数据量的 DEM 数据进行集群的并行绘制测试,得出绘制加速比随着数据量的增加而增加的结论,这对于大数据量的 DEM 数据的并行处理具有重要意义.

关键词:WCF;DEM;并行绘制

中图分类号:TP391.4 **文献标志码**:A **文章编号**:1672-9102(2014)03-0079-04

Parallel rendering of DEM data based on the WCF framework

FENG Zhiyuan, LI Chaokui, ZHANG Qiang

(National-local Joint Engineering Laboratory of Geospatial Information Technology,
Hunan Province Engineering Laboratory of Geospatial Information, Hunan University of Science and Technology, Xiangtan 411201, China)

Abstract: In a computer cluster, using the basic WCF communication architecture, a wide range of DEM data was parallel rendering. According to the task allocation strategy on master node, it was assigned to each server in the cluster; the server used task concurrency model of the WCF; the master node directly distributed tasks to an instance of service which need not put it into the message queue of waiting, to improve the efficiency of parallel rendering of DEM data. Through the different amount of DEM data parallel drawing test on cluster, the rendering acceleration ratio increased with the growing amount of data. For a large amount of data, DEM data parallel processing had important significance.

Key words: WCF; DEM; Parallel Rendering

大部分客户难以承受高性能工作站的昂贵价格,采用计算机集群(集群由控制节点和服务器节点组成)对地理大数据进行并行处理,会是一个较好的选择.集群对于大数据处理不仅价格大大降低,而且在数据处理效率方面要比高性能单机处理效率高得多,因此采用集群对地理大数据进行并行处理具有较高的性价比.

需要集群对大数据进行并行处理时,通过任务分配策略把任务分配给各绘制服务器;服务器端根据分配的任务,对本地磁盘上的备份数据进行一个一个地进行并行计算,从而达到并行处理的目的.采用 WCF 构架的并发模式,可以并行对任务进行分配处理,从而提高整个集群的任务处理效率.

集群采用 WCF 构架模式,每个服务器端采用 WCF 的并发任务处理;控制节点对分配的任务进行并发处理时,不再要求服务器端先把任务放到消息队列中,直接分配给服务实列,对任务进行处理,可以大大提高任务的绘制效率.

收稿日期:2013-05-15

基金项目:国家自然科学基金资助项目(41271390);湖南省自然科学基金资助项目(12JJ9023);湖南省研究生科研创新项目(CX2013B405);地理空间信息国家测绘局重点实验室开放基金资助项目(201330)

通信作者:李朝奎(1967-),男,湖南汉寿人,博士,教授,博士生导师,主要从事三维地理建模及其应用研究. E-mail: chkl_hn@163.com

1 并行绘制机制

1.1 并行绘制体系结构

目前,经典的 3 种并行绘制体系结构[1-3]中,Sort - first,Sort - last 是主流体系结构. 也有使用两者的混合体系结构[2-3]. Sort - first 体系结构中,几何处理之前要做归属判断,显示屏中的图像空间区域被分割成多个子区域并分配给不同的绘制服务器进行处理. Sort - last 则不对图像空间区域进行划分,而是只将显示屏幕中场景数据等分为若干子数据集,然后对子数据集进行划分(此划分不受空间格局限制)[4];因而可以保证各处理器的绘制单元数量基本相等,从而保证负载平衡.

Sort - first 的优点是各流水线之间完整且相互独立;缺点是需要进行图元归属判断. 对显示屏幕划分并非空间均匀划分,容易造成各处理器负载不平衡. Sort - last 的优点是把场景中空间对象划分为若干个子数据集,对划分后各数据集的显示屏幕不再进行划分,而且没有任何冗余处理;缺点是需要深度合成,给网络数据传输带来巨大的压力.

1.2 任务划分

常用的自适应任务分配算法很多主要有基于几何的数据分析自适应负载平衡算法(Roble 算法[5]、Whelan 的 median - cut 算法[6]、Whitman 的自顶向下分解算法[7]和 MADH 算法[8])和基于时空转换的负载平衡算法[9]. MADH 算法,综合了 Whitman 和 Whelan 算法的优点,采用网格覆盖屏幕,对于包含同一空间对象(占有 N 个网格)的网格,每个网格的权重为 $1/N$. 可以根据不同的权重值对整个屏幕进行任意屏幕的划分,使得划分的任务数量等于处理的进程数的整数倍;另外容易使各个绘制服务器的负载达到平衡,该方法比较灵活.

采用高效的任务分配方法:在任务分配的时候,总是把任务分配表中的任务分配给服务器中累积时间最少的进程,从而达到每个处理器都能正常工作,并实现负载平衡. 如图 1:进程 1、进程 2 和进程 3 的任务累计时间分别为 t_1,t_2 和 t_3,若同时 $t_2 < t_1 < t_3$,此时任务表中的当前"任务 1",需要分配给"进程 2"来执行,依次循环,直到任务表中的任务分配完毕.

对划分好的任务之间,小粒度的任务与大粒度的任务进行自动合并,使得各个任务之间的绘制时间尽量保证一致,通过任务之间的自动调整,使得各个处理器得到的负载基本平衡.

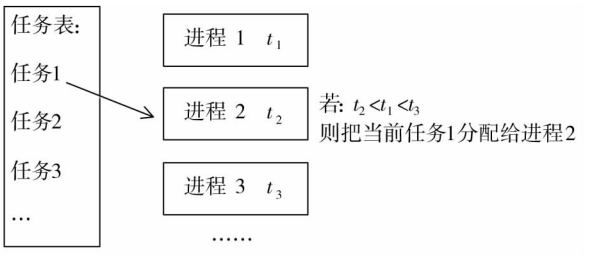

图 1 高效的任务分配

1.3 WCF 并发模式

在 sort - first 体系结构下,各个绘制流水线之间相互独立,便于并行处理. 集群中各个服务器采用 WCF 的并发模式优势:第一、服务端不管理服务,而且不管在任何情况都不会同步访问服务实例;第二、服务实例支持并发调用,而不会把客户端的消息放到消息队列中,而是一旦到达立即分发给服务实例,对任务进行分配,从而提高并行绘制效率.

WCF 的并发模式主要分为以下几个方面:

1)集群并行构架是基于 WCF 的通信基础架构[10-12],不但能提供 Web Service,还可以提供 Remoting, Enterprise Services, MSMQ (Microsoft message queue)等,目标是为这些技术提供统一的编程模型,将编程与通信基础结构分开. 基础构架可以分 3 个层次结构:集群客户端代理、集群计算节点和分布式存储.

第一、集群客户端代理[13],是集群服务器端为客户端发送请求提供的一个访问接口. 通过该接口,使得客户端与服务端建立可信连接.

添加引用服务,配置终结点信息,包含服务地址、HTTP 协议绑定、注册服务的绑定和服务契约;终结点的配置文件[14]如下:

```
< endpoint address = " http://localhost: * * * * * * /Service. svc"  binding = " wsHttpBinding"
bindingConfiguration = " WSHttpBinding _ IService1 "
contract = " ParallelDisplayService. IService1"
    name = " WSHttpBinding_IService1" >
```

接下来,在客户端与服务端就可以创建会话:

```
SessionStartInfo info = new SessionStartInfo( "
* * * * * * * ", " * * * * * * * * ");
DurableSession session = DurableSession.
CreateSession( info);
```

建立会话以后,还要对会话访问机制创建安全的传输;在此采用 TCP 协议建立安全绑定传输:

```
NetTcpBinding binding = new NetTcpBinding
```

(SecurityMode. Transport);

这时,就可以在客户端创建一个客户端代理:

BrokerClient < IService1 > client = new BrokerClient < IService1 > (session, binding);

通过客户端代理,客户端就可以调用服务端的服务实例. 在文中实验里,通过与服务端建立会话,并向服务端传送相应服务端的绘制指令,以启动服务,完成服务绘制工作.

第二、集群计算节点又称"集群绘制节点",是集群中成分布式的承载数据处理服务的一些节点. 每一个计算节点,都留有整个场景数据的备份,无需从客户端调用数据,通过接收客户端的绘制指令,直接从本地读取数据,并进行计算绘制.

第三、分布式存储,是对大范围的场景数据储存的形式,分布在各个计算节点上. 计算节点接收到绘制指令时,可以直接对本服务器上的相应数据进行绘制和处理,减少数据从客户端调用在网络传输上增加开销.

2)WCF 的控制器[15]:是由消息路由模块、负载平衡模块和服务器探测模块等组成.

消息路由模块:监听客户端的请求;把请求传递给节点,启动服务;然后消息回应传递给监听模块.

集群上的负载平衡模块:通过消息路由监听到服务端的负载情况,在客户端可以根据每个服务器的负载情况,有目的给负载小的服务器分配任务.

服务探测模块:主要是对服务器端的状态进行探测,即使更新各服务器的状态表,使得下一次分配任务时,可以参考各服务器的状态进行负载的分配.

3)服务并发模式:服务端的 ServiceBehavior 特性的 ConcurrencyMode 属性负责管理服务实例的并发访问:

Public enum ConcurrencyMode { Single, Reentrant, Multiple}

当服务被设置为 ConcurrencyMode. Single, WCF 会禁止并发调用. 其他的服务实例被放在消息队列[16]中等候获得同步锁才能被调用(如图 2).

当服务被设置为 ConcurrencyMode. Multiple, 服务端不管理服务,而且不管在任何情况都不会同步访问服务实例;这时服务实例支持并发调用,而不会把客户端的消息放到消息队列中,而是一旦到达立即分发给服务实例(默认最大并发调用值为 16,并不是匹配所有客户端的调用数量;可以根据需要设置并发调用值).

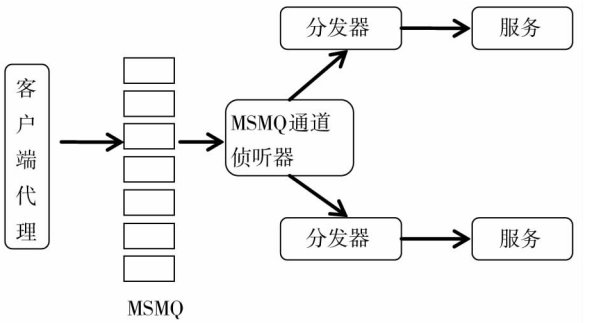

图 2　WCF 的消息队列调用

2　并行绘制实验

1)实验硬件环境

3 台计算节点和一台代理节点,配置:处理器 Intel(R) Xeon(R) CPU E5620 @2.40GHz (2 处理器),内存 8.00 GB,64 位操作系统

2)实验软件环境

头节点配置环境:VS2010 开发环境、HPC Server 2008 操作系统、HPC Pack 2008 R2 SDK, MPI. Net 和 MPI. NET Runtime.

计算节点配置环境:HPC Server Pack 2008 R2,HPC Server 2008 操作系统、MPI. NET Runtime 和 net Framework 4.0.

3)从不同级别的数据量(0.344, 2.030, 3.620, 5.020, 10.000 G)的 DEM 数据,对并行绘制的性能进行测试,它们的加速比如表 1 所示:

表 1　不同数据量加速比测试结果

序号	数据量 /G	并行绘制时间 /h:m:s.ms	串行绘制时间 /h:m:s.ms	加速比 /%
1	0.344	0:00:49.935	0:02:06.407	2.53
2	2.030	0:01:06.500	0:12:46.928	11.53
3	3.620	0:01:07.972	0:21:17.892	18.80
4	5.020	0:01:28.585	0:29:39.667	20.09
5	10.000	0:02:37.639	1:03:23.411	24.13

其中,3.620 G 的 DEM 并行绘制结果图如图 3 所示. 不同数据量的加速比如图 4 所示.

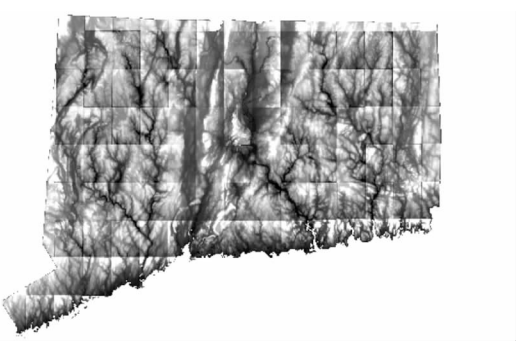

图 3　绘制结果图

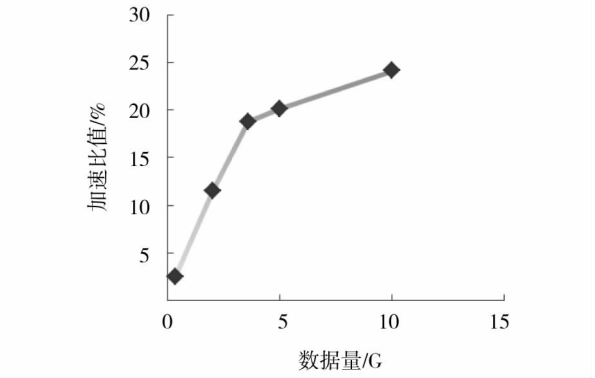

图 4 不同数据量加速比

3 结论

文中采用的是 sort - first 流水线体系结构,对任务进行自适应任务划分,并对划分的任务采用高效的任务分配方法进行任务分配;在服务器端采用 WCF 的并发模式,把任务分发给服务实例. 实验表明:采用基于 WCF 并发架构的集群,对不同级别的 DEM 数据量进行并行绘制测试(加速比如图 4 所示),并行绘制的加速比随着数据量的增加而增加. 采用集群对大数据量的 DEM 数据进行并行绘制,对提高数据的处理效率具有重要意义.

参考文献:

[1] 谭同德. 基于多核 PC 集群的并行绘制系统研究与实现[D]. 河南郑州:郑州大学,2010.

[2] Samanta R, Funkhouser T, Li K, et al. Sort - First parallel rendering with a cluster of PCs[M]. New Jersey, USA: In Eurographics/SIGGRAPH workshop on Graphics hardware, 2000.

[3] Yang J, Shi J Y, Jin Z F, et al. Design and implementation of a large - scale hybrid distributed graphics system [C]//Proceedings of Fourth Eurographics Workshop on Parallel Graphics and Visualization. Switzerland: Eurographics Association Aire - la - Ville, 2002.

[4] 王观武. 基于 GPU 集群系统的并行绘制技术研究[D]. 长沙:国防科技大学,2010.

[5] Roble D R. A load balanced parallel scanline Z buffer algorithm for the iPSC hypercube [C]//The 1st International Conference PIXIM 88. Paris, France: Editions Hermes, 1988.

[6] Daniel W. A multiprocessor architecture for real - time computer animation[D]. Pasadena, California: California Institute of Technology, 1985.

[7] Whitman S. Dynamic load balancing for parallel polygon rendering[J]. IEEE Computer Graphics and Applications, 1994, 14(4): 41 - 48.

[8] Mueller C. The Sort - First rendering architecture for high - performance graphics [C]//Proceedings of the 1995 Symposium on Interactive 3D Graphics. Chapel Hill, USA: University of North Carolina, 1995.

[9] 金哲凡. 保留模式图形并行绘制研究[D]. 杭州:浙江大学,2004.

[10] 罗丹. 基于 WCF 的遗留系统并行构架的设计与实现[D]. 杭州:浙江大学,2011.

[11] Lowy J. Programming WCF services[M]. California, USA: O'Reilly, 2007.

[12] Klein J. Professional WCF programming[M]. Beijing: John Wiley & Sons, 2009.

[13] Yang R, Yang K S, Kafatos M, et al. Value range queries on earth science data via histogram clustering[C]//In First International Workshop on Temporal, Spatial, and Spatio - Temporal Data Mining. Lyon, France: Lecture notes in computer science, 2001.

[14] 吴清寿. 基于 WCF 的分布式系统模型研究与实现[J]. 吉林师范大学学报,2012,33(3):61 - 64.

[15] 侯彦娥. 基于 WCF 的异构系统间数据交互研究[J]. 河南大学学报(自然科学版),2013,43(4):104 - 108.

[16] 冯志红,赵拥军,李光茂. 消息调度模型下一种改进的双缓存地图漫游算法[J]. 测绘科学,2013,38(4):86 - 90.