Computers & Geosciences 82 (2015) 78–88

Contents lists available at ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

A virtual tile approach to raster-based calculations of large digital elevation models in a shared-memory system

Ahmet Artu Yıldırım [a,*], Dan Watson [a], David Tarboton [b], Robert M. Wallace [c]

[a] Department of Computer Science, Utah State University, Logan, UT, USA
[b] Utah Water Research Laboratory, Utah State University, Logan, UT, USA
[c] US Army Engineer Research and Development Center, Information Technology Lab, Vicksburg, MS, USA

ARTICLE INFO

Article history:
Received 12 May 2014
Received in revised form
7 April 2015
Accepted 27 May 2015
Available online 30 May 2015

Keywords:
Multithreaded parallel digital elevation model analysis
Pit filling algorithm
User-level virtual memory management

ABSTRACT

Grid digital elevation models (DEMs) are commonly used in hydrology to derive information related to topographically driven flow. Advances in technology for creating DEMs have increased their resolution and data size with the result that algorithms for processing them are frequently memory limited. This paper presents a new approach to the management of memory in the parallel solution of hydrologic terrain processing using a user-level virtual memory system for shared-memory multithreaded systems. The method includes tailored virtual memory management of raster-based calculations for datasets that are larger than available memory and a novel order-of-calculations approach to parallel hydrologic terrain analysis applications. The method is illustrated for the pit filling algorithm used first in most hydrologic terrain analysis workflows.

1. Introduction

Grid digital elevation models (DEMs) are commonly used in hydrology to derive information related to topographically driven flow (Moore et al., 1991; Hengl and Reuter, 2009; Wilson and Gallant, 2000). When dealing with large digital elevation model (DEM), datasets, computational efficiency and memory capacity become important considerations. Prior work in Tarboton (2014) has advanced parallel methods for terrain processing using a message passing interface (MPI) approach that allows memory to be distributed across multiple processors in medium-sized cluster computers (Tesfa et al., 2011; Wallis et al., 2009; Tarboton et al., 2009).

In desktop computers, virtual memory systems are the standard approach to working with data that is too big to fit in memory. Operating systems typically implement virtual memory using page files that hold on disk contents of memory. However repeated swapping (thrashing) occurs when these get large because the system has limited general capability to anticipate the pages needed next. Most operating systems implement the virtual machine in kernel that makes it difficult, sometimes impossible, to change its functionality and page replacement policy. This necessities the implementation of a user-level tailored virtual memory system to handle a programs' locality better for fine-grain control (Engler et al., 1995).

There is a need to process large DEMs on desktop computers that are often limited in total memory. This is the primary problem addressed in this work. It is also desirable to have parallel terrain analysis algorithms that use multi-threading to take advantage of common multi-core processors (Marowka, 2011) for greater efficiency. This is a secondary consideration in this work.

This paper presents a new approach to the management of memory and the parallel solution of the raster-based computations for shared-memory multithreaded systems, such as desktop computers. The contributions of this method in the context of parallelism are a tailored user-level tile based virtual memory manager for raster-based calculations for datasets that are larger than available memory and a novel order-of-calculations approach to parallel hydrology analysis applications.

We implemented a modified version of the Planchon and Darboux pit filling algorithm (Planchon and Darboux, 2002) as implemented in Tarboton (2014) and Wallis et al. (2009) as an application of our tiled virtual memory manager and evaluated its effectiveness for pit removal in DEMs of varying size, with a varying number of operating system threads and memory capacity. The results demonstrate several benefits over the use of standard virtual memory approaches.

Furthermore, this study examines a load-balancing technique to minimize the idle times which might occur in case of uneven load between compute threads due to data variability. We used the GDAL library (GDAL Development Team, 2014) to enable a wide

* Corresponding author.
E-mail addresses: ahmetartu@aggiemail.usu.edu (A.A. Yıldırım), dan.watson@usu.edu (D. Watson), dtarb@usu.edu (D. Tarboton), robert.m.wallace@usace.army.mil (R.M. Wallace).

http://dx.doi.org/10.1016/j.cageo.2015.05.014
0098-3004/

range of raster file formats within the implemented memory manager. We implemented our memory manager using the Microsoft Windows 7 operating system as it is widely used for desktop Geographic Information System terrain analysis and enabling the processing of large DEMs on Windows systems was a goal of this work. The source code of the virtual memory project and the pit-filling algorithm can be found at https://bitbucket.org/ahmetartu/hydrovtmm.

The paper is organized as follows: Section 2 provides background and literature review. Section 3 gives specifics of the modified Planchon and Darboux Algorithm used here. Section 4 describes the design of the multi-threaded tiled virtual memory manager for raster-based calculations that is contributed here. Section 5 gives performance results. Finally, we discuss conclusions based on the obtained results in Section 6.

2. Background

This review addresses three subjects that are needed to set the context for this work. First we review existing DEM pit filling approaches focusing on the Planchon and Darboux (2002) Algorithm modified and used in this work. We then examine other efforts that use parallel methods for hydrologic terrain analysis and general methods for virtual memory management to enable the processing of large datasets that do not fit in physically available computer memory.

Pits, defined as grid cells or sets of grid cells completely surrounded by higher grid cells, often occur during DEM production and are generally considered to be artifacts of the data collection process (Jenson and Dominque, 1988). Drainage conditioning to remove pits is an important preprocessing step in the hydrologic analysis of DEMs, and is representative of a broad class of raster-based algorithms (e.g. Tarboton et al., 1991; Moore et al., 1991) designed to determine topographically driven flow. Once drainage conditioning has been performed, a DEM that has no pits is referred to as being hydrologically conditioned. The most common approach to drainage conditioning is pit filling, whereby pit grid cells are raised to a level where they are at a minimum equal to the lowest surrounding grid cell and can drain. A well-known effective pit filling algorithm is described by Planchon and Darboux (2002). Pit removal using the Planchon and Darboux (PD) algorithm is one of the multiple hydrologic terrain analysis functions in the TauDEM package. Time-complexity of the direct implementation of the PD algorithm is reported to grow with the number of cells N in DEM as $O(N^{1.5})$ (Planchon and Darboux, 2002). Planchon et al. also provided an improved implementation that embedded a recursive dry upward cell function that was reported to achieve a time-complexity of $O(N^{1.2})$ (Planchon and Darboux, 2002). Alternative pit filling algorithms, some claiming better efficiency, have been presented by others (Wang and Liu, 2006; Barnes et al., 2014; YongHe and WanChang, 2009).

The Planchon and Darboux (PD) approach fills pits by covering the whole surface with a layer of "water" up to a level greater than or equal to the highest point in the DEM, then removes the excess water in an iterative manner. Doing so, the algorithm naturally leaves the water in the depressions at the height of their outlet. Let $Z \in \mathbb{R}^2$ be the set of input elevation points (i.e., the input DEM with size m, where each member is an elevation point $x_i, 1 \le i \le m$) and let $W \in \mathbb{R}^2$ be the output DEM consisting of "filled" elevation points y_i. The goal of the PD algorithm is to increase each elevation point x_i with a minimal difference of elevation. The PD algorithm initializes all grid cells to a large value (greater than the highest point in the domain). It then uses iterative scans across the domain to lower elevation values to the lowest elevation greater than or equal to the original terrain hydrologically conditioned so that they drain to one of their neighbors.

Tarboton (2014) and Wallis et al. (2009) implemented a parallel version of the PD algorithm. This adopted a distributed memory domain partitioning approach to parallelism and divided the domain into horizontal stripes, one stripe per parallel process. The PD algorithm was applied separately to each stripe in parallel, with a step to exchange information across stripe boundaries at the end of each iteration so as to ensure convergence to the same global solution as obtained by a serial implementation. The original PD algorithm and the Wallis et al. parallel implementation visit each grid cell on each iteration. Scans of the grid cycle through all eight possible combinations of row and column scan orders. PD also offered an improved implementation that used a recursive dry upward tree search each time a cell was set to the original elevation to enhance efficiency. However each iterative pass across the DEM still examined each grid cell.

Subsequent to the Wallis et al. (2009) work, the TauDEM team (Tarboton, 2014) identified the visiting of each grid cell on each iteration as an inefficiency and developed a stack based approach whereby unresolved grid cells are placed on a stack on the first scan, then removed from the first stack on each subsequent scan and placed on a second stack. Stacks are then switched. This limits the scanning to two directions rather than eight, but was found to result in a speedup of a factor of 2 for small datasets and 4.3 for a modestly large 1.5 GB dataset in comparison to the eight combination full grid scanning. The benefits of the stack thus seem to outweigh the inefficiency of fewer scan directions. The TauDEM team did not evaluate the recursive dry upward approach of the improved PD algorithm. Recursive methods use the system stack to expand system memory, posing a challenge for memory management in large data computations. They also pose a challenge for a domain partitioned parallel approach as cross partition calls are less predictable. They are also hard to implement on a stack as they would require additional code to track the stack position of each grid cell and to handle changing the order in which grid cells on the stack are processed. The two direction stack based modified PD algorithm was incorporated into the publicly released version of Tarboton (2014) that was the starting point for this work. The focus of this paper is on virtual memory management for large DEM data, and the modified PD algorithm as used by Tarboton (2014) and Wallis et al. (2009) is used as an example to illustrate a general approach.

Beyond pit filling several parallel computing technologies have been employed in the implementation of the hydrologic algorithms to achieve higher performance by effectively utilizing available processors in the system including MPI for distributed memory architectures (Do et al., 2011; Tesfa et al., 2011; Wallis et al., 2009), OpenMP (Xu et al., 2010) or low-level standard threading library for multi-threaded shared memory architectures, and NVIDIA CUDA for general-purpose computing on graphics processing units (GPUs) (Wang et al., 2012; Ortega and Rueda, 2010). Xu et al. (2010) present a grid-associated algorithm to improve the performance of a D8 algorithm (O'Callaghan and Mark, 1984) via OpenMP technology which is a thread-level standard of parallel programming. CUDA algorithms for drainage network determination are presented by Ortega and Rueda (2010), achieving up to 8× speed-up improvement with respect to corresponding CPU implementation. Do et al. introduced a parallel algorithm to compute the global flow accumulation in a DEM using MPI (Do et al., 2011) where hierarchical catchment basins are computed by means of a parallel spanning tree algorithm to model the flow of water. Each processor computes a local flow direction graph using the *D8* flow routing model.

There have been a number of general approaches to better manage virtual memory. The Tempest (Reinhardt et al., 1994)