Parallel calculation of LS factor for regional scale soil erosion assessment

Kai Liu [1], Guoan Tang [2]

[1]Key Laboratory of Virtual Geographic Environment (Nanjing Normal University), Ministry of Education, Nanjing, 210023, China
Telephone: +86 13770866548
Email: liukaigis@gmail.com

[2]Key Laboratory of Virtual Geographic Environment (Nanjing Normal University), Ministry of Education, Nanjing, 210023, China
Telephone: +86 13776623891
Email: tangguoan@njnu.edu.cn

1. Introduction

In many studies, USLE/RUSLE has been employed for the quantitative assessment of soil erosion at large watersheds or even on a regional scale. At the same time, the development of an acquisition method for spatial data means that the geospatial dataset size is increasing at an exponential rate and high resolution DEM data for large regions are becoming more available. As a result, the computational time increases significantly and to the level becoming a bottleneck for applying these models over large areas (Jiang et al., 2013). It is limited by computer storage and calculation ability that traditional serial processing cannot meet user demand due to the long response time. Among the factors in the Universal Soil Loss Equation (USLE) and the revised USLE (RUSLE), the extraction of the LS factor, which reflects the influence of terrain on soil erosion, is a key issue in the applications of these models. Currently, two methods are adopted for LS factor calculation on regional scale. One method employs a set of prediction rules to generate LS factor based on a number of attributers which control the landform types such as materials, climate and regional geomorphology. Another extracts the topographic index based on low resolution DEMs, and then gets available values via a scale transformation. However, both methods could only be regarded as an expedient measure, the fast and accurate calculation of LS factor on regional scale is yet to be developed.

Developments in computer technology have improved computation ability by using parallel processing. Recently, parallel technology has been widely used in digital terrain analysis, such as parallel drainage extraction, parallel visibility analysis, and parallel hydrological analysis. In order to make the parallel programming become easy, some raster-based programming libraries are proposed such as Parallel Raster Processing Programming Library (pRPL) (Guan and Clarke 2010) and Parallel Raster-based Geocomputation Operators (PaRGO) (Qin et al, 2014). Among the existing literatures, Message Passing Interface (MPI) is widely used to parallelize algorithms for its adaptable to various parallel computing environments and the rich programming interfaces. Compared with the serial algorithms, the MPI parallel algorithms can achieve a huge improvement in processing time (Tesfa et al., 2011). There is, however, little published research on the LS factor calculation using parallel computing.

The aim of this paper is to propose a parallel approach that can be applied on a regional scale calculation of the LS factor. The workflow for LS factor calculation contains algorithms for flow direction, flow accumulation, drainage network, slope, slope length and the LS factor. Experiments on a multi-node system show that the proposed parallel algorithm allows efficient calculation of the LS factor at a regional scale with a massive dataset.

2. Methodology

2.1 Approach to parallelization

In the LS factor calculation process, all the algorithms can be divided into two types, local algorithms and global algorithms, based on the existence of data dependency during the computation process. In local algorithm, the calculation for each cell is independent, which means the calculation results rely only on the input data and can be completed after traversing the input data once. The local algorithm can be further divided into two types: 1) point computation, such as the algorithm for drainage-network definition and factor calculation, which has no relevance with other grids; 2) regional computation, such as the algorithm for slope and flow direction, in which the computation process needs a limited region. Regional computation depends on the center cells and its neighbor cells and different algorithm employs different scope transition, the Moore neighborhood are used for both slope algorithm and flow direction algorithm in LS factor calculation. Compared with the local algorithms, the processing of global algorithm is not independent, which means the calculation of most cells may rely on the calculation results of other cells.

The existing researches prove that local algorithms have better parallelism and communication strategy should be designed for the parallelization of global algorithms to accommodate the influence of data dependency. The parallel strategies in this paper are designed according to the algorithmic characters discussed above, including the decomposition method to maintain the integrity of the results, optimized workflow to reduce the time taken for exporting the unnecessary intermediate results and the communication strategy to improve the communication efficiency.

2.2 Programming implementation

The MPI parallel algorithm in this paper was developed in C++ programming language. Fig. 1 shows an integral flowchart of the parallelization workflow in the extraction of the LS factor and it is described in the following steps.

Step 1: Each process is initialized with the number of processes and parameters involving the algorithm.

Step 2: The input dataset is subdivided into subdomains and the result file is created by process 1.

Step 3: Each process reads data from the input file according to its anchor point.

Step 4: Subtasks are implemented in each process.

Step 5: If the algorithm is the global type, the buffer-communication-computation strategy should be used to make sure that all the cells finish their calculations.

Step 6: Each process writes the computation result to the result file.

Step 7: Each process frees memory and the task is completed.

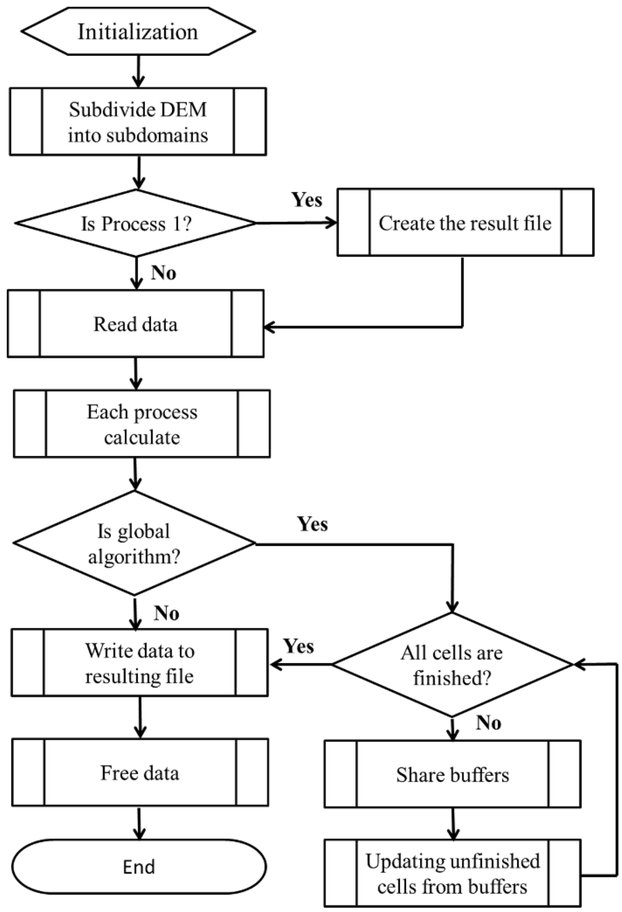

Fig. 1. Parallelization workflow

3. Experiments and results

In order to evaluate its accuracy and effectiveness, the parallel algorithm were conducted on an 80-core cluster composed of 10 diskless nodes that were connected through an Ethernet network with Gigabit (1000 Mbit/s) speed. Each node had dual quad-core Intel(R) Xeon(R) E5620 processors (2.40GHz, 16GB RAM) under a Linux operating system. Fig. 2 shows the run time taken to complete each steps of the LS factor calculation using both serial algorithm and the parallel MPI algorithm. Between 1 and 16 processes, the execution time of both the 90-m dataset and 30-m dataset are significantly reduced, which are shown in Fig. 2a and Fig. 2b, respectively. There are three kinds of speedups shown in Fig. 3, the ideal speedup, total time speedup and the compute time (total time minus the time for disk I/O) speedup. Because of the differences in serial algorithms and the parallel strategy, there are clearly variations in total time speedups of different algorithms. As shown in Fig. 4, the speedups of local algorithms are obvious higher than that of global algorithms by using the smaller dataset. However, the differences between local and global algorithms get narrower by using larger dataset

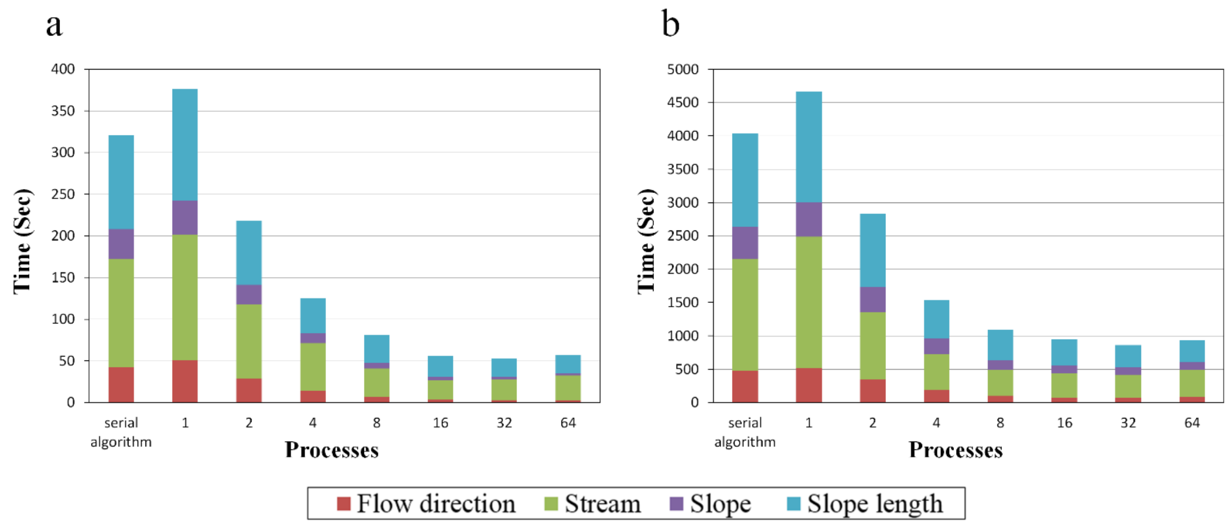

Fig. 2. Time taken to parallel algorithms for LS calculation with different DEM datasets: (a) 90-m dataset (b) 30-m dataset

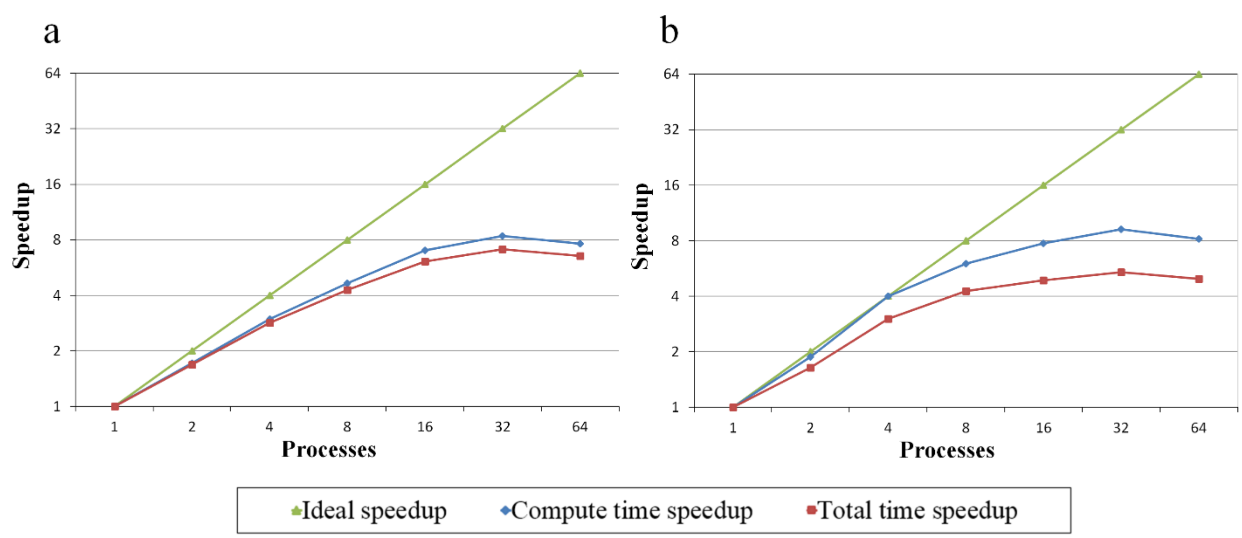

Fig. 3. A comparison of compute time speedup and total time speedup of parallel algorithms for LS calculation with different DEM datasets: (a) 90-m dataset (b) 30-m dataset

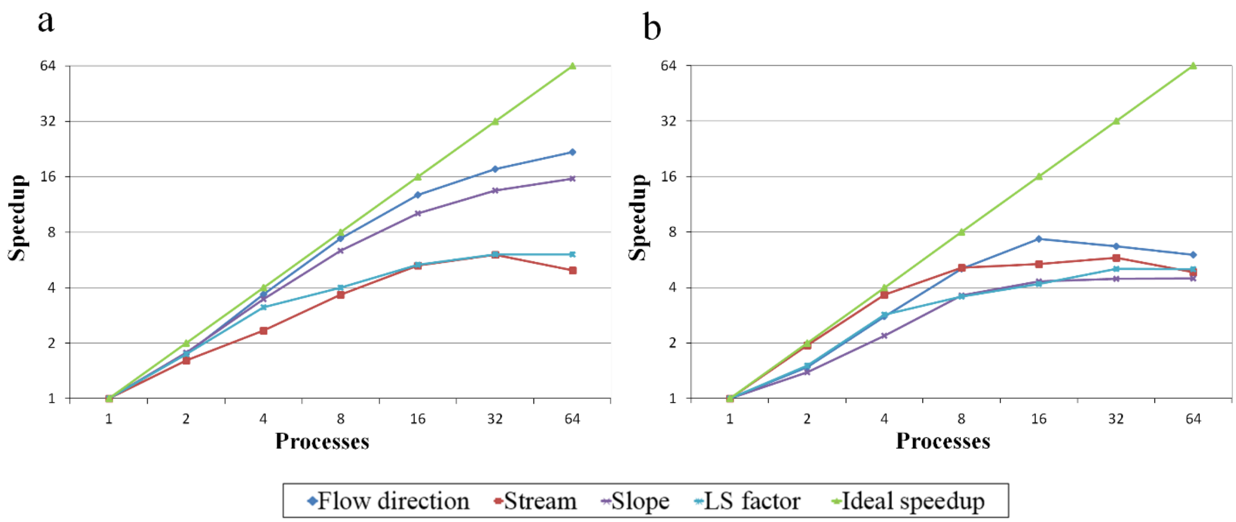

Fig. 4. A comparison of speedup of parallel algorithms for LS calculation with different DEM datasets: (a) 90-m dataset (b) 30-m dataset

4. Conclusions

With the improvement of DEM resolution and the extension of the research area, serial algorithms are insufficient, sometime even incapable to process the massive terrain datasets. In this paper, a MPI algorithm is developed for calculating the LS factor which in turn includes the calculation of slope, flow direction, stream, and final LS factor. From the experimental results on the multi-node cluster the following findings are observed: (1) the parallel implementation of the LS factor calculation dramatically reduces the computation time, even with the datasets which completely overwhelms the serial recursive implementation; (2) Due to the influence of communication overhead, the parallel efficiency of global algorithms is inferior to that of local algorithms; and (3) the parallelized LS factor algorithm achieves a higher compute time speedup with the increase of data volume especially when the number of process is not very large. However the increase of the I/O overhead has a strong negative influence on the total time speedup.

5. References

Guan, Q.F. and Clarke, K.C., 2010. A general-purpose parallel raster processing programming library test application using a geographic cellular automata model. *International Journal of Geographical Information Science*, 24 (5), 695–722.

Jiang, L., Tang, G.A., Liu, X.J., Song, X.D., Yang, J.Y., Liu, K., 2013. Parallel contributing area calculation with granularity control on massive grid terrain datasets. *Computers & Geosciences*. 60, 70–80.

Qin, C.Z., Zhan L.J., Zhu A.X., Zhou, C.H., 2014. A strategy for raster-based geocomputation under different parallel computing platforms. *International Journal of Geographical Information Science*, 28 (11), 2127-2144.

Tesfa, T.K., Tarboton, D.G.,Waston, D.W., Schreuders, K.A.T., Baker, M.E.,Wallace, R. M., 2011. Extraction of hydrological proximity measures from DEMs using parallel processing. *Environmental Modeling &Software*. 26(12), 1696–1709.