Environmental Modelling & Software 80 (2016) 175–184

Contents lists available at ScienceDirect

Environmental Modelling & Software

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

A two-level parallelization method for distributed hydrological models

Junzhi Liu [a], [b], A-Xing Zhu [a], [b], [c], [d], *, Cheng-Zhi Qin [b], [c], **, Hui Wu [e], [f], Jingchao Jiang [e], [f]

[a] Key Laboratory of Virtual Geographic Environment, Ministry of Education, Nanjing Normal University, 1 Wenyuan Road, Nanjing, Jiangsu 210023, China
[b] Jiangsu Center for Collaborative Innovation in Geographical Information Resource Development and Application, 1 Wenyuan Road, Nanjing, Jiangsu 210023, China
[c] State Key Lab of Resources and Environmental Information System, Institute of Geographic Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing, 100101, China
[d] Department of Geography, University of Wisconsin–Madison, Madison, WI 53706, USA
[e] Hangzhou Dianzi University, 115 Wenyi Road, Hangzhou, Zhejiang, 310012, China
[f] Smart City Research Center of Zhejiang Province, 115 Wenyi Road, Hangzhou, Zhejiang, 310012, China

ARTICLE INFO

Article history:
Received 29 October 2015
Received in revised form
28 February 2016
Accepted 28 February 2016
Available online xxx

Keywords:
Distributed hydrological model
Two-level parallelization
Multi-core cluster
Sub-basin
Basic simulation unit

ABSTRACT

This paper proposes a scalable two-level parallelization method for distributed hydrological models that can use parallelizability at both the sub-basin level and the basic simulation-unit level (e.g., grid cell) simultaneously. This approach first uses the message-passing programming model to dispatch parallel tasks at the sub-basin level to different nodes with multi-core CPUs in the cluster. Each node is responsible for some of the sub-basins. Parallel tasks for each sub-basin at the basic simulation-unit level are then dispatched to multiple cores within each node using the shared-memory programming model. A grid-based distributed hydrological model was parallelized to demonstrate the performance of the proposed method, which was tested in different scenarios (e.g., different data volume, different numbers of sub-basins). Results show that the proposed two-level parallelization method had better scalability than the parallel computation at sub-basin level alone, and the parallel performance increased with data volume and the number of sub-basins.

Software availability

Program title: TwoLevelParHydroModel.

Description: Two-level parallel computing for distributed hydrological modeling.

Developer: Dr. Junzhi Liu and Prof. A-Xing Zhu.

Platform: Linux.

Source language: C++

Cost: Free.

Availability: Contact the developer.

1. Introduction

There are several emerging trends in distributed hydrological modeling, which plays an important role in watershed research and management (Borah and Bera, 2004). First, high spatial resolutions are needed to characterize the detailed spatial distribution of hydrological variables such as soil moisture and soil erosion (Rojas et al., 2008; Chen et al., 2013). Second, long-term simulations are needed to assess the impacts of land-use change and climate change on watershed sustainable development (Bhaduri et al., 2001; Barnett et al., 2005). Third, integrated simulations coupling multiple geographic processes are needed in order to evaluate the effects of watershed management practices comprehensively (Zoltay et al., 2010). To conduct such distributed hydrological

---

* Corresponding author. Key Laboratory of Virtual Geographic Environment, Ministry of Education, Nanjing Normal University, 1 Wenyuan Road, Nanjing, Jiangsu 210023, China.

** Corresponding author. State Key Lab of Resources and Environmental Information System, Institute of Geographic Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing, 100101, China.

E-mail addresses: axing@njnu.edu.cn (A.-X. Zhu), qincz@lreis.ac.cn (C.-Z. Qin).

http://dx.doi.org/10.1016/j.envsoft.2016.02.032
1364-8152/

modeling over long periods, a large amount of computation are required. This makes the parallelization of distributed hydrological models an inevitable choice (Vivoni et al., 2011).

In distributed hydrological modeling, watersheds are divided into different types of simulation units, such as sub-basins, grid cells and hydrological response units (Dehotin and Braud, 2008). It is a common way to dispatch computing tasks of different simulation units to multiple processors for parallel computing (Li et al., 2011). For example, parallel-computing methods at different levels (e.g., sub-basin or grid cell) using different types of computer hardware (e.g., cluster or multi-core CPU [Central Processing Unit]) have been proposed and proven to be effective with speedup ratios ranging from 2 to 80 (Kollet and Maxwell, 2006; Vivoni et al., 2011; Wang et al., 2011; Ran et al., 2013; Yalew et al., 2013; Burger et al., 2014; Liu et al., 2014).

However, the scalability of parallel computing using the parallelizability among simulation units at a single level is usually limited. For example, there exist maximum speedup ratios (MSRs) for parallel computing at the sub-basin level (Liu et al., 2013). When the number of processors exceeds certain thresholds, the speedup ratio will not keep increasing due to the load imbalance caused by limited numbers of sub-basins and the dependences among them (Wang et al., 2012; Liu et al., 2013). Parallel computing at the grid-cell level also has scalability problems mainly due to communication overhead (Liu et al., 2014). How to address the scalability problem and improve the efficiency of parallel computing for distributed hydrological models has become an important research topic.

This paper proposes a two-level parallelization method for distributed hydrological models by simultaneously utilizing the parallelizability at both the sub-basin and the basic simulation unit levels, and in this way improve parallel-computing scalability. Section 2 describes the basic concept of the two-level parallelization method. Section 3 describes the implementation of the method, taking a grid-based distributed hydrological model as an example. Section 4 validates the effectiveness of the proposed method through a case study. Section 5 concludes and discusses future research directions.

2. Basic concept of the two-level parallelization method

For distributed hydrological modeling, a watershed can be divided into spatial units of different levels. The spatial units from top to bottom are usually sub-basins, hillslopes and basic simulation units such as grid cells and hydrological response units (Band et al., 2000). Parallel computing for distributed hydrological modeling can be conducted at different levels. The granularity of parallel tasks, which refers to the number of computations for each parallel task and the communication overhead among parallel tasks, are different for parallel computing at different levels. Parallel computing at the sub-basin and hillslope levels is coarse grained, with a relatively low number of parallel tasks and low communication overhead, and it is suitable to run on computer clusters. Parallel computing at this level has good scalability with respect to hardware, but suffers from limited parallelizability caused by the limited number of parallel tasks and the dependences among them (Wang et al., 2012). In contrast, parallel computing at the basic simulation-unit level (e.g., grid-cell level) is fine-grained, with a relatively large number of parallel tasks but also with high communication overhead. Although there is good parallelizability at this level, it requires an extensive level of communication and so need to run on shared-memory hardware. The hardware environment is usually the limiting factor for parallel computing at this level (Liu et al., 2014). The coarse- and fine-grained parallel computing methods described above are complementary on the aspects of parallelizability and hardware. Combining them to conduct two-level parallel computing may lead to better parallel-computing performance (Zhao et al., 2013).

Because parallel computing at the sub-basin level is suitable to run on computer clusters, and at the basic simulation-unit level it is suitable to run on shared-memory hardware, multi-core cluster (i.e. a cluster having nodes with multi-core CPUs), is selected as the hardware platform for the two-level parallel computation in this paper. With the development of multi-core CPUs, most current computer clusters are multi-core clusters, which made the proposed method in this study widely applicable. The architecture of the two-level approach is as follows: first, parallel computing at the sub-basin level is conducted among multiple nodes in a computer cluster using the message-passing programming model, and then parallel computing at the basic simulation-unit level is conducted among multi-cores within each node using the shared-memory programming model (Fig. 1). It is worth noting that although a job management systems like PBS (Portable Batch System) could also be used to conduct the coarse-grained parallel computation, the intense interaction among parallel tasks at the sub-basin level requires a flexible communication mechanism and makes the message-passing parallel computation (e.g. using MPI (Message Passing Interface)) a necessary choice.

3. Implementation of the two-level parallelization method

The two-level parallelization method is a combination of parallel computing at the sub-basin level and at the basic simulation-unit level. To illustrate the implementation of this method, a grid-based distributed hydrological model was parallelized as an example.

3.1. Description of a grid-based distributed hydrological model

The grid-based distributed hydrological model we used was developed by Liu et al. (2014), which integrates several existing methods to conduct event-based hydrological simulation in semi-arid watersheds. The hydrological processes simulated in this model include interception, infiltration, surface depression, overland-flow routing and channel-flow routing. The interflow- and groundwater-flow processes were omitted because runoff generation is dominated by infiltration excess overland flow in semi-arid areas.

The infiltration process was simulated using a quadratic approximation of the Green-Ampt method (Li et al., 1975). The interception and depression processes were simulated using methods in the Wetspa model (Liu and De Smedt, 2005). For simulation of the interception process, the fill-and-spill mechanism was used and the maximum interception storage was calculated using a statistical equation containing leaf area index, vegetal species, and date (Liu and De Smedt, 2004). To simulate the depression process, depression and overland flow were allowed to occur simultaneously even if excess rainfall was less than the depression storage, which was estimated by an empirical equation suggested by Linsley et al. (1975).

Both overland-flow routing and channel-flow routing were performed sequentially from upstream to downstream according to the single-flow direction defined by the D8 method (O'Callaghan and Mark, 1984). Surface flow was simulated by a one-dimensional kinematic wave model combined with the Manning's equation (Chow et al., 1988). The equation for surface-water depth is