Journal of Hydrology 535 (2016) 737–747

Contents lists available at ScienceDirect

Journal of Hydrology

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

Double-layer parallelization for hydrological model calibration on HPC systems

Ang Zhang [a], Tiejian Li [a,*], Yuan Si [a], Ronghua Liu [a,b], Haiyun Shi [a,c], Xiang Li [a,b], Jiaye Li [a], Xia Wu [d]

[a] State Key Laboratory of Hydroscience and Engineering, Tsinghua University, Beijing 100084, China
[b] China Institute of Water Resources and Hydropower Research, Beijing 100038, China
[c] Department of Civil Engineering, The University of Hong Kong, Pokfulam, Hong Kong, China
[d] PetroChina Beijing Natural Gas Pipeline Company, Beijing 100012, China

ARTICLE INFO

Article history:
Received 3 June 2015
Received in revised form 22 October 2015
Accepted 8 January 2016
Available online 14 January 2016
This manuscript was handled by Geoff Syme, Editor-in-Chief, with the assistance of Bellie Sivakumar, Associate Editor

Keywords:
Digital Yellow River Integrated Model
Genetic algorithm
HPC job scheduling
Parallel computing
Parameter calibration

SUMMARY

Large-scale problems that demand high precision have remarkably increased the computational time of numerical simulation models. Therefore, the parallelization of models has been widely implemented in recent years. However, computing time remains a major challenge when a large model is calibrated using optimization techniques. To overcome this difficulty, we proposed a double-layer parallel system for hydrological model calibration using high-performance computing (HPC) systems. The lower-layer parallelism is achieved using a hydrological model, the Digital Yellow River Integrated Model, which was parallelized by decomposing river basins. The upper-layer parallelism is achieved by simultaneous hydrological simulations with different parameter combinations in the same generation of the genetic algorithm and is implemented using the job scheduling functions of an HPC system. The proposed system was applied to the upstream of the Qingjian River basin, a sub-basin of the middle Yellow River, to calibrate the model effectively by making full use of the computing resources in the HPC system and to investigate the model's behavior under various parameter combinations. This approach is applicable to most of the existing hydrology models for many applications.

© 2016 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

Distributed hydrological models have been extensively used in water resource management (Beven, 2001; Cibin et al., 2014; Singh and Prevert, 2002). A significant goal of research efforts in hydrological modeling is to improve the accuracy of the model applications. Parameters that quantify the hydrological mechanism are the key factors that influence the accuracy of model simulations (Muleta and Nicklow, 2005). However, some of the parameters are costly or cannot be feasibly determined by using field measurements directly. Therefore, model calibration, which is defined as the adjustment of model parameters within recommended ranges to optimize the agreement between observed data and simulated results, is an essential procedure prior to the application of hydrological models (Cooper et al., 2007; Tolson and Shoemaker, 2007). Moreover, parameter sensitivity analysis accompanied by model calibration helps model users and developers understand the hydrologic processes and enhance the adaptability of the simulation models (Bahremand and De Smedt, 2008; Zhan et al., 2013).

Trial-and-error is a traditional way to determine model parameters. However, this tedious technique requires the considerable experience of modelers and researchers and is therefore much more difficult for new users and new models. In contrast, automated calibration is more efficient than manual calibration and has been more frequently used in recent years (Liu, 2009). In an automated method, a specified search scheme is used to find the most suitable combination of parameters and is guided by the results of an evaluation of each simulation (Madsen, 2000). The performance of automated model calibration depends on the choice of the search strategy and the exploitation of computing resources.

Various search strategies for the optimal combination of parameters have been devised and achieved for model calibration. One such development is the use of the genetic algorithm (GA) (Goldberg, 1989; Holland, 1975; Konak et al., 2006; Wang, 1991), which has been reported as an efficient and robust means of model calibration, including both conceptual and distributed hydrological models. Most importantly, the parallel nature of GA makes it

* Corresponding author. Tel.: +86 10 62789469; fax: +86 10 62772463.
E-mail addresses: zhanga11@mails.tsinghua.edu.cn (A. Zhang), litiejian@tsinghua.edu.cn (T. Li), siy13@mails.tsinghua.edu.cn (Y. Si), waxle@sina.com (R. Liu), shihaiyun@tsinghua.edu.cn (H. Shi), ideal.thu@gmail.com (X. Li), li-jy13@mails.tsinghua.edu.cn (J. Li), wuxiazuishuai@163.com (X. Wu).

http://dx.doi.org/10.1016/j.jhydrol.2016.01.024
0022-1694/© 2016 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

suitable for parallel computing. Another strategy is the shuffled complex evolution (SCE-UA) algorithm (Duan et al., 1992), which combines a local downhill algorithm with a global shuffling strategy. Recently, different techniques were coupled with artificial neural network (ANN) to improve the ANN performance in the estimate of daily flows (Chau and Wu, 2010; Taormina and Chau, 2015; Wang et al., 2015b; Wu et al., 2009). Other approaches for the calibration of popular hydrological models, such as the Soil and Water Assessment Tool (SWAT), have also been proposed in recent years (Cibin et al., 2010; Cibin and Chaubey, 2015; Wu and Liu, 2012, 2014; Zhang et al., 2013).

With the increasing application of automated model calibration techniques, the computing time problem has been increased by the hundreds or even thousands of times that models are executed. Therefore, determining how to exploit more computing resources for model calibration has become a crucial issue. Cheng et al. (2005) proposed a hybrid method that combines a parallel genetic algorithm with a fuzzy optimal model for model calibration, implemented the method on massive parallel computers with single instruction multiple data stream and reduced the time–cost of calibration while maintaining similar results. Sharma et al. (2006) indicated that parallel versions of model calibration algorithms can be successfully applied to complex hydrological models by employing high performance computing (HPC) systems and the computational times on two computer clusters containing 128 and 48 processors, respectively, have been effectively reduced.

Meanwhile, parallelization techniques, such as the Message Passing Interface (MPI) and the Open Multi-Processing (OpenMP) application program interface, have been widely applied to hydrological models themselves to reduce their execution times (Chapman et al., 2007; Wu et al., 2013). A parallelized model can exploit every processor core in a multi-core computer or a cluster of computers with simultaneously executed model processes, one on each core. For each model process, a sub-domain of a river basin is simulated. According to the pattern that a basin is depicted, hydrological models can be divided into several categories, including subbasin-based, hillslope–channel based, and grid-based models. For subbasin-based hydrological models, their parallelization could be realized by decomposing a basin into large subbasins, each of which constituted of a number of small subbasins. In contrast, the parallelization of grid-based hydrological models is much more complicated, since their domain decomposition can be achieved according to either subbasins or regular sub-grids. However, for subbasin-based parallelization of hydrological models, their improvements in simulation speed are limited by the longest upstream-to-downstream flow routing calculation that must be carried out in serial along the main stem of a river basin (Li et al., 2011; Tang et al., 2010).To date, the major challenge for the calibration of parallel hydrological models is to break the limitation of the inevitable serial simulation along the main stem, especially when the calibration is carried out for a large river basin with high-resolution units, and for a large number of parameters. Therefore, to meet the challenge, a higher layer parallelism of model execution with different parameter combinations is highly promising to improve the overall efficiency of model calibration. The primary objective of this study is to propose a double-layer parallelization for hydrological model calibration. The expected benefits of the proposed methods are (i) to reduce the time required for model calibration, and (ii) to facilitate the analysis of parameter sensitivity.

In this paper, the example parallel hydrological model that to be calibrated is the physically-based Digital Yellow River Integrated Model (DYRIM). The DYRIM can simulate the hydrological and sediment processes of a large river basin based on hillslope–channel units in the drainage network extracted from digital elevation model (DEM) data (Wang et al., 2007, 2012a). Former studies of the DYRIM have proved its capability of hydrological simulations in the Yellow River basin and more other basins (Shi et al., 2015; Wang et al., 2015a). Recently, to improve the preparedness of hydrological simulation for any river basin in the world, we have extracted the drainage networks of most large river basins from the 30-m-resolution ASTER GDEM using a high-efficient eight-direction method (Bai et al., 2015), and composed a global drainage network database named as HYDRO30, following the resolution of its source DEM. A modified binary-tree codification method (Li et al., 2010) was developed to code and identify each river reach in the drainage networks.

For the case study, the upstream of the Qingjian River with a drainage area of 930 km^2 is selected. The Qingjian River is a tributary of the Yellow River, located on the highly erodible Loess Plateau in north China. There are nearly 8000 river reaches (i.e., hillslope–channel units) in the extracted drainage network of the upper Qingjian River basin. The average area of the hillslope–channel units is about 0.1 km^2. The high-resolution drainage network can result in a great increase of the computational time of physically-based hydrological simulation when the basin is larger, and thus, the double-layer parallelization is acutely needed to improve the overall efficiency for the calibration of parallel hydrological models.

2. Methods

2.1. Framework of the double-layer parallelization

The optimization of model parameters is parallelized in two layers on an HPC system. A dynamic sub-basin decomposition method (Li et al., 2011) was developed to parallelize the hydrological simulation of the DYRIM, which contributes to the lower-layer parallelism. The MPI standard is adopted to realize the lower-layer parallelism, mainly because it is the dominant technique to develop parallel programs on distributed memory systems that most HPCs belong to. The job scheduling functions of an HPC system are used to manipulate simultaneous model executions with different parameter combinations in a same generation of an optimization algorithm, which contributes to the upper-layer parallelism. Thus, the system of the double-layer parallelization can be built as shown in Fig. 1. Therefore, three key questions should be solved: (i) How to submit several simulation jobs and recognize their status on an HPC system? (ii) How to assign different hydrological parameters for different simulation jobs? (iii) How to save the simulation results and calculate the relevant evaluation criteria? These questions will be addressed by the technological innovations proposed in the following subsections.

2.2. Genetic algorithm and its parallelization using the HPC job scheduling

The genetic algorithm (GA), firstly proposed by Holland (1975), is a kind of heuristic techniques for solving optimization problems. Based on the mechanics of natural selection and evolution, GAs have been developed into a powerful search approach. GAs have the following advantages: (1) GAs are a multiple-point-based search approach, (2) the search scheme of GAs is directly guided by the objective function, and (3) GAs employ probabilistic transition rules to avoid local optima. Because of these advantages, GAs are recognized as a robust method for complex problems (Wendt et al., 2010; Wang et al., 2012b).

Among other optimization algorithms the GA was chosen in this study because of its stability, natural parallelism and problem-independence. The GA implementation employed in this paper treats the parameters to be optimized as real numbers with simu-

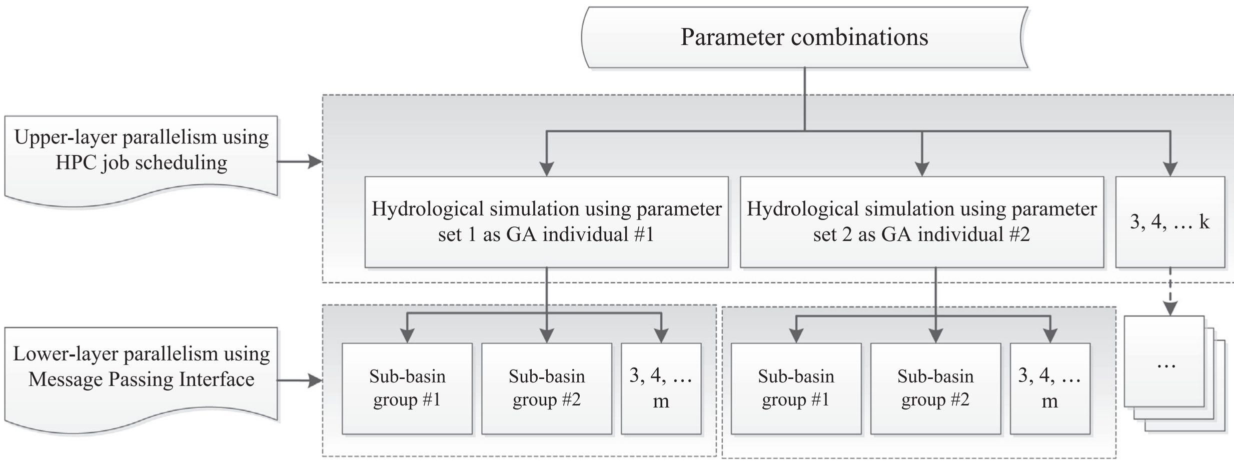

Fig. 1. Framework of the double-layer parallelization.

lated binary crossover and real-parameter mutation (Deb et al., 2000). This technique promises the parameters independent of the GA and easy to be optimized. The efficiency of the GA itself will not be discussed in this paper.

A GA starts with a population of candidate solutions, which are also referred to as chromosomes or individuals. The chromosomes are combined to reproduce fitter chromosomes by three genetic operators, i.e., selection, crossover and mutation. Based on a probability criterion, the selection operator selects some high fitness chromosomes to be used for crossover. The crossover operator combines different pieces of genetic information from the selected chromosomes. The mutation operator allows some pieces of the genetic information to be randomly changed within allowable ranges. As iterations successively proceed, the fitness of a population progressively improves until no further improvement can be found. In particular, GAs become more attractive when the evaluation of an individual takes considerable computation time.

High performance computing (HPC) is a branch of computer science, and it generally implies a computing system and environment that uses many processors (Hager and Wellein, 2010). HPC research aims to build supercomputers, study parallel algorithms and develop relevant software. Most HPC systems consist of a cluster of computers that are interconnected by a high performance network (e.g., the Quad Data Rate (QDR) InfiniBand network), which promises high-bandwidth and low-latency data transfer among computers. In hydrological modeling, an HPC is generally used when a parallel hydrological model is applied to a large river basin (Wu et al., 2013) or an automated parameter calibration model is applied for a serial hydrological model (Zhang et al., 2013).

In this paper, the double-layer parallelization for hydrological model calibration is proposed to fully exploit the parallelism in both the optimization model and the hydrological model itself. A flow chart of the proposed method using an HPC system is shown in Fig. 2. The outer layer is the GA. When the GA needs to evaluate the fitness of various model parameter combinations, four procedures are used to manage multiple parallel DYRIM simulations.

First, model parameters are updated in the DYRIM database, with one database user for each GA individual. Second, the job scheduler of the HPC system is called to put a number of DYRIM jobs into the job list of the HPC. A job scheduler is the software interface of an HPC system, which allows users to submit, monitor and manipulate their calculation jobs. Using the job scheduler is the most convenient and high-efficient way to access HPCs. For model calibration, the number of jobs is equal to the population of the GA. If the HPC has enough available computing resources

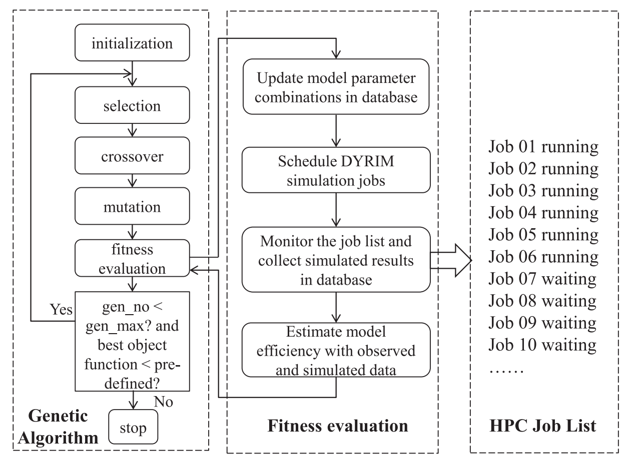

Fig. 2. The flow chart of the model calibration using the job scheduling functions of an HPC system.

to allocate those DYRIM simulations, all of the jobs will be performed simultaneously. If not, some jobs will be running, and the others will be waiting. Therefore, the third procedure is to monitor the job list and collect the simulation results from each finished job. When all of the jobs have completed, the last procedure makes the model efficiency estimations using the observed data and proposes the fitness evaluation list. After the fitness evaluation, a new generation of the GA is proposed to explore more parameter combinations until the stop criterion or the maximum number of generations is reached.

Fig. 3 presents the pseudo-code of the job management functions through the HPC job scheduling. The questions mentioned before were solved by the functions in the coding. The functions that are independent of the hydrological model and the optimization algorithm are indicated. Therefore, the advantages of a simple interface and good compatibility can be achieved.

3. Application

3.1. Key parameters of the DYRIM

DYRIM takes each river reach, which includes two to three hillslopes and one channel, as the basic hydrological unit because of the different hydrological response mechanisms on hillslopes and in channels. The runoff-yield model is established on each hillslope

```
#import <Microsoft.Hpc.Scheduler.tlb>
#import <Microsoft.Hpc.Scheduler.Properties.tlb>
//the type libraries of the HPC scheduler interface, which are included in the Microsoft HPC Pack 2012 SDK.
class JobManagement//Each job runs with its own database user, has its own JobManagement class instance, and can be identified
by its database UserId or the pointer pJob.
{
public:
        bool Initialize(struct parameter Parameters, string UserId, string Password);//initialize a JobManagement instance with assigned
                model parameters and database user name and password.
        struct criteria Do();//do the job and returns the evaluation criteria
private:
        IScheduler* pScheduler;//pointer to the HPC scheduler
        ISchedulerJob* pJob;//pointer to a job
        ISchedulerTask* pTask;//pointer to a task
        string thisUserId;
        string thisPassword;
        struct parameter thisParameters;
        struct criteria thisCriteria;
        bool SetParameters();//set parameters in the DYRIM model database for a new job
        bool IsFinished();//monitor the job running status and do following works
        bool ObtainSimulations();//obtain the simulation results of a job
        bool CalcCriteria();//calculate the evaluation criteria of a job's results
}
bool JobManagement::IsFinished()
{
        JobState pJobState;
        string StateInfo;
        while (true)
        {
                pJob->get_State(&pJobState);
                StateInfo = GetJobStateString(pJobState);
                if(StateInfo == "Submitted" || StateInfo == "Queued" || StateInfo == "Running" || StateInfo == "Finishing")//the job is
                        waiting or running
                        Sleep(2000);//sleep for a while before checking the job' s state again
                else if(StateInfo == "Finished")
                {
                        ObtainSimulations();
                        CalcCriteria();
                        break;
                }
                Else//including StateInfo == "Failed" || StateInfo == "Canceled" || StateInfo == "Canceling"
                        exit(ERROR_CODE5);
        }
}
struct criteria JobManagement::Do()
{
        HRESULT hr;
        SetParameters();
        hr = pScheduler->Connect(string HPC_Head_IP);//connect to the HPC Cluster Manager
        if(FAILED(hr)) exit(ERROR_CODE1);
        hr = pScheduler->CreateJob(&pJob);//create a new job
        if(FAILED(hr)) exit(ERROR_CODE2);
        hr = pJob->CreateTask(&pTask);//create a new task
        if(FAILED(hr)) exit(ERROR_CODE3);
        pTask->put_CommandLine(CommandLine);//set task parameters
        pTask->put_WorkDirectory(WorkDirectory);
        pTask->put_StdOutFilePath(OutputFilePath);
        pTask->put_StdErrFilePath(ErrorFilePath);
        pTask->put_MaximumNumberOfCores(MaxNumberofCores);
        pTask->put_MinimumNumberOfCores(MinNumberofCores);
        pJob->AddTask(pTask); //add the task to the job
        hr = pScheduler->SubmitJob(pJob, HPCUserId, HPCPassword);//submit the job, which will automatically start running when
                the HPC have enough available resource;, otherwise, the job should be in the Queued state
        if(FAILED(hr)) exit(ERROR_CODE4);
        IsFinished();
        return thisCriteria;
        }
```

Fig. 3. Pseudo-code of the job management functions through the HPC job scheduling.

unit, where the soil mass is divided into two layers: topsoil and subsoil (Fig. 4). The bottom of the topsoil layer is parallel to the hillslope surface, and the subsoil layer is triangle-shaped in a longitudinal cross-section of the hillslope. The runoff yield model can basically reflect both the infiltration-excess and storage-excess mechanisms, and it runs at fine time steps (e.g., 6 min). Infiltration-excess runoff on the surface is the most important hydrological process considered in the model, along with related

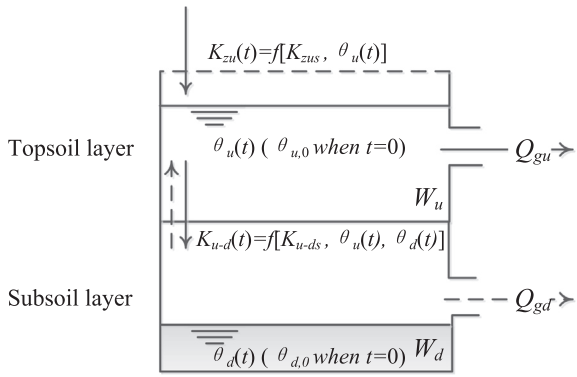

Fig. 4. The hillslope runoff-yield model and the main parameters that are used in DYRIM. In this figure, t is time, W_u is the water storage of the topsoil, Q_{gu} is the topsoil drainage, W_d is the water storage of the subsoil and Q_{gd} is the subsoil drainage. K_{zu} is the vertical conductivity of the topsoil layer, which is a function of K_{zus} (the vertical saturated conductivity of the topsoil layer) and $\theta_u(t)$ (the topsoil moisture). $K_{u\text{-}d}$ is the vertical conductivity between the topsoil and subsoil, which is a function of $K_{u\text{-}ds}$ (the vertical saturated conductivity between the topsoil and subsoil), $\theta_u(t)$ and $\theta_d(t)$ (the subsoil moisture) (from Li et al. (2009a)).

Table 1
The key adjustable parameters in DYRIM.

Parameter	Description	Range of values and step for optimization
K_{zus}	Vertical saturated conductivity of topsoil layer (m/h)	[0.001, 0.01] @ 0.00001
$K_{u\text{-}ds}$	Vertical saturated conductivity between topsoil and subsoil (m/h)	[0.001, 0.1] @ 0.00001
$\theta_{u,0}$	Topsoil initial moisture (m^3/m^3)	[0.001, 0.6] @ 0.00001
R_m	Reference manning's coefficient for river reaches	[0.001, 0.04] @ 0.00001

processes such as vegetation interception, evapotranspiration, ground water discharge and water redistribution between the two soil layers. Runoff from each hillslope flows into the associated channel and then gathers along the drainage network to the outlet.

The parameters of DYRIM can be divided into two groups (Li et al., 2009a, 2009b). The first group includes invariant parameters that are used to describe the properties of the land use and soil types, including the field capacity of the topsoil and subsoil layers. The invariant parameters can be determined from field measurements and handbooks and have less influence on the simulated basin runoff. The other group includes all of the sensitive and adjustable parameters, which must be calibrated before model application using the observed rainfall and runoff data.

The representative adjustable parameters are listed in Table 1, and the important roles of the first three parameters in the runoff-yield process are shown in Fig. 4. The vertical saturated conductivity of the topsoil layer (K_{zus}) controls the surface infiltration rate and primarily influences the infiltration-excess runoff. The vertical saturated conductivity between the topsoil and subsoil ($K_{u\text{-}ds}$) controls the rate of water redistribution between the two soil layers and further acts upon the surface infiltration and evapotranspiration sub-processes in DYRIM. The topsoil initial moisture ($\theta_{u,0}$) is the initial state of the topsoil layer. Moreover, the river manning coefficient (R_m) is related to the river routing processes. Therefore, the combination of those four parameters is optimized with the ranges and steps shown in Table 1.

3.2. Study area

The study area is the upper Qingjian River basin (109°12′–109°44′E, 37°01′–37°19′N). The upper Qingjian River is a tributary of the Yellow River and has a drainage area of 930 km^2. The length of the main stream is 55.6 km. The upper Qingjian River basin is located in a semi-arid region with a mean annual rainfall of 486 mm; the rainfall occurs as short-duration, high-intensity torrential rains in the flood season in the summer (from July to September).

There is one hydrological station in the upper Qingjian River basin, Zichang. There are 11 rainfall stations that are located within and adjacent to the river basin. The spatial distribution of the summer rainfall is complex. Therefore, the upstream sub-basin Zichang station was chosen for the model calibration because of its higher density of rainfall stations. The drainage network and the distributions of the hydrological and rainfall stations of the upper Qingjian River basin upstream Zichang station are shown in Fig. 5. All of the meteorological and hydrological data that were used in this paper were provided by the Hydrographic Bureau of the Yellow River Conservancy Commission.

Model calibration in the study area is more difficult because this semi-arid river basin suffers more complex and sensitive hydrologic processes than in temperate and wet climates. Infiltration-excess flow highly depends on the status of the topsoil layer and varies acutely during the initial phase of each rainfall–runoff event. Therefore, there is a greater demand for short time-step rainfall data for model calibration. Most of the recorded rainfall data in the upper Qingjian River basin are in 2-h time-steps. Although some of the records have finer time-steps, they were all aggregated to 2 h. The time-step for the model simulation and the evaluation of the results was set to 6 min. Two typical rainfall events in 2001 and 2002 were chosen to calibrate the model.

3.3. Evaluation criteria

The Nash–Sutcliffe coefficient of efficiency (NSE) has been used as one of the very popular indices to assess the predictive power of hydrological models. The NSE expresses the correlation of the simulated and measured runoffs as (Nash and Sutcliffe, 1970):

$$NSE = 1 - \left(\sum_{i=1}^{n}(O_i - C_i)^2\right) \Bigg/ \sum_{i=1}^{n}\left(O_i - \overline{O}\right)^2, \tag{1}$$

where C is the simulated data, O is the measured data, and the subscript i represents the sequential number of the simulated and measured data series. NSE approaches 1.0 if the simulated values are quite close to the measured values. An NSE greater than 0 indicates that the means of the model predictions and measured values are still close. An NSE less than 0 indicates that the simulated values cannot be a good prediction of the measured sequence.

For the simulation of flood events, the relative error of the peak discharge value (Re_Q) is commonly used as:

$$Re_Q = |C_{max} - O_{max}|/O_{max}, \tag{2}$$

where the subscript max represents the peak.

Speedup (S_p) and parallel efficiency (E_p) are the two main numerical measures that are used to evaluate the performance of a parallel algorithm (Scott et al., 2005). S_p and E_p are used to indicate the speed advantage and the effectively used fraction of time, which are defined, respectively, as

$$S_p = T_1/T_p \text{ and} \tag{3}$$

$$E_p = S_p/p, \tag{4}$$

where T_1 and T_p are the wall-clock times of the simulation using a corresponding serial model and p processes in parallel, respectively.

The evaluation results of the simulations during the calibration process, including the model parameters and the evaluation criteria values, are automatically stored in output files. They are further used for model validation, parameter sensitivity analysis and the analysis of computing performance.

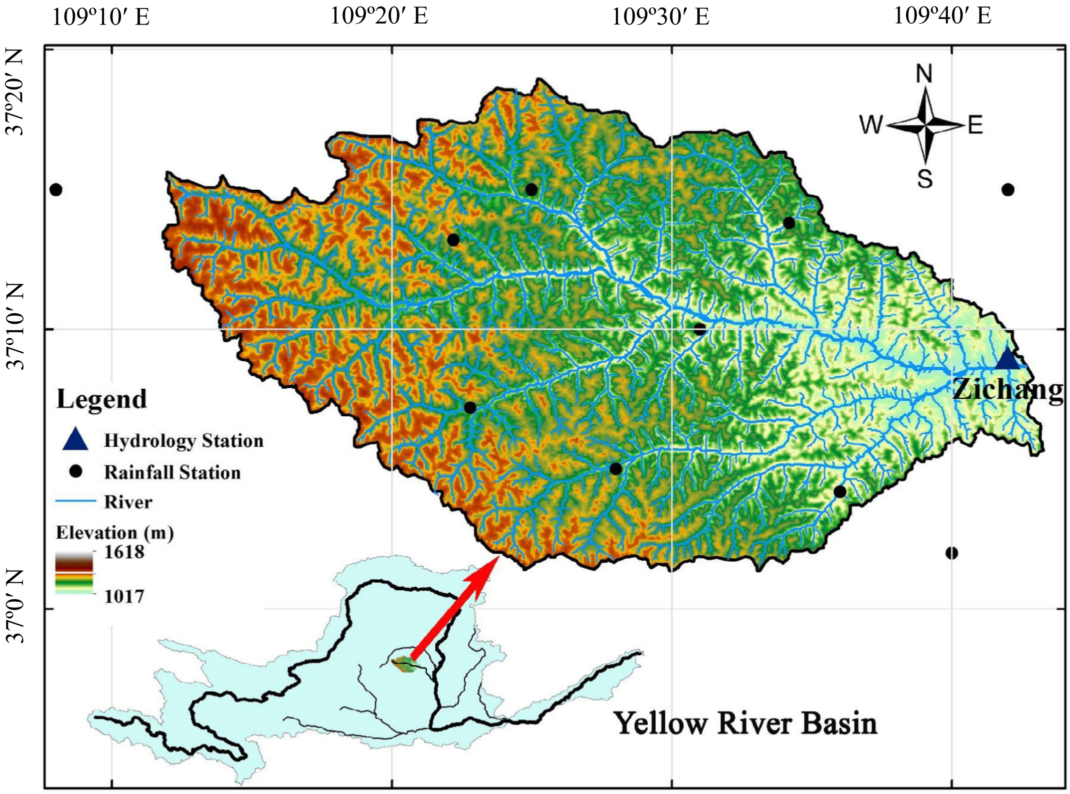

Fig. 5. The upper Qingjian River basin upstream Zichang station and the distribution of the hydrology and rainfall stations.

4. Results

4.1. Lower-layer parallel performance of the DYRIM

The lower-layer of the proposed method, the parallel hydrological model, was first tested on a Windows HPC Server 2012 system with 20 computer nodes. Each node had 24 logical processor cores, and generally, one processor core executed one model process. For parallel hydrological models, the computational time generally is reduced when more processor cores are used. The greatest advantage of using an HPC system for simulations is that a large number of processor cores can be easily used. However, allocating more processor cores for one simulation job means that fewer jobs can be carried out at the same time. Therefore, the number of cores that are allocated for one job is a key factor that needs to be determined; the number of cores has a significant impact on the general efficiency of the double-layer system.

There are three types of processes in the parallel DYRIM, namely, the master process, the slave computing processes, and the data transfer process (Li et al., 2011). The number of slave computing processes (p) for the calculation is the total number of all of the processes minus two, which are the master process and the data transfer process. The cumulative serial time consumption of the DYRIM can be divided into three categories: the time for calculation, the database access time and the communication time. In a given simulation of the upper Qingjian River basin, the number of split-off sub-basins remained unchanged, therefore the cumulative serial time for the calculation was stable (Fig. 6a). However, because of the increasing intensity of the inter-node communications and database accesses that were associated with more computing processes, the cumulative serial times for communication and database access increased with increasing p.

The T_p consumed by one execution of DYRIM for the upper Qingjian River basin for the year 2002 event is plotted against the number of slave computing processes (p) in Fig. 6b. The T_p dropped rapidly with the increasing p in the initial phase, but it remained approximately 70 s when p was greater than seven. S_p reached 4.3 when 22 slave computing processes were used and was unable to further increase (Fig. 7), because in addition to the

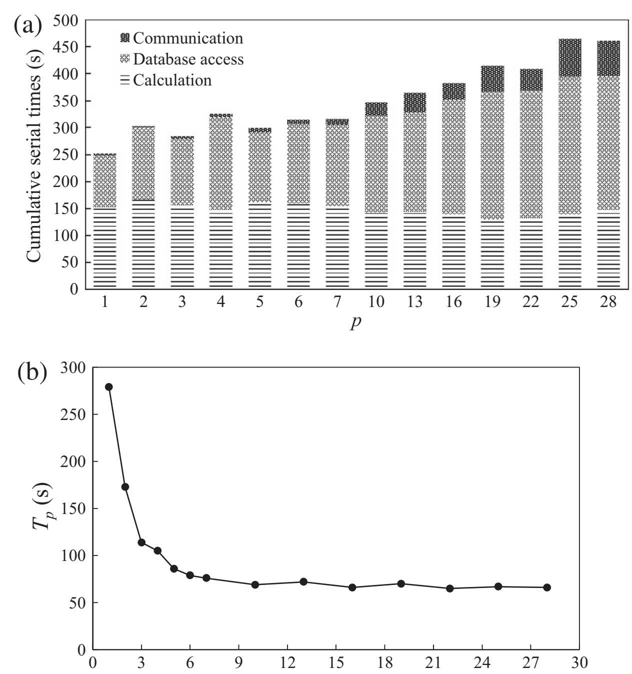

Fig. 6. (a) Different portions of the cumulative serial time for different numbers of slave computing processes. (b) Wall-clock time for different computing processes.

increasing communication and database access times, there was the topological constraint of the river basin. Namely, the longest upstream-to-downstream flow routing calculation along the main stem must be carried out serially. Furthermore, the drop of E_p was caused by the same reasons. Therefore, for applications, there would be a balance between speedup and parallel efficiency. Nevertheless, parallelization greatly improves the efficiency of a single hydrological simulation.

4.2. Double-layer parallel performance of model calibration

The proposed double-layer parallel system was further implemented on the Microsoft Windows Azure cloud computing platform, allocated with a total number of 80 processor cores. From the performance test of the parallel hydrological model, it can be seen in Fig. 7 that the parallel efficiency declines when more computing processes are used. On the contrary, the coarse grain parallelism in the GA is not limited by the topological constraint of a river basin and is more promising to contribute a high efficiency to the double-layer system. Considering the number of individuals in the GA, four processor cores were allocated for each hydrological simulation to test the proposed system. Therefore, at most 20 jobs, namely all 20 individuals in a generation of the GA, could be carried out simultaneously with the allocated resources. With the Windows Azure configuration, the simulation time of the DYRIM with four cores (i.e., two computing processes) for the year 2002 event in the upper Qingjian River basin was 130 s, which was a little shorter than the previous test application shown in Fig. 6b.

To evaluate the performance of the double-layer parallelism, tests were conducted for one generation of the GA, namely 20 DYRIM simulations for the year 2002 event in the upper Qingjian River basin. The total time consumption of the 20 serial simulations (one job at one time) was 2600 s (Fig. 8). As the number of allocated cores increased, more jobs could be submitted at the same time, and the wall-clock time, T_p, for the 20 simulations decreased. For example, it cost 180 s to finish one round of 10 concurrent simulation jobs, then the T_p of the total 20 simulations was 360 s. Finally, the T_p of the double-layer system with 20 concurrent simulation jobs on 80 processor cores was 238 s, and the upper-layer related parallel efficiency, E_p, was 54.6%.

The upper-layer related parallel efficiency decreased with the increase of concurrent simulation jobs (Fig. 8). The loss of performance was generally caused by the increase in the database and communication loads. Moreover, the parallel efficiency was relatively high when the least number of concurrent jobs is configured to complete all the jobs with a same number of job cycles. For example, using seven concurrent jobs is more efficient than eight or nine because the same three job cycles must be carried out for 20 GA individuals. Obviously, when the number of individuals is divisible by the number of concurrent jobs, a high efficiency will

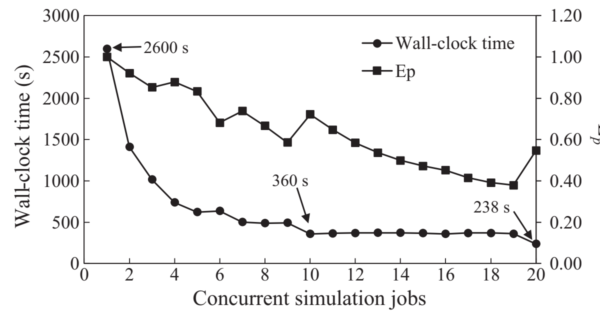

Fig. 8. The time consumptions and the upper-layer related parallel efficiencies of the double-layer system with different numbers of concurrent simulation jobs tested on the Window Azure system for the upper Qingjian River basin.

be obtained. Therefore, to make full use of the HPC resources, there is an optimization problem among the total number of processor cores to be allocated to the double-layer system, the number of cores for one simulation, and the number of GA individuals. In general, if the proposed double-layer parallel system is well configured, a considerable amount of wall-clock time for model calibration can be saved.

4.3. Parameter sensitivity analysis

First, only the parameter K_{zus} (vertical saturated conductivity of topsoil layer) was calibrated for the 2002 rainfall–runoff event, while the other parameters were empirically determined. The response curves of the evaluation criteria, namely *NSE* and Re_Q, with respect to K_{zus} are shown in Fig. 9. There was one peak value of *NSE* in the selected range of K_{zus}, when it equaled 0.005 m/h. The minimum absolute value of Re_Q was found when the K_{zus} was equal to 0.0037 m/h, which was not coincident with that for the peak *NSE*. Solutions for wrapping a multi-objective problem into a GA can be found in previous studies (e.g., Ahmadi et al., 2014; Deb, 2001) and will not be discussed in this paper.

Second, both of the two key parameters, K_{zus} and $K_{u\text{-}ds}$ (the vertical saturated conductivity between the topsoil and subsoil), were calibrated for the 2002 rainfall–runoff event. The response surface of *NSE*, which was derived from the calibration records, is shown in Fig. 10. The comprehensive impact of the two key parameters on the runoff results can be directly observed, where a hilltop can be seen.

Finally, all of the four key parameters were calibrated for both of the 2002 and 2001 events with 20 individuals in 10 generations. In hydrological model calibrations, there is a dilemma that the optimal parameter values will be different when several history events are calibrated together. Sensitivity analysis is useful to

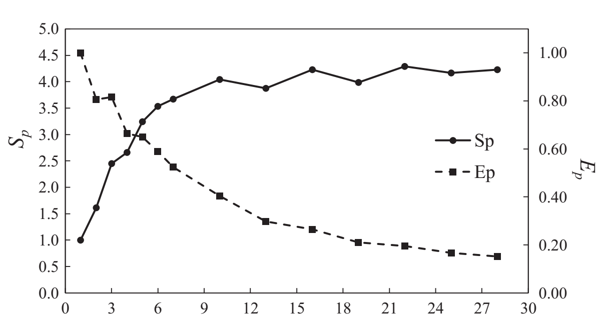

Fig. 7. Speedup and parallel efficiency for different numbers of slave computing processes.

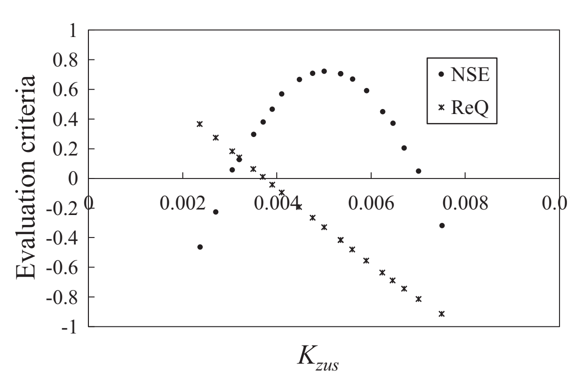

Fig. 9. The response curves of the evaluation criteria with respect to the parameter K_{zus} in the 2002 rainfall–runoff event.

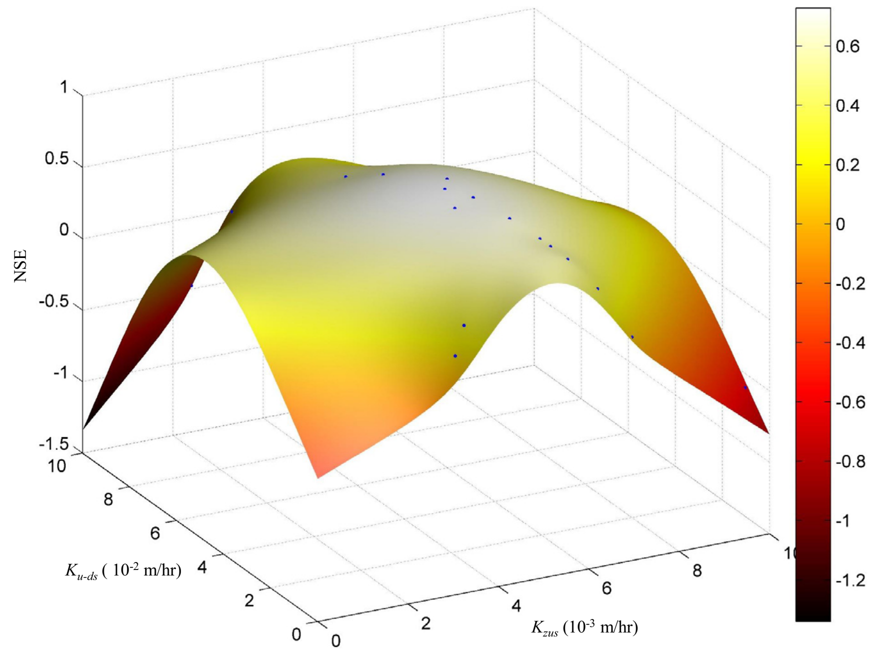

Fig. 10. The response surface of *NSE* with respect to the parameters K_{zus} and $K_{u\text{-}ds}$ in the 2002 rainfall–runoff event.

choose the acceptable parameter values. In this study, the parameters were analyzed using the General Likelihood Uncertainty Estimation (GLUE) method, which is a popular statistical technique to intuitively quantify the uncertainty of model predictions and the sensitivity of parameters (Beven and Binley, 1992). The sensitivity results of the two events are shown in Fig. 11, in which similar distributions of the *NSE*s can be observed. In Fig. 11a, the *NSE* values first show a significant increasing trend as K_{zus} increases, and then they decrease when K_{zus} exceeds a certain value. In Fig. 11b, the *NSE* values show a consistent converging trend with the increase of $K_{u\text{-}ds}$. Throughout the entire selected range of $K_{u\text{-}ds}$, the *NSE* values were always high. However, larger $K_{u\text{-}ds}$ values are recommended, which may help to reduce the occurrence of quite disappointing results. The sensitivity results of the other two parameters are also shown in Fig. 11(c) and (d). The parameters of the topsoil initial moisture and the manning's coefficient of river reaches have less influence than those about infiltration process.

Referring to Fig. 11, the optimal parameters were K_{zus} = 0.005 m/h, $K_{u\text{-}ds}$ = 0.057 m/h, $\theta_{u,0}$ = 0.212 m^3/m^3 and R_m = 0.01. Fig. 12(a) shows the simulated flow discharges of the two events at the Zichang Station with a time step of six minutes compared with the observed data. Similar matches between the simulated and observed discharges of the two events can be seen in Fig. 12(a), and the *NSE* values were 0.65 and 0.72, respectively. Considering the higher magnitude of the year 2001 event, it is reasonable that its simulation result was more sensitive to the model parameters, as shown in Fig. 11.

After calibrating the DYRIM, three more rainfall–runoff events in the years 2002 and 2006 are selected as the verification cases and all of the *NSE* and Re_Q are shown in Table 2. As we can see in Fig. 12, the simulated flow discharges have a flatter shape than the observed values, with an earlier start point of each event. And this leads to the higher error of Re_Q for the calibration and verification cases. This was mainly caused by the rough temporal resolution of the rainfall input, whose time step was two hours. Although the rainfall-excess process in this river basin could be well simulated with the six-minute time step in the model simulation, it was difficult to reproduce the real shape of the flow discharge with the two-hour rainfall data.

4.4. Discussion

The proposed double-layer parallel system used the GA for the parameter optimization of the hillslope–channel based DYRIM. While the DYRIM is parallelized by subbasin-based domain decomposition, a number of parallel DYRIM simulations with different parameter combinations are executed simultaneously on an HPC system to obtain the objective function values for GA individuals in parallel. This double-layer parallelization would contribute to a remarkable promotion of efficiency in the entire calibration, particularly for hillslope–channel based and subbasin-based models, whose subbasin-based parallelization is only limited by the serial simulation along the main stem of a river basin.

To the best of our knowledge, this is the first work to propose such a double layer parallelism for hydrological model calibration. This contribution primarily aimed on the construction and realization of the double layer system, rather than an optimization algorithm. Moreover, the two layers of parallelization are independent from each other. The lower layer can be replaced by any parallel hydrological model using the MPI standard. The upper layer of parallelization is capable of incorporating other optimization algorithms, such as the SCE-UA, because most of those algorithms do not introduce dependencies among the individual simulations in a same generation.

The proposed double-layer parallel system showed its great advantage for the calibration of parallel hydrological models, and would countervail the efficiency loss caused by the inevitable serial simulation along the main stem of a river basin. The time consumptions and the upper-layer related parallel efficiencies of the double-layer system with different numbers of concurrent simulation jobs were analyzed. The concurrent simulation jobs rely on a same database, and then some efficiency loss of the upper layer is caused by the database load, similar with that of a single job. To achieve an overall high-efficiency of the double-layer system, a tradeoff between the two layers can be found. When the number of cores assigned to one hydrological simulation increases, the parallel efficiency will decrease mainly because of the serial simulation along the main stem, determined by the converging structure of a drainage network. When the number of concurrent

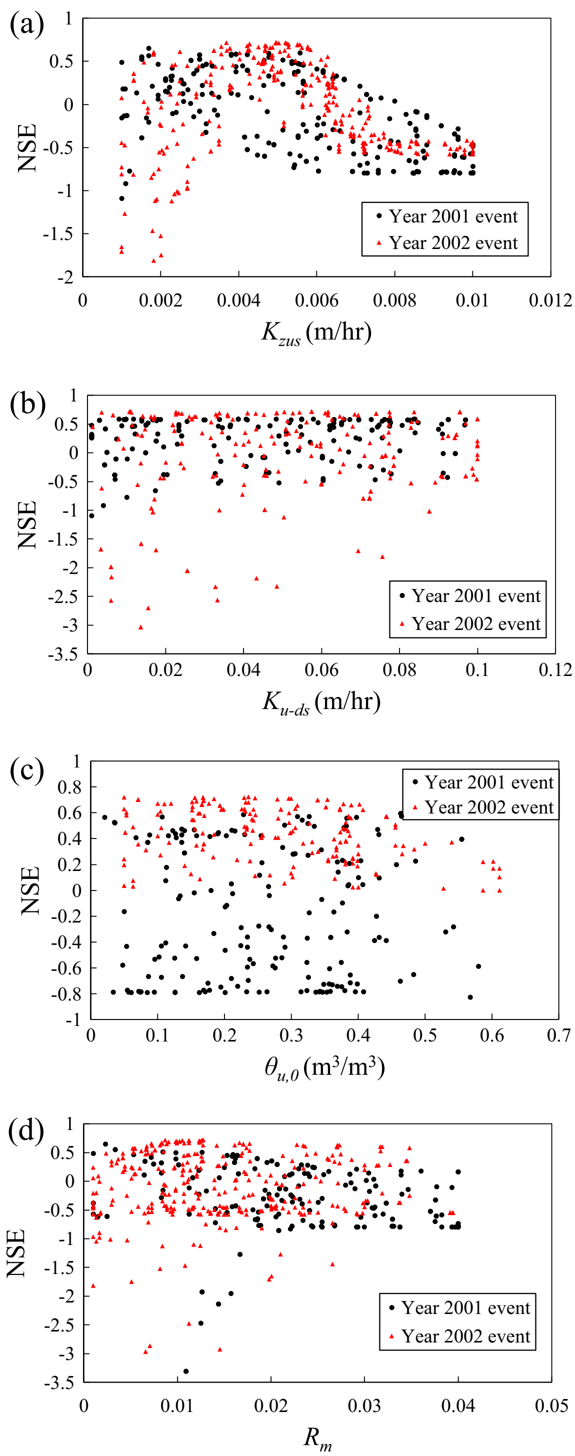

Fig. 11. Scatter plots of the *NSEs* using the General Likelihood Uncertainty Estimation (GLUE) method for the two key parameters: (a) K_{zus} and (b) $K_{u\text{-}ds}$ (c) $\theta_{u,0}$ (d) R_m.

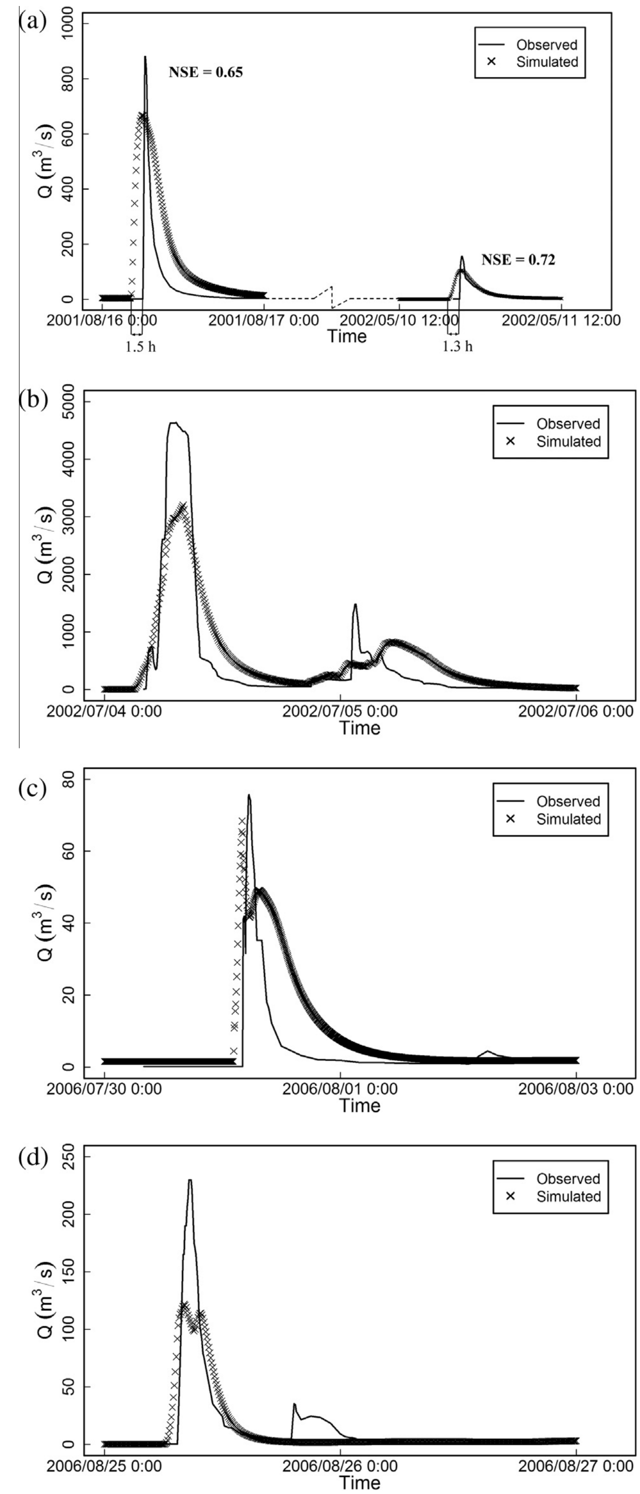

Fig. 12. Observed and simulated runoff in the calibration and verification period: (a) simulation results in calibration period, (b) simulation results in verification period of 2002 event, (c) simulation results in verification period of 2006.07 event and (d) simulation results in verification period of 2006.08 event.

simulations increases, the upper-layer efficiency will also decrease because of the increasing communication traffic and database connections that can be tolerated. Therefore, for a given number of processor cores available in an HPC system, there should be an optimal relation between the number of concurrent jobs and the number of cores assigned for each job. This relation should be explored in further research. For the DYRIM used in this paper, the loss of upper-layer efficiency is generally caused by the increasing load of its database. To improve its efficiency, using a distributed file system for the input/output of the DYRIM is a possible way.

The results of parameter sensitivity analysis indicate that the parameters of the infiltration process have higher sensitivity, while the parameter of the river routing process is not so obvious. The

Table 2
The *NSE* and Re_Q values during the period of model calibration and verification.

	Rainfall–runoff event	*NSE*	Re_Q (%)
Calibration	2001.8.16	0.65	24
	2002.5.10–5.11	0.72	33
Verification	2002.7.4–7.5	0.75	31
	2006.7.30–8.2	0.51	14
	2006.8.25–26	0.60	47

results are not sensitive with the topsoil initial moisture, mainly because the topsoil will be quickly moistened by rainfall. Furthermore, more parameters and additional factors, including the spatial distribution of rainfall, have large impacts on hydrological simulation results. The proposed system can be further used to investigate the influence of such factors.

For the precision of the simulation results, with the scale ranging from monthly to daily, former studies of the DYRIM have achieved pretty good results and the *NSE* could easily reach a satisfactory value, for example, higher than 0.85 (Wang et al., 2015a). What we are endeavoring to do in this paper is to achieve the simulation of flood events with an hourly scale. In the DYRIM, the temporal resolution of runoff simulation is six minutes, but each rainfall data point with a two-hour step is uniformly assigned to corresponding simulation time steps. The time step of rainfall may have a considerable impact on the results, resulting in the flatter shape of simulated flow discharges processes. To better simulate such short-duration high-intensity rainfall–runoff events, rainfall observations with higher spatial and temporal resolutions are badly needed. When finer rainfall records are obtained from concurrent meteorological stations and satellite remote sensing, e.g., with a time step as fine as one minute and a spatial resolution at eight kilometers, the time step of hydrological models will be even shorter than six minutes, and the size of hillslope–channel units will be smaller than 0.1 km^2. In such a situation, the computational consumption of hydrological model calibration will increase dramatically, and the proposed double layer parallelism will be one of the necessities.

5. Conclusions

This paper proposed a double-layer parallel system for hydrological model calibration. The lower-layer was parallelized in a hydrological model by decomposing a river basin into sub-basins, and the upper-layer parallelization was achieved as simultaneous hydrological simulations with different parameter combinations in a certain generation of the genetic algorithm by using job scheduling functions on an HPC system. As an example, the proposed system was applied to optimize some key parameters of a parallel hydrological model, the DYRIM. Moreover, the GLUE method was used to determine the most appropriate parameter values among different simulated events. The results demonstrated that the proposed system can make full use of the computing resources in an HPC system and exploit the parallelisms in both the optimization algorithm and the hydrological model itself. By using the proposed system, both the accuracy and efficiency of model calibration can be achieved. Furthermore, the job management procedure proposed in this paper can be further applied in more complicated circumstances, and then the proposed system can be used for more parallel calibrations. Multiple optimization processes, including different simulated periods, different river basins, and different evaluation criteria can be run in one HPC system simultaneously. The proposed method is also applicable to most of the existing hydrology models parallelized using the MPI standard. Moreover, the Microsoft Windows Azure cloud computing platform was employed here because it is easily accessible, other commercial HPC facilities are also compatible with the proposed method.

Acknowledgments

The study is supported by the National Key Technology R&D Program of China (Nos. 2012BAB02B02, 2013BAB05B03), the State Key Laboratory of Hydroscience & Engineering (No. 2011-KY-4) and the China National Petroleum Corporation (No. 2013B-3410-0508). Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the authors and funders. We are also grateful to the two anonymous reviewers who offered insightful comments leading to the improvement of this paper.

References

Ahmadi, M., Arabi, M., Ascough II, J.C., Fontane, D.G., Engel, B.A., 2014. Toward improved calibration of watershed models: multisite multiobjective measures of information. Environ. Modell. Softw. 59, 135–145.
Bahremand, A., De Smedt, F., 2008. Distributed hydrological modeling and sensitivity analysis in Torysa watershed, Slovakia. Water Resour. Manage. 22, 393–408.
Bai, R., Li, T.J., Huang, Y.F., Li, J.Y., Wang, G.Q., 2015. An efficient and comprehensive method for drainage network extraction from DEM with billions of pixels using a size-balanced binary search tree. Geomorphology. http://dx.doi.org/10.1016/j.geomorph.2015.02.028.
Beven, K., Binley, A., 1992. The future of distributed models: model calibration and uncertainty prediction. Hydrol. Process. 6 (3), 279–298.
Beven, K.J., 2001. Rainfall–runoff Modelling. John Wiley & Sons Press, Chichester.
Chau, K., Wu, C., 2010. A hybrid model coupled with singular spectrum analysis for daily rainfall prediction. J. Hydroinform. 12 (4), 458–473.
Chapman, B., Jost, G., Van Der Pas, R., 2007. Using OpenMP: Portable Shared Memory Parallel Programming. The MIT Press, Massachusetts, USA, p. 353.
Cheng, C.T., Wu, X.Y., Chau, K.W., 2005. Multiple criteria rainfall–runoff model calibration using a parallel genetic algorithm in a cluster of computers. Hydrol. Sci. 50, 1069–1087.
Cibin, R., Athira, P., Sudheer, K.P., Chaubey, I., 2014. Application of distributed hydrological models for predictions in ungauged basins: a method to quantify predictive uncertainty. Hydrol. Process. 28 (4), 2033–2045.
Cibin, R., Chaubey, I., 2015. A computationally efficient approach for watershed scale spatial optimization. Environ. Modell. Softw. 66, 1–11.
Cibin, R., Sudheer, K.P., Chaubey, I., 2010. Sensitivity and identifiability of stream flow generation parameters of the SWAT model. Hydrol. Process. 24, 1133–1148.
Cooper, V.A., Nguyen, V.T.V., Nicell, J.A., 2007. Calibration of conceptual rainfall–runoff models using global optimisation methods with hydrologic process-based parameter constraints. J. Hydrol. 334, 455–466.
Deb, K., 2001. Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons Press, Chichester.
Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2000. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. Lecture Notes Comput. Sci. 1917, 849–858.
Duan, Q.Y., Sorooshian, S., Gupta, V.J., 1992. Effective and efficient global optimization for conceptual rainfall–runoff models. Water Resour. Res. 28, 1015–1031.
Goldberg, D., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, Massachusetts, USA.
Hager, G., Wellein, G., 2010. Introduction to High Performance Computing for Scientists and Engineers. CRC Press.
Holland, J., 1975. Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, Michigan, USA.
Konak, A., Coit, D.W., Smith, A.E., 2006. Multi-objective optimization using genetic algorithms: a tutorial. Reliab. Eng. Syst. Saf. 91, 992–1007.
Li, T.J., Wang, G.Q., Chen, J., 2010. A modified binary tree codification of drainage networks to support complex hydrological models. Comput. Geosci. 36 (11), 1427–1435.
Li, T.J., Wang, G.Q., Chen, J., Wang, H., 2011. Dynamic parallelization of hydrological model simulations. Environ. Modell. Softw. 26, 1736–1746.
Li, T.J., Wang, G.Q., Huang, Y.F., Fu, X.D., 2009a. Modeling the process of Hillslope soil erosion in the Loess Plateau. J. Environ. Inform. 14 (1), 1–10.
Li, T.J., Wang, G.Q., Xue, H., Wang, K., 2009b. Soil erosion and sediment transport in the gullied Loess Plateau: scale effects and their mechanisms. Sci. China Ser. E: Technol. Sci. 52 (5), 1283–1292.
Liu, Y., 2009. Automatic calibration of a rainfall–runoff model using a fast and elitist multi-objective particle swarm algorithm. Expert Syst. Appl. 36, 9533–9538.
Madsen, H., 2000. Automatic calibration of a conceptual rainfall–runoff model using multiple objectives. J. Hydrol. 235, 276–288.

Muleta, M.K., Nicklow, J.W., 2005. Sensitivity and uncertainty analysis coupled with automatic calibration for a distributed watershed model. J. Hydrol. 306 (1–4), 127–145.

Nash, J.E., Sutcliffe, J.V., 1970. River flow forecasting through conceptual models. J. Hydrol. 10, 282–290.

Scott, L.R., Clark, T., Bagheri, B., 2005. Scientific Parallel Computing. Princeton University Press, New Jersey, USA, 374 pp.

Sharma, V., Swayne, D., Lam, D., Schertzer, W., 2006. Auto-calibration of hydrological models using high performance computing. In: Proceedings of the Third Biennial Meeting of the International Environmental Modelling and Software Society, Vermont, USA, 6 pp.

Shi, H.Y., Li, T.J., Liu, R.H., Chen, J., Li, J.Y., Zhang, A., Wang, G.Q., 2015. A service-oriented architecture for ensemble flood forecast from numerical weather prediction. J. Hydrol. 527, 933–942.

Singh, V.P., Prevert, D.K., 2002. Mathematical Models of Large Watershed Hydrology. Water Resources, Highlands Ranch, Colo., p. 891.

Tang, G., D'Azevedo, E.F., Zhang, F., Parker, J.C., Watson, D.B., Jardine, P.M., 2010. Application of a hybrid MPI/OpenMP approach for parallel groundwater model calibration using multi-core computers. Comput. Geosci. 36 (11), 1451–1460.

Taormina, R., Chau, K.W., 2015. ANN-based interval forecasting of streamflow discharges using the LUBE method and MOFIPS. Eng. Appl. Artif. Intell. 45, 429–440.

Tolson, B.A., Shoemaker, C.A., 2007. Cannonsville reservoir watershed swat2000 model development, calibration and validation. J. Hydrol. 337 (1–2), 68–86.

Wang, H., Zhou, Y., Fu, X.D., Gao, J., Wang, G.Q., 2012a. Maximum speedup ratio curve (MSC) in parallel computing of the binary-tree-based drainage network. Comput. Geosci. 38, 127–135.

Wang, G.Q., Fu, X.D., Shi, H.Y., Li, T.J., 2015a. Watershed sediment dynamics and modeling: a watershed modeling system for yellow river. In: Yang, C.T., Wang, L.K. (Eds.), Advances in Water Resources Engineering, Handbook of Environmental Engineering, vol. 14. Springer International Publishing.

Wang, G.Q., Wu, B.S., Li, T.J., 2007. Digital yellow river model. J. Hydro-environ. Res. 1 (1), 1–11.

Wang, Q.J., 1991. The genetic algorithm and its application to calibrating conceptual rainfall–runoff models. Water Resour. Res. 27, 2467–2471.

Wang, W.C., Chau, K.W., Xu, D.M., Chen, X.Y., 2015b. Improving forecasting accuracy of annual runoff time series using ARIMA based on EEMD decomposition. Water Resour. Manage. 29 (8), 2655–2675.

Wang, W.C., Cheng, C.T., Chau, K.W., Xu, D.M., 2012b. Calibration of Xinanjiang model parameters using hybrid genetic algorithm based fuzzy optimal model. J. Hydroinform. 14 (3), 784–799.

Wendt, K., Cortés, A., Margalef, T., 2010. Knowledge-guided genetic algorithm for input parameter optimisation in environmental modelling. Procedia Comput. Sci. 1 (1), 1367–1375.

Wu, C.L., Chau, K.W., Li, Y.S., 2009. Methods to improve neural network performance in daily flows prediction. J. Hydrol. 372 (1), 80–93.

Wu, Y.P., Li, T.J., Sun, L.Q., Chen, J., 2013. Parallelization of a hydrological model using the message passing interface. Environ. Modell. Softw. 43, 124–132.

Wu, Y.P., Liu, S.G., 2012. Automating calibration, sensitivity and uncertainty analysis of complex models using the R package Flexible Modeling Environment (FME): SWAT as an example. Environ. Modell. Softw. 31, 99–109.

Wu, Y.P., Liu, S.G., 2014. A suggestion for computing objective function in model calibration. Ecol. Inform. 24, 107–111.

Zhan, C.S., Song, X.M., Xia, J., Charles, T., 2013. An efficient integrated approach for global sensitivity analysis of hydrological model parameters. Environ. Modell. Softw. 41, 39–52.

Zhang, X.S., Beeson, P., Link, R., Manowitz, D., Izaurralde, R.C., Sadeghi, A., Thomsona, A.M., Sahajpalc, R., Srinivasan, R., Arnold, J.G., 2013. Efficient multi-objective calibration of a computationally intensive hydrologic model with parallel computing software in Python. Environ. Modell. Softw. 46, 208–218.