Dear Reviewer,

Thank you very much for reviewing our manuscript. We also want to thank you for your insightful comments, which were very valuable and helpful in improving the quality of our manuscript. We have studied the comments carefully and have made corresponding revisions. Our responses to your comments are listed as follows. The comments are shown in blue, and our responses are shown in black.

Sincerely,
Qiaoying Lin, Ph.D.

Department of Resources and Environmental Sciences, Quanzhou Normal University, Donghai Street 398, Quanzhou, Fujian 362000, China

Reviewer #2:
General comments:

The revised manuscript is greatly improved, especially the introduction section. Most of the comments have been well addressed. However, I still have some specific comments.
The basic idea of this manuscript is to alleviate the development burden of hydrological modelers to achieve high-performance watershed modeling without reconstruction of model code, which is novel and clearly stated. The implementation based on the SWAT model, i.e., GP-SWAT, must be helpful for the scientific community. Overall, I am glad to suggest an acceptance for publication after a minor revision.

Reply: Thank you for your constructive comments. We have addressed all of your comments in the revised manuscript.

Specific comments:
1. In Line 53-54, the author introduced three types of parallelization strategies, such as model-level, submodel-level, and spatial-decomposition. But, in my view, the author has confused the spatial-decomposition method with the submodel-level, i.e., Line 64-79 should be the spatial-decomposition method, or more precisely, the spatial(-temporal) decomposition method, and Line 80-90 should be the submodel-level method. I mean, the so-called submodel level is a special case derived from the spatial(-temporal) decomposition method. In such a case, each submodel is a full model executed on one part of the watershed (i.e., subbasin). Besides, each parallelization type should have a short and precise definition. Please consider my suggestion.

Reply: Thank you very much for your suggestion. We have revised the introduction section thoroughly to avoid possible confusion regarding the parallelization strategies introduced. In the introduction section, we briefly review the broad model parallelization method and then narrow it to a special model parallelization method, i.e., the spatial-decomposition method. When reviewing the spatial-decomposition method, we first introduced spatial-decomposition implemented through model reconstruction, which is a possible limitation. We then described another spatial-decomposition method

performed without model reconstruction, which has advantages (reducing the modeler's workload and providing an alternative to cooperate with recent advanced information technologies and resources). We finally outlined the goal and scheme of the research presented in this paper. The introduction section is written as follows:

"With the enhanced availability of high-resolution remote sensing data and long periods of hydrometeorological data, hydrologists are increasingly building high-fidelity hydrological models to investigate water availability (Liang et al., 2020), water quality (Fang et al., 2020), climate change (Cai et al., 2016), and watershed management options (Jayakody et al., 2014;Qi and Altinakar, 2011;Lee et al., 2010). However, these hydrological models, which contain detailed representations of real-world systems and processes, can demand large computational budgets and require prohibitively high execution times, ranging from minutes to days (Razavi et al., 2010). Because modeling practices such as model calibration and uncertainty analysis usually involve thousands of model evaluations or more, they may sometimes become computationally prohibitive or even infeasible (Razavi and Tolson, 2013). Thus, the effective use of computationally expensive simulations remains a challenge for many applications involving a large number of model simulations.

In general, there are four broad types of research methods for alleviating the computational burden associated with computationally expensive model applications: (1) utilizing metamodeling approaches (Chandra et al., 2020;Sun et al., 2015), (2) developing computationally efficient algorithms (Humphrey et al., 2012;Joseph and Guillaume, 2013), (3) opportunistically avoiding model evaluations (Razavi et al., 2010), and (4) utilizing parallel computing technologies and infrastructures (Yang et al., 2020;Huang et al., 2019;Wu et al., 2013;Wu et al., 2014;Zamani et al., 2020). The first, second, and third ideas above share the same goal of reducing computational demand by using lightweight surrogate models, by decreasing the number of model simulations, and by terminating model execution early when the simulation result is poorer than expected, respectively. The fourth idea adopts a different strategy of boosting model application performance by optimizing the efficiency of computational resource utilization.

Among these methods, model parallelization is the most frequently adopted. It has been extensively applied to optimize the efficiency of generic modeling activities, such as model calibration (Zhang et al., 2013;Ercan et al., 2014;Gorgan et al., 2012), sensitivity analysis (Khalid et al., 2016;Hu et al., 2015), uncertainty analysis (Zhang et al., 2016;Wu and Liu, 2012;Zamani et al., 2020) and the identification of beneficial management practices (Liu et al., 2013). For spatially explicit models, it is possible to decompose a large-scale model into multiple smaller models and, in parallel, to simulate the independent smaller models to further improve model performance (hereafter, for simplicity, this spatial-decomposition and simulation method is referred to as the spatial-decomposition method). Though our literature review, we found that the spatial-decomposition method is usually performed through model reconstruction. For example, Wu et al. (2013) improved the performance of the Soil and Water Assessment Tool (SWAT) model by distributing subbasin simulations to different computational cores through the message passing interface (MPI). Wang et al. (2013) developed the temporal-spatial discretization method (TSDM), in which the parallelization degree of subbasins is exploited to the maximum extent by properly organizing the simulation sequences of dependent subbasins. To boost the performance of the fully sequential dependent hydrological model (FSDHM), Liu et al. (2016) adopted the MPI to perform subbasin-level model parallelization in a computer cluster. Based on the MPI and OpenMP frameworks, Zhu et al. (2019) introduced the spatially explicit integrated modeling system

(SEIMS) to perform model parallelization in a computer cluster consisting of multiple nodes. However, this parallelization method is relatively complex, as it requires a throughout model reconstruction to enable the parallel simulation of model components, to perform the communication among components that is necessary for integrating the model results, and to deal with issues such as failover and load balance. As a result, a steep learning curve is expected for modelers who are unfamiliar with the model source codes. Although there are some parallel computation frameworks available that can facilitate this method, e.g., the Open MPI and the OpenMP application programming interface (API), it is still a very tedious and time-consuming process.

For acyclic models, it is possible to perform model decomposition without model reconstruction. Taking the SWAT model as an example, a large-scale watershed model involving multiple subbasins can be split into multiple smaller models, each of which consists of only one subbasin (hereafter referred to as subbasin models). The stream flow and chemical loadings from upstream subbasins can be treated as boundary conditions of their downstream subbasins, which can be incorporated as point sources for these downstream subbasins. Through proper organization of the simulation of these models, a result identical to that of the original model can be obtained. In this strategy, upstream subbasin models must be simulated before downstream subbasin models; however, sibling subbasin models can be simulated in parallel to optimize model performance (for detailed information about the implementation of subbasin-level parallelization for the SWAT model, readers should refer to Yalew et al. (2013) and Lin and Zhang (2021)). Therefore, it can gracefully avoid much of the workload (e.g., failover, task management and balancing) that modelers face when consulting the model reconstruction method. Additionally, this aspect results in great opportunities to cooperate with recent advanced information technologies (ITs) and resources. For example, it is possible to loosely couple this spatial-decomposition method and parallel frameworks for processing big data (such as Hadoop, Spark and Flink) to perform model parallelization and alleviate the burden placed on modelers to address low-level programming tasks such as failover as well as task management and balancing. It also provides an economic alternative to deploy these solutions on cloud-based facilities, such as Azure HDInsight, Amazon Web Services Elastic Map Reduce (EMR), and Google Dataproc. Because these parallel frameworks are so universal in the IT industry, many cloud providers currently offer a convenient, on-demand support environment for these frameworks.

In this study, we propose a two-level (watershed- and subbasin-level) model parallelization scheme based on a combination of the graph-parallel Pregel algorithm and model spatial domain decomposition. The objective of this study is to create a simulation-accelerated tool for the SWAT model by adopting both watershed-level and subbasin-level parallelization, without model reconstruction. We hope that this tool will help IT practitioners or modelers improve model performance without requiring specific domain knowledge of the hydrological model. In accordance with this scheme and goal, a graph-parallel simulation tool for SWAT (named GP-SWAT) has been developed using an open-source general-purpose distributed cluster computing framework, Spark. GP-SWAT has been assessed in two sets of experiments to demonstrate its potential to accelerate single and iterative model simulations at different parallelization granularities when implemented on a computer running the Windows operating system (OS) and on a Spark cluster consisting of five computational nodes. Experiment set one was conducted to illustrate that GP-SWAT can be used to perform subbasin-level model parallelization using a multicore computer running the Windows OS, while in experiment set two, GP-SWAT was assessed for iterative model runs. For each experiment in the latter set, subbasin- and watershed-level parallelization schemes were employed to execute 1000 model

simulations with one to five parallel tasks implemented on each computational node. In each of the test cases, GP-SWAT was evaluated based on four synthetic hydrological models representing different input/output (I/O) burdens."

2. The title used "a…simulation framework", but the introduction only listed some parallelization strategies (or named parallelization schemes). I would suggest introducing existing hydrological modeling frameworks based on parallel computing and raise their weakness. I think that will be the answer to the second comment of #referee 1 (Line 95: It's better to state why this research wants to propose a new parallelization scheme?). Also, in the main text, the author used "a two-level parallelization scheme", why not "framework", and what is the difference?

Reply: Thank you for your valuable suggestion. We have unified these phases to eliminate any possible confusion. The proposed method that is able to perform two-level model parallelization in this study is referred to as the scheme, and the software we developed accordingly is referred to as the two-level model parallelization tool. The introduction section is concisely revised to focus on the spatial-decomposition method. When the limitations of spatial-decomposition through model reconstruction are outlined, the advantages and disadvantages of using existing hydrological modeling frameworks such as the Open MPI and the OpenMP application programming interface (API) are briefly introduced as follows:

"However, this parallelization method is relatively complex, as it requires a throughout model reconstruction to enable the parallel simulation of model components, to perform the communication among components that is necessary for integrating the model results, and to deal with issues such as failover and load balance. As a result, a steep learning curve is expected for modelers who are unfamiliar with the model source codes. Although there are some parallel computation frameworks available that can facilitate this method, e.g., the Open MPI and the OpenMP application programming interface (API), it is still a very tedious and time-consuming process."

3. The authors claimed that "indeed, the actual speedup ratio that can be achieved is largely dependent on the structure of the stream network." and "The intention of using two study areas in this study was to demonstrate how stream network complexities can affect GP-SWAT performance". Although the revised manuscript added some more descriptions of the two study areas, I cannot find the quantitative or qualitative analysis of the different stream networks' structures and the consequent result differences. So, I may suggest only retain the Jinjiang study area. Or, if the author can give a calculation method of theoretical speedup ratio considering the structure of stream networks and the available computing resources, that will be much valuable to adopt the two distinct study areas.

Reply: We agree. Only the Jinjiang study area is retained in the revised manuscript. Statements pertaining to the Harp Lake catchment and stream network structure have been removed throughout the manuscript. We are planning a new study to investigate how the structure of the stream network and the organization of the directed acyclic graph can affect the performance of GP-SWAT. We hope this issue can be well addressed in our future research. The calculation of the theoretical speedup is included and is defined as follows:

$$Speed_{ref} = Sub_{num} / \sum_{i=1}^{n} Ceil(Count_i / PT_{num}), \quad (2)$$

where sub_{num} is the number of subbasins of the hydrological model under test, n is the total number of supersteps, i denotes the i-th superstep, $Count_i$ is the number of subbasin models simulated in the i-th superstep, PT_{num} is the number of parallel tasks performed at an executor, and *Ceil* is a function that returns the smallest integer value that is greater than or equal to a predetermined parameter value.