

Calibration of SWAT Hydrological Models in a Distributed Environment Using the gSWAT Application

Victor Bacu, Danut Mihon, Teodor Stefanut, Denisa Rodila, Dorian Gorgan

Computer Science Department
Technical University of Cluj-Napoca
Cluj-Napoca, Romania

{victor.bacu, vasile.mihon, teodor.stefanut, denisa.rodila, dorian.gorgan}@cs.utcluj.ro

Abstract—Topics such as the sustainability and vulnerability of land management practices on water quality and quantity are very important in these days both for decision makers and for citizens. The enviroGRIDS FP7 project addresses some of these topics in the Black Sea Catchment area. One of the software tools developed in this project is gSWAT. It allows the calibration of SWAT hydrological models in a flexible development environment and uses distributed computational infrastructures to speed-up the simulations. The development of SWAT (Soil Water Assessment Tool) hydrological models is a well-known procedure for the hydrological specialists and this paper highlights, from the end-users point of view, the scenarios related with the calibration procedures available in the gSWAT application.

I. INTRODUCTION

Currently a lot of effort is put into topics such as sustainability and vulnerability of land management practices on water quality and quantity. Both decision makers and citizens are interested in these aspects. The enviroGRIDS [1] project, funded by the European Commission (EC) through 7th Framework Programme (FP7) aimed at building capacity in the Black Sea region providing specialists, decision makers and citizens with tools and applications specialized in processing spatial data, processing and visualization of satellite images, calibration and simulation of hydrologic models, etc.

One of the software applications developed in this project is gSWAT, targeting the calibration of SWAT [2] hydrologic models. In the frame of the project a very complex SWAT model of the Black Sea catchment basin has been developed, which required a complex calibration process. For small and medium scale models the calibration process can be easily performed on a desktop computer and in a reasonable amount of time. But for complex models this process is very difficult to be made in this way, mainly because of the size of the model (and the space that is needed in order to store all the results) and in executing all the required simulations in a reasonable amount of time. The gSWAT application addresses this issues and allows a flexible calibration process of complex (but not only) SWAT hydrologic models in a Web based environment. The user has access to high power computational resources and storage space. The execution of simulations is performed in a distributed environment, Grid.

A distributed infrastructure offers high power computation and storage resources, but the access to them is difficult for many users mainly because the interaction with this kind of infrastructure is not made in a graphical manner. For this reason the gSWAT application is developed as a Web

application to allow users to access and use the computational resources provided by the Grid infrastructure in the process of hydrologic model calibration. Management of processes, data distribution, task parallelization, monitoring, load balancing, authentication and authorization, scalability represents topics that are solved transparently by the gSWAT application from the user point of view.

In this paper we are presenting the scenarios related with the calibration process available in the gSWAT application. In section 2 are presented notions related to hydrologic models, calibration process, and execution. Several working sessions are presented in section 3. Section 4 presents the architecture and the module of the gSWAT application. Section 5 presents the interoperability aspect of the application by using services and the way in which this is implemented on some particular case.. The performance evaluation is discussed in section 6, section 7 presenting the conclusions.

II. HYDROLOGIC MODELS

Hydrological models are widely used for water resource planning, flood prediction, water quality, etc. They represent, in a simplified manner, the hydrological cycle which can be used for hydrological prediction. Three phases are required in order to provide a good hydrological model: development, calibration and evaluation. Model calibration aims at selecting the best values for model parameters so that the real hydrological behavior can be simulated [3]. Most hydrological models have two types of model parameters, namely physical parameters (represents physical properties of the catchment, which can be measured) and process parameters (represents characteristics which cannot be measured). The objective function measures the difference between the simulated output of the hydrological model and the measured output and in general is based on least squares or maximum likelihood methods.

A classification of hydrological models based on their model structure, spatial distribution, stochasticity, and spatial-temporal application is presented in [4]. Metric models such as Data Based Mechanic (DBM) [5] and Artificial Neural Networks (ANN) [6] are based on observations. ANN uses measured rainfall and runoff data to map the behavior of the rainfall-runoff processes. Physic-based models are using the equations of motion in order to represent hydrological processes. The hybrid physically-based-conceptual models aim at simplifying the model structure.

In the enviroGRIDS project the Soil and Water Assessment

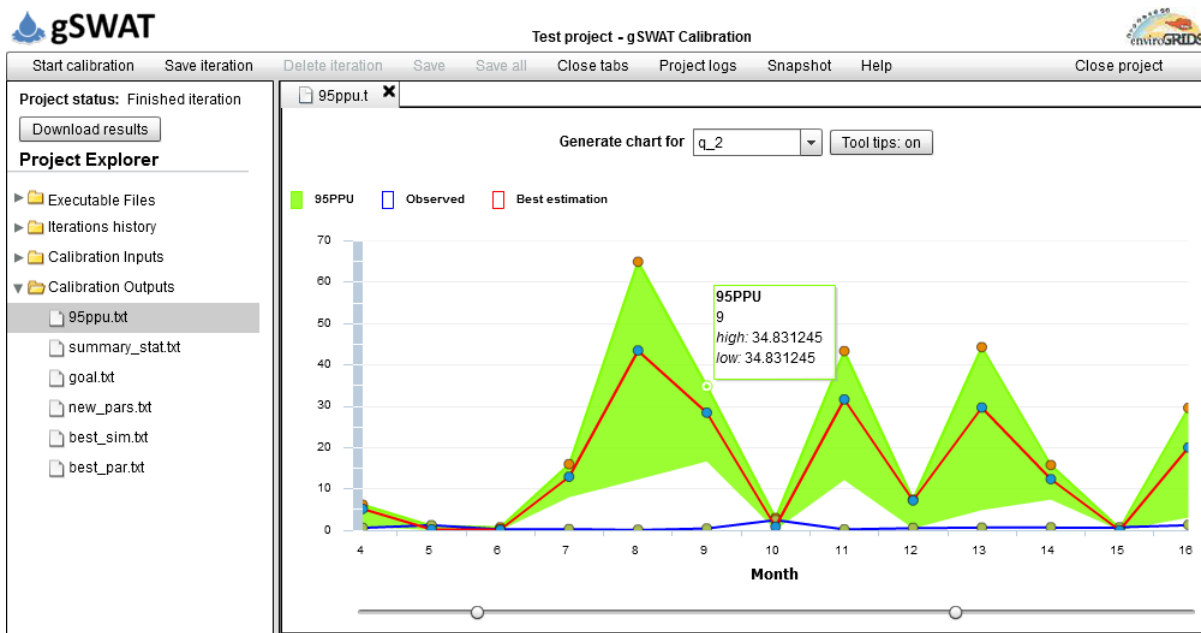


Fig. 1. Output visualization in gSWAT.

Tool (SWAT) has been used to model and simulate the Black Sea catchment basin. SWAT is a continuous simulation model that operates on a daily time step and quantifies the impact of land management practices on water quality and vegetation growth. The calibration and uncertainty analysis is a very important step in the flow of creating a SWAT model. In the enviroGRIDS project the Sequential Uncertainty Fitting program SUFI-2 [7] was used. One advantage of using this algorithm is that the simulations are independent one from another, meaning that we can achieve a high level of parallelism. It allows analyzing a large number of parameters which can be specified by the users.

A recent research paper [12] showed that the Grid technology is suitable for hydrology domain mainly for reducing the processing time. Different studies, such as in [13] and [14] prove that a Grid infrastructure, using efficient planning mechanisms, can lead to an increase of system performances. In [15] the authors present a parallelization framework for hydrological models calibration, but at a reduced scale, using a 24 CPUs cluster. A method involving Message-Passing Interface (MPI) is presented in [16]. A comparative analysis of three method of parallelization of 2D hydraulic models is presented in [17]. The usage of GPUs for processing a 2D flood simulation model is presented in [18] and [19]. Other methods of parallelization are described in [20] and [21].

III. WORKING SESSIONS

A. Projects

gSWAT is a Web based application supporting the calibration of complex SWAT hydrological models. It offers both computational resources to minimize the time needed to calibrate the models and storage resources to access remotely the SWAT models and also the results of the calibration process. This application is exposed to the users similar to the Software as a Service (SaaS) level from Cloud. The complexity of the

underlying computational infrastructure is hidden and the users can focus on the calibration process rather than aspects related to Grid computing.

A project in gSWAT represents a SWAT model together with other information related to it. The first step to create a new project is to define the project name and description. After this step the user specifies the SWAT model that will be uploaded to the gSWAT server. At the server side the SWAT model is remapped to the structure needed for the calibration process, meaning a new directory structure. The new structure is after that archived and stored on the Storage Element (SE), the LFN (Logical File Name) for the archive being updated in the database. A feedback with the status of this process is provided to the user.

A calibrated model is obtained after a set of iteration steps, each iteration step consisting in executing a variable number of simulations. For each calibration project only one iteration step is the active one, meaning that the user can start only one execution at a time for a calibration process. When starting a new iteration process the user has the possibility to save the previous one, and has access to all iterations that are already executed.

B. Process execution and monitoring

Only one iteration step can be active (in execution) for each calibration project at a time. From the user's point of view the complexity of the calibration process execution over the Grid infrastructure is transparent. From the graphical interface the user selects the start calibration button which will trigger the execution of the steps already detailed in a previous section. Before starting the execution the user should modify all the input parameters that will have an impact on the results. The gSWAT database is periodically interrogated in order to provide users with feedback about the execution (in terms of total execution time and number of completed simulations).

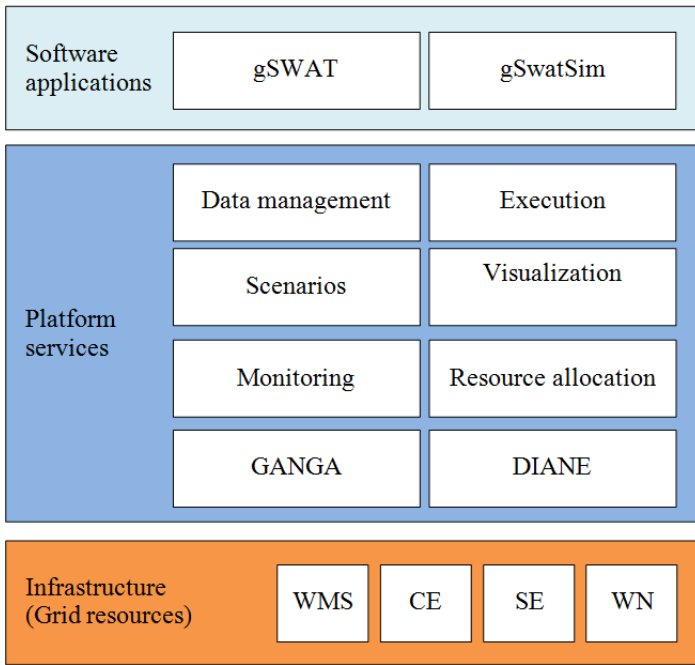


Fig. 2. gSWAT architectural modules.

The user has the possibility to stop the execution of the current iteration step by clicking the stop calibration button. This will trigger the cancelling of all the Grid jobs and cleaning of the current iteration step intermediary files.

C. Input and output data visualization

After the internal structure of the SWAT model is created the user has the possibility to explore it by using the graphical user interface. The text files can be edited directly in the text editor which supports opening multiple files at the same time, and basic operations such as save file, save files, redo, undo, copy and paste, etc.

The output results can be visualized as text or as charts. The chart module parses the 95ppu.txt file and output this data in a graphical manner. The chart presents the best estimated parameters values together with the observed values (see Figure 1). The user has the possibility to adjust the horizontal axis which represents the temporal scale. All the output data can be downloaded, as an archive, by the user. This archive is created on the fly when the user tries to download it.

IV. GSWAT APPLICATION

The gSWAT application [22], [23] is based on the client-server architectural model and uses Web 2.0 technologies in order to provide a flexible calibration interface for different categories of users, such as hydrology specialists or students.

By exposing an intuitive graphical interface, the gSWAT application overcomes the command line based interface exposed by gLite [8]. GANGA [9] offers a flexible programming interface and facilitates the accessibility to Grid infrastructures. DIANE (Distributes ANalysis Environment) [10] provides an efficient usage of Grid infrastructures and it is based on the master-slave paradigm. The gSWAT application is using both

GANGA and DIANE to provide a flexible environment and to minimize the execution time.

A. General architectures

The architecture is composed of three layers, where each layer provides different functionalities (presented in Figure 2). The distributed infrastructure that is used to minimize the calibration time is the Grid infrastructure. The services layer offers services both for the graphical user interface and for other applications that are interconnected with it, such as BASHYT. The graphical user interface is built in Adobe Flex and being a web based interface it can be used from different devices (such as desktops, laptops or even tablets). The layers are similar to the ones in Cloud computing, the infrastructure level can be mapped to the Infrastructure as a Service (IaaS), the Platform services can be mapped to the Platform as a Service (PaaS) and the software applications can be mapped to the Software as a Service (SaaS). An experimental study of migration of scientific applications (where the experiment was made on the gSWAT application) from Grid to Cloud Cluster infrastructure was presented in [11].

B. gSWAT Modules

1) *Data management*: In gLite a Storage Element (SE) offers a uniform access to various data storage resources (such as disk or tape) and allows users and applications to store/retrieve data in a very simple manner. From the users point of view the file location is hidden, he has access to files based on a logical file name. The data could be replicated to several SEs in order to minimize the transfer cost or to

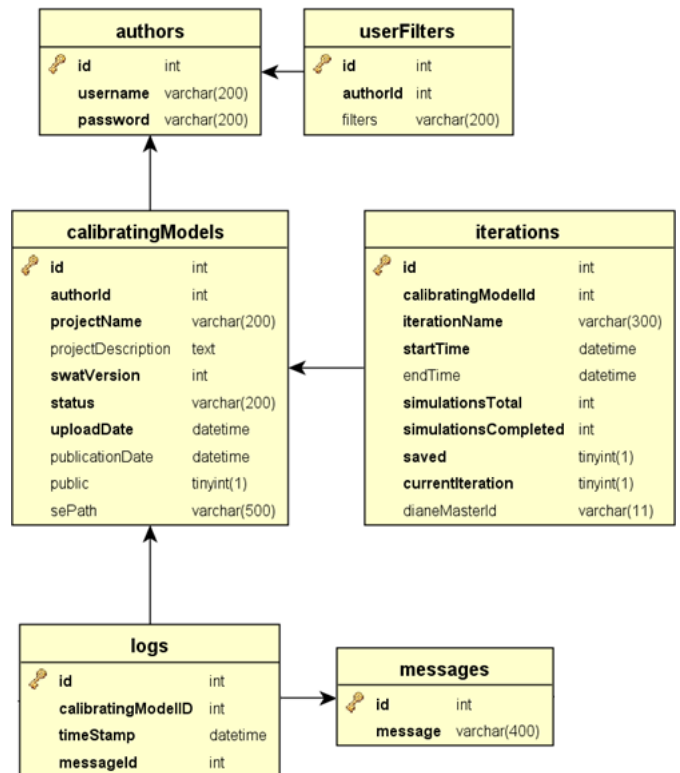


Fig. 3. Database structure.

increase the availability of data. Files are shared by the users in a Virtual Organization (VO) and are protected by security mechanisms. In such environment the files (data) are written once and they cannot be modified, the only solution for doing this is by removing and replacing the files. The protocol that is used by the SEs is GSIFTP which offers a high-speed, reliable and secure data transfer.

This module is responsible mainly for exchanging data to and from the Grids Storage Element. It offers services related to this functionality which provides a transparent access to data resources for the users. A specific data structure is needed by the calibration process and this module creates the necessary directory structure and store the SWAT model to the SE. Another service provides the results from the execution of calibration.

Figure 3 presents the database that stores all the information related to projects, iterations, etc. In gSWAT, each hydrological model is represented in the database as a calibrating model. The most important information about calibrating model are the SWAT version (is used to know which executable is needed in order to execute the model simulations), the logical file path (is used to retrieve the SWAT model from the SE after each job is started on a WN), status (is modified by the execution module to update the state of the calibration process and is used by the graphical user interface to inform the users about the current state).

The status of the calibrating model could be one of the followings:

- 1) Empty the project doesn't have a valid SWAT model attached to it;
- 2) Uploading the SWAT model archive is fetched to the gSWAT server, validated and transformed to the structure needed by the calibration process and finally uploaded to the SE;
- 3) Incomplete uploading the SWAT model is not valid, or another problem occurred when storing the model on the Grid repository (missing Grid certificates, problem in communicating with LFC server, transfer error, etc.);
- 4) Loaded the project contains a valid SWAT model stored on the SE and on which the calibration process can start;
- 5) Finished - the current iteration execution is completed and the model can be used to define and execute scenarios;
- 6) Running a iteration execution is currently ongoing;
- 7) Incomplete iteration some errors occurred during the execution (bad SWAT model, missing files, etc.).

For each calibrating model there can be zero to many iterations steps, but only one is currently active (meaning is in running). The users have the option to visualize all the input and output data related to one iteration. The start and finish time of the execution is stored in the database, the execution time for each individual simulations can be retrieved from the output files. The number of simulations that are completed is updated by the monitoring module and is reflected in the graphical user interface.

In a dynamic environment, such as Grid, errors can occur at different level, data or execution. In order to minimize the

possible errors due to data the data management module tries to detect and recover the execution.

2) *Execution*: In order to validate a SWAT model a complex calibration process is being conducted, this process being completed when a calibration criteria is satisfied. By performing a variable number of iteration steps we try to accomplish this goal. In each iteration step several simulations of the SWAT model are executed (independently on the other ones) by performing 3 phases (presented in Figure 4): pre-processing, actual execution and post-processing.

Because the complexity of the pre-processing phase is not very high this phase is performed at the server side, once for each iteration step. The user has the possibility to modify some parameters of the SWAT model by defining intervals from which, by using the Latin hypercube sampling method, new parameters values are generated. The outcome of this phase is a list of new parameters values (one list for each simulation needed) which will be propagated in the next step in the SWAT model.

The most complex step is the actual execution of simulations. The execution module uses DIANE and GANGA to interact with Grid jobs. DIANE is used to start and manage the execution of simulations. Each simulation is mapped in DIANE as a task which will be executed on a Grid WN. GANGA starts the Grid jobs and connects to DIANE master

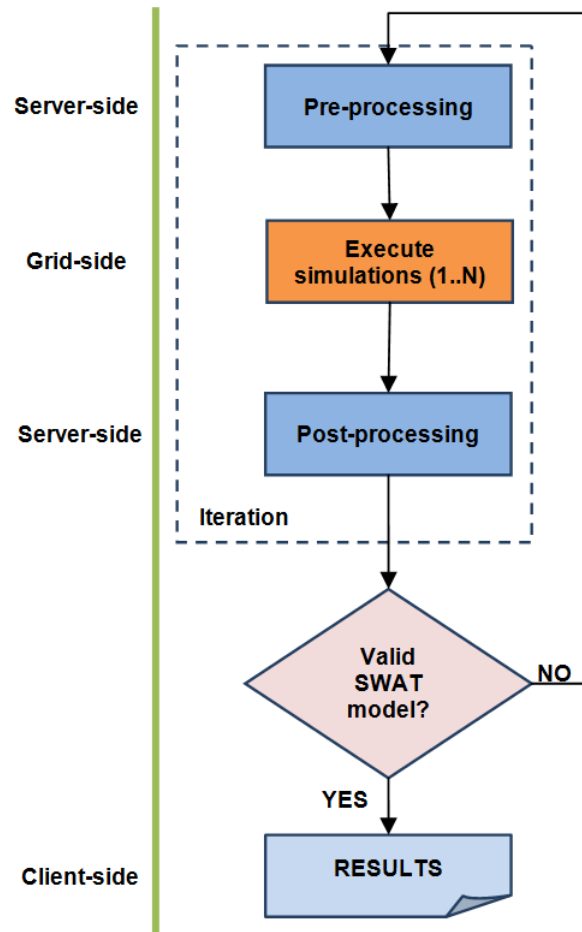


Fig. 4. Calibration steps.

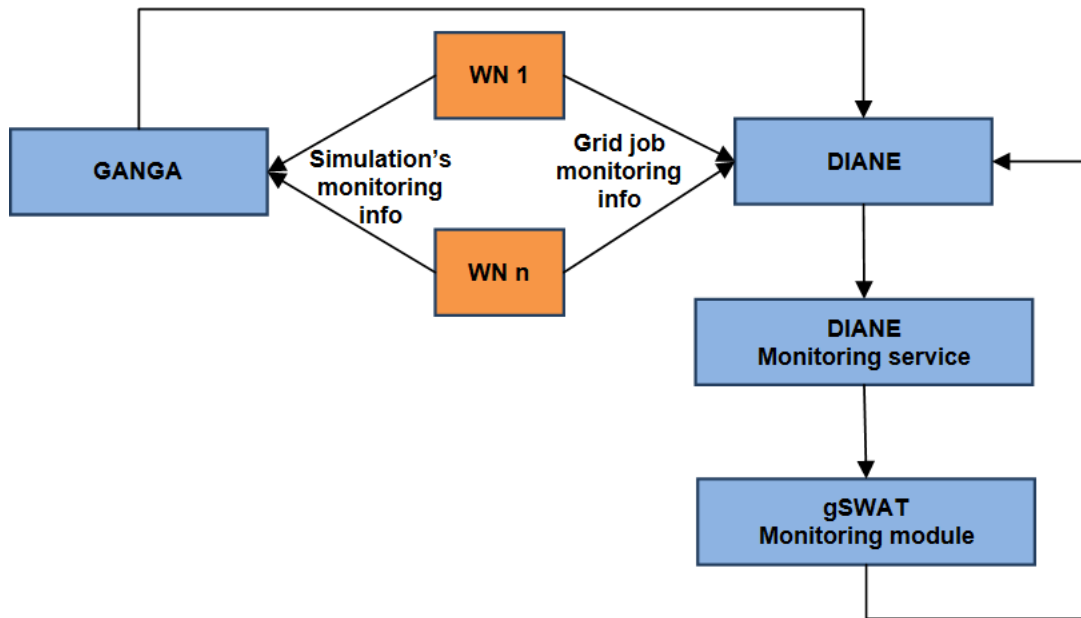


Fig. 5. Monitoring module flow of actions.

in order to receive tasks to be executed. Because the number of simulations that should be performed is very high (varying in general from 200 to 1000) and in order to minimize the number of Grid resources that are used each Grid job will execute one or more simulations.

The following steps are performed by the execution module:

- Define the script that will be executed by the Grid job. This script will copy locally the SWAT model archive (stored on the SE), extract the files, modify the model parameters accordingly to the new values generated in the pre-processing phase, execute the SWAT simulation and in the end archive and send back the SWAT outputs;
- Define the DIANE script that maps the simulations to be performed to tasks;
- Start a new DIANE master for each iteration step on a different port to which the Grid jobs (DIANE workers) will connect;

- Start Grid jobs using GANGA. After this each job will connect to the DIANE master and will receive tasks (simulations) to be executed;
- Monitor the execution of the simulations and store this info to the gSWAT database from which the graphical user interface will provide feedback to the users;
- Download the output results from each simulation at the server side.

The final phase is the post-processing which is also executed at the server side and creates the output for the current iteration step based on the output results provided by each simulation. In the graphical user interface the users have the possibility to visualize, in a graphical manner, the results or to download the files.

3) *Scenarios*: Scenarios can be defined starting from a calibrated SWAT model to highlight different aspects regarding the modeled catchment basin. The gSWATSim module allows the execution of basic scenarios which are created by modifying some of the model parameters. Similar to the execution module, it uses the Grid infrastructure to run the scenarios. It offers a complex execution and management solution and also the possibility to integrate some of the functionalities in other applications.

The database related to scenarios stores information such as: scenario name, scenario description, scenario fingerprint, SWAT version, status, scenario location on the Storage Element and scenario execution output location on the Storage Element. Scenario execution means the execution of only one simulation. The output could be fetched to other applications, in order to visualize to results in a graphical manner. The following steps are performed by this module in order to run scenarios:

- Start the DIANE master;

Log messages - Demo SWAT model	
Messages	Time stamp
Finish the iteration process	Fri Oct 19 12:12:34 GMT+03
Start the iteration process	Fri Oct 19 12:01:55 GMT+03
Loaded project. You successfully upload the TxtInOut folder	Fri Oct 19 11:59:48 GMT+03
Processing files (copy model to Grid)	Fri Oct 19 11:58:02 GMT+03
Uploading the model is finished	Fri Oct 19 11:58:02 GMT+03
Start uploading model. Please wait ...	Fri Oct 19 11:57:59 GMT+03
Empty project. Please upload a TxtInOut folder	Fri Oct 19 11:56:10 GMT+03

CLOSE

Fig. 6. Detailed log messages.

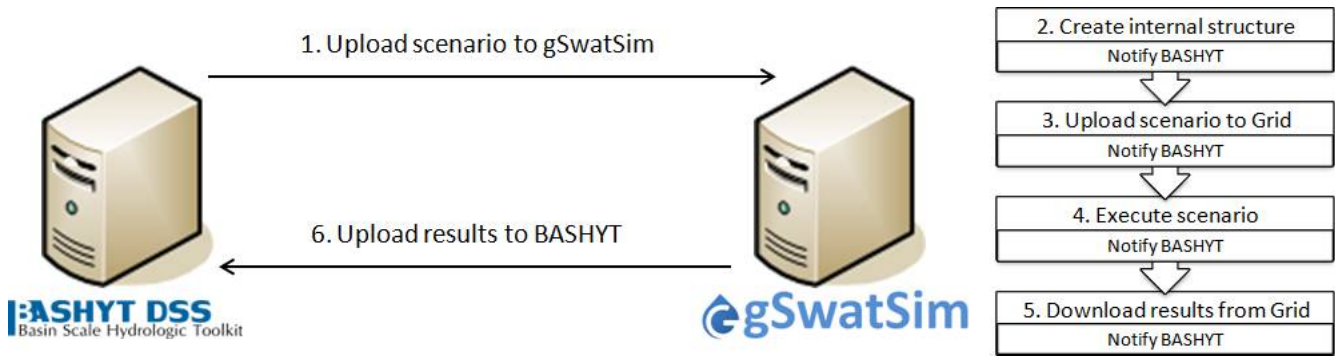


Fig. 7. gSwatSim and BASHYT service based interoperability.

- Start one GANGA worker;
- Execute the job on the Worker Node (WN) (meaning copy locally the SWAT scenario archive from the SE, execute SWAT simulation, archive the output results and upload them to SE).

4) *Monitoring*: The monitoring module is used both to check the execution flow and to provide feedback to users. Each calibration project has attached a status field (in the database) that characterizes the current situation. This status is updated based on commands initiated by users (uploading a new project, change the project information, etc.) or based on the execution of the iterations. Every job that is executed in the Grid environment has one of the following states: submitted, waiting, ready, scheduled, running, done, cleared. These states are reached with the successful execution; other states are reached with the failure of execution. The status information that is provided by the gSWAT application is different that the states reachable by Grid jobs. These states were presented in a previous section.

The DIANE monitoring system is used by the gSWAT application to gather information regarding the execution flow (meaning the number of simulations that were successfully executed). The flow of interaction between the different components of the system is presented in Figure 5. The DIANE master connects to the monitoring server (which is gridmsg101.cern.ch) and the monitoring messages are sent automatically to it. The DIANE master updates the status based on the information received from the Grid WNs. Two levels of execution status is available, one from DIANE which is responsible for providing info at a higher level, (meaning at simulation level) and the other one from GANGA which is used to monitor the Grid jobs and provide info at a lower level (which is important mainly to recover the execution if some error occurred). The Diane Dashboard application makes available all the monitoring messages in JSON format. The JSON format is used to transfer structured data between a server and a web application. The monitoring module has incorporated a JSON parser that update the gSWAT database with relevant information, such as start time, end time (if it is available), total simulations, completed simulations. At predefined time interval, the information is updated also in the graphical user interface.

The status of the calibration projects offers only limited information about it. Beside this, the user has access to more

detailed information about the progress of the calibration process in the form of system logs. Every time a calibration project changes his status, much more detailed message info is stored in the database together with a timestamp (used to be able to order the messages). The user can visualize system logs related to a single calibration project or for all of his calibration projects (Figure 7).

5) *Resource allocation*: The calibration process for large scale SWAT models is quite complex (mainly because of the size of the model and the number of simulations that are needed to be performed) and in order to minimize the execution time and also to improve the usage of Grid resources the resource allocation module [25] selects the optimum number of resources that are needed. The model complexity is defined based on the number of files, model size and an estimated complexity provided by the specialist in hydrology. Other important aspects are the availability of the Grid resources (free WN) and also the number of users that are using the application. The steps followed by this module are the following: gathering requirements (specified complexity, number of files, model size, etc.), discovering Grid resources (available WNs, waiting jobs, etc.) and determining the necessary resources (based on the requirements and the available resources).

The actual execution time has the following mathematical expression:

$$TotalExecutionTime = T(resourceAllocation) + T(copyModel)$$

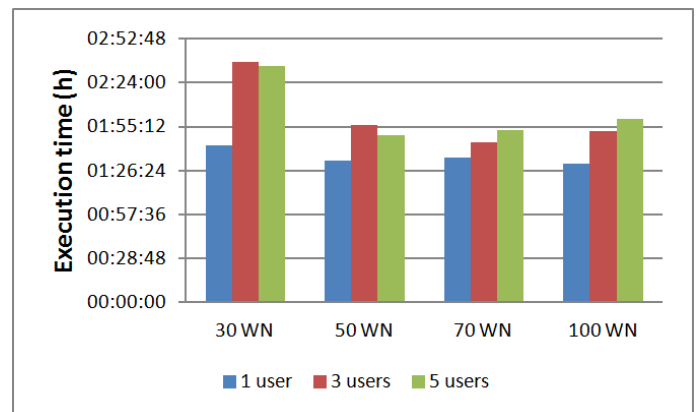


Fig. 8. The variation of execution time with the number of users.

$$+ T(\text{propagateModifications}) + T(\text{actualExecutionOfSWAT}) + T(\text{getOutputData})$$

V. gSWATSIM INTEROPERABILITY

The gSWATSim [26] exposes a collection of REST Web Services [27] that allows the user to create new projects (scenarios), to modify some information about the projects (such as project name, description, etc.), to run scenarios, to upload output results to BASHYT, etc.

BASHYT [28] offers in a Web based interface the possibility to produce reports for SWAT models in a flexible manner. The interoperability between gSWATSim and BASHYT brings some advantages:

- scenarios are developed in a flexible environment by using BASHYT functionalities;
- by using GRID capabilities, gSwatSim speeds up the processing (simulation) of large scenarios;
- the results can be visualized by using BASHYT dedicated tools and modules.

The interoperability between gSwatSim and BASHYT is presented in Figure 6. The first step is to upload scenario to gSwatSim. At server side the internal structure is created and BASHYT is notified about it. After that the scenario is archived and uploaded to SE from where it will be available. The next step is to execute the scenario and store the results to SE. The results are downloaded by gSwatSim from the Grid and uploaded to BASHYT. Notification messages are sent to BASHYT each time the status of the scenario execution changes. In the end the output results can be visualized in BASHYT.

VI. PERFORMANCE EVALUATION

A. Distributed infrastructure

By using a distributed infrastructure we gain computational power, efficient storage solution and flexibility. From the user point of view the access to the distributed infrastructure is made automatically. The computational resources needed by the gSWAT application are provided by the enviroGRIDS project VO. Currently three CEs are providing resources for it but the main CE is ce01.mosigrid.utcluj.ro providing 128 physical CPUs, with a 1024 logical CPUs. This VO is using one SE (se01.mosigrid.utcluj.ro) with a storage capacity of 13 TB. Being a production site, and not just a test site, the availability of resources is not constant (the resources are shared with other VOs), this being reflected on the experiments that were made on it. A comparative analysis of parallel execution of SWAT hydrological model on multicore and Grid architectures is presented in [24].

B. Black Sea catchment basin calibration results

The gSWAT application is addressed to specialists in hydrology to help them to calibrate complex SWAT model. It can also be used as a teaching tool in workshops related to SWAT and calibration. The total area of the Black Sea Basin is around 2.3 million km² with rivers from 23 countries. A

complex SWAT model consists of a very high number of files (at least 1.000.000 files).

For the first experiments we have used a small scale model. The size of the SWAT model archive stored on SE is 256 MB and the size of the extracted archive is 327 MB. The number of input files, without the ones from the backup directory, is 17,990 files and the number of the hydrological sub-basins is 1,629. The number of input parameters for this model was 14. The variables for this experiments are the number of simulations (100, 500 and 1000 simulations), and the number of allocated WNs (30, 50, 80 and 100 WNs).

1) *gSWAT scalability with the number of user:* A first experiment targets the scalability of the application in terms of number of users that are performing calibrations. In Figure 8 is represented the influence of the number of users on the overall execution time of the calibration process. A first remark is that the calibration time when only one user is running the application is lower than when 3 or 5 users are also performing a calibration. This is obvious because only one user is using the Grid resources. It is also important to notice that even though the execution time increases with the number of users it is not a linear increasing. The overall execution time is higher mainly because the number of Grid resources is not scaled with the number of users and the Grid services have to manage more jobs. The number of Grid resources was fixed and the other VOs could use them as well, reducing in this way the number of possible computational resources for gSWAT. In all cases the overall execution time decreases when adding more computational resources even though more users are performing calibrations.

2) *gSWAT scalability with the number of computational resources:* Another experiment aims to show what is the influence of the number of computational resources used (WNs) on the overall execution time. When adding more resources the execution time should decrease. The improvement is not in all cases proportional with the additional computational resources that are used. In Figure 9 are presented the results. The execution time decreases when adding more resources, the decrease is accentuated better when the number of simulations is higher. The trend is the same even if the number of simulations is 100, 500 or 1000, proving in this way the scalability of the application with the number of simulations and with the number of computational resources. In some cases even though we add more resources the speedup is small and it shows that is not always a good idea to add more resources.

Table 1 presents the speedup (by percentage values) gained by increasing the computational resources from 30 to 50 WNs, from 50 to 80 WNs and from 80 to 100 WNs. If we increase the number of WNs from 30 to 50 we gain 44% for 100 simulations. However, the number of resources needed is with 67% more. The speedup gained by increasing the computational resources from 80 to 100 (and any number of simulations),

TABLE I. SPEEDUP PERCENTAGES

$S_n = T_1/T_n, n = \#WN$	100 Sims	500 Sims	1000 Sims
50 WNs / 30 WNs	44%	83%	40%
80 WNs / 50 WNs	14%	31%	57%
100 WNs / 80 WNs	3%	12%	13%

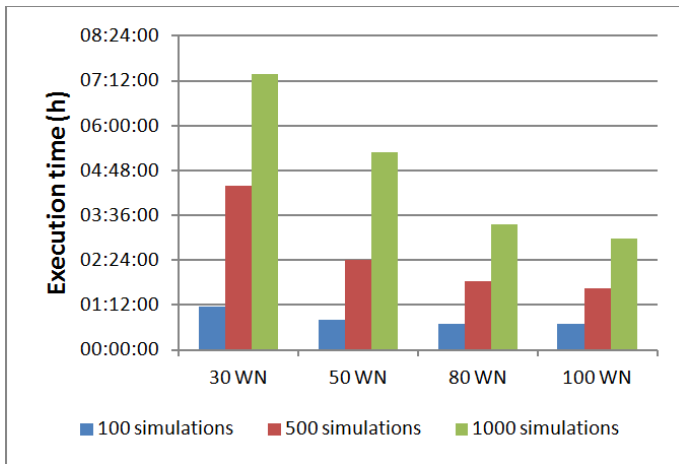


Fig. 9. Total execution time.

is not very high compared with the 25% additional resources that are needed. Figure 10 presents the execution time per simulation. In all cases (variation of the number of simulations) the execution time for one simulation decreases (keeping the same trend) when we use more computational resources.

Figure 11 presents the submission time, which is constant (around 13 seconds) and does not depend on the number of computational resources used or on the number of simulations that were executed. The submission process consists in all the steps performed by the gSWAT application before the execution of simulations can begin. Even though the submission time is constant the impact on the total execution time is different.

For the complex SWAT model we have executed 8 iteration steps, each iteration step requiring 200 simulations. Because of the complexity of the model we split the execution of each iteration step in 4 blocks of 50 simulations. The average execution time for one iteration step was around 170 hours, meaning a virtual execution time per simulation of around 50 minutes. The actual execution time for one simulation was around 40 hours. The increase of performance is in this case a significant one, execution of all the simulations on only one computer is impossible in this case in a reasonable amount of

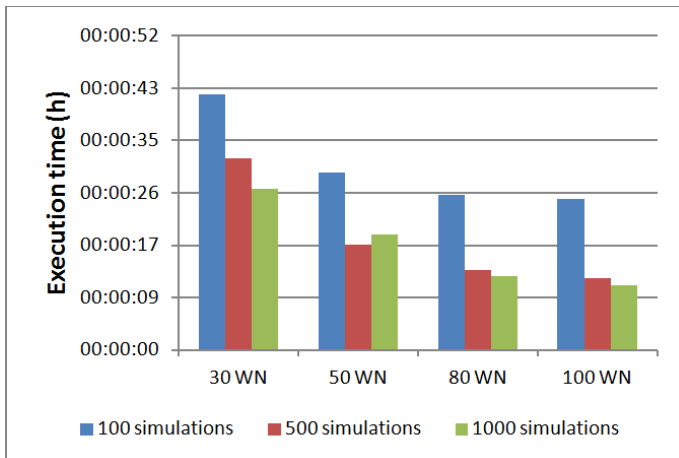


Fig. 10. Execution time per simulation.

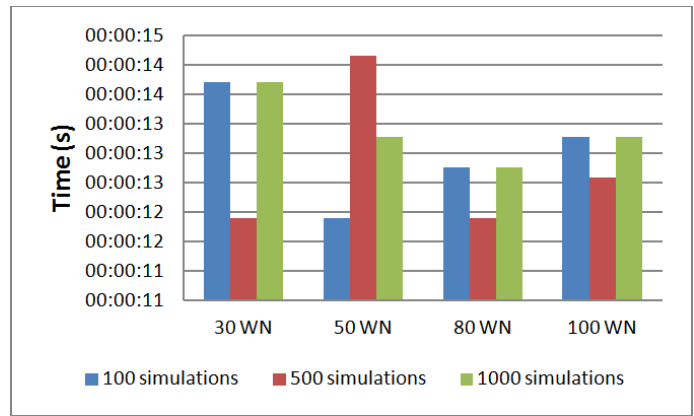


Fig. 11. Submission time for one job.

time. The execution times for each simulation are different but there are no significant differences regarding the total execution time (see Figure 12 where results from three iteration steps are presented). The minimum and maximum execution time for each simulation block varies mainly because of the availability of Grid resources.

For a complex model where the number of files is very high (more than 1.000.000 files) we have to start fewer jobs on the same physical machine. The execution of the simulation needs to read and write in this case many files, and if multiple jobs are executed on the same physical machine, they will make concurrently access to the hard-disk and the execution time will grow excessively. In some cases the execution of one or more simulations takes longer than the execution of the other ones (as is the case of the second simulation block for the second iteration steps presented in Figure 12). The availability of the Grid resources is the cause for this higher execution time but as can be seen the impact is not significant. This experiment proves that in this case (calibration of complex models) the Grid offers a very good solution, decreasing very much the time needed to execute all the simulations required by the calibration process.

VII. CONCLUSIONS

Complex SWAT hydrologic models are used to assess the sustainability and vulnerability of land management practices on water quality and quantity. The gSWAT application offers a flexible environment to calibrate SWAT models over distributed infrastructures such as Grid. The execution time could be minimized by running several simulations in parallel, on different WNs. In some cases (according to the number of simulations or the model complexity) the speedup obtained by increasing the number of computational resources is quite small. The experiments proved that the calibration process can benefit by the scalability offered by the Grid infrastructure.

ACKNOWLEDGMENT

This research is supported by the FP7 enviroGRIDS Project funded by the European Commission, through the Contract 226740.

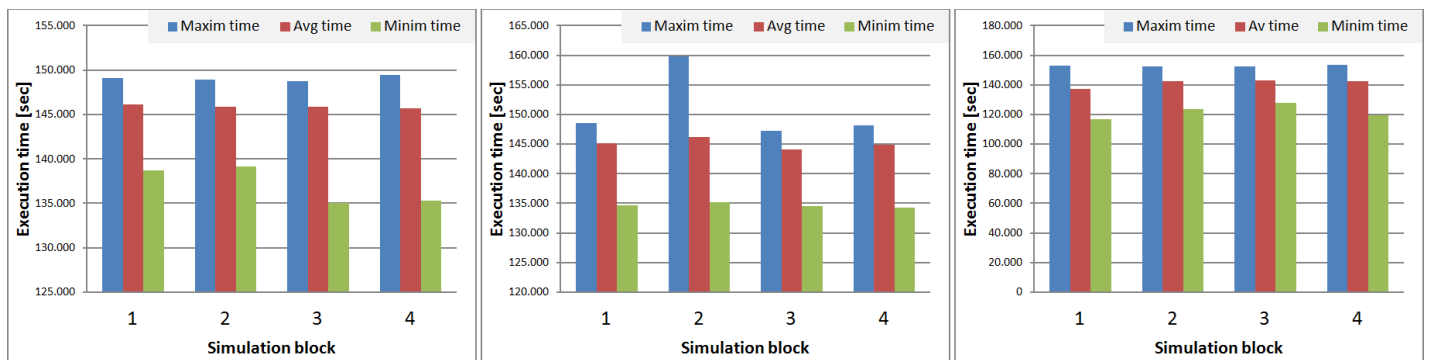


Fig. 12. Execution times for complex SWAT models (results for three iteration steps using 4 simulation blocks for each one).

REFERENCES

- [1] enviroGRIDS project, <http://www.envirogrids.net/>
- [2] SWAT: Soil and Water Assessment Tool, available at: <http://swatmodel.tamu.edu/>
- [3] C. Moore, and J. Doherty, "Role of the calibration process in reducing model predictive error", *Water Resour. Res.*, 41, 2005.
- [4] I. G. Pechlivanidis, B. Jackson, N. McIntyre and H.S. Wheater, "Catchment scale hydrological modelling: A review of model types, calibration approaches and uncertainty analysis methods in the context of recent developments in technology and applications", *Global NEST Journal*, 13(3), pp. 193-214, 2011.
- [5] P. C. Young, A. J. Jakeman and D. A. Post, "Recent advances in the data-based modelling and analysis of hydrological systems", *Water Science and Technology*, 36(1), pp. 99-116, 1997.
- [6] C. W. Dawson, R. J. Abrahart, A. Y. Shamseldin and R. L. Wilby, "Flood estimation at ungauged sites using artificial neural networks", *Journal of Hydrology*, 319(1-4), pp. 391-409, 2006.
- [7] K. C. Abbaspour, C. A. Johnson, and M. Th. van Genuchten, "Estimating Uncertain Flow and Transport Parameters Using a Sequential Uncertainty Fitting Procedure", *J. Vadose Zone*, 3 (4), pp. 1340-1352, 2004.
- [8] gLite, <http://glite.cern.ch/>
- [9] J.T. Moscicki, F. Brochu, J. Ebke, U. Egede, J. Elmsheuser, K. Harrison, R.W.L. Jones, H.C. Lee, D. Liko, A. Maier, A. Muraru, G.N. Patrick, K. Pajchel, W. Reece, B.H. Samsel, M.W. Slater, A. Soroko, C.L. Tan, D.C. van der Ster, M. Williams, "Ganga: A tool for computational task management and easy access to Grid resources", *Computer Physics Communications*, Volume 180, Issue 11, November 2009, pp. 2303-2316, ISSN 0010-4655, 2009.
- [10] J.T. Moscicki, DIANE - Distributed Analysis Environment for GRID enabled Simulation and Analysis of Physics Data, *Nuclear Science Symposium Conference Record*, pp. 1617 - 1620, Vol.3, ISSN 1082-3654, 2003.
- [11] L. Biro, V. Bacu, D. Rodila, L. Barabas, D. Gorgan, "Grid to cloud migration of scientific applications, using dynamically created cloud clusters", *Intelligent Computer Communication and Processing (ICCP)*, 2012 IEEE International Conference on , pp.335,340, 2012.
- [12] G. Lecca, M. Petitdidier, L. Hluchy, M. Ivanovic, N. Kussul, N. Ray, and V. Thieron, "Grid computing technology for hydrological applications", *Journal of Hydrology*, 403(1-2), pp. 186 - 199, 2011.
- [13] F. Pop, "A fault tolerant decentralized scheduling in large scale distributed systems", Chapter in: *Handbook of Research on P2P and Grid Systems for Service-Oriented Computing: Models, Methodologies, and Applications*, edited by: Antonopoulos, N., Exarchakos, G., Li, M., and Liotta, A., Information Science Reference (IGI Global), pp. 566-588, 2010.
- [14] B. Simion, C. Leordeanu, F. Pop, and V. Cristea, "A hybrid algorithm for scheduling workflow applications in grid environments (icpdp)", In *Proceedings of the 2007 OTM confederated international conference on On the move to meaningful internet systems: CoopIS, DOA, ODBASE, GADA, and IS - Volume Part II, OTM07*, pp. 1331-1348, Berlin, Heidelberg, 2007.
- [15] E. Rouholahnejad, K. Abbaspour, M. Vejdani, R. Srinivasan, R. Schulin, and A. Lehmann, "A parallelization framework for calibration of hydrological models", *Environ. Model. Softw.*, 31, pp. 2836, 2012.
- [16] T. Li, G. Wang, J. Chen, and H. Wang, "Dynamic parallelization of hydrological model simulations", *Environ. Model. Softw.*, 26(12), pp. 1736-1746, 2011.
- [17] J. Neal, T. Fawcett, P. Bates, and N. Wright, "A comparison of three parallelisation methods for 2d flood inundation models", *Environ. Model. Softw.*, 25(4), pp. 398-411, 2010.
- [18] A. J. Kalyanapu, S. Shankar, E. R. Paradyjak, D. R. Judi, and S. J. Burian, "Assessment of gpu computational enhancement to a 2d flood model", *Environ. Model. Softw.*, 26(8), pp. 1009-1016, 2011.
- [19] B. Singh, E. R. Paradyjak, A. Norgren, and P. Willemsen, "Accelerating urban fast response lagrangian dispersion simulations using inexpensive graphics processor parallelism", *Environ. Model. Softw.*, 26(6), pp. 739-750, 2011.
- [20] Y. Cui, B. E. Vieux, H. Neeman, and F. Moreda, "Parallelisation of a distributed hydrologic model", *Int. J. Comput. Appl. Technol.*, 22(1), pp. 42-52, 2005.
- [21] V. Sharma, D. A. Swayne, D. Lam, and W. Schertzer, "Parallel shuffled complex evolution algorithm for calibration of hydrological models", In *Proceedings of the 20th International Symposium on High-Performance Computing in an Advanced Collaborative Environment, HPCS 06*, pages 30, Washington, DC, USA, 2006.
- [22] D. Gorgan, V. Bacu, D. Mihon, T. Stefanut, D. Rodila, P. Cau, K. Abbaspour, G. Giuliani, N. Ray, A. Lehmann, "Software platform interoperability throughout enviroGRIDS portal", *Selected Topics in Applied Earth Observations and Remote Sensing (JSTARS)*, IEEE Journal of, 5(6), pp. 1617-1627, 2012.
- [23] D. Gorgan, V. Bacu, D. Mihon, D. Rodila, K. Abbaspour, E. Rouholahnejad, "Grid based calibration of SWAT hydrological models", *Natural Hazards and Earth System Science (NHES)*, 12(7), pp. 2411-2423, 2012.
- [24] D. Rodila, V. Bacu, D. Gorgan, "Comparative parallel execution of SWAT hydrological model on multicore and grid architectures", *International Journal of Web and Grid Services (JWGS)* 8, 3, pp. 304-320, 2012.
- [25] V. Bacu, D. Gorgan, "Grid application oriented computational resource allocation strategy", *High Performance Computing and Simulation (HPCS)*, 2012 International Conference on, pp.581-587, 2012.
- [26] V. Bacu, D. Mihon, T. Stefanut, D. Rodila, D. Gorgan, P. Cau, S. Manca, "Grid based services and tools for hydrological model processing and visualization", In *Symbolic and Numeric Algorithms for Scientific Computing (SYNAS)*, 2011 13th International Symposium on, pp. 291-298, 2011.
- [27] R. T. Fielding, "Architectural Styles and the Design of Network-Based Software Architectures", Ph.D. Dissertation, University of California, Irvine, 2000.
- [28] S. Manca, C. Soru, P. Cau, G. Meloni, M. Fiori, A multi model and multiscale, GIS oriented Web framework based on the SWAT model to face issues of water and soil resource vulnerability. Presentation at the 5th International SWAT Conference, August 3-7, 2009