International Environmental Modelling and Software Society (iEMSs)
2010 International Congress on Environmental Modelling and Software
Modelling for Environment's Sake, Fifth Biennial Meeting, Ottawa, Canada
David A. Swayne, Wanhong Yang, A. A. Voinov, A. Rizzoli, T. Filatova (Eds.)
http://www.iemss.org/iemss2010/index.php?n=Main.Proceedings

Parallel computing of a large scale spatially distributed model using the Soil and Water Assessment Tool (SWAT)

S. G.Yalew[a], A. van Griensven[a], L. Kokoszkiewicz[b]

[a]UNESCO-IHE Institute for Water Education, Department of Hydroinformatics and Knowledge Management, 2601 DA Delft ,The Netherlands (a.vangriensven@unesco-ihe.org)
[b]CERN - European Organization for Nuclear Research
email: lukasz.kokoszkiewicz@cern.ch

Abstract: The Soil and Water Assessment Tool (SWAT) has been used widely for large scale applications, reaching entire continents. Within the EU funded EnviroGrids project, a detailed application of SWAT on the Black Sea Basin is envisaged using high resolution data. In order to support the computation, the model is run on a computer grid. The use of the SWAT allowed for such computations with little adaptations to the source. A 3-step procedure is needed. In the first step, a program is run in order to split the model into several sub-models. Afterwards, the sub-models are run in parallel. In a last step, the outputs of the sub-basins are collected at a central computer and the routing is performed.
High computations are also needed when simulations have to be repeated, such as for sensitivity, calibration and uncertainty analysis. In these cases, the simulations are repeated for different parameter sets. In this paper, we discuss the gridification of the algorithm "LH-OAT" that performs sensitivity analysis and has been linked to the SWAT model. The results show a clear improvement in calculation time. Nevertheless, it is concluded that the parallel computing of a distributed model is mainly beneficial for large scale applications with high resolution, while running the sensitivity analysis algorithm has more general and obvious benefit. In a next step, the gridification will be optimised depending on the application and the overheads that are due to submission and receiving of files, as well as potential waiting times for executions on the grid.

***Keywords**:* Parallel computing, grid computing, hydrological model, SWAT

1. INTRODUCTION

The Soil and Water Assessment Tool (SWAT) is an increasingly popular large scale watershed modelling software. The scale of the problems that use this modelling software seems to have increased to a remarkably large degree over time. The whole of continental US and the whole of Africa were modelled on separate occasions using SWAT with commendable results.

The effort being carried out by the EU funded 'EnviroGRIDS@Black Sea Basin' project is another of such pushes that require large scale SWAT models to be run and deliver at near real-time. The 'EnviroGRIDS@Black Sea Basin' is a project among whose aims are linking, gathering, storing, managing and distributing key private or hard-to-find environmental data in an easily accessible, shareable, and process-able format. Developing

the access to real time data from sensors and satellites, and an early warning system that will inform in advance decision makers and the public about risks to human health, biodiversity and ecosystems integrity, agriculture production or energy supply provoked by climatic, demographic and land cover changes are among the goals of the project. The project is working to make such tools and services to be available on a grid infrastructure. It intends to use the Soil and Water Assessment Tool (SWAT) to create high resolution hydrological models of the entire Black Sea Catchment.

This push for large scale modelling, or the need for near real-time output, however, comes at the expense of a huge computation power. This expense will prove even costlier whenever there is a need to iteratively run such very large scale models for common practices in modelling such as calibration and uncertainty analysis. Several endeavors for the reduction of computation time in large scale processes have been made, both related to SWAT and other model codes.

This study describes two types of gridification for hydrological modelling: one where a single simulation is run on a grid by splitting the model into sub-models. This concept is applicable for the computation of large scale spatially distributed SWAT models, thereby reducing the computation time, among other advantages, by parallelization of components or processes. In a second category of the grid application, multiple simulations of a single model are split over the computation grid, as may be needed for model parameter calibration, sensitivity and/or uncertainty analysis tools. In such types of applications,1000 CPUs may be made to compute 1 simulation each instead of having to do 1000 simulations on a single PC.

1.1 Grid computing

Since its inception in the late 1990s, grid computing has advanced so rapidly to take an appealing acceptance by major institutes and projects worldwide. In recent years, Grids have emerged as wide-scale distributed infrastructures that support the sharing of geographically distributed, heterogeneous computing and storage resources.

Coordinated by CERN (the European Organization for Nuclear Research) and funded by the European Commission, the EGEE project (Enabling Grids for E-sciencE) aims to set up a worldwide grid infrastructure for science. The EGEE Grid initially focused on two well-defined application areas, Particle Physics and Life Sciences, due to the fact that these communities were already Grid aware and ready to deploy challenging real applications at the beginning of the project. A range of applications are supported by this grid currently.

Many grids are used for e-science: enabling projects that would be impossible without massive computing power. Scientists can now share data, data storage space, computing power, and results easier than ever before via the grid which enables researchers to tackle bigger questions: from disease cures and disaster management to global warming and the mysteries of the universe.

1.2 EGEE Grid infrastructure

The EGEE Grid (Enabling Grid for E-sciencE) is currently the largest multi-science Grid infrastructure in operation which federates over 250 resource centres world-wide, providing more than 68,000 CPU's and more than 20 Petabytes of storage. This infrastructure is used by several thousands of scientists distributed in over 200 virtual organizations with access to EGEE resources that is managed by the gLite1 middleware.

The EGEE grid, like any other grid infrastructure, has its own middleware and access protocols. It demands the software to be run on the Grid to be developed in specified (parallelized) formats for better computational efficiency on the grid environment Whereas new software tools might be developed with these requirements in mind, existing scientific legacy codes or applications, the majority of which are not grid-aware yet, often require major code restructuring before they could run on the grid.

"Gridification" is adapting applications to include new layers of grid-enabled software. An application that ordinarily runs on a stand-alone PC must be "gridified" before it can run on a grid. Just like "webifying" applications to run on a web browser, grid users need to "gridify" their applications to run on a grid (GridCafe2). Once gridified, thousands of people will be able to use the same application and run it on interoperable grids.
For example, a gridified data analysis application will be able to:

- Obtain the necessary authentication credentials to open the files it needs
- Query a catalogue to determine where the files are and which grid resources are able to do the analysis
- Submit requests to the grid, asking to extract data, initiate computations, and provide results.
- Monitor progress of the various computations and data transfers, notifying the user when analysis is complete, and detecting and responding to failures (collective services).

In this study, we will gridify SWAT on the EGEE grid and demonstrate it with an auto-calibration routine for a water quality model.

2. GRIDIFYING A SWAT MODEL

2.1 Soil and Water Assessment Tool (SWAT)

The Soil and Water Assessment Tool (SWAT) is a physically based model that can simulate water quality and quantity at the watershed scale (Neitsch et al., 2005). SWAT computes continuous on a daily time step. The model is primarily designed to predict impacts of management on water, sediment, and agricultural chemical yield in gauged and un-gauged basins. It allows for the watershed delineation into a large number of sub watersheds. The major modules in the model include hydrology, erosion/sedimentation, plant growth, nutrients, pesticides, land management, stream routing, and pond/reservoir routing. Input climate (precipitation, maximum and minimum temperature, relative humidity, wind speed, solar radiation) are required on a daily temporal resolution, although the most recent version of the model allows also for hourly input files.

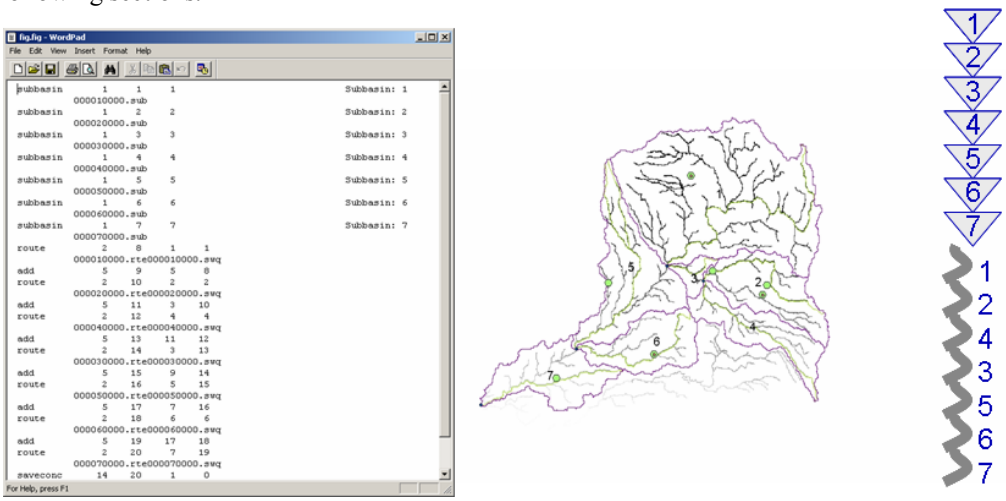
A watershed modelled using SWAT is partitioned into different required and/or optional objects of subunits such as sub-basins, reaches/main channel segments, impoundments/reservoirs on the main channel network and point sources. First, a watershed in SWAT is subdivided into sub-basins which are, thus, the first level of the subdivision. Sub-basins are defined by geographical positions in the watershed and are spatially related to one another (SWAT User's Manual, Version 2000). All sub-basins drain into the river network where water is routed from upstream to downstream reaches. Sub-basins will contain at least one, and theoretically as many as unlimited number of, HRUs (Hydrologic Response Units) and a main channel or reach. Furthermore, impoundments, a pond and/or a wetland, may also be defined in a sub-basin.

Secondly, the land area in a sub-basin may further be divided into hydrologic response units, so called HRUs. Hydrologic response units are portions of a sub-basin that possess unique land use, management, or soil attributes (SWAT User's Manual, Version 2000). Unlike in the case of sub-basins, no spatial relationship or interaction can be specified among HRUs. Sediment, chemicals or nutrient loadings from each HRU are computed independently and then summed up to determine the total load from the sub-basins.

A watershed model should also incorporate one reach or main channel associated with each sub-basin. This channel carries loadings from the sub-basin or outflow from the upstream reach segments into the network of the watershed in the associated reach segment.

2.2 Parallelization of the SWAT model

The parallelization of the Soil and Water Assessment Tool (SWAT) models for Grid execution is analyzed with various approaches. The general principle of this study might simply be stated as: divide-compute-merge. In other words, it follows the general principle of dividing or splitting a large scale, high resolution SWAT model into several small model components or subcomponents and computing all in parallel on the Grid, after which outputs from all these components or subcomponents are collected and merged for presenting results similar to that of the original, undivided model. Methodological questions such as ”On what level can one split SWAT models for optimal efficiency: sub-basin or HRU level?” are among the first that require some analysis to answer. This study focuses on sub-basin level splitting for speculative reasons that HRU level splitting might increase overhead while splitting and merging, and communication overheads while computing on the grid. Further questions that require precise analysis are: From which files do we retrieve all the information for independent computations of such sub-basin level splits? Do we have to split all sub-basin specific information into independent sub-basin files or could we still maintain some common input files for all the sub-basins as they are in the original model? What is the advantage or disadvantage of splitting or maintaining those common input files? This study will try to give answers to these and other questions in the following sections.



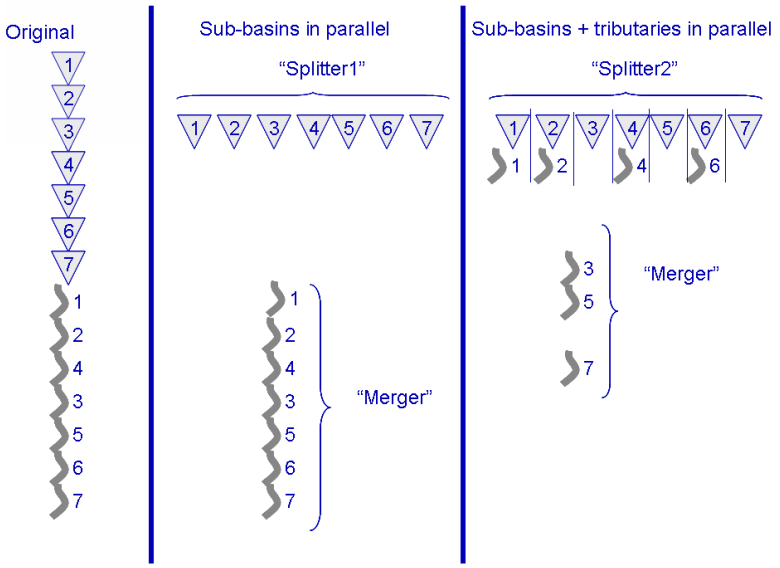
(a) (b) (c)

Figure 1: The “Fig file” (a) for the SWAT model of the Nzoia catchment (Kenya) (b) and the corresponding conceptual schematization for this file (c) , triangles represent the sub-basin and the lines represent the reaches (c).

Within a watershed, the sub-basin processes are computed independent from each other and within the sub-basin, the HRUs are computed independently as well. On the other hand, the computations of the reaches depend on the outputs of the sub-basins computation, but not the other way around. In other terms, reaches in sub-basins are supposed to incorporate loads from sub-basin processes, and thus are dependent on them. The reaches also depend on each other (downstream reaches depend on the outputs of the upstream reaches). For that reason, the parallelization is performed at the level of the sub-basins only, and not for the reaches. All HRUs within a sub-basin share the same weather input file, which are the largest model input files in SWAT.. An HRU level splitting, thus, would mean the processing of these big input files during splitting, which is computationally costly. It was chosen not to go up to HRU level splitting of SWAT models in this study for the simple reason that doing so would reduce the gain in computation time. The level of sub-basins is for that reason much more efficient in exchanging of input- and output files.

Watershed configuration in SWAT models is stored in a file called “fig-file”. This file consists of a number of “command lines” for each time step to be computed in sequence. Among the most important command lines are “sub-basin” that computes the sub-basin processes, “route” for the computation of the processes in a reach and “add” that sums outputs from previous command lines. A typical structure of the file is given in the example of figure 1a, where the first block represent lines for the computations of the sub-basins (7

in total), followed by a block with several "route" and "add" commands. The latter represents the river network.



(a) (b) (c)

Figure 2: The original configuration "sequence" (a), the split SWAT models "paral1"(b)and "paral2" (c)

In the original SWAT code, the sequence of computation is as in Figure 2(a), all sub-basins, and all HRUs within the sub-basins, are computed first followed by the main channel networks/reaches. These computations are iterated for every day of every year stated for the simulation period. Two approaches of parallelization were devised in this study. In a first parallelization configuration, all sub-basins are computed in parallel ("paral1" in figure 2b), followed by all reaches in series. In a second configuration, the reaches of upstream sub-basins were computed during the parallelization process of the sub-basins in sequence to the sub-basin they belong to, followed by the reaches that depend on upstream reaches "paral2", as presented in figure 2c.

2.3 Software tools used for parallelization of the SWAT model

In total 5 programs are used:.

(1) Swat2005.exe: the original SWAT model code
(2) SWATgrid.exe: adapted SWAT model code
(3) Splitter1 or Splitter2: written for the splitting SWAT models(python/java)
(4) SWATmerger.exe: written for merging sub-basin outputs(python/java)
(5) Ganga: A job submission tool for the EGEE grid (python)

SWAT2005.exe is the original SWAT code and is used for the computation of the sub-basins (and the routing of the upstream reaches in the second configuration). **SWATgrid.exe** is an adapted version of the SWAT model that reads in the output files from the models of the sub-basins run on the grid, and initiates the add/route commands. The major change with the adapted SWAT code used for this purpose from the original is that it allows the routing of stream networks by skipping sub-basin computations.
Splitter1 splits the model into several sub-models. It creates a model for each sub-basin. The following files needed to be adapted:

- Fig.fig: this file is reduced to one command line for the sub-basin computation and one to save the outputs to a file.
- The precipitation (*.pcp) file contains only the rainfall gage data that is used in the sub-basin
- Only the input files that are use, or copied to the folder represent the sub-basin, including the executable program swat2005.exe

- An adapted configuration file is created for the model that will read the outputs of the sub-basins that performs the routing.

Splitter2 is similar to Splitter1, except that it includes a routing command and a routing input file in the configuration for routing of upstream tributaries.
SWATmerger.exe copies the output files of the sub-basins into a single folder, and calls the swatgrid.exe code.
Ganga(a python script) is a grid user interface for specifying and for submitting jobs to the grid which connects to the grid, submits the files to the grid machines and copies the results back to the PC of the user.

3. Parallelization of the LH-OAT Sensitivity Analysis Algorithm

3.1 The LH-OAT algorithm

Sensitivity analysis is the process of determining the rate of change in the model output with respect to changes in the model input parameter. It is needed to determine how the hydrological model outcomes are sensitive to their parameter. Furthermore, most hydrological models are very complex and over-parameterized. The identification of the most important parameters so as to reduce time and complexity, before carrying out time-intensive activities such as model calibration, requires a faster and efficient way of sensitivity analysis tool and technique.

In this study, parallelization of the LH-OAT software tool which implements the Latin Hypercube-One factor At a Time method (van Griensven et al, 2005) is carried out.

The LH-OAT tool performs a Latin Hypercube (LH) sampling followed by One-Factor-At-a-Time (OAT) sampling (Figure 3). It starts with taking N Latin Hypercube sample points for N intervals, and then varying each LH sample point P times by changing each of the P parameters one at a time.
The method operates iteratively. Each loop starts with a latin-hypercube point. At each Latin Hypercube point j, a partial effect Si,j for each parameter ei is calculated as (in percent):

$$S_{i,j} = \left| 100 * \left(\frac{M(e_1,...,e_i*(1+f_i),...,e_P) - M(e_1,...,e_i,...,e_P)}{\frac{M(e_1,...,e_i*(1+f_i),...,e_P) + M(e_1,...,e_i,...,e_P)}{2}} \right) / f_i \right| \quad \textbf{(1)}$$

where M(.) refers to the model functions, f_i is the fraction by which the parameter ei is changed (a predefined constant) and j refers to a Latin Hypercube point. In equation 1, the parameter was increased with the fraction f_i, but it can also be decreased since the sign of the change is defined randomly. Therefore a loop requires P+1 runs.
A final effect is calculated by averaging these partial effects of each iteration for all Latin Hypercube points (thus for N loops). The method is very efficient, since N intervals (user defined) in the LH method require a total of only N*(P+1) runs.
The final effects can be ranked with the largest effect being given rank 1 and the smallest effect being given a rank equal to the total number of parameters analysed. Oftentimes during a sensitivity analysis for a particular dataset, some parameters have no effect on model predictions or performance. In this case they are all given a rank equal to the number of parameters.

This method combines the robustness of the Latin Hypercube sampling method which ensures that the full range of all parameters has been sampled with the precision of a One-factor-At-a-Time(OAT) sampling method. The OAT sampling method assures that the changes in the output in each model run can be unambiguously attributed to the parameter altered in the input.

3.2 Parallelization of the LH-OAT algorithm

A LH-OAT application requires a number of simulations (equal to number of intervals for the Latin Hyper cube times the number of parameters +1). These simulations are independent from each other and can thus easily be parallelised in order to perform the simulations, and to collect the objective function or a particular model output of interest, for each of these simulations. In our example we look at the average flow and the average sediment load at the outlet as an output of interest.

The parallelization is performed by splitting the file with the parameter values into several files. Each of these files controls a batch execution that can be done on a particular processor. All the result files, containing the average flow and sediment load value for the simulations, are merged to one file (named sensobjf.out).

3.3 Software Tools used

The following software tools are used for the parallelization of the LH-OAT algorithm :
(1) **LH_OAT_sampling.exe:** a program that samples the parameter sets, based on parameter ranges and the number of intervals in the Latin Hypercube
(2) **SensSplitter.py:** a python script that creates subfolders, each of which containing the model and a file containing parameter files. The folder also contains a file with specifications of how the parameters should be changed (changepar.dat). In addition, the "ICLB" variable within master watershed file of SWAT models (file.cio) should get a code-value '5' indicating that a batch file (batchpar.dat) with parameter sets should be performed.
(3) **swat2005.exe**, that the original SWAT model
(4) **SensMerger.py-** a python script that merges the files from the sub-parameter sets into one file (sensobjf.out)
(4) **LH_OAT_analysis**: an algorithm that produces sensitivity rank for parameters, based on the parameter sets (senspar.out) and the objective functions (objpar.out)

After splitting the SWAT models and the LH-OAT parameter sets on a local machine using the respective splitter tools developed in this study, each sub-component is submitted to multiple high performance computing machines on the EGEE grid. The results are collected and presented using the respective merger tools developed for this same purpose in this study. The outcomes of the tests are depicted and discussed below.

4. RESULTS

4.1 Parallelization of SWAT Models

The computation times for the model are depicted in tables 1 to 4 below. In table 1, a small model, presented in figure 1, with 7 sub-basins simulating data for 43 years run on 7 CPUs on the grid, shows a speedup ratio of 2.96 in computation. A larger model built for the Balaton Lake basin, containing 204 sub-basins simulating data for 16 years, run on 204 CPUs, achieved a speedup of 1.4 in computation time. From these results, it seems valid to conclude that the speedup in computation times is higher when the number of simulation period is longer. This is because execution time for splitting and merging of the models is more sensitive to number of sub-basins than number of years of simulation.

Table 1 Computation time results for a very small model.

7 sub-basins, 7 HRU's:	Computation time (s)		Number of CPUs	Speedup
Full model ("sequence")	32			
Parallelistion Experiment	Approach I	Approach II		
Splitting	1.2	1.4		
Sub-basin	3.3	5		
Merging	6.3	4.4		
Parallel computing	10.8	10.8	7	2.96

Another interesting scenario is the same model with 7 sub-basins but a much higher resolution, that is, multiple HRUs per sub-basin. This scenario, with a speedup of 6.3 as shown in Table 2, demonstrates a clear gain in computation time for high resolution models, which is the objective of this study.

4.2 Parallelization of LH-OAT Sensitivity Analysis Algorithm

The computation time is equal to the total computation time divided by the number of sub-models that are created for the parallelization. As shown in the table below, splitting 330 parameter sets [i.e., 10*(32+1)] for independent simulations on 330 processing units on (CPUs) on multiple HPCs on the grid gives a speedup ratio of 7.3. Similar to the parallelization of high resolution SWAT models, this result shows a clear gain in computation time.

Table 2 Computation time results in seconds for different tasks

Task	Computation for 1 CPU in s (m)	Speedup for gridcomputing (#CPU's)
7 sub-basins, 7 HRU's	**32**	**2.96 (7)**
7 sub-basins, 1390 HRU's	**2671**	**6.9 (7)**
204 sub-basins, 204 HRU's	**249**	**1.4 (204)**
Sensitivity analysis	**2835**	**7.3 (330)**

5. CONCLUSION

The 'gridification' of the Soil and Water Assessment Tool allows for running the model on a grid environment. The results show an improvement of the performance in computation time, too. Nevertheless, there may be a considerable loss of time during the job submission procedure between the user's PC and the grid which results in communication overhead, especially with very small models. It is therefore unlikely that grid computing would become very beneficial for small to mid-size cases. Nevertheless, for large cases, especially with multiple of HRUs per sub-basin, grid computing may gain a lot in computation time. In addition, it may also solve memory problems that may originate from running large models on a single computer. The applicability of gridified algorithms for multiple simulations, such as for sensitivity analysis, calibration and uncertainty analysis is more straightforward.

An important limitation of the present GRID-computations is that there are no interactions possible between the computer nodes. For that reason, we could not account for any relationships between the sub-basins or between rivers and sub-basins (eg water transfers). Also for integrated modelling, where model-dependencies are present by default, such grid computations may not be helpful in reducing the high computation load. In a next step, tests of the parallelization tools and techniques used so far in this study on multiple processors on the same HPC machine will be experimented.

ACKNOWLEDGEMENTS

The authors wish to thank the EU/FP7 EnviroGRIDS@BlackSeaBasin project for the financial support.

REFERENCES

Foster, I. and Kesselman, C. editors, "The Grid: Blueprint for a New Computing Infrastructure", Elsevier, 2004.

Neitsch, S.L, Arnold, J.G., Kiniry, J., Williams, J.R. "Soil and Water Assessment Tool Theoretical Documentation, Version 2005", Grassland ,Soil and Water Research Labratory, Agricultural Research Service and Blackland Research Center, Texas Agricultural Research Station, Temple, Texas, 2005.

van Griensven, A., Meixner, T, Grunwald, S, Bishop, T, Diluzio, M, Srinivasan, R. "A global sensitivity analysis tool for the parameters of multi-variable catchment models." *Journal of Hydrology*, 324: 10-23, 2006.