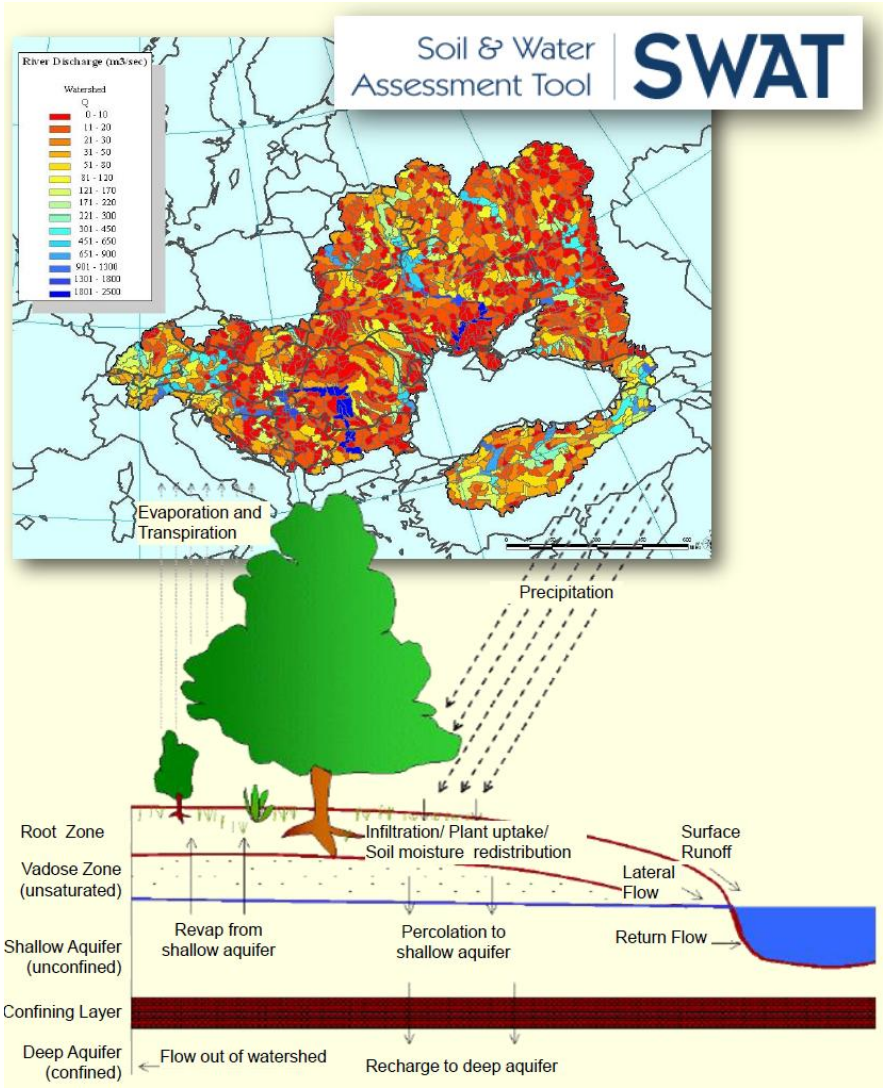
gSWAT

Victor Bacu, Dorian Gorgan, Denisa Rodila,
Danut Mihon, Teodor Stefanut
Computer Science Department
Technical University of Cluj-Napoca
Victor.Bacu@cs.utcluj.ro

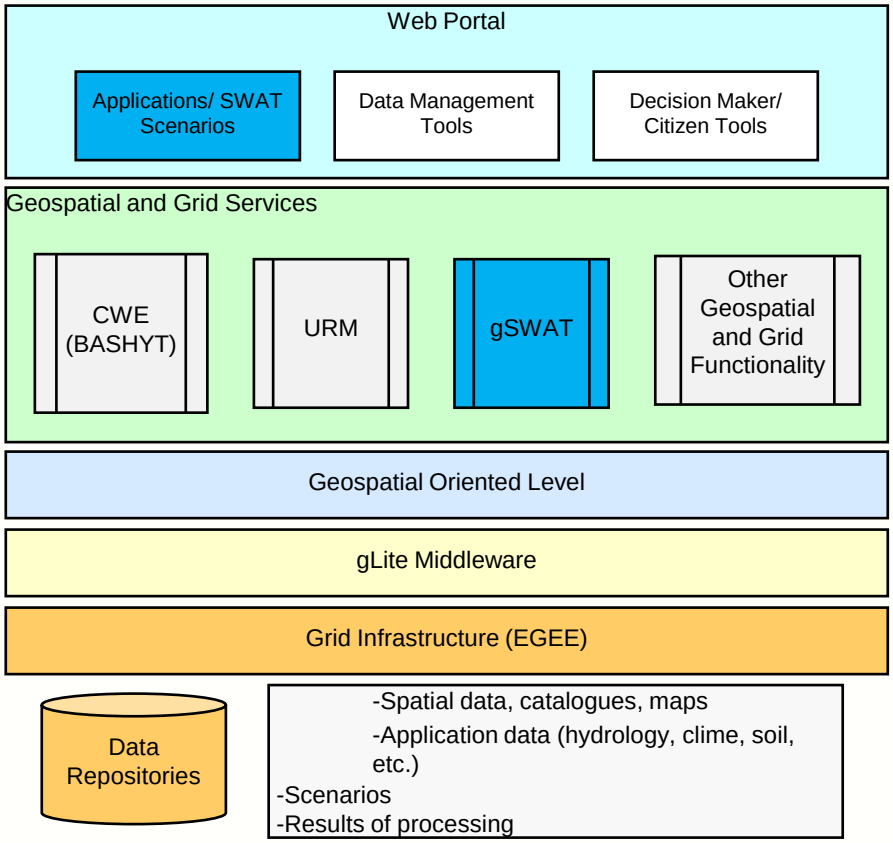
SWAT Overview

- SWAT (Soil Water Assessment Tool)
  - hydrological model
  - operates on a daily time step
  - used for predicting the water resources, sediment, and chemical yields in a specific watershed
- *Input data*: weather, soil properties, topography, vegetation, and land management practices of the watershed
- SWAT estimates the impact of land management practices on water quantity and quality in complex watersheds
- The SWAT model must pass through a careful calibration and uncertainty analysis



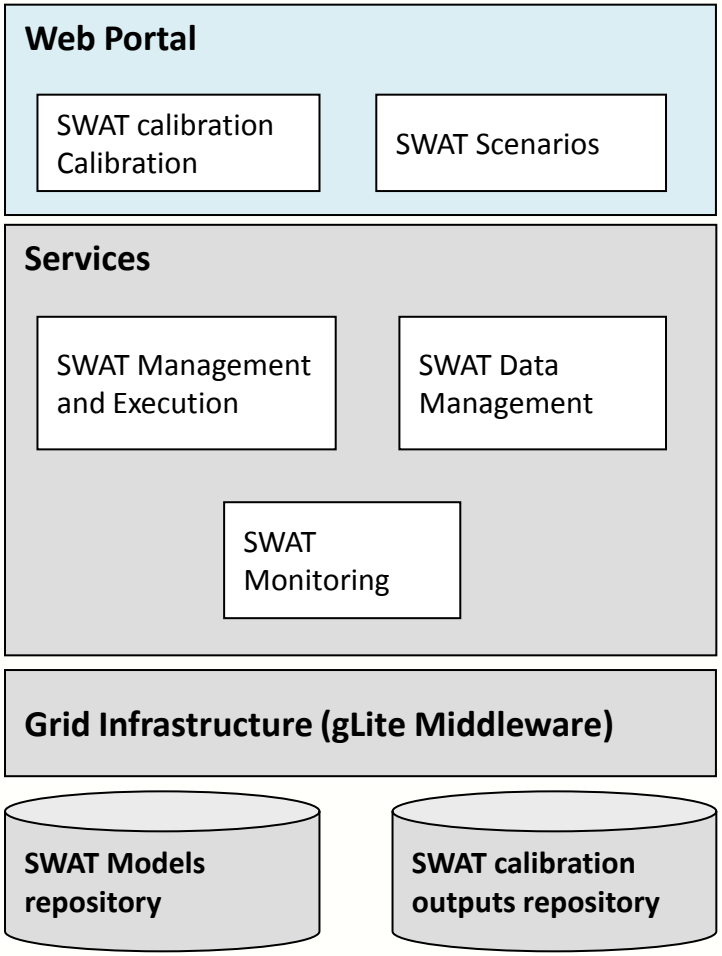
gSWAT Overview

- Main functionalities:
  - Calibration of the SWAT models (on GRID infrastructure)
  - Execution of scenarios based on a calibrated SWAT model (on GRID infrastructures)
- The calibration process uses the SUFI2 uncertainty analysis routine
- Web application



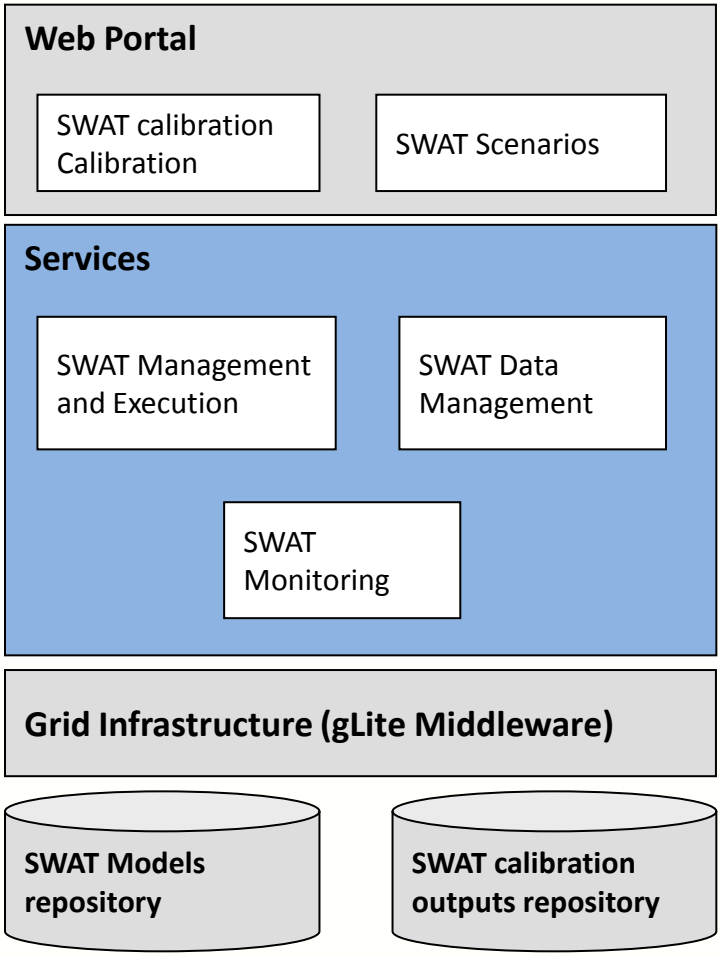
gSWAT Architecture

- Web Portal
  - Developed in Adobe Flex 4
  - Enable different categories of users to use the calibration interface using efficient and intuitive user interaction techniques



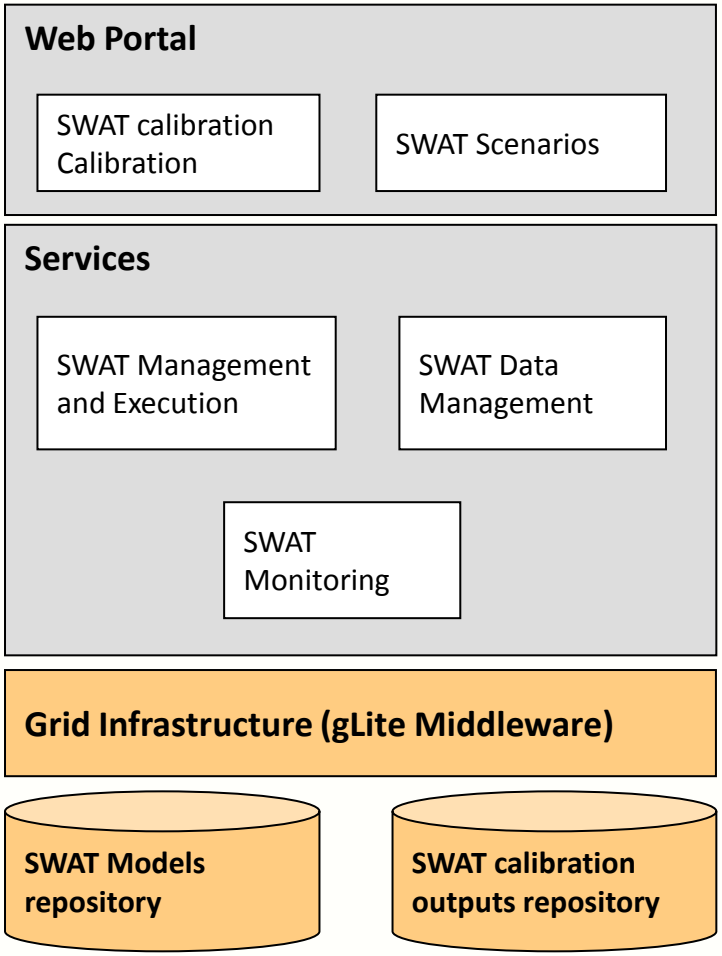
gSWAT Architecture

- Services
  - **SWAT Management and Execution** – used to create new calibration projects, upload SWAT models, start new iterations, save finished iterations, delete iterations, etc.;
  - **SWAT Data Management** – structures the data as is necessary for the calibration process, upload data to Grid in Storage Element, download output data;
  - **SWAT Monitoring** – monitors the execution of the calibrations, update the iteration status, etc.



gSWAT Architecture

- Grid infrastructure
  - Resources
    - Worker Nodes – computational resources
    - Storage Element – storing resources
  - Ganga - frontend for job definition and management
  - Diane – support a more efficient usage of the distributed computing infrastructures



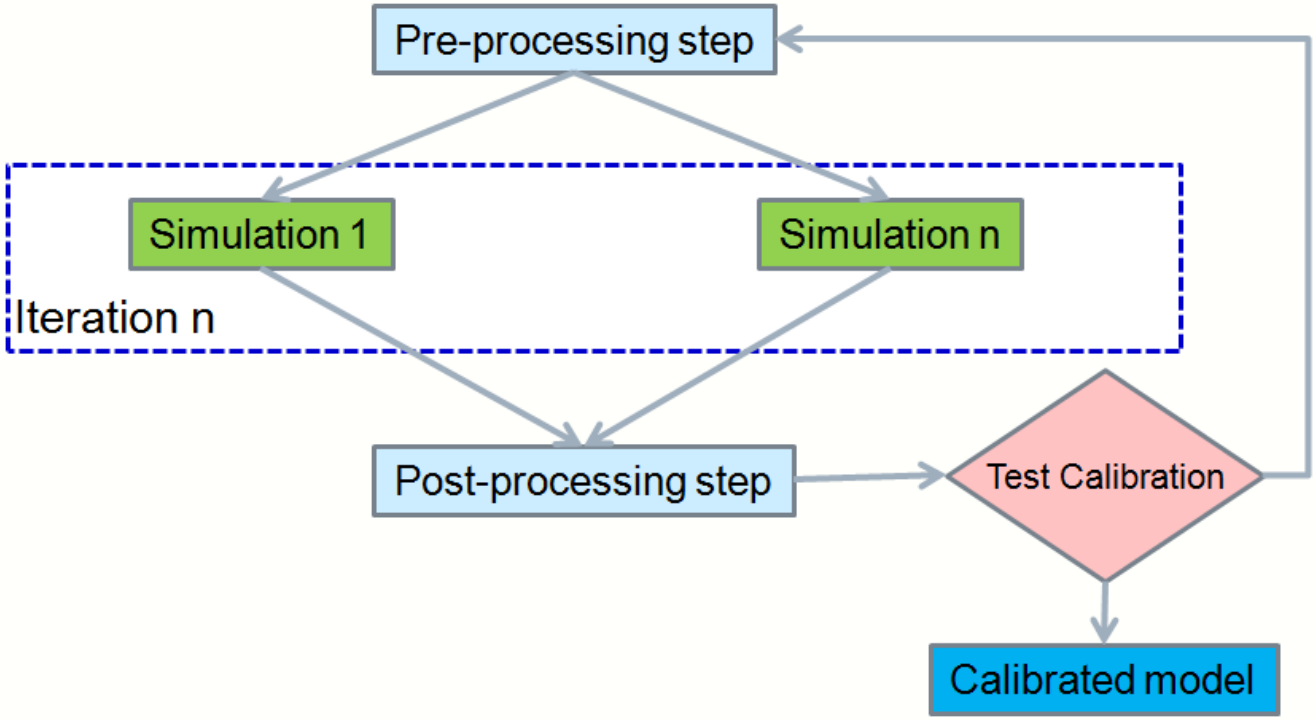
gSWAT Functionalities

- Management of the SWAT calibration projects
  - Create new project
  - Upload SWAT model
  - Delete project
  - Modify calibration parameters
- Management of the execution of the calibration iterations
  - Start and monitor iterations
  - Save and delete completed iterations
- Output visualization
  - Graphical visualization

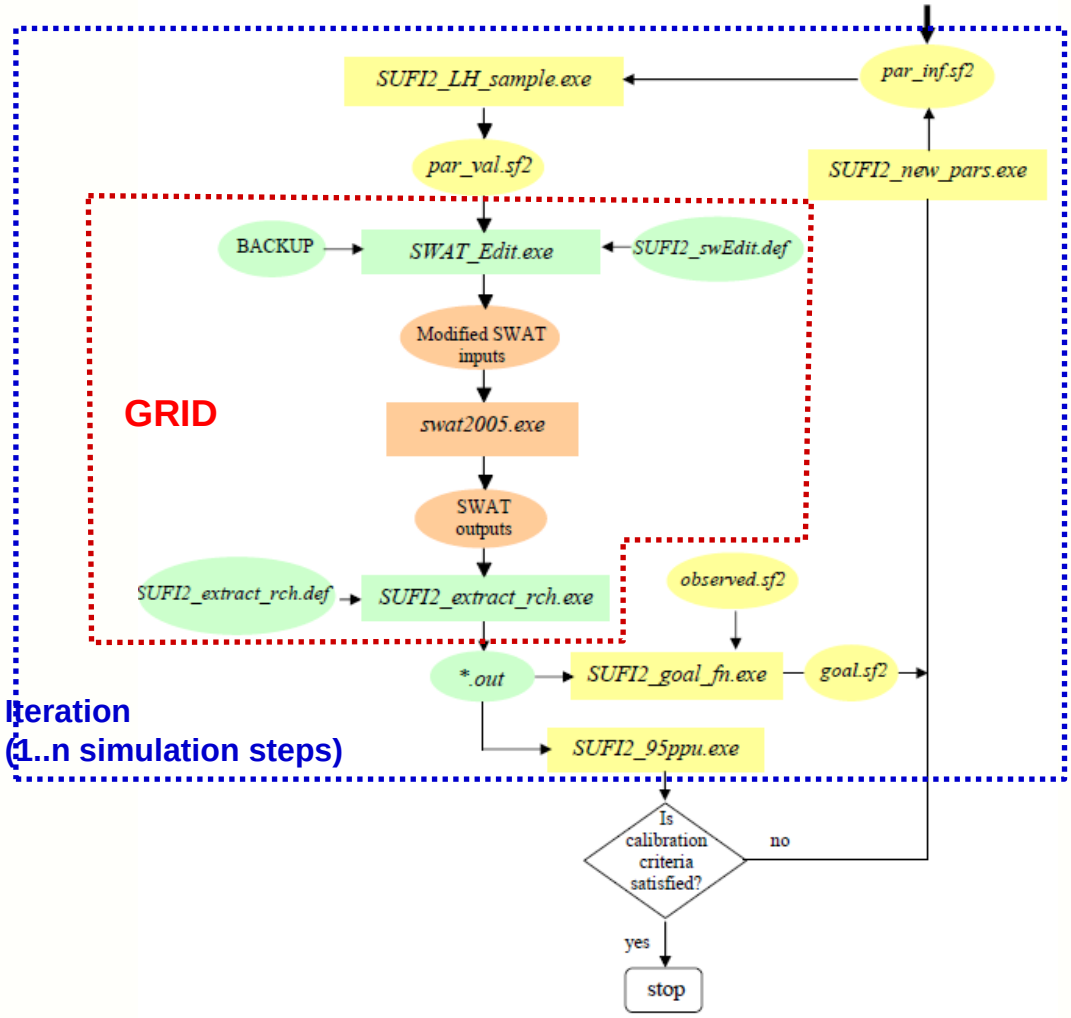
SWAT Calibration Process

- **The calibration process** - execution of several iterations until the calibration criteria is satisfied.
- Each **iteration** process is composed of a number of simulations.
- Each **simulation** is independent on the other simulations.
- The user can choose between several objective functions.
- After performing an iteration the user can change the type of the objective function to see the effect of this objective function.

Processing steps



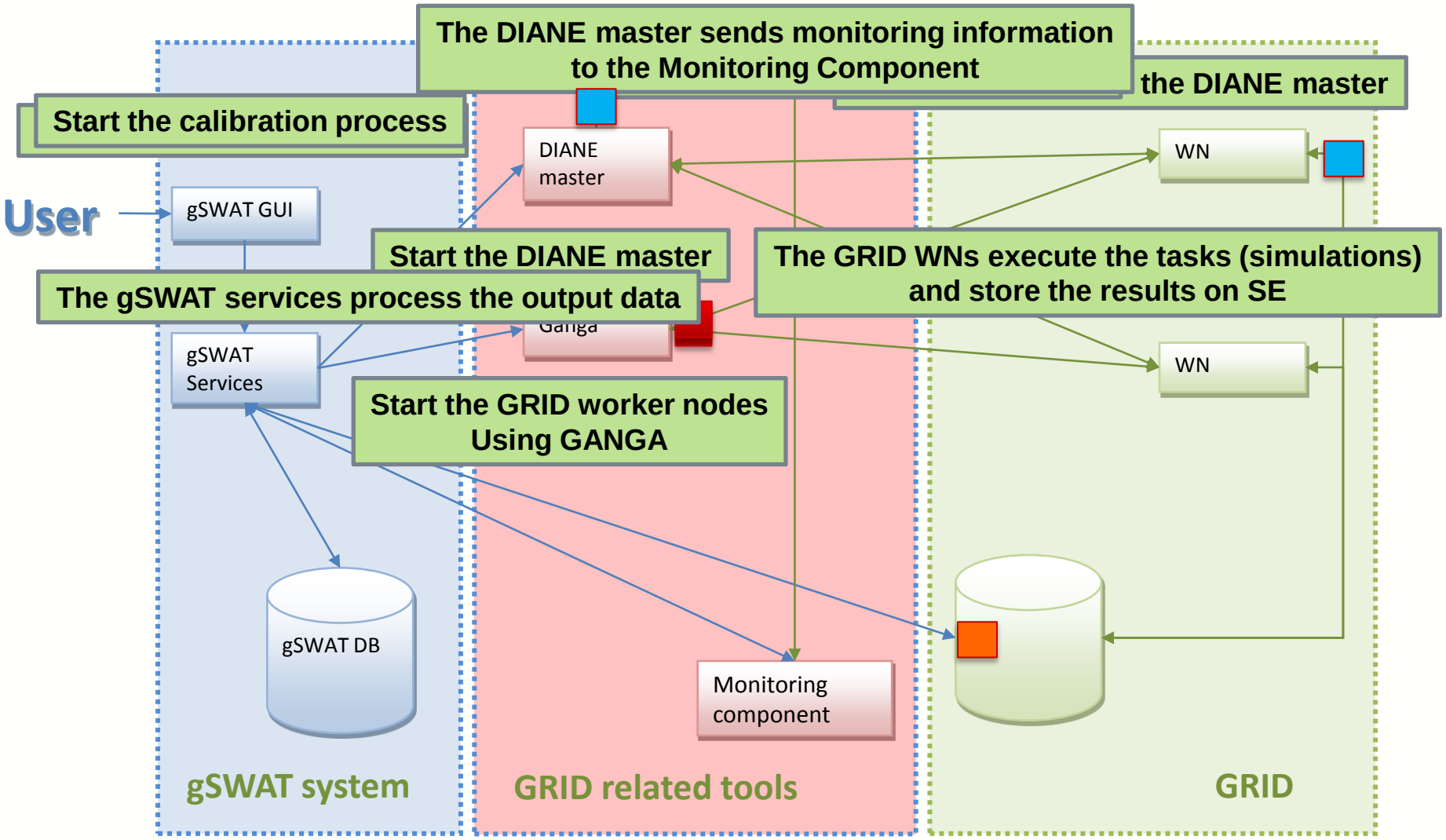
The calibration process



gSWAT Execution Flow

1. Create the job script
2. Create the DIANE script
3. Start the DIANE master
4. Start the Grid workers using GANGA
5. Monitor the execution of the tasks (simulations)
6. Download the output data

gSWAT Execution Flow



Executing simulations on Grid

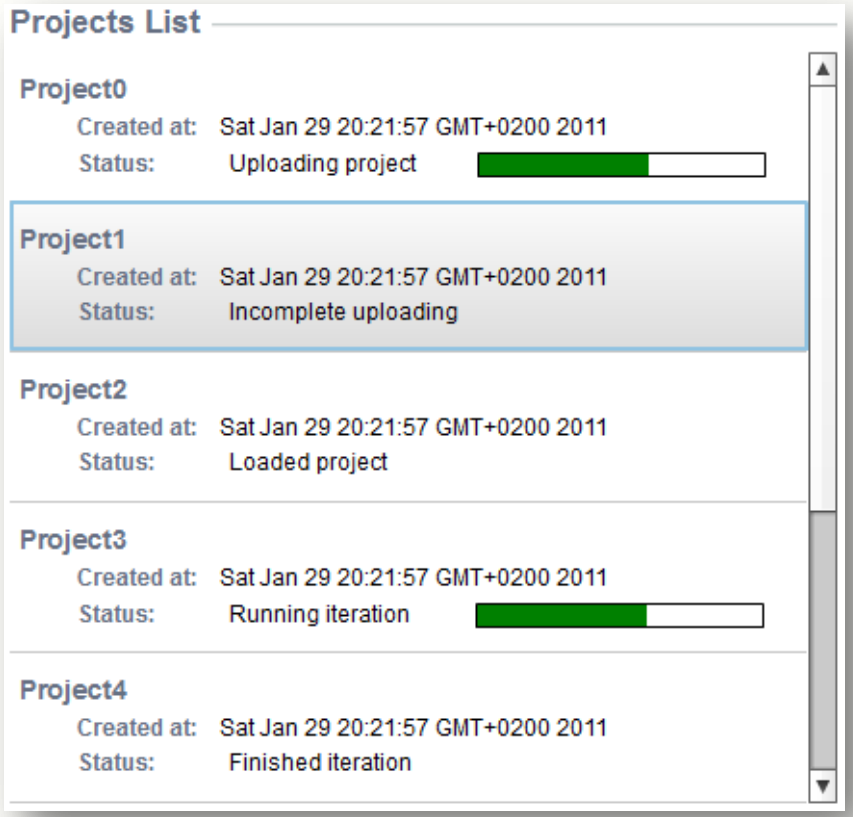
- The jobs are launched using DIANE and Ganga
- On the Grid Worker Node we perform the following steps:
  - Copy locally the SWAT model archive;
  - Execute the SWAT simulation;
  - Archive and store the SWAT outputs.

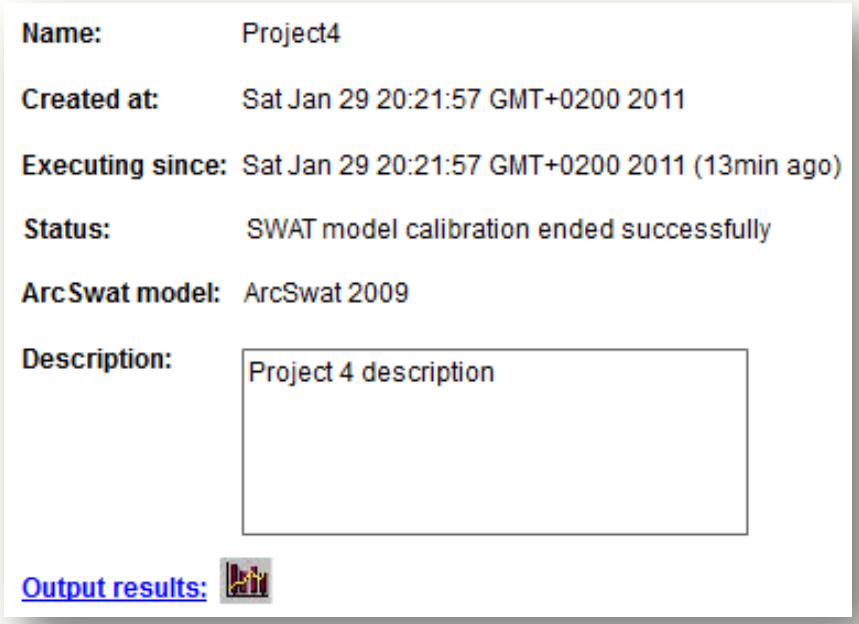
gSWAT

- Data management
  - The SWAT model is stored on a Storage Element
  - The Grid jobs will copy locally the input SWAT model from the Storage Element
- Monitoring component
  - DIANE monitoring component (http://dianemon.cern.ch)
  - The gSWAT monitoring component parses the JSON
  - Store the information in the gSWAT database

gSWAT User Interface

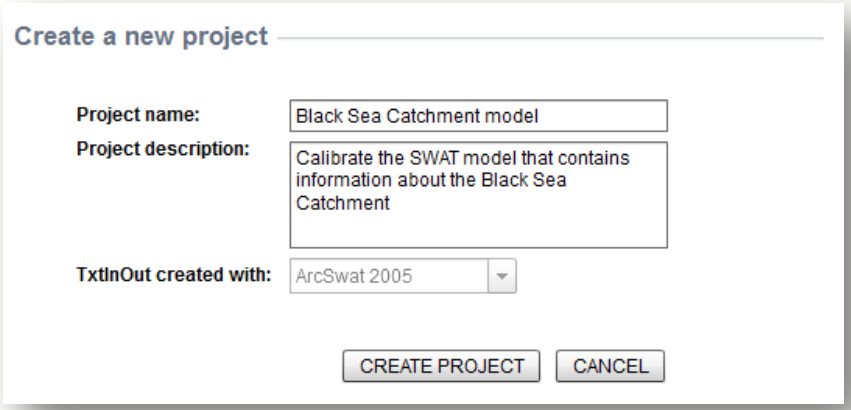
- Project list and detailed information on the selected project



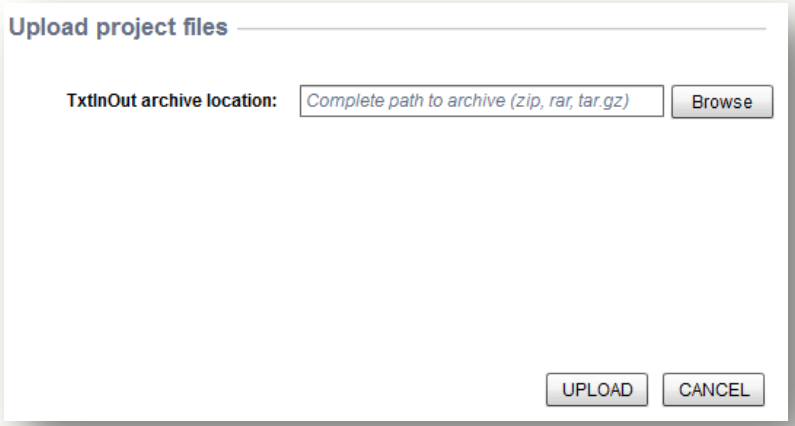


Create a new project

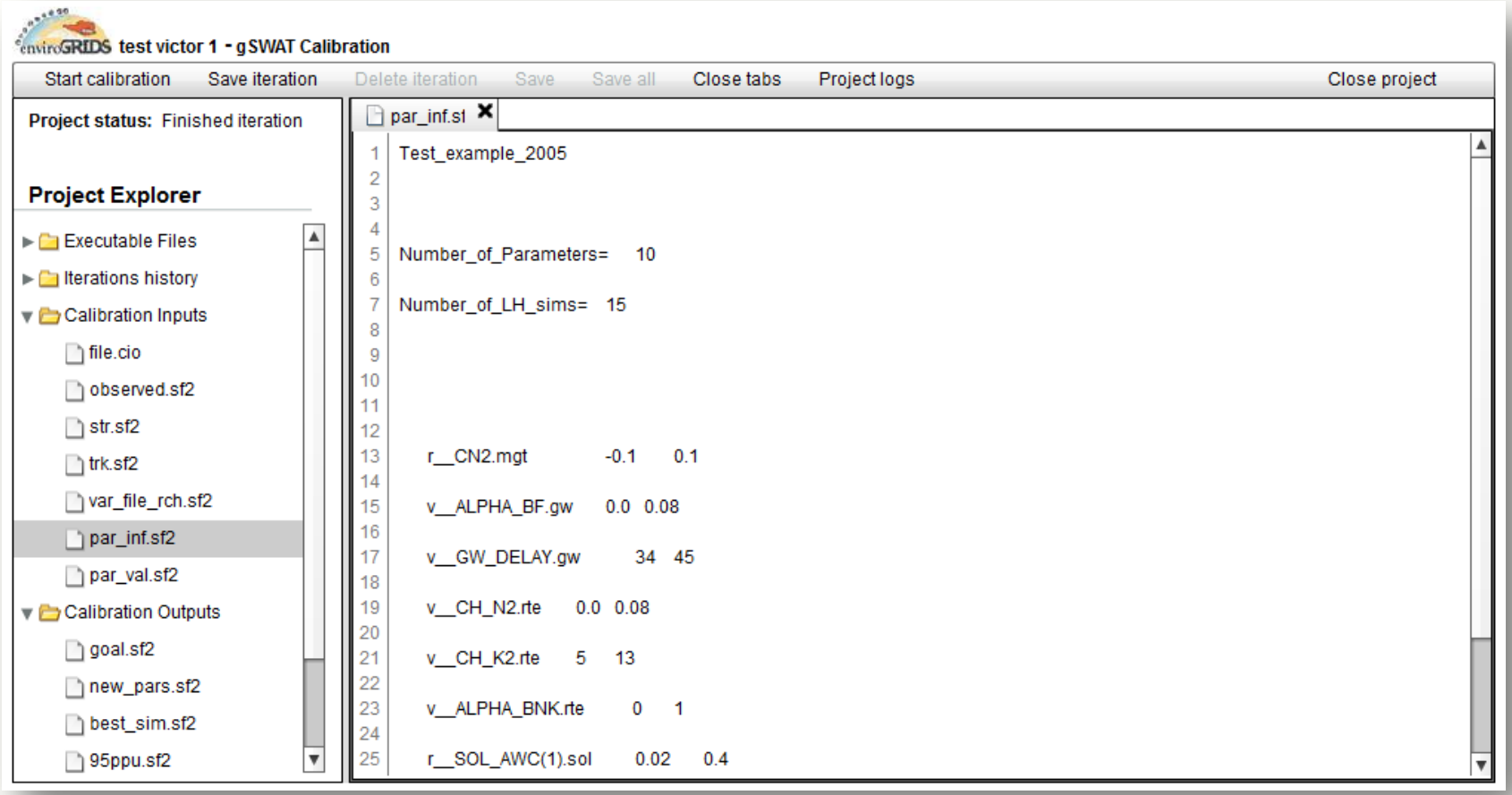
- Step 1: Define general project information



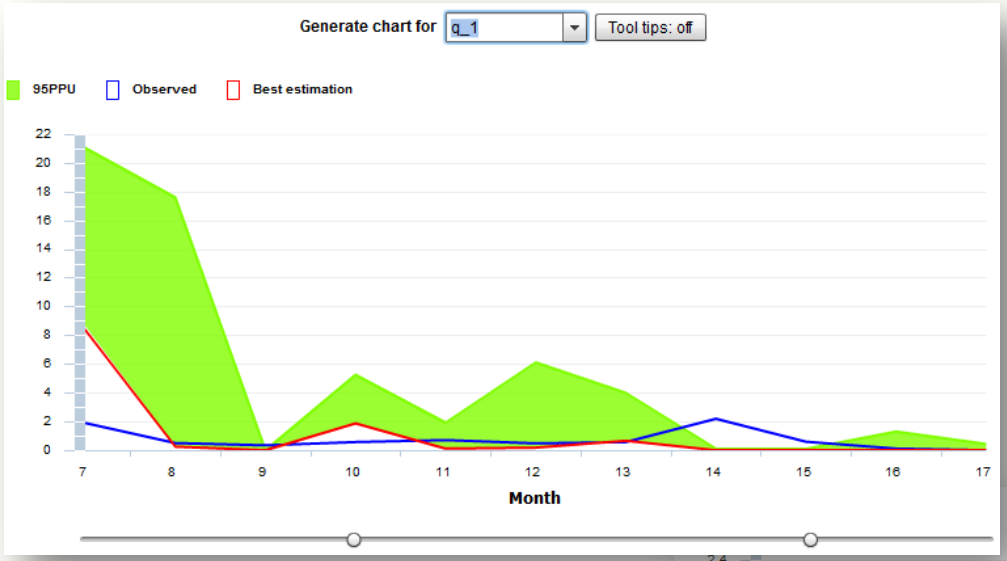
- Step 2: Select the SWAT model to be uploaded

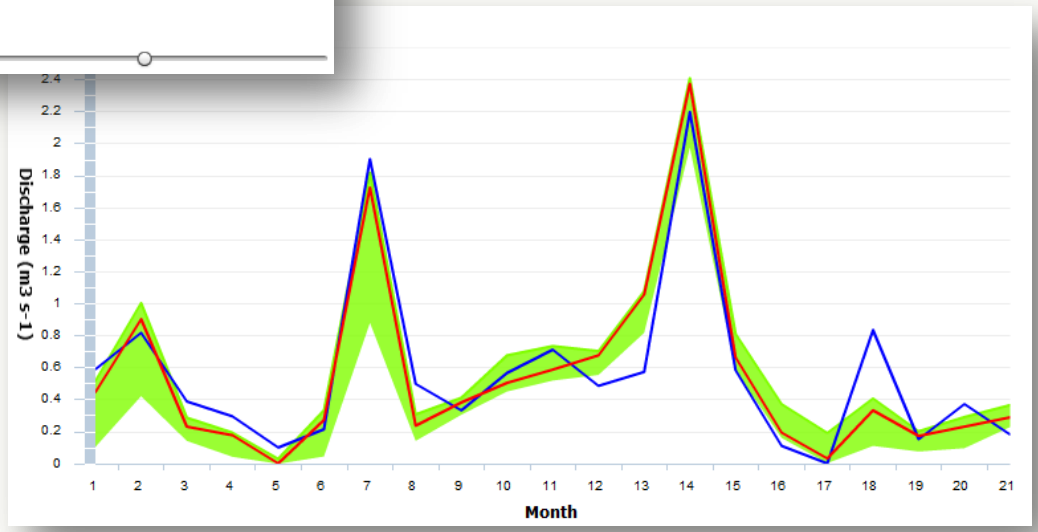


Modify SUFI2 calibration parameters



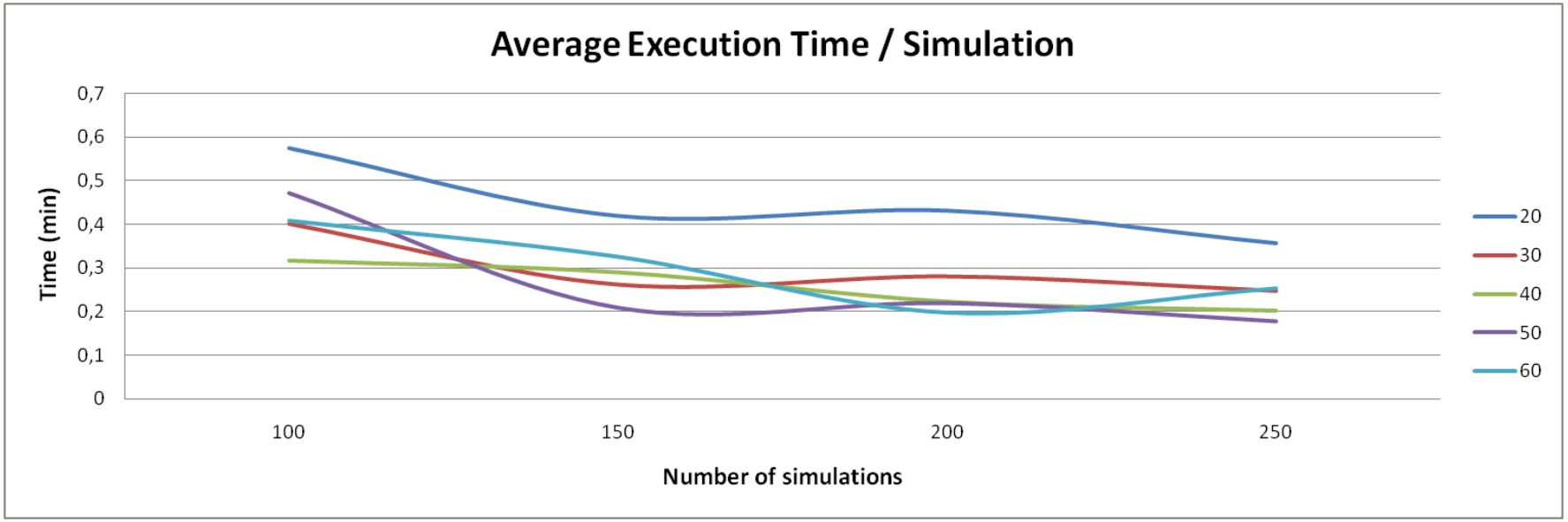
Calibration Output Visualization





Performance

- vo.gear.cern.ch VO
- Variables: the number of worker nodes and the number of SWAT simulations



Future work

- The possibility to define and run scenarios
- Export the calibrated SWAT model to other systems, such as BASHYT, for output visualization

Thank you for your attention!

Questions?

Victor Bacu, Dorian Gorgan, Denisa Rodila,
Danut Mihon, Teodor Stefanut
Computer Science Department
Technical University of Cluj-Napoca
Victor.Bacu@cs.utcluj.ro