SWAT Gridification

Title	SWAT Gridification
Creator	Kokoszkiewicz Łukasz (CERN)
Creation date	11.05.2010
Date of revisions	13.05.2011 - Gregory Giuliani 16.05.2011 - Gregory Giuliani 17.05.2011 - Nicolas Ray 18.05.2011 - Jamie Shiers
Subject	SWAT, Gridification, EnviroGRIDS VO, Monitoring
Status	Final
Type	Word document
Description	Explanation of the SWAT gridification process, guidelines for using Ganga in conjunction with Diane software to run SWAT jobs on the grid. EnviroGRIDS VO facts and monitoring tools.
Contributor(s)	Łukasz Kokoszkiewicz, Andrew Maier, Massimo Lamanna, Ann van Griensven, Seleshi Yalew, Karim Abbaspour, Elham Rouholahnejad
Rights	Public
Identifier	Gridified code (D2.8)
Language	English
Relation	Geospatial and Grid Services, EnviroGRIDS portal

Contents

1 List of figures

2 Abstract

This document describes the SWAT gridification task aiming in providing users with possibility to run their SWAT related calculations on the grid. The document highlights SWAT gridification schemas, EnviroGRIDS VO related information and monitoring tools.

3 Executive summary

One of the tasks in EnviroGRIDS project was to gridify SWAT (Soil and Water Assessment Tool), the most important tool used by the hydrological community.

The gridification, in this respect, was focused on searching proper use cases where calculations can be split to sub-jobs, run separately on different nodes and merged at the end. We were successful in finding three opportunities for this kind of calculations: the SWAT model parallelization, LH-OAT (Latin Hypercube One factor At a Time) parallelization and the SWAT-CUP sufi2 algorithm parallelization.

The EnviroGRIDS VO (Virtual Organization) was created to make using of the grid resources easier for all EnviroGRIDS partners. After getting a proper grid certificate and registering to the VO every project member can run their SWAT calculations on the grid.

Along with the SWAT gridification we developed two generic monitoring tools for Ganga and Diane software based on the, newly developed, hBrowse framework that can be setup to monitor any hierarchical data up to four levels deep.

This document highlights the SWAT gridification achievements and tries to serve as a small guide in starting of using grid resources by EnviroGRIDS members.

4 Introduction

4.1 Purpose and scope

The purpose of this document is to provide information about the SWAT gridification including three main use cases, the EnviroGRIDS VO and monitoring tools. This document provides the exact description of the gridification/parallelization process of the SWAT application, used software and the main subjects that was encountered during the process.

The SWAT gridification will provide users with the grid infrastructure to run their simulations, calibrations and uncertainty analysis more effectively.

The document also provides relationship of the SWAT gridification to the EnviroGRIDS system.

4.2 Document structure

- *Chapters 1-3* – Introduction and short summary
- *Chapter 4* – Relation with the Envirogrids system, especially the EnviroGRIDS portal
- *Chapter 5* – Generic explanation of the nature of software gridification
- *Chapter 6* - Explanation of why we need to gridify some tools in the context of enviroGRIDS, and who are the current users, the potential ones in the EG consortium, as well as outside the consortium
- *Chapter 7* – Status of the gridification process, comprehensive explanation of the use cases
- *Chapter 8* – Status of the EnviroGRIDS VO, registration procedure

- *Chapter 9* – Presentation of the new generic jobs monitoring system
- *Chapter 10* – Conclusions and recommendations
- *Chapter 11* – References

5 Relation with EnviroGRIDS system

The gridified SWAT code is in direct relation to the EnviroGRIDS portal. All of the use cases can be combined with the system if needed. Gridified SWAT code is the part of the Geospatial and Grid Services of the Web Portal and will serve users to efficiently develop and run SWAT models.

6 Gridification explanation

A gridification is the process in which the application or process that could run only on a single machine is changed so it can run faster on the grid infrastructure.

6.1 Requirements

- Executable files have to be runnable on the UNIX based operating systems
- We have to be able to split the particular use case (problem we are solving) on the separate, independent parts that can be run on the grid and merge the results at the end.

6.2 In practice

Every application can be run on the grid as long as the algorithm it uses can be parallelized. First step in each gridification process should be (if not already done) to recompile executables to a UNIX executable files. The second step is to create the splitter and merger applications that would split the problem (model) onto separate parts that can be run on a separate nodes and merge the results when grid run completes.

7 The purpose of SWAT gridification

The requirements of the hydrological community are growing very fast. SWAT models become larger and larger so the computing power of a single desktop machine becomes not enough. Single SWAT runs are not a big problem, at the time being, but their calibration is completely different story. A model that takes two hours to simulate, on the single machine, have to be run several hundreds of times to calibrate (for a single calibration run). That's over a month. The SWAT gridification can shorten this time to several hours including grid overhead (waiting for a node, copying and extracting operations).

For the EnviroGRIDS project and for the entire hydrological community gridification will help to achieve ultimate project goals faster and will improve quality of the results.

For the time being, the SWAT gridification proved itself to be very useful in tasks like calibration and uncertainty analysis of a SWAT models. Potential user base for this technology are all users that have anything to do with a SWAT models development and not only inside the EnviroGRIDS project but in entire hydrological, SWAT related, community.

8 Presentation/explanation and status of gridification process

8.1 Short SWAT Introduction

The Soil and Water Assessment Tool is a river basin, or watershed, scale model and was developed to predict the impact of a land management practices on water, sediment and agricultural chemical yields in a large complex watersheds with varying soils, land use and management conditions over long periods of time.

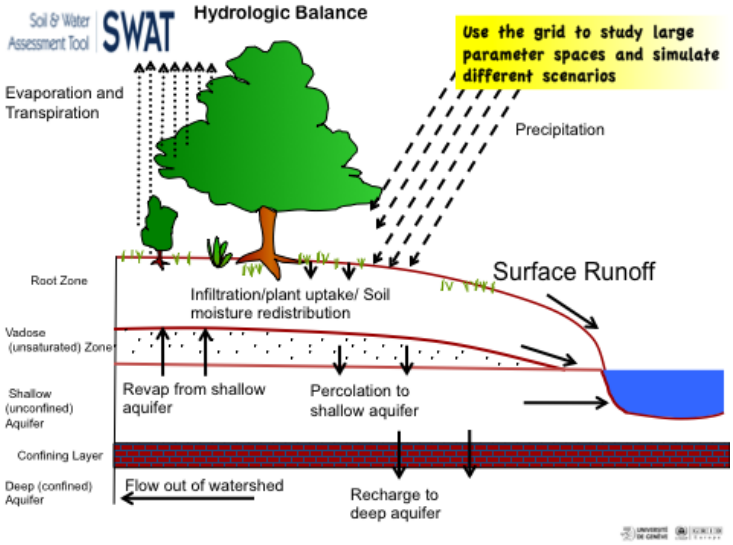

Figure 8-1: Hydrologic balance

- It is widely used around the world by entire hydrology community.
- SWAT and related software is constantly developed in many institutions around the world since the early 1990s
- SWAT can model every part of an hydrological model across the full landscape under study

Every SWAT model can cover everything from a single river fragment with its neighboring land to an entire water system of a continent. Larger models consists of many, so called, sub-basins that are usually single rivers (or river fragments) or lakes with the neighboring land determined by a runoff surfaces and a rivers routing.

8.2 Gridification schemas

The Gridification of SWAT and related software ware focused on searching most usable use cases for grid utilization in SWAT model development workflow. The use cases had to be chosen based on their parallelization potential and usefulness. During research three gridification schemas were developed.

8.2.1 SWAT model gridification

The ‘sub-basin based model splitting followed by routing’ method was developed first. The main idea is to split a model itself based on sub-basins. Pre and post processing of a model is required to complete splitting and merging steps. After splitting, each sub-model can be run separately (in parallel) on the grid. The sub-basin based parallelization is very interesting for large models.

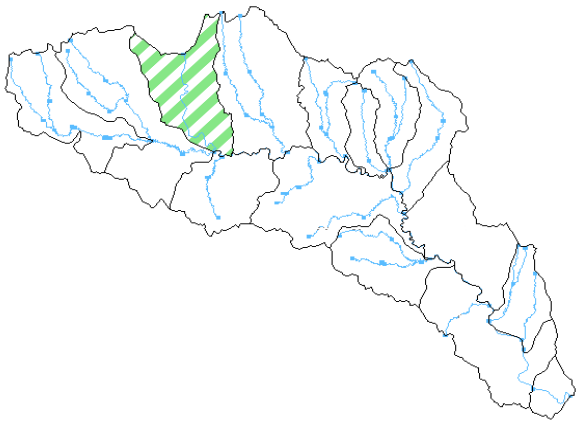

Figure 8-2: Example visualization of the SWAT model consisting of a number of sub-basins, each of these sub-basins can be run on the grid

As was observed, during many test runs, the SWAT model sub-basin based grid executions show some promises but to be fully utilized it requires much bigger jobs.

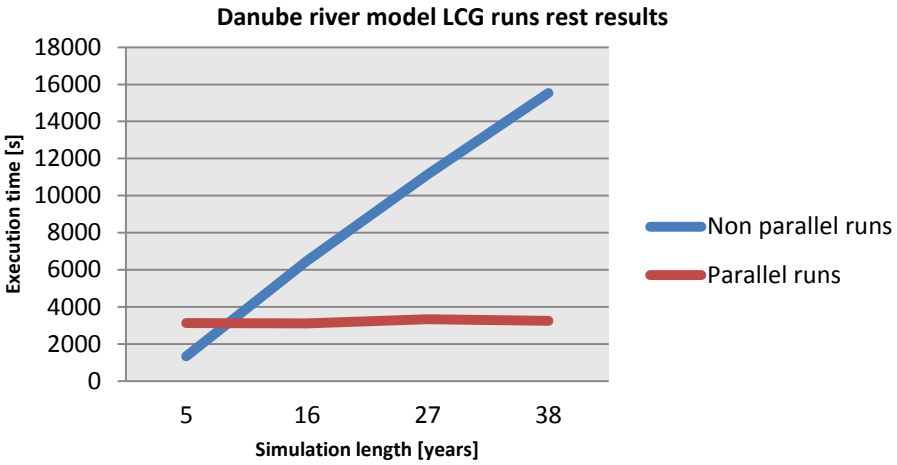

Figure 8-3: 15K HRUs model runs with the different simulation lengths

Ideal situation would be when a single sub-basin execution time takes more than half an hour; in this kind of situations the grid overhead would be negligible. This gridification schema was described in great detail in a recent "DISTRIBUTED COMPUTATION OF LARGE SCALE SWAT MODELS ON THE GRID" paper submitted to the "Environmental Modelling & Software".

8.2.2 LH-OAT algorithm gridification

The LH-OAT is an uncertainty analysis method. In an uncertainty analysis we are sorting out parameters sensitive to selected real world physical process (e.g. nitrogen transport). Then an effective model calibration can be applied. The mechanism behind the LH-OAT method parallel run is very similar to the SWAT-CUP parallelization. We simply run many (usually several hundreds) SWAT runs with a different parameters. The potential performance growth in this gridification schema compared to a non-parallel run is the same as in the, already utilized, SWAT-CUP sufi2 algorithm.

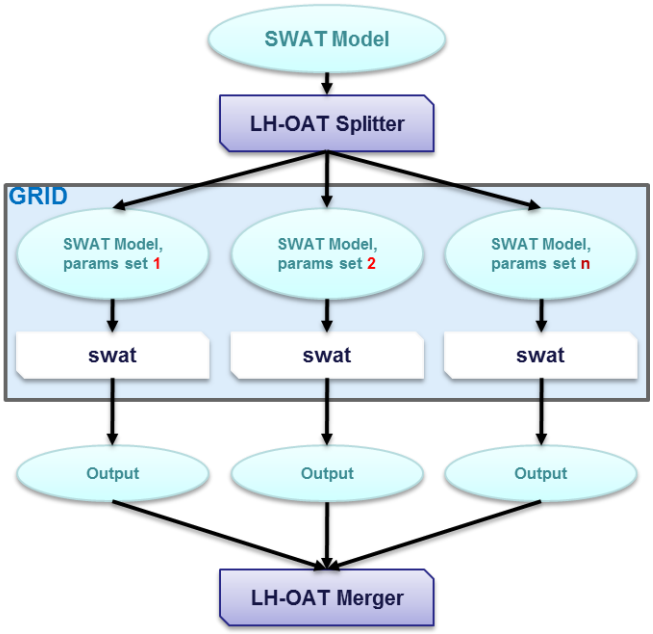

Figure 8-4: The LH-OAT Splitter creates copies of the original model with a different parameter sets, each model copy is run on the grid and the output is merged after calculation completes

8.2.2.1 LH-OAT Algorithm test runs

LH-OAT uncertainty analysis Ganga test run:

- Splitting (parallel execution) provided by Delft
  - Full workload takes about 48 minutes on an average desktop machine
  - Split in ~300 subjobs. Typical execution time for a subjobs ~10-20 s
- We got a workflow which can be split in 309 jobs
  - Average job execution time on a single desktop machine is about 3 s
  - Dominant effects on the execution time (for such short tasks) are:
    - Splitting: 5min (to be verified)
    - Merging time: 7sec
    - Job queuing time (in batch queues)
    - Infrastructure instabilities

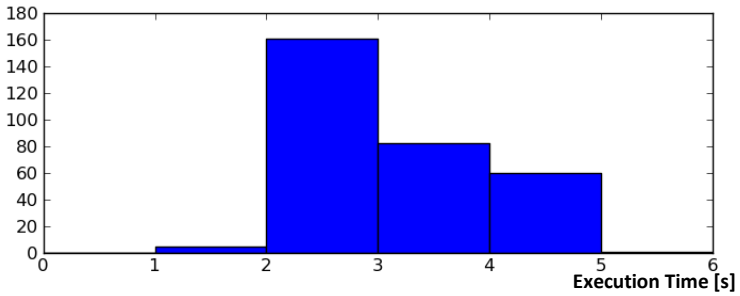

Figure 8-5: Calibration jobs in a function of execution time

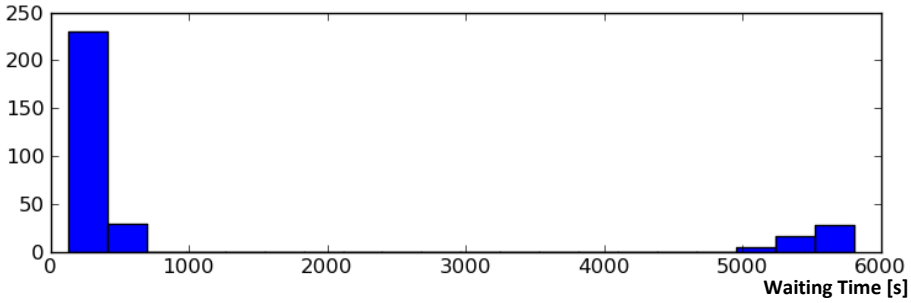

Figure 8-6: Number of jobs in a function of waiting time. Some jobs can be blocked in a queue due to congestion problems

Observations:

- Queuing time can be dominating
  - Execution/waiting time proportion would be better for a longer jobs
- 1 per mille failure observed

LH-OAT uncertainty analysis Diane test run:

- Parallel execution (309 tasks)
  - Splitting and merging steps take about 5min for this model
  - Average job execution time on a single desktop machine is about 3 s
  - Depending on the status of the Grid worker we immediately went down to:
    - Entire parallel execution in this case took about 14 minutes
    - Having more workers ready to accept tasks:
      - Parallel run took 41s
      - Entire parallel execution in this case took about 6min (dominated by the preliminary operations)
  - Dedicated tests to verify reliability
    - No failures observed over 10^4 jobs

Conclusions:

- Diane is very efficient in running large number of job
- In the case when one or more jobs might fail DIANE provides automatic jobs resubmission

8.2.3 SWAT-CUP sufi-2 algorithm gridification

Sufi-2 is the only SWAT-CUP algorithm that can be parallelized. It uses only one copy of a model but, on every simulation, different parameters set are set up. On each node we can run one or more simulations (eg. we can increase the number of simulations per node to reach execution time that is not that dominated by a grid overhead). This parallelization method gives the greatest promises. It's very widely used among community and has already implemented mechanisms to run specific simulation with a defined parameters set. The potential performance gain compared to a non-parallel method is very significant. Giving that calibration wants to run 500 SWAT runs and that we have a big swat model that's execution time would dominate a grid overhead, the run can be close to 500 times faster.

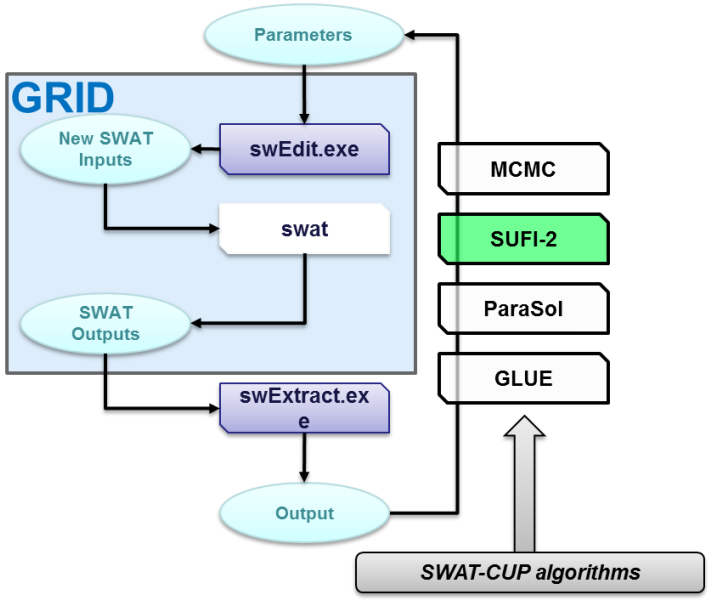

Figure 8-7: Each sufi-2 grid job will take iteration number as an input parameter. Based on the iteration number swEdit.exe script will modify model parameters accordingly. After all parameters being modified SWAT simulation is run and the output is written

Gridification of of SWAT-CUP sufi2 algorithm concludes of:

- Making Swat_Edit.exe file independent from .NET framework using mono library command mkbundle in UNIX environment
- recompiling of SUFI2_execute_L09.exe to run a Swat_Edit.exe
- creating a proper diane run file depending on requirements of a specific implementation

We have "ready to go" example script set, to run the swat sufi2 calibration efficiently from command line. This example can be used as a starting point for integration of the sufi2 algorithm into EnviroGRIDS portal.

8.2.3.1 Sufi-2 algorithm test results

The project, that takes about 60000 seconds to complete on an average desktop machine, was used. In the gridified sufi-2 algorithm there is no splitting stage and in this case there were 500 jobs where each of them took about 2 minutes to complete.

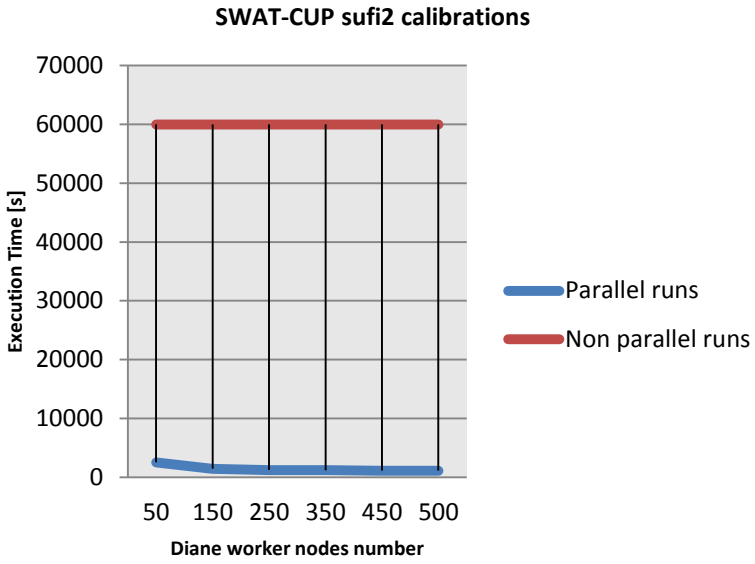

Figure 8-8: Performance improvement when using grid for SWAT model calibration

The tests were run using Diane. We can observe a clear performance gain when using gridified code of the sufi-2 algorithm as calibration proceeded over 50 times faster and even greater increase in performance is expected with a bigger SWAT models. With a bigger model, natural grid overhead and worker nodes submission time will be close to negligible so, with a SWAT model growth, the entire parallel execution time will converge to a single sub-job execution time.

8.3 Software used

For the SWAT jobs submission and management GANGA and DIANE software is used.

8.3.1 Ganga

Ganga is an easy-to-use frontend for job definition and management, implemented in Python. It has been developed to meet the needs of various communities. Ganga allows trivial switching between testing on a local batch system and large-scale processing on Grid resources.

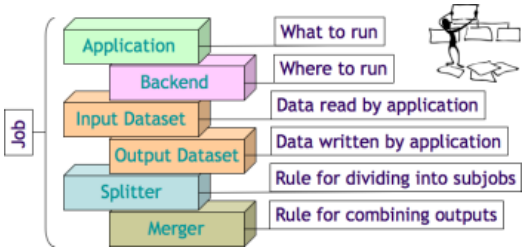

Figure 8-9: Ganga job structure

A job in Ganga is constructed from a set of building blocks. All jobs must specify the software to be run (application) and the processing system (backend) to be used.

8.3.2 Diane

Diane is a job execution control framework for parallel scientific applications. In respect of SWAT, helps with submitting large number of jobs. The automatic control and scheduling of computations on a set of distributed worker nodes leads to an improvement of the quality of service of the EGI/LCG Grid. The tool allows to:

- Reduce the application execution time by using the resources more efficiently,
- Reduce the user work overhead by providing fully automatic execution and failure management,
- Efficiently integrate local and Grid resources.

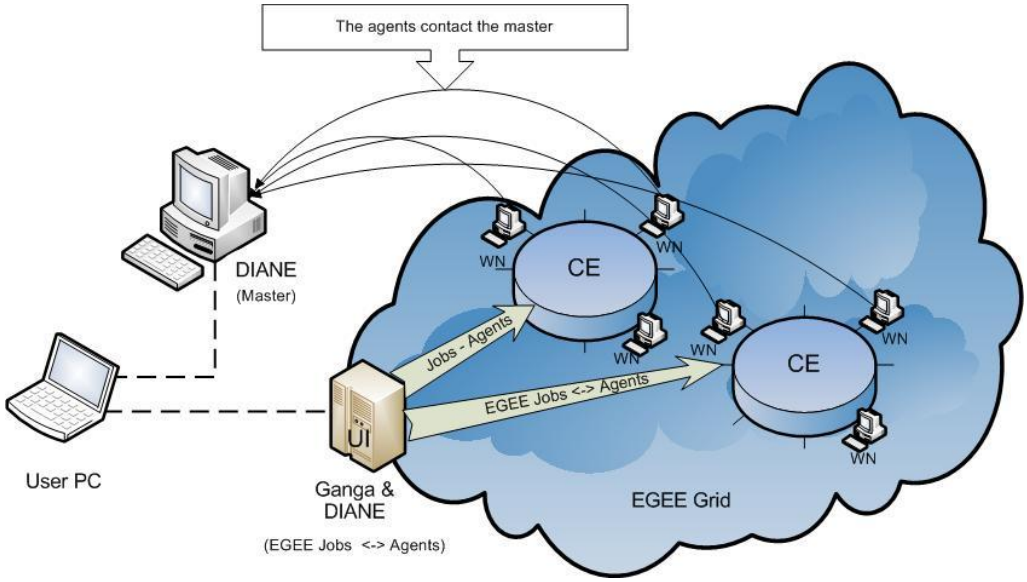

Figure 8-10: Diane

When using Diane we create Diane master process using Diane run file. The run file is defining all of the input files and executable file. After that, Diane workers can be submitted. Diane uses Ganga as submission backend and Diane worker nodes are, in fact, Ganga jobs to which the master can send tasks (in this case SWAT gridification scheme jobs).

9 EnviroGRIDS VO

9.1 Purpose

The EnviroGRIDS VO was created to make grid resources more accessible for a members of the project. It also gives the project more control over use of the resources and provides possibility to easily add new computing sites. From the consortium point of view, creating of the new VO was the only logical choice.

9.2 Registration

Every project member that wants to run his SWAT jobs on the grid needs to register to the EnviroGRIDS VO. The first step is to obtain a proper grid certificate form the specific national Certificate Authority (CA). Here is the list of websites that can help obtaining the certificate:

- http://lcg.web.cern.ch/LCG/digital.htm - CERN website describing the process of obtaining certificates
- http://www.eugridpma.org/members/worldmap - World map of CAs

After getting the grid certificate and installing it on the browser user should register using the VOs registration page:

- https://lcg-voms.cern.ch:8443/vo/envirogrids.vo.eu-egee.org/vomrs

Registration procedure takes few minutes and requires the following information:

- **DN** – Title of members certificate
- **CN** – Issuer of members certificate
- Email Address
- Phone number

After registration, the member needs to wait for the VO manager approval.

9.3 VO information

The most important information about EnviroGRIDS VO can be found on the VOs CIC Identity Card available under the following URL (website also requires grid certificate):

- https://cic.egi.eu/index.php?section=vo&vo=envirogrids.vo.eu-egee.org

To configure software to use the EnviroGRIDS VO, information from this website are needed:

- https://voms.cern.ch:8443/voms/envirogrids.vo.eu-egee.org/configuration/configuration.action

10 Jobs monitoring tools

Along with the SWAT use cases gridification the generic, easy to use and highly customizable job monitoring framework was developed. It's called hBrowse framework.

The hBrowse Framework (www.hbrowse.net) is a new kind of generic open source monitoring application. It's the html/javascript client that can be combined with any kind of server as long as it can send json formatted data. The whole application can be setup using just one settings file.

It was designed to meet the needs of various communities connected to a grid computing. It is strongly configurable and easy to adjust and implement accordingly to a specific community needs. Each part of this software (dynamic tables, user selection etc.) is in fact a separate plugin which can be used separately from the main application. It was especially designed to meet the requirements of Atlas and CMS users as well as to use it as a bulked Ganga monitoring tool.

The hBrowse framework was also used to create 2 monitoring tools for Ganga and Diane:

- http://gangamon.cern.ch/ganga
- http://dianemon.cern.ch/diane

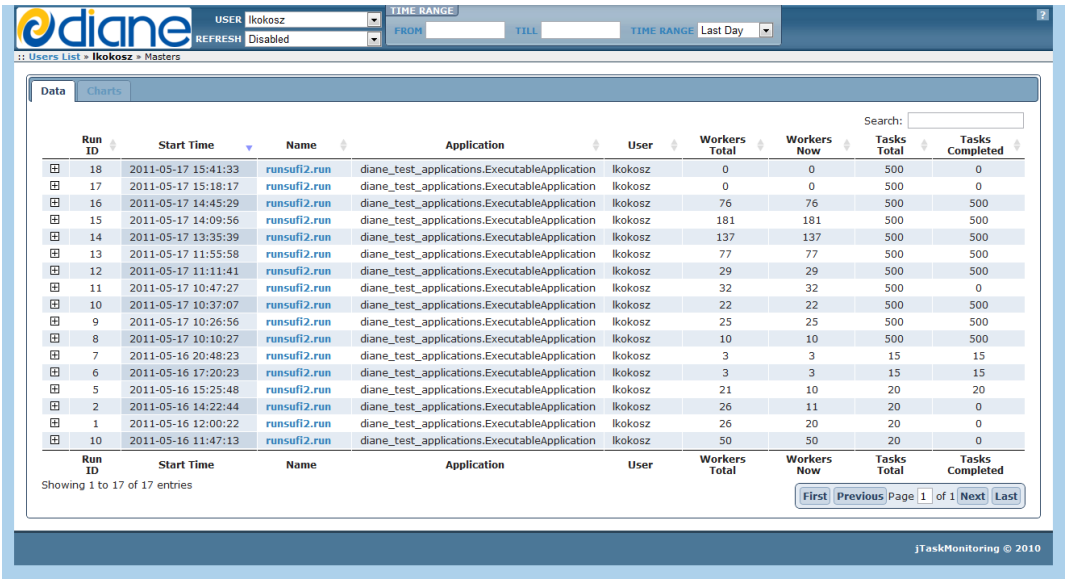

Figure 10-1: Diane monitoring tool

Users get this tools completely free when using Ganga and Diane all that needs to be done to activate them is to properly setup the Ganga configuration file (.gangarc)

- https://twiki.cern.ch/twiki/bin/view/ArdaGrid/GangaDIANEOperationsProcedures#Instructions_for_users

The hBrowse project was started as a small monitoring application only for Ganga but quickly evolved to the generic tool that can be used by various communities to create their own monitoring systems.

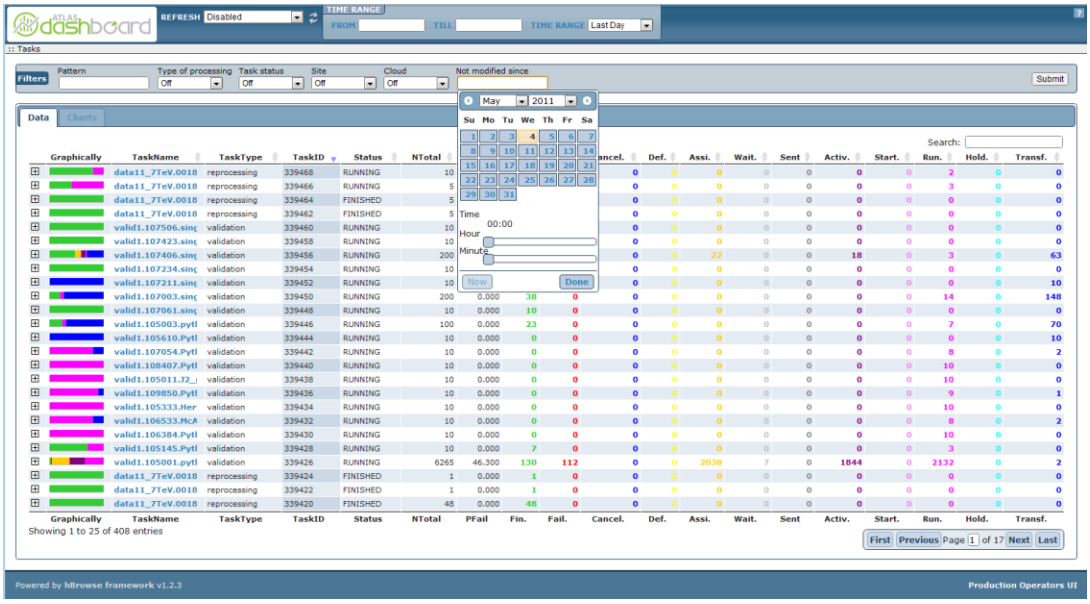

Figure 10-2: Atlas (LHC experiment) production operators UI monitoring tool

10.1 Architecture

The hBrowse framework uses many common technologies that make it very easy to modify by a developers that are not involved in the original project. It uses jQuery and many of its plugins. The main jQuery plugin component used by the application is Datatables which handles tables rendering and interactivity

- Framework is simply a JavaScript client
- Different server layers can be applied (server layer is not part of the system)
- Server layer serves only as a data source.
- Client and server communicates using JSON data transfer format
- JSON in this case is a perfect solution because of its syntax:
    - in most cases it is the same as in JavaScript and Python
- The format of the JSON server output is not important because it can be easily translated to the data format expected by the client application

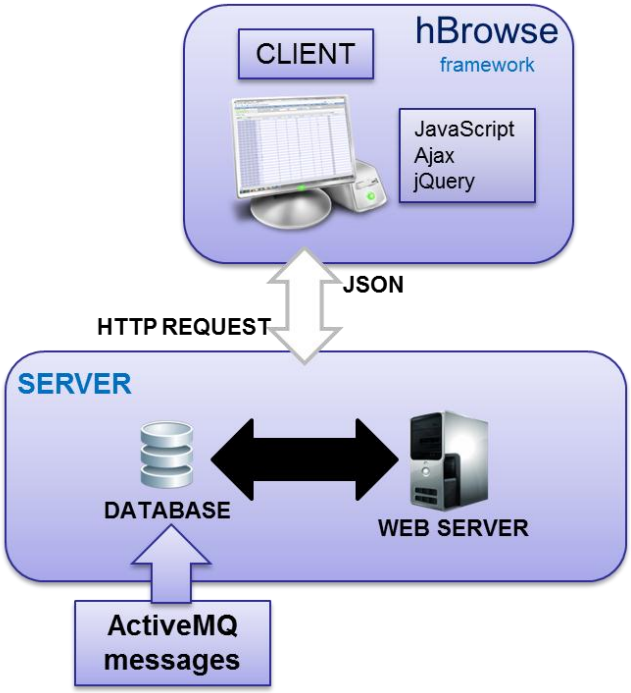

Figure 10-3: Usual system architecture based on hBrowse framework

Application works in a loop of mutual dependency between 3 main aspects:

- Bookmarking URLs
- Data model
- View

Basically when we want to change the content of the page we will first modify the URL. The change like this will be noticed by an application and, based on the URL, internal data model will be updated. At the end, view will be generated based on the data stored in a model:

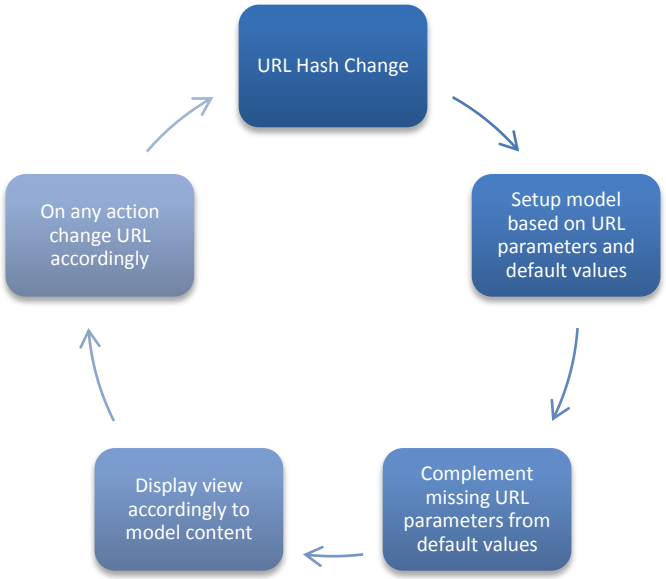

Figure 10-4: hBrowse working loop

The application data are stored inside the Data model which is defined in data.js file. Settings let you access the Data model in various places so it's important to understand its structure. Data stored inside the Data model are split onto 2 parts, operating data and cached data. Operating data is used to display proper information on the screen and they consists information like a user name, time periods, interface settings etc. The cached data are used in various situations like charts drawing or additional requests etc.

The application consists of four main modules that are used in presentation layer

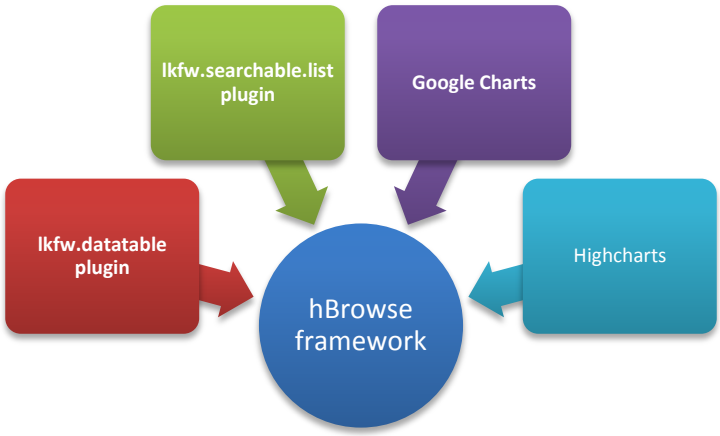

Figure 10-5: hBrowse main system components

- **lkfw.datatable plugin** - jQuery datatables plugin overlay that expands its original functionality.
- **lkfw.searchable.list plugin** - jQuery livesearch plugin overlay fitted to the needs of the monitoring application.
- **Google Charts** - Framerowk uses google chart service to draw data plots.
- **Highcharts** - It's a second library used by the hBrowse framework to draw charts, it brings a lot of interactivity level compared to the Google Charts

10.2 Functionalities

These are the main features of the hBrowse framework:

- Highly configurable environment
  - Almost every aspect of the application can be setup inside the settings file including table rows, information that is showed inside expanded rows, charts, filters etc.
- Bookmarking support
- History support
- Easy search
- Google and Highcharts charts support
- Fully configured data filters
- Independent from the server side part of the application
- 3 main views (users selection, tables level one and table level 2)
- Data caching

11 Conclusions and recommendations

The SWAT gridification process was proceeding quite smoothly from the beginning. Thanks to the Ganga and Diane software that handles jobs submission, data transfers and error handling the main effort could be put on the SWAT use cases research.

As was mentioned before, three gridification schemas were taken into account giving the greatest chances of actually being useful for the community. First of them, the SWAT model gridification is quite interesting for a very large models (large in sense of calculation time). The point here is to have a model that's calculation time is much greater than grid overhead (time necessary for submission, file transfers, preparation of output etc.).

The LH-OAT algorithm gridification gives some promises and, after additional optimizations by interested community, can be as useful as the gridification of SWAT-CUP sufi2. It wasn't developed pass the testing stage due to lack of interest from the EnviroGRIDS community.

The SWAT-CUP sufi2 algorithm gridification is a great success of this deliverable. It was properly tested and gives great possibilities in a SWAT models development. This gridification schema is currently used in the gSWAT Calibration application developed by our partners from Technical University of Cluj-Napoca.

The EnviroGRIDS VO is up and running from autumn 2010 and right now have eight members. Giving that interest in the grid SWAT calculations are growing, we expect many more in the coming months.

The great success of the SWAT gridification was creation of a generic monitoring tool for various communities. Right now, entire Ganga and Diane users community is using tools created using the hBrowse application and a many monitoring applications based on this framework is being developed at the moment.

11.1 Recommendations

To achieve a long term support for the SWAT application running on the grid infrastructure it is absolutely necessary for the SWAT developers to officially support UNIX versions of SWAT executable file.

12 References

DataTables (table plug-in for jQuery) [Online]. - www.datatables.net.

DIANE: Distributed Analysis Environment [Online]. - http://it-proj-diane.web.cern.ch/it-proj-diane.

DISTRIBUTED COMPUTATION OF LARGE SCALE SWAT MODELS ON THE GRID [Journal] / auth. Seleshi G Yalew, PhD Researcher; Ann van Griensven, PhD, Associate Professor;Nicolas Ray; Getnet D Betrie; Lukasz Kokoszkiewicz;. - 2010.

Ganga: Gaudi/Athena and Grid Alliance [Online]. - http://ganga.web.cern.ch/ganga.

hBrowse - LHC Grid Job/Task monitoring generic application [Online]. - hbrowse.net.

Highcharts - Interactive JavaScript charts for your webpage [Online]. - www.highcharts.com.

jQuery: The Write Less, Do More, JavaScript Library [Online]. - jquery.com.

SWAT: Soil and Water Assessment Tool [Online]. - http://swatmodel.tamu.edu/.