An OGC-WS Framework to Run FireStation on the Grid

António Pina[1], Bruno Oliveira[1], Joel Puga[1], Ricardo Marques[1], Alberto Proença[1]

Departamento de Informática, Universidade do Minho
{pina,boliveira,joel,rjm,aproenca}@di.uminho.pt

Abstract. The CROSS-Fire project aims to develop a grid-based framework as a risk management decision support system for the civil protection authorities, using forest fires as the main case study and FireStation as the standalone CAD application that simulates the fire spread over complex topography. The CROSS-Fire initial tasks focussed on the development of a parallel version of the fire simulator engine, P-FireStation, and its porting into the EGEE grid environment, G-FireStation. The main contribution of this communication lies on the definition of an OGC-WS framework to enable a basic set of standard geospatial services that will allow Firestation to be interoperable with standard-based Spatial Data Infrastructures.

The P-FireStation version explores the inherent parallel environment offered by clusters at each site of the EGEE grid, to support larger data sets and to improve the accuracy of the predictions. This parallel version relies on the MPI protocol and supports larger data sets taking advantage of the MPI parallel I/O facilities. The G-FireStation version integrates EGEE grid facilities, namely the gLite data management services and tools to access data, the AMGA gLite grid metadata catalogue to manage the simulation I/O data, and the WatchDog tool to monitor and provide data for the interactive control of simulations.

CROSS-Fire decision support system will be implemented as a web portal that presents at its core a layer of OGC protocols to access, add and process data: WCS for interoperable access to geospatial coverages, WFS for retrieving geographic features and WPS for geospatial processing services and control of the execution of simulations. The portal is also a connection point for desktop clients with GIS and CAD capabilities.

The approach will make G-FireStation more interoperable, allowing to access different data providers and publishing output data for further processing, following the guidelines of the EC CYCLOPS project.

Keywords: Civil Protection, Fire Simulation, OGC-Web Services, EGEE

1 Introduction

Current geo-referenced computing and communication technologies are becoming robust and mature to be accepted as decision aid tools by the Civil Protection (CP) authorities in emergency calls. The CP authorities need fast and reliable tools to remotely access real time geo-referenced data and to estimate the risk spread well in advance, in order to lead the operational procedures with the active involvement of public and private organizations.

A forest fire is a typical example of a critical environmental disaster in Mediterranean countries, whose efficient fight requires the best use of the available computational and communication power. FireStation is a standalone desktop simulator of the fire growth over complex topography, developed at University of Coimbra, that has been tested in the past by Portuguese fire brigades [1].

CROSS-Fire is a research project, funded by the Portuguese National Grid Initiative (NGI) and led by University of Minho in cooperation with the FireStation original team at University of Coimbra and the Portuguese CP Authority [2]. It aims to develop a grid-based framework as a risk management decision support system for the CP authorities, using forest fires as the main case study. Its initial tasks include the porting of the FireStation (FS) simulator to a computing cluster environment (P-FireStation, P-FS) and then to the grid (G-FireStation, G-FS). The work carried out may also be seen as another effort to prove that CP applications can (i) benefit from the underlying grid infrastructure to a faster access to improved risk/disaster predictions and that (ii) the adoption of a standard geospatial layer makes CP applications more interoperable, providing access to different data providers and publishing output data for further processing.

The Open Geospatial Consortium Web Services (OGC-WS, [3]) are being adopted worldwide, as the technology to support the development of complex distributed applications over grid platforms, and to deal with data from many different sources, including live sensors. OGC services provide the conceptual tools for modelling the geospatial information that is shared in CP applications during its entire lifecycle. European and international projects and initiatives are designing, developing, implementing, and using Spatial Data Infrastructure (SDI) as information infrastructures oriented to geographic data. The Infrastructure for Spatial Information in Europe (INSPIRE, [4]) is an EC initiative to build an European SDI beyond national boundaries that aims to provide relevant, harmonized and quality geographic information, to handle issues concerning data formats, metadata, and network services to enable a basic set of standard geospatial services.

The EC FP6 CYCLOPS project (Cyber Infrastructure for Civil protection Operative Procedures, [5]) adopted the EGEE grid platform [6] to support the CP operative procedures aiming to refine EGEE standards and protocols within CP specific requirements. It has proposed an architectural framework built on top of the EGEE middleware gLite [7], to make use of the grid processing and storage capabilities, which include a set of geospatial services accessible through standard interface services. The approach has been tested and validated through the implementation of a grid-enabled prototype application, G-RISICO, for wild fire risk assessment, used by the CP authorities in Italy.

This communication presents an OGC-WS framework to run FS on the grid. Next section gives an overview of the desktop version of the FS simulator. Third and fourth sections describe the proposed parallel model and implementation of P-FireStation (P-FS), followed by the devised approach to the G-FireStation (G-FS) gridification. The fifth section presents the devised OGC-WS Framework that will support the G-FS simulation environment. This communication closes with a discussion of the overall work and suggestions for future work.

2 FireStation Approach

The FS is an integrated system aimed at the simulation of fire growth over a complex topography [1]. It has been used by Portuguese fire brigades under the CP authorities in the past decade. The original standalone FS version integrated a module for wind field generation (following a linear model called NUATMOS), as well as a module for the computation of the Fire Weather Index, from the Canadian system. The software was originally written in MDL, a specific C language of Microstation with built-in subroutines for the design of window-based interfaces and generation of visualization elements both in 2D and 3D space. For the CROSS-Fire project, the frontend modules were separated from the backend ones, and a more accurate wind field estimator (using a Navier-Stokes solver called CANYON) was added. The main efforts in this project were to parallelize these FS backend modules and to port the resulting P-FS to the grid.

FS requires three different types of input data to simulate fire propagation: (i) a description of the terrain features (e.g., morphology, land occupation), (ii) meteorological data, namely the wind conditions affecting the selected region, and (ii) initial conditions and some control parameters (see figure 1), [1].

The terrain information follows the BEHAVE model [8] that divides the terrain into cells, over which its properties are assumed to be constant. The description of the topography and fuel characteristics of the terrain is registered in two files that store the altitude and the fuel type for each cell. The information about the wind conditions, at mid-flame high, affecting the terrain, is previously produced by the wind field generation module. Finally, some control data such as the ignition points and the stopping control parameters criteria have to be passed to the application. The stopping criteria can be either the maximum simulation time, the maximum burnt area or the maximum number of burnt cells.

2.1 Fire Propagation Model

The fire growth simulation is a process of contagion between burning and non-burning cells, carried out in the following way: at a generic time instant, the time the fire takes to propagate from each burning cell to its (non-burning) neighbours is computed. This calculus is based on the cells' wind information, slope and fuel characteristics. Each non-burning neighbour cell is thus assigned a number which represents the shortest time instant at which ignition will take place as a result of contagion from the already burning cells. In the next iteration, the non-burning cell with the shortest time assigned becomes a burning cell, and the propagation process is repeated. This model, based on a Dijkstra's dynamic programming algorithm, leads to a time progression which may not be constant, i.e., time may step non-uniformly, following the contagion process. For the present work, it is interesting to look inside the core of the application: the fire propagation algorithm, shown in algorithm 1.1.

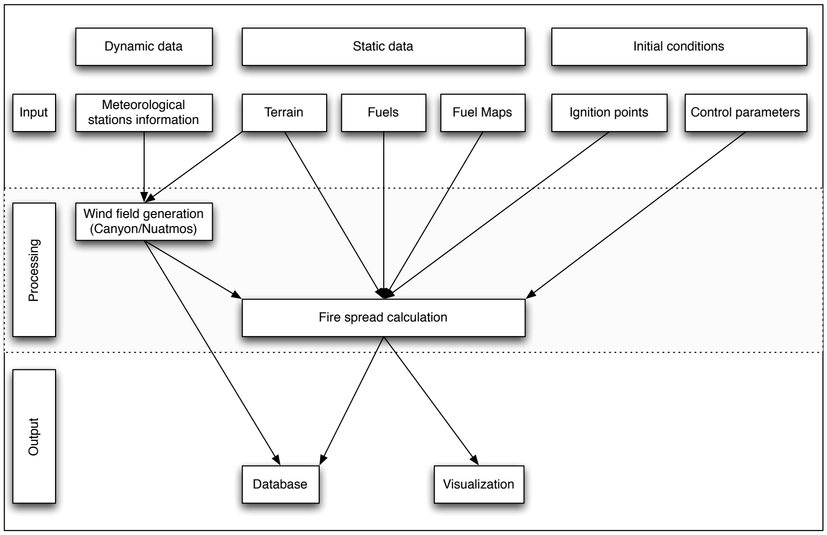

Fig. 1. Components of FireStation simulation model

Listing 1.1. Fire propagation algorithm

```
while(ExistPredictions && !StoppingCriteriaMatched) {
/* 1. Fetches the cell set to burn in the shortest period of time */
  C = GetNextBurningCell();
/* 2. The present time value is set to the ignition time of cell C */
  PresentTime = GetBurningPrediction(C);
/* 3. Cell C is burned */
  BurnCell(C);
/* 4. Refreshes the number of cells burned and the total burned area: */
  TotalBurnedCells = TotalBurnedCells + 1;
  BurnedArea = TotalBurnedCells * CellArea;
/* 5. Computes the fire propagation from C to its neighbour cells in the
    form of predictions
      and incorporates the cells and the predictions in the list of
          predictions: */
  ComputePropagation(C);
/* 6. Makes a set of tests to decide whether the simulation will be
    stopped or not */
  if(Any Stopping Criteria is Verified)
      { StoppingCriteriaMatched = True; } (End If)
} (End while)
```

The fire propagation algorithm consists of a cycle which executes while there are still cells marked to burn and none of the stopping criteria is verified. If there are no cells marked to burn, then it means that there is no fire in the area of calculus, so the simulation can terminate. If any of the stopping criteria is reached, then the simulation also stops. The output of the fire growth simulation is a list of the burnt cells, and its respective ignition time. This output can be post-processed to obtain other type of results such as the fire spread rate and direction, the burnt area and its perimeter, and the linear intensity of the fire front.

3 P-FireStation Model

The goal of the parallelization phase was to produce a prototype P-FS which can take advantage of the computational power available in a cluster environment, by using multiple nodes to simulate the fire growth in parallel. The starting point was a C version of FS containing only the fire propagation module.

A domain decomposition parallelization strategy was applied to assign a subset of cells, obtained by partitioning the whole region to each processor used in the simulation. Each processor would then compute the fire growth in its own subset. The eventual fire propagation to a neighbour subset has to be communicated to the processor in charge of it, so that the neighbour processor is aware that the fire is burning in its terrain, and can simulate the fire spread from that point on (see figure 2). When all the processors terminate the simulation, the results should be combined.

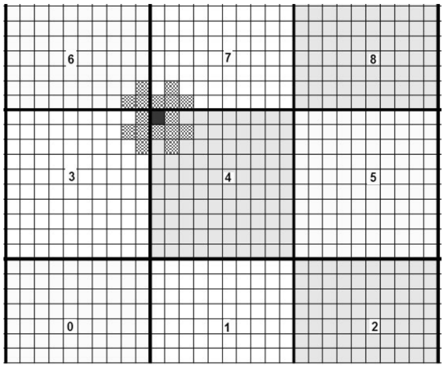

Fig. 2. Fire propagation to neighbours

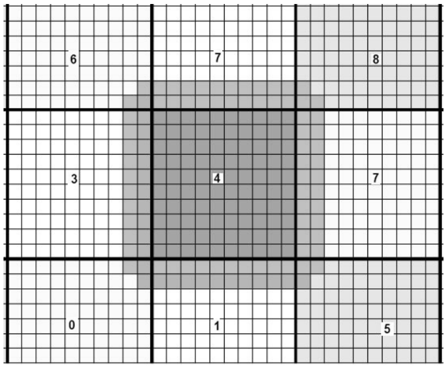

Fig. 3. Ghost area of a sub-terrain

To compute the fire spread in its own sub-terrain, each processor needs more information than just the one corresponding to its sub-terrain. When a processor is computing the fire spread from a burning cell (the darkest cell in the top-left corner of processor 4 in figure 2) it has to make predictions for its 16 neighbour cells (checkboard pattern). However, because some of these cells do not belong to its sub-terrain, each processor needs to have not only information about its local cells, but also about all the cells from the neighbours processores for which it can possibly make predictions.

One of the improvements to the desktop version of FS is the ability to visualize the fire spreading into the newly burnt cells, as soon as these are computed. To support this type of monitoring, each processor should communicate to an external application the cells marked as burnt at the end of each iteration, without waiting by the end of the simulation.

P-FS implementation was based on the design of an almost non-intrusive layer of MPI functions, tags and derived data types that leaves most of the original sequential code unchanged. The input/output data stages were implemented in MPI-IO, using local arrays with ghost area [9], due to the potentially large data files involved in the simulation.

4 G-FireStation

The G-FS version was developed aiming to exploit some grid capabilities, namely: (i) to get updated data from other organizations (such as meteorological data, ”coverages”, reported burnt regions), (ii) to manage large data input/output files, (iii) to create a large database of simulation results and (iv) to allow faster than real-time monitoring of the fire growth. The first goal has not been addressed yet.

To achieve the other goals, the parallel version was enhanced to integrate a set of EGEE grid facilities, required by this type of application, including the use of (i) gLite for job description, job submission, output retrieval and data management, (ii) data management services and tools to ease the location, access and transfer of data, (iii) the AMGA [10] metadata catalogue to manage the simulations input/output data and (iv) the WatchDog tool [11], to monitor and to provide data for the interactive control of simulations.

The G-FS job description language showed in algorithm 1.3 describes the main tasks that are executed by the G-FS simulation:

1. Download the large data files from a grid storage element;
2. Start the job monitoring tool (WatchDog);
3. Execute the P-FS, using WatchDog to report the fire growth evolution;
4. Register the results in grid storage elements after termination;
5. Add an entry with the information about this execution in AMGA database.

The job description file makes use of the mpi-start mechanism, which explains why the executable is the wrapper script, and not the P-FS binary file. The wrapper and the hook files are included in the input sandbox that also carries: the parametric data for the simulation (*ignition.dat*, *control.dat* and *fuelmodels.dat*); scripts for downloading/uploading data from/to the storage element, respectively (*fireStation-download-input.sh* and *fireStation-upload-output.sh*); a script to add an entry of the execution in the executions' metadata database (*synchronize-amga-se.sh*); and the WatchDog scripts to monitor the simulation.

Listing 1.2. G-FS job description file

```
Type = ‘‘Job’’;
JobType = Normal;
CPUNumber = 12;
Executable = fireStation-start-wrapper.sh;
Arguments = fireStation MPICH;
StdOutput = std.out;
StdError = std.err;
InputSandbox= {fireStation-start-wrapper.sh, fireStation, ignition.dat,
    Control.dat, fuelmodels.fls, fireStation-hooks.sh, fireStationdownload
    -input.sh, fireStation-upload-output.sh, synchronize-amga-se.sh };
OutputSandbox = { std.err,   std.out };
Requirements = Member(MPI-START,  other.
    GlueHostApplicationSoftwareRunTimeEnvironment)
&& Member(MPICH,  other.GlueHostApplicationSoftwareRunTimeEnvironment)
&& (!RegExp(ce.cp.di.uminho.pt,other.GlueCEUniqueID));
```

This example job will run in 12 nodes with the mpi-start mechanism enabled and follows the specific requirements.

The hooks file includes the *pre_run* and *post_run* functions. During the pre-run phase, input data are downloaded from the storage elements into the working nodes, the WatchDog is started, and the P-FS binary file is given permissions to execute. During the *post_run* phase the WatchDog is turned off, the result is uploaded to the storage elements and a new entry is added to the AMGA database.

Listing 1.3. G-FS pre- and post-processing phases

```
# This function will be called before the MPI executable is started.
        pre_run_hook () {
        echo Executing pre hook.
# Get the Input Files from the Storage Element
        chmod 777 fireStation-download-input.sh
        . fireStation-download-input.sh
# Start WatchDog
        ./watchdog.ctrl start
# Give the application permitions to execute
        chmod 777 fireStation
        echo Finished the pre hook.
        return 0 }
# This function will be called before the MPI executable is finished.
        post_run_hook () {
        echo Executing post hook.
# Stop WatchDog
        ./watchdog.ctrl stop
# Uploading the output form the CE to the SE
        chmod 77 fireStation-upload-output.sh
        . fireStation-upload-output.sh
# Synchronising Amga database with the SE
        chmod 777 synchronize-amga-se.sh
        . synchronize-amga-se.sh jobid.amga
        echo Finished the post hook.
        return 0 }
```

5 OGC-WS Framework

The goal of the CROSS-Fire project is not only to build a grid-oriented software platform to provide faster computation of the natural hazard spread, but also to deliver a complete decision-support system, that may be used by CP authorities to take actions based on the results of the simulations, or evaluate different scenarios by exploiting parameters variations. It will also be of great value to create and manage a database of past simulations.

The central piece of the decision-support system is a web portal where many players can connect (see figure 4). CP authorities and other entities (e.g., the City Council officers) may access the portal to request new simulations, or to update information concerning the input data required to estimate the risk of the natural hazard, such as fuel distribution and terrain maps. They can also consult the portal to access the past simulations to validate the predictions with actual field data. Scientists and the academic community can also use the portal to request simulations, in periods with no natural hazards activity, supporting their experiences with live data and the computational power delivered by the grid.

Within pre-defined limits, reporters and citizens can also view the state of a simulation, and can even update the information required for the models under

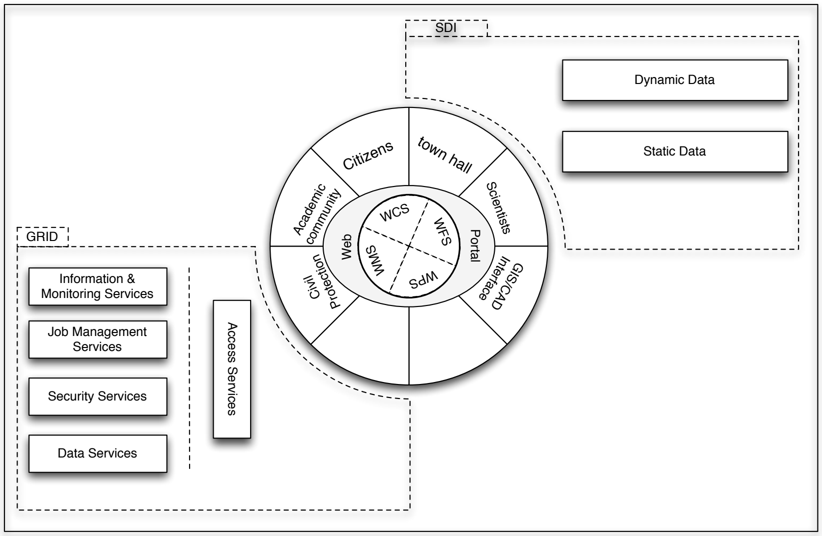

Fig. 4. Decision-support system architecture

well defined conditions. Other players can later connect to the portal, on a need basis, allowing a continuous evolution of the system.

The portal presents many layers. At its core, the portal has a layer of OGC protocols to access and add data, and control the execution of simulations. The main point of interaction with the portal will be a WPS service. WPS is an OGC standard used to make consistent inputs and outputs for geospatial processing services. In this model, WPS will provide a mechanism to access the grid facilities for processing and data management, and to request geospatial information, from an SDI, required to estimate the natural hazard spread.

The portal is also the connection point for desktop clients with GIS and CAD capabilities. Desktop clients offer a richer interactive environment benefiting at the same time of all services provided by the portal.

5.1 Process Flow

Figure 5 shows an application process flow from the operator's point of view.

The operator of the GIS/CAD interface starts by selecting the appropriate information to run the simulation, either by querying an SDI or by issuing a request for data to the portal's WPS, that will then contact the SDI to get the required data (1,2). This information concerns both dynamic and static data, namely information regarding wind stations, terrain, fuel and fuel models. WPS has the capability to convert the required information into an appropriate format, to conform to the target application specification.

Once all required input data are available, including the control parameters – ignition points and termination mode – a processing request is sent to the portal's

WPS (3). According to the parameters in the request, the WPS will choose the execution model for the simulation. After launching the correspondent jobs on the Grid it creates a new simulation entry in the metadata database (4, 4’, 4”).

The FS version supplied to the CROSS-Fire project includes two different execution models for the calculation of wind field: the linear model NUATMOS and the Navier-Stokes solver CANYON. The complexity of the CANYON mathematical model requires more computational resources, making it a very good candidate for parallelization to reduce the overall simulation time.

The architecture framework that is presented here supports other application and execution models for natural risk assessment and management, such as flash floods, volcanic eruptions, tsunamis, or even other fire spread models.

The jobs responsible for the generation of the wind field and for the fire spread calculation may use standard grid facilities, such as the data management LFC tools, to preserve and exchange information while the simulation evolves. Another important player is the WatchDog tool that may be used to continuously monitor output data from the simulation and copy them into the appropriate, pre-defined, storage element. Cumulatively, as the execution proceeds, new entries can be inserted into the AMGA metadata database (5,6). This type of logging mechanism may later be used for post-mortem execution, or, for instance, to continue a new simulation from any point in time of a previous one.

The portal’s WPS once notified of the existence of new data output from the simulation (7) fetches it directly from the AMGA database server (8) and returns it to the GIS/CAD interface(9).

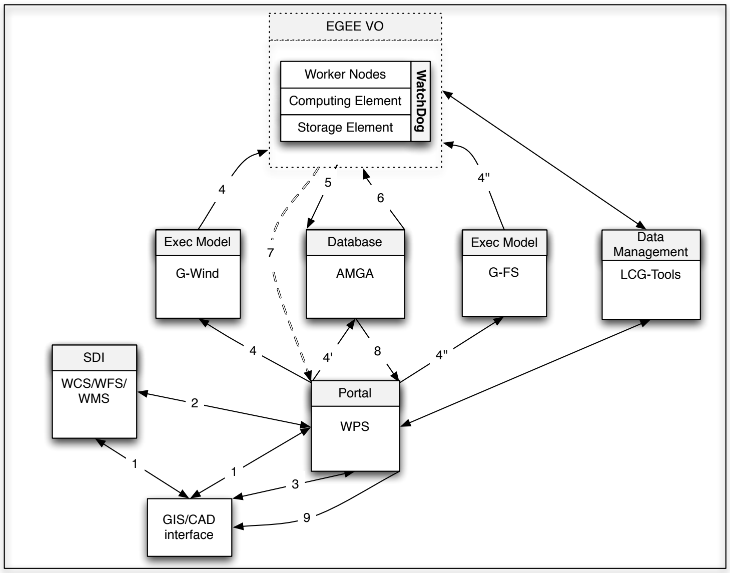

Fig. 5. The process flow

The data management tools are used whenever necessary to move data from or into the grid.

6 Conclusions

Both parallel P-FS and grid G-FS versions of FireStation were successfully implemented, confirming that the grid can provide a fast and reliable response to emergency situations, with large input/output data demands, with support for soft real-time monitoring, and keeping track of previous simulations through a grid oriented metadata database.

This development work was linked to the G-RISICO approach followed in the CYCLOPS project, which assesses the wild land fire risk over the Italian territory. However, while G-FS considers and estimates the fire spread over the cell borders on partitioned regions, G-RISICO does not consider interactions between adjacent partitioned regions. Current G-FS version already integrates several EGEE grid facilities and services, such as the data management services and tools to locate, access and transfer data, the AMGA metadata catalogue to manage the simulations I/O data, and the WatchDog tool to monitor and to control the interactive execution of simulations.

Although the CROSS-Fire project was designed aiming the Portuguese territory, as an extension to previous collaboration with the national CP authority in the control of forest fires, it is currently being extended to Europe and Latin America, through international cooperation within the CYCLOPS consortium and as a partner in the E-science grid facility for Europe and Latin America (EELA-2). These extensions will allow the study and investigation of the requirements to run CP applications in a larger grid infrastructure that spawns continents, and the evaluation of the inter-operability between different/complementary grid infrastructures.

Later versions of G-FS will hide grid complexity and offer to a non-specialist a standard way to exploit and enable geospatial services for data access and processing, such as OGC WCS, WFS and WPS. Grid capabilities will be further exploited to provide faster multi-simulation execution, to manage large input/output data files, to create a database of simulation results and to support interactive control of the simulations through a richer graphical user interface.

The planned G-FS user interface that is currently being evaluated is the gvSIG, a multilingual, open source GIS editing tool, developed by the Valencian Community (Spain) and co-funded by the EC FEDER program, that can handle both vector and raster data, and with a solid user base. The gvSIG conforms to the INSPIRE framework requirements for managing geospatial information, supports multiple file formats, DBMS and web protocols, and includes an SDI client for standard connection to OGC services, accessing data and being able to overlap and merge in gvSIG map views. A discovery service client is also provided within gvSIG, which can be used to locate data resources within an SDI.

Acknowledgements

This work is supported by the FCT grant GRID/GRI/81795/2006 awarded to the CROSS-Fire project, under the Portuguese National Grid Initiative.

Authors also acknowledge the First EELA Grid School contribution, whose programme, based both on lectures and hands-on practices, covered in-depth the main topics of Grid technologies needed to create the first G-FS prototype.

References

1. AMG Lopes, MG Cruz, and DX Viegas. FireStation - an integrated software system for the numerical simulation of re spread on complex topography. Environmental Modelling and Software, 17(3):269285, 2002.
2. CROSS-Fire Collaborative Resources Online to Support Simulations on Forest Fires : a Grid Platform to Integrate Geo-referenced Web Services for Real-Time Management, GRID/GRI/81795/2006.
3. Open Geospation Consortium http://www.opengeospatial.org/
4. INSPIRE http://inspire.jrc.ec.europa.eu/
5. CYCLOPS Technical Annex, Cyber-Infrastructure for Civil Protection Operative Procedures (CYCLOPS), SSA project N. 031874, March 2006.
6. Enabling Grids for E-sciencE (EGEE) http://www.eu-egee.org/
7. Stephen Burke, Simone Campana, Patricia Méndez Lorenzo, Christopher Nater, Roberto Santinelli, Andrea Sciabà, gLite 3.1 User Guide, Manuals Series
8. Patricia L. Andrews. BEHAVE: fire behavior prediction and fuel modeling system-BURN Subsystem, part 1. General Technical Report INT-194. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Research Station. 130 p., 1986
9. Using MPI-2: Advanced Features of the Message Passing Interface W Gropp, R Thakur, E Lusk - 1999 - MIT Press Cambridge, MA, USA
10. Nuno Santos and Birger Koblitz, Metadata services on the Grid, Proceedings of the X International Workshop on Advanced Computing and Analysis Techniques in Physics Research - ACAT 05
11. Riccardo Bruno - INFN Catania, A watchdog utility to keep track of job execution, https://grid.ct.infn.it/twiki/bin/view/PI2S2/WatchdogUtility