A worldwide Rainfall Hydrologic Analysis Through Grid Computing

O. Terzo, L. Mossucca
Istituto Superiore Mario Boella
Via P. C. Boggio 61, Torino, Italy
Email: {terzo, mossucca}@ismb.it

A. Albanese, R. Vigna, N. P. Premachandra
Politecnico di Torino
Corso Duca degli Abruzzi 24, Torino, Italy
Email: {adriana.albanese, rossella.vigna, prasanna.nishshanka}@polito.it

***ABSTRACT*—This paper describes a new approach for intensive rainfall data analysis. The ITHACA Early Warning system is conceived to give an alert in advance about the occurrence of heavy rainfalls around the world which can be used by WFP or other humanitarian assistance organizations to evaluate the event and understand the potentially floodable areas where their assistance is needed. This system is based on precipitation analysis and it uses rainfall data from satellite at worldwide extent. The Tropical Rainfall Measuring Mission and Multisatellite Precipitation Analysis dataset is used in this project. This product is delivered in near real-time for monitoring the current rainfall condition over the world. Considering the great deal of data to process, this paper presents an architecture solution based on Grid Computing technics. Our focus is on the advantages of using a distributed architecture in terms of performances for this specific scenario.**

***Keywords*-grid computing, e-science, early warning system, distributed architecture, scheduling, river basins, hydrological analysis.**

I. INTRODUCTION

The Early Warning System is developed by ITHACA (Information Technology for Humanitarian Assistance, Cooperation and Action)[1] in collaboration with Istituto Superiore Mario Boella (ISMB), under a special request made by WFP (World Food Programme) [2], the food aid branch of the United Nations, in order to increase efficacy in approaching emergency preparedness related to flood events. This completely automated system runs in river basin scale having a global coverage by using 3B42 and 3B42RT satellite rainfall data products of Tropical Rainfall Measuring Mission (TRMM) [3]. The 3 hourly real-time 3B42RT data with some statistical adjustments are used to detect critical rainfall events and to create alerts in near real-time.

Once alerts are triggered, they are automatically mapped using the informative layers extracted from a Spatial Data Infrastructure [4]. The overall situation can be visualized on a web application, in order to offer an easy access to the data during the emergency also for WFP's local offices. As mentioned before, all the procedures in realtime (including the Extraction Transformation and Loading procedures, the rainfalls analysis and data migration for the web application), are managed through the grid architecture, that will be described in the next paragraphs.

The paper is organized as follows: Section 2 discusses the rainfall analysis procedures, according to the geographical area subdivision. Section 3 explains the context system considerations and the choice of grid technologies to process data. Section 4 explains the architecture design and software agents functionalities. Section 5 is related to the performances results obtained using the Grid architecture. Future developments are exposed in the last section of this paper.

II. THE PROCESSING CHAIN OF REAL TIME TRMM RAINFALL OBSERVATION

The rainfall dataset used in this project is the Tropical Rainfall Measuring Mission (TRMM) Multisatellite Precipitation Analysis (TMPA).

For the historical database, the 3B42 product dataset has been loaded in a data warehouse architecture. That has been proved as suitable to manage more than 10 billions of records, performing complex data analysis and queries.

For the analysis of rainfalls in near realtime the 3B42RT product dataset was used.

The preliminary step for the analysis is ETL process (Extraction Transformation and Loading) structured through the following steps:

1) The extraction of data from ftp NASA servers [5] and converting data into a suitable format.
2) The transformation of data to fit operational needs (which can include quality levels). The transformation stage applies a series of rules or functions to the extracted data in order to convert them into the chosen format. These gridded files are transformed into ascii format starting from .hdf format (hierarchical data format) for the historical data set (TRMM 3B42 from 1998 to 2009) and from .bin format for the real time data (TRMM 3B42RT). Both data are resampled using a geographical layer of hydrographical basin (Hydro1k from USGS) [6], in order to perform the analysis in river basin scale.
3) Only for the historical dataset there is an additional step constitued by a loading procedure that uses a bulk load tool (sql loader)

After these pre-elaboration procedures, the rainfall data are analyzed through a hydrological model based on depth-duration-frequency curves [7], which allows the calculation of cumulated rainfalls in a specific period for each hydrographical basin. The rainfall historical data are used to define these reference curves to which the cumulated rainfall in real-time are compared with.

The first base data for the hydrological analysis is the TRMM rainfall data, having 3 hours temporal resolution and 0.25 degre sptail resolution (from 50 degrees south to 50 degrees North latitude)

The second one is the geographical surface area from the GIS watershed layer of HYDRO1k, a geographical database which provides a global coverage of topographically derived datasets including drainage basins. The watershed layer shown on Figure 1, is hierarchically organized through a territorial subdivision at different levels (from one to six).

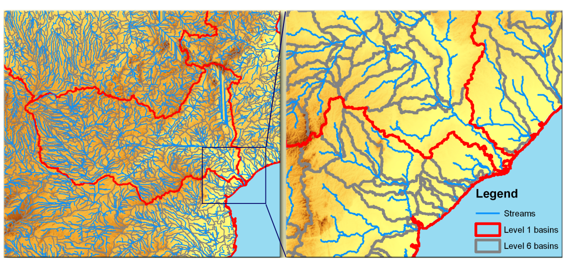

Figure 1. Drainage basin overview

The level 1 basins corresponding to large geographical regions (47 in total) are subdivided in 19570 level 6 basins. All geographical basins are referred to every river catchments: the intersection of the level 6 basins with TRMM grid has generated 311999 micro geographical entity. The rainfall values are resampled on the basis of these entities.

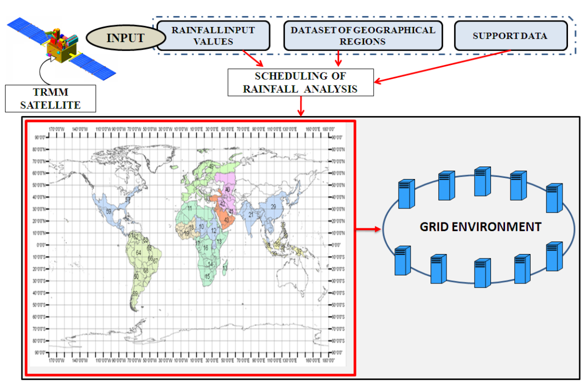

Figure 2. Flood Grid process schema

The real time rainfall analysis was split in 5 main steps.

1) Every 3 hours the system downloads satellite rainfall data product from the ftp NASA server.
2) Then, data are transformed in .txt format and are geographically located by adding the hydrological basin reference. The average of records contained on each file is 312000.
3) The system can now distribute the last data rainfall file on grid nodes for the elaboration (Figure 2).
4) For all geographical regions the system proceeds to the dispatching of the last rainfall values on each hydrological basin.
5) The next step runs the hydrological model to detect critical rainfall events.

   The Real Time System is conceived to analyze the real time rainfall and detect areas affected by critical rainfall in a time that does not exceed three hours.

A web GIS interface was developed (Figure 3) in order to obtain an immediate visualization of alerted areas [8].

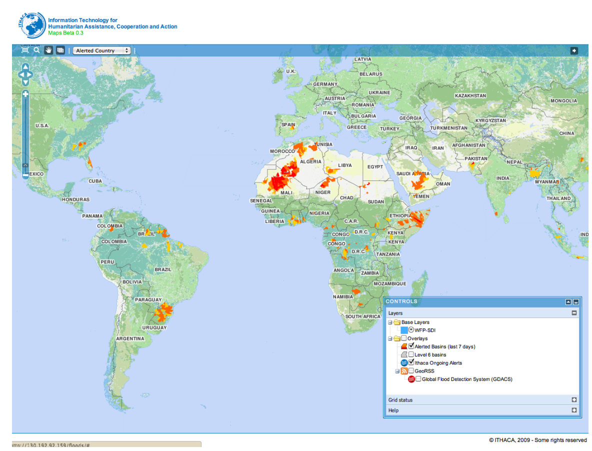

Figure 3. Web site overview

III. ARCHITECTURE MOTIVATIONS

The main goal of this project was to merge rainfall data analysis from satellite at worldwide extent and Grid computing technologies in order to reduce the time required for a global rainfall analysis.

The grid rainfall processing is an integrated system devoted to handle and process rainfall data from TRMM satellite 3B42RT. The general purpose of the entire infrastructure was to share the computational resources. The expected amount of data received every day was approximatively 2.5 millions rainfall values, corresponding to 312000 values each 3 hours.

The first goal was to find a solution that took in consideration the worldwide analysis requirements.

The second major requirement was related to the hydrological analysis in a time not exceeding three hours, since after this time the next values of rainfall are downloaded and prepared for a new analysis.

In consideration of this a grid technology was suitable to give a flexible architecture that can process such a huge quantity of data reducing the total processing time for the real time rainfall analysis.

On this specific context this technology solves the scalability problem, thus keeping the capability to process all data on time.

The hydrologic model is constantly improved in term of complexity and the needs in term of MIPS (Million Instruction Per Second) increase.

It has been decided to create a distributed architecture by using the Globus Middleware [9][10][11], an open source Grid software toolkit used for building Grid systems and applications. This technology has been applied to computationally intensive scientific analysis. The Globus software is addressed for the challenging problems in distributed resource sharing and is a fundamental enabling technology for Grid architecture. The toolkit includes software services and libraries for resources, jobs management, security and file management.

IV. ARCHITECTURE DESIGN

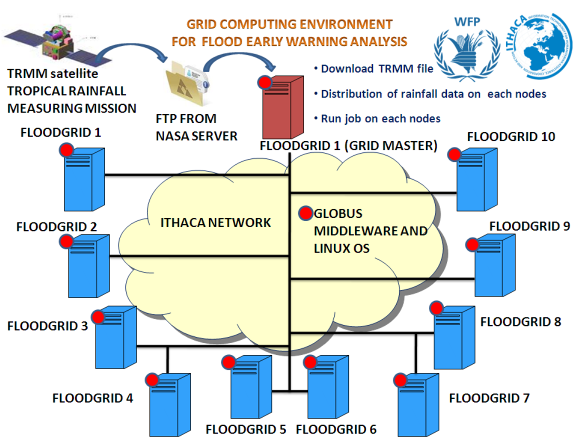

Figure 4. Flood Grid architecture

The Flood Grid Process Architecture is composed of 11 nodes. In the infrastructure grid nodes are Dual core 2 GHz with 2 GB RAM. The operating system is Ubuntu 8.10 and 9.04. Hence machines have the same characteristics and performances are very similar in term of computational capabilities. All nodes, grid Master and worker nodes was connected through a common IP protocols inside the intranet of the Ithaca research center: a master nodes and 10 worker nodes for the execution of jobs. The GridMaster was configured as GRIIS Grid Index Information Services and 4 main tasks were defined.

- Monitoring the availability of the worker nodes.
- The capabilities, every 3 hours, to download and generate the last txt rainfall data file from NASA ftp server through a software agent named Get TRMM rainfall agent.
- The capabilities to send the last rainfall value on each nodes.
- The capabilities to run the jobs for the hydrological model on each nodes through a specific Global Run Job agent

Each worker node was configured to run the hydrologic model.

A. *The Software Agents and Scheduler*

In order to guarantee the scalability of the system and the monitoring capabilities of the grid network, some agents were developed [12]. The agents are an emerging solution for providing a flexible, autonomous and intelligent software components. Using software agents, grid master node never sends request on gridded nodes to get information about grid services status. With this approach all agents are independent, preventing deadlocks and starvation. Finally all agents are autonomous The four agents are:

- Resources Agent: Installed on each node. The main function is monitoring the services status on nodes and sending periodically the status on the grid master node. The parameters sent on grid master are the space available on nodes, the grid services status like gridftp status and job management status.
- TRMM Rainfall Agent: is composed of two main functions, the first of which is to download the data from NASA ftp server and to transform them from binary to txt file format. The second one is distribute the last realtime rainfall data on each node on the grid architecture.
- Global Scheduler Agent[13][14]: Installed only on the grid master node, this agent performs firstly a resource discovery in order to verify the availability of grid service on each node. It does so by querying the information of nodes status stored on the database.

  The second stage is a resources selection implemented by applying the job requirements definition. In our scenario some selection requirements have been defined: the capabilities of nodes to send and receive files, the capabilities to execute the hydrologic algorithm and finally that the last running analysis is not in progress.

  The third task on a running time was dedicated to the monitoring progress of job execution. Two ways were developed: an automatic way by querying the job status on each node every 10 seconds and one time notification. In this case user interaction is needed, since in normal condition the automatic mechanism is disabled.

Finally, the last stage, job submission [15][16] by running the Job Scheduler Agent on each node involved on the process analysis. The attention for this software agent was the frequency of execution (3 hours), due to the delivery of rainfall data by the NASA server ftp. It is worth noting that so far the analysis requirement have not evolved in respect to the change of hydrological model algorithm: so, the analysis requirement was considered static.

- Job Scheduler Agent: Installed on each node, this specific agent was developed with some main functions. When the master node submits the run command to this agent, its first task is to verify the availability of rainfall data input to be analyzed. If some files were found the process starts distributing rainfall data on each region and subsequently staging hydrologic model for the analysis. The last task, related to the new critical events found, is the sending of the new alerts on the grid master node. We can note that if no rainfall data input were found, at the first step, the agent goes into sleep mode waiting for the next submission of job command from the grid master.

B. *Architecture Consideration*

As explained before, in this specific context the global coverage of earth was split in 312000 micro geographical areas. In a typical scenario based on a single machine each area is analyzed in a sequential mode: for N area to be analyzed the N+1 area could be processed only when the previous analysis was completed. This sequential scenario is defined Single Node Infrastructure (SNI).

In another scenario each geographical entity is referred to a hierarchical structure: large regions are subdivided in 3 levels (macro, micro and short geographical entity) thus making possible to parallelize the entire analysis of the earth considering a Multiple Node Infrastructure (MNI). This allows to analyze worldwide geographical region at the same time with a parallelization of the analysis on different nodes. In Figure [5], the comparison of processing time between the two different architectures is shown. On both examples the start time is set equal to t_0 and in the MNI event execution ends at $t_0 + \Delta_t$, in SNI it finishes at $t_0 + \alpha_t$, with $\Delta_t < \alpha_t$.

V. Improving Performances

Here it is shown a mathematical representation of a Grid Network. A Grid is composed of a set of resources nodes N.

$$G = [N_1, N_2, N_3, ..., N_n]$$

In the context of the paper it's also considered a set of Hydrological Basin:

$$H_{Basin} = H_{Basin1}, H_{Basin2}, H_{Basin3}, ..., H_{Basin\eta}$$

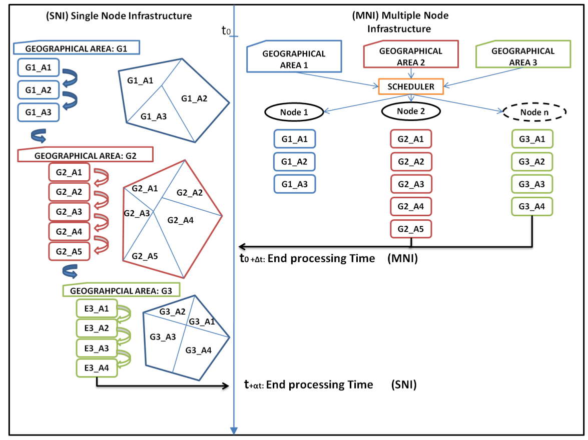

Figure 5. Scheduling approach SNI-MNI

each of them needs a process time for the rainfall distribution:

$$T_{dist}$$

A transfert time between grid master node to each worker nodes was defined

$$T_{ftp} = T_{ftp1}, T_{ftp1}, T_{ftp1}, ..., T_{ftpN}$$

Some test were implemented using the two different architectures: a SNI and MNI, in order to evaluate performances. The aim of the test was to obtain an estimation for the total processing time.

Execution time on SNI is:

$$T_p = \beta_{et} + T_{dist} + (H_{Basin} * \eta)$$

Time execution on MNI is:

$$T_p = \beta_{et} + (T_{ftp} * N) + \frac{T_{dist} + (H_{Basin} * \eta)}{N}$$

where:

$$T_p = TotalTimeProcess$$
$$H_{Basin} = HydrologicalBasinProcess$$
$$\beta_{et} = GeographicalTransformationbeforeanalysis$$
$$T_{dist} = TimeProcessforbasinRainfallDistribution$$
$$T_{ftp} = TimetosendRainfallFilesonEachNode$$
$$\eta = NumberofGeographicalBasins$$
$$N = GridNodeNumbers$$

In Table 1, average estimation time for the execution of a single basin on each node is reported. The test was realized for a sample of 5000 basins for the two contexts MNI and SNI. An extra time of the process due to the file transfer time are observed.

In Figure 6, the comparison between the total processing time for rainfall analysis at a worldwide range is shown . It can be observed that the total execution time decreases when the number of nodes increases. In

Nodes	Proc. Elab SNI (sec)	Elab MNI (sec)
Floodgrid 1	2.23	4.12
Floodgrid 2	2.25	4.19
Floodgrid 3	2.31	4.25
Floodgrid 4	2.18	4.16
Floodgrid 5	2.36	4.21
Floodgrid 6	2.23	4.22
Floodgrid 7	2.45	4.13
Floodgrid 8	2.31	4.22
Floodgrid 9	2.22	4.09
Floodgrid 10	2.20	4.01

Table I
MNI AND SNI ESTIMATION TIME FOR PROCESS A SINGLE BASIN

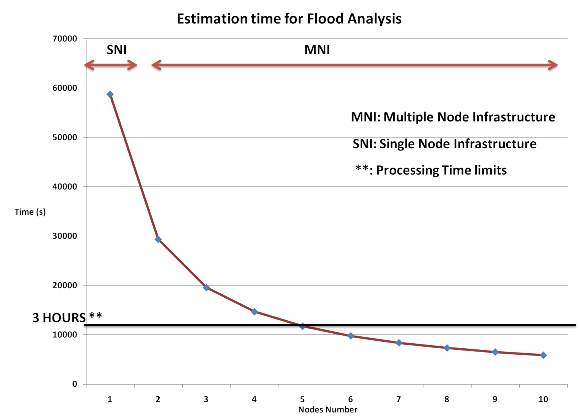

Figure 6. Processing Time MNI-SNI

an architecture composed of one node that corresponds of a Non Distributed Architecture scenario the total time is 58700 seconds (near 16 hours), clearly not an acceptable solution considering the needs in terms of time consumption requirements.

In a distributed scenario composed of two nodes 29300 seconds (over 8 hours) are consumed; as we can see from Figure 6 the highest the number of nodes, the lowest the execution time. It can also be observed that the critical number of nodes to satisfy the time requirements is 5. A gain of processing time could be obtained, obviously by exceeding the 5 node numbers on the Grid Network. We can also notice the asymptote behavior trend following the increasing number of nodes. In consideration of this the architecture proposal was based on a 10 nodes for the data elaboration and one node for the grid management.

VI. Conclusion And Future Works

In this paper we have presented our approach to reduce the total execution time in a specific context. Our application scenario was related to the calculation of rainfall critical events at a worldwide range: this involves a huge quantity of data to be analyzed; moreover, a big constraint is represented by the limitation in terms of process time for the elaboration of input data. In consideration of the requirements we have evaluated the difference between a multiple and a single node infrastructure in terms of performances. In spite of the extra time due to the transfer of files on each node, we could note that the choice of grid technology enables to reduce time consumption and increases general performances.

Another crucial consideration is the scalability of the system, since the hydrological model evolved in term of complexity. A direct consequence of this is that the time to process rainfall analysis increases; on this case, the distributed architecture guarantees the capabilities of rain data elaboration in 3 hours by adding new nodes.

As future works we plan to improve the scheduling of the jobs on the nodes by using multiprocessing technics based on threading in order to parallelize jobs execution.

References

[1] Ithaca Information Technology for Humanitarian assistance Cooperation and Action http://www.ithaca.polito.it

[2] WFP (World Food Programme) http://www.wfp.org

[3] Huffman G J, Adler R F, Bolvin D T, Gu G, Nelkin E J, Bowman K P , Hong Y, Stocker E F, Wolff D B (2007) The TRMM Multi-satellite Precipitation Analysis: Quasi-Global, Multi-Year, Combined-Sensor Precipitation Estimates at Fine Scale. J Hydrometeor 8(1):38-55

[4] A. Ajmar, F. Perez, O. Terzo, *WFP Spatial Data Infrastructure (SDI) implementation in support of emergency management*, Proc XXI Congress ISPRS (Int Soc Photogrammetry Remote Sens), 2008.

[5] ftp://trmmopen.gsfc.nasa.gov/pub/merged/mergedIRMicro.

[6] HYDRO1k ,http://eros.usgs.gov/products/elevation/gtopo30/hydro/index.html

[7] A. Albanese, F. Giorgi, N. P. Premachandra, O. Terzo, R. Vigna, *Application of an early warning system for heavy rainfall*, Proceedings of the Second International Conference on Earth Observation for Global Changes (EOGC 2009), Chengdu, China, 25-29 May 2009.

[8] E. Agosto, S. Dalmasso, *Ithaca geospatial services on a FOSS framework*, Proc GI4DM19-25, 2009.

[9] I. Foster,Kesselman ,The Grid2: Blueprint for a New Computing Infrastructure,2003

[10] Globus,"http://www.globusconsortium.org/

[11] Berman, Fox, Hey, *Grid Computing Making the Global Infrastructure a Reality*,Wiley,2005

[12] P. Gradwell, Grid Scheduling with Agents, Department of Computer Science, University of Bath

[13] Sofia Dimitriadou, Helen Karatza, *Job Scheduling in a Distributed system Using Backfilling with Inaccurate Runtime Computation*, CISIS, 2010

[14] R. Yahyapour, Philip Wieder, Gid Scheduling Use cases, 2006

[15] F Xhafa, S. Pllan, L.Barolli, Grid and P2PMiddleware for Scientific Computing Systems, CISIS, 2010

[16] Kurowsky, Nabrzyzki, Scheduling jobs on the Grid-Multicriteria approach, Computational Methods in Science and Technology, 2006