



*Conference Proceedings Paper – Remote Sensing*

## **An Integrated GeoAgro Webtool for Spatial Data Visualization and Dissemination**

**Jalal Eddin Omari and Chandrashekhar M. Biradar**

International Center for Agricultural Research in Dry Areas (ICARDA), CGIAR

E-Mails: j.omary@cgiar.org (J.O.); c.biradar@cgiar.org (C.B.)

Author to whom correspondence should be addressed; E-Mail: j.omary@cgiar.org;

c.biradar@cgiar.org; Tel.: +962-6-5903120; Fax: +962-6-5525930.

*Published: 25 June 2015*

---

**Abstract:** Today, with massive flow of the spatial and non-spatial database, enormous field data collection with enhanced electronic protocols in agricultural research have led to assemblage of massive caches of digital information on agriculture that describes the context-options for technological as well as socio-economic interventions at multiple scales and scenarios. However such tools and web services are often limited to certain geographical regions and/or specific thematic domain and do not exist for inherent data lacking regions such as agro-ecosystem in the dry areas of the developing countries. In a bid to improve digital agro-ecosystem services in the dry areas and develop a system which is economical and easy to replicate at national level, the International Center for Agricultural Research in Dry Areas (ICARDA) recognizes that Geoinformatics in an integrated agro-ecosystem is critical in obtaining adequate data at an appropriate time, it can help making precise decision about planting, growing, harvesting and marketing, which leads to increased net return while decreasing environmental impact. In order to achieve this ambitious goal, the foremost step is to make spatio-temporal information at multiple scales available at the finger tips in the era of smart phones and personal computers. ICARDA developed number of geospatial tools and web portals for data access, visualization and sharing to help researcher and decision makers at various scales to influence agricultural research, programs and policies. In this paper we discuss latest trends and advances in an integrated web-based spatial data processing, visualization and sharing platform for agro-ecosystem research, application and outreach. The workflow uses the hybrid multifarious protocols for data archiving, processing, visualization and sharing. An interactive user friendly end-user segment allows to query and process various biophysical parameters. Information generated either from static and/or dynamic layers

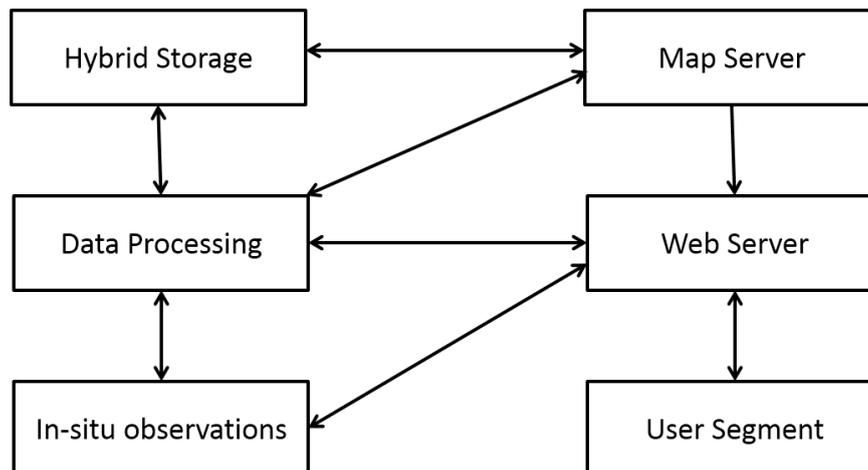
allow to move from mapping basic land units to quantitative description of the various components needed to improve overall system sustainability and productivity, and to develop and monitor key indicators across a range of cross-cutting themes needed to improve food security and livelihoods in dry areas of the world.

**Keywords:** Spatial data; Geoinformatics; Webtools; Visualization; Agro-ecosystems;

## 1. Introduction

With ever increasing human population, the increased demand for food and fiber needs has put tremendous pressure on land resources. Dry areas of the developing world occupy about 41% of the earth's land area, and are home to 2.5 billion people and 1.5 billion livestock while these areas have limited natural resources, land degradation and frequent droughts severely challenge food production on sustainable basis. Quantification of such dynamics and drivers with account of vast and often inaccessible geographic extents is limited or not possible with conventional approach. With advent of earth observation data at multiple scales on a near real-time basis over a cloud sources, in conjunction with increased computational and data speed have enriched the spatial data analytics towards 'precision-decision' making capacity. However such information are still limited to experts in the field of remote sensing and computational sciences and certainly are not in easily usable forms for non-spatial experts such as farmers, breeders, decision makers, etc. at their fingertips.

In order to address such limitation, scientists at ICARDA developed remote sensing based inputs with an integrated web-tool for data request, processing, visualization and dissemination. Any end-user who is capable of operating web browser will be able to find or generate on-demand request without the need to deal with an intermediate process. The generated datasets will be available at multi-temporal scales from farmscape to landscape levels. These tools overcome complex algorithms and workflows at end user segment to transform data into useful products and information directly usable and compatible formats. A simple workflow of an integrated GeoAgro web-tool is depicted below (Fig. 1)

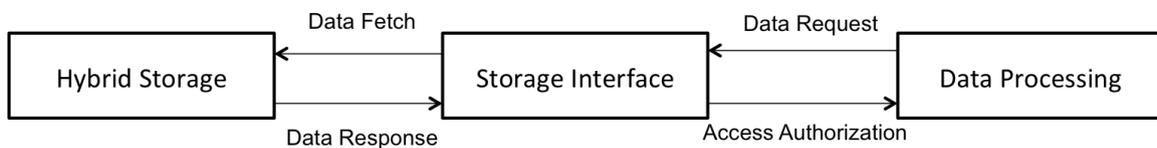


**Fig. 1** Key components of an integrated web-tool

## 2. Hybrid Storage

Tool uses hybrid storage to achieve fastest input and output (IO) operations. This storage consists of local storage area network (SAN) archive system where all raw and processed data are available. This SAN is connected to high-performance computing (HPC) cluster within the same LAN and with high-speed fiber channels. The open access cloud storage uses Amazon Web Services (AWS) S3<sup>1</sup>, with its recently enabled Landsat<sup>2</sup> 8. Integrated web-tool uses such resource as raw and near-real-time inputs to achieve on the fly data processing rather than downloading required data into our SAN for local processing, this has enabled faster and updated data source.

To provide the best way to extend tool components and include new parts, all system components interact with each other via integrated interfaces, for example in hybrid storage system, mechanism has been set not to allow direct communication between data processing and data storage, instead everything should go through the data storage interface which manages access roles and unified response through access interface structure (AIS) schema (Fig. 2).



**Fig. 2** Data flow in the access interface structure (AIS)

As part of hybrid storage, we developed an AWS search and download tool in order to search and download available tiles based on multiple attributes criteria, this criteria consists of geographical location (longitude and latitude or path and row), date and cloud coverage. This tool is developed using C#<sup>3</sup> and provides user friendly graphical interface and command line. <http://geoagro.icarda.org/awsl8.html> (Fig. 3)

<sup>1</sup> <http://aws.amazon.com/public-data-sets/landsat/>

<sup>2</sup> <http://landsat.usgs.gov/>

<sup>3</sup> <http://www.microsoft.com/net>

**Fig. 3** Screenshot of Landsat 8 tool

### 3. Data Processing:

On the fly data processing is the most complicated component of the tool. It's responsible about doing all the data transformation from raw formats into useful informational products (e.g. vegetation indices, change detection, etc.). The power of this component comes from two parts which have been explained below with taking an example of most simple vegetation index such as Normalized Differential Vegetation Index (NDVI):

1. Processing Algorithms: the data is processed based on predefined algorithms, these algorithms define the required input and then generate final output. Such algorithms are varies from simple two-band vegetation index/ratio to more complicated indices such as plant pigments, pest risk and actual evapotranspiration (ETa). The simple algorithm of NDVI uses two-band of Landsat 8 (Band4-Red and Band5-NIR) to generate float or scaled NDVI surfaces. On the fly algorithms calculate normalized ratio of the two bands to generate NDVI as:

$$NDVI = ( NIR - R ) / ( NIR + R )$$

Some other even more complex algorithms require inputs from cloud as well as physical storage, such algorithms require large amount of processing time and resources, such as drought monitoring, climate data downscaling, yield forecasting and technology out scaling. Most of these algorithms are developed using IDL<sup>4</sup>, Python<sup>5</sup> including GDAL<sup>6</sup> and ArcPy<sup>7</sup> modules and by using C# in some cases.

<sup>4</sup> <http://www.exelisvis.com/ProductsServices/IDL.aspx/>

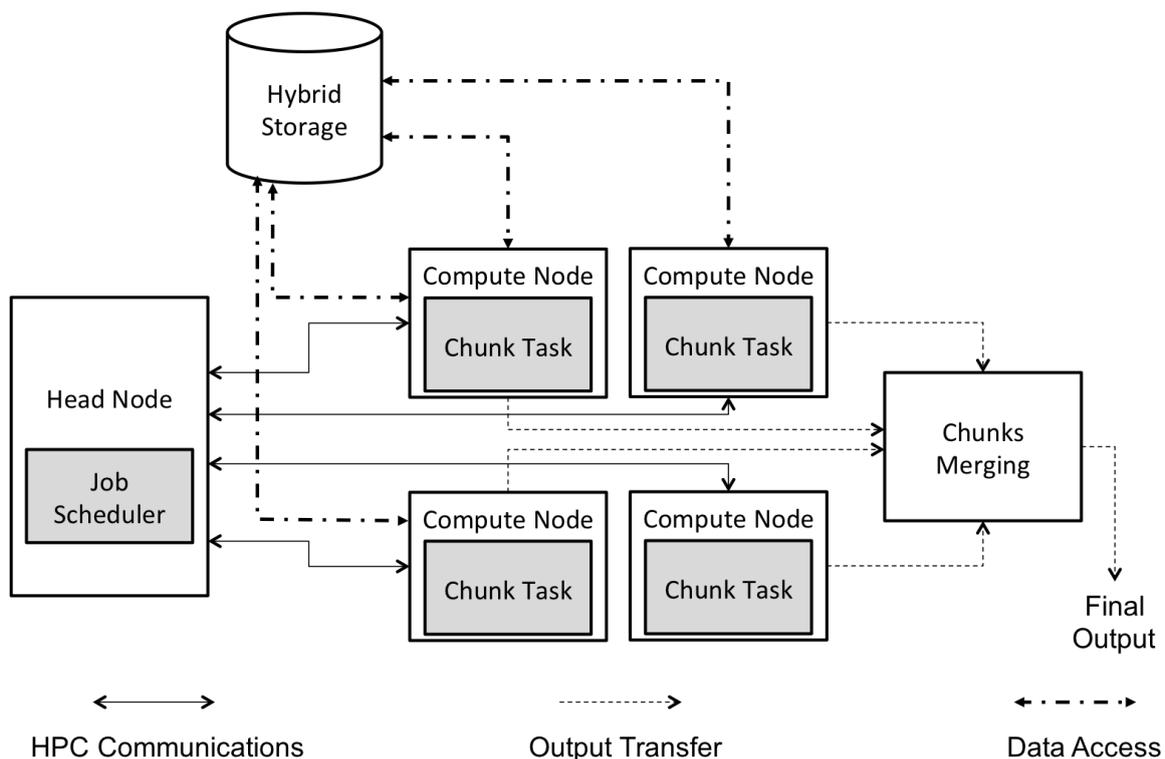
<sup>5</sup> <https://www.python.org/>

<sup>6</sup> <http://www.gdal.org/>

<sup>7</sup> <http://blogs.esri.com/esri/arcgis/2010/02/12/what-is-arcpy/>

2. High Performance Computing (HPC) Infrastructure: the need of processing large-scale surfaces for longer time-periods such as single Landsat tile at given location over three decades or daily climate data downscaling at 1km resolution surfaces for the last 35 years require huge computational processing by using parallel techniques, however under very limited hardware resources. We discovered many HPC solutions and decided to use Microsoft HPC Server<sup>8</sup>. It is compatible with our network infrastructure and provides one of the best parallel computing management tools, also the availability of good API<sup>9</sup> made it easier for us to adapt it. This infrastructure consists of 16 high performance machines and very powerful server. As total our HPC contained 148 CPU cores and 400GB of RAM. Task management is done by custom made C# application and Excel Macros (Fig .4).

The main process is split based on geographical locations, so each process covers a part of the whole surface and then all outputs are merged into one final output.



**Fig. 4** HPC infrastructure and chunks processing

After processing is done, all outputs are transferred to mapping server in order to make them available to other systems including desktop and web systems. This process is done in automatic way in order to complete the entire workflow from raw data to final useful information. We are using ArcGIS for Server as our main mapping server solution and as in other parts of the system, we developed a map server interface. This helped us in two ways:

<sup>8</sup> <https://technet.microsoft.com/en-us/library/cc514029.aspx>

<sup>9</sup> Application program interface (<http://www.webopedia.com/TERM/A/API.html>)

1. Access Control: due to the different data access levels, we had to find a way to control this access and map server interface is the best place to do this control.
2. Expandability: dealing with map server as interface will make it easier to add new servers without affecting other components. And also since different mapping servers have different access methods, this interface will manage standardizing system internal access and response schemas.

#### 4. Data Visualization:

This part defines the interactivity between users and the system. The system displays all available data through a web tool which combines all data sources in user friendly interface. In the past, at most instances, access was limited to desktop and laptop devices, however recently a good amount of traffic comes from tablets and mobile devices, and each one of these devices has a different display settings and screen dimensions. It is necessary to build a display interface to accommodate wide array of the devices using *Responsive Design*<sup>10</sup> framework. But the challenge was to fit one of these frameworks into spatial data visualization due to gridded nature of the database and that each grid has its own attributes and information. The *Twitter Bootstrap* framework<sup>11</sup>, 3<sup>rd</sup> version is used, it provides the best practices to design the webpage for different screen sizes without the need to generate different HTML code.

The web visualization tool uses latest and open source online libraries and servers such as OpenLayers 3<sup>12</sup>, jQuery<sup>13</sup>, Django<sup>14</sup>, Zend Framework<sup>15</sup> and MySQL<sup>16</sup>. Data for this tool comes from the map server that handles all processed data. When the user wants to visualize any surface, the process query is initiated through browser and transferred to the web server, web server will check user request and forward it to map server (through map server interface) and then the requested data is returned via Web Map Service (WMS)<sup>17</sup>. WMS is a standard protocol for serving georeferenced map images over the internet that are generated by a map server using data from a spatial database. The Open Geospatial Consortium (OGC)<sup>18</sup> released WMS in April 2000 and all map servers support this service. The purpose of this service is to transfer all spatial data from map server to web page in bitmap format e.g. PNG, GIF or JPEG. In addition, vector graphics can be included: such as points, lines, curves and text, expressed in SVG or WebCGM format (Fig. 5).

---

<sup>10</sup> [http://en.wikipedia.org/wiki/Responsive\\_web\\_design](http://en.wikipedia.org/wiki/Responsive_web_design)

<sup>11</sup> <http://getbootstrap.com/>

<sup>12</sup> <http://openlayers.org/>

<sup>13</sup> <https://jquery.com/>

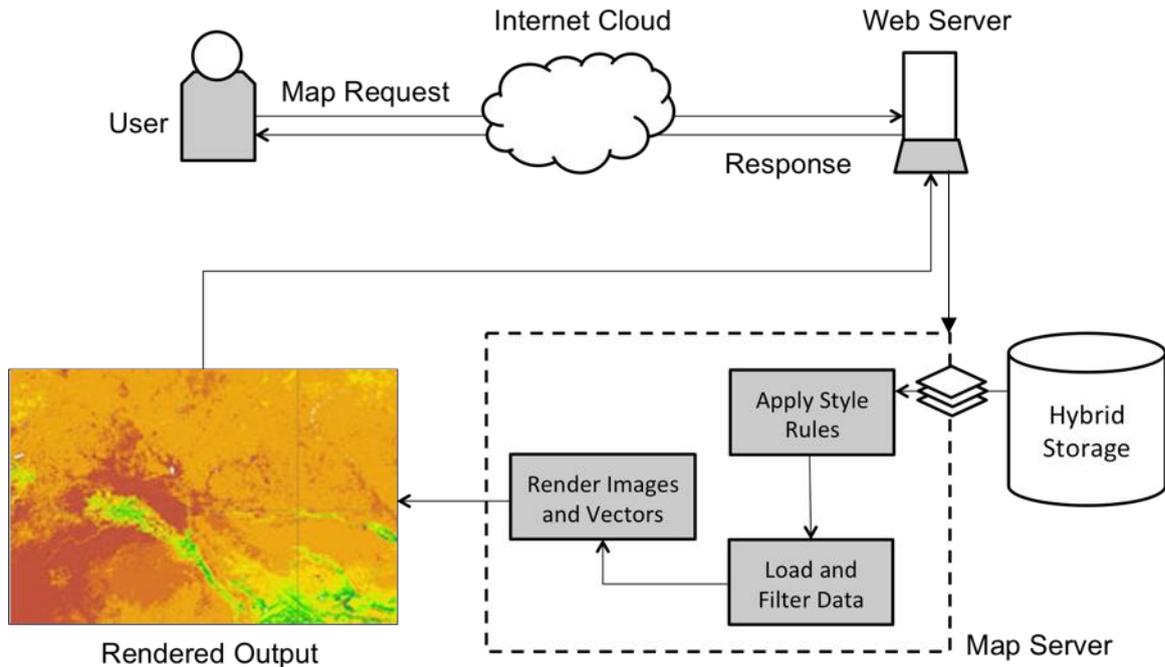
<sup>14</sup> <https://www.djangoproject.com/>

<sup>15</sup> <http://framework.zend.com/>

<sup>16</sup> <https://www.mysql.com/>

<sup>17</sup> <http://www.opengeospatial.org/standards/wms>

<sup>18</sup> <http://www.opengeospatial.org/>



**Fig. 5** Map request workflow

The tool uses the WMS response to add the layer(s) to OpenLayers object included into the web page. OpenLayers provides a flexible way to add multiple layers from different sources and in different projections. Beside the WMS response, server will return other related information like attributes or legend to display it in its placeholder. JQuery is used to manage all web page components' functionality like hiding, collapsing, showing, etc. Interface has also an option to incorporate in-situ observation or ground-truth dimension to the data. This data comes in two types: Georeferenced Images and Collected Surveys, which helps in generating new outputs and their validation.

The web tool backend serves business logic behind the tool. It manages user requests and their responses and also defines all user roles and access control. This backend process uses PHP and Python, both are served through Apache web server (mod\_proxy<sup>19</sup> is used to serve python). In both languages we used professional frameworks. Zend Framework (PHP<sup>20</sup>) and Django (Python). All non-spatial data is stored into MySQL server and served through PDO<sup>21</sup> and MySQL-python<sup>22</sup> module.

If the user requested data that is not available into map server but system already knows the algorithm to generate it. A processing request will be sent from web server to the processing server, this request is analyzed based on one of the following two scenarios;

<sup>19</sup> [http://httpd.apache.org/docs/2.2/mod/mod\\_proxy.html](http://httpd.apache.org/docs/2.2/mod/mod_proxy.html)

<sup>20</sup> <http://php.net/>

<sup>21</sup> <http://php.net/manual/en/book.pdo.php>

<sup>22</sup> <https://pypi.python.org/pypi/MySQL-python>

1. Short Process: in this case server will immediately send request to HPC server and waits for the output to be generated. This output shouldn't take long time and once it's ready, the response will be returned back to web page to be served and visualized.
2. Long Process: the process in other cases will take long time to be implemented due to the complex algorithms, and in this case user will be notified about processing queue and required time and then send a notice of completion via an email.

## 5. Conclusion

The main purpose of the this initiative is develop user friendly and cost effective spatial data processing, access and visualization system for data as well as a resource for limited regions. The integrated web tools provide new dimension for remote sensing data visualization and dissemination using both hybrid and open access protocols. It enables ease of use for multiple stakeholders at various levels. The HPC infrastructure from unused or idle computational resources increased the speed of the data augmentation and delivery. The backend architecture of the interactive tool is complex due to array of the spatial-temporal scales and interaction between multiple systems and servers.

This paper shares the knowledge and experience gained in the first phase of the system development, and shows its potential application in agro-ecosystems. It also provides a discussion platform for co-learning protocols for further development.

## 6. References and Notes

1. Astatsryan, H.; Hayrapetyan, A.; Narsisian, W.; Saribekyan, A.; Asmaryan, Sh.; Saghatelyan, A.; Muradyan, V.; Guigoz, Y.; Giuliani, G.; Ray, N. An interoperable web portal for parallel geoprocessing of satellite image vegetation indices. *Earth Science Informatics* **2014**. doi: 10.1007/s12145-014-0165-3.
2. Wang, D.; Zhao, Z.; Shaw, S-L.; Ragghianti, G; Wei, Y. Building a High Performance ArcGIS Cluster and its Application to Climate Data Processing, Proceedings of the GeoComputation 2013 Conference, Wuhan, China, 23 - 25 May 2013.
3. Faria, S.; Pendelberry, S. Hawker, J.S. ArcGIS in a High Performance Computing (HPC) Virtual Environment, Proceedings of 2010 Esri International User Conference, California, USA, 12-16 July 2010.

© 2015 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).