# HiGIS - When GIS Meets HPC

Lu Liu[1], Anran Yang[1], Luo Chen[1], Wei Xiong[1], Qiuyun Wu[1], Ning Jing[1]

[1]Department of Information Engineering, National University of Defence Technology
Yanwachi 137, Changsha, China
Telephone: (+86)73184573480
Email: {luliu, yanganran10, luochen, xiongwei, wqy, ningjing}@nudt.edu.cn

## 1. Introduction

Classic Geographic Information Systems (GIS) e.g. ESRI ArcGIS, Quantum GIS running as a standalone desktop program take good advantage of the computing power of personal computers. The performance of such kind of GIS with regards to the geodata throughput, visualization as well as spatial analysis is restricted by the capability (number of cores, CPU frequency, memory speed, I/O latency and bandwidth) of the local computer. Although there are some server-based systems e.g. ArcGIS Server, Geoserver, Mapserver etc. can provide OGC-standardized web services, they put their main focus on providing mapping (WMS) or geodata service (WFS and WCS) via HTTP. However, to do a complicated geospatial computing mission is still a tough nut to crack. In recent years, some online GIS-like system emerged as the rapid rise of cloud computing. The typical representatives include GISCloud, CartoDB, ArcGIS Online, etc. The current systems did good job in mapping or geodata sharing, while rarely touched the area of geocomputation which sometimes is both compute and data-intensive.

What will happen when GIS meets high-performance computing (HPC)? Wang (2010) have done explorative and solid work in this area, and proposed CyberGIS. To promote the performance of geocomputation by leveraging the power of HPC, we have been making great efforts in the past two years to build a GIS with full functions in terms of geodata management, visualization and especially, high-performance geospatial computing tool set. The name of the system is HiGIS. The backend of HiGIS is running on an HPC environment, while the frontend includes a desktop and a web client which is fairly thin and cross-platform. There are hundreds of parallel geospatial computing tools provided in HiGIS. Geodata, tools and geoprocessing models are treated as the same, and abstracted as the concept **geoapp** in HiGIS. They are the basic elements in the ecosystem around HiGIS.

## 2. HiGIS = HPC + GIS

The main goal of HiGIS is to improve the performance of time-consuming GIS operations by utilizing parallel computing in an HPC environment, as well as to provide such enhanced GIS ability to as many users as possible via personalized, light-weighted and cross-platform client programs.

The server-side software stack of HiGIS includes system services supporting the three GIS core functions - spatial analysis, visualization and data management. The corresponding software modules are **higine** (HiGIS Engine), **hiart** (HiGIS Cartography) and **hipo** (HiGIS Repository) respectively.

**higine** is responsible for processing the geospatial computing requests like sent from client programs. In HiGIS, such kind of computing requests are normalized as workflows even the flow contains only one step of some fundamental analysis. **higine** can resolve and execute the submitted workflows. To ease the submission of geospatial computing requests from clients, **higine** defines a group of public data structures and service interfaces including *submit*, *cancel*, *get_status*, etc. Rather than directly forward the computing requests to the operating system, **higine** has to take care of the executing sequence of the submitted jobs, as well as how many computing resources should be assigned to each job because the underlying HPC environment is so different from a commodity personal computer running local operating system. Consequently, **higine** communicates with a batch job scheduler commonly used in HPC environments to schedule the computing resources and keep the load balanced.

**hiart** provides interfaces for registering and styling geospatial data for cartographic visualization. The visualized geodata in HiGIS is served by Tile Map Service (TMS) standard. When a piece of geodata is visualized, its spatial reference system is normalized to Web Mercator system (EPSG:3785). The reason for this normalization is to avoid on-the-fly map reprojection.

**hipo** encapsulates the access to the geospatial catalogue and metadata repository and provides a unified interface to both the other server-side components and client programs. When users import their own data to HiGIS, or export a piece of geodata to their local device, the data access interface will be invoked. The public service interfaces defined in **hipo** are mainly creation, retrieval, update and deletion (CRUD) operations of the data stored in the repository.

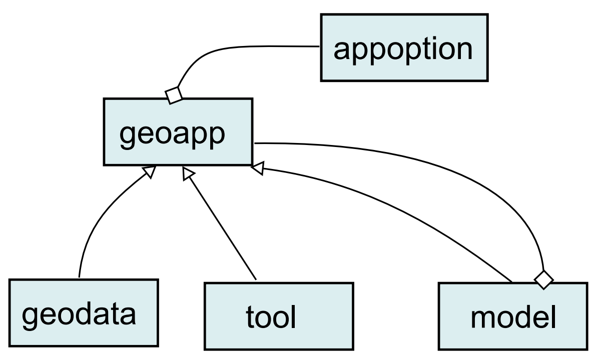


Figure 1. **Geoapp** object model in HiGIS.

The concept of **geoapp** is proposed to abstract the geospatial computing-related resources in HiGIS (see Figure 1). A **geoapp** object acts as a "producer", with some appoptions to determine what to produce. A geodata object represents a geospatial dataset with options like querying geographic extent to produce part of the dataset. A tool object represents an executable algorithm or a utility program, which produce the output or side effect of the execution. A model is composed of simple **geoapps**, e.g. geodata and tools. A model can produce composite results according to its definition and internal logics.

There is a group of excellent research teams from National University of Defence Technology, Institute of Geographic Sciences and Natural Resources Research, Institute of Computing Technology in Chinese Academy of Sciences, Wuhan University, Nanjing Normal University, Nanjing University, Peking University and North East University in

China jointly doing research and development work for the HiGIS project. The team have contributed the initial high-performance spatial analysis tools taking great use of parallel computing powered by HPC. The toolbox contains from fundamental analysis tools like buffer and overlay analysis, digital terrain analysis, spatial data transformation, to advanced geocomputation tools like intelligent algorithms, geostatistics and even complicated watershed modelling, etc.

The usage of HiGIS is a little bit different from the traditional GIS. First of all, an account for each user is necessary for the purpose of personalizing the **geoapp** resources owned by the user, as well as the working perspective of the system. HiGIS users may import their own datasets to the system while taking full control of the access level of the data. Besides the user-imported data, HiGIS provides large amount of free geospatial datasets acquired from other public data sources for high-performance geocomputation processing and research. Users may search and select the tools that fit for their processing requirements in the HiGIS **geoapp** repository. To launch a geospatial computing mission, users can either directly submit their jobs by providing necessary parameters via the GUI of a tool, or construct a complicated HPC workflow with an embedded model designer. **higine** working at the server-side will take care of the processing of jobs and job flows. Users can monitor the detailed running information of their jobs in a dashboard. When the submitted geospatial computing mission is completed, the result dataset can be visualized immediately if necessary.

So far, the 1.0 version of HiGIS has already been released but is deployed in an internal HPC cluster with 32 compute nodes in National University of Defence Technology. The running system provides strong support to solve some large-scale geocomputation problems. For example, the job of watershed modelling for an area in the south of Fujian province in China (~60000 square km) is finished within 5 minutes with 24 processes (Liu et al. 2013).

## 3. Visions of HiGIS

In the near future, HiGIS will be online and providing geocomputation services to public. Our visions of HiGIS include:

- **HiGIS in the Cloud.** The online version of HiGIS will be deployed in an HPC environment powered by cloud computing infrastructure, e.g. Cluster Compute instance in Amazon EC2.
- **Extended Data Compatibility.** The visualization and processing component of HiGIS will be more compatible to various types of data, e.g. spatio-temporal data, user-defined data.
- **Development Support.** An open development suite including mapping, processing and data access API will be provided to enable geoapp-development work with the support of HiGIS. Such a suite will help form a geocomputation ecosystem around HiGIS.
- **Geodata Version Control System.** A distributed geodata version control system will be included to support managing and tracing the change of a geospatial dataset when doing data editing.

- **Online Cartography.** We plan to add the support of a complete and elegant map styling language like CartoCSS[1] to enable online cartography and re-rendering on-the-fly.

## 4. Acknowledgements

This work is funded by National High-tech R&D Program of China (863 Program) (No. 2011AA120306) and Natural Science Foundation of Hunan Province (grant No. 12jj068). The development of HiGIS takes a lot of resources from open-source projects in related fields, e.g. Thrift[2], GDAL[3], TORQUE[4], mapnik[5], etc. We sincerely appreciate the community and developers of these projects. Support from numerous colleagues and experts from the other research teams under the umbrella of this 863 bundle project is gratefully acknowledged.

## 7. References

Wang S., 2010. A cyberGIS framework for the synthesis of cyberinfrastructure, GIS, and spatial analysis. *Annals of the Association of American Geographers* 100, 535–557.

Liu J, Zhu A, Liu Y, Qin C, Chen L, Wu H, Yang L, 2013, Parallelization of a grid-to-grid routing algorithm based on grids layering. *Journal of National University of Defense Technology*. to be published.

[1] CartoCSS | MapBox. http://www.mapbox.com/carto
[2] Apache Thrift. http:// http://thrift.apache.org/
[3] GDAL - Geospatial Data Abstraction Library. http://www.gdal.org/
[4] TORQUE Resource Manager. http://www.adaptivecomputing.com/products/open-source/torque/
[5] Mapnik. http://mapnik.org/