Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi
Dokuz Eylul University Faculty of Engineering Journal of Science and Engineering

Basılı/Printed ISSN: 1302-9304. Elektronik/Online ISSN: 2547-958X

An Open Source Geographical Information System Tool for Province Administration

İl Özel İdareleri için Yeni Bir Açık Kaynak Coğrafya Bilgi Sistemi Yazılım

Ömer Savaş [1], **Gıyasettin Özcan** [2*]

[1] Kütahya İl Özel İdaresi, 43020, Kütahya
[2] Bursa Uludağ Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, BURSA
Sorumlu Yazar / Corresponding Author *: gozcan@uludag.edu.tr

Geliş Tarihi / Received: 11.08.2018
Kabul Tarihi / Accepted: 01.10.2018
DOI:10.21205/deufmd.2019216112
Araştırma Makalesi/Research Article

Atıf şekli/ How to cite: ÖZCAN, G., SAVAŞ, Ö. (2019). An Open Source Geographical Information System Tool for Province Administration. DEUFMD, 21(61), 113-123.

Abstract

We introduce a new open source Geographical Information System tool for public administration. Our tool contains utilities for the geographical analysis of public governance that supply services to urban, suburban, and rural areas. We present an architecture that handles content heavy geographical image data. In order to achieve fast computation time and ensure security, we present a layered architecture. The tool exploits open source accelerators and interfaces. We also consider client-side development and provide an open source front-end web framework. We enabled spatial data generation and presentation utilities. The results denote that our cache usage speeds up the image creation time by 8x. Field experiences show that our Geographical Information System tool contributes to the province development strategies in a fast, reliable, and scalable form. The tool is still in use by Kütahya Province Administration.

Anahtar Kelimeler: *Computer Engineering, Geographical Information System , Layered Architecture, Data Generation and Data Presentation, Cache Support*

Öz

Bu çalışmada İl özel İdare yönetimi için geliştirilen yeni bir açık kaynak Coğrafi Bilgi Sistemi aracı tanıtılmaktadır. Araç kentsel, banliyö ve kırsal alanlara hizmet veren kamu yönetimine coğrafi analizi için yazılım öğeleri içermektedir. Bu amaçla resim verilerini işleyen bir mimari sunulmuştur. Hızlı hesaplama süresi elde etmek ve ağ güvenliğini sağlamak için katmanlı bir mimari kullanılmıştır. Araç açık kaynak hızlandırıcıları ve arayüzleri açık kaynaklı bir ön uç web çerçevesi sağlar. Mekansal veri üretimi ve sunum yardımcı programlarını sağlanmıştır. Sonuçta, önbellek kullanımlarımız resim oluşturma süresini 8 kat azaltmıştır. Alan deneyimleri, Coğrafi Bilgi Sistemi aracımızın hızlı, güvenilir ve ölçeklenebilir bir biçimde il gelişim stratejilerine katkıda bulunduğunu göstermektedir. Araç halen Kütahya İl Özel İdaresi tarafından kullanılmaktadır.

Keywords: *Bilgisayar Mühendisliği, Coğrafya Bilgi Sistemleri, Katmanlı Mimari, Veri Oluşturma ve Veri Sunma, Önbellek Desteği*

1. Introduction

Developing internet technology has become the dominant factor for the end users that require easy access to media data. Information processing technologies have enable fast and reliable access to even the most complex encyclopaedic data such as spatial context. In order to present data and location jointly, Geographic Information Systems have emerged.

A Geographic Information System, GIS stores, retrieves, analyses, manipulates and presents geographic data. GIS provides location information and related verbal information. Therefore, form of the GIS data is spatial. GIS is available in many areas including vehicle-tracking systems, GPS devices, from decision support systems and strategy planning [1-3]. User profiles of GIS exist in a wide range, from individual to corporations and even governments.

Since locations of GIS devices are not centralized, GIS implementations generally require network communication. For instance, a company may need to track all their vehicles on road. In addition, a government should need to collect spatial information of rural and urban areas by means of network devices and store the data in a centralized server.

GIS is a broad term that refers to various domains, applications, and processes. One of the first definition of GIS, proposed in 1968, attempted to design a system for regional planning [4]. The primary task of the study was the integrated analysis of social, economic, and land data for rural development. Afterwards, researchers considered theoretical concepts of spatial data handling and practical software applications. Reader can found advancements on GIS history in [5-6].

Benefits of rural life, urban life and industry encouraged the GIS applications. Literature analyzed and presented various GIS technologies including data capture, data representation, data restructuring, projections, accuracy and related information that comes from different sources.

A GIS should integrate at least four components: hardware, software, data, and management [7]. For efficiency, all components should be harmonious. Hardware is the component that process GIS. Network transmission medium, CPU, RAM, and cache are some of the most important components of the GIS. Since GIS commonly executes spatial data, implementations require heavy computational power. Therefore, we must utilize hardware components efficiently.

Second component, software, provides tools that retrieve, store, analyze and display GIS information. For retrieval, GIS software needs efficient storage, indexing, query management, visualization and easy access strategies.

Third component is data, and we can be retrieve it from various locations and adapters. We should access the data via in its simplest, and nearest form.

Finally, an efficient GIS must also contain efficient manipulation and management technologies. For instance, GIS should introduce efficient scale methods that minimize computation expenses and memory consumption. Even further, some queries may comprise proximity analysis such as looking for most suitable locations, frequency analysis of some goods or “what if” queries.

A GIS must introduce efficient visualization tools. The tools must handle maps and graphs. In most cases, the maps should integrate with reports, 3-D views and images.

Commercial GIS tools generally introduce better software capabilities. However, biggest drawback of the commercial GIS tools is the high price.

In order to overcome high price problem, it is possible to construct license free GIS solutions by freeware GIS components. Literature denotes various license-free GIS tools [8]. However, those tools aim specific problems. In contrast we aim to construct a compact GIS by collecting different components of freeware utilities. However, integration of GIS components introduced by different software developers is difficult because of consistency issues.

In terms of local governments with budget constraints, satisfactory and open source GIS tools are highly demanded. In addition, the local governments need tools that can contribute to rural and urban development of the province.

In this study, we consider the GIS requirements of a local government. Particularly, we consider Kütahya province of Turkey and its geographical information management. We present an open source and efficient GIS tool that aim to satisfy local government requirements.

The tool enables tools to manage urban and rural regions of the province as well as area populations, asphalt and paved roads, springs, title deeds, forests, farmlands...etc. It ensures efficient network communication among mobile devices and servers.

Implemented tool introduces data generation and data presentation utilities. It also enables data access by data purchasing, reading from the ground with GPS, drawing over the satellite image. We also considered the computational burden of spatial data processing. In order to speed up GIS computation we present a layered architecture. Furthermore, we improved the speed by means of cache mechanisms.

2. Materials and Methods

In this study, we design a cross-platform web application, which executes as a complete GIS. It provides necessary tools and applications to local governments and municipalities which have to deal with both rural and urban development. The tool does not require any plug-in or commercial licensed components. Even further, the design does its best to ensure Open Geospatial Consortium standards [9].

Particularly, the tool has been physically utilized in Kütahya province of Turkey, where provincial administration of Kütahya requires reports and services that are rendered throughout the city borders. To achieve these goals, geographic information datasets should be generated, manipulated, stored and displayed in proper forms.

In order to ensure flexibility, we designed a GIS architecture with layers, where a layer can be replaced by another equivalent. Furthermore, the tool offers distributed computing. In order to improve efficiency, it exploits cache mechanisms that exists in open source market [10]

In addition to flexibility advantage, layered architecture has at least two more contributions to our method. These are speed, and security [11]. In terms of spatial database, parallel data processing improves computation and ensures time speed up. In other words, the layered architecture enables us distributed computing [12].

Layered architecture also ensures security of the system since direct access to the data is not possible [11]. In contrast, distant users can make direct access to only web servers. Correspondingly, data access is managed in the internal layer structure of the GIS. For flexibility, our architecture consists of database, map server and, web application layers. The layered architecture enables distributed computing and high performance spatial data processing. A detailed explanation about the layered architecture is explained in the following subsections.

2.1. General overview of architecture

A general view of our GIS architecture and corresponding layers are shown in Figure 1. In the figure, A PC or a mobile device can access to the GIS by means of a server, which is named as web application. The web application contains software that handle internet communication. The web application is able to store and manipulate database for naive queries. If needed, a security server can be aligned in front of Web server as well.

The operating system layer is on the bottom of the architecture and manages both the GIS database and the web application layer that controls network communication. Also it manages the data transfer protocols among Map Server and database, DB.

The Map Server [13] functions between the web application and database. When a GIS query is requested from a distant host, the query initially reaches the web application layer. If Web application cannot respond to the query, it makes contact with database, DB, via operating system protocols.

Database queries can be routed to web application via the Map Server. Database server keeps the spatial data and interact with Map Server when needed. The data is stored by means of index algorithms for fast storage and retrieval. Spatial data is stored in Hard disk drives. Due to computer architectural limitations, storage and retrieval of data from hard disk drives are slow [14].

As can be seen in the layered architecture of the model, distant hosts cannot access databases directly and consequently database security is enhanced. Also, layered architecture aims to ensure that high density network communication processes do not overwhelm operating system or internal database; therefore, the workload of database servers can be reduced.

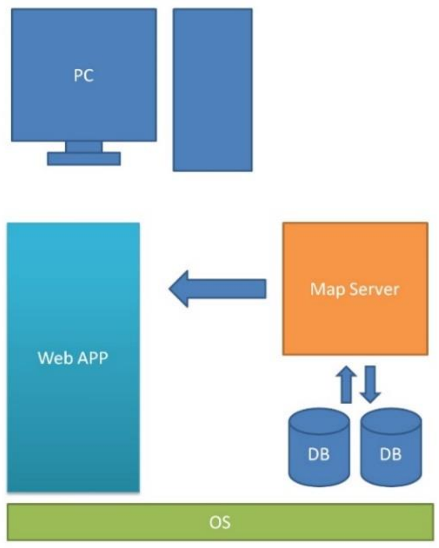


Figure 1. General Overview of the Layered Model

2.2. Database layer

Inside database layer, we need to select most efficient open access database system. For open source database administration, MySQL and PostgreSQL introduce powerful database query utilities [15]. Depending on the data size, alternative engines can be incorporated for database management systems. For example, the myISAM engine is faster for small scale data sets. On the other hand, InnoDB is better if database tables contain large numbers of records or GIS data set has spatial features [16]. PostgreSQL and PostGIS are also alternative open source database management tools. To select the most convenient option, we compared their performances by means of 2GB GIS data. Since a real example needs to handle large size data, we loaded the both database candidate with the same GIS data set.

In the Table 1, seven different SQL Queries are executed. Selected queries are some of the most common database queries. These are table creation, create unique index unique, create unique index updates key, database population, table scan and aggregate function operation. Results in Table 1 denote that PostgreSQL presents better latency performance for our SQL queries that satisfy provincial requirements. The results in table guided us to prefer PostgreSQL.

Literature denotes that PostgreSQL introduces wide range of beneficial features [15]. It has both object and relational database management system feature. In terms of extensible and standards compliance, PostgreSQL is very powerful. It can handle large workloads and handle concurrent users.

Table 1. Latency based comparison of Mysql ve Postgresql for large size data load (>2GB)

SQL Query	MySQL	Postgre SQL
create_tables	0,06	0,02
create_idx_uniques_key	112,25	29,46
create_idx_updates_key	119,33	41,14
populateDatabase	1063,95	331,65
table_scan	4,13	0,64
agg_func	2,9	1,77

2.3. Map server layer

Map Server is a web based software that is used to share GIS maps. In our tool, the map server layer has been set between DBMS and Web application. This layer paves the path for spatially enabled internet applications. Map Server inputs the database query results, extract the spatial features of data and outputs services. Since stored text information is transformed into spatial format in this layer, it may require high computation time. Therefore, selecting the most efficient open source Map Server may contribute to GIS performance decently.

We scanned the literature and decided that two open source map servers are convenient for our implementation: MapServer [13] and GeoServer where latter can be downloaded from "http://geoserver.org/" without fee. In terms of Kutahya province requirements, the principal efficiency factor is the response time. We compared the GeoServer and MapServer by response time to select most convenient map server. To do this, we utilized two data sets as shown in Table 2.

In Table 2, Dataset_1 contains 10.000 geographic points, whereas Dataset_2 10.000.000 geographic points. We compared Map Servers by

PostGIS and ShapeFile by means of geospatial vector data formats [9]. As result, we have presented four different comparison criteria and showed in each column of Table 2. Results in Table 2 denote that GeoServer yields faster response time for each criteria.

Particularly, GeoServer minimizes the response time by 12X, when large size Shape File is processed. Therefore we preferred exploit GeoServer at Map server layer.

Table 2. Query Response Time of Map Server and GeoServer in milliseconds

	PostGIS		ShapeFile	
Data set	Data Set1	Data Set2	Data Set1	Data Set2
Map Server	50	47	39	386
Geo Server	42	42	27	33

2.4. Web application layer

The web application layer of our architecture manages the following requirements for our implementation. The requirements are as follows:

- displays the maps on the browser
- runs the application on all mobile and desktop browsers
- runs the synchronization, runs database operations, runs relevant libraries.
- caches frequently used query data.

For display, we exploit Javascript and its map libraries. Besides, OpenLayers introduces satisfactory Javascript library to load, display and render maps from multiple sources. Due to requirements of the province, it is necessary to run the application across multi platforms, including desktop and mobile. To ensure cross platform capabilities, we used Bootstrap as a front end framework. In order to synchronize the tools and data, we exploit CodeIgniter, which is an MVC framework running on PHP [17].

The GIS must be improved by cache mechanisms as well. Cache is a physical memory which enable fast access to information [14]. For efficiency, we cache the maps that have been generated by Map Server since generation of a map from textual information causes big latency. When the same request come from another user, the Varnish cache fetches the result from its cache and sends the response without directing to GeoServer.

2.5 Improving the Scale

When the workload of the Map servers increases, we'll need additional computational power to integrate inside our layered architecture. In order fulfill the computational requirements of Map Server, we present two techniques. First technique is the integration of a cache, which keeps most frequently used spatial data for future usage. Our second technique is based on parallel computation.

We exploit open source Varnish cache [18] as HTTP accelerator, which stores data in virtual memory and designed for content heavy dynamic web pages [19]. The Varnish software utilizes round robin and random director as load balancing and presents us an efficient cache mechanism for geo-spatial data.

Varnish cache helps us to design parallel computation when the workload is high. When a single Map Server and Data Server is not sufficient to compute geo-spatial data, additional servers can be integrated to our layer based architecture by means of Varnish cache. As can be seen in Fig. 2, all requests will first be directed to varnish cache port inside Map Server. Thereafter, Varnish cache distributes the load evenly to the internal servers.

As can be seen in the Fig. 2, Servers are divided into two groups, where some of them are utilized as Map Servers, and others are used as Data Servers. Data Servers are supported by memory caches that keep frequently used data.

In Fig. 2, there exist multiple groups of servers in the architecture. The primary group is data and application servers. The servers in this group have the same characteristics and they are connected to a common database, using a common cache. When a server needs information, it scans the cache looks. If the information is not available in the cache, computationally expensive GIS results will be generated by the database server, and results will be sent to the client. GIS results are also written to the cache simultaneously. If a server updates the data in the database, the corresponding data in the cache server is deleted.

Requests related to the first group are not cached with the varnish service. If the incoming requests are not map requests it will be assigned to the most convenient server, based on load balancing. Within the distributed workload, a few more servers can be added to the data server group immediately when the application encounters an unexpected density. Even if it is deemed necessary, the weight coefficient can be determined according to the performances of the servers and some requests can be directed to other servers.

The second group of servers are the map servers. These servers should have the same hardware and software properties and do not store data on their memory. When the map data request arrives, the Varnish service checks the cache first, if the data are not in the cache, the map server will make a request to the idlest server of the second group of servers. Response of the server is sent to the client and cached by Varnish. In this way, the time-consuming GIS data generation can be fetched from cache without reading the map server. The servers in this group are automatically selected by load balancing, as in the first group.

In summary, open source Varnish cache implementation is not only used as a cache, but also it behaves like a gate between Web application and internal servers. It also manages the workload of Servers, which are shown in Figure 2.

3. Fundamental Utilities

Particularly, our tool is composed of two utility features. These are GIS data generation utilities, and GIS data presentation utilities. The GIS data generation utilities enable spatial data generation from mobile or desktop scanners. On the other hand, data presentation utilities provide facilities to read the spatial data and to transform into visual form for presentation. In the following subsections, we explain these utilities.

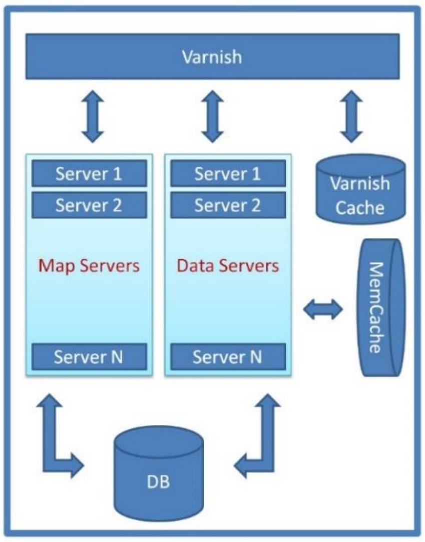
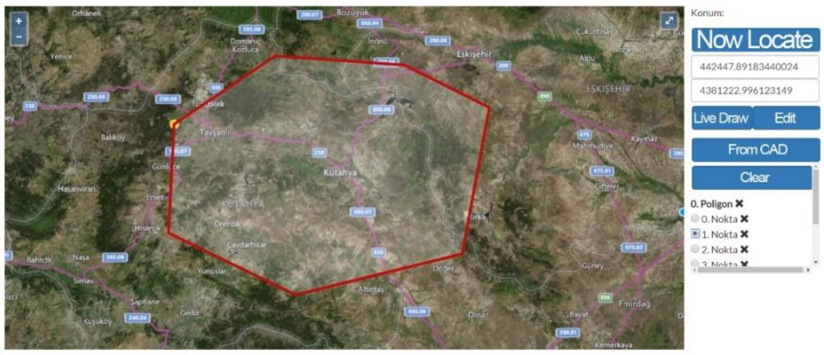


Figure 2. Network connection with Varnish cache model

3.1. Data generation utility

Particularly, in our tool, users can generate geographical data by logging into a desktop browser or to a mobile side application without any other software component. There is a data-adding screen where a user can add Bing or Google map as a pad on a map window. From this screen, user can generate point, line and polygon data by clicking on the satellite image. In addition, users can generate their own type of projections that they want to export, without depending on the format in the database.

If desired, users can transfer the geographical data or the coordinates from another CAD application and insert into data generation utility tool. In addition, if a land staff logs in from a mobile browser and clicks the live drawing button; the tool generates geographical data using the phone's location. In Figure 3, the user generates geographical data by walking around the polygon on line.



442447.89183440024

4381222.996123149

Figure 3. Data Generation from coordinates

3.2. Presentation with different symbols

During the geographical analysis of the province, users need to distinguish locations, roads,

Villages, etc. For instance, distinguishing paved roads and asphalt roads is necessary. Therefore, that investment strategy of the province can be determined in a clear form. In order to fulfill such requirements, our GIS presentation utility should introduce flexibility. For this goal, our tool enables to present Presentation with different symbols utility.

Whether user generates the data within implementation or imports one from another source, our web application is able to convert data into visual map form and present the results in an embedded map. When presenting this, user can define the symbol method using the SLD, which is a geographic data styling language.

In Figure 4, user can make village settlement spots larger or smaller. User can see the villages that are located in green colored forests. Additionally, users can change the roads where the gray color represent asphalt roads and brown roads are brown. User has also ability to change the color.

3.3. Geographical or textual search property

User can also search for geographic data in the system using geographic criteria. We present several ways for search property. In terms of geographical search, a user can draw a new geographic object and bring it into the intersection, which contains the query object.

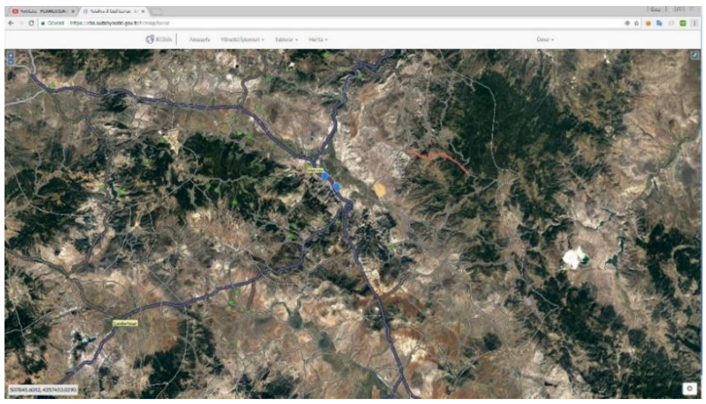


Figure 4. Presentation of villages and roads with different symbols

The user can also fetch the nearest objects as a customized search method and then export the data at the point when it is clicked on the map.

Search by means of a Point, Line and Polygon is available. The tool fetches and presents the corresponding nearest objects. After that, tool returns the clicked objects.

In terms of textual search, user writes the name of the location. The tool returns a map searches the name in the database. The result returns a map centering the searched query.

3.4. Interactive map updates

Our tool is interactive. In other words, geographic data from other sources can also be added as a layer on the map. The added layer can also be a dynamic geographic data layer, such as a vehicle tracking system. Figure 5 illustrates an interactive map usage. User of the tool is added as a layer on the map. The user can mark the vehicle, whose license plate and other information can be stored in the database. The user can track its position dynamically.

The interactive map updates are compatible to Web Map Services, WMS [20]. WMS introduces HTTP interface that fetches geo-registered web images from geo-spatial databases. Such feature improves the flexibility of our tool. In addition, the interactive map updates also run in harmony with Web Feature Services, WFS. The interface of WFS allows to request geo-spatial data from internet by means of platform independent calls. As a result, communication between mobile devices and servers yield efficient outputs. The tool supports various satellite images. However, we set the Bing satellite image by default in the tool.

3.5. Updates

Particularly, user can maintain the updates by two approaches. The first approach is periodic update. The system accesses the remote server during a period based on user definitions; it queries the remaining geographic data and places the data on the screen. The second approach is based on map movement. In this case, the tool does not permit periodical updates. When the user scrolls the map and zooms in and out, the tool displays by fetching the current data from the server. The second approach is suitable for rarely updated locations.

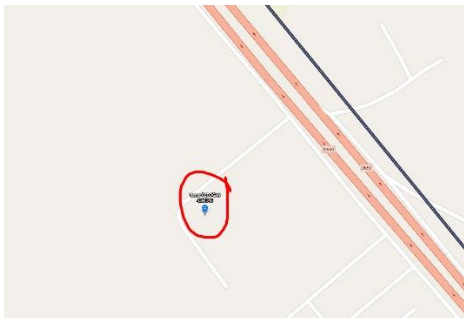


Figure 5. Vehicle tracking update of a user by means of adding a dynamic geographic data layer

The tool can also add verbal data provided by remote services. If there is an associated geographical data that you are verbally requesting, you can bring the data onto the map. For example, tool can visually present a title deed area on the map, provided by the title deed service with the person's name.

3.6. Legend and filtering features

Inside a map, we may need to use symbols to present brief explanation. However, users do not need to know the meaning of the symbols before. In order to solve the problem, we define map legends. A map legend presents visual explanation to symbols on the map.

User can display legends separately for each layer or display the legends for all data in a batch. By means of legends, we can easily define the borders, roads, road maintenance, bridges, vents, retaining walls, tunnels, residential areas, etc. at the same time.

In database, filtering is the manipulation of the data to make it favorable for the user. Filtering outputs the records that ensure qualifying criteria. With filtering feature, it is possible to transform the general map into a report screen.

For each layer we can generate separate filters. Unless the filter deactivation, the tool shows the map as filtered. For example, for planning purposes, it may be necessary to present the villages in a county with a population of over 100. When user executes such query via "Filtering feature", the tool introduces the necessary map that shows the villages that provides the condition.

3.7. 3D world feature

One of the most versatile feature of the GIS tool is the presentation of 3D world feature. 3D world feature can be used for planning and presentation. For planning, geographical properties of a region can be seen more clearly. So that optimal investment locations can be determined by 3D world feature. For instance, forestry regions and easiest access paths to the forests can be observed by 3D model.

In the tool, we represent asphalt roads with blue color and forestry regions with green color. So that, a strategic investment plan can be initiated by the 3D model. 3D model is a commonly preferred tool for presentation. In order to explain the requirements, 3D tools present simple proofs to explain the needs.

4. Speeding up Our GIS Tool with Cache Utilities

There exist several computational burdens during GIS program executions. One of the reason is based data features. Because of the size of locational data, GIS implementations utilize large size datasets. On the other hand, recorded GIS data are generally complex. For instance, user can represent locations by polygons. Even further records size of a polygon is not deterministic and resources can be stored in a distributed database. Therefore, data fetch operations from various resources cause latency.

Introducing new features to the GIS implementations also increases computation time. For instance, introducing the legend feature causes extra processing capacity. In order to present features such as Update, legend, or filtering, complex computational procedures are necessary.

One of the most fundamental computational burden of GIS occurs when it transforms textual data into geographical form or vice versa. In order to present GIS data, each GIS component and its coordinate address are stored in a textual database. When needed, the tool fetches textual GIS information from database. Depending on the coordinates, tool places the component into visual map. In order to ensure the procedures, complex image processing procedures are necessary.

To speed up the GIS performance, cache utilities are good option [21-22]. Cache is a memory that stores frequently used geographical data on the map. Given that information is in cache, we gain time since computer do not need to fetch textual data from database. We also do not need to transform textual data into visual form and align on the map.

In terms of writing a web application, scalability is one of the main issue. GIS implementation designers need to consider whether the application is able to serve mass users with available equipment. When we embed the cache utilities in the architecture in a relevant form, our web based GIS; implementation yields fast data presentation and generation opportunities. To strengthen our claim, we support the details of cache implementations with experimentation and empirical results.

In this study, we have use three different cache utilities for efficient GIS implementation. For a web based GIS implementation, we prefer the use of the following open source cache utilities: APC, Varnish cache and CodeIgniter cache. Each cache utility introduces specific advantage. We explain them in the following subsections.

4.1. Optimization with APC Cache and CodeIgniter Cache

APC, Alternative Php cache, is an open source cache for Php. It provides robust framework for caching and reduces pitfalls of interpreter languages. Our tool utilizes APC, since we exploit an interpreter language, Php during web application design. APC stores the opcodes in memory and consequently reduces the access to disk. As a result, the implementation reduces both file access and opcode generation costs.

For efficiency, we exploit CodeIgniter, which is an open source Php framework, containing pre-written methods and classes. It enables to cache the pages in their fully rendered state. We aim to speed up caching with CodeIgniter cache.

In this study, we modified the indexing technique of the cached files by means of kernel development. Particularly, we sorted the cached files according to index order, similar to URL indexing. Thus, search among cached files becomes faster. This technique also discriminates cached data of different languages. In order to evaluate caching, we queried the villages of the province, where a page lists 10 villages.

We repeated the same query three times, where first execution does not exploit cache. At second iteration, we exploited only APC. At last query, we used both APC and Codeigniter cache. We present the page generation time in Table 3.

Table 3 has three lines, where each line denotes a specific experiment. First line is the result, where experiment does not utilize cache. The table denotes that maximum duration time occurs when we do not utilize caching. Second line utilizes only APC. Results show that APC speeds up page generation time. We observe the maximum improvement when we utilize APC and Codeigniter caches collaboratively. In this case, we observe 25X improvement on page generation time when compared to page generation without cache.

Table 3. Map Server Side Benchmark Results for APC, CodeIgniter Cache

State	Page Generation Time (ms)
Without cache	0,1955
With cache	0,1046
With APC+CodeIgniter Cache	0,0069

4.2. Optimization with varnish cache

In order to handle dynamic GIS tools for content heavy information we exploited Varnish software, which is also open source. Varnish is efficient in high profile dynamic web sites [18]. Varnish is preferred since it supports high-traffic efficiently by means of heavily threaded architecture. Varnish utilizes virtual memory and enables harmony among operating system and utilized GIS caching [19].

Recall that Varnish cache takes place at the front side of the Map Server, where operating system manages the incoming network data. As explained in Fig. 2, the network connections are initially access Varnish cache. Our architecture deliberately put the Varnish on the front side of Map layer to prevent situations where operating system caches data, while Map Server writes the same data to disk for evaluation, we made a GIS query for the village roads in Kütahya province. We present the picture generation time in Table 4. In the table, response time of the Varnish cache was 8 milliseconds. When compared to best page generation of 69 milliseconds, results imply that utilization of Varnish cache speeds up the picture generation by approximately 8X.

Table 4. Map Server Side Benchmark Results for Varnish Cache

State	Picture Generation Time (ms)
Without cache	69
With Varnish cache	8

5. Discussion and Conclusion

A GIS must handle spatial data, which is content heavy. Furthermore, GIS should ensure mobile access and GPS management utilities to present efficient network access. Consequently, development of a GIS software solutions is becoming expensive and commercial licenses of GIS solutions require high annual prices.

In this study, we developed an open source GIS tool for provincial governments. The architecture of our GIS tool exploits open source resources. We use open source database, cache, and GIS server utilities. We introduce a new layered-based GIS architecture and integrate the utilities in a frame with web programming.

Our GIS tool introduces both data generation and data presentation utilities. In terms of data generation, we introduce various options for the users. User can either import data coordinates from another platform, or exploit GPS and mobile phones to determine the coordinates. A user can also select locations from a map.

In terms of data presentation, we introduce various opportunities. Using geographic styling language features, we introduce options to discriminate roads, villages or other properties by colouring and symbols. In addition, we introduce geographical or textual search properties. Interactive map updates, legends, 3D world views are some of the most essential parts of the application.

The architecture of the GIS tool supports both desktop and mobile platforms. Because of mobile platform opportunities, we can collect and analyze geographic information in the field.

To enable fast response to the user queries, we utilized cache methods. Particularly, we improved content heavy dynamic web pages by means of Varnish cache. Furthermore, we ensured harmony among APC and CodeIgniter caches to reduce GIS computation. Feedback in Kütahya province indicates that our GIS tool ensures efficient solution.

Acknowledgements

During development of the GIS tool, governance of Kütahya supplied all dataset. The authors of this study are grateful to the government officials of the province.

References

[1] Aloquili, O., Elbanna, A. and Al-Azizi, A. 2009. Automatic vehicle location tracking system based on GIS environment, IET software, Vol. 3, no. 4, pp. 255-263. DOI: 10.1049/iet-sen.2008.0048

[2] Pradhan, B. 2013. A comparative study on the predictive ability of the decision tree, support vector machine and neuro-fuzzy models in landslide susceptibility mapping using GIS, Computers & Geosciences, Vol. 51, pp. 350-365. DOI: 10.1016/j.cageo.2012.08.023

[3] Siljander, M., Venäläinen, E. Goerlandt, F. and Pellikka, P. 2015. GIS-based cost distance modelling to support strategic maritime search and rescue planning: a feasibility study, Applied Geography, Vol. 57, pp. 54-70. DOI: 10.1016/j.apgeog.2014.12.013

[4] Tomlinson, R.F. 1968. A Geographic Information System for regional planning, in land evaluation, CSIRO Symposium Organized in Cooperation with UNESCO, Melbourne, Australia 200-210.

[5] Foresman, T.W. 1998. The History of Geographic Information Systems: Perspectives from the Pioneers, 1st Edition. Prentice HALL PTR, New Jersey, 397 s.

[6] Goodchild, M.F. 2010. Twenty years of progress: GIScience in 2010, Journal of Spatial Information Science, Vol. 1, pp. 3-20. DOI: 10.5311/JOSIS.2010.1.2

[7] Heywood, I., Cornelius, S., and Carver, S. 2006. An Introduction to Geographical Information Systems, 3rd Edition, Prentice Hall, Essex, 426 s.

[8] Steiniger, S., and Hunter, A.J.S. 2012. Free and open source GIS software for building a spatial data infrastructure, in: Proc Geospatial free and open source software in the 21st century, pp. 247-261, Nantes, France

[9] Reichardt, M. 2017. Open Geospatial Consortium Standards, 1st Edition, The International Encyclopedia of Geography, Wiley, USA,

[10] Wilson, P.R., Johnstone, M.S., Neely, M, and Boles, D. 1995. Dynamic storage allocation: A survey and critical review. International Workshop on Memory Management, Scotland, September 27-29, 1-78.

[11] Elmasri, R. 2008. Fundamentals of database systems, 6th edition, Addison-Wesley, 1172 s.

[12] Chervenak, A., Foster, I., Kesselman, C., Salisbury, C. and Tuecke, S. 2000. The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets, Journal of network and computer applications, Vol. 23, pp. 187–200, DOI: 10.1006/jnca.2000.0110

[13] Vatsavai, R. R., Burk, T. E., & Lime, S. 2008. University of Minnesota Map Server ss 1197 – 1205., Shekhar, S., & Xiong, H. ed. 2008. Encyclopedia of GIS, Springer US, 1369 s.

[14] Patterson, D., and Hennessy, J.L. 2006. Computer Architecture A Quantitative Approach, 4th Edition, Morgan Kaufmann, 704 s.

[15] Rowe, L.A., and Stonebraker, M.R. 1987. The POSTGRES data model, 13th International Conference on Very Large Data Bases, September 1-4, England, 83-96.

[16] Comparing PostgreSQL vs. MySQL, 2012, http://posulliv.github.io/2012/06/29/mysql-postgres-bench/ Accessed: 2014-06-30.

[17] Blanco, J.A., and Upton, D. 2009. CodeIgniter 1.7., Packt Publishing, Birmingham, UK, 282 s.

[18] Velázquez, F., Lyngstøl, K., Heen, T.F., and Renard, J. 2018. Varnish Book, https://book.varnish-software.com/4.0/

[19] Tanenbaum, A.S. and Bos, H. 2014. Modern operating systems, 4th edition, Pearson, 1072s.

[20] Scharl, A., and Tochtermann, K. 2009. The Geospatial Web: How Geobrowsers, Social Software and the Web 2.0 are Shaping the Network Society, Springer Science & Business Media, isbn: 1846288266

[21] Li, H., Nakazato, H., and Ahmed, S.A. 2017. Request Expectation Index Based Cache Replacement Algorithm for Streaming Content Delivery over ICN, Future Internet, vol. 9, no. 4, 83, DOI: 10.3390/fi9040083

[22] Paul, S., Fei, Z, 2001. Distributed caching with centralized control, Computer Communications, Vol. 24, no.2, pp. 256-268, DOI: 10.1016/S0140-3664(00)00322-4