

Comparing speed of Web Map Service with GeoServer on ESRI Shapefile and PostGIS

Jan Růžička

Institute of geoinformatics,
VSB-TU of Ostrava
17. listopadu 15, 708 33, Ostrava-Poruba, Czech Republic
jan.ruzicka.vsb@gmail.com

Abstract

There are several options how to configure Web Map Service using several map servers. GeoServer is one of most popular map servers nowadays. GeoServer is able to read data from several sources. Very popular data source is ESRI Shapefile. It is well documented and most of software for geodata processing is able to read and write data in this format. Another very popular data store is PostgreSQL/PostGIS object-relational database. Both data sources has advantages and disadvantages and user of GeoServer has to decide which one to use. The paper describes comparison of performance of GeoServer Web Map Service when reading data from ESRI Shapefile or from PostgreSQL/PostGIS database.

Keywords: PostGIS, ESRI Shapefile, GeoServer, Web Map Service, performance.

Introduction

Size of the spatial data grows every day and their management is more and more complicated. Geographic information systems has moved from file based systems via database managed data to distributed data management. All three possible ways how to manage spatial data are still available and used. There are advantages and disadvantages in a case of all of them. For example is very easy to copy single file (group of files) to another user in comparison to copy whole database or whole distributed system. Or it is very difficult to store large sized data to single file (for example there are limit about 4GB for ESRI Shapefile format [4]). Or it is very difficult to perform some queries on large data on single computer. For example Kepka and Ježek mentioned another disadvantage of PostGIS: “PostgreSQL with PostGIS plays a role of the flagship of Open Source RDBMS but there is just limited possibility of simple and fast queries visualisation” [5]. Several advantages of PostGIS are mentioned for example by Stěhule [12]

GeoServer [1] can produce map outputs according to Web Map Service specification [9]. Lot of projects are based on ESRI Shapefile format. Lot of projects are based on PostgreSQL/PostGIS system. The question on which was research based was: “How fast will be GeoServer when serving maps according to WMS if the source of data is PostgreSQL/PostGIS table or ESRI Shapefile?”. According to results of Adam Shreier work [11]. I expected that GeoServer will be faster when dealing with ESRI Shapefile than with PostgreSQL/PostGIS table. I have discovered that this is not true and results show that there are other aspects that must be considered. I can conclude that running Web Map Service with GeoServer on ESRI Shapefile or PostgreSQL/PostGIS are comparable.

Methods

The data from COSMC (Czech office for surveying, mapping and cadastre) mentioned by Med and Souček in [8] that represents parcels in the Czech republic were used for this research. The services were tested by Horák, Růžička and Ardielli [3]. Number of records was about 15 millions and number of vertices was about 200 milions. The ESRI Shapefile files has about 8 GB together (cca 4 GB for geometry (SHP file) and 4 GB for attributes (DBF)).

The data were stored in ESRI Shapefile and in PostgreSQL/PostGIS table. For the ESRI Shapefile was created quad tree index. Shptree tool [7] with default options was used to build index. For PostgreSQL/PostGIS table were created two tables one without spatial index and one with GIST spatial index [6].

WMS based on ESRI Shapefile and on PostgreSQL/PostGIS table have been prepared. There were three types of WMS for PostGIS table, one for table without index, one for table with index and one for table with index connected via JNDI technology [10].

Following software and hardware configuration were used to perform testing.

- Intel(R) Core™4 2.4GHz
- 48 GB RAM
- SSD disks
- Ubuntu Server 14.04
- GeoServer 2.7
- PostgreSQL 9.3
- PostGIS 2.1
- Jmeter

According to recommendations [2] I have eventually tuned Postgrruzicka2.bbl.eSQL and run test on table with index again. Tuning was as described in Table 1.

Table 1: Tuning parameters

Parameter	Without pg_tune	With pg_tune
default_statistics_target	default	50
constraint_exclusion	default	on
checkpoint_completion_target	default	0.9
effective_cache_size	default	32GB
work_mem	default	288MB
wal_buffers	default	8MB
checkpoint_segments	default	16
shared_buffers	128 MB	11GB
max_connections	100	80

Five tests were done on WMS. Each test took 4 hours.

- ESRI Shapefile

- PostGIS without GIST index
- PostGIS with GIST index
- PostGIS with GIST index on JNDI
- PostGIS with GIST index on tuned PostgreSQL

Results

Table 2 shows responses of Web Map Service running on GeoServer with different data sources. There are minimum, maximum and average response times in seconds.

Table 2: Responses of Web Map Service

	Min (s)	Max (s)	Avg (s)	Std (s)
PostGIS without index	6.8	51.0	16.9	2.1
PostGIS with index	0.2	12.0	2.7	1.7
PostGIS with index via JNDI	0.2	15.9	3.1	2.0
PostGIS with index on tuned PostgreSQL	0.1	12.0	2.4	1.5
ESRI Shapefile	0.2	25.0	1.7	2.8

The figure 1 shows minimum response time for each configuration. The figure 2 shows maximum response time for each configuration. The figure 3 shows average response time for each configuration.

Table 3 shows number of responses of Web Map Service running on GeoServer with different data sources. The limits were specified 5 and 10 seconds.

Table 3: Number of responses over limit

	% > 5 s	% > 10 s
PostGIS without index	100.00	99.8
PostGIS with index	9.39	0.09
PostGIS with index via JNDI	16.45	0.47
PostGIS with index on tuned PostgreSQL	5.62	0.02
ESRI Shapefile	10.57	0.12

The figure 4 shows number of responses over 5 seconds in percents.

The figure 5 shows number of responses over 10 seconds in percents.

The figure 6 shows distribution of response time for each type of connection

Conclusion

I can conclude that running Web Map Service with GeoServer on ESRI Shapefile or PostgreSQL/PostGIS are comparable. There are not significant differences in average time of

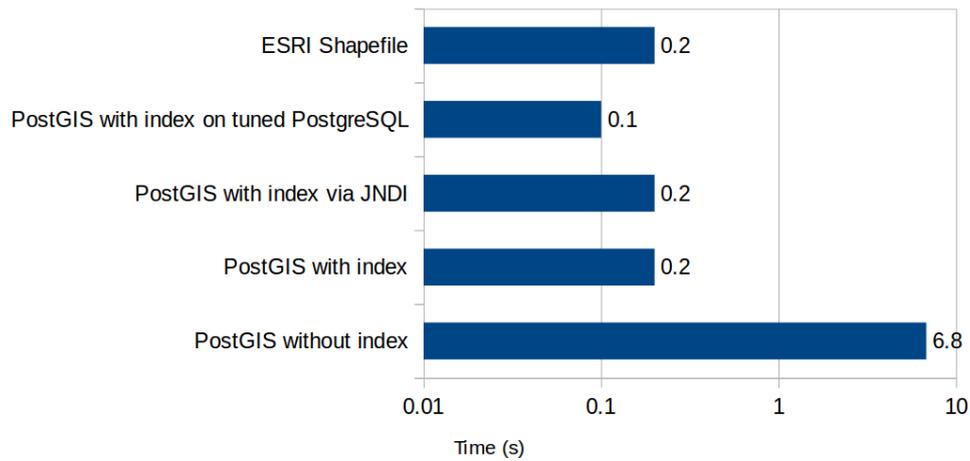


Figure 1: Minimum time for response

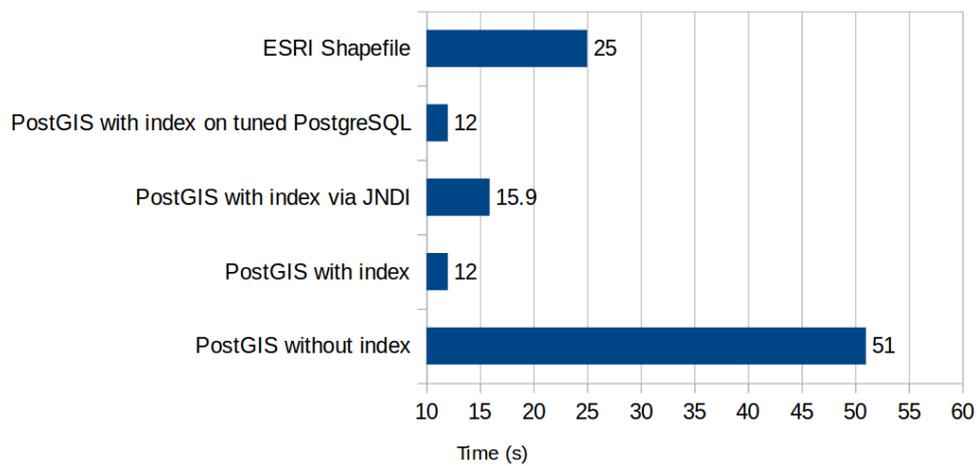


Figure 2: Maximum time for response

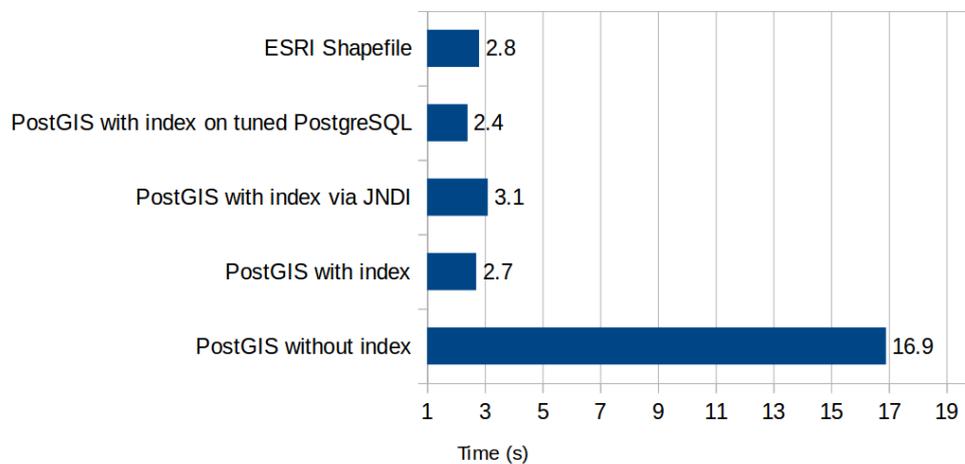


Figure 3: Average time for response

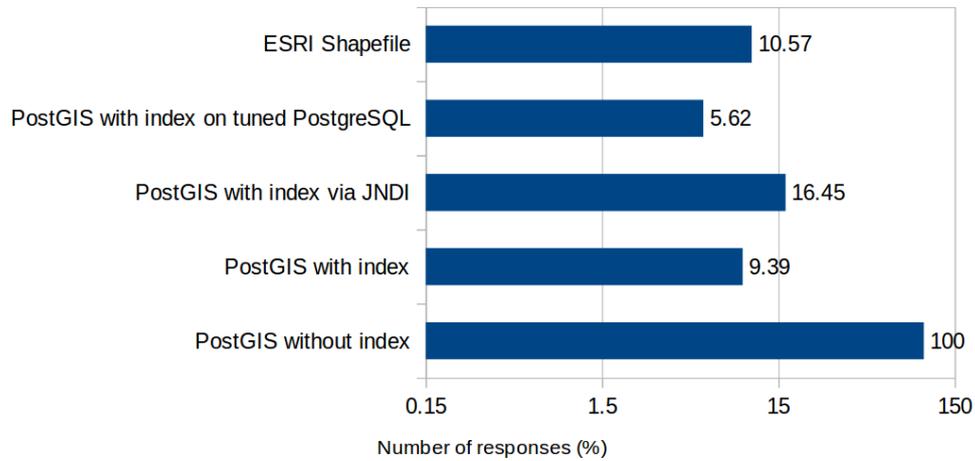


Figure 4: Number of responses higher than 5 s

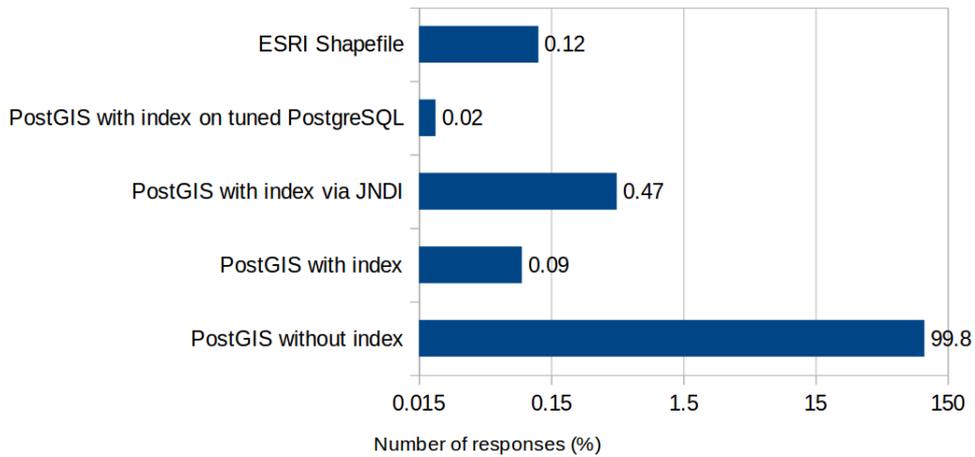


Figure 5: Number of responses higher than 10 s

response (compare 2.8 and 2.4 seconds). There are significant differences in maximum response time (compare 25 s and 12 s). There are significant differences in number of responses over 5 s (compare 10 % and 5 %).

I can conclude that connecting PostgreSQL/PostGIS via JNDI is slower than connecting PostgreSQL/PostGIS without JNDI. The difference is very significant mainly in number of responses over 5 s (compare 16 % and 5 %).

I can conclude that tuning PostgreSQL could speed up WMS (compare 2.7 s and 2.4 s for average response time or 10 % and 5 % for number of responses over 5 s).

I did not tested any other type of index than GIST for PostgreSQL/PostGIS. It may be done in future. I did not tested any other type of index than quad tree index on ESRI Shapefile and I did not tested any than default options for building both indexes.

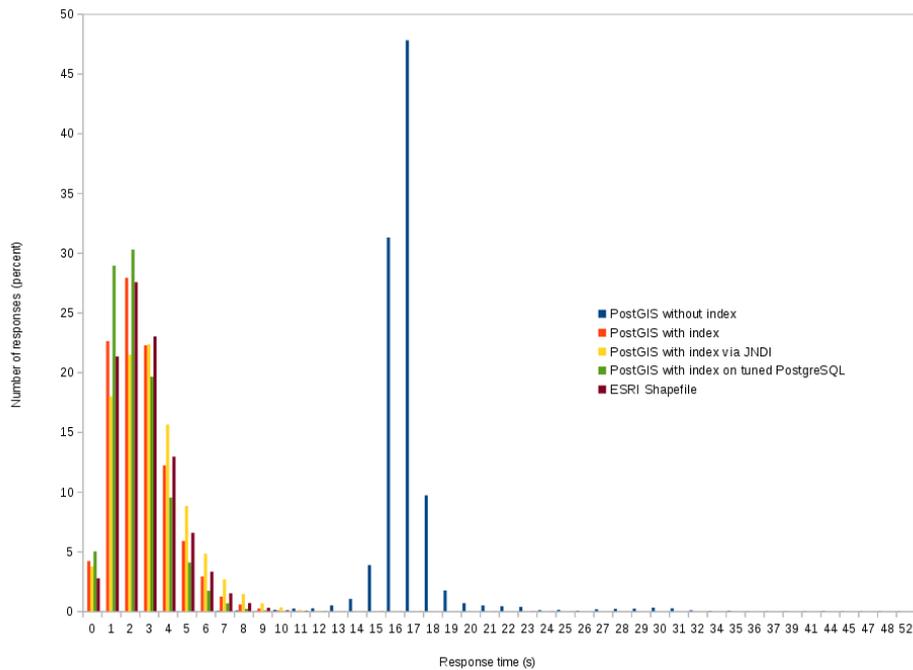


Figure 6: Distribution of time responses

References

- [1] Geoserver.org. *GeoServer*. URL: <http://geoserver.org/>.
- [2] The PostgreSQL Global Development Group. *Performance Optimization - PostgreSQL wiki*. URL: https://wiki.postgresql.org/wiki/Performance_Optimization.
- [3] Jiří Horák, Jan Růžička, and Jiří Ardielli. “Performance Testing of Download Services of COSMC”. In: *Geoinformatics FCE CTU 10* (Nov. 2013), pp. 5–14. DOI: [10.14311/gi.10.1](https://doi.org/10.14311/gi.10.1).
- [4] Environmental Systems Research Institute. *ESRI Shapefile Technical Description*. URL: <https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>.
- [5] Michal Kepka and Jan Ježek. “Web client for PostGIS—the concept and implementation”. In: *Geoinformatics FCE CTU 11* (Dec. 2013), pp. 63–76. DOI: [10.14311/gi.11.5](https://doi.org/10.14311/gi.11.5).
- [6] M. Leslie. *Section 8: Spatial Indexing*. URL: <http://revenant.ca/www/postgis/workshop/indexing.html>.
- [7] MapServer. *Shptree*. URL: <http://mapserver.org/utilities/shptree.html>.
- [8] Michal Med and Petr Souček. “Development and testing of INSPIRE themes Addresses (AD) and Administrative Units (AU) managed by COSMC”. In: *Geoinformatics FCE CTU 11* (Dec. 2013), pp. 77–83. DOI: [10.14311/gi.11.6](https://doi.org/10.14311/gi.11.6).
- [9] OGC/ISO. *Web Map Service*. URL: <http://www.opengeospatial.org/>.
- [10] Oracle. *Java Naming and Directory Interface (JNDI)*. URL: <http://www.oracle.com/technetwork/java/jndi/index.html>.

- [11] Adam Schreier. *Porovnání rychlosti mapového serveru GeoServer při přístupu k různým datovým skladům [online]*. Bachelor's thesis. 2014 [cit. 2016-04-05]. URL: <http://theses.cz/id/jqqz7x/>.
- [12] Pavel Stěhule. "PostGIS pro vývojáře". In: *Geoinformatics FCE CTU 2* (Dec. 2007), pp. 71–90. DOI: [10.14311/gi.2.11](https://doi.org/10.14311/gi.2.11).