

Comparação do desempenho de extensões espaciais de SGBD: PostGIS e SpatiaLite

Rodrigo Evangelista Delgado
Pedro Lucas Lopes Zimmermann
Ivanildo Barbosa

Instituto Militar de Engenharia - IME
Praça General Tibúrcio, 80 - 22290-270 - Rio de Janeiro - RJ, Brasil
rodrigoedelgado@gmail.com, pedro_zimmermann@hotmail.com, ivanildo@ime.eb.br

Abstract. Spatial data has been created at an increasingly rate due to the availability of smaller and cheaper GPS devices. This amount of data makes feasible the spatiotemporal analysis over the behavior of the monitored objects. This paper compares the potentiality of two commonly used spatial extensions of Database Management Systems (DBMS): PostGIS and SpatiaLite. The former is the most known, widely used and studied, by the open-source community, universities and large companies. The latter is said to be an alternative when the database is small and the queries are not very complex. In order to compare the spatial extensions features, a database was generated in each of the DBMS, spatial and non-spatial queries were performed, and the time elapsed to return the results was analyzed. Other criterion to consider for comparing extensions were the storage size, in MB, of both the databases and the functions provided in each one to check if they pass the OGC requirements in terms of spatial functionalities. As a conclusion about the analysis of the main aspects compared, the authors show that SpatiaLite is a viable option in less complex scenarios whilst PostGIS is recommended in applications that demand more rapid processing for larger amounts of data.

Palavras-chave: Spatial DBMS, database performance, OGC SQL functions, Bancos de Dados Espaciais, performance de bancos de dados. Funcionalidades SQL OGC.

1. Introdução

A geração de dados espaciais tem crescido nos últimos anos em virtude da miniaturização e da popularização de dispositivos dotados de receptores GPS. Esse novo volume de dados permite implementar análises sobre o comportamento espaço-temporal dos objetos monitorados.

A criação, o gerenciamento e a manipulação de bancos de dados são implementadas por Sistemas de Gerenciamento de Banco de Dados (SGBD). Um SGBD espacial oferece tipos de dados e consultas espaciais, provendo índices espaciais e algoritmos eficientes para a manipulação da informação geográfica (Güting, 1994).

Na análise de um banco de dados espacial verifica-se que podem existir dados convencionais e dados espaciais. Dados convencionais descrevem as características, e dados espaciais descrevem a localização e a forma geométrica dos objetos espaciais. Objetos espaciais geralmente estão associados a um domínio espacial, seus limites geográficos e são geralmente modelados para representar elementos do mundo real usando formas geométricas simples, tais como pontos, linhas ou polígonos.

O *Open Geospatial Consortium* (OGC) especificou um esquema SQL que suporta o armazenamento, recuperação, consulta e atualização de feições geoespaciais com geometria simples (OGC, 2010). São abrangidas funcionalidades referentes à geometria e ao sistema de referência das coordenadas da própria feição e outras que identificam o relacionamento entre pares de objetos, mesmo que não sejam da mesma natureza.

Atualmente, *PostGIS* e *SpatiaLite*, extensões espaciais do *PostgreSQL* e *SQLite* respectivamente, são exemplos de SGBD espaciais amplamente utilizados que podem ser citados. As duas extensões além de serem compatíveis com os padrões OGC, possuem um grande número de funcionalidades, fácil importação e exportação de dados, integração com SIG (Sistemas de Informação Geográfica) e se diferenciam em características que são o foco do presente estudo.

O objetivo deste artigo é comparar as duas extensões espaciais do *PostgreSQL* e *SQLite*, com base na quantidade de funções suportadas, espaço de armazenamento e velocidade de processamento de consultas SQL alfanuméricas e espaciais. A Seção 2 apresenta a metodologia empregada na avaliação do desempenho, descrevendo a base de dados de teste e as condições de realização dos testes. Os resultados obtidos são apresentados e discutidos na Seção 3. A Seção 4 contém as considerações finais deste trabalho.

2. Metodologia de Trabalho

2.1 Bases de Dados para avaliação

Os dados empregados nesta comparação foram importados de cinco arquivos disponibilizados, em formato CSV (*Comma Separated Values*), pelo Portal de Dados Abertos da Prefeitura Municipal do Rio de Janeiro, contendo 106 Hotéis, 382 Monumentos da cidade, 20 Praias, 18 Museus e 136 Pontos turísticos e culturais.

A Figura 1 ilustra uma amostra da categoria de hotéis, em formato tabular (a) e como mapa (b).

[Baixar](#)

Hotéis

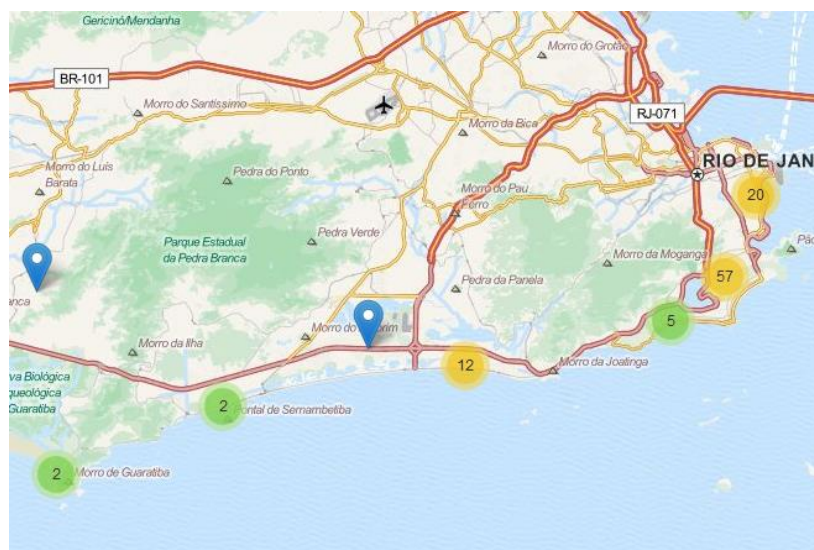
URL: http://dadosabertos.rio.rj.gov.br/api/Turismo/apresentacao/csv/hotelaria__csv

Uma lista dos hotéis da cidade do Rio de Janeiro, com informações sobre localização, categoria, telefone, fax, email, cnpj e também línguas faladas.

Grid Graph Map 106 records 0 - 100 Search data ... Go Filters

Nome	Logrado...	Número	Bairro	Telefone	Fax	Categori...	Email	Latitude	Longitud...	Acomod...	Acomod...
Acapulco ...	Rua Gust...	854	Copacaba...	3077-2000	2275-3396	4	reservas...	-22.964128	-43.173263	2	0
Aeroporto...	Avenida B...	280	Centro	3575-7321	2544-1231	4	jorge.chav...	-22.910842	-43.170983	0	0
Ambassa...	Rua Sena...	25	Centro	2297-7181	2220-4783	3		-22.910786	-43.177045		
American...	Rua Hum...	244	Humaitá	2537-4714	2122-3000	Flat	amerflat...	-22.958998	-43.200804		
Angrense	Travessa ...	25	Copacaba...	2548-0509	2548-0509	2	reserva@...	-22.971645	-43.187833		
Apa	Rua Repú...	305	Copacaba...	2548-8112	2256-3628	3	apa@apa...	-22.966521	-43.182836	0	0
Argentina	Rua Cruz ...	30	Flamengo	2558-7233		3	luzia@arg...	-22.936734	-43.175197	0	0
Arpoador ...	Rua Fran ...	177	Ipanema	2523-0060	2511-5094	4	reservas...	-22.987825	-43.193911	0	0
Astoria C...	Rua Repú...	345	Copacaba...	2545-9090		4	diretoria...	-22.966225	-43.183029	2	0
Atlântico ...	Rua Sigu...	90	Copacaba...	2548-0011	2235-7941	4		-22.966946	-43.186320		
Atlântico ...	Rua Sant...	15	Copacaba...	2549-8090	2549-8090	Flat		-22.972564	-43.186315		
Atlântico ...	Rua Raul ...	94	Copacaba...	2522-0037		Flat		-22.984755	-43.191473		

(a)



(b)

Figura 1. Extrato de dados na categoria de Hotéis.

Como o formato CSV não suporta geometrias, foi necessário construir a geometria do tipo ponto com base nos valores de latitude e longitude presentes nas respectivas colunas.

2.2 Critérios para avaliação do desempenho

As funções especificadas em OGC (2010) para feições do tipo ponto são empregadas na definição de sua geometria, na identificação de relacionamentos espaciais, no cálculo de distâncias, operações de conjuntos e definições de vizinhança (*buffer zone*). O primeiro critério de comparação entre as extensões é o suporte a essas funções.

O segundo critério de comparação é o espaço de armazenamento ocupado pelas tabelas preenchidas com os dados importados dos arquivos.

O terceiro critério é o tempo de execução das consultas envolvendo funcionalidades espaciais:

- Distância entre pontos distintos – Foram calculadas as distâncias entre os hotéis e as praias;
- Latitude e longitude do objeto – Foram consultadas as coordenadas dos hotéis;
- Coordenadas UTM (Fuso 23S) – Foram transformadas as coordenadas da camada de hotelaria.
- Definição de vizinhança de raio 5 km – Foi utilizada como referência a camada de hotéis.

O quarto critério é o tempo de execução das consultas que não envolvem funcionalidades espaciais:

- Exibir os dados existentes em uma tabela – Foi utilizada a tabela de hotelaria.
- Exibir uma tabela obedecendo uma restrição – Foram consultados os hotéis situados em Copacabana.
- Exibir uma tabela ordenada obedecendo uma restrição – Foram consultados os hotéis de Copacabana, ordenados pelo nome.
- Exibir a quantidade de elementos que obedecem uma restrição – Foram consultados quantos hotéis havia em Copacabana.
- Exibir o SRID de uma tabela – Foi consultado o SRID da camada de monumentos.
- Exibir o tipo de geometria de uma tabela – Foi consultado o tipo de geometria da tabela de museus.

Todos os testes foram realizados em um computador pessoal com processador Intel i7 2.20 GHz, com 8.00 GB de memória RAM.

3. Resultados e Discussão

Os resultados da obtidos após a execução dos testes estão apresentados em tabelas.

A Tabela 1 apresenta o tamanho, em *Mb*, dos bancos de dados gerados com os dados importados. Observa-se que o espaço ocupado em disco pela base de dados do Spatialite é menor que o ocupado pela gerada pelo PostGIS. Não foi encontrado um recurso para obter o tamanho ocupado individualmente pelas tabelas no Spatialite. Empregando o aplicativo pgAdmin, é possível obter essa quantidade. Contudo, a criação de um banco de dados espacial implica a necessidade de tabelas adicionais, que também ocupam espaço em disco.

Tabela 1. Tamanho do banco de dados preenchido com os dados importados

Tamanho <i>PostGIS</i> (<i>Mb</i>)	Tamanho <i>Spatialite</i> (<i>Mb</i>)
11	3,7

A Tabela 2 indica o suporte das extensões espaciais às funcionalidades especificadas em OGC (2010). Observa-se que ambas as extensões cumprem com as determinações da OGC e com uma sintaxe bastante semelhante.

As Tabelas 3 e 4 apresentam, respectivamente, o tempo decorrido na execução das consultas espaciais e não-espaciais indicadas na seção 2. Observa-se que o PostGIS apresenta desempenho superior em relação ao Spatialite, porém a execução das consultas em ambas as extensões foi praticamente instantânea, com o volume de dados empregado nos testes.

Tabela 2. Suporte às funcionalidades especificadas em OGC (2010)

Funcionalidade OGC	Sintaxe <i>PostGIS</i>	Sintaxe <i>Spatialite</i>
ST_Dimension	ST_Dimension(<i>geom</i>)	Dimension(<i>geom</i>)
ST_GeometryType	ST_GeometryType(<i>geom</i>)	GeometryType(<i>geom</i>)
ST_AsText	ST_AsText(<i>geom</i>)	AsText(<i>geom</i>)
ST_AsBinary	ST_AsBinary(<i>geom</i>)	AsBinary(<i>geom</i>)
ST_SRID	ST_SRID(<i>geom</i>)	SRID(<i>geom</i>)
ST_IsEmpty	ST_IsEmpty(<i>geom</i>)	IsEmpty(<i>geom</i>)
ST_IsSimple	ST_IsSimple(<i>geom</i>)	IsSimple(<i>geom</i>)
ST_Boundary	ST_Boundary(<i>geom</i>)	Boundary(<i>geom</i>)
ST_Envelope	ST_Envelope(<i>geom</i>)	Envelope(<i>geom</i>)
ST_Equals	ST_Equals(<i>geom1,geom2</i>)	Equals(<i>geom1,geom2</i>)
ST_Disjoint	ST_Disjoint(<i>geom1,geom2</i>)	Disjoint(<i>geom1,geom2</i>)
ST_Intersects	ST_Intersects (<i>geom1,geom2</i>)	Intersects (<i>geom1,geom2</i>)
ST_Touches	ST_Touches(<i>geom1,geom2</i>)	Touches(<i>geom1,geom2</i>)
ST_Crosses	ST_Crosses(<i>geom1,geom2</i>)	Crosses(<i>geom1,geom2</i>)
ST_Within	ST_Within(<i>geom1,geom2</i>)	Within(<i>geom1,geom2</i>)
ST_Contains	ST_Contains(<i>geom1,geom2</i>)	Contains(<i>geom1,geom2</i>)
ST_Overlaps	ST_Overlaps(<i>geom1,geom2</i>)	Overlaps(<i>geom1,geom2</i>)
ST_Relate	ST_Relate(<i>geom1,geom2</i>)	Relate(<i>geom1,geom2</i>)
ST_Distance	ST_Distance(<i>geom1,geom2</i>)	Distance(<i>geom1,geom2</i>)
ST_Intersection	ST_Intersection(<i>geom1,geom2</i>)	Intersection(<i>geom1,geom2</i>)
ST_Difference	ST_Difference(<i>geom1,geom2</i>)	Difference(<i>geom1,geom2</i>)
ST_Union	ST_Union(<i>geom1,geom2</i>)	GUnion(<i>geom1,geom2</i>)
ST_SymDifference	ST_SymDifference(<i>geom1,geom2</i>)	SymDifference(<i>geom1,geom2</i>)
ST_Buffer	ST_Buffer(<i>geom,dist</i>)	Buffer(<i>geom,dist</i>)
ST_ConvexHull	ST_ConvexHull(<i>geom</i>)	ConvexHull(<i>geom</i>)
ST_X	ST_X(<i>pt</i>)	X(<i>pt</i>)
ST_Y	ST_Y(<i>pt</i>)	Y(<i>pt</i>)
ST_Z	ST_Z(<i>pt</i>)	Z(<i>pt</i>)
ST_M	ST_M(<i>pt</i>)	M(<i>pt</i>)

Tabela 3. Tempo decorrido na execução das consultas espaciais

Consulta	Tempo <i>PostGIS</i> (ms)	Tempo <i>Spatialite</i> (ms)
Distância entre pontos distintos	35	47
Latitude e longitude do objeto	31	30
Coordenadas UTM (Fuso 23S)	33	47
Definição de vizinhança de raio 5 km em torno dos monumentos	47	63

Tabela 4. Tempo decorrido na execução das consultas não-espaciais

Consulta	Tempo <i>PostGIS</i> (ms)	Tempo <i>SpatiaLite</i> (ms)
Exibir uma tabela	18	28
Exibir uma tabela obedecendo uma restrição	17	25
Exibir uma tabela ordenada obedecendo uma restrição	12	36
Exibir a quantidade de elementos que obedecem uma restrição	13	36
Exibir o SRID de uma tabela	12	24
Exibir o tipo de geometria de uma tabela	14	21

4. Conclusões

O objetivo do artigo é comparar as duas extensões espaciais do PostgreSQL e SQLite, com base na quantidade de funções suportadas, espaço de armazenamento e velocidade de processamento de consultas SQL alfanuméricas e espaciais.

Foi observado que ambas as extensões oferecem total suporte às funcionalidades especificadas em OGC (2010), porém o número de funções não-especificadas definidas no PostGIS é bastante superior ao do SpatiaLite. Quanto ao espaço de armazenamento, a extensão SpatiaLite ocupou menos espaço em disco, isso se deve principalmente ao fato anteriormente mencionado sobre o número de funções implementadas em ambos. Os valores apresentados nas tabelas 3 e 4 permitem concluir que a extensão PostGIS permite a execução mais rápida de consultas, tanto espaciais como não-espaciais, apesar disso, a resposta do SpatiaLite é bastante rápida (praticamente instantânea) também.

Pode-se concluir portanto que o uso do SpatiaLite em aplicações que não exijam consultas muito complexas e com uma base de dados pequena é justificável, pois a extensão cumpre com o papel desejado de maneira eficaz, ocupando menos espaço em disco quando comparada ao PostGIS.

Referências Bibliográficas

Güting, R. H. An Introduction to Spatial Database Systems. *VLDB Journal*, p.357–399,1994.

Herring, J.R. **OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 2: SQL option**. Disponível em: < <http://www.opengeospatial.org>>. 2010.

PostGIS 1.5 Manual. Disponível em: < <http://postgis.net/docs/manual-1.5/>>. Acesso em: 16.out.2014.

Prefeitura Municipal do Rio de Janeiro, **Portal de Dados Abertos**. Disponível em <<http://data.rio.rj.gov.br>>. Acesso em: 16.out.2014.

SQL Functions Reference List. Disponível em: < <http://www.gaia-gis.it/gaia-sins/spatialite-sql-4.2.0.html>>. Acesso em: 18.out.2014.