Análise Quantitativa sobre o Desempenho na Indexação de *Big Geospatial Data* em Ambiente de Nuvem Computacional

João Bachiega Jr.[1]

[1]Instituto de Ciências Exatas - Departamento de Ciência da Computação - CIC
Universidade de Brasília (UnB) - Brasília - DF - Brasil

joao.bachiega.jr@gmail.com

Abstract. *With the growth of spatial data volume, known as Big Geospatial Data, some tools have been developed to allow the processing of this data in an efficient way, but for this it is fundamental to index the databases. The cloud computing has computational power and several other characteristics that are adherent to the execution of this type of application. This paper presents a quantitative analysis of the performance of indexing task on big geospatial data using SpatialHadoop tool in a test scenario provisioned in a cloud environment.*

Resumo. *Com o crescimento do volume de dados espaciais, conceituado como Big Geospatial Data, algumas ferramentas foram desenvolvidas para permitir o processamento desses dados de forma eficiente, mas para isso é fundamental a indexação das bases de dados. A computação em nuvem possui poder computacional e diversas outras características que são aderentes para a execução deste tipo de aplicação. Este trabalho apresenta uma análise quantitativa sobre o desempenho da indexação de grandes volumes de dados geográficos por meio da ferramenta SpatialHadoop em um cenário de testes provisionado em ambiente de nuvem.*

1. Introdução

O enorme volume de dados geográficos gerados e disponibilizados nos últimos anos, conceituado como *Big GeoSpatial Data*, tem motivado pesquisadores a encontrarem uma solução para o processamento desses dados [1], e também a disponibilização de poder computacional capaz de suprir as necessidades geradas por estas aplicações, o que é encontrado na computação em nuvem, que é um modelo que possibilita acesso sob demanda a um vasto conjunto de recursos computacionais, oferecendo alto poder computacional.

Para que estas aplicações tenham um bom desempenho em relação ao tempo de processamento, uma tarefa importante é a indexação do conjunto de dados. Quando executado em um *cluster* computacional, a quantidade de nós tem papel significativo no tempo de execução desta tarefa.

Neste sentido, este artigo tem o objetivo de fazer uma análise quantitativa sobre o desempenho da tarefa de indexação para grandes volumes de dados geográficos, executados em ambiente de nuvem computacional, realizando comparações entre tipos diferentes de indexações e entre configurações distintas de *cluster*.

Assim, este artigo está estruturado, em mais cinco seções. A Seção 2 apresenta o conceito de *Spatial Cloud Computing*. Na Seção 3 são apresentadas as características

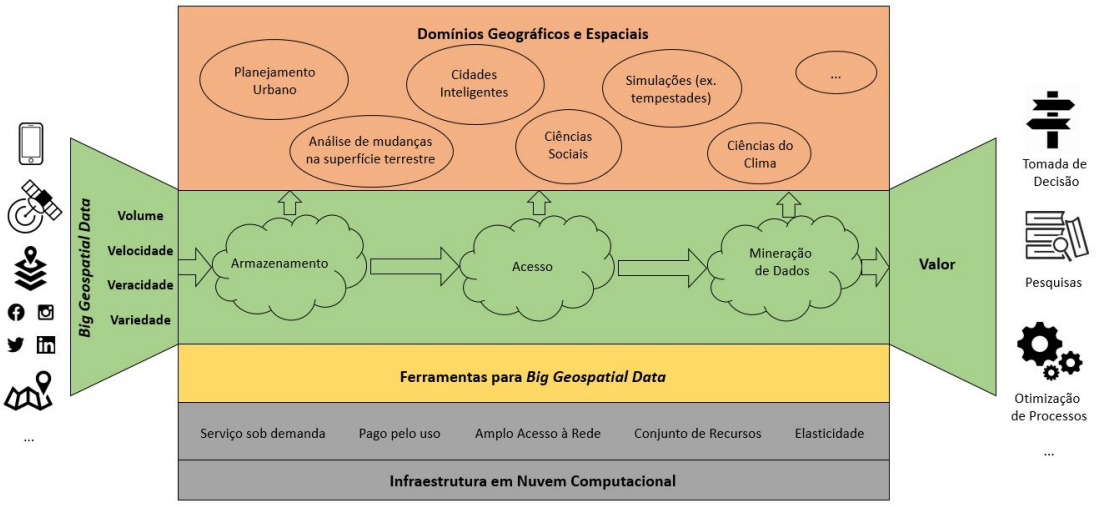

Figura 1. Utilização da Computação em Nuvem para o Processamento de *Big Geospatial Data*, adaptado de [1].

do *SpatialHadoop*. Em seguida, os métodos para indexação são apresentados na Seção 4. A Seção 5 apresenta a análise quantitativa realizada, detalhando os testes executados e resultados obtidos. Por fim, a conclusão deste artigo é apresentada na Seção 6.

2. Spatial Cloud Computing

O termo *Big Geospatial Data* é um paradigma emergente para a grande quantidade de informações geográficas que tem-se disponível a cada segundo devido a utilização de Sistemas de Informações Geográficas (SIG), atingindo *petabytes* de informações a cada dia [2]. Estes dados são gerados das mais diversas formas, tais como em mídias sociais, dispositivos móveis, satélites, entre outros. Além disso, esses dados têm sido gerados de uma maneira cada vez mais acelerada. O desafio de transformar grandes volumes de dados no resultado esperado, exige requisitos de armazenamento, de acesso, de análise e de mineração dos dados.

A Computação em Nuvem é um modelo de entrega de poder computacional em forma de serviço que possui características que permitem o processamento de grandes volumes de dados, tais como a elasticidade para o provisionamento de recursos; o alto poder computacional obtidos através do compartilhamento de recursos; o amplo acesso que permite uma rápida comunicação; a obtenção de recursos de acordo com a demanda; e, por fim, a tarifação baseada apenas nos recursos que foram utilizados [3].

Por todas estas características, alguns autores como Li *et al.* [3] e Yang *et al.* [1], afirmam que a computação em nuvem é bastante aderente ao processamento de *Big Geospatial Data*, definindo o termo *Spatial Cloud Computing* para o ambiente em que a infraestrutura da computação em nuvem é utilizada em prol da geração de valor para domínios geográficos. Assim, como pode ser observado na Figura 1, tendo como insumo um grande volume de dados obtidos das mais variadas fontes, as ferramentas para o processamento de *big geospatial data* se apoiam na infraestrutura da nuvem computacional para armazenamento, acesso e mineração dos dados, com o objetivo de atingir resultados que agreguem valor às organizações, favorecendo a tomada de decisão e a otimização de

processos.

3. SpatialHadoop

O processamento de grandes volumes de dados espaciais tem demandado não apenas recursos computacionais robustos, mas também métodos eficientes. Nos últimos anos, diversas aplicações foram desenvolvidas utilizando os conceitos de *Hadoop* para otimizar o processamento desses dados, tais como: *GIS Tools on Hadoop* [4] e *Hadoop-GIS* [5]. Em [6] foi apresentado o *SpatialHadoop*, um *framework* que está incorporado no *Hadoop*, implementando as funcionalidades espaciais no seu interior e também utilizando índices espaciais. Desta forma, o *SpatialHadoop* tem um bom desempenho quando comparado com as demais aplicações existentes até então.

O núcleo do *SpatialHadoop* consiste em quatro camadas, as quais são [6]:

- Camada de Linguagem: o *SpatialHadoop* utiliza o *Pigeon*, uma linguagem *SQL-like* que suporta os tipos de dados padrões do *Open Geospatial Consortiums* (OGC);
- Camada de Operações: encapsula a implementação de diversas operações espaciais que utilizam os índices espaciais, e os novos componentes na camada de *MapReduce*;
- Camada *MapReduce*: para ser capaz de lidar com arquivos indexados espacialmente, introduz dois novos componentes na camada de *MapReduce* – *SpatialFileSplitter* e *SpatialRecordReader*;
- Camada de Armazenamento: adiciona índices espaciais para superar uma limitação do *Hadoop*, que provê suporte apenas para arquivos não indexados do tipo *heap*. Para isso, organiza o seu índice em dois níveis, indexação global e local.

Assim como todas as aplicações que utilizam o conceito de *MapReduce*, o *SpatialHadoop* é executado em um *cluster* composto por pelo menos um *master node*, responsável pelo gerenciamento das tarefas, e por diversos *datanodes*, que serão os servidores executores das operações.

4. Indexação para *Big Geospatial Data*

O processamento de operações espaciais é fortemente influenciado pelo uso de estruturas de dados e algoritmos de pesquisa conhecidos como Métodos de Acesso Multidimensionais (MAM) [7]. Estes métodos são projetados para atuarem como um caminho otimizado aos dados espaciais com base em um conjunto definido de predicados sobre os atributos. Neste sentido, o espaço indexado é organizado de tal forma que a recuperação dos objetos espaciais contidos em uma área particular requeira apenas o acesso a objetos próximos a esta área, em oposição a análise do conjunto completo de objetos.

Ao longo do tempo, diversas pesquisas foram realizadas no intuito de melhorar as formas de indexação dos dados espaciais. As mais simples são as árvores binárias, como AVL, Red-Black e Splay Tree [8]. Após isto, diversas outras foram propostas, sendo as principais: KD-Tree [9], R-Tree [10], Hilbert R-Tree [11], Grid [12], e R+-Tree [13].

Neste trabalho são comparados os índices espaciais *Grid* (Figura 2 (a)) e *R-Tree* (Figura 2 (b)), ambos suportados pelo *SpatialHadoop* [6]. A indexação *Grid* estende o

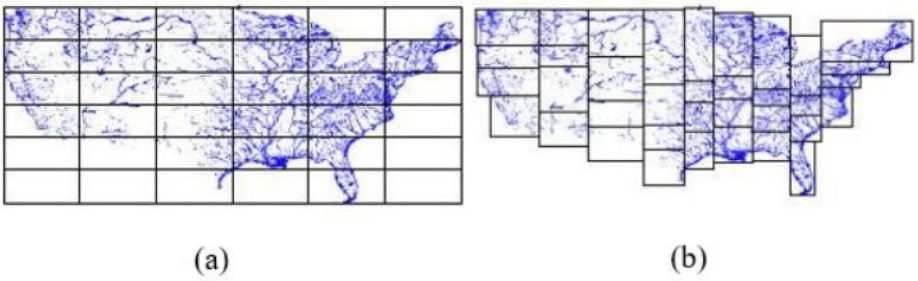

Figura 2. (a) Indexação Grid [6]. (b) Indexação R-Tree [6].

conceito de *Hashing* para duas dimensões, dividindo o espaço segundo uma grade retangular, onde cada célula, correspondente a um *bucket*, é associada a uma página, uma área no disco [12]. A associação é feita por meio do diretório, que é uma matriz bidimensional onde seus elementos possuem o endereço de uma página.

Já a indexação *R-Tree* é uma estrutura hierárquica dinâmica, que utiliza o *Minimum Bounding Retangle* (MBR) para organizar objetos de acordo com a posição espacial e, desta forma, fiquem armazenados próximos uns dos outros. A principal função da *R-Tree* é reduzir o espaço de consulta, descartando os objetos que não fazem parte do predicado espacial selecionado. Um grande benefício proporcionado pela *R-Tree* é que sua estrutura permite que vários nós que não fazem parte do critério de busca sejam descartados, diminuindo o acesso a disco [10].

5. Testes e Resultados

Um ambiente de testes foi configurado no provedor Amazon AWS, utilizando o serviço *Elastic Map Reduce (EMR)* [1], que é oferecido especificamente para a construção de aplicações que processam grandes volumes de dados. Os testes foram realizados em duas configurações de *cluster*: a primeira contendo 1 *master node* e 2 *datanodes* e a segunda configuração com 1 *master node* e 4 *datanodes*. Todos os nós foram configurados com 4 vCPUs e 14Gb de memória. Por simplicidade na implementação, o *EMR File System* [2] foi utilizado como recurso de armazenamento dos dados.

5.1. Bases de Dados

Para a realização dos testes foram utilizados quatro conjuntos de dados, conforme apresentado na Tabela 1.

Tabela 1. *Datasets* Utilizados nos Testes.

Dataset	Conteúdo	Qtde. Registros
BUILDINGS	Contornos de construções no mundo	115 milhões
WAYS	Estradas mapeadas no mundo	164 milhões
RAILS	Ferrovias mapeadas no mundo	181 milhões
OBJECTS	Objetos geográficos mapeados no mundo	263 milhões

[1]https://aws.amazon.com/pt/emr
[2]http://docs.aws.amazon.com/ElasticMapReduce/latest/ReleaseGuide/emr-fs.html

Todas as bases de dados são representações do mundo real e foram extraídas do *OpenStreetMap*. Elas estão disponíveis para download em `http://spatialhadoop.cs.umn.edu/datasets.html`.

5.2. Resultados e Análises Quantitativas

Os experimentos foram realizados com o objetivo de obter o tempo de execução das indexações *Grid* e *R-Tree* nas bases de dados em um *cluster SpatialHadoop* com 2 ou 4 *datanodes*. Os resultados representam a média de cinco execuções. Por ser executado em ambiente de nuvem computacional, os valores são apresentados em minutos, uma vez que esta é a medida de tarifação do provedor utilizado.

Baseado nos resultados obtidos, foram aplicadas 3 técnicas de análise quantitativa de dados, conforme demonstrado na Tabela 2.

Tabela 2. Experimentos Executados.

Técnica Aplicada	**Objetivo**
Intervalo de Confiança	Avaliar se o desempenho das indexações *Grid* e *R-Tree* são significativamente diferentes.
Regressão Linear	Estabelecer um modelo de regressão linear entre o tempo de execução e o tamanho da base de dados.
Projeto 2^2 Fatorial	Determinar qual fator é significativo para o tempo da indexação.

O detalhamento das análises quantitativas aplicadas serão apresentados nas seções a seguir. Todos os dados coletados e os cálculos detalhados estão disponíveis para consulta em `https://github.com/joaobachiegajr/MetQuant`.

5.2.1. Intervalo de Confiança

Um intervalo de confiança é uma amplitude de valores que tem a probabilidade de conter o valor verdadeiro da população [14]. Neste sentido, foram realizados 5 experimentos comparando o tempo de execução das indexações *Grid* e *R-Tree* para a base de dados *RAILS*, que contém 181 milhões de registros, utilizando uma configuração de *cluster* com 4 *datanodes*. A Tabela 3 apresenta os resultados coletados.

Tabela 3. Resultados Coletados para Intervalo de Confiança.

Experimento	**Grid (minutos)**	**R-Tree (minutos)**
1	56	56
2	58	57
3	56	56
4	59	57
5	55	56

Aplicando as definições e fórmulas recomendadas em [14] e os dados obtidos apresentados na Tabela 4, é possível afirmar, com 90% de certeza, que não há diferença entre as indexações considerando esta base de dados e esta configuração de *cluster*.

Tabela 4. Resultados Obtidos para IC de 90%.

Métrica	***Grid***	***R-Tree***
Média	56,80	56,40
Desvio Padrão	1,64	0,55
Diferença das médias	0,40	
Desvio padrão das diferenças (s)	0,77	
IC	0,40 $\pm$ 1,65	

Embora o índice de confiança a ser utilizado recomendado por Jain [14] seja entre 90% e 95%, mas com o intuito apenas de validar os resultados obtidos, foram repetidos os cálculos considerando um intervalo de confiança de 50% e, ainda assim, o zero estava contido no intervalo, o que reafirma que as indexações não são significativamente diferentes.

5.2.2. Regressão Linear

A regressão linear quantifica a relação entre uma ou mais variáveis preditoras e uma variável de resultado [14]. Sendo assim, foi realizada uma análise de 5 observações do tempo de execução da indexação *Grid* em um *cluster* com 2 *datanodes*, para todas as bases de dados descritas na Seção 5.1, conforme apresentado na Tabela 5.

Tabela 5. Resultados Coletados para Regressão Linear.

Base de Dados	Registros em milhões	Experimentos (em seg)				
		1	2	3	4	5
Buildings	115	51	50	51	50	50
Ways	164	76	76	76	77	75
Rails	181	79	79	80	80	79
Objects	263	86	87	87	87	87

A aplicação dos métodos para regressão linear objetivaram estabelecer um modelo entre o tempo de execução da indexação e a quantidade de registros da base de dados geográfica. Os resultados obtidos para a indexação *Grid* são apresentados abaixo:

$n = 4$
$\overline{x} = 180,75$
$\overline{y} = 73,15$
$\Sigma xy = 55460,00$
$\Sigma x^2 = 142051,00$
$\Sigma x = 723,00$
$\Sigma y = 292,60$
$\Sigma y^2 = 22155,00$
$b_1 = 0,23$
$b_0 = 32,25$
$SST = 751,07$
$SSE = 169,04$
$R2 = 0,77$
$MSE = 84,52$
$se = 9,19$
$t = 1,886$
$s_{b_0} = 16,25$
$s_{b_1} = 0,09$

Com isso, temos que o intervalo a 90% para s_{b_0} é $(1,61; 62,90)$ e o intervalo a 90% s_{b_1} é $(0,06; 0,39)$. Logo, o modelo desejado, isto é, o *tempo de indexação* dada a

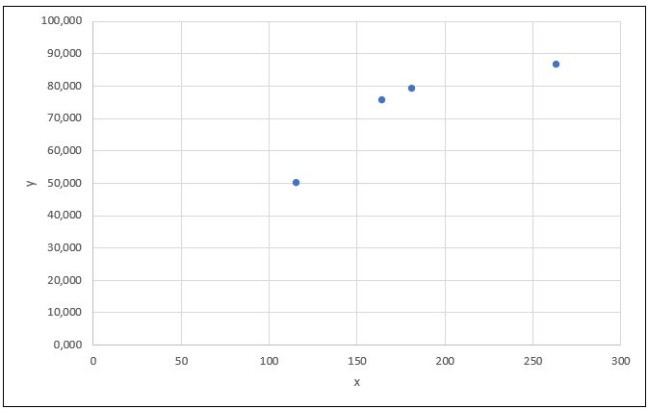

Figura 3. Teste de Comportamento Linear.

quantidade de registros é:

$$\text{Tempo de Indexação} = 32,25 + 0,23 \times \text{Quantidade de Registros} \quad (1)$$

Desta forma, com 90% de confiança, é possível afirmar que os dois parâmetros são significativos, dado que os intervalos de confiança para s_{b_0} e s_{b_1} não contém o valor 0.

No entanto, ao gerar um gráfico para realizar a inspeção visual do modelo, conforme apresentado na Figura 3, foi possível verificar que os dados não seguem um comportamento linear e, consequentemente, o modelo é inadequado para o conjunto de dados.

Com o objetivo de definir um modelo linear, foi realizada uma transformação logaritmica. Desta forma, os valores considerados para a análise foram os apresentados na Tabela 6.

Tabela 6. Resultados Transformados por log.

Base de Dados	Registros em milhões	Experimentos				
		1	2	3	4	5
Buildings	115	1,71	1,70	1,71	1,70	1,70
Ways	164	1,88	1,88	1,88	1,89	1,88
Rails	181	1,90	1,90	1,90	1,90	1,90
Objects	263	1,93	1,94	1,94	1,94	1,94

Ainda assim, o gráfico se manteve sem um comportamento linear. Foi realizado, então, o teste de independência no qual foi possível constatar, baseado na tendência parabólica, que existe uma dependência entre o erro e as várias de predição, conforme apresentado na Figura 4.

5.2.3. Projeto 2^2 Fatorial

O projeto fatorial é uma técnica bastante utilizada quando se tem duas ou mais variáveis independentes, também chamadas de fatores [14]. Para este trabalho, esta técnica será

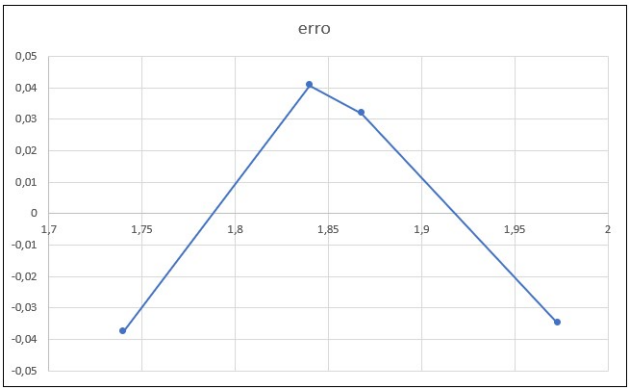

Figura 4. Teste de Independência.

utilizada para determinar qual fator é significativo para o tempo da indexação de uma base de dados geográficas.

Considerando as bases utilizadas neste trabalho, foi considerado um volume *baixo* de registros para as bases de dados que possuem até 180 milhões de registros. A caracterização *alta* foi atribuída para as bases de dados que possuem mais que 180 milhões de registros.

O primeiro fator (X_A) será a quantidade de *datanodes*, que pode assumir os valores -1 para *2 datanodes* e 1 para *4 datanodes*. O segundo fator (X_B) refere-se a quantidade de registros em uma base de dados, atribuindo-se -1 para *Baixo* e 1 para *Alto*.

A Tabela 7 foi construída para o desenvolvimento da técnica 2^2 Fatorial.

Tabela 7. Projeto 2^2 Fatorial.

	2 Datanodes (-1)	4 Datanodes (+1)
Baixo (-1)	59	40
Alto (+1)	90	56

Considerando estes valores, as equações resultaram nos seguintes valores:

$q_0 = 61,25$
$q_A = -13,25$
$q_B = 11,75$
$q_{AB} = -3,75$

Por fim, temos que o fator (X_A), que representa o número de datanodes, é o que tem o maior impacto ($\pm$ 13, 25).

De fato, o aumento de datanodes em um cluster SpatialHaddop tem impacto direto no tempo de execução da indexação das bases de dados, conforme já demonstrado em outros trabalhos [15], [16], [17].

6. Conclusão

A indexação tem papel fundamental no desempenho de aplicações que processam *Big Geospatial Data*, uma vez que é a tarefa que exige maior poder computacional e tempo

de processamento. No entanto, considerando o escopo dos dados utilizados neste trabalho, e considerados os testes realizados e a análise quantitativa desenvolvida, foi possível observar que, não existem diferenças significativas entre as indexações *Grid* e *R-Tree*.

Além disso, não foi possível estabelecer um modelo linear que seja capaz de estimar o tempo da indexação de uma base de dados geográfica considerando a quantidade de registros a ser processado. No entanto, utilizando uma análise do Projeto Fatorial 2^2, foi possível observar que a variação da quantidade de *datanodes* em um *cluster SpatialHadoop* é significativo para o tempo de execução da indexação.

Para trabalhos futuros é sugerida a execução de mais experimentos, variando ainda mais as bases de dados, o volume de registros de cada base e ainda, configurações mais diversificadas de *cluster*. Com mais dados disponíveis, análises quantitativas mais adequadas e robustas podem ser executadas, gerando resultados significativos que contribuam para o estado da arte.

Referências

[1] Chaowei Yang, Manzhu Yu, Fei Hu, Yongyao Jiang, and Yun Li. Utilizing cloud computing to address big geospatial data challenges. *Computers, Environment and Urban Systems*, 61:120–128, 2017.

[2] Ahmed Eldawy and Mohamed F Mokbel. The era of big spatial data. In *Data Engineering Workshops (ICDEW), 2015 31st IEEE International Conference on*, pages 42–49. IEEE, 2015.

[3] Ang Li, Xiaowei Yang, Srikanth Kandula, and Ming Zhang. Cloudcmp: comparing public cloud providers. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 1–14. ACM, 2010.

[4] Eric Hoel and Mike Park. Big data: Using arcgis with apache hadoop. *Esri International Developer Summit*, 2014.

[5] Ablimit Aji, Fusheng Wang, Hoang Vo, Rubao Lee, Qiaoling Liu, Xiaodong Zhang, and Joel Saltz. Hadoop gis: a high performance spatial data warehousing system over mapreduce. *Proceedings of the VLDB Endowment*, 6(11):1009–1020, 2013.

[6] Ahmed Eldawy and Mohamed F Mokbel. Spatialhadoop: A mapreduce framework for spatial data. In *Data Engineering (ICDE), 2015 IEEE 31st International Conference on*, pages 1352–1363. IEEE, 2015.

[7] Ricardo Rodrigues Ciferri. Análise da influência do fator distribuição espacial dos dados no desempenho de métodos de acesso multidimensionais. *Tese de Doutorado em Ciencia da Computacao. Centro de Informatica. Universidade Federal de Pernambuco*, 2002.

[8] Volker Gaede and Oliver Günther. Multidimensional access methods. *ACM Computing Surveys (CSUR)*, 30(2):170–231, 1998.

[9] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[10] Antonin Guttman. R-trees: A dynamic index structure for spatial searching. *SIGMOD international conference on Management of data*, 14(2), 1984.

[11] Ibrahim Kamel and Christos Faloutsos. Hilbert r-tree: An improved r-tree using fractals. *International Conference on Very Large Databases (VLDB)*, 1993.

[12] Jürg Nievergelt, Hans Hinterberger, and Kenneth C Sevcik. The grid file: An adaptable, symmetric multikey file structure. *ACM Transactions on Database Systems (TODS)*, 9(1):38–71, 1984.

[13] Timos Sellis, Nick Roussopoulos, and Christos Faloutsos. The r+-tree: A dynamic index for multi-dimensional objects. *International Conference on Very Large Databases (VLDB)*, 1987.

[14] Raj Jain. *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*. John Wiley & Sons, 1990.

[15] Joao Bachiega, Marco Reis, Aleteia Araujo, and Maristela Holanda. A cost-efficient method for big geospatial data on public cloud providers. *The Ninth International Conference on Advanced Geographic Information Systems, Applications, and Services*, pages 25–31, 2017.

[16] Joao Bachiega, Marco Reis, Aleteia Araujo, and Maristela Holanda. Cost optimization on public cloud provider for big geospatial data. *Proceedings of the 7th International Conference on Cloud Computing and Services Science*, pages 54–62, 2017.

[17] Joao Bachiega, Marco Reis, Aleteia Araujo, and Maristela Holanda. An architecture for cost optimization in the processing of big geospatial data in public cloud providers. *IEEE International Congress on Big Data*, pages 190–197, 2018.