UNIVERSIDADE FEDERAL DE JATAÍ (UFJ)
INSTITUTO DE CIÊNCIAS EXATAS E TECNOLÓGICAS (ICET)
CURSO DE CIÊNCIA DA COMPUTAÇÃO

Ahmed Subhi Sousa

Distribuição de dados para processamento de multijunções espaciais usando o método *Gain-Loss*

Jataí, Goiás

2024

Ahmed Subhi Sousa

Distribuição de dados para processamento de multijunções espaciais usando o método *Gain-Loss*

Monografia apresentada ao curso de Ciência da Computação do Instituto de Ciências Exatas e Tecnológicas da Universidade Federal de Jataí (UFJ), como requisito para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Thiago Borges de Oliveira

Jataí, Goiás

2024

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFJ.

Sousa, Ahmed Subhi
Distribuição de dados para processamento de multijunções espaciais usando o método Gain-Loss / Ahmed Subhi Sousa. - 2024.
42 f.: il.

Orientador: Prof. Dr. Thiago Borges de Oliveira.
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Jataí, Instituto de Ciências Exatas e Tecnológicas, Ciência da Computação, Jataí, 2024.
Bibliografia.
Inclui siglas, abreviaturas, símbolos, gráfico, tabelas, lista de figuras, lista de tabelas.

1. Multijunção espacial. 2. Sistemas distribuídos. 3. Distribuição de Dados. 4. Gain-Loss. I. Oliveira, Thiago Borges de, orient. II. Título.

CDU 004

UNIVERSIDADE FEDERAL DE JATAÍ
INSTITUTO DE CIÊNCIAS EXATAS E TECNOLÓGICAS

DECLARAÇÃO DE APROVAÇÃO DA VERSÃO FINAL

Declaro que o(a) discente Ahmed Subhi Sousa do curso de Bacharelado em Ciência da Computação foi aprovado(a) na defesa do Trabalho de Conclusão de Curso (TCC) com o título final Distribuição de dados para processamento de multijunções espaciais usando o método Gain-Loss na data de 13/12/2024 e efetuou todas as correções pertinentes sugeridas pela banca examinadora, composta pelo seguintes membros:

Orientador(a)	Thiago Borges de Oliveira
Membro 1	Italo Tiago da Cunha
Membro 2	Flávio Ferreira Borges

Declaro ainda que a versão final anexada a este processo está adequada para ser devidamente depositada em repositório institucional.

Observação

Esta declaração deve ser assinada pelo(a) orientador(a)

Documento assinado eletronicamente por **THIAGO BORGES DE OLIVEIRA**, **Professor do Magistério Superior**, em 18/12/2024, às 19:44, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufj.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0372160** e o código CRC **68C8BB2C**.

Dedico este trabalho à minha família que sempre me apoiou.

Agradecimentos

Agradeço ao meu orientador, professor Thiago Borges de Oliveira, pela orientação, paciência e dedicação ao longo deste trabalho. Agradeço também à minha família, pelo apoio e pela confiança em minha jornada. Ao professor Flávio Ferreira Borges e à empresa PhD Sistemas, que proporcionaram os recursos necessários para a realização dos experimentos em nuvem na plataforma Azure.

"Você nunca sabe que resultados virão da sua ação. Mas se você não fizer nada, não haverá resultados."
(Mahatma Gandhi)

Resumo

Este trabalho analisa a eficiência do método *Gain-Loss* (GL) para distribuição de dados espaciais em sistemas distribuídos, utilizando o sistema DGEO. O problema está na complexidade e nos altos custos de processamento associados às consultas de multijunção espacial em grandes volumes de dados espaciais, evidenciando o uso de *clusters* de computadores. Nesse contexto, o objetivo foi investigar a eficiência do GL em comparação com o método *Round-Robin* (RR), considerando tempo de execução e consumo de rede em um sistema distribuído com conjuntos de dados reais. A metodologia envolveu a execução de sete consultas espaciais sobre um cluster com quatro máquinas virtuais, utilizando *datasets* reais e três métodos de escalonamento diferentes. Os resultados mostraram que o GL superou o RR em consultas no tempo de execução, mas apresentou desempenho inferior quanto ao tráfego de rede, ainda enfrentando desafios na otimização das consultas.

Palavras-chaves: *Multijunção espacial; Sistemas distribuídos; Distribuição de Dados; Gain-Loss.*

Abstract

This study analyzes the efficiency of the Gain-Loss (GL) method for spatial data distribution in distributed systems, utilizing the DGEO system. The problem lies in the complexity and high processing costs associated with multiway spatial join queries over large volumes of spatial data, highlighting the need for computer clusters. In this context, the objective was to investigate the efficiency of GL in comparison with the Round-Robin (RR) method, considering execution time and network consumption in a distributed system with real datasets. The methodology involved executing seven spatial queries on a cluster with four virtual machines, using real datasets and three different scheduling methods. The results showed that GL outperformed RR in query execution time but exhibited inferior performance in terms of network traffic, still facing challenges in query optimization.

Keywords: *Multiway Spatial Join; Distributed systems; Data Distribuition; Gain-Loss.*

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

CPU	*Central Processing Unit*
DistGeo	Plataforma de Geoprocessamento Distribuído de Operações Espaciais
DGEO	Sistema Distribuído de Processamento Geográfico
DHCP	*Dynamic Host Configuration Protocol*
GEOS	*Geometry Engine - Open Source*
GHz	GigaHertz
GL	*Gain-Loss*
GPS	Sistema de Posicionamento Global (*Global Positioning System*)
GR	*Greedy Algorithm*
IBGE	Instituto Brasileiro de Geografia e Estatística
ICET	Instituto de Ciências Exatas e Tecnológicas
LAPIG	Laboratório de Processamento de Imagens e Geoprocessamento
LP	*Linear Programming*
LR	*Langrangian Relaxation*
LTS	*Long Term Support*
Mb/s	Megabit por segundo
MB	Megabyte
RAM	*Random Access Memory*
RPM	Método do Ponto de Referência (*Reference Point Method*)
RR	*Round-Robin*
RSL	Revisão Sistemática de Literatura
SBDE	Sistemas de Banco de Dados Espaciais
SHP	*Shapefile*
SIG	Sistema de Informação Geográfica

SOL	SBC OpenLib
SSH	*Secure Shell*
STR	*Sort-Tile-Recursive*
UFG	Universidade Federal de Goiás
UFJ	Universidade Federal de Jataí

Sumário

1 Introdução

1.1 Motivação

Devido à popularização e acessibilidade de dispositivos como *smartphones*, satélites, *drones*, veículos agrícolas e demais tecnologias que integram o uso do Sistema de Posicionamento Global (GPS), houve um aumento significativo de dados espaciais a serem processados nos últimos anos (OLIVEIRA et al., 2023). O processamento desses dados espaciais é realizado, frequentemente, com consultas espaciais, as quais fazem uso de algoritmos computacionais complexos. Tais algoritmos demandam tempo de execução relevante, seja por sua complexidade ou pelo volume de dados. Para reduzir o tempo de processamento das consultas, estudos tem adotado o processamento distribuído utilizando *clusters* de computadores (BACELLAR, 2010).

Um tipo de consulta espacial que tem sido bastante estudada em sistemas de banco de dados espaciais (SBDE) é a junção espacial, que consiste na operação de combinar objetos espaciais de dois *datasets* [1](JACOX; SAMET, 2007), que satisfaçam um predicado espacial – como sobreposição, interseção ou proximidade. Além da junção, a multijunção espacial também é uma das consultas espaciais de maior relevância, por lidar com algoritmos ainda mais complexos, devido ao fato de combinar objetos de três ou mais *datasets* distintos, também com base em um predicado espacial específico (OLIVEIRA; COSTA; RODRIGUES, 2015). Por exemplo, utilizando *datasets* de cobertura vegetal, produtividade do solo e níveis de nutrientes, a junção espacial realizada a partir de um predicado de interseção poderia resultar em áreas onde seja necessário uma correção de solo, ou áreas com potencial para produção.

Devido à complexidade dessas consultas e ao tamanho dos *datasets*, é necessário particionar os dados em várias máquinas (ELDAWY et al., 2021). O uso de *clusters* de computadores em sistemas distribuídos para o processamento de consultas espaciais permite a distribuição do custo de processamento entre diversas máquinas, otimizando o uso de recursos computacionais, reduzindo significativamente o tempo de execução das consultas, como também melhorando a eficiência e a escalabilidade do sistema, tornando-o capaz de lidar com volumes crescentes de dados espaciais de maneira mais eficaz, de modo a aumentar o poder computacional sem crescer demasiadamente os custos financeiros, pois se o processamento fosse realizado em apenas uma máquina, seria necessário um hardware mais potente capaz de lidar com o processamento (BACELLAR, 2010).

Os algoritmos de distribuição de dados são importantes para as consultas espaciais, tendo em vista que afetam o tempo de processamento e a utilização da rede. Logo, é

[1] *Datasets* são um conjunto de dados.

importante que se partilhe os dados de forma balanceada entre os nós do *cluster*, para que todos os recursos sejam utilizados de maneira uniforme, evitando processamentos intensos. Porém, também é essencial que diminua a necessidade do uso da banda de rede.

A colocalização usando a localização inerente do dados espaciais pode ajudar nesse sentido, já que, nas junções espaciais, um servidor do *cluster* precisa de todos os dados úteis para a consulta. Caso um dado espacial necessário não esteja localmente em um servidor, é preciso que servidores façam uma troca de informações. Entretanto, estudos recentes comprovaram que a colocalização afeta, em certo grau, o tempo de execução (OLIVEIRA et al., 2013; OLIVEIRA et al., 2023). Ou seja, um *cluster* extremamente balanceado em relação aos dados precisará de troca de informações entre os servidores, aumentando o uso da rede e a latência do início do processamento; já com os dados colocalizados espacialmente, utilizará menos da banda de rede e terá menor latência, porém os servidores poderão ficar desbalanceados em relação ao processamento.

Nesse sentido, o trabalho de Tonon e Oliveira (2019) propôs o método *Gain-Loss* de distribuição de dados, baseado em algoritmos da árvore R^0, desenvolvida por Xia e Zhang (2005), com objetivo de equilibrar a colocalização e o balanceamento. Contudo, foram feitos testes em ambiente controlado, entre este método em comparação com outros. Apesar de não ter sido encontrado um ponto de equilíbrio entre paralelismo e colocalização, obteve-se resultados satisfatórios com um balanceamento regular e uma redução do uso de recursos computacionais.

O Sistema Distribuído de Processamento Geográfico (DGEO) (OLIVEIRA, 2017) é uma aplicação para processar dados espaciais de forma distribuída, desenvolvida na Universidade Federal de Goiás, em linguagens C e Go, aprimorada e mantida na UFJ desde então. O DGEO emprega a biblioteca GEOS para processar os predicados das consultas de junção espacial, e os recursos da linguagem Go para as processar de forma paralela e distribuída.

O método *Gain-Loss* foi integrado no sistema distribuído DGEO (OLIVEIRA, 2017), no contexto do projeto de pesquisa intitulado Desenvolvimento e Integração de Estruturas de Dados e Algoritmos para Processamento Eficiente da Multijunção Espacial em Sistemas Distribuídos, projeto PI02366-2018, desenvolvido no Curso de Bacharelado em Ciência da Computação/ICET/UFJ.

1.2 Objetivo do Trabalho

Esse trabalho teve como objetivo investigar a eficiência do método *Gain-Loss* de distribuição de dados espaciais, quanto ao tempo de execução das consultas e o volume de comunicação da rede em um sistema distribuído com conjuntos de dados reais (DGEO).

Os objetivos específicos foram:

- Elaborar consultas de multijunção espacial distribuídas envolvendo *datasets* reais;
- Configurar um *cluster* de computadores para executar o sistema DGEO;
- Executar os experimentos;
- Tabular e analisar os resultados dos experimentos para concluir sobre a eficiência do método *Gain-Loss*.

1.3 Contribuição do Trabalho

A principal contribuição deste trabalho foi a avaliação da eficiência do método *Gain-Loss* (GL) em um ambiente distribuído real, utilizando o DGEO para executar consultas de multijunção espacial. Diferentemente de estudos anteriores, que utilizaram *datasets* sintéticos e ambientes controlados, este trabalho aplicou o método GL a *datasets* reais. A análise dos resultados experimentais forneceram informações sobre vantagens e desvantagens do GL em consultas diferentes.

1.4 Organização da Monografia

O trabalho está dividido em quatro capítulos, descritos resumidamente a seguir: O Capítulo 2 apresenta o referencial teórico, contextualizando os conceitos utilizados. No Capítulo 3 aborda os trabalhos relacionados, explicando os critérios de busca e análise, apresentando um resumo dos trabalhos selecionados que se destacaram, seguido de uma comparação entre eles. No Capítulo 4 detalha a metodologia utilizada na pesquisa, explicando como os experimentos foram conduzidos, além de apresentar os resultados obtidos, com análises sobre o tempo de execução e o volume de dados trafegados na rede das consultas realizadas. Por último, o Capítulo 5 apresenta as conclusões da pesquisa e uma breve descrição dos trabalhos futuros.

2 Referencial Teórico

2.1 Introdução

Neste capítulo são apresentados os conceitos para melhor compreensão do trabalho. A seção 2.2 aborda a definição de dados espaciais, sua representação e aplicação em consultas espaciais, além do uso de Sistemas de Informação Geográfica (SIG). A seção 2.3 apresenta o conceito de processamento distribuído de dados espaciais, essenciais para lidar com multijunções espaciais. A seção 2.4 apresenta métodos de distribuição de dados espaciais. Por fim a seção 2.5 explora sobre o DGEO, uma aplicação desenvolvida para realizar consultas espaciais de forma paralela e distribuída, integrando os métodos de distribuição.

2.2 Dados Espaciais

Os dados espaciais, sinônimo de dados geográficos ou geoespaciais, indicando sua relação com objetos próximos ou na superfície da Terra, são uma representação de objetos geográficos de interesse do mundo real, como ruas, edifícios, lagos e países, juntamente com suas localizações (FITZ, 2018). Esses dados são considerados multidimensionais e complexos, exigindo técnicas especializadas para sua manipulação em bancos de dados espaciais e podem ser representados espacialmente, permitindo a visualização e análise de informações de maneira gráfica, através de imagens, mapas temáticos e planos de informações (OLIVEIRA, 2017). A estrutura desses dados pode ser do tipo matricial, dividindo o espaço em células, ou vetorial, como demonstra a Figura 1, na qual o espaço possui pontos representados pelos cruzamentos, as linhas pelas ruas e os polígonos pelos lotes. Apesar da imagem na figura ser um objeto matricial, refere-se aqui aos objetos vetoriais que possibilitaram a construção da imagem.

Os dados espaciais são capazes de representar qualquer referência espacial que se deseja estudar e analisar, por exemplo, uma fazenda, um planeta, um ecossistema ou componentes de uma placa de circuito impresso. Independentemente da referência espacial, os dados podem ser convertidos para uma representação planar (HUISMAN; BY et al., 2009).

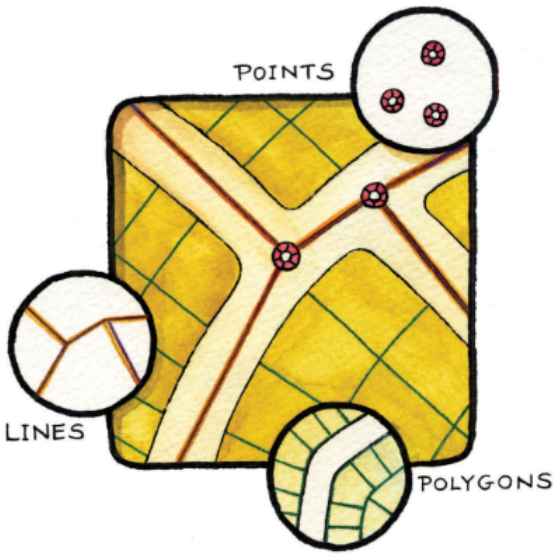

Figura 1 – Representação vetorial de pontos, linhas e polígonos.

Fonte: (CAMPBELL; SHIN, 2012)

2.2.1 Consultas Espaciais

Para entender o conceito de consultas espaciais é importante conhecer sobre análise espacial. A análise espacial é uma manipulação de dados com o objetivo de valorizá-los, apoiar decisões e para revelar padrões e variações que não estavam evidentes, ou seja, transforma dados espaciais brutos em informações espaciais úteis (LONGLEY et al., 2005).

Nesse contexto, as consultas espaciais são métodos fundamentais para extrair informações de um *dataset*. Elas permitem comparar *datasets* distintos para obter informações adicionais e retornam um conjunto de objetos espaciais (por exemplo, objetos de ponto, linha e região) que satisfazem um determinado predicado topológico (por exemplo, cruza, sobrepõe, dentro, encontra) considerando um objeto de pesquisa. (OLIVEIRA, 2017; CARNIEL, 2018)

Três tipos importantes de consultas espaciais são:

1. Consulta de janela: uma operação em que uma seleção geométrica é aplicada a um único *dataset*, resultando em um conjunto de objetos que sobrepõem uma região ou janela específica, geralmente retangular. A Figura 2 ilustra uma consulta de janela, onde uma área retângular é selecionada em um *dataset* que representa parte de uma cidade;

2. Junção Espacial: é uma operação de sobreposição aplicada a dois *datasets*, onde um objeto espacial de um *dataset* é combinado com um objeto do outro *dataset* se suas geometrias satisfizerem um predicado espacial, como interseção ou cobertura. A Figura 3 representa uma junção espacial, combinando um *dataset* ilustrado pelo

Figura 2 – Consulta de janela.

Fonte: (FRANÇA, 2018)

retângulo particionado com um *dataset* retratado pelo círculo também particionado. Essa junção satisfaz a um predicado espacial de intersecção, permitindo identificar os objetos que se sobrepõem entre as duas geometrias; e

Figura 3 – Junção espacial.

Fonte: (CAMPBELL; SHIN, 2012)

3. Multijunção Espacial: é uma junção espacial que envolve um número arbitrário de *datasets*, agrupados em pares, onde cada par possui um predicado espacial específico.

2.2.2 Sistema de Informação Geográfica

Um Sistema de Informação Geográfica (SIG) é um tipo de programa de computador utilizado para organizar, analisar, visualizar e compartilhar dados de diferentes períodos históricos e em diversas escalas. Ele é composto por hardware, software, dados e peopleware, sendo uma ferramenta útil em várias áreas, desde climatologia, epidemiologia e arqueologia até planejamento urbano e consultoria política (CAMPBELL; SHIN, 2012).

A Figura 4 representa o conceito de SIG utilizando a metáfora de um bolo em camadas, onde cada camada corresponde a um tema geográfico específico. Essas camadas são organizadas de maneira empilhada e, juntas, formam uma representação detalhada da realidade espacial, facilitando a análise e compreensão dos dados. A camada mais inferior

representa o mundo real, que é a base de todo o sistema, incluindo elementos físicos como rios, montanhas, construções e terrenos. Acima dessa camada, a de uso da terra demonstra como o espaço é utilizado, indicando áreas urbanas, agrícolas, florestais ou industriais.

Seguindo para cima, encontramos a camada de elevação, que representa as variações de relevo e altitude do terreno. Em seguida, há a camada de parcelas, que subdivide o espaço em unidades menores, como lotes ou propriedades. Logo acima, a camada de ruas mostra as vias de circulação. A camada de clientes ocupa o topo da estrutura, destacando a distribuição de pontos específicos, como a localização de pessoas.

Essa organização em camadas permite ao SIG integrar informações de diferentes naturezas e analisar as relações espaciais entre elas. Por exemplo, é possível observar a influência da elevação no uso do solo ou a proximidade entre ruas e os locais de interesse, como clientes ou edifícios. Dessa forma, o modelo de camadas não apenas facilita a visualização dos dados, mas também possibilita análises complexas ao vincular informações espaciais e atributos específicos de cada objeto.

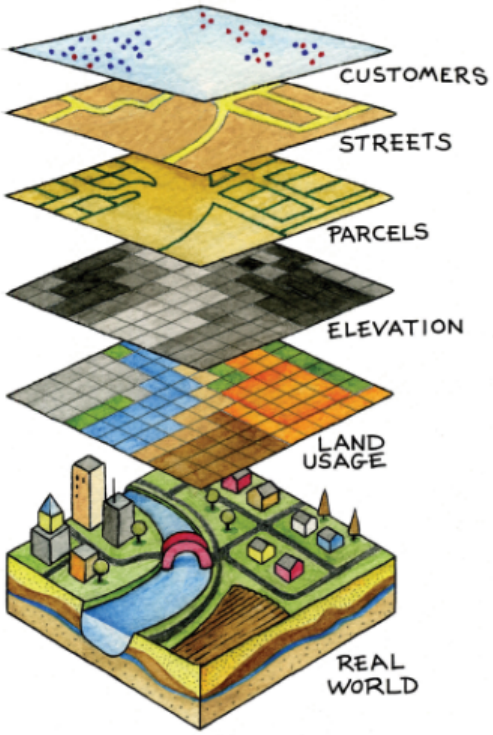

Figura 4 – SIG como um bolo de camadas.

Fonte: (CAMPBELL; SHIN, 2012)

As aplicações de um SIG abrangem áreas como fenômenos climáticos, humanos, sociais e econômicos, entre outros. No planejamento urbano, por exemplo, um SIG pode ser utilizado para mapear e analisar a expansão da cidade, identificar áreas de risco e planejar infraestrutura (FITZ, 2018). A título de exemplo de empregabilidade, em 2011, houve um terremoto no Japão e os SIGs ajudaram socorristas nas operações de resgate,

mapear áreas e infraestruturas severamente danificadas, priorizar necessidades médicas e a localizar abrigos temporários (CHANG, 2019).

2.3 Processamento Distribuído de dados espaciais

O processamento distribuído de dados espaciais é uma abordagem essencial para lidar com a grande quantidade de dados envolvida em consultas de multijunção espacial. Usar apenas um computador para esse tipo de processamento pode ser inviável devido à alta utilização da capacidade de processamento. Em vez disso, a utilização de *clusters* de computadores permite que as consultas sejam realizadas de forma distribuída, reduzindo o tempo de execução das consultas espaciais complexas (OLIVEIRA, 2017).

A Figura 5 ilustra uma rede distribuída onde um usuário interage com um sistema composto por vários nós, cada um com seu próprio sistema operacional e aplicação. Esses nós, identificados como *n1, n2, n3, ..., nk*, formam um cluster escalável, onde estão conectados por uma rede interna que facilita a comunicação, a transferência de dados e o processamento paralelo. O nó mestre gerencia e coordena as tarefas, atuando como intermediário entre o usuário e os nós escravos, distribuindo as solicitações e mantendo comunicação com a rede interna para o processamento e com a rede externa, que conecta o nó mestre ao usuário.

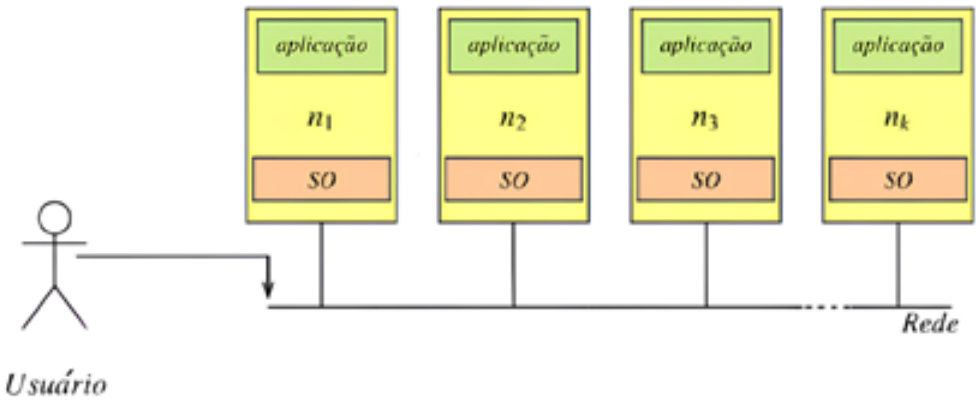

Figura 5 – Exemplo da organização de um *cluster*

Fonte: (FRANÇA, 2018)

Esse processo pode ser dividido em três etapas: Particionamento, Distribuição e Escalonamento. Cada uma dessas etapas impacta na outra, conforme discute-se a seguir.

Na etapa de Particionamento, os dados espaciais são divididos em partes menores, conhecidas como partições. Esse processo é crítico, pois o objetivo é criar partições que sejam balanceadas, evitando sobrecarregar um único nó do *cluster*. Um particionamento eficiente leva em consideração a distribuição espacial dos dados, tentando minimizar a

sobreposição, o que pode afetar negativamente o desempenho da consulta. A sobreposição de objetos, ocorre quando um objeto é dividido entre várias partições, resultando em replicações que aumentam o tamanho do conjunto de dados e podem gerar resultados duplicados (ZEIDAN; VO, 2022).

A Figura 6 compara dois métodos de particionamento. O STR, proposto por Leutenegger, Lopez e Edgington (1997), organiza os dados em blocos, apresentando uma taxa de utilização inferior a 90%. Já o R*-Grove, proposto por Eldawy et al. (2021), otimiza o uso do espaço, alcançando taxas de utilização superiores a 95

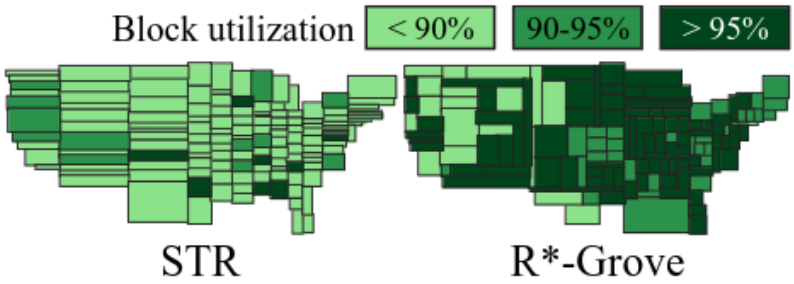

Figura 6 – Métodos de particionamento STR e R*-Grove

Fonte: (ELDAWY et al., 2021)

A etapa de Distribuição envolve a alocação dessas partições entre os nós do *cluster*. Nesta fase, métodos como *Round-Robin* (MUTENDA; KITSUREGAWA, 1999), *Proximity Area* (OLIVEIRA et al., 2013) e *Gain-Loss* (TONON; OLIVEIRA, 2019) são utilizados para determinar a melhor maneira de distribuir as partições, pois a escolha do método de distribuição impacta diretamente no balanceamento da carga e na eficiência do processamento.

Um dos principais problemas enfrentados na etapa de distribuição de dados espaciais é a duplicação de resultados (DITTRICH; SEEGER, 2000). Este problema ocorre quando diferentes nós do *cluster* relatam os mesmos resultados de junção espacial devido à replicação dos dados entre as partições. Dittrich e Seeger (2000) propuseram uma solução para o problema de duplicação de resultados chamada Método do Ponto de Referência (RPM). Esse método consiste em identificar as células que poderiam potencialmente relatar resultados duplicados e permitir que apenas uma delas o faça.

A etapa do escalonamento possui como objetivo o escalonamento da execução das consultas de junção espacial de forma que garanta que a carga das tarefas sejam distribuídas de maneira balanceada entre os nós e minimize o uso da rede. Porém, a alocação uniforme de tarefas pode aumentar a comunicação entre os nós, impactando negativamente o uso da rede, e o oposto também, pois quando há um baixo uso da rede pode resultar em uma alocação desigual das tarefas (OLIVEIRA, 2017).

Logo, cada uma dessas etapas impactam diretamente uma nas outras: um particionamento pouco eficaz pode resultar em uma distribuição ineficiente, que por sua vez pode dificultar um escalonamento equilibrado.

2.4 Métodos de distribuição de dados espaciais

Métodos de distribuição de dados espaciais são técnicas utilizadas para distribuir um conjunto de dados espaciais. Em um ambiente distribuído esses métodos são essenciais para otimizar a execução de consultas espaciais, garantindo eficiência na manipulação de grandes volumes de dados geoespaciais.

2.4.1 Round-Robin

O método mais comum e simples de distribuição de partições em um ambiente distribuído é o *Round-Robin*. Este método distribui as partições de forma uniforme entre os nós do *cluster*, promovendo um balanceamento de carga eficiente. A distribuição é feita de maneira alternada, seguindo uma lista circular de nós, o que garante que cada nó receba aproximadamente a mesma quantidade de partições. No entanto, uma desvantagem significativa do método *Round-Robin* é que ele não considera a colocalização das partições. Isso significa que objetos espaciais que se sobrepõem frequentemente são alocados em nós diferentes, o que requer a replicação e transferência desses objetos pela rede para a execução de junções espaciais (OLIVEIRA, 2017).

2.4.2 Proximity Area

O método de distribuição *Proximity Area* (OLIVEIRA et al., 2013), busca colocalizar objetos espaciais próximos para reduzir a comunicação na rede durante operações de junção espacial. Este algoritmo apresenta um parâmetro de balanceamento, onde valores entre 0 e 1 permitem ajustar a intensidade com que os objetos são atraídos para determinados nós do *cluster*. Com um valor baixo, como k = 0,1, o *Proximity Area* busca distribuir os objetos de forma mais uniforme entre os nós (dados balanceados em volume/quantidade). Em contrapartida, com um valor alto, como k = 0,9, o método permite distribuições que priorizam a colocalização.

2.4.3 Árvore R^0

A árvore R*, proposta por Beckmann et al. (1990), pode ser usada como um método de acesso em sistemas de banco de dados que organizam tanto pontos multidimensionais quanto dados espaciais. Incorporando novos conceitos baseados na redução da área, margem e sobreposição dos retângulos de diretório, a árvore R* foi muito robusta contra distribuições de dados desfavoráveis.

A árvore R^0 é uma estrutura aprimorada da árvore R*, proposta por Xia e Zhang (2005), para melhorar a eficiência das operações espaciais. Essa árvore incorpora técnicas que manipulam *outliers*, que são objetos localizados longe dos outros ou que têm uma grande extensão espacial, para otimizar o desempenho das consultas espaciais.

2.4.4 Gain-Loss

O método de distribuição *Gain-Loss* é uma técnica desenvolvida para otimizar a distribuição de dados espaciais em ambientes de *cluster*, baseada nas métricas de ganho e perda da árvore R^0. Para alocar um novo objeto, o método *Gain-Loss* avalia a perda de inserção em cada nó e escolhe aquele que resulta na menor perda. Além disso, para manter o paralelismo e o balanceamento de carga, o método implementa uma fórmula que determina a quantidade mínima de objetos por nó, com base no número total de objetos e nós disponíveis. Se um nó ultrapassa o limite de objetos, os objetos são redistribuídos para um novo nó. Para fazer isso, é avaliado o conjunto de objetos que maximiza o ganho ao ser removido. O conjunto de objetos que resulta no maior ganho é escolhido, removido do nó original e realocado em um nó vazio. Em situações onde todos os nós estão ocupados, a inserção é realizada no nó que resulta em menor perda (TONON; OLIVEIRA, 2019).

2.5 DGEO

Sistema Distribuído de Processamento Geográfico, ou simplesmente DGEO, é uma aplicação de processamento distribuído de dados espaciais, desenvolvida inicialmente em um projeto de pesquisa na Universidade Federal de Goiás (OLIVEIRA, 2017) e, posteriormente, aprimorado e mantido no contexto do projeto de pesquisa intitulado Desenvolvimento e Integração de Estruturas de Dados e Algoritmos para Processamento Eficiente da Multijunção Espacial em Sistemas Distribuídos, projeto PI02366-2018, desenvolvido no Curso de Bacharelado em Ciência da Computação/ICET/UFJ. Implementada utilizando as linguagens de programação C e Go, a aplicação faz uso da biblioteca GEOS (*Geometry Engine - Open Source*) para realizar consultas de junção espacial de maneira paralela e

distribuída.

O principal objetivo do DGEO é gerenciar eficientemente consultas espaciais distribuídas e executar a junção ou multijunção espacial, utilizando dados reais, configurações de consulta desejadas e características específicas do *cluster*. A aplicação é capaz de estimar os custos associados às consultas, retornando valores de custo de processamento da CPU e a quantidade de dados transmitidos pela rede.

Atualmente, o sistema DGEO pode utilizar dois algoritmos de distribuição de dados: o *Round-Robin*, que distribui os dados de maneira uniforme sem considerar a localização espacial dos objetos e o *Gain-Loss*, que é baseado na análise dos ganhos e perdas associados à distribuição dos dados (TONON; OLIVEIRA, 2019).

Para avaliar o desempenho dos diferentes algoritmos de distribuição de dados na aplicação DGEO, este trabalho focou na comparação entre os algoritmos *Round-Robin* e *Gain-Loss*. O algoritmo *Round-Robin* foi utilizado como *baseline*, por ser uma abordagem tradicional e amplamente utilizada. Em contrapartida, o algoritmo *Gain-Loss*, baseado na análise dos ganhos e perdas associados à distribuição dos dados, foi analisado para verificar seu desempenho em relação ao *baseline*. O algoritmo *Proximity Area*, conforme demonstrado no artigo de Tonon e Oliveira (2019), ficou fora do escopo deste trabalho devido ao seu desempenho inferior e ao tempo limitado da pesquisa, que impede sua implementação no DGEO.

3 Trabalhos relacionados

3.1 Introdução

Neste capítulo, são apresentados os critérios de busca adotados para selecionar os trabalhos relacionados a este estudo, assim como os critérios de análise utilizados para identificar os mais relevantes. Além disso, é fornecido um breve resumo dos trabalhos selecionados, destacando suas contribuições e abordagens. Por fim, é realizado um comparativo entre esses trabalhos, demonstrando semelhanças, diferenças e a evolução das soluções propostas no contexto da pesquisa.

3.2 Critérios de busca

Para o levantamento bibliográfico apresentado neste texto, foi realizada uma revisão narrativa de literatura. As buscas foram realizadas em artigos selecionados pelo grupo de pesquisa, que já havia conduzido uma Revisão Sistemática de Literatura (RSL) inicial. Além disso, foram utilizadas fontes de pesquisa como Google Acadêmico e SOL (SBC OpenLib), com destaque para o Google Acadêmico, que apresentou resultados mais relevantes. As *strings* de busca utilizadas foram "distribuição de dados espaciais", "consulta de multijunção" e "sistema distribuído", bem como "algoritmo de distribuição dos dados espaciais".

3.3 Metodologia de análise

Para analisar os trabalhos relacionados a este estudo, os seguintes critérios foram adotados:

3.3.1 Multijunção Espacial

O primeiro critério analisa artigos que abordam sobre consultas de multijunção espacial, fundamental para a análise e interpretação dos dados espaciais, visto que todos os resultados do experimentos realizados dependem desse tipo de consulta.

3.3.2 Sistemas Distribuídos

O segundo critério escolhido são trabalhos que tratam de sistemas distribuídos, devido ao grande volume de dados dos *datasets* e à complexidade das consultas. Essa abordagem permite repartir os custos de processamento entre várias máquinas, e a utilização de *cluster* permite esse processo.

3.3.3 Gain-Loss

O terceiro critério avalia se os trabalhos implementaram ou analisaram o método de distribuição *Gain-Loss*. Desenvolvido para otimizar a distribuição de dados espaciais em clusters, o *Gain-Loss* equilibra colocalização e balanceamento, reduzindo o custo computacional e otimizando o uso da rede durante o processamento de consultas espaciais.

3.3.4 Dados Reais

O último critério verifica se os experimentos realizados no trabalho utilizaram *datasets* reais em vez de dados sintéticos. A utilização de dados reais é necessário para validar a aplicabilidade e a eficiência dos métodos propostos em cenários práticos.

3.4 Trabalhos analisados

Com base nos critérios apresentados anteriormente, foram analisados 22 trabalhos relacionados dos quais foram selecionados 4 que se destacaram.

3.4.1 Processamento Distribuído de Operações de Junção Espacial com Bases de Dados Dinâmicas para Análise de Informações Geográficas

Oliveira et al. (2013) propôs um novo método de distribuição de dados chamado *Proximity-Area*, desenvolvido para otimizar o processamento de consultas de junção espacial distribuídas em *datasets* dinâmicos e volumosos, colocalizando objetos espaciais próximos para reduzir a comunicação na rede durante operações de junção espacial. Os experimentos foram feitos na Plataforma de Geoprocessamento Distribuído de Operações Espaciais (DistGeo), um sistema de geoprocessamento distribuído baseado em comunicação *peer-to-peer*, em um ambiente de *cluster* com dados espaciais reais.

O método *Proximity-Area* é analisado tanto em termos de balanceamento quanto de colocalização de dados, sendo comparado ao método *Round-Robin*. Os resultados mostraram melhorias no desempenho com redução no tráfego de rede e do tempo de execução.

3.4.2 Efficient Processing of Multiway Spatial Join Queries in Distributed Systems

A tese de (OLIVEIRA, 2017) aborda sobre o processamento de consultas de multijunção espacial em sistemas distribuídos. Uma de suas contribuições é o desenvolvimento do DGEO, uma aplicação para processar dados espaciais de forma distribuída, com o principal objetivo de gerenciar eficientemente consultas espaciais distribuídas e executar a junção ou multijunção espacial, utilizando dados reais, configurações de consultas desejadas e características específicas do *cluster*.

Além disso, o trabalho apresenta três métodos de escalonamento implementados no DGEO: *Linear Programming* (LP), *Lagrangian Relaxation* (LR) e *Greedy Algorithm* (GR). Esses métodos buscam reduzir significativamente o consumo de recursos no processamento das consultas. Enquanto o LP e GR são recomendados para cenários onde pequenas consultas ou consultas *ad-hoc* são predominantes, o LR oferece o melhor equilíbrio entre eficiência e qualidade, sendo mais indicado para consultas complexas e de grande volume.

Os experimentos realizados com o DGEO demonstraram que a aplicação é capaz de lidar com grandes volumes de dados e realizar consultas de multijunção espacial de forma eficiente, além de estimar os custos a essas consultas, retornando valores de custo de processamento da CPU e a quantidade de dados transmitidos pela rede.

3.4.3 Métodos de Distribuição de Dados para Processamento Distribuído de Multijunções Espaciais

Tonon e Oliveira (2019) propuseram o método *Gain-Loss* de distribuição de dados, baseado em algoritmos da árvore R^0, desenvolvida por Xia e Zhang (2005), com objetivo de equilibrar a colocalização e o balanceamento em consultas de multijunção espacial em sistemas distribuídos. Os experimentos foram feitos em ambiente controlado, não distribuído, entre o método proposto, *Gain-Loss*, comparado ao *Round-Robin* e *Proximity-Area*. Os resultados mostraram que o método GL apresentou uma redução do uso de recursos computacionais, principalmente uso de rede e tempo de processamento, apesar de não ter sido encontrado um ponto de equilíbrio entre paralelismo e colocalização.

3.4.4 Beast: Scalable Exploratory Analytics on Spatio-temporal Data

Neste trabalho de Eldawy et al. (2021) é apresentado o *Beast*, um sistema para realizar análises exploratórias em grandes dados espaço-temporais. A proposta oferece três operações para a exploração de dados, sendo elas a consulta de intervalo, junção espacial e estatísticas zonais. O *Beast* fornece uma técnica eficiente de particionamento chamada R*-Grove, que distribui as partições de maneira balanceada entre os nós do *cluster*, e ainda oferece suporte a outras técnicas populares de particionamento, como *Grid, STR* e *Kd-tree*. A escolha do melhor particionador depende das características do conjunto de dados, distribuição e requisitos de consulta, para isso é utilizado de aprendizagem profunda que seleciona automaticamente uma técnica de particionamento adequada que promete a melhor métrica de otimização.

O sistema é baseado em aprendizado profundo para selecionar automaticamente o particionador mais eficiente que promete a melhor métrica de otimização, visando otimizar o desempenho das análises espaço-temporais. Os experimentos com dados em grande escala mostraram uma evidência clara da escalabilidade do sistema.

3.5 Resumo Comparativo

Analisando os trabalhos relacionados, é possível observar que todos contribuem para a área de distribuição de dados espaciais em sistemas distribuídos. Com objetivos de proporem soluções para um melhor balanceamento de carga, colocalização de dados e consequentemente uma melhor eficiência no processamento de consultas espaciais, utilizando de protótipos para chegarem nessas soluções. Os trabalhos de Oliveira et al. (2013), Tonon e Oliveira (2019) e Eldawy et al. (2021) abordam sobre métodos de distribuição de dados em sistemas distribuídos, como o *Proximity-Area*, *Gain-Loss* e o R*-Grove, que buscam uma redução do uso da rede, tempo de processamento e escalabilidade, onde os dois primeiros são capazes de processar consultas de multijunção e o terceiro por enquanto apenas consultas de junção espacial. Já o trabalho de Oliveira (2017) aborda sobre o desenvolvimento do DGEO e métodos de escalonamento em sistemas distribuídos, conseguindo gerenciar consultas com diferentes métodos de distribuição e escalonamento.

Este trabalho diferencia-se por avaliar a eficiência do método *Gain-Loss* em um ambiente distribuído com *datasets* reais, algo não explorado por Tonon e Oliveira (2019), que utilizaram dados sintéticos e ambientes controlados. Em contrapartida, Eldawy et al. (2021) e Oliveira et al. (2013) utilizaram dados reais, mas não exploraram o método GL, tornando este trabalho um complemento para a validação do método em um ambiente distribuído real.

A Tabela 1 apresenta uma comparação entre os trabalhos relacionados e este trabalho, de acordo com os critérios análise: Multijunção Espacial, Sistemas Distribuídos, *Gain-Loss* e Dados Reais. Observa-se que os trabalhos analisados contribuem de diferentes maneiras para o campo do processamento de dados espaciais. Enquanto Oliveira et al. (2013) e Eldawy et al. (2021) se destacam pelo uso de sistemas distribuídos com dados reais, eles não abordam o método *Gain-Loss*, que é o foco central do trabalho de Tonon e Oliveira (2019), embora este tenha sido avaliado em um ambiente controlado com dados sintéticos. A tese de Oliveira (2017) se diferencia por integrar multijunções espaciais em sistemas distribuídos, mas sem utilizar o método *Gain-Loss*.

Tabela 1 – Comparativo entre trabalhos

	Multijunção Espacial	Sistemas Distribuídos	*Gain-Loss*	Dados Reais
Oliveira et al. (2013)	✗	✓	✗	✓
(OLIVEIRA, 2017)	✓	✓	✗	✓
Tonon e Oliveira (2019)	✓	✓	✓	✗
Eldawy et al. (2021)	✗	✓	✗	✓
Este trabalho	✓	✓	✓	✓

4 Avaliação e Testes

4.1 Introdução

Neste capítulo, detalha-se a metodologia empregada na pesquisa, incluindo o ambiente de execução e suas configurações, os dados espaciais utilizados, as consultas espaciais executadas, o tamanho do *cluster* e como foi conduzida a avaliação dos experimentos. Em seguida, são discutidos os resultados obtidos, com análises sobre o tempo de execução e o volume de dados trafegados na rede das consultas realizadas.

A metodologia utilizada é baseada na metodologia já empregada para utilizar esse tipo de sistema, conforme os trabalhos de Oliveira (2017), e outras monografias desenvolvidas no âmbito do projeto de pesquisa, como Pita (2017). Trata-se de uma metologia consolidada e suficiente para estabelecer as métricas necessárias para para concluir sobre a hipótese, conforme detalhada a seguir.

4.2 Configurações do ambiente de execução

Os experimentos foram executados em um *cluster* constituído por 4 máquinas virtuais da plataforma Azure, interligadas por uma rede virtual exclusiva, com capacidade de 12500 Mb/s. Devido a limitações de cota na plataforma, que não puderam ser alteradas a tempo dos experimentos, foram configurados dois tipos de máquinas virtuais, com as características descritas na Tabela 2, todas as 4 com o sistema operacional Ubuntu Server 24.04 LTS, versão de 64 bits. Para conectar ao cluster, utilizou-se um notebook convencional, que não participa do experimento, exceto pela conexão remota ao *cluster*.

Tabela 2 – Descrição do *hardware* do *cluster*.

Item	**Características**
Quantidade de Máquinas	4
CPU das máquinas 1 e 2	AMD EPYC 7763 64-Core Processor 2.45 GHz
CPU das máquinas 3 e 4	AMD EPYC 9V74 80-Core Processor 3.7 GHz
Cache das máquinas 1 e 2	2 MB
Cache das máquinas 3 e 4	4 MB
RAM	8 GB
Sistema Operacional	Ubuntu Server 24.04 LTS

Para viabilizar a realização dos experimentos, todas as máquinas foram configuradas com o protocolo SSH, usando a ferramenta de acesso múltiplo OpenSSH. Essa ferramenta

viabilizou a troca de informações entre as máquinas do *cluster* através de uma interface de linha de comando (*gnome-terminal*), viabilizando o envio de comandos diretos ou múltiplos às máquinas do *cluster* via acesso remoto, com autenticação por chaves privadas.

Para executar o algoritmo, foi necessária a instalação, em todas as máquinas, da aplicação DGEO. A aplicação precisa das bibliotecas libgdal, libgeos++ e libglib2.0. As bibliotecas foram instaladas em todas as máquinas do *cluster*, além das atualizações de pacotes e outras configurações de rede.

4.2.1 Dados utilizados

Para avaliar a proposta, foi selecionado um conjunto de *datasets* reais, obtidos no site do IBGE, do Laboratório LAPIG do Instituto de Estudos Sócio-Ambientais da UFG e do *Digital Chart of the World*. Os *datasets* utilizados são diversificados, e suas características são apresentadas na Tabela 3. Foram incluídos *datasets* diversificados, incluindo polígonos representando vegetação, municípios, alertas de desmatamento no cerrado, culturas e represas de água, além de linhas que representam rodovias, hidrografia, ferrovias, contorno de relevo e hidrografia mundial. Cada *dataset* possui características específicas, como o número de objetos espaciais (Cardinalidade) e o tamanho dos arquivos SHP em *megabytes* (MB).

O formato SHP é utilizado para armazenar dados geoespaciais vetoriais. Ele consiste em três arquivos principais: o arquivo .shp, que contém a geometria dos objetos espaciais (pontos, linhas ou polígonos); o arquivo .shx, que serve como um índice para a geometria; e o arquivo .dbf, que armazena os atributos dos objetos em formato tabular.

Tabela 3 – *Datasets*

Nome	**Sigla**	**Tipo**	**Cardinalidade**	**Tamanho SHP(MB)**
Vegetação	V	Polígonos	2.140	4,7
Municípios	M	Polígonos	5.564	38,8
Alertas desmat. cerrado	A	Polígonos	32.578	11,2
Rodovias	R	Linhas	51.646	15,2
Hidrografia	H	Linhas	226.963	64,5
Culturas	C	Polígonos	123.746	69,3
Ferrovias	F	Linhas	194.261	28,7
Represas de água	RA	Polígonos	338.860	136,7
Contorno de relevo	CR	Linhas	703.574	572,5
Hidrografia Mundial	HM	Linhas	943.638	243,2

4.2.2 Consultas espaciais

Os experimentos incluíram as consultas listadas na Tabela 4. Cada uma dessas consultas formam uma multijunção espacial real, incluindo todos os *datasets* na Tabela 3. O predicado espacial de cada junção é a interseção com o primeiro *dataset*. O símbolo ⋈ foi empregado para denotar a operação de junção espacial. Cada linha da tabela apresenta uma consulta composta por múltiplas junções espaciais, e a coluna "Característica Principal" descreve o tipo de cada consulta, destacando se a junção envolve, por exemplo, polígonos e linhas, ou se os *datasets* são pequenos e colocalizados. Já a coluna "Cardin. Junção" informa a cardinalidade da junção, ou seja, o número de resultados obtidos da consulta.

Tabela 4 – Consultas espaciais

Sigla	Consulta	Característica Principal	Cardin. Junção
Q1	F ⋈ HM ⋈ C	Junção com polígonos e linhas	2.313
Q2	A ⋈ HM ⋈ F ⋈ C	Conjunto resultante seletivo	168
Q3	HM ⋈ F ⋈ CR	Maiores *datasets* de linhas	2.659
Q4	HM ⋈ CR ⋈ RA	Os três maiores *datasets*	771
Q5	V ⋈ A ⋈ M	*Datasets* pequenos, colocalizados	36.479
Q6	HM ⋈ CR ⋈ F ⋈ R	Todos *datasets* de linhas	3.139
Q7	RA ⋈ C ⋈ A ⋈ M	Todos *datasets* de polígonos	26.128

4.2.3 Resumo da Avaliação

Durante os experimentos, foram capturadas as seguintes métricas:

- Tempo de execução da consulta em milissegundos;
- Volume de comunicação da rede em *kilobytes*, mensurada na entrada de fluxo na interface de rede de cada máquina.

4.2.4 Execução da aplicação

Com a aplicação DGEO já instalada em todas as máquinas do *cluster*, a mesma foi executada em cada uma dessas máquinas. Em cada experimento a ser testado, todos os processos associados às aplicações foram interrompidos e reiniciados para evitar o uso repetido de dados armazenados em cache do sistema operacional, da aplicação ou do *hardware*.

Para garantir maior confiabilidade dos resultados e evitar distorções com máquinas vizinhas na plataforma, foram realizadas sete execuções para cada consulta. Dessa forma, foram realizadas e reiniciados sete testes para cada consulta mencionada na Tabela 4, descartando o menor e o maior valor e calculando a média entre os cinco restantes, registrando-a conforme a metodologia empregada por Oliveira, Costa e Rodrigues (2015).

A execução da aplicação consistiu em dois passos:

1. Executar a aplicação nos nós;
2. Executar a aplicação na máquina cliente, que inicia a execução da consulta.

A execução inicial da aplicação DGEO ocorreu ao inicializar o arquivo dgeo no terminal com o comando: `./dgeo`. O servidor ficou em estado de espera até que o cliente começou a processar os dados enviados pela aplicação. O argumento `-intf=p4p1` determina a interface de rede que será analisada em relação ao volume de dados trafegados na rede. Sem essa configuração, a interface de *loopback* é definida como a interface padrão de medição.

A próxima etapa consistiu em executar o aplicativo do cliente no terminal com o comando: `./cclient Q1.txt`. Executar o arquivo `./cclient` permite que o cliente se comunique com os servidores para transferir os dados da consulta escolhida, que nesse caso é a consulta Q1. Os dados dos *datasets* a serem processados estão nas instruções do arquivo `Q1.txt`.

Os experimentos foram realizados com os seguintes argumentos: `MW_OPTMETHOD`, para escolher um método de particionamento, que pode ser `BS`, atualmente chamado de `GR` (*Greedy Algorithm*), `LP` (*Linear Programming*), `LR` (*Lagrangian Relaxation*); e `MW_DISTR_METHOD`, para escolher um método de distribuição, podendo ser `RR` (*Round-Robin*) e `GL` (*Gain-Loss*). Por padrão, o método de distribuição utilizado é o `RR`, e o escalonador é o `LP`.

4.3 Análise dos Resultados Obtidos

Nesta seção apresenta uma análise detalhada dos resultados obtidos através dos experimentos. Foram comparados os métodos de distribuição espacial *Gain-Loss* e *Round-Robin*, utilizando diferentes algoritmos de escalonamento (LR, LP e GR), quanto ao tempo de execução das consultas e o volume de comunicação da rede. Os dados obtidos foram apresentados por meio de gráficos, com o objetivo de facilitar a análise e interpretação dos resultados.

Analisando os gráficos da Figura 7, que comparam o tempo de execução em

milissegundos entre os métodos RR e GL, é possível observar que o método GL apresentou um desempenho superior em 71.43% das consultas quando utilizado em conjunto com os métodos LR e LP. Em contrapartida, quando comparado com o método RR, utilizando o GR, o GL obteve um desempenho inferior em 14,29% das consultas.

O método GL apresentou um desempenho consideravelmente inferior nas consultas combinado com o GR. Segundo Oliveira et al. (2023), o método GR é recomendado para consultas menores e *ad-hoc*. Isso explica os resultados melhores obtidos pelo GR nas consultas Q2 e Q5, que possuem *datasets* pequenos e são mais seletivas. Porém, como a maioria das consultas envolvem *datasets* maiores, o GR se mostrou inferior aos outros métodos, comprometendo tanto o tempo de processamento quanto o volume de dados trafegados na rede na maior parte das situações.

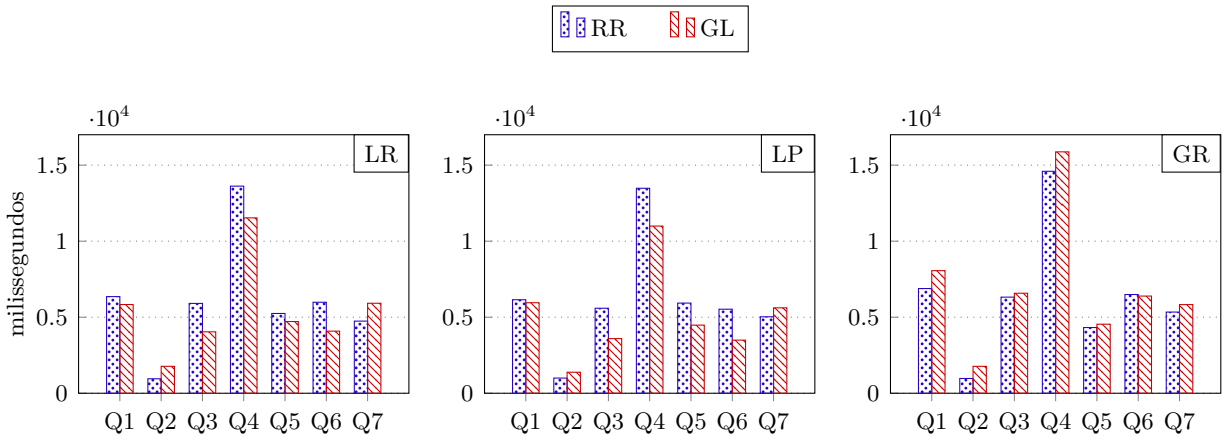

Figura 7 – Comparação do tempo de processamento entre os métodos RR e GL, utilizando os três métodos de escalonamento LR, LP e GR.

Ao analisar os gráficos da Figura 8, que comparam a quantidade total de dados em kB trafegados na rede entre os métodos RR e GL, utilizando os três métodos de escalonamento LR, LP e GR, observa-se que o GL obteve um desempenho inferior ao RR em 42,86% das consultas utilizando os métodos LR e LP, e em 28,57% das consultas utilizando o GR.

A consulta Q2 apresentou métricas mais baixas, com ambos os métodos apresentando um tempo de processamento e a quantidade de dados trafegados na rede baixos, porém a cardinalidade resultante da consulta foi baixa. Esse fator pode ter influenciado no maior tempo de processamento no GL por ter recursos para colocalizar os objetos.

Em relação as consultas Q3, Q4 e Q6, todas consultas com maior volume de dados, é possível que o método GL conseguiu reduzir o tempo de processamento, indicando que houve um balanceamento da carga entre os nós do *cluster*, porém essa melhoria ocorreu ao custo de um maior tráfego de dados na rede – oportunidade que o escalonamento encontrou devido as características da distribuição dos dados pelo método GL. Na consulta Q7, o oposto ocorreu: o GL conseguiu reduzir o tráfego de dados na rede, indicando que ele

colocalizou os dados, porém, isso resultou em um maior tempo de processamento.

Na consulta Q1 e, principalmente, na Q5, o método GL, em conjunto com os escalonadores LR e LP, conseguiu manter um melhor equilíbrio entre o tempo de processamento e o volume de dados trafegados na rede. Esse desempenho foi mais evidente na Q5, onde os *datasets* já estavam colocalizados, facilitando a otimização do GL.

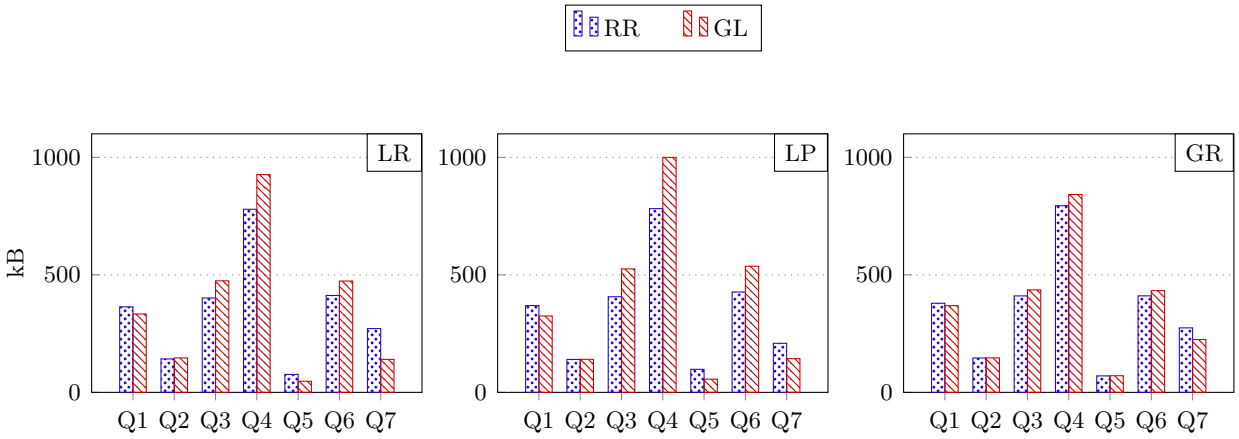

Figura 8 – Comparação do volume de dados trafegados na rede entre os métodos RR e GL, utilizando os três métodos de escalonamento LR, LP e GR.

5 Conclusões e Trabalhos Futuros

5.1 Introdução

Este capítulo apresenta as conclusões obtidas a partir dos resultados da avaliação do método *Gain-Loss* em comparação ao método *Round-Robin*. Além disso, são discutidas as desvantagens observadas nos experimentos com *datasets* reais. Por fim, são sugeridos os trabalhos futuros, que inclui o aprimoramento do método GL e a incorporação de novos algoritmos de distribuição ao DGEO para otimizar o balanceamento de carga e a colocalização de dados em consultas espaciais.

5.2 Conclusões

Neste trabalho foi realizada uma avaliação comparativa da eficiência do método de distribuição espacial *Gain-Loss* em relação ao método *Round-Robin*, utilizando diferentes métodos de escalonamento. Para alcançar o objetivo, realizamos experimentos utilizando sete consultas diferentes, cada uma composta por múltiplos *datasets*. Foram analisadas as seguintes métricas: tempo de execução, na qual o GL demonstrou um desempenho superior ao RR em 52,38%, e o volume de dados trafegados na rede das consultas, onde o GL obteve um desempenho de 45,24%.

Comparando os resultados obtidos por esse trabalho aos resultado obtidos por Tonon e Oliveira (2019), que utilizaram *datasets* sintéticos e um ambiente controlado é possível observar diferenças significativas no comportamento do GL em um ambiente de *cluster* com *datasets* reais.

No trabalho de Tonon e Oliveira (2019), o GL se mostrou eficaz na redução da sobreposição de dados, apresentando resultados superiores ao RR, enquanto o RR demonstrou ser eficaz na questão de balanceamento. Porém, esses experimentos foram realizados com *datasets* sintéticos pouco volumosos, contendo 500 objetos. Por outro lado, os experimentos deste trabalho, realizados com *datasets* reais e mais volumosos, com o menor *dataset* contendo 2.140 objetos, apresentaram resultados mais variados.

Por exemplo, em consultas de alta cardinalidade, como Q3, Q4 e Q6, o GL apresentou redução no tempo de processamento, apesar de aumentar o custo de rede. Já na consulta Q7, o GL reduziu o tráfego de rede, mas resultou em maior tempo de processamento, destacando os desafios de equilibrar colocalização e balanceamento. Por outro lado, o GL apresentou bons resultados em consultas de baixa cardinalidade, como Q5, e em consultas seletivas, como Q2, mas enfrentou desafios em consultas com alta cardinalidade.

Dessa forma, os resultados do GL em um ambiente com dados reais não foi de acordo com o esperado, visto que ele obtinha os melhores resultados no quesito consumo de rede e obteve apenas 45,24% de desempenho nesse quesito. Assim, esse método ainda pode passar por mudanças e adaptações para que se torne mais eficiente.

5.3 Trabalhos futuros

Para trabalhos futuros, sugere-se o aprimoramento do método *Gain-Loss* para conseguir manter um balanceamento e colocalização eficientes no ambiente com dados reais. Repetir os experimentos para tamanhos de *clusters* maiores e incrementalmente, é recomendado para explorar efetivamente as possibilidades do método em conjunto com os escalonadores.

Outro trabalho futuro pode ser a incorporação do método de distribuição R∗-Grove ao DGEO como uma nova abordagem de comparação com o *Gain-Loss* e *Round-Robin*, considerando as características de balanceamento e colocalização do GL. O GL foi desenvolvido para minimizar a sobrecarga de rede e otimizar a distribuição espacial, assim como o R∗-Grove também busca maximizar a eficiência em consultas espaciais. A análise comparativa poderia trazer dados sobre qual método é mais eficiente para diferentes cenários, como alto volume de dados e cargas de trabalho variáveis.

Referências

BACELLAR, H. V. Cluster: Computação de alto desempenho. *Universidade de Campinas*, 2010. Citado na página 15.

BECKMANN, N. et al. The r*-tree: An efficient and robust access method for points and rectangles. In: *Proceedings of the 1990 ACM SIGMOD international conference on Management of data*. [S.l.: s.n.], 1990. p. 322–331. Citado na página 25.

CAMPBELL, J.; SHIN, M. *Geographic information system basics*. [S.l.]: 2012 Book Archive, 2012. Citado 3 vezes nas páginas 19, 20 e 21.

CARNIEL, A. C. Spatial indexing on flash-based solid state drives. In: *PhD@ VLDB*. [S.l.: s.n.], 2018. Citado na página 19.

CHANG, K. tsung. *Introduction to Geographic Information Systems*. 9. ed. McGraw-Hill, 2019. ISBN 9781260502206; 1260502201. Disponível em: <libgen.li/file.php?md5=c7f74c53bede69d995c022b5de1bfcfa>. Citado na página 22.

DITTRICH, J.-P.; SEEGER, B. Data redundancy and duplicate detection in spatial join processing. In: IEEE. *Proceedings of 16th International Conference on Data Engineering (Cat. No. 00CB37073)*. [S.l.], 2000. p. 535–546. Citado na página 23.

ELDAWY, A. et al. Beast: Scalable exploratory analytics on spatio-temporal data. In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. [S.l.: s.n.], 2021. p. 3796–3807. Citado 4 vezes nas páginas 15, 23, 30 e 31.

FITZ, P. *Geoprocessamento sem complicação*. Oficina de Textos, 2018. ISBN 9788579750489. Disponível em: <https://books.google.com.br/books?id=eiJHDwAAQBAJ>. Citado 2 vezes nas páginas 18 e 21.

FRANÇA, A. G. *Precisão da Estimativa de Seletividade de Tarefas de Junção Espacial Distribuída usando Histogramas de Euler*. 63 p. Monografia — Universidade Federal de Goiás, Regional Jataí, Jataí, GO, Brasil, 2018. Citado 2 vezes nas páginas 20 e 22.

HUISMAN, O.; BY, R. A. de et al. Principles of geographic information systems. *ITC Educational Textbook Series*, v. 1, p. 17, 2009. Citado na página 18.

JACOX, E. H.; SAMET, H. Spatial join techniques. *ACM Transactions on Database Systems (TODS)*, Acm New York, NY, USA, v. 32, n. 1, p. 7–es, 2007. Citado na página 15.

LEUTENEGGER, S. T.; LOPEZ, M. A.; EDGINGTON, J. Str: A simple and efficient algorithm for r-tree packing. In: IEEE. *Proceedings 13th international conference on data engineering*. [S.l.], 1997. p. 497–506. Citado na página 23.

LONGLEY, P. A. et al. *Geographic Information Systems and Science (2005)(22nd ed.)(en)(536s)*. 2. ed. Wiley, 2005. ISBN 9780470870006,0470870001,047087001X. Disponível em: <http://gen.lib.rus.ec/book/index.php?md5=70112ac0deee51b0edec281c4891aabf>. Citado na página 19.

MUTENDA, L.; KITSUREGAWA, M. Parallel r-tree spatial join for a shared-nothing architecture. In: IEEE. *Proceedings 1999 International Symposium on Database Applications in Non-Traditional Environments (DANTE'99)(Cat. No. PR00496)*. [S.l.], 1999. p. 423–430. Citado na página 23.

OLIVEIRA, S. de et al. Processamento distribuído de operações de junção espacial com bases de dados dinâmicas para análise de informações geográficas. *XXXI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, p. 12, 2013. Citado 6 vezes nas páginas 16, 23, 24, 28, 30 e 31.

OLIVEIRA, T. B. de. *Efficient Processing of Multiway Spatial Join Queries in Distributed Systems*. Tese (Doutorado) — Instituto de Informática, Universidade Federal de Goiás, Goiânia, GO, Brasil, 11 2017. Disponível em: <http://repositorio.bc.ufg.br/tede/handle/tede/8033>. Citado 11 vezes nas páginas 16, 18, 19, 22, 23, 24, 25, 29, 30, 31 e 32.

OLIVEIRA, T. B. de et al. Scheduling distributed multiway spatial join queries: optimization models and algorithms. *International Journal of Geographical Information Science*, v. 37, n. 6, p. 1388–1419, 2023. Citado 3 vezes nas páginas 15, 16 e 36.

OLIVEIRA, T. B. de; COSTA, F. M.; RODRIGUES, V. J. S. Definição de Planos de Execução Distribuídos para Consultas de Junção Espacial usando Histogramas Multidimensionais. In: *Proceedings of the Brazilian Symposium on Databases*. Petrópolis, RJ, Brazil: [s.n.], 2015. p. 89–100. Citado 2 vezes nas páginas 15 e 35.

PITA, L. C. *Avaliação do uso de Histogramas Espaciais para Particionamento de Dados em Sistemas Distribuídos*. 51 p. Monografia — Universidade Federal de Goiás, Regional Jataí, Jataí, GO, Brasil, 2017. Citado na página 32.

TONON, G. S.; OLIVEIRA, T. B. de. Gain-loss: Método de distribuição de dados para processamento distribuído de multijunções espaciais. In: *Proceedings of XX Geoinfo*. São José dos Campos, SP, Brasil: MCTIC/INPE, 2019. p. 274–279. Citado 8 vezes nas páginas 16, 23, 25, 26, 29, 30, 31 e 38.

XIA, T.; ZHANG, D. Improving the R*-tree with Outlier Handling Techniques. In: *Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*. Bremen, Germany: [s.n.], 2005. p. 125–134. Citado 3 vezes nas páginas 16, 25 e 29.

ZEIDAN, A.; VO, H. T. Efficient spatial data partitioning for distributed k nn joins. *Journal of Big Data*, Springer, v. 9, n. 1, p. 77, 2022. Citado na página 23.