

Otimizador de Particionamento Espaço-Temporal para Séries de Dados Matriciais*

Ana P. B. Pereira¹, Leonardo D. Fuzikawa^{1†}, Geovani P. Santos¹, Daniel S. Kaster¹

¹Departamento de Computação - Universidade Estadual de Londrina (UEL)
Caixa Postal 10.011 – 86.057-970 – Londrina – PR – Brasil

{ana.barbosa.pereira, leonardo.daiki, geovani.pereira, dskaster}@uel.br

Abstract. *The management of matrix data requires partitioning strategies due to its large volume, such as spatiotemporal partitioning, which can be applied to matrix data series. Although systems exist to manage such data, user intervention is generally required to define partitioning configurations. This work presents a spatiotemporal partitioning optimizer for matrix data in order to make partitioning decisions based on workload statistics. Experiments were conducted with workloads arranged in different contexts of spatial and temporal overlap, where the proposed method reduced the number of block accesses by 57.66% on average.*

Resumo. *O gerenciamento de dados matriciais exige estratégias de particionamento devido ao seu grande volume, como o particionamento espaço-temporal, que pode ser aplicado a séries matriciais. Embora existam sistemas que lidam com esses dados, geralmente é necessária a intervenção do usuário para definir as configurações dos particionamentos. Este trabalho propõe um otimizador de particionamento espaço-temporal para séries de dados matriciais a fim de tomar decisões de particionamento com base na carga de trabalho. Foram realizados experimentos com cargas de trabalho dispostas em diferentes contextos de sobreposição espacial e temporal, onde o método proposto reduziu o número de acessos a bloco em 57,66%, em média.*

1. Introdução

Áreas como sensoriamento remoto, meteorologia e astronomia produzem grandes volumes de dados matriciais. Dados *raster*, por exemplo, organizam informações em grades regulares de pixels associadas a coordenadas espaciais e são empregados em aplicações de monitoramento agrícola e detecção de mudanças ambientais [Fu et al. 2024, Miller et al. 2024]. A incorporação de uma dimensão temporal dá origem às séries de dados matriciais, sobre as quais são frequentes consultas espaço-temporais, como, por exemplo, o cálculo do acumulado de chuva em uma região ao longo de um intervalo de tempo. Devido ao volume desses dados e à natureza das operações realizadas, estratégias de particionamento são necessárias para organizar os dados fisicamente e reduzir o número de blocos acessados durante a execução das consultas.

*Este trabalho teve o suporte financeiro do CNPq e da Fundação Araucária e apoio da Espectro Ltda e do projeto NAPI-NORTE - Centro de Inteligência Artificial (CIA-Agro).

†Os dois primeiros autores possuem igual contribuição neste trabalho.

O gerenciamento de dados matriciais é suportado por sistemas como PostGIS [PostGIS 2026], Oracle Spatial [Oracle 2026], Rasdaman [Baumann et al. 2021] e SciDB [Brown 2010]. O desempenho das consultas depende das estratégias de particionamento adotadas, que determinam como os dados são organizados fisicamente. Porém, as estratégias de particionamento disponíveis nesses sistemas são, em geral, definidas manualmente pelo usuário. Abordagens adaptativas existentes, como AQWA [Aly et al. 2015] e GeoBalance [Soltani et al. 2022], buscam ajustar o particionamento com base nas consultas observadas, mas não são voltadas a séries matriciais, tendo enfoque na otimização em ambientes distribuídos. Portanto, mesmo em sistemas com suporte a organização do particionamento, a decisão permanece sob responsabilidade do usuário.

Diante desse cenário, este trabalho propõe um otimizador de particionamento espaço-temporal para séries de dados matriciais, capaz de determinar automaticamente quando e onde aplicar operações de particionamento com base na análise da carga de trabalho. A abordagem monitora uma janela de consultas, identifica regiões espaço-temporais mais requisitadas e avalia o custo de configurações de particionamento, selecionando aquela que minimiza os acessos à memória secundária. Experimentos conduzidos em um ambiente de simulação comparando o método proposto com otimizações homogêneas, compreendendo 144 baterias de testes sobre dados do Sentinel-2 [Drusch et al. 2012], mostraram que a proposta superou as otimizações homogêneas, evidenciando o potencial da abordagem para automatizar decisões de particionamento.

2. Conceitos e Trabalhos Relacionados

O modelo de dados matriciais é uma representação de dados baseada em uma grade bidimensional composta por um número finito de valores. Um exemplo é o *raster*, dado geoespacial normalmente utilizado para armazenar registros de satélites. Com a dimensão temporal, formam-se séries matriciais, nas quais cada instante está associado a uma matriz. Sobre essas séries podem ser realizadas consultas espaço-temporais, que recuperam dados de uma região em um intervalo de tempo e acessam blocos, que são as unidades físicas de armazenamento. Como o custo da consulta depende da organização dos dados em disco, estratégias de particionamento são usadas para reduzir o número de blocos acessados [Baumann et al. 2021].

Na literatura, os métodos de particionamento de dados matriciais são divididos em duas linhas principais. No particionamento espacial [Hojati et al. 2024, Baumann et al. 2021], cada matriz da série temporal é particionada de forma isolada, sem ligação direta com os dados dos instantes adjacentes. Essa abordagem é usada por sistemas de dados espaciais consolidados, como PostGIS [PostGIS 2026] e Oracle Spatial [Oracle 2026], além de sistemas para dados *raster* e multidimensionais, como Rasdaman [Baumann et al. 2021] e SciDB [Brown 2010]. Já o particionamento espaço-temporal [Hu et al. 2020, Li et al. 2017] reúne em um bloco dados de uma região por um período, de modo que consultas sobre vários instantes obtenham os dados necessários com menos acessos a blocos. Estratégias recentes propõem o particionamento espaço-temporal dinâmico, que combina subdivisão do espaço e realocação dos dados de instantes subsequentes nas sub-regiões geradas [Santos and Kaster 2025]. Nesse caso, uma região A com dados em T_i e T_{i+1} pode ser dividida em A_1 e A_2 , reagrupando os dados por subárea nos blocos $A_{1_{T_i, T_{i+1}}}$ e $A_{2_{T_i, T_{i+1}}}$. Esse processo é similar ao usado em estruturas de indexação espacial, como a *k-d-tree*. Contudo, esses sistemas delegam ao usuário a

definição do particionamento físico, sem sugestões automáticas.

Existem abordagens que buscam adaptar o particionamento à carga de trabalho. O AQWA [Aly et al. 2015] implementa um mecanismo adaptativo para dados vetoriais em *clusters* Hadoop, utilizando *k-d-trees* para redistribuir partições com base na distribuição das consultas observadas. O GeoBalance [Soltani et al. 2022] emprega um algoritmo para migrar incrementalmente regiões entre partições em sistemas distribuídos, respondendo a variações na carga em tempo real. Ambos identificam padrões de acesso para orientar decisões de particionamento, porém não consideram simultaneamente as dimensões espacial e temporal e não são diretamente comparáveis à proposta deste trabalho.

Portanto, há uma carência na literatura de mecanismos que, a partir da análise da carga de trabalho, identifiquem automaticamente quando e onde aplicar particionamentos espaço-temporais em séries de dados matriciais. Este trabalho propõe um método de otimização capaz de realizar essas decisões de particionamento de forma automática.

3. Otimizador de Particionamento Espaço-Temporal

Esta seção descreve a abordagem proposta para monitoramento, análise e tomada de decisão visando otimizar as operações de particionamento espaço-temporal. O processo é dividido em três etapas principais, conforme ilustrado na Figura 1.

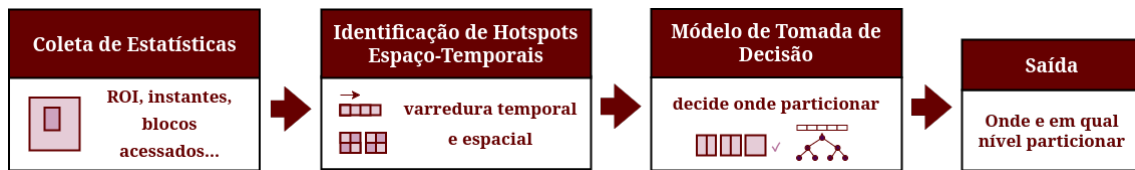


Figura 1. Etapas do processo de otimização espaço-temporal proposto.

Dado um conjunto de dados com n *rasters* cobrindo os instantes de tempo T_0 a T_{n-1} , a primeira etapa realiza a **Coleta de Estatísticas**. Para isso, utiliza-se um vetor de instantes de tempo onde, para cada instante T_i é mantida uma *k-d-tree* cuja raiz representa toda a região espacial daquele instante, como ilustra a Figura 2. A profundidade inicial delimita o tamanho do bloco não particionado, contendo apenas um instante de tempo. Cada nó da estrutura armazena as coordenadas (x_{bi}, y_{bi}) e (x_{bf}, y_{bf}) do MBR do bloco b , uma lista de consultas (L_q) que intersectam o nó e, para cada consulta q , as coordenadas da MBR da consulta (x_{qi}, y_{qi}) e (x_{qf}, y_{qf}) , o intervalo temporal $[T_{qi}, T_{qf}]$ e um identificador único de consulta u_q . Dessa forma, a raiz de cada *k-d-tree* contém o total de consultas q que acessam aquele instante.

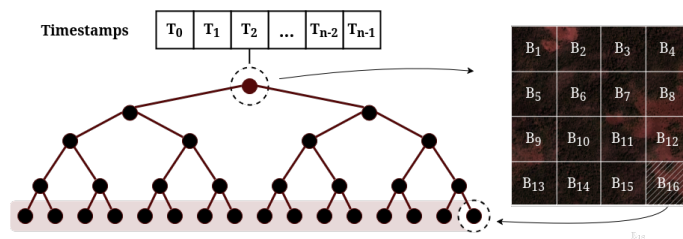


Figura 2. Estrutura de armazenamento de estatísticas de consulta.

Após uma quantidade de consultas determinada por um parâmetro, é executada a segunda etapa de **Identificação de Hotspots Espaço-Temporais**. Inicialmente, realiza-se uma varredura sequencial das raízes das k - d -trees, de T_0 a T_{n-1} , para identificar os instantes mais requisitados. O tamanho do intervalo analisado é de 2^P , assumindo particionamentos recursivos no padrão da k - d -tree, em que P é o nível do particionamento, a fim de detectar o maior intervalo considerado *hotspot*. Em seguida, para cada intervalo identificado, realiza-se uma varredura espacial nos nós folha da estrutura da Figura 2, a fim de determinar os blocos mais requisitados dentro daquele intervalo temporal. O resultado é um conjunto de *hotspots* espaço-temporais, cada um definido por uma região espacial (x_{hi}, y_{hi}) e (x_{hf}, y_{hf}) e um intervalo temporal $[T_{hi}, T_{hf}]$.

Os *hotspots* são então enviados como entrada ao **Módulo de Tomada de Decisão**. O módulo constrói uma estrutura auxiliar que contém um vetor de k - d -trees de profundidade P , onde cada célula é associada a um instante temporal. Essa estrutura representa as diferentes possibilidades de particionamento espaço-temporal, em que cada nó armazena a quantidade de acessos realizados naquela região espacial e intervalo de tempo pelas consultas da carga de trabalho capturada nas estatísticas. O módulo realiza uma análise *what-if*, associando a cada nó o custo da execução da carga de trabalho considerando um particionamento no nível correspondente. Se o custo do nível é superior ao custo do particionamento de nível inferior da região e intervalo de tempo equivalente, o processo segue recursivamente aquele ramo. Caso o particionamento não reduza o número de acessos, a possibilidade de otimização é descartada.

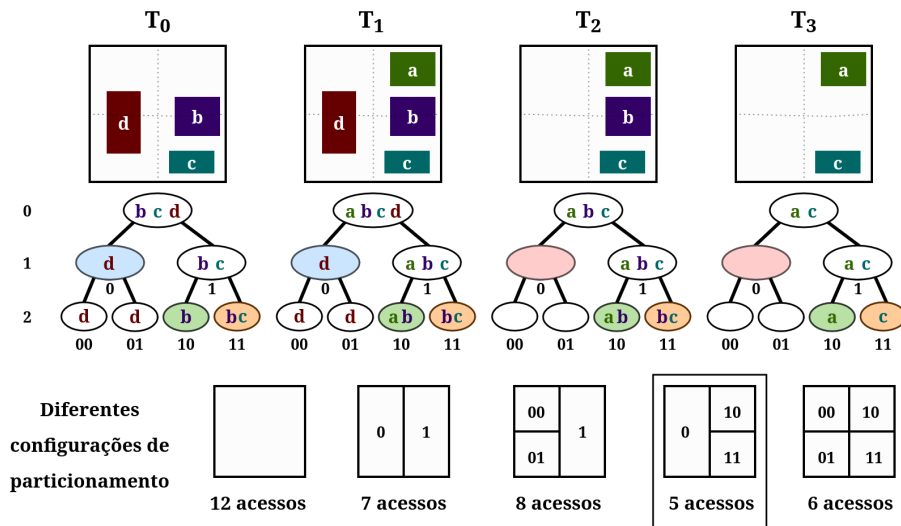


Figura 3. Análise de custos e simulação *what-if* dos particionamentos.

A Figura 3 mostra um exemplo onde há um *hotspot* espaço-temporal de 4 instantes de tempo $[T_0, T_3]$, ou seja, $P = 2$, e 4 consultas a , b , c e d . O nó raiz indica as consultas que acessam o bloco sem particionamento. O segundo nível, indica as consultas que acessam blocos considerando o primeiro nível de particionamento espaço-temporal. A consulta d , por exemplo, aparece na raiz de T_0 e T_1 , demandando dois acessos sem particionamento, mas aparece apenas à esquerda no nível 1, demandando apenas um acesso com esse particionamento, pois agruparia em um único bloco a região nos instantes T_0 e T_1 (destacado em azul na árvore). Em sequência, a análise *what-if* decide pelo particio-

namento de nível 2 no ramo à direita, pois as regiões 10 e 11, que abrangem de $[T_0, T_3]$, permitem a maior redução de acessos, onde as consultas a e c poderiam ser executadas com apenas um acesso (blocos agrupando os nós verdes e laranjas, respectivamente) e a consulta b com dois acessos. A parte inferior da Figura 3 mostra as diferentes possibilidades de particionamento que podem ser realizadas para este cenário. Sem particionar, as consultas demandariam 12 acessos a blocos. Já no melhor particionamento, conforme destacado, o número de acessos é reduzido para 5. Essa configuração é gerada ao final da análise, onde são escolhidos os particionamentos equivalentes aos nós 0, 10 e 11.

4. Experimentos

Para os experimentos, foi desenvolvido um ambiente de simulação de particionamento espaço-temporal, para ser posteriormente acoplado ao sistema [Santos and Kaster 2025]. O principal objetivo foi comparar a quantidade de acessos a blocos entre a estratégia proposta e otimizações homogêneas, aplicadas uniformemente em todo o conjunto de dados, em três categorias, baseadas na quantidade de instantes de tempo agrupados por particionamento, sendo eles 2, 8 e 16 instantes. Como referência, também foi avaliada a execução sem particionamento espaço-temporal. O conjunto de dados utilizado contém imagens de satélite Sentinel-2 [Drusch et al. 2012], referentes à região T22KDV, no Paraná, com dimensões de 10980×10980 pixels e compreende temporalmente uma série de 68 *rasters*.

Na carga, os *rasters* foram particionados espacialmente, como normalmente ocorre nos bancos de dados espaciais, em blocos de 687×687 pixels. Na sequência, foram executadas baterias de testes para as diferentes estratégias de particionamento, utilizando consultas variando o tamanho da região de consulta (50×50 e 300×300 pixels), e a extensão temporal (2, 8 e 16 instantes), totalizando seis combinações. A carga de trabalho foi organizada com um padrão dominante de consulta (70%), fixando-se o tamanho da região de consulta e o número de instantes de tempo, e um padrão complementar (30%), formado pelas demais combinações. Também foram consideradas variações de sobreposição entre consultas: (a) sem sobreposição, (b) sem sobreposição espacial e com sobreposição temporal, (c) com sobreposição espacial e sem sobreposição temporal e (d) com sobreposição espacial e temporal. Todas as combinações foram consideradas como padrão dominante e complementar, para todas as variações de sobreposição, totalizando 144 baterias de testes. Cada bateria compreendeu 100 consultas cuja posição da região de consulta e o instante inicial de cada consulta foram sorteados aleatoriamente.

4.1. Resultados

A Figura 4 apresenta o desempenho das estratégias de particionamento considerando diferentes cargas de trabalho. O eixo X apresenta os padrões predominantes (70% da carga de trabalho), diferenciando-se pelo tamanho da região de consulta e pela extensão temporal envolvida. A métrica utilizada para avaliação e comparação dos métodos foi o total de acessos a disco demandados pela carga de trabalho, sem considerar o uso de *cache*.

Observa-se que a abordagem proposta supera os particionamentos homogêneos em todos os cenários. Os particionamentos homogêneos foram consideravelmente mais eficientes que a execução sem particionamento espaço-temporal. Entre as estratégias homogêneas, o melhor desempenho foi obtido com particionamentos de 16 instantes, alcançando reduções de até 76.56%, enquanto o método proposto superou esse resultado no

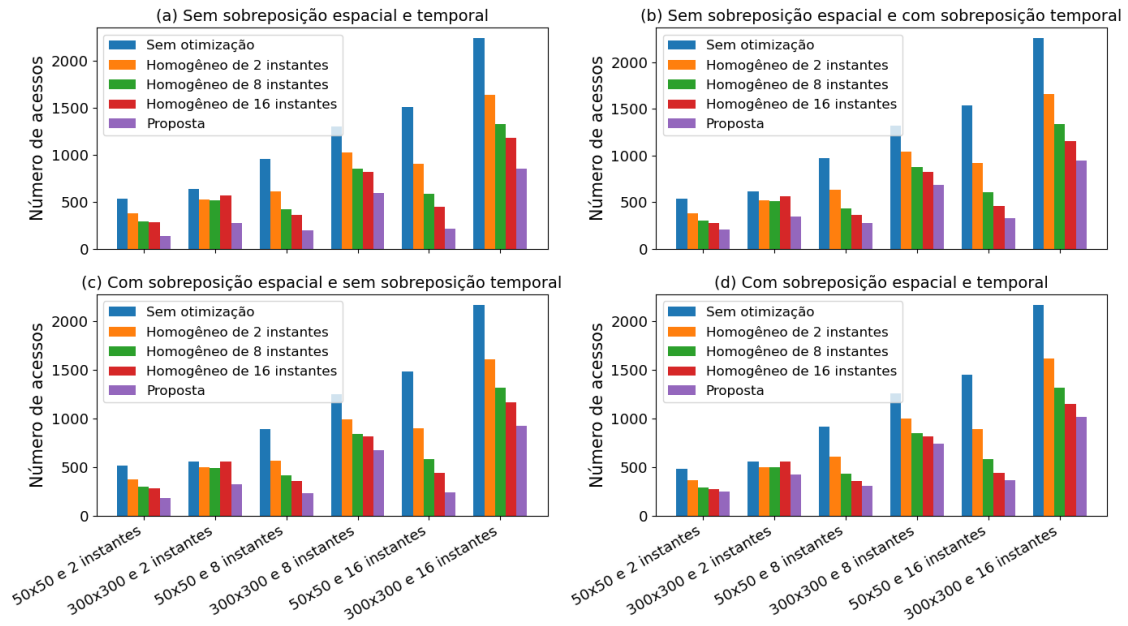


Figura 4. Resultados dos experimentos comparando os acessos a bloco de diferentes estratégias de otimização.

mesmo cenário, atingindo 85.82%. Os maiores ganhos da abordagem proposta ocorrem em cenários com múltiplos instantes de tempo, nos quais há redução maior de acessos a bloco em relação às demais estratégias. Além disso, a proposta apresenta melhor desempenho em cenários sem sobreposição espacial e/ou temporal, nos quais há maior isolamento de contextos e menor conflito entre consultas, pois em cenários com sobreposição, algumas consultas podem ser beneficiadas e outras penalizadas por um determinado particionamento. No melhor caso, o otimizador chegou a 93.56% de redução, contra no máximo 73.46% das demais estratégias no mesmo cenário. Em média, as reduções foram de 25.99% para 2 instantes, 38.38% para 8 instantes, 41.04% para 16 instantes e 57.66% para a abordagem proposta, mostrando a capacidade de gerar configurações de particionamento efetivas e adequadas a diferentes cargas de trabalho.

5. Conclusão

Este trabalho¹ apresentou uma nova abordagem de otimização do processo de particionamento espaço-temporal para séries de dados matriciais. A proposta coleta estatísticas das consultas, identifica *hotspots* espaço-temporais e toma decisões de particionamento utilizando uma árvore de análise de custo com base na carga de trabalho. Resultados experimentais indicam uma média de redução de acesso a blocos de 57.66% na abordagem proposta, onde alternativas homogêneas obtiveram reduções médias de, no máximo, 41.04%, demonstrando o potencial da proposta em otimizar particionamentos espaço-temporais de forma transparente ao usuário. Trabalhos futuros incluem a implementação da abordagem proposta no ambiente de [Santos and Kaster 2025] e testes em cargas de trabalho reais.

¹O uso de Inteligência Artificial foi limitado à revisões gramaticais.

Referências

- Aly, A. M., Mahmood, A. R., Hassan, M. S., Aref, W. G., Ouzzani, M., Elmeleegy, H., and Qadah, T. (2015). Aqwa: adaptive query workload aware partitioning of big spatial data. *Proceedings of the VLDB Endowment*, 8(13):2062–2073.
- Baumann, P., Misev, D., Merticariu, V., and Huu, B. P. (2021). Array databases: Concepts, standards, implementations. *Journal of Big Data*, 8:1–61.
- Brown, P. G. (2010). Overview of scidb: large scale array storage, processing and analysis. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 963–968.
- Drusch, M., Del Bello, U., Carlier, S., Colin, O., Fernandez, V., Gascon, F., Hoersch, B., Isola, C., Laberinti, P., Martimort, P., et al. (2012). Sentinel-2: Esa’s optical high-resolution mission for gmes operational services. *Remote sensing of Environment*, 120:25–36.
- Fu, Y., Zhu, Z., Liu, L., Zhan, W., He, T., Shen, H., Zhao, J., Liu, Y., Zhang, H., Liu, Z., et al. (2024). Remote sensing time series analysis: A review of data and applications. *Journal of Remote Sensing*, 4:0285.
- Hojati, M., Roberts, S., and Robertson, C. (2024). Dstree: A spatio-temporal indexing data structure for distributed networks. *Mathematical and Computational Applications*, 29:42.
- Hu, F., Yang, C., Jiang, Y., Li, Y., Song, W., Duffy, D. Q., Schnase, J. L., and Lee, T. (2020). A hierarchical indexing strategy for optimizing apache spark with hdfs to efficiently query big geospatial raster data. *International Journal of Digital Earth*, 13:410–428.
- Li, Z., Hu, F., Schnase, J. L., Duffy, D. Q., Lee, T., Bowen, M. K., and Yang, C. (2017). A spatiotemporal indexing approach for efficient processing of big array-based climate data with mapreduce. *International Journal of Geographical Information Science*, 31(1):17–35.
- Miller, L., Pelletier, C., and Webb, G. I. (2024). Deep learning for satellite image time-series analysis: A review. *IEEE Geoscience and Remote Sensing Magazine*, 12(3):81–124.
- Oracle (2026). Oracle spatial and graph. <https://www.oracle.com/database/spatial-database/>.
- PostGIS (2026). Postgis — spatial and geographic objects for postgresql. <https://postgis.net/>.
- Santos, G. P. and Kaster, D. S. (2025). Particionamento dinâmico espaço-temporal para séries de dados matriciais. In *Simpósio Brasileiro de Banco de Dados (SBBD)*, pages 549–562. SBC.
- Soltani, K., Padmanabhan, A., and Wang, S. (2022). Geobalance: workload-aware partitioning of real-time spatiotemporal data. *GeoInformatica*, 26(1):67–94.