What is a quadtree and how it works

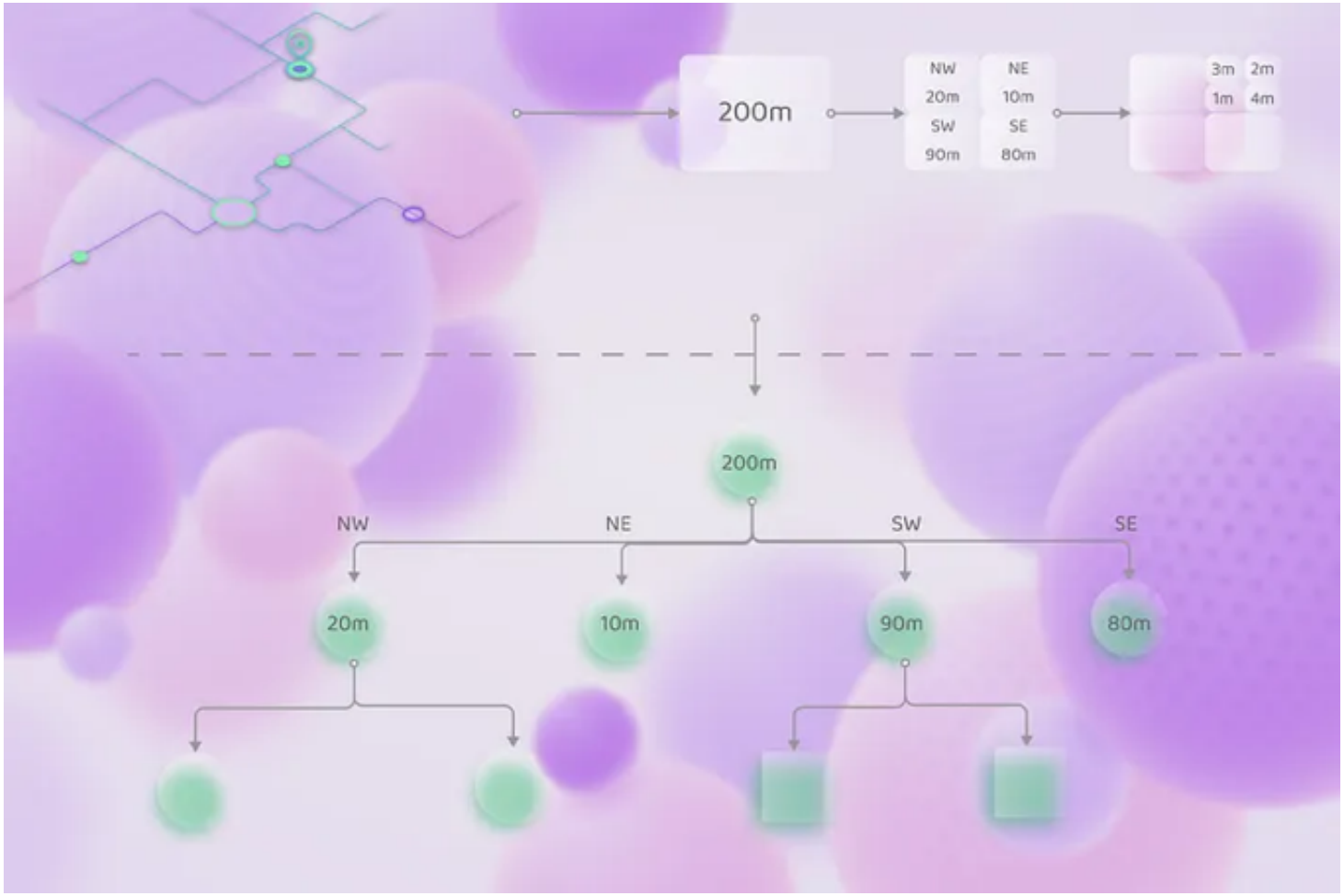
Roman Glushach Follow 3 min read · May 23, 2023

20

Quadtree is a tree-based data structure that recursively partitions a two-dimensional space into four equal quadrants or regions. This structure is used to represent and store spatial data such as points, lines, and polygons in a more efficient way than using a simple list or array. A quadtree is a type of tree data structure where each node has at most four children. The root node represents the entire 2D space, which is divided into four quadrants. Each quadrant is represented by a child node of the root. The child nodes can also be further divided into four quadrants if they contain more than one point.

This process continues recursively until all points are contained in individual leaves of the tree.

A quadtree is a way of organizing a two-dimensional space by breaking it down into smaller and smaller parts. It starts by dividing the space into four equal quadrants, and then it continues to subdivide each quadrant into four more quadrants, until all the subdivisions meet certain criteria.



To create a quadtree, we need to define the boundaries of the space we want to partition, and a function that determines how to split a space into four quadrants. We also need to decide how to store the data associated with each leaf node, and how to traverse the tree to access or modify the data. There are different types of quadtrees, such as region quadtrees, point quadtrees,

line quadtrees, and curve quadtrees, depending on the type and shape of the data they represent.

When using quadtree will be beneficial

- it can adapt to the distribution of the data, creating finer partitions where the data is dense and coarser partitions where the data is sparse
- it can reduce the search space and improve the efficiency of queries that involve spatial regions or nearest neighbors
- it can compress images by representing similar regions with fewer bits, reducing the storage space and transmission time

When using quadtree disadvantageous

- it can create a lot of small nodes and increase the memory overhead if the data is very irregular or noisy
- it can be difficult to implement and maintain, especially for dynamic data that changes frequently
- it can introduce artifacts or errors in image compression if the partitioning is too coarse or the quantization is too low

Quadtree has many applications in computer science, including:

- Image compression: Quadtree can be used to compress images by dividing an image into rectangular regions and replacing each region with its average color
- Collision detection: Quadtree can be used to detect collisions between objects in a game or simulation by checking if the bounding boxes of the objects intersect
- Location-based services: Quadtree can be used to store and query geospatial data efficiently, such as finding all points within a given distance of a location

Conclusion

Quadtree is a powerful data structure that enables efficient storage and querying of spatial data. Its recursive structure allows for fast searching and manipulation of large datasets. Understanding how quadtree works is essential for anyone working with spatial data.

Data Visualization | Datastrucutre | Design Patterns | Maps | Trees

Written by Roman Glushach

760 followers · 0 following

Follow

Senior Software Architect & Engineer Manager at Freelance

No responses yet

I Ishu

What are your thoughts?

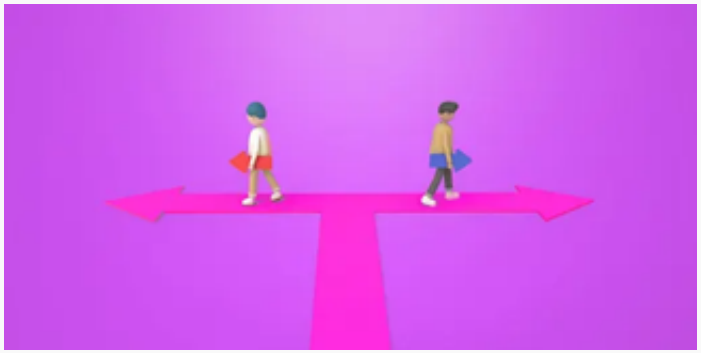
More from Roman Glushach

Roman Glushach

Comparing NPM, YARN, and PNPM Package Managers: Which One is...

If you are a Frontend or Full Stack developer, you probably have used one or more of thes...

Jun 28, 2023 264 6

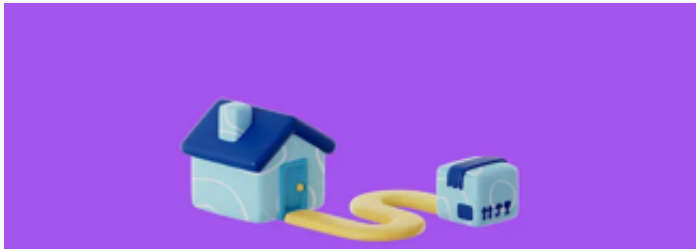


Roman Glushach

Understanding Hexagonal, Clean, Onion, and Traditional Layered...

Anarchitecture refers to the segmentation of a system into parts, the organization and...

Oct 10, 2023 207 1



Roman Glushach

Domain-Driven Design (DDD): A Guide to Building Scalable, High-...

Domain-driven design (DDD) is a software design approach that focuses on modeling...

Oct 6, 2023 238 5

Roman Glushach

The Evolution of Kafka Architecture: From ZooKeeper to...

Kafka's architecture has recently shifted from ZooKeeper to a quorum-based controller th...

Jul 5, 2023 185 1

See all from Roman Glushach

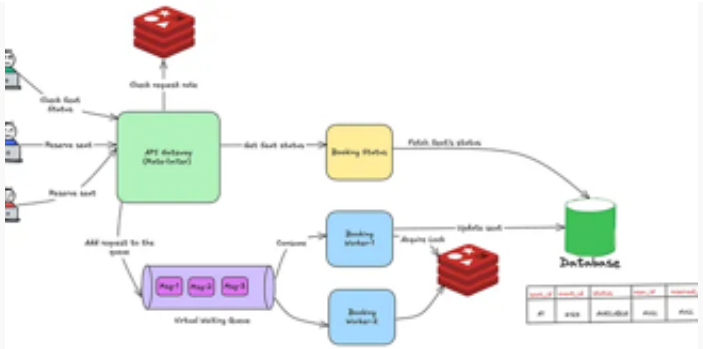
Recommended from Medium

Sri Charan Nama

Geohash vs Quadtree: Choosing the Right Spatial Index for Locati...

Understanding the core differences and when to use each approach in system design

Jun 30 5



In ITNEXT by Animesh Gaitonde

Solving Double Booking at Scale: System Design Patterns from Top...

Learn how Airbnb, Ticketmaster, and booking platforms handle millions of concurrent...

Oct 8 1.8K 22

In Level Up Coding by Fareed Khan

Building an Agentic Deep-Thinking

Tosny

7 Websites I Visit Every Day in

Building an Agentic Deep-Thinking RAG Pipeline to Solve Complex...

Planning, Retrieval, Reflection, Critique, Synthesis and more

6d ago 974 9

7 Websites I Visit Every Day in 2025

If there is one thing I am addicted to, besides coffee, it is the internet.

Sep 23 5.5K 193

The CS Engineer

Forget JSON—These 4 Data Formats Made My APIs 5× Faster

A 1.2 KB JSON payload transformed a smooth request into a 120 ms wait. Switching format...

Sep 30 1.2K 35

Michal Malewicz

I was wrong about Liquid Glass.

It's actually amazing—here's why.

6d ago 546 20

See more recommendations