

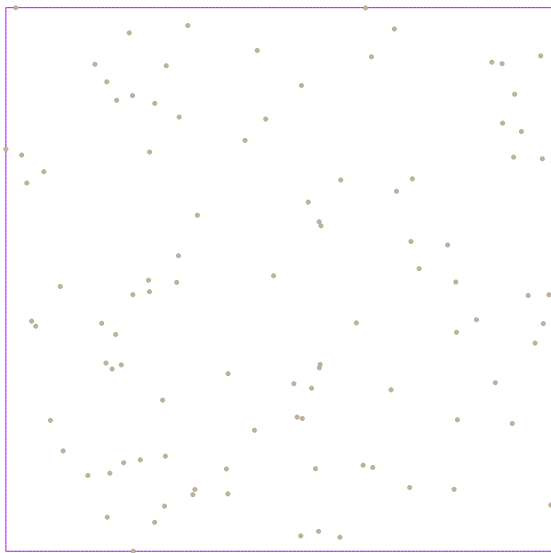
# PFLOCK Report

Andres Calderon

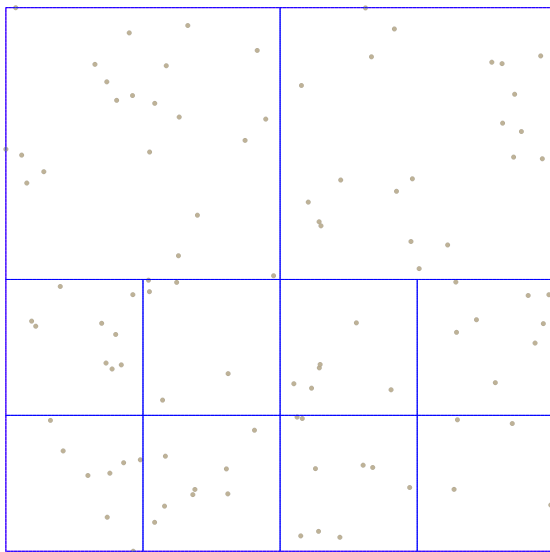
University of California, Riverside

May 18, 2020

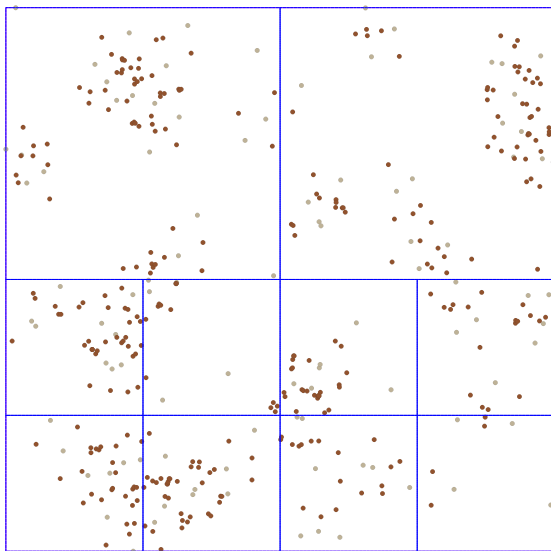
# Points...



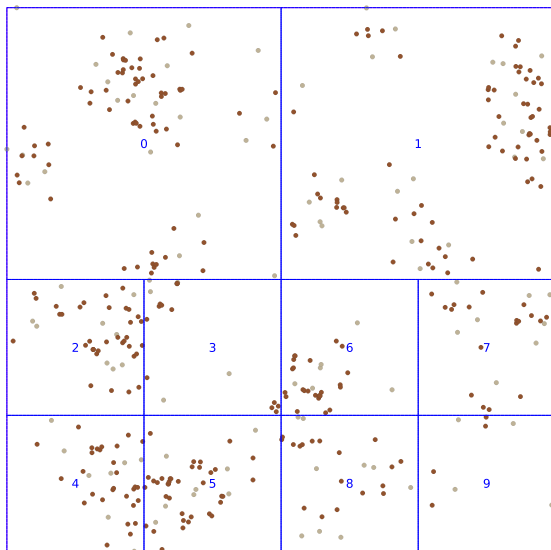
# Global Partitions...



# Centers...



# Global Parititons...

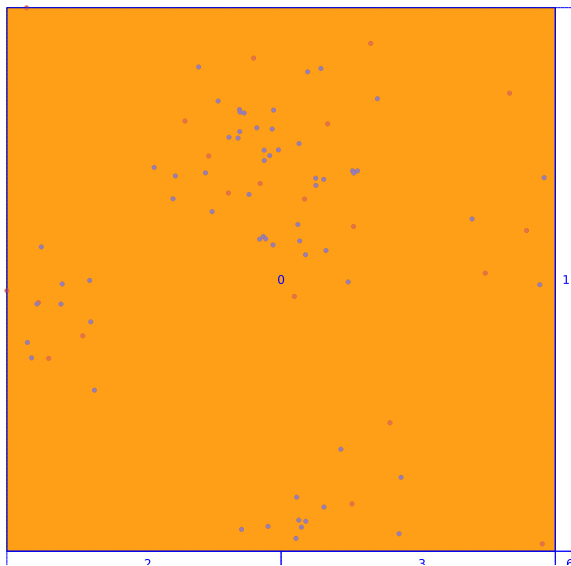


# Local Quadtree...

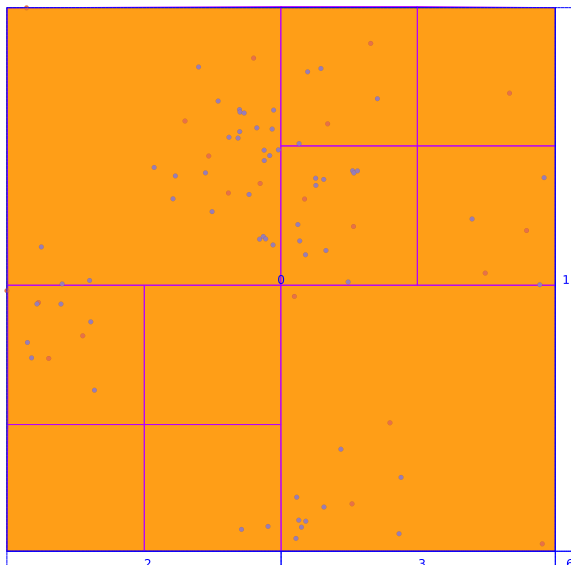
Input: A sequence of Points (I have marked which ones are points and which ones are centers)

1. Take a sample from the sequence
2. Build a quadtree with the sample
3. Query the quadtree with the rest of the sequence to extract the corresponding leaf or leaves.
  - 3.1 In case of a center, query its corresponding envelope.
4. Match items in the same leaf and combine centers and points.

# Partition 0...



# Local Quadtree...





# What is the problem?

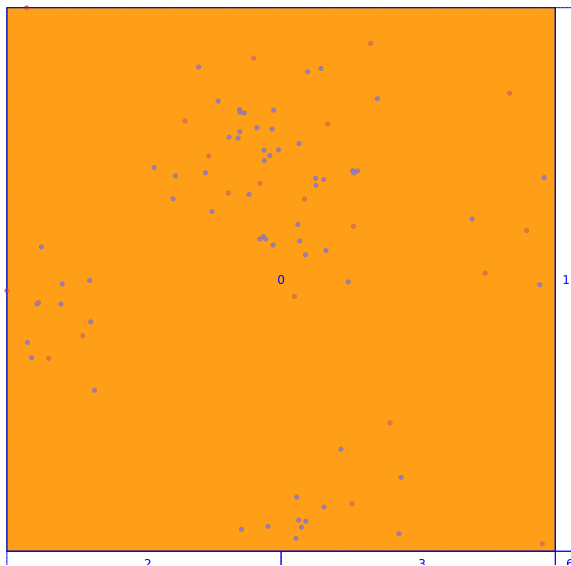
- ▶ Construction of the quadtree takes most of the time.
- ▶ Difficult to tune good quadtrees. Most of the time it generates very large quadtrees.

# Local Grids...

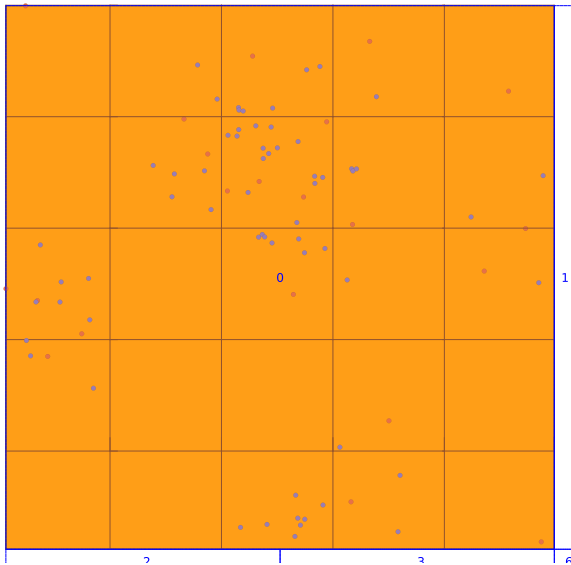
Input: An index of Points (I can query the index to extract points and identify if it is a point or center)

1. Create a set of regular grids based on the extension of the Partition's boundary.
2. For each grid:
  - 2.1 query the index to retrieve the associated items.
  - 2.2 combine points and centers.

# Partition 0...



## Local Grid...



# What is the problem?

- ▶ The JTS quadtree provided by GeoSpark as local index cannot retrieve leaves' MBRs.
- ▶ I can avoid the use of the local index but I have to solve the replication of centers manually.

# What's next?

- ▶ Double-check the local quadtree strategy.
- ▶ Work on local grids but avoid using an extra index.
- ▶ Explore other libraries: RTrees or KDBTrees.