PFLOCK Report

Andres Calderon

University of California, Riverside

April 13, 2020

Partition based join...

1. Partition centers and points using the same set of global grids (Quadtree grids).

2. At each global grid, take a sample of centers and build a local quadtree.

3. For each point in a global grid:
   3.1 compute the position of that point according to the grids defined by the local quadtree.

4. For centers in the same global grid:
   4.1 compute the position of each center according to local grids and manage possible replication.

5. For each local grid, filter pairs which do not satisfy the distance condition.

Experiment setup...

- ▶ Dataset: LA_50K (time interval 320).
- ▶ $\varepsilon = 45$ and $\mu = 3$.
- ▶ Global partitioning: Quadtree, number of partitions: 256 (≈ 750 grids).
- ▶ Number of nodes: 12, cores per node: 9 (108 total cores).
- ▶ Varying values of capacity (maximum items per grid) and fraction (size of the sample respect to # of points in grid) for the local grids.

Results...

Execution time by maximum items per grid (capacity)

Results...

Execution time by size of the sample (fraction)
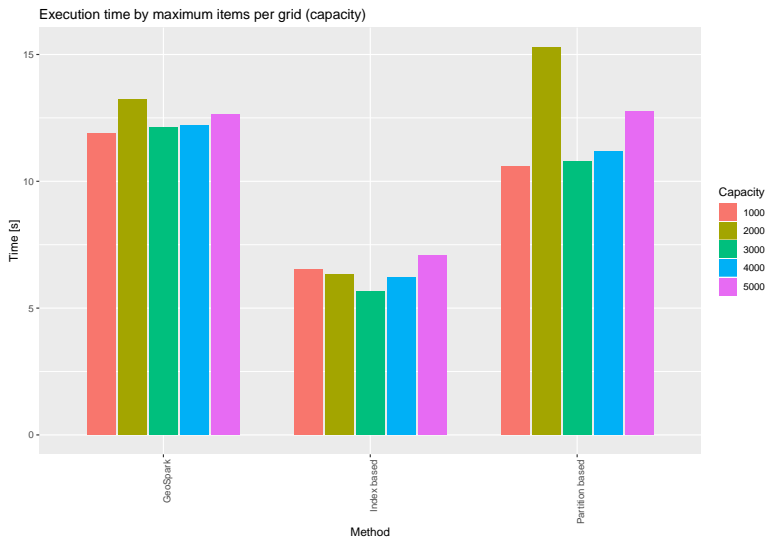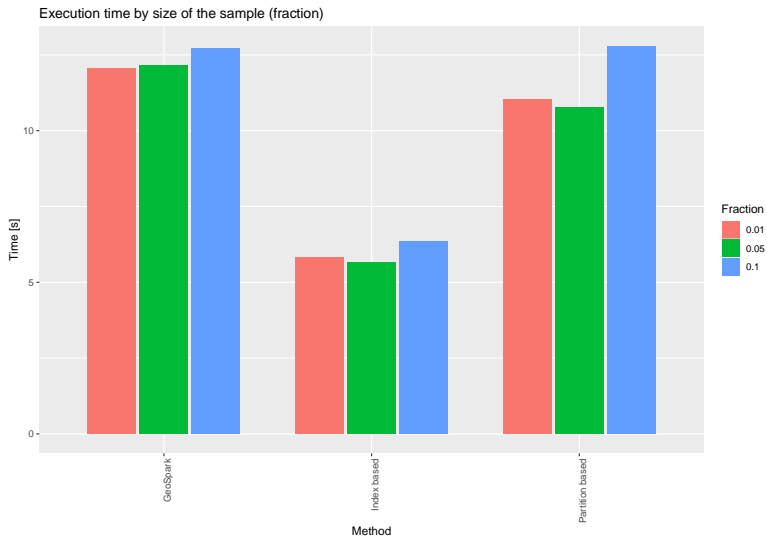
What's next?

- Optional:
  - Avoid the sampling during local quadtree creation and querying.
  - Try KDB-Tree as local partitioner.
- Continuing with index-based join and integrate with previous code.