

CG_Hadoop: computational geometry in MapReduce.

Ahmed Eldawy, Yuan Li, Mohamed F. Mokbel and Ravi Janardan
SIGSPATIAL'13.

Andres Calderon

October 18, 2016

Agenda

1 Background

2 Computational Geometry Operations

- Union
- Skyline
- Convex Hull
- Farthest Pair
- Closest Pair

3 Experiments

4 Conclusions

Agenda

1 Background

2 Computational Geometry Operations

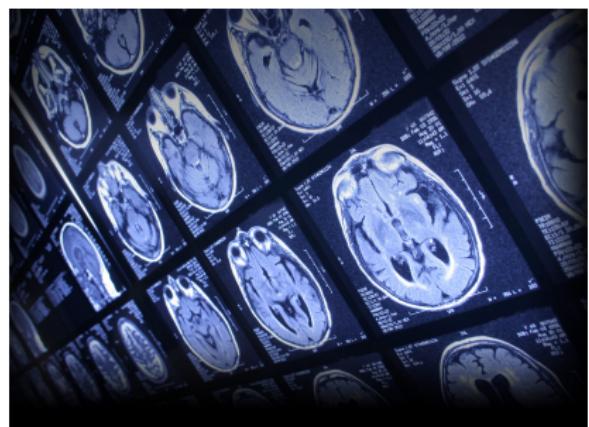
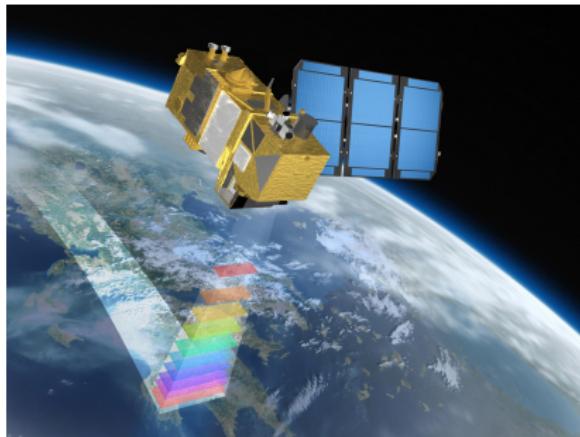
- Union
- Skyline
- Convex Hull
- Farthest Pair
- Closest Pair

3 Experiments

4 Conclusions

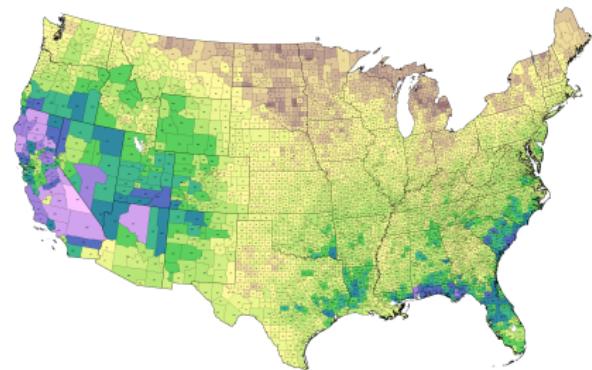
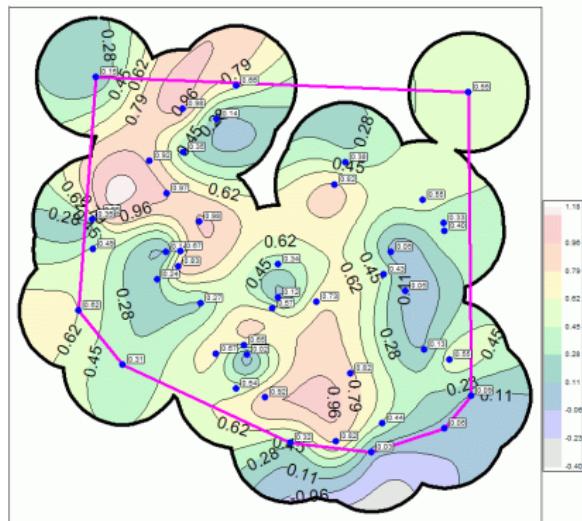
Introduction

- Hadoop is widely used on many applications...
 - Machine learning, Graph processing, Sorting...
- But we are watching a big [spatial] data explosion...
 - satellites, smart phones, space telescopes, medical devices...



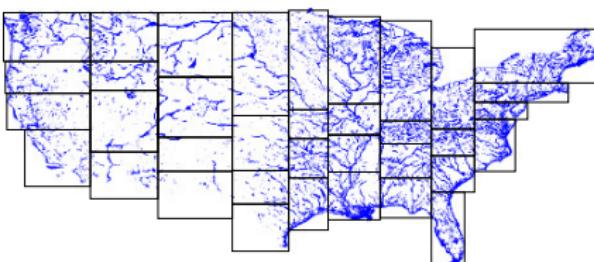
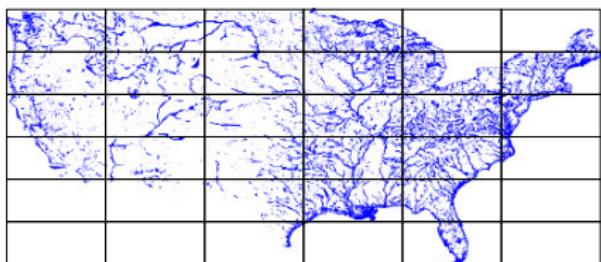
Introduction

- Computational geometry operations are quite costly.
 - Convex hull - 4 billion points - up to three hours.
 - Polygon union - 5 million polygons - around one hour.



SpatialHadoop

- Extend Hadoop for spatial analysis.
 - Efficient operation for spatial querying.
 - Support for spatial types (point, polygons, rectangle).
 - Spatial partitioning and indexing techniques (Grid, R-tree, R+-tree).
 - Open source (<http://spatialhadoop.cs.umn.edu/>).



(Eldawy and Mokbel, 2015)

CG_Hadoop

- Apply a divide and conquer approach.
- Exploit spatial indexing and early pruning (SpatialHadoop).
- Achieve up to 260x increase performance.
- Part of the operation layer of SpatialHadoop.
- Run under Hadoop, but not the same performance.

CG_Hadoop

- Main idea:
 - ① Partition the input.
 - ② Filter partitions that do not contribute to the answer when it is possible.
 - ③ Apply the algorithm locally for each partition.
 - ④ Merge the local answers to compute a global result.

Agenda

1 Background

2 Computational Geometry Operations

- Union
- Skyline
- Convex Hull
- Farthest Pair
- Closest Pair

3 Experiments

4 Conclusions

Agenda

1 Background

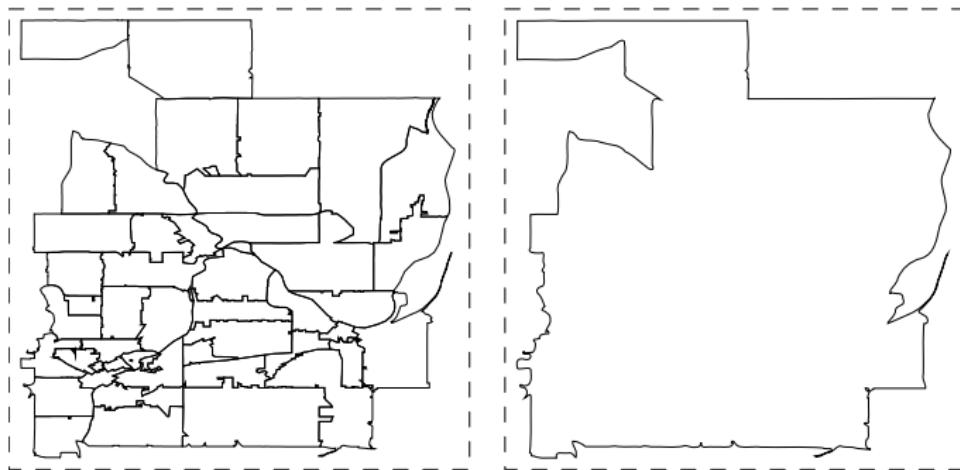
2 Computational Geometry Operations

- Union
- Skyline
- Convex Hull
- Farthest Pair
- Closest Pair

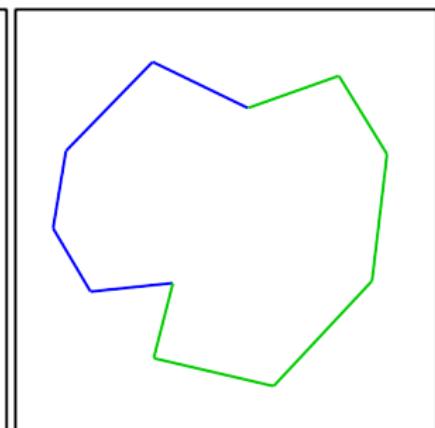
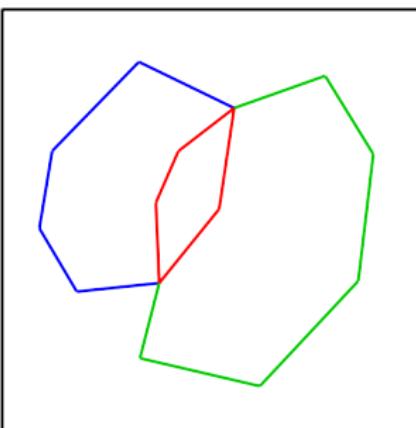
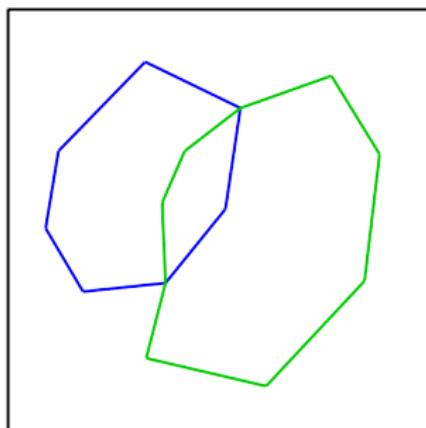
3 Experiments

4 Conclusions

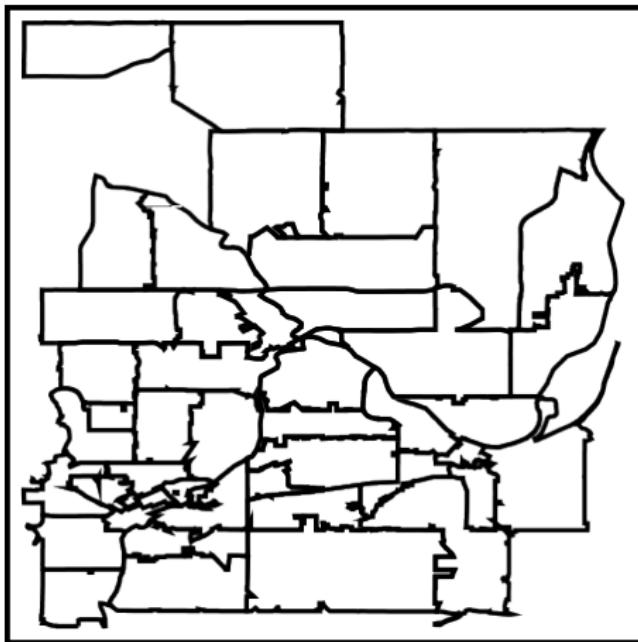
Union



Traditional algorithm

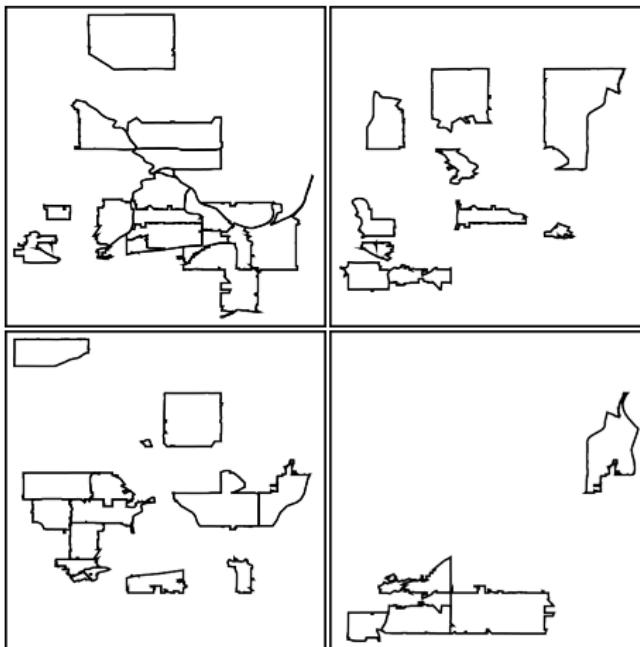


Hadoop - Partition



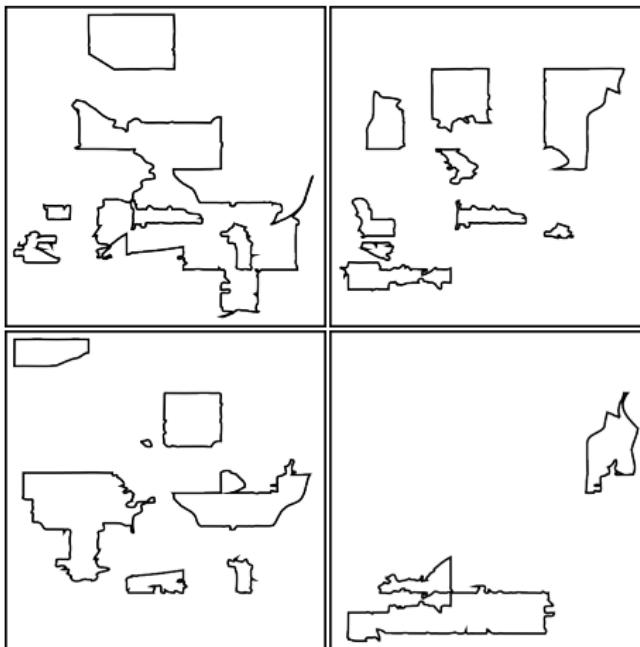
Partition Local Global

Hadoop - Partition



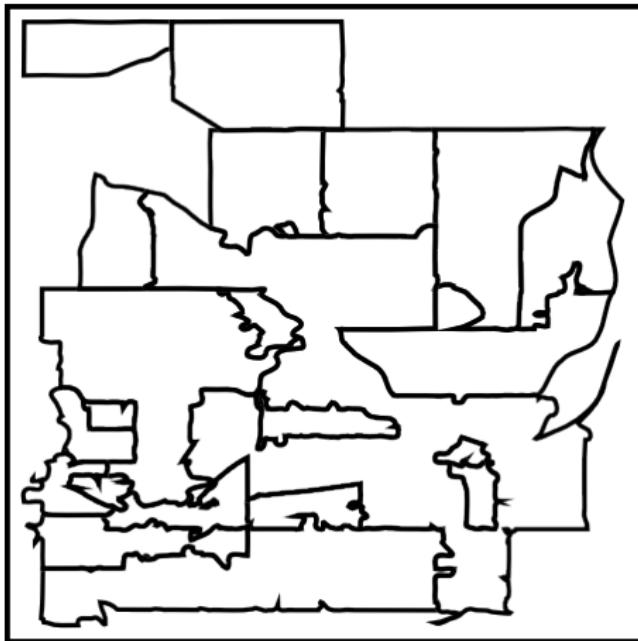
Partition Local Global

Hadoop - Local union



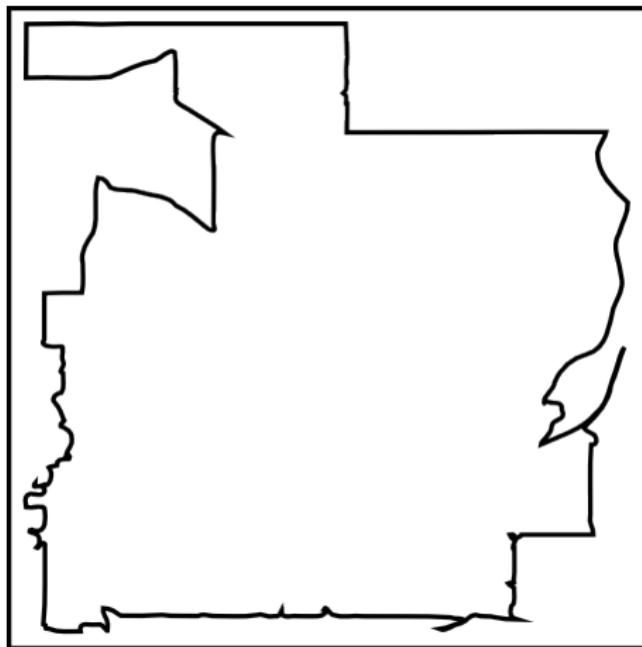
Partition **Local** Global

Hadoop - Global union



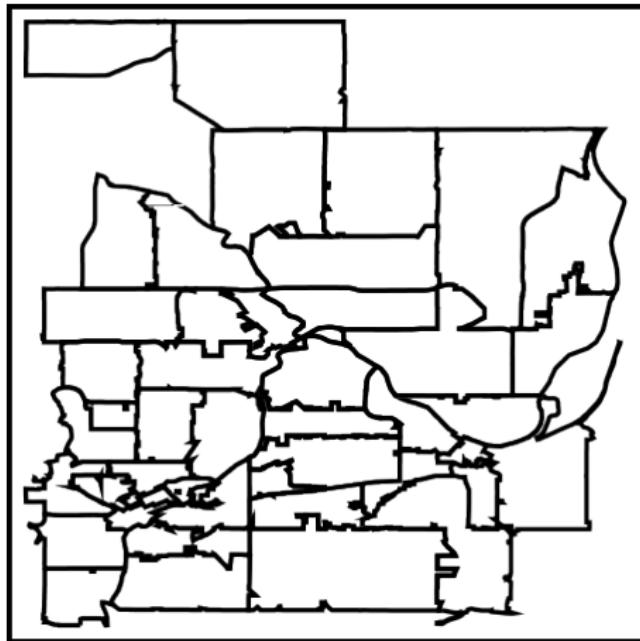
Partition Local **Global**

Hadoop - Global union



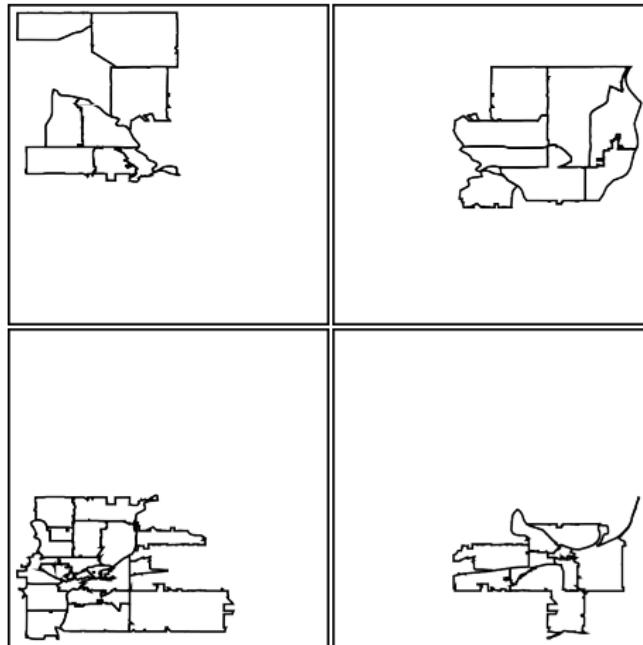
Partition Local **Global**

SpatialHadoop - Partition



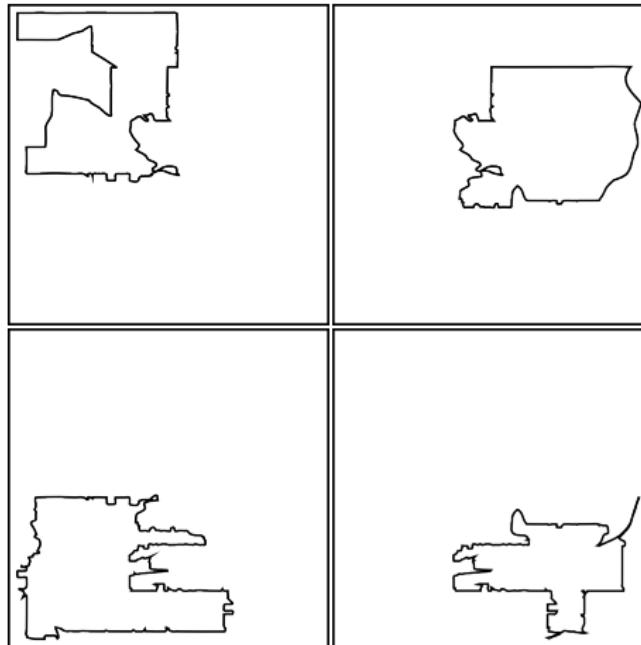
Partition Local Global

SpatialHadoop - Partition



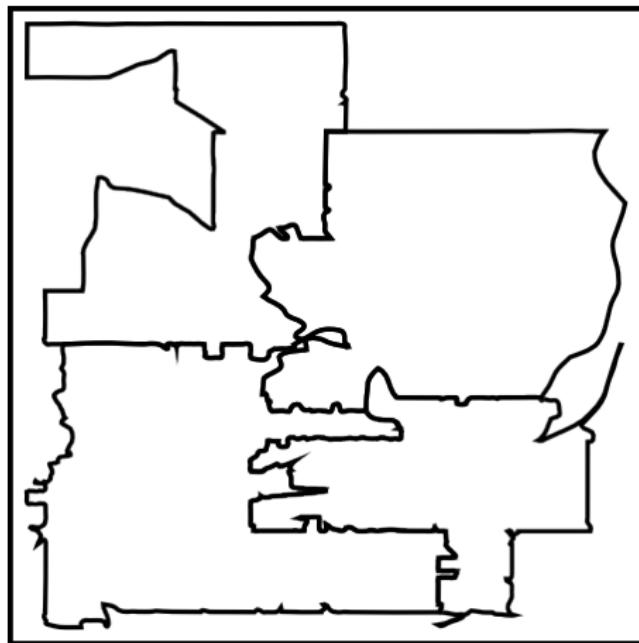
Partition Local Global

SpatialHadoop - Local union



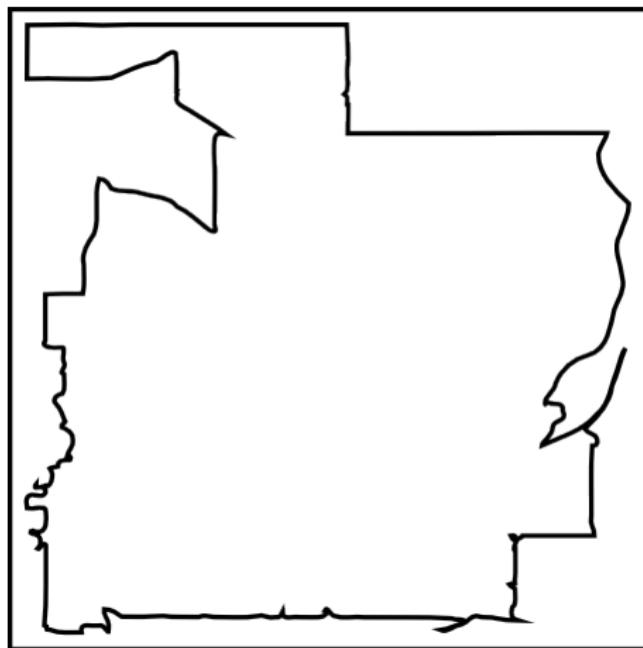
Partition **Local** Global

SpatialHadoop - Global union



Partition Local **Global**

SpatialHadoop - Global union



Partition Local **Global**

Agenda

1 Background

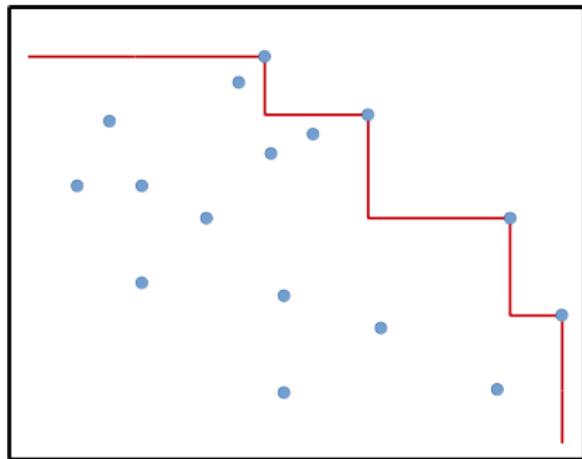
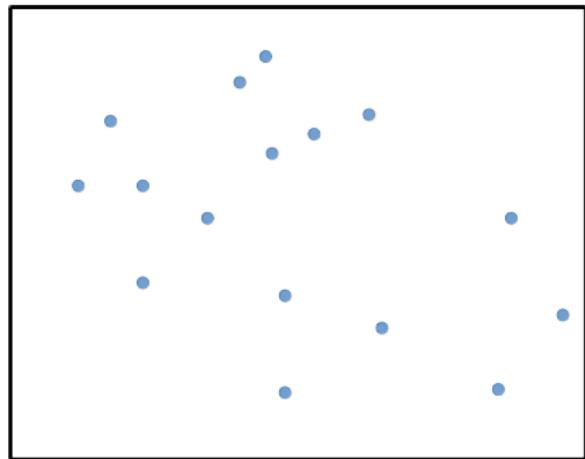
2 Computational Geometry Operations

- Union
- Skyline
- Convex Hull
- Farthest Pair
- Closest Pair

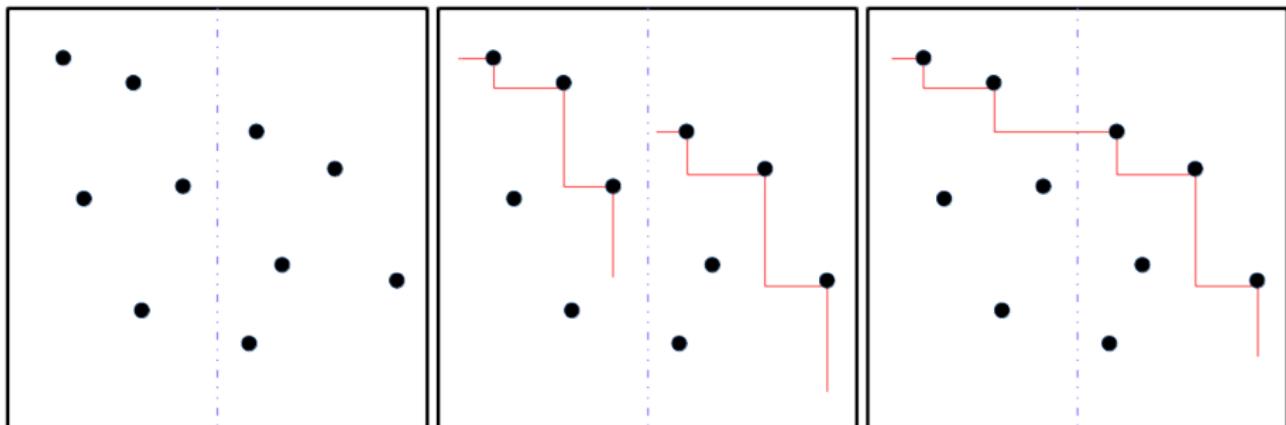
3 Experiments

4 Conclusions

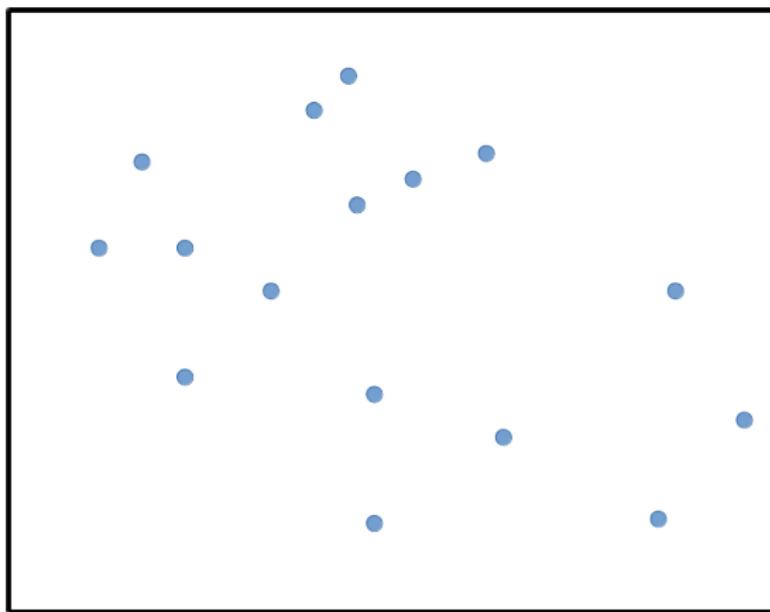
Skyline



Divide and conquer approach

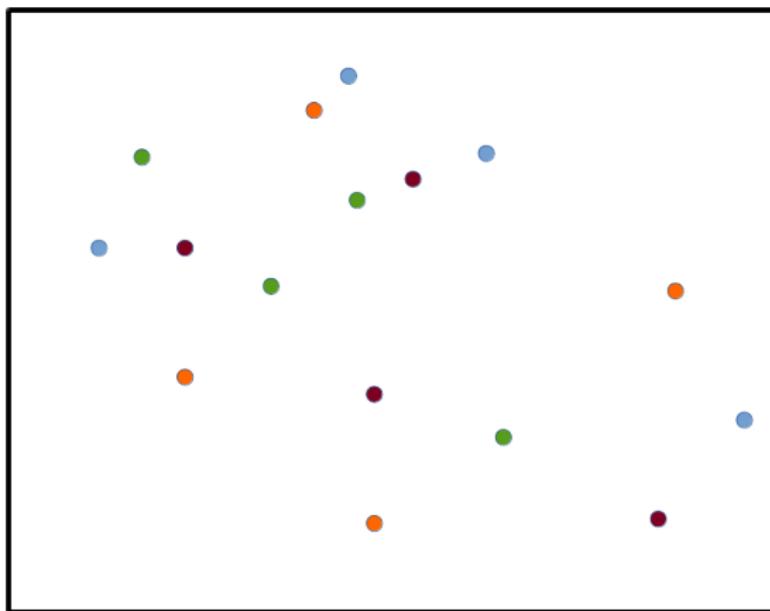


Hadoop - Partition



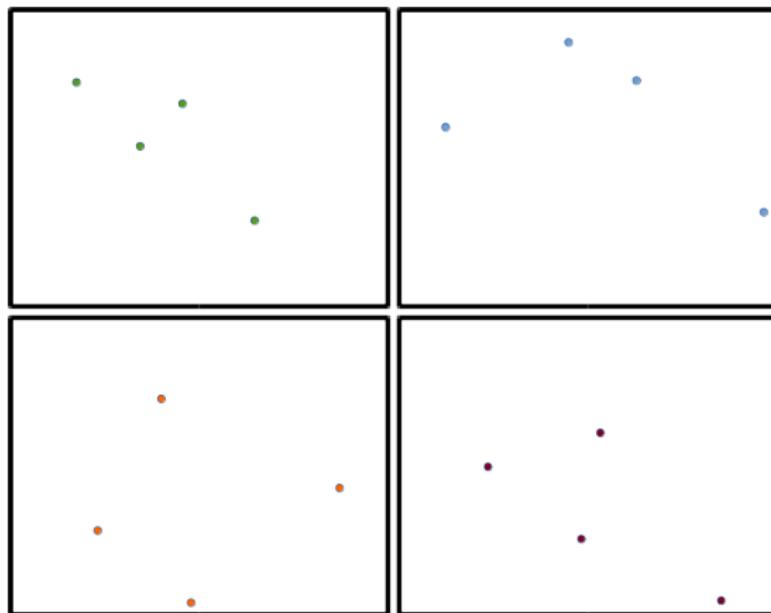
Partition Local Global

Hadoop - Partition



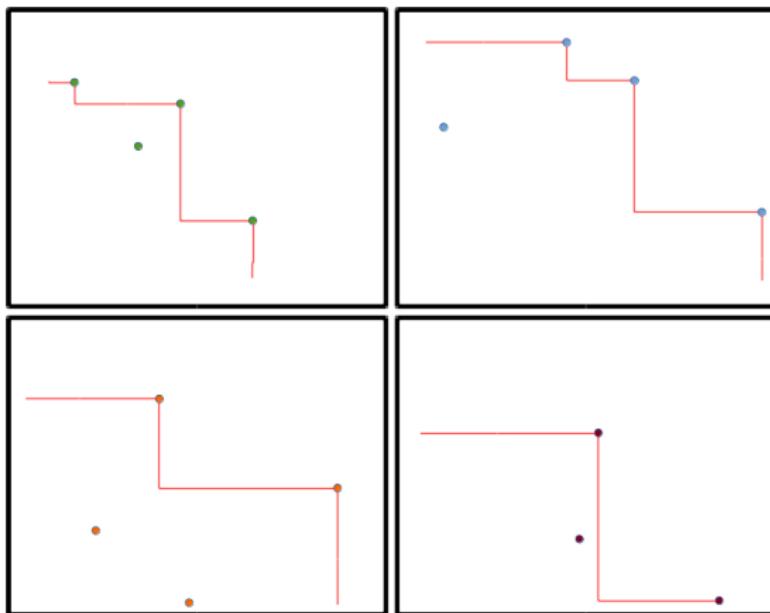
Partition Local Global

Hadoop - Partition



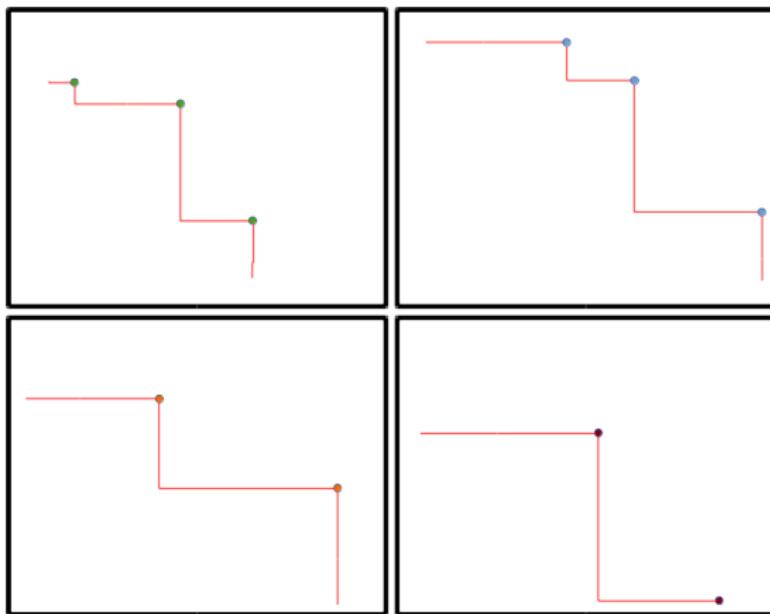
Partition Local Global

Hadoop - Local



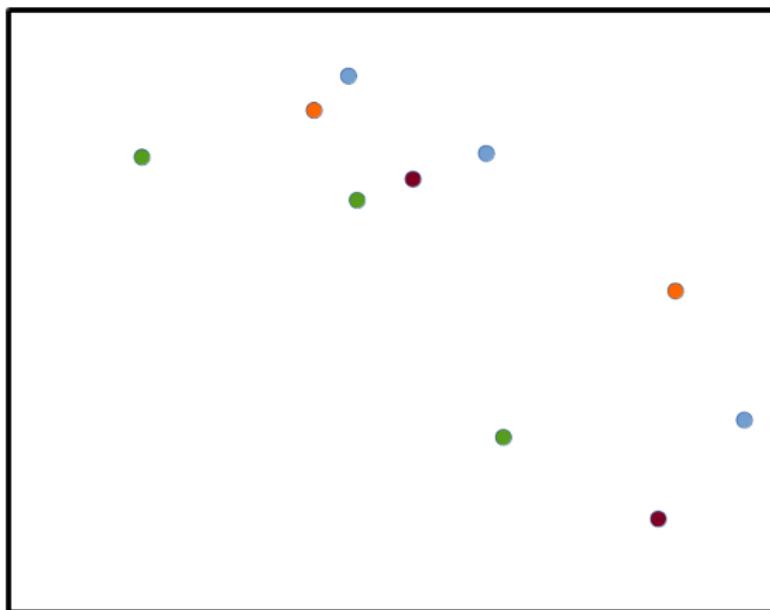
Partition **Local** Global

Hadoop - Local



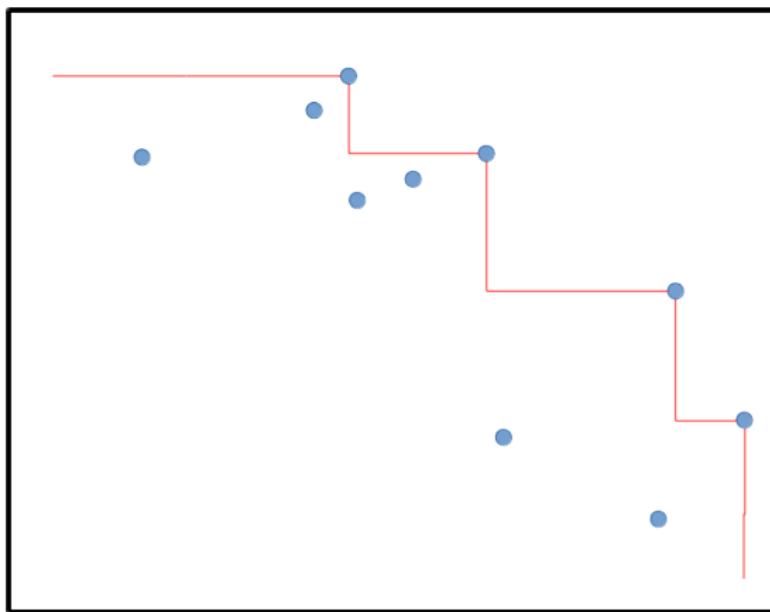
Partition **Local** Global

Hadoop - Global



Partition Local **Global**

Hadoop - Global



Partition Local **Global**

Skyline in SpatialHadoop

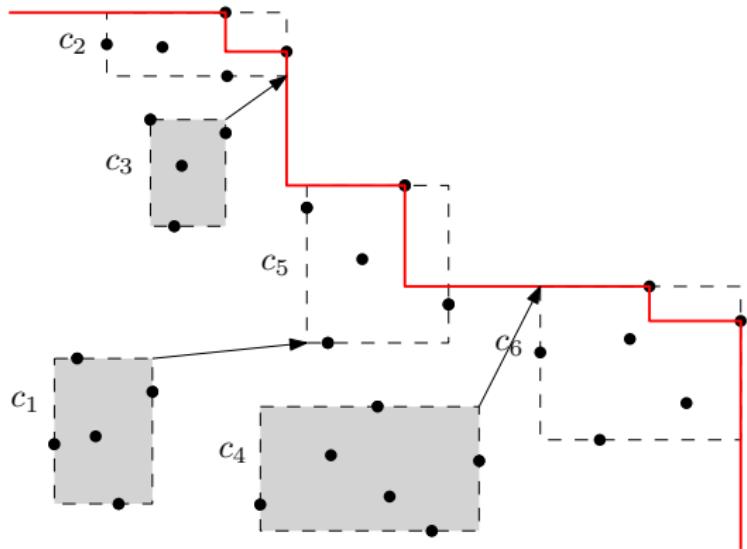
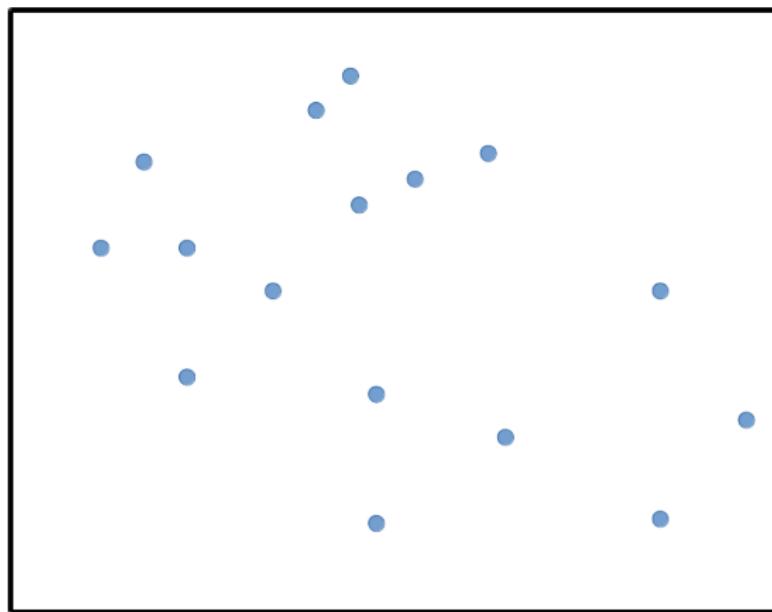


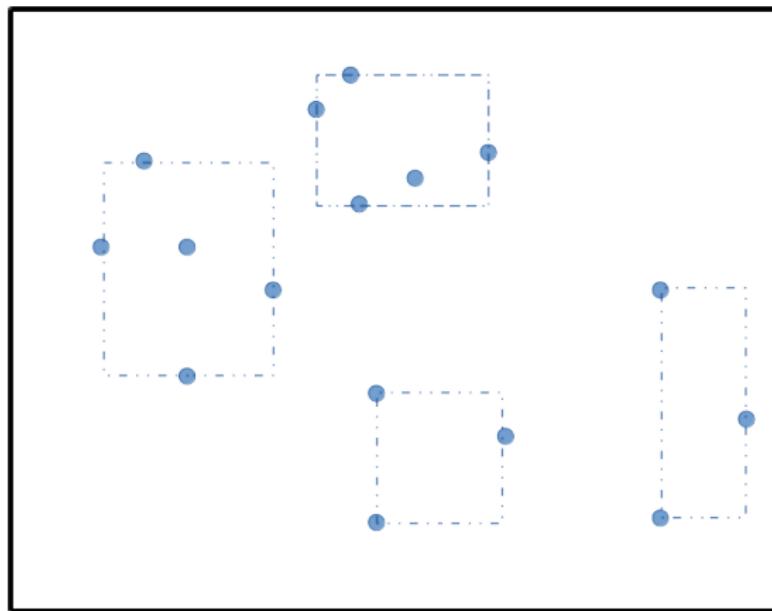
Figure 4: Skyline in SpatialHadoop

SpatialHadoop - Partition



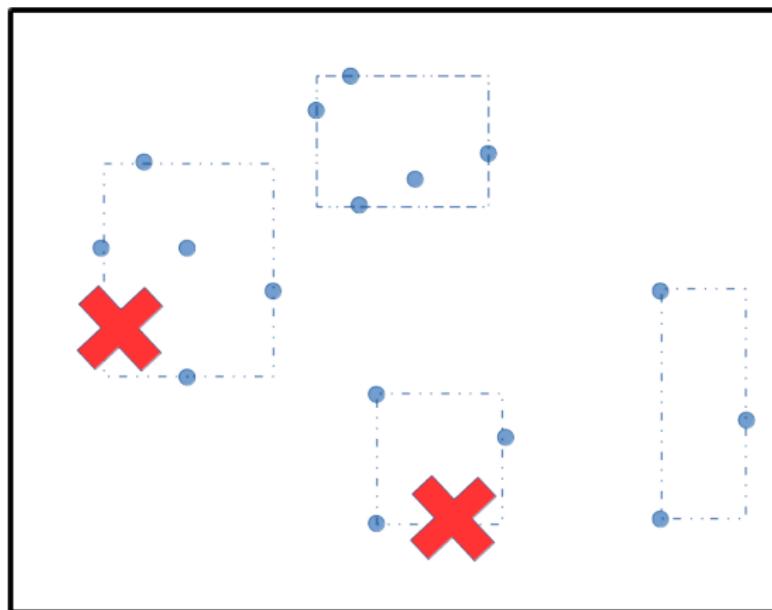
Partition Pruning Local Global

SpatialHadoop - Partition



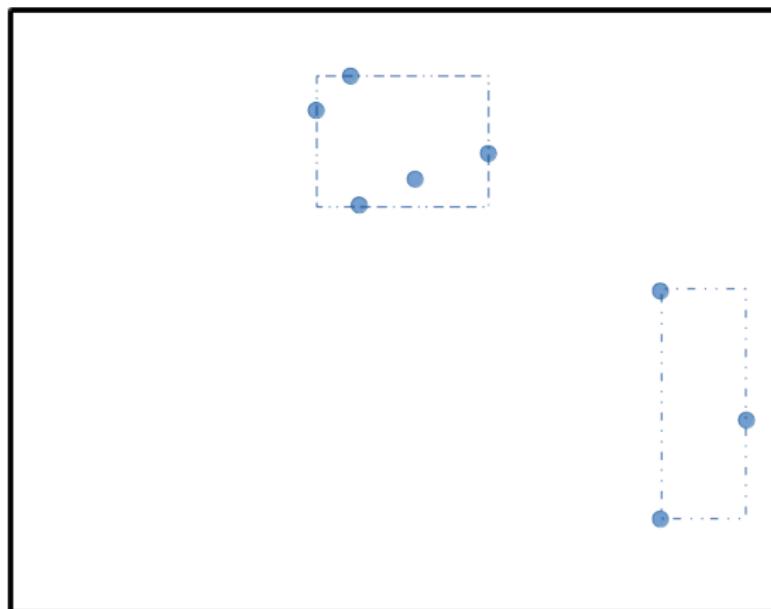
Partition Pruning Local Global

SpatialHadoop - Pruning



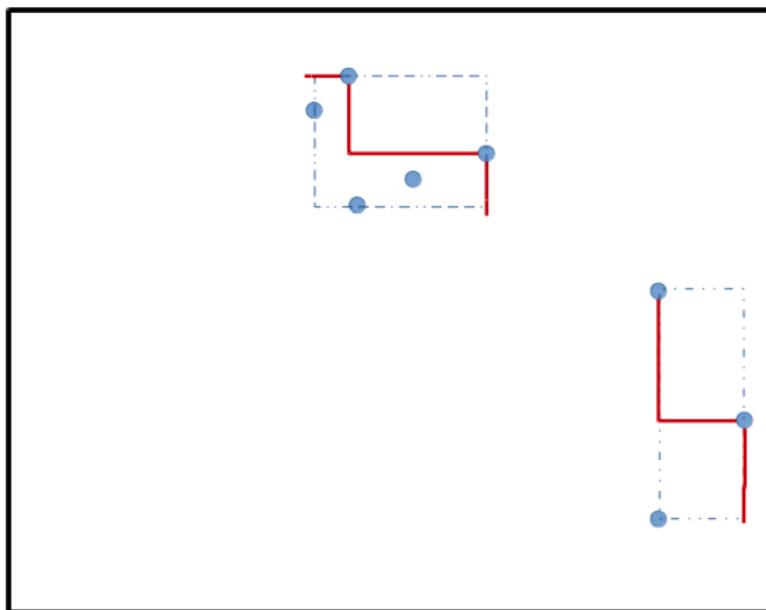
Partition **Pruning** Local Global

SpatialHadoop - Pruning



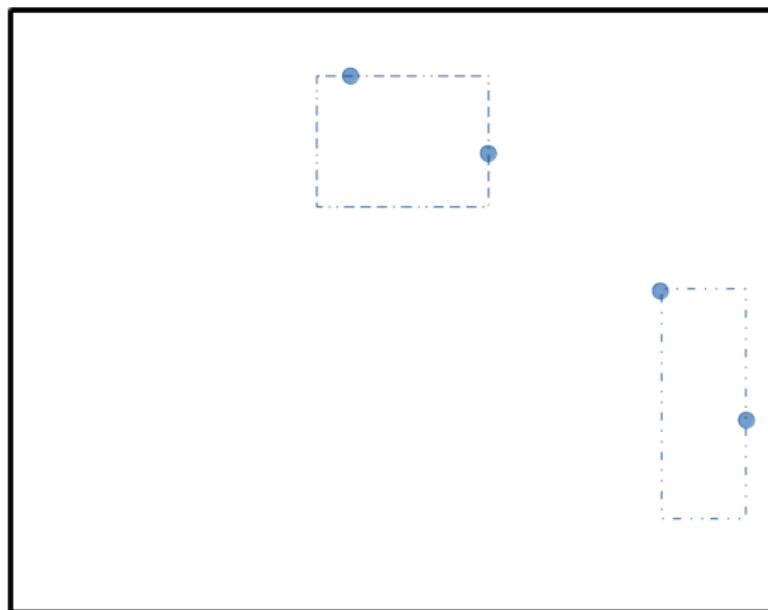
Partition **Pruning** Local Global

SpatialHadoop - Local



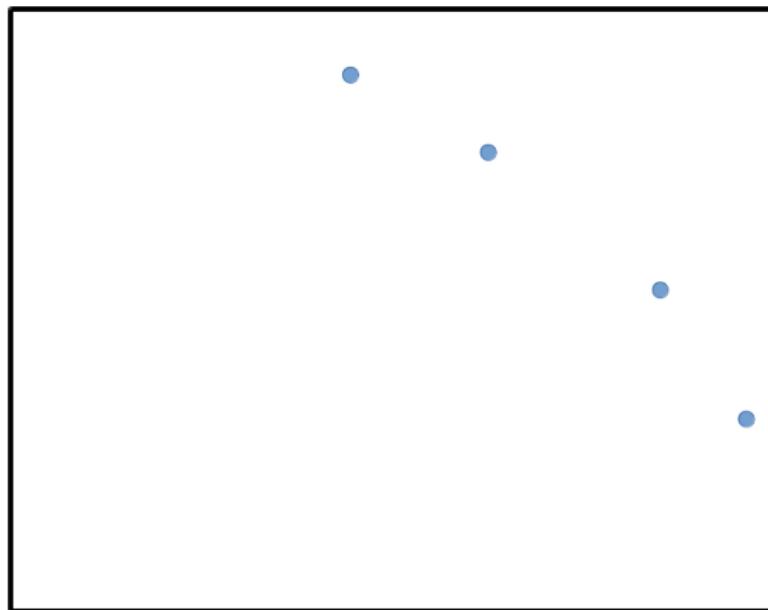
Partition Pruning **Local** Global

SpatialHadoop - Local



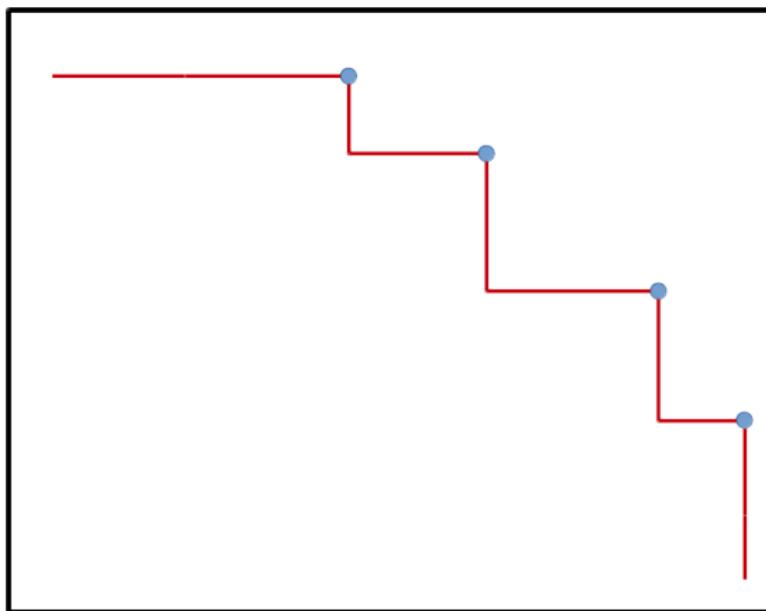
Partition Pruning **Local** Global

SpatialHadoop - Global



Partition Pruning Local **Global**

SpatialHadoop - Global



Partition Pruning Local **Global**

Agenda

1 Background

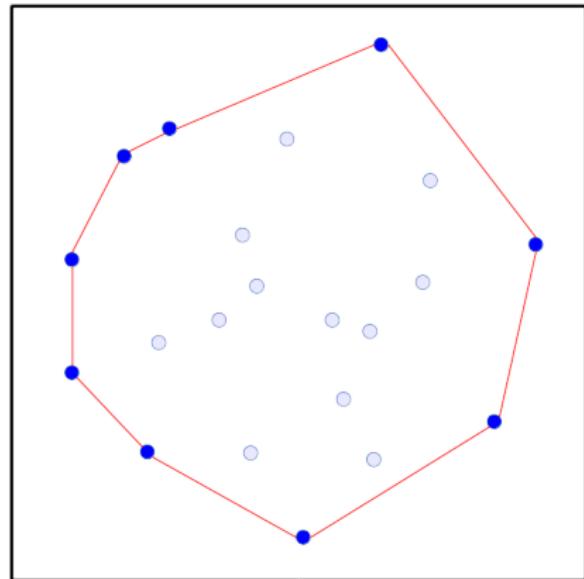
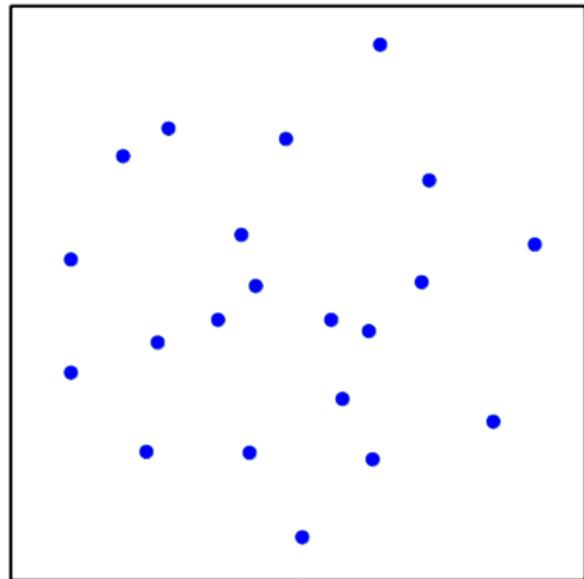
2 Computational Geometry Operations

- Union
- Skyline
- Convex Hull
- Farthest Pair
- Closest Pair

3 Experiments

4 Conclusions

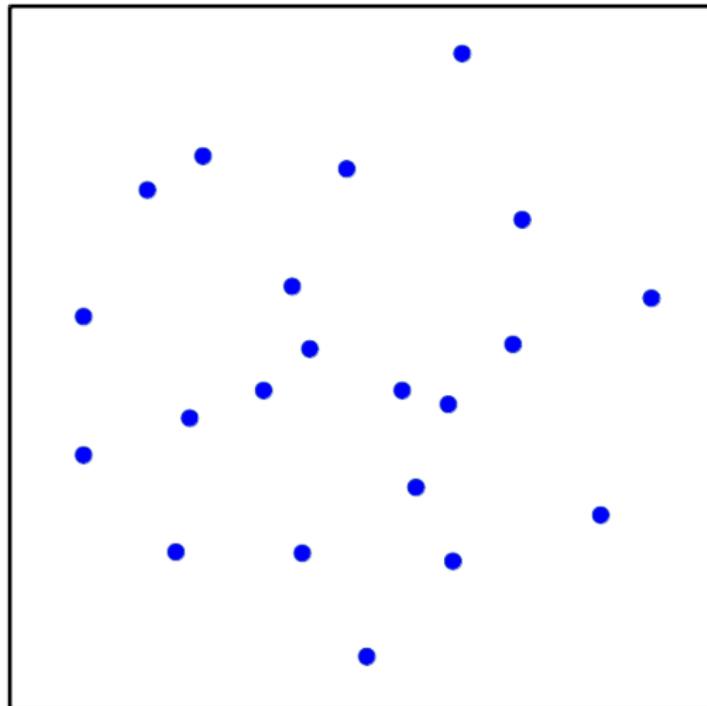
Convex Hull



Andrew's Monotone Chain algorithm

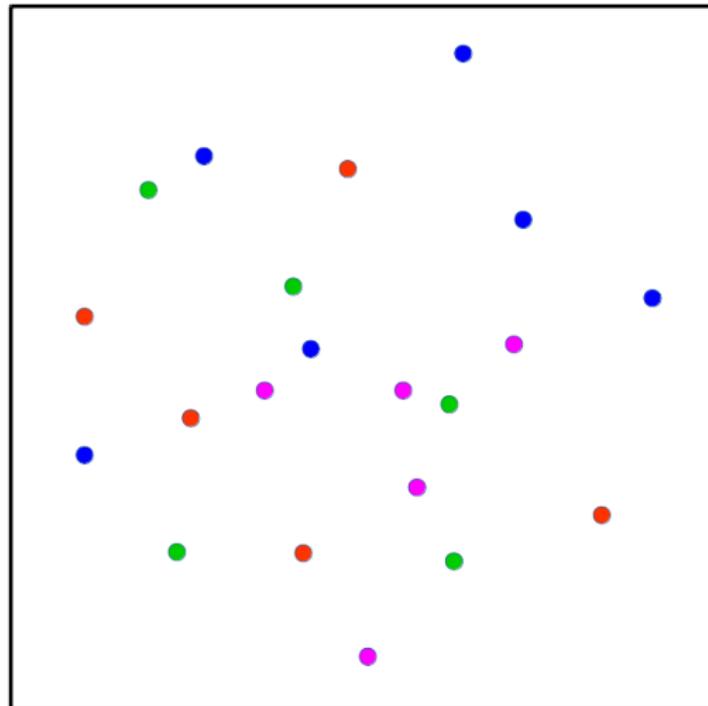
https://en.wikibooks.org/wiki/Algorithm_Implementation/Geometry/Convex_hull/Monotone_chain

Hadoop - Partition



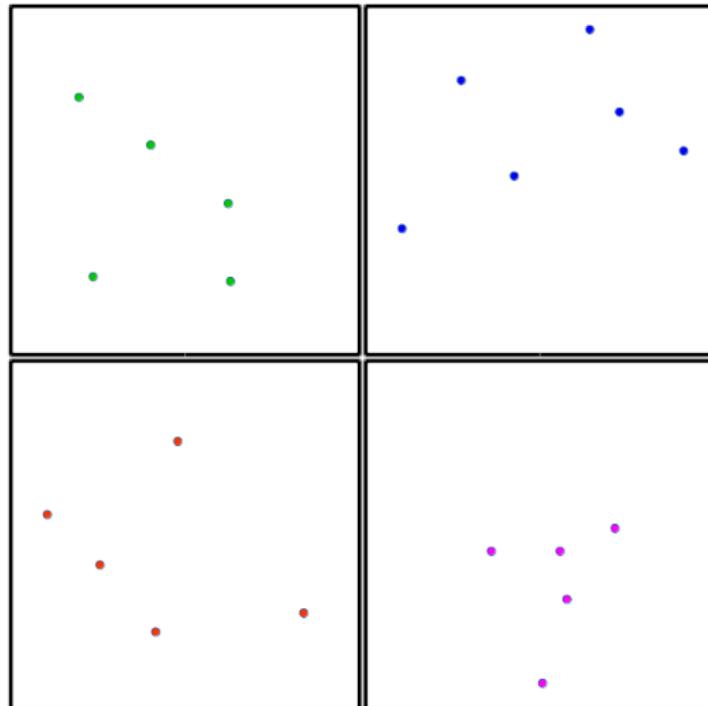
Partition Local Global

Hadoop - Partition



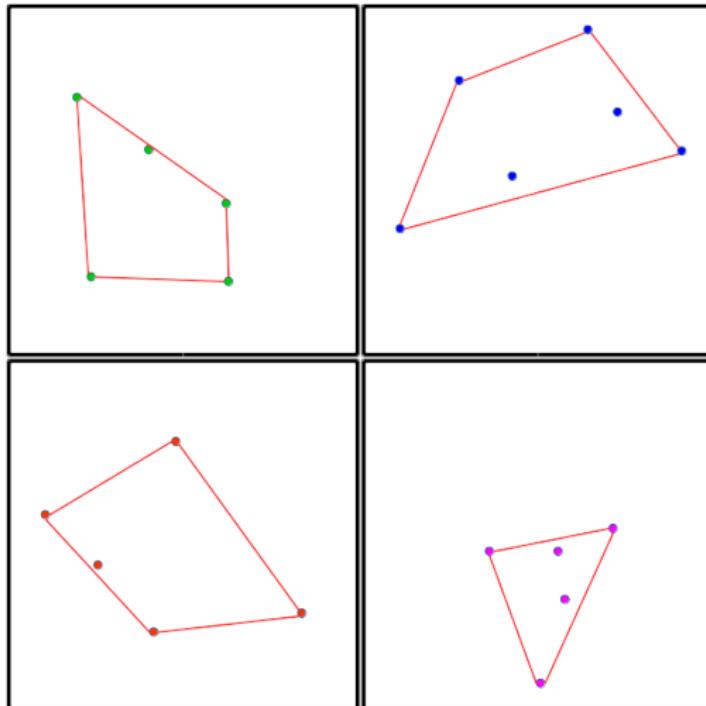
Partition Local Global

Hadoop - Partition



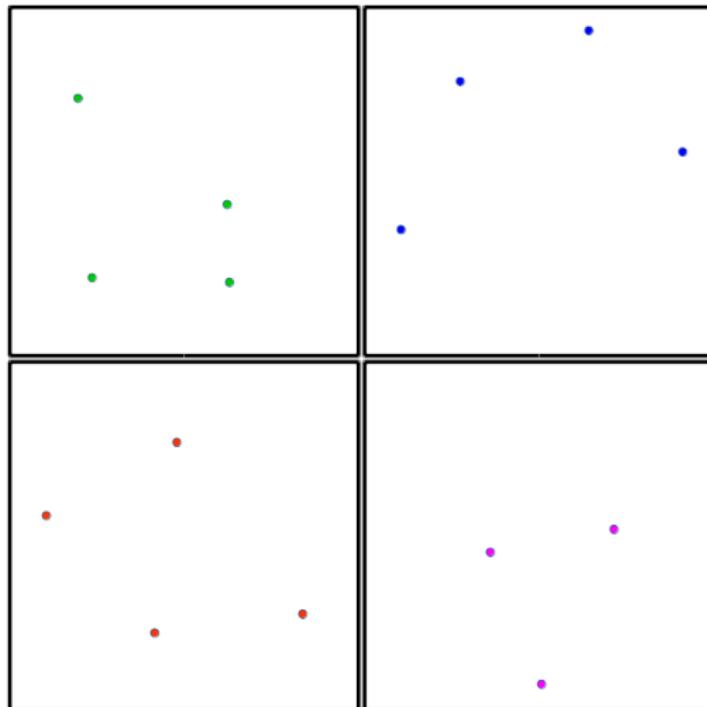
Partition Local Global

Hadoop - Local



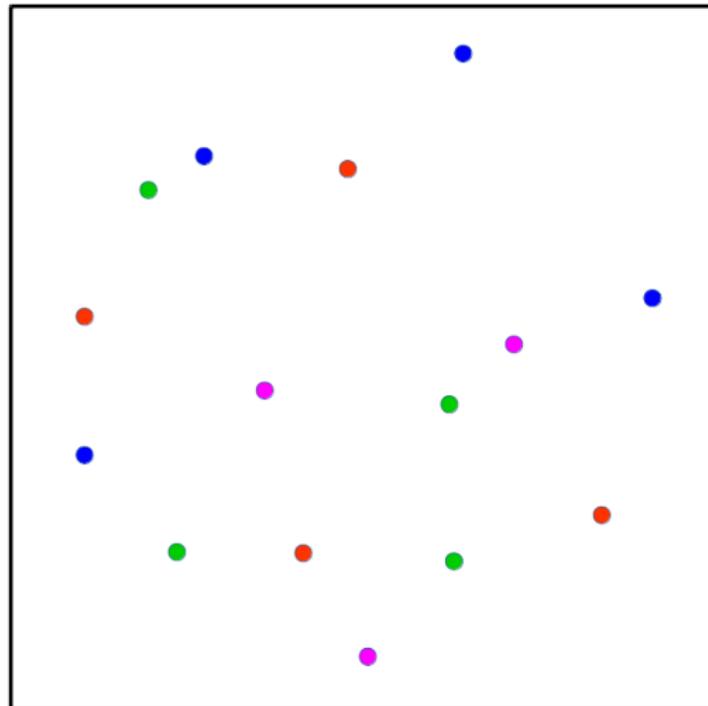
Partition **Local** Global

Hadoop - Local



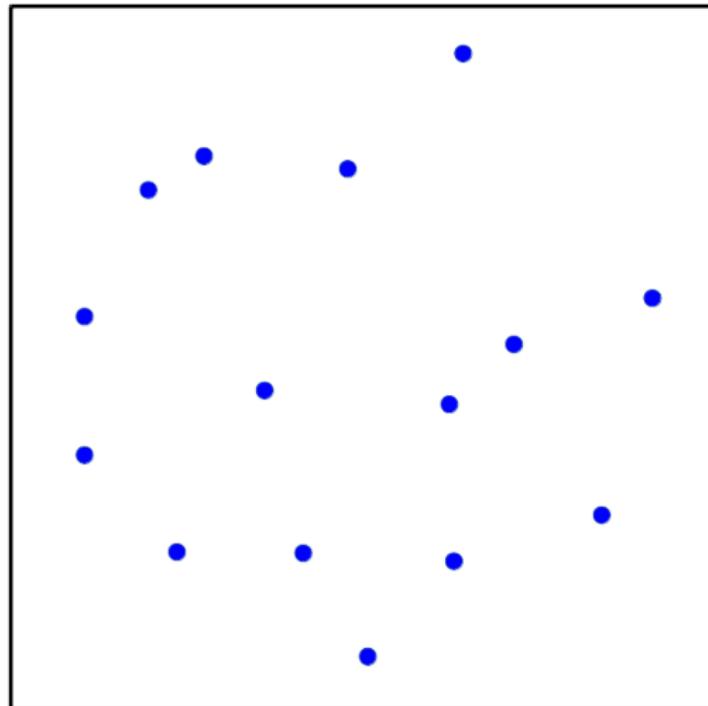
Partition **Local** Global

Hadoop - Global



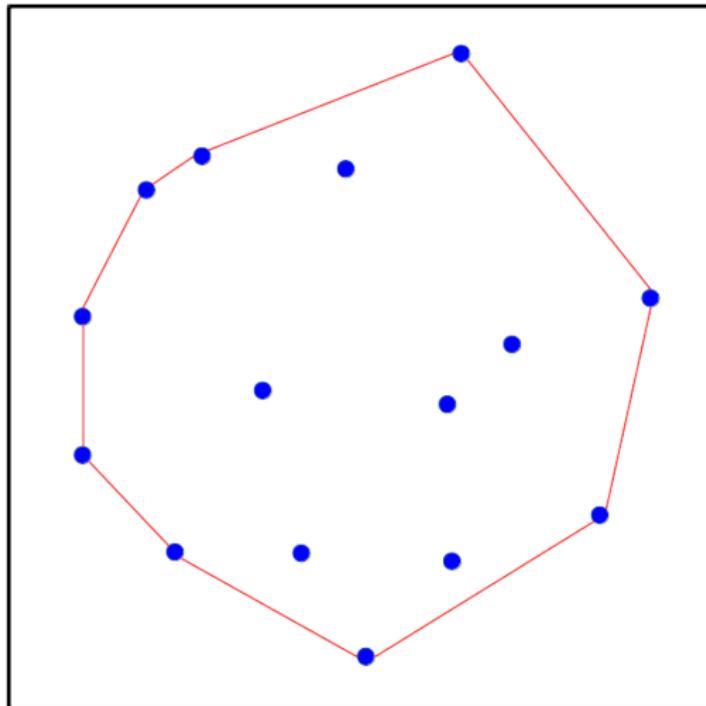
Partition Local **Global**

Hadoop - Global



Partition Local **Global**

Hadoop - Global



Partition Local **Global**

Convex hull in SpatialHadoop

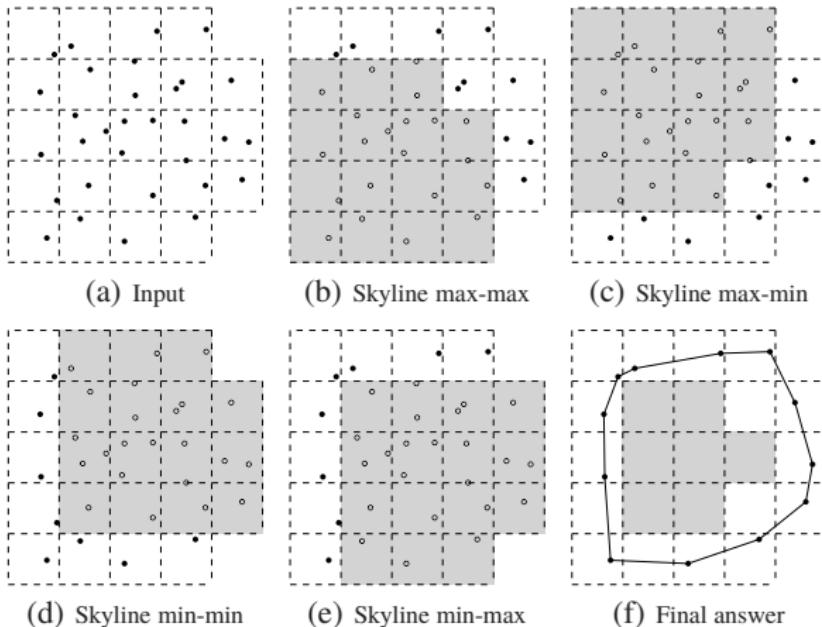
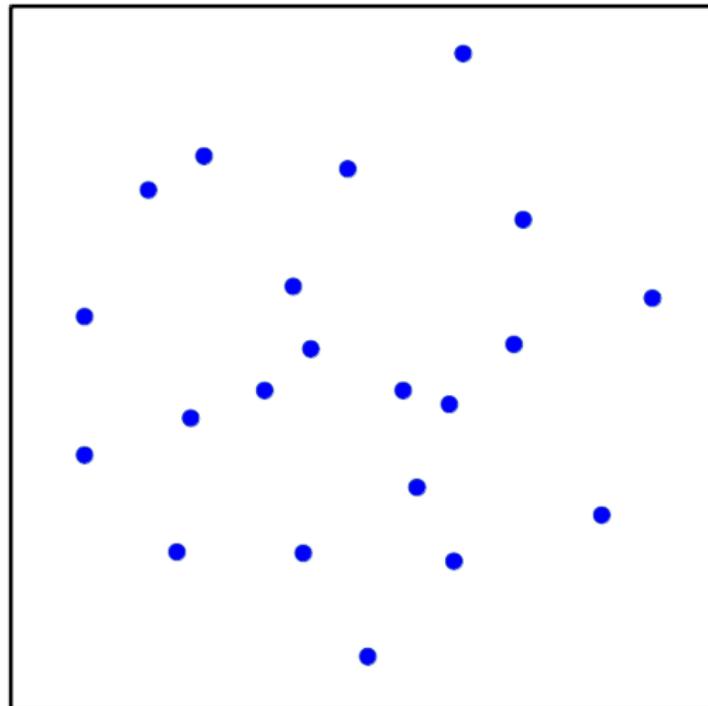


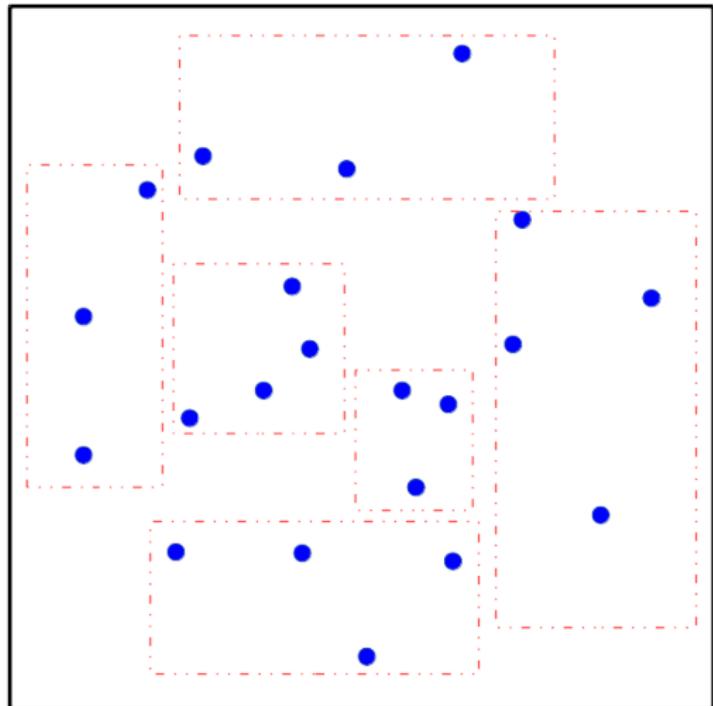
Figure 5: Convex hull in SpatialHadoop

SpatialHadoop - Partition



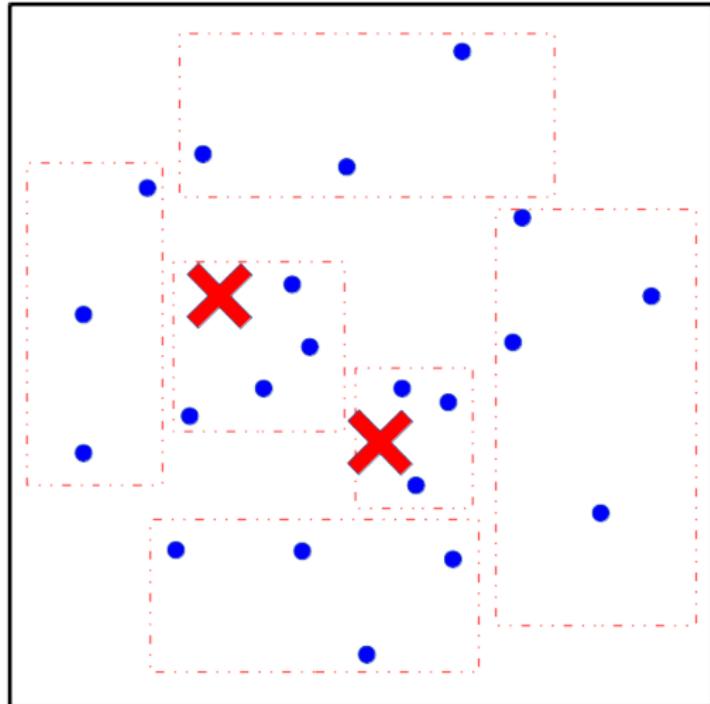
Partition Pruning Local Global

SpatialHadoop - Partition



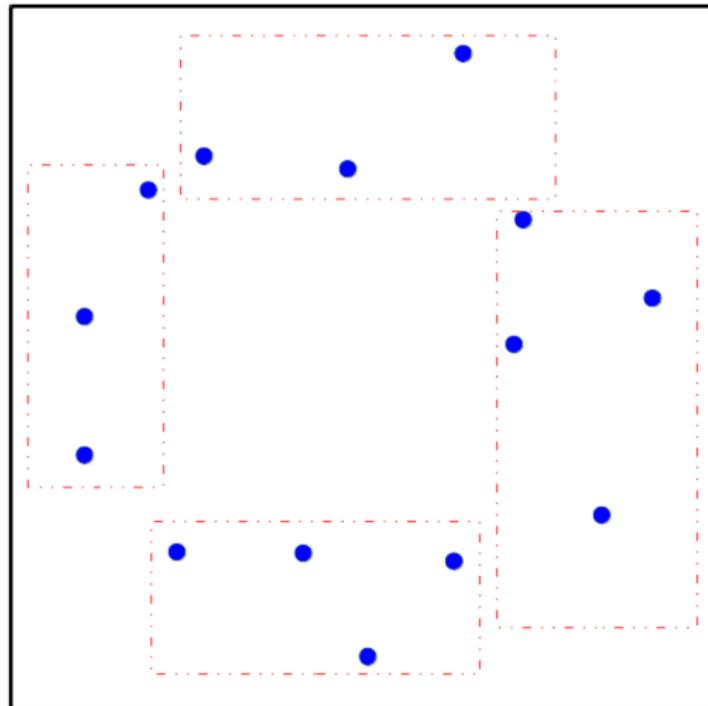
Partition Pruning Local Global

SpatialHadoop - Pruning



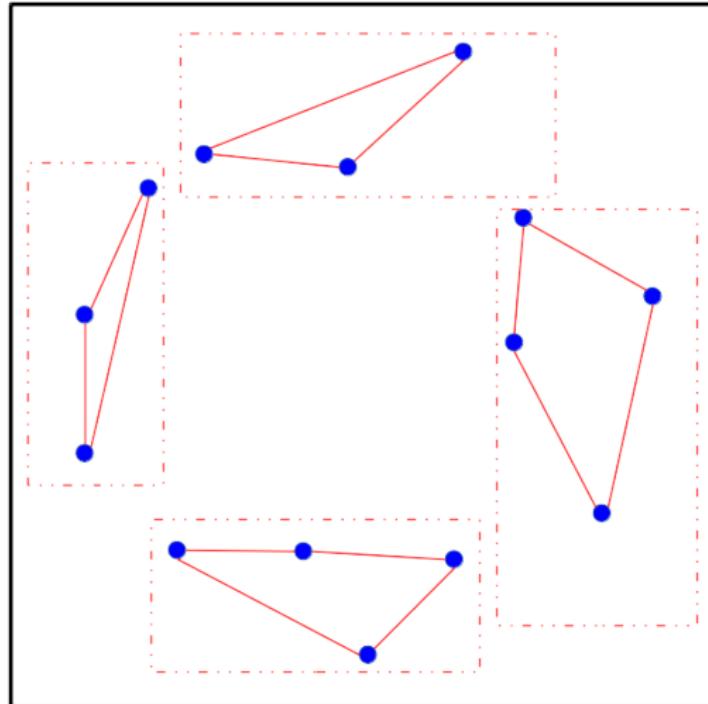
Partition **Pruning** Local Global

SpatialHadoop - Pruning



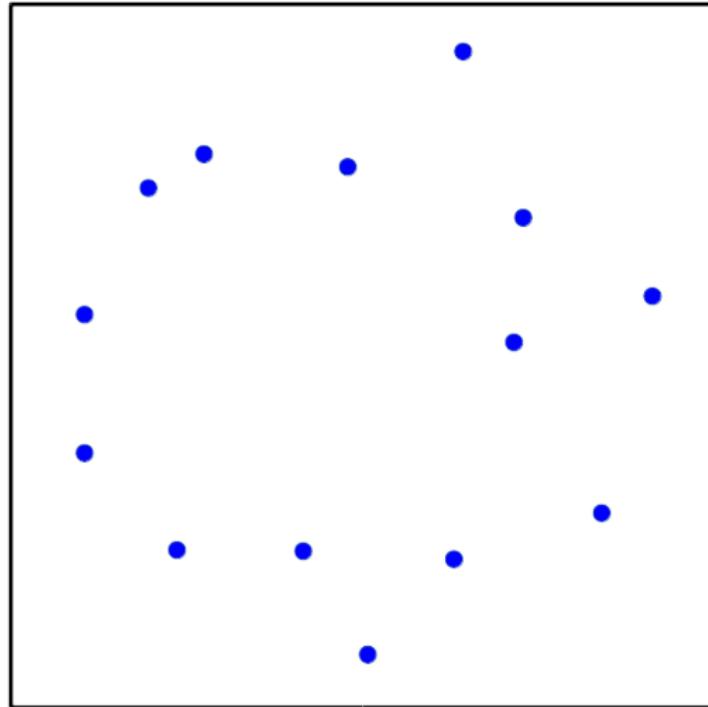
Partition **Pruning** Local Global

SpatialHadoop - Local



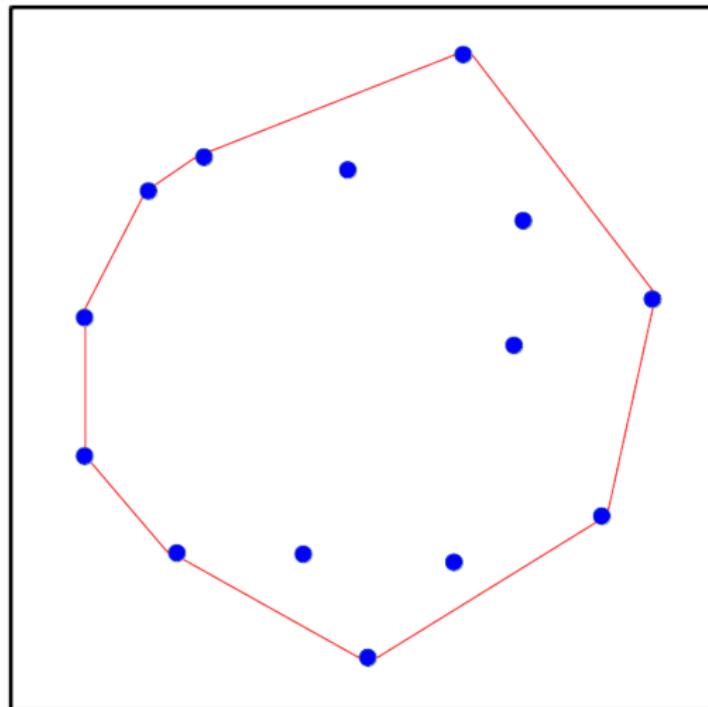
Partition Pruning **Local** Global

SpatialHadoop - Global



Partition Pruning Local **Global**

SpatialHadoop - Global



Partition Pruning Local **Global**

Agenda

1 Background

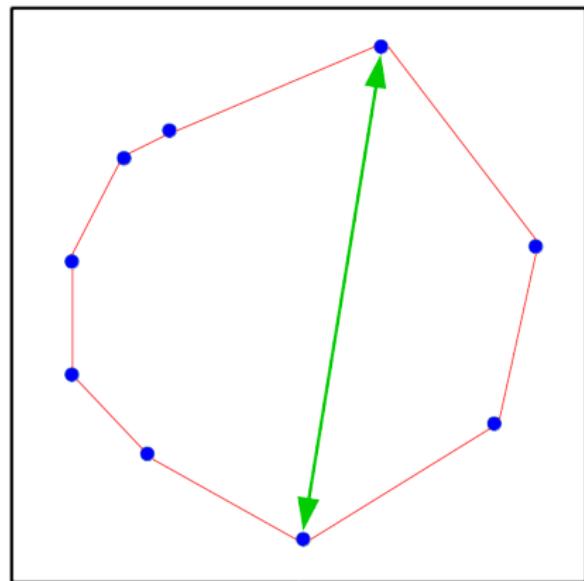
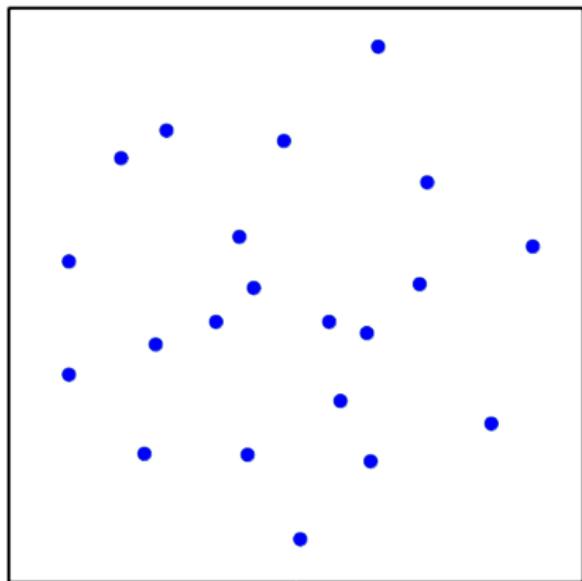
2 Computational Geometry Operations

- Union
- Skyline
- Convex Hull
- Farthest Pair
- Closest Pair

3 Experiments

4 Conclusions

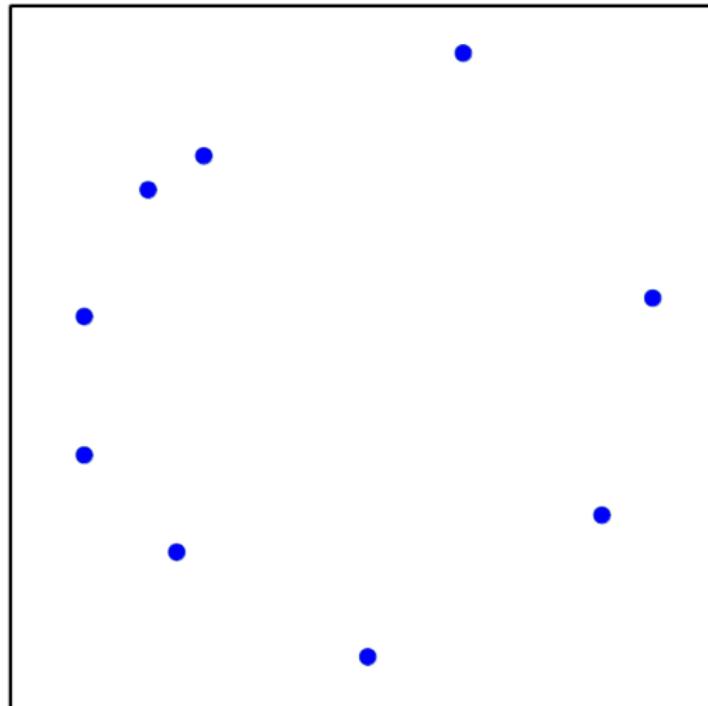
Farthest Pair



Rotating Calipers method

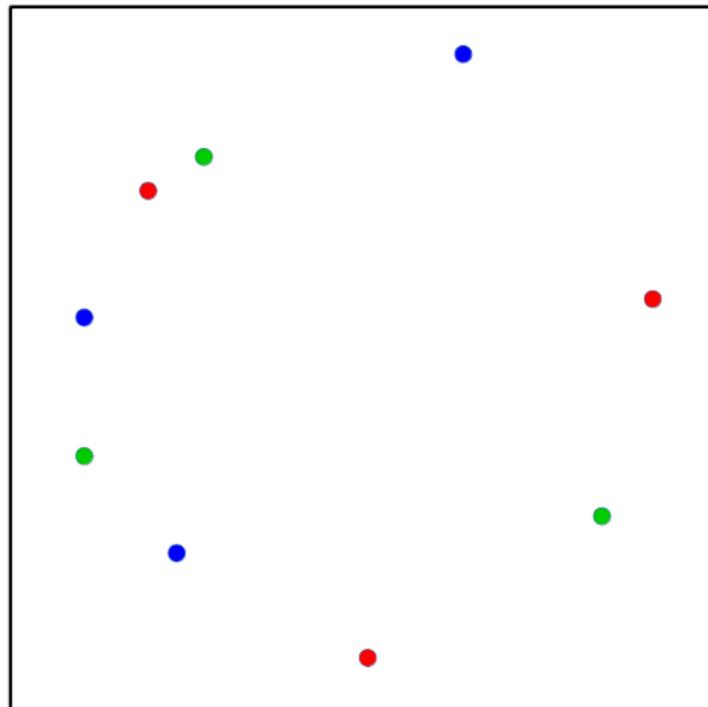
<http://www-cg.r1.cs.mcgill.ca/~godfried/research/calipers.html>

Hadoop - Partition



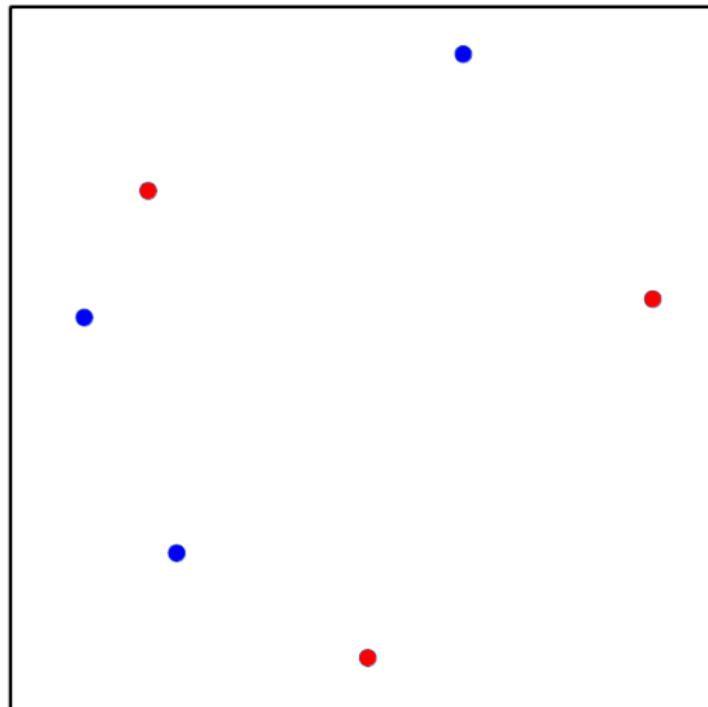
Partition Local Global

Hadoop - Partition



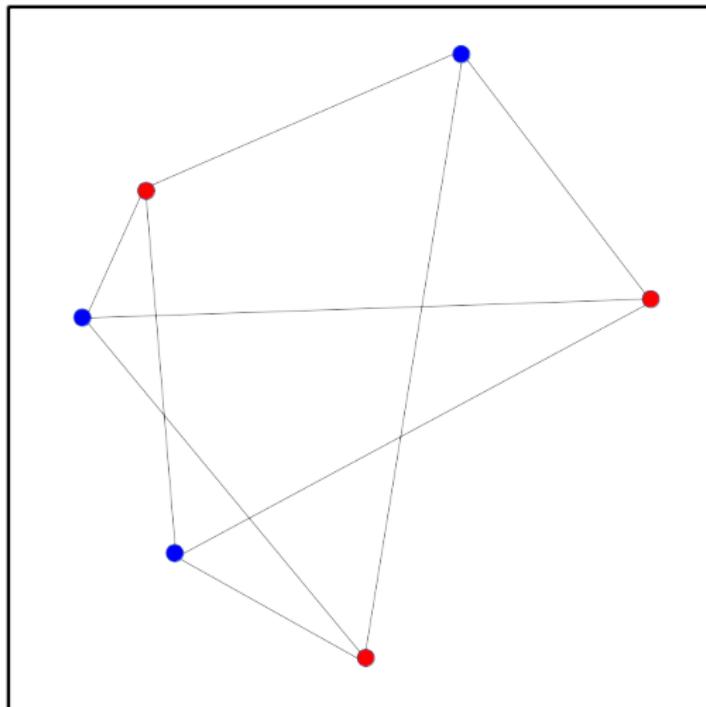
Partition Local Global

Hadoop - Local



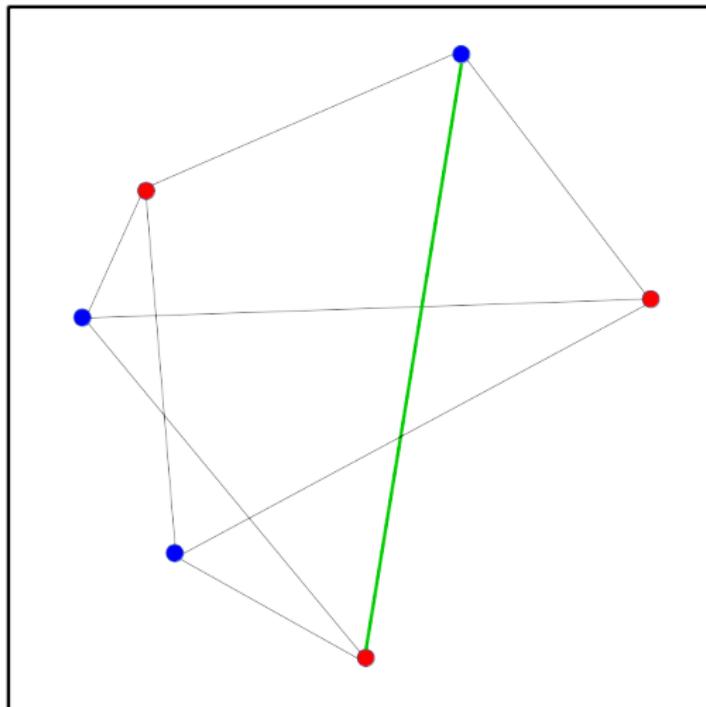
Partition **Local** Global

Hadoop - Local



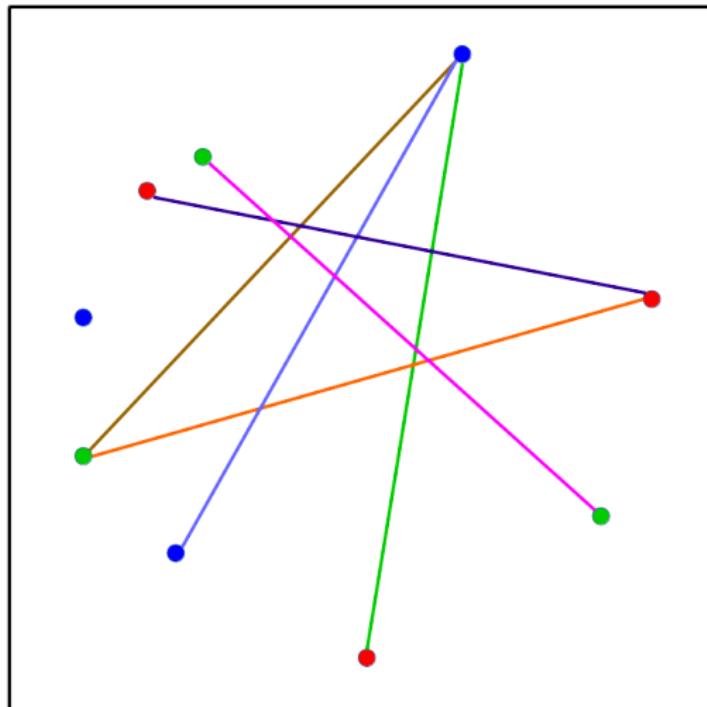
Partition **Local** Global

Hadoop - Local



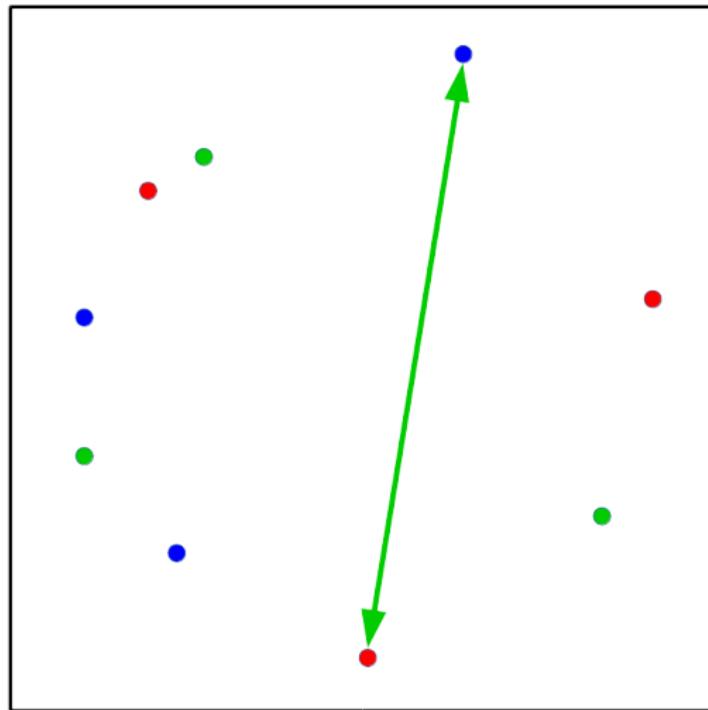
Partition **Local** Global

Hadoop - Global



Partition Local **Global**

Hadoop - Global



Partition Local **Global**

Farthest pair in SpatialHadoop

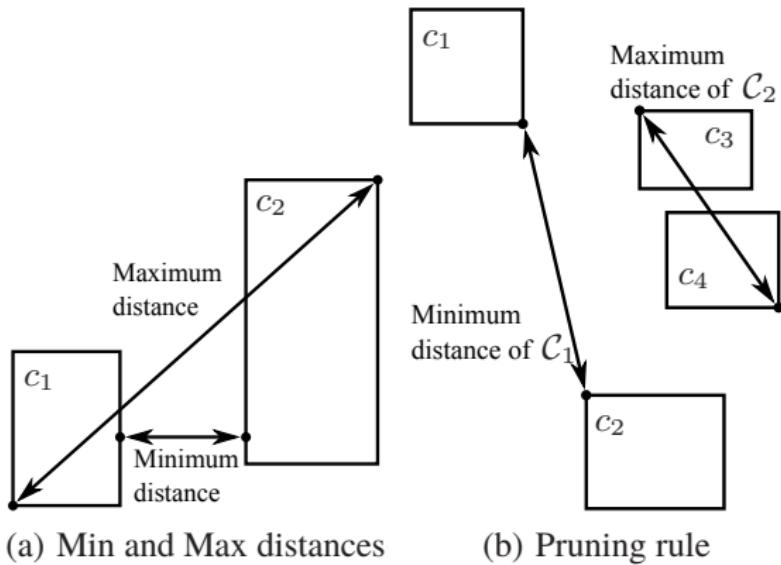
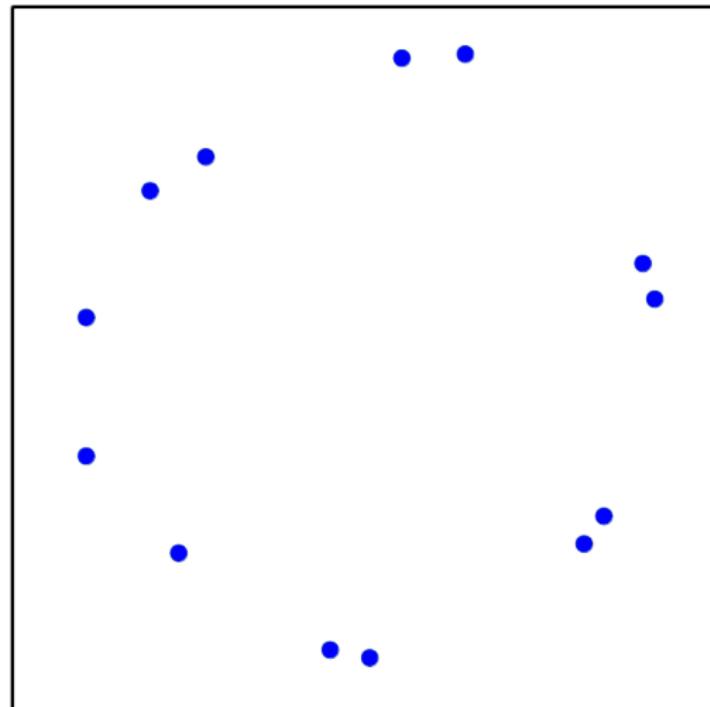


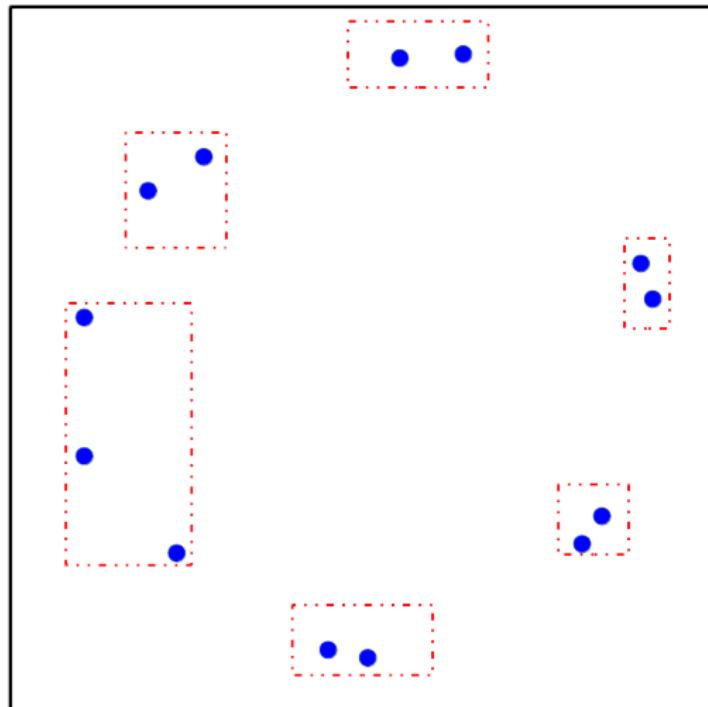
Figure 6: Farthest pair algorithm in SpatialHadoop

SpatialHadoop - Partition



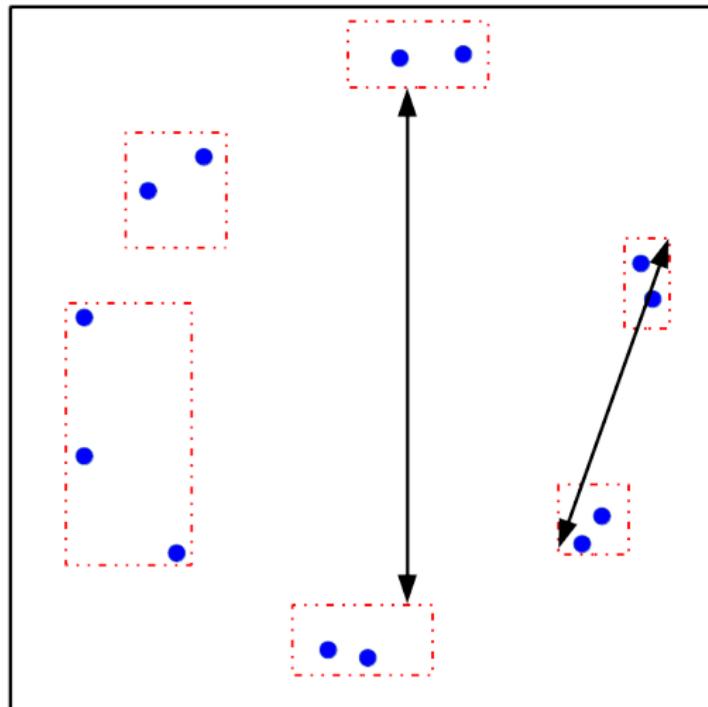
Partition Pruning Local Global

SpatialHadoop - Partition



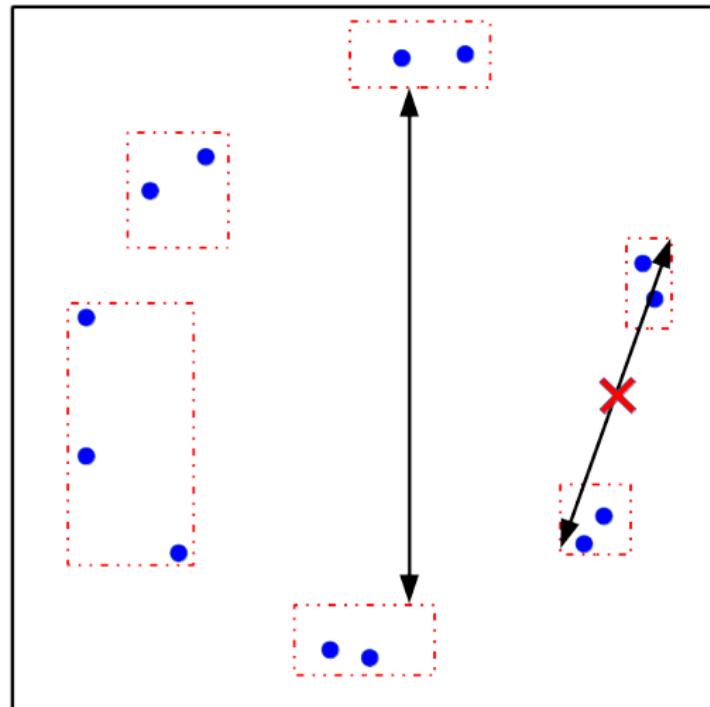
Partition Pruning Local Global

SpatialHadoop - Pruning



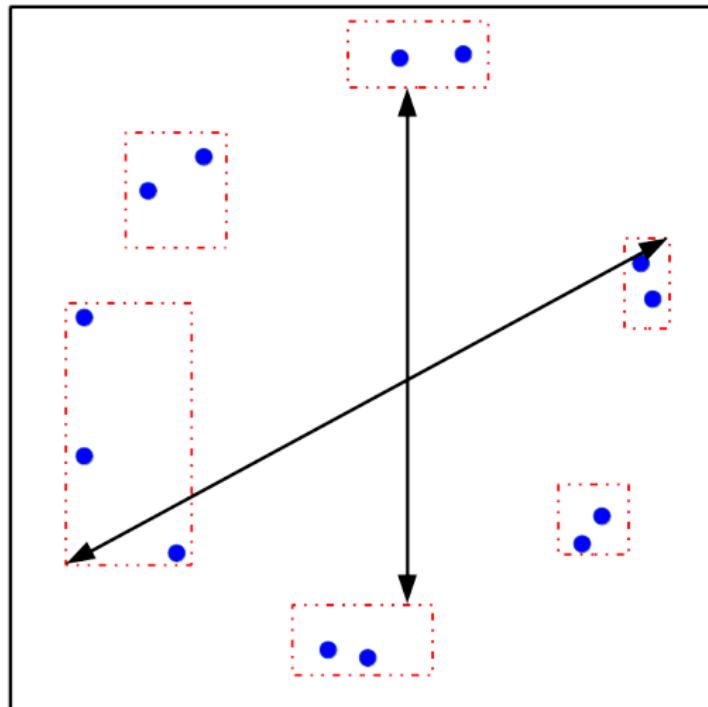
Partition **Pruning** Local Global

SpatialHadoop - Pruning



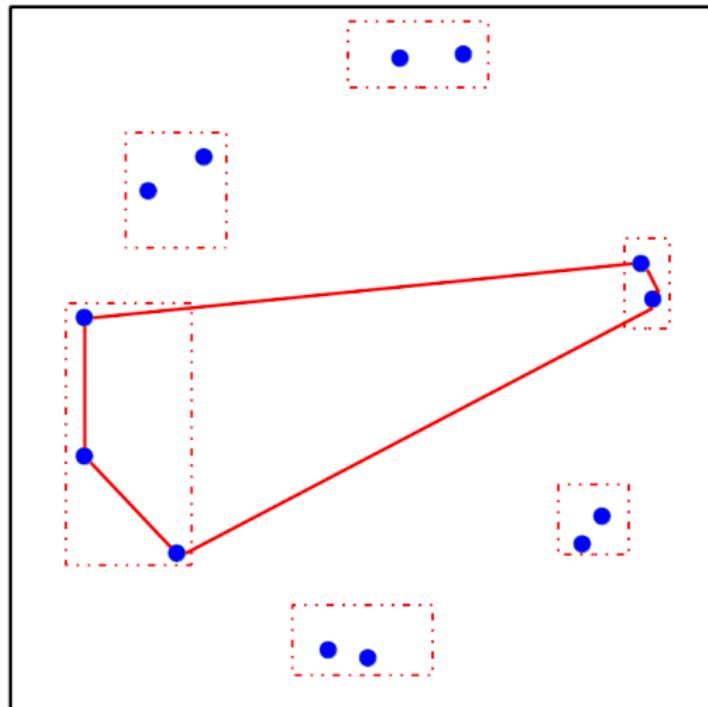
Partition **Pruning** Local Global

SpatialHadoop - Pruning



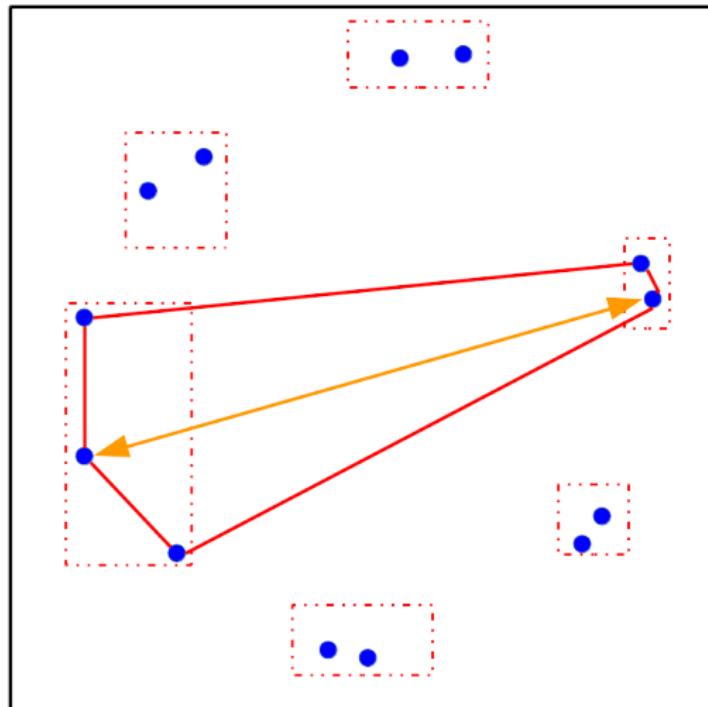
Partition **Pruning** Local Global

SpatialHadoop - Local



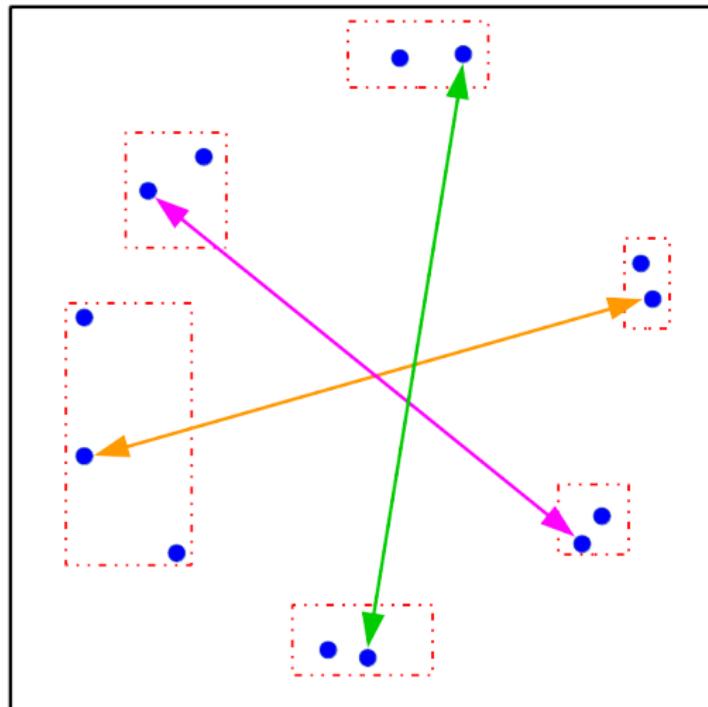
Partition Pruning **Local** Global

SpatialHadoop - Local



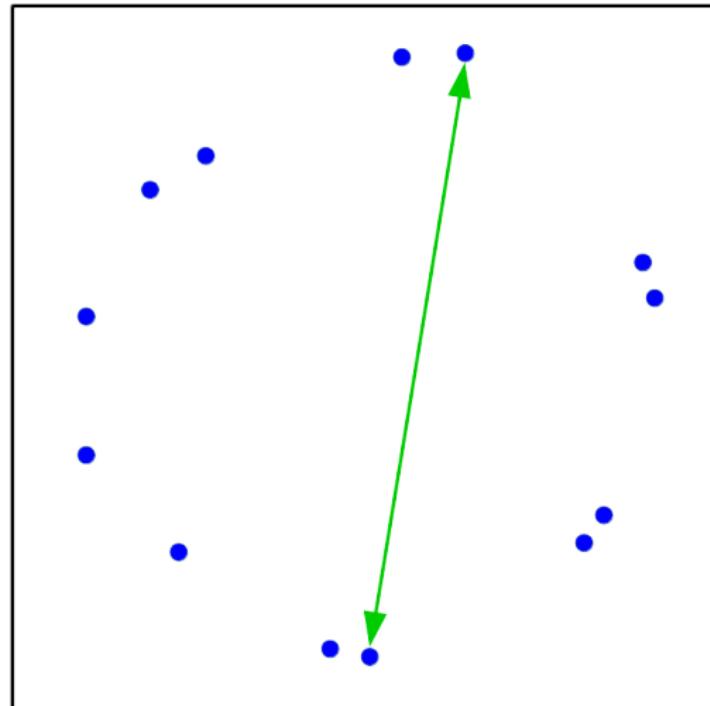
Partition Pruning **Local** Global

SpatialHadoop - Global



Partition Pruning Local **Global**

SpatialHadoop - Global



Partition Pruning Local **Global**

Agenda

1 Background

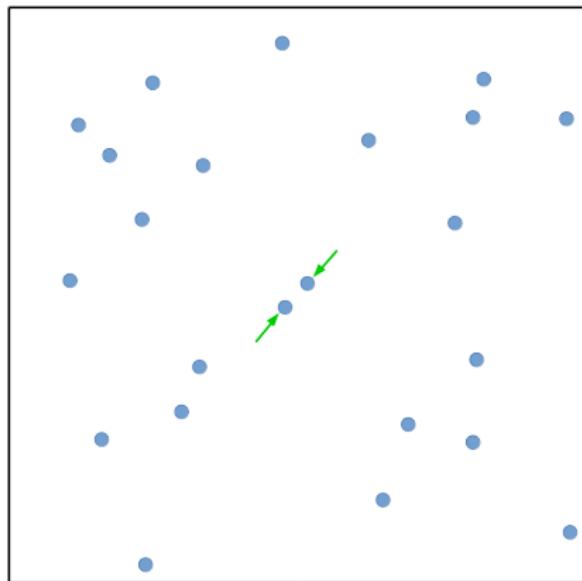
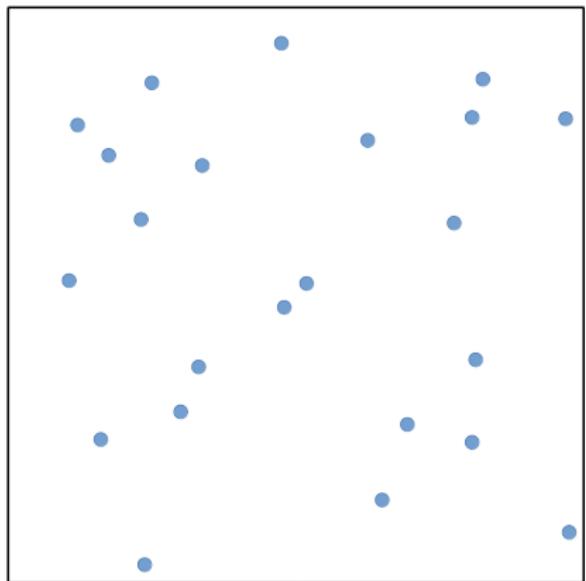
2 Computational Geometry Operations

- Union
- Skyline
- Convex Hull
- Farthest Pair
- Closest Pair

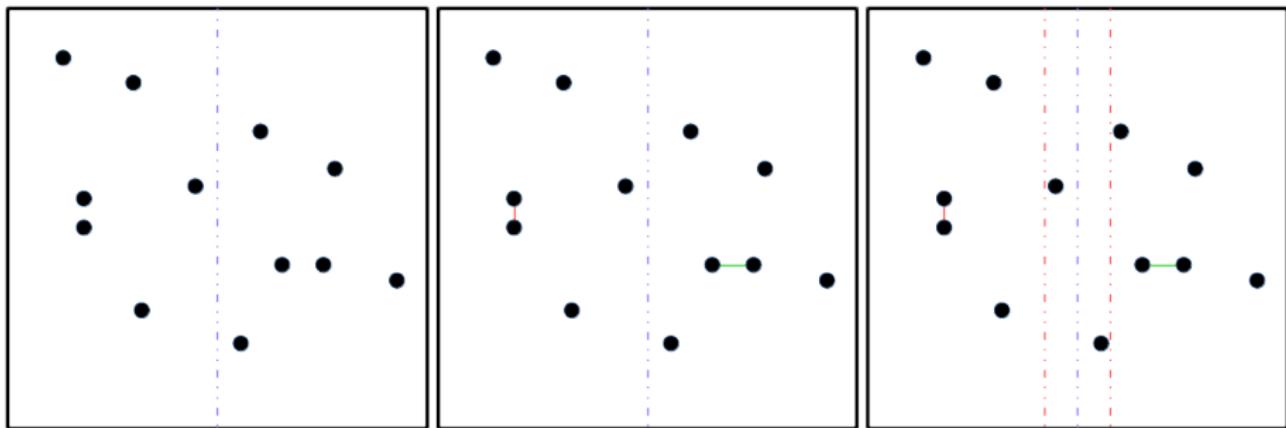
3 Experiments

4 Conclusions

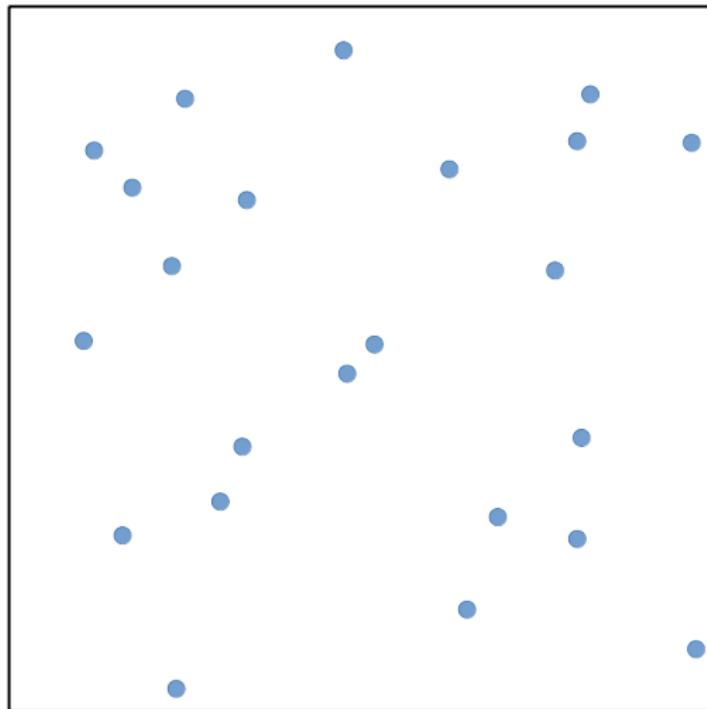
Closest Pair



Divide and conquer approach

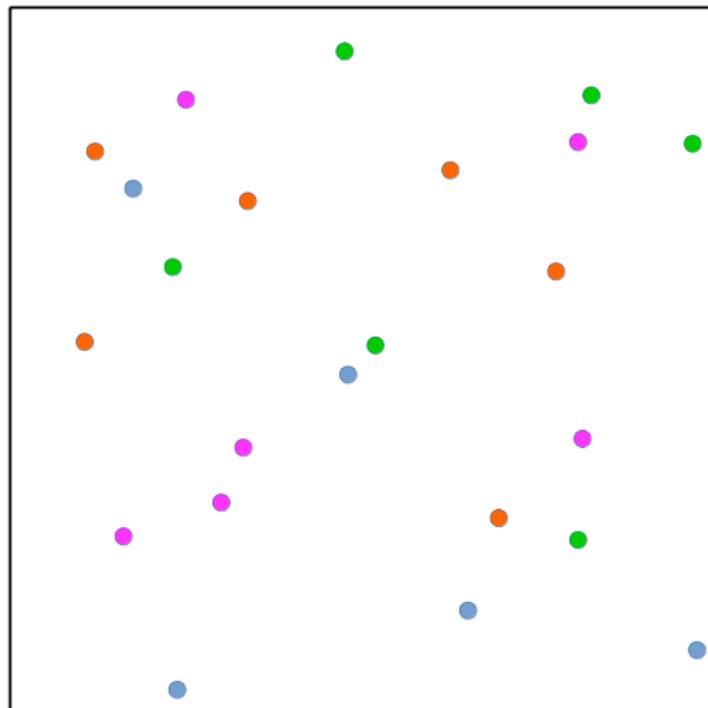


Hadoop - Partition



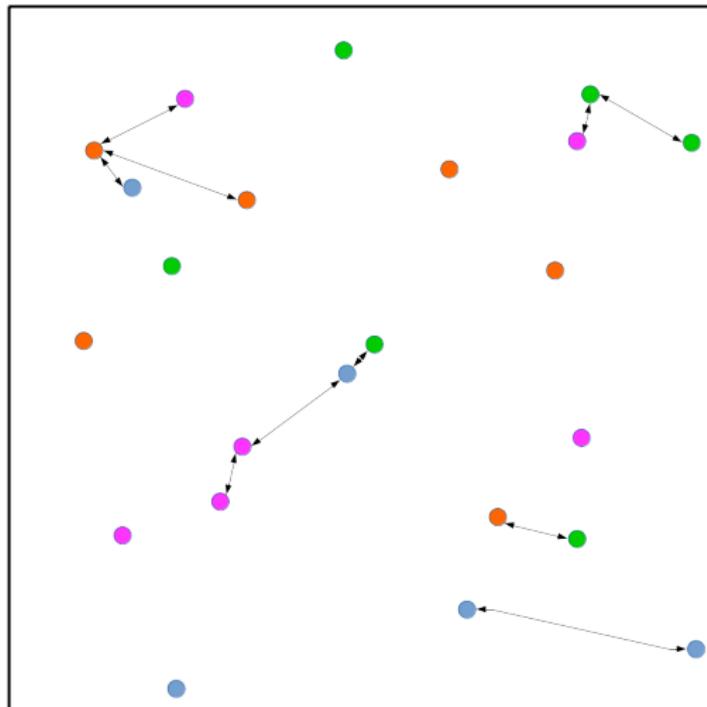
Partition Local Global

Hadoop - Partition



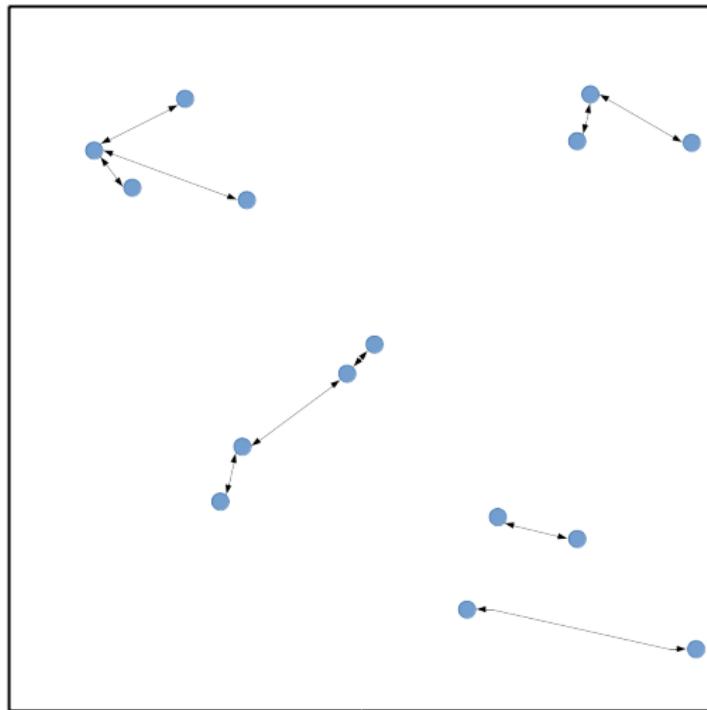
Partition Local Global

Hadoop - Local



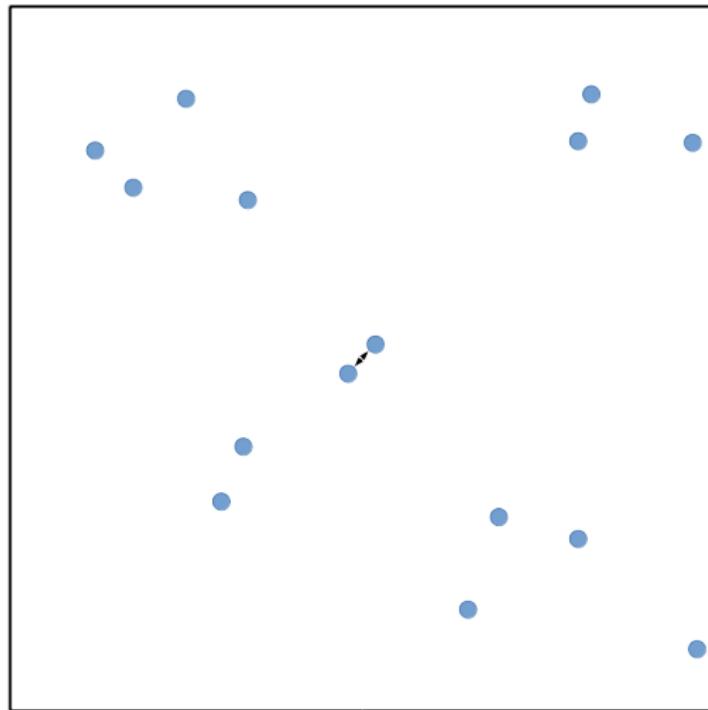
Partition **Local** Global

Hadoop - Global



Partition Local **Global**

Hadoop - Global



Partition Local **Global**

Closest pair in SpatialHadoop

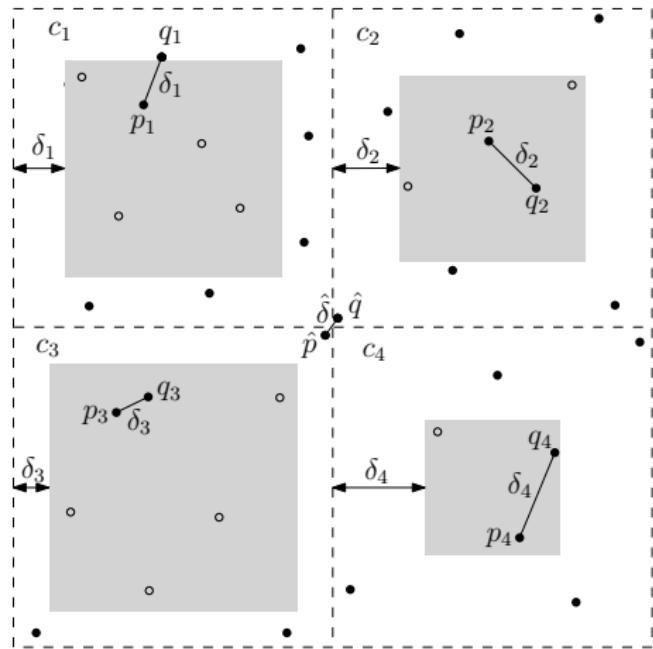
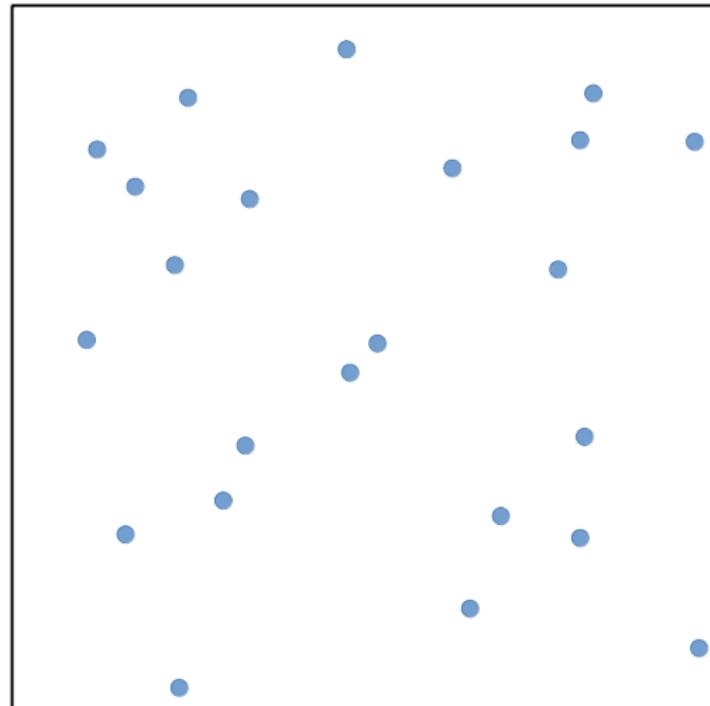


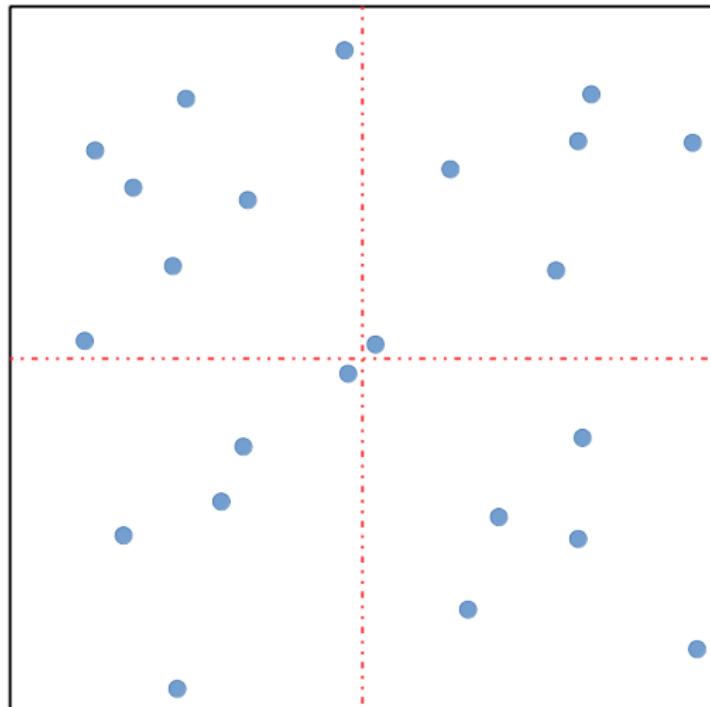
Figure 7: Closest Pair in SpatialHadoop

SpatialHadoop - Partition



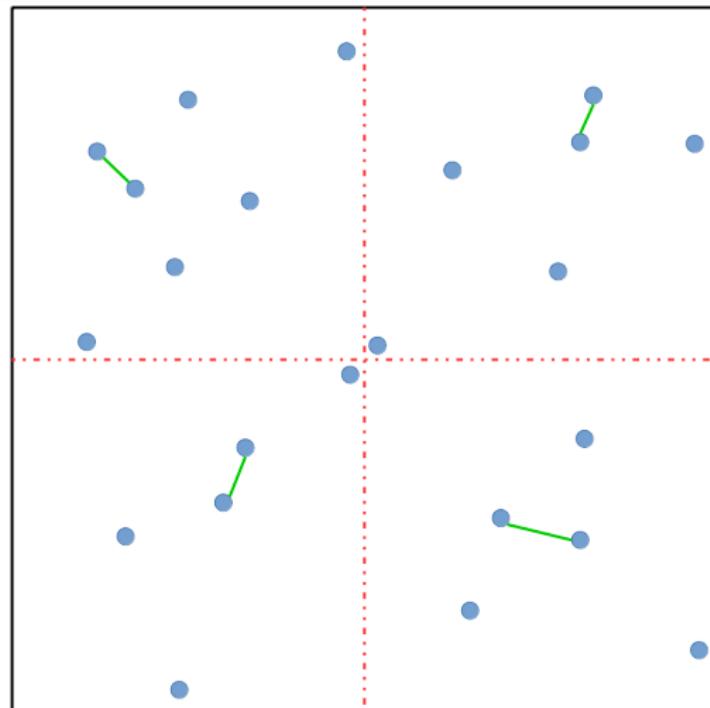
Partition Local Pruning Global

SpatialHadoop - Partition



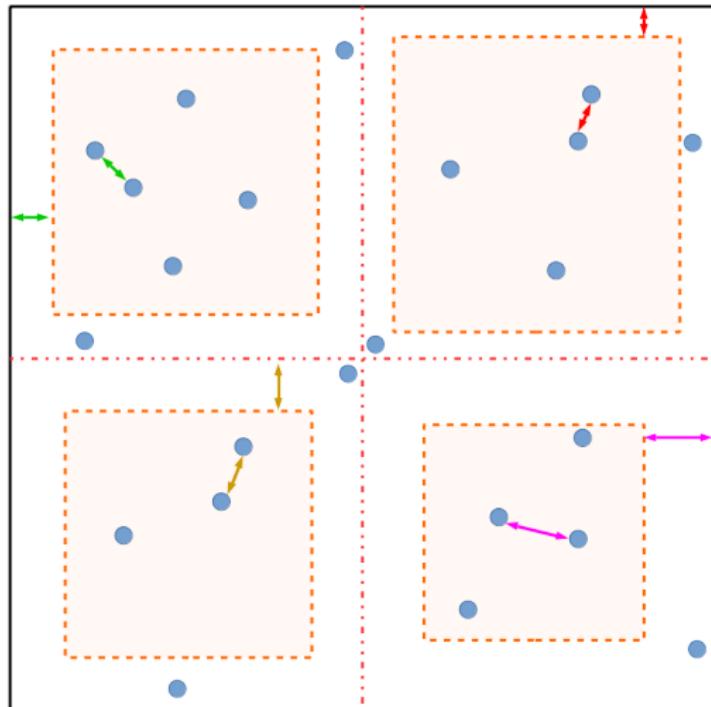
Partition Local Pruning Global

SpatialHadoop - Local



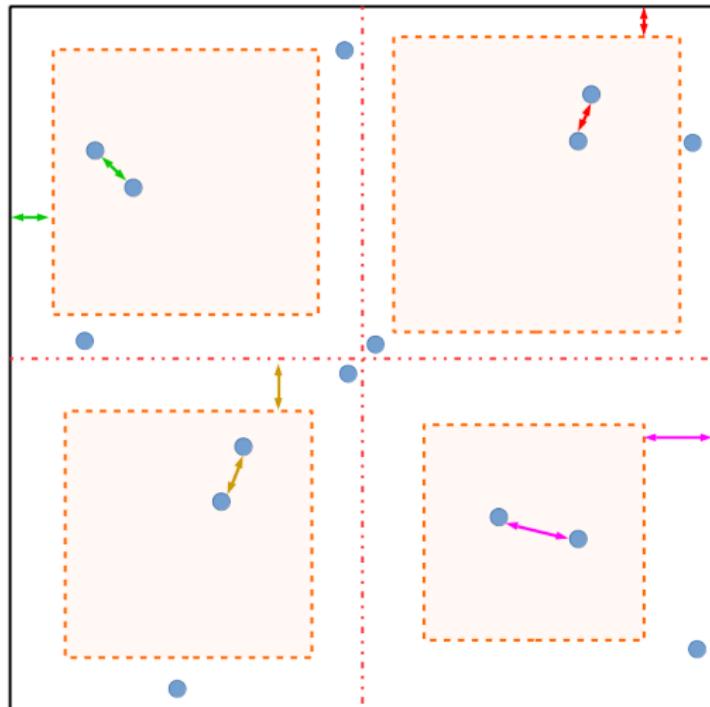
Partition **Local** Pruning Global

SpatialHadoop - Local



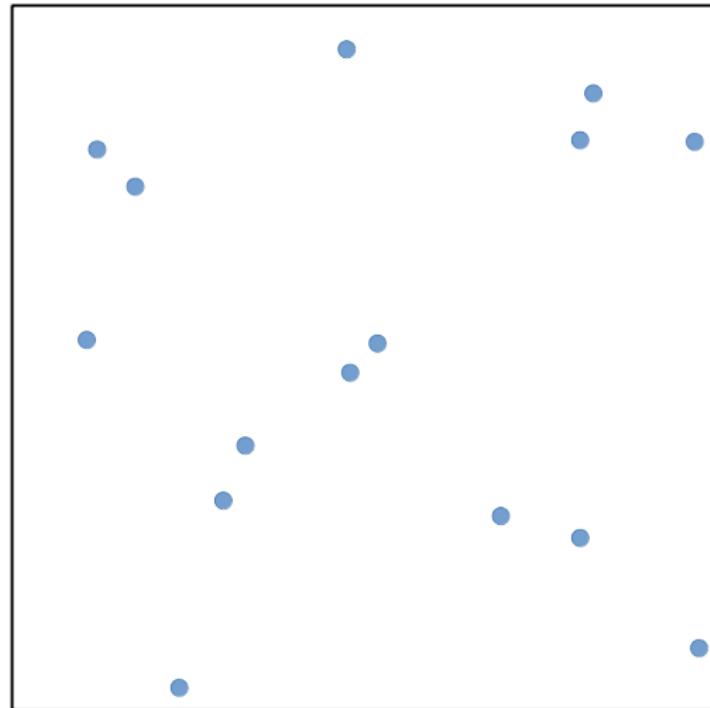
Partition **Local** Pruning Global

SpatialHadoop - Pruning



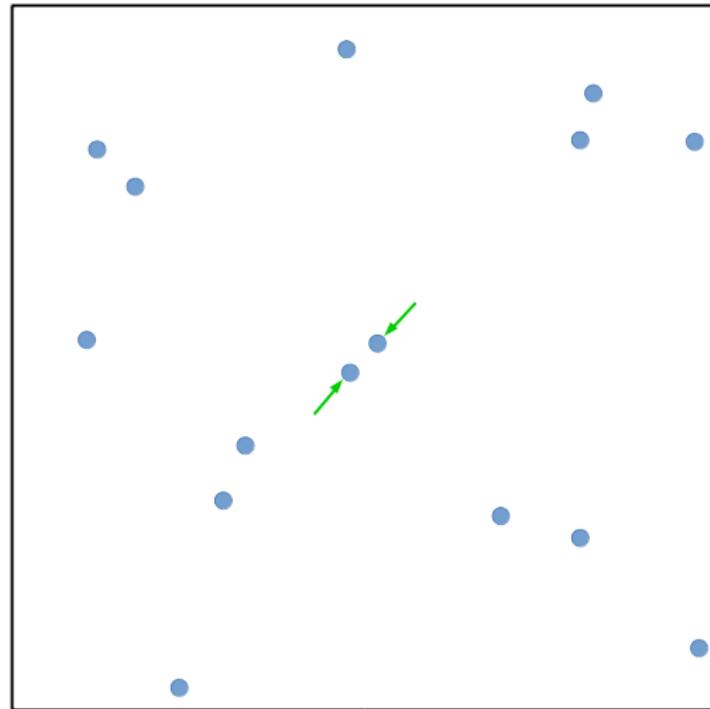
Partition Local **Pruning** Global

SpatialHadoop - Global



Partition Local Pruning **Global**

SpatialHadoop - Global



Partition Local Pruning **Global**

Agenda

1 Background

2 Computational Geometry Operations

- Union
- Skyline
- Convex Hull
- Farthest Pair
- Closest Pair

3 Experiments

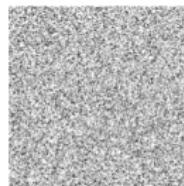
4 Conclusions

Setup

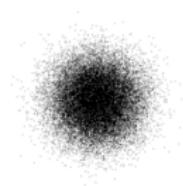
- Cluster of 25 nodes:
 - HDD from 50GB to 200GB.
 - RAM from 2GB to 8GB.
 - Processors 2.2GHz to 3GHz
- Single machine:
 - HDD 2TB.
 - RAM 16GB.
 - Processor 3.4GHz.

Datasets

- Real datasets (from OpenStreetMap):
 - OSM1: 164M polygons, 80GB.
 - OSM2: 1.7B points, 52GB.
- Synthetic dataset:
 - SYNTH: 3.8B points, 128GB.
 - Five different distributions.



(a) Uniform



(b) Gaussian



(c) Correlated

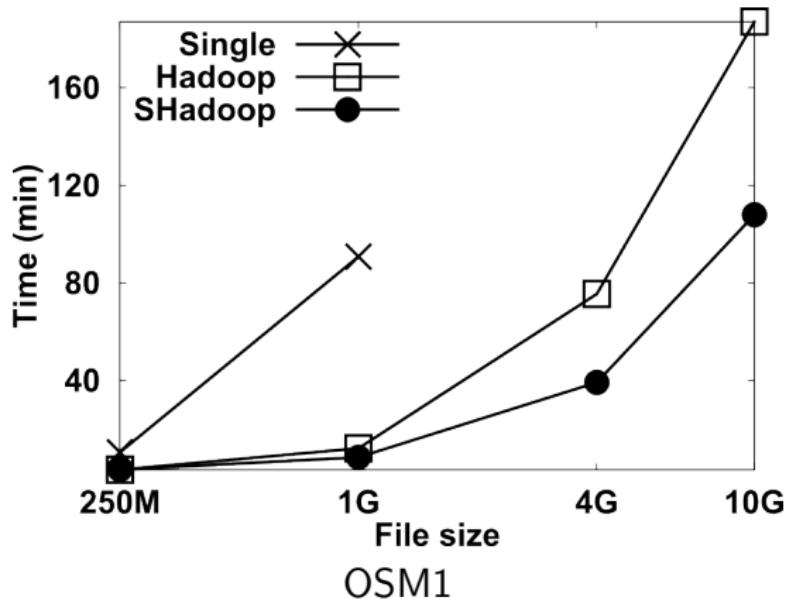


(d) Anti-correlated

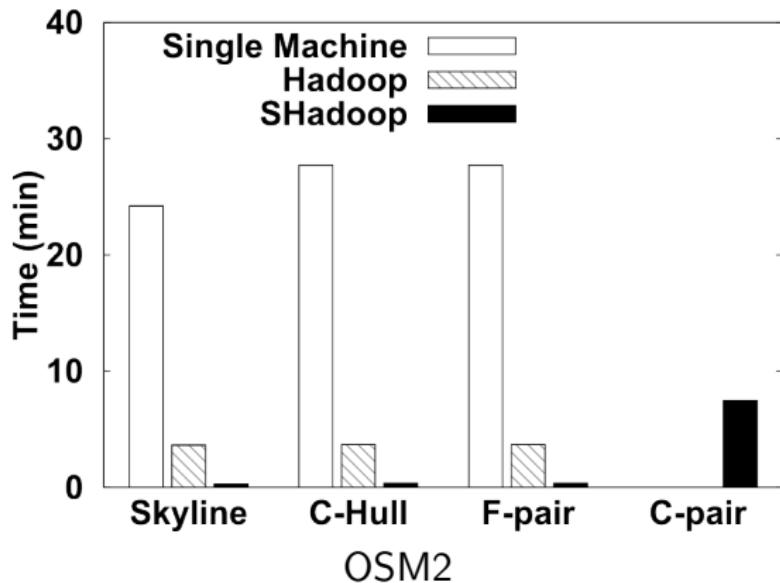


(e) Circular

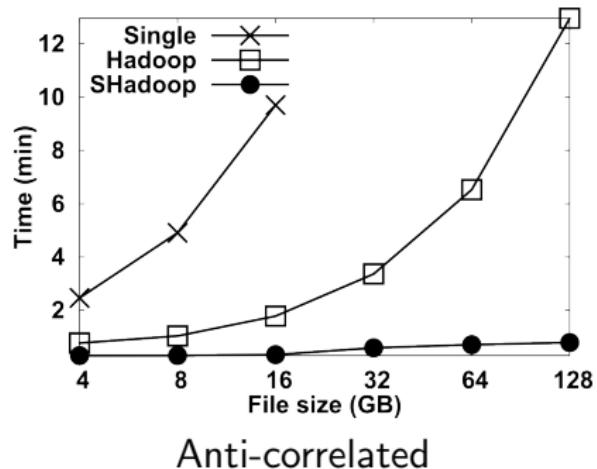
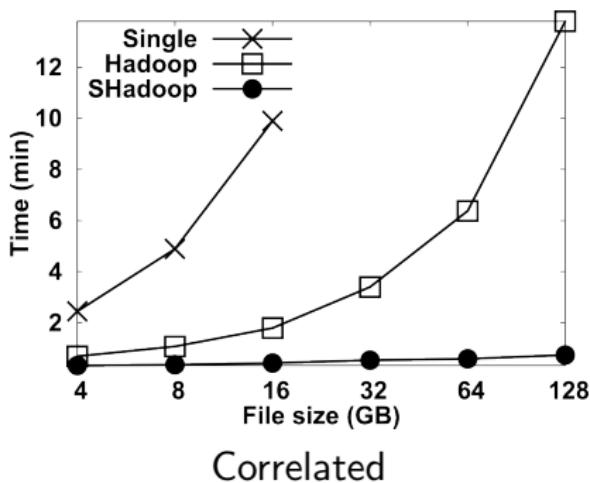
Union



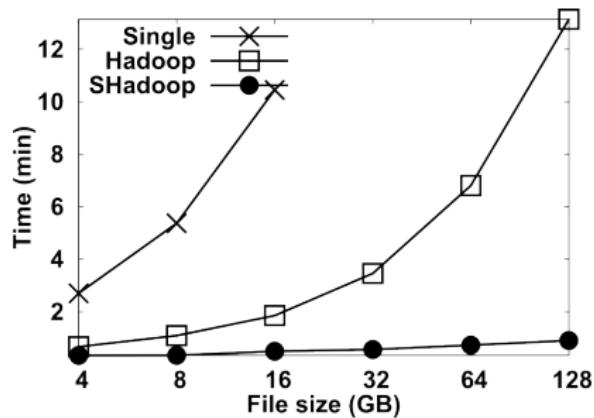
Others



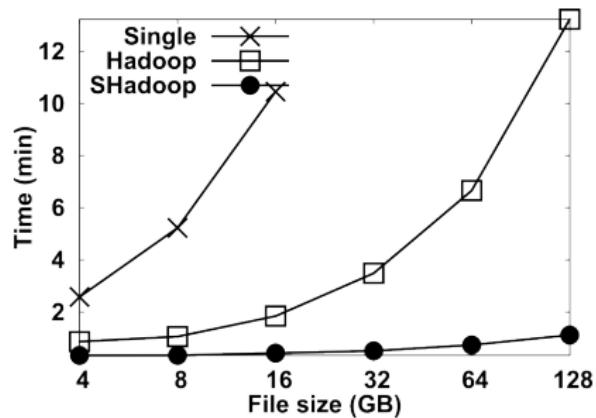
Skyline



Convex hull

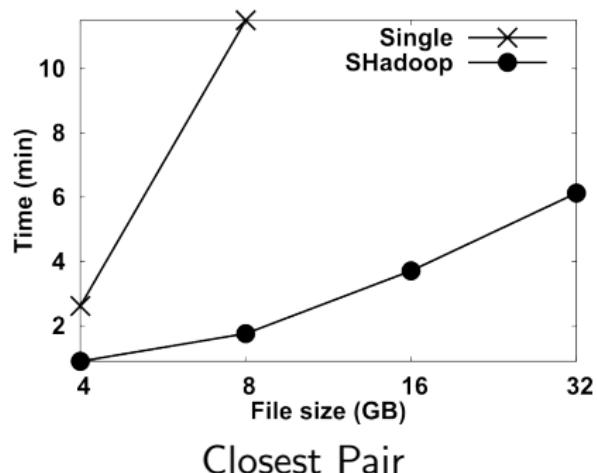
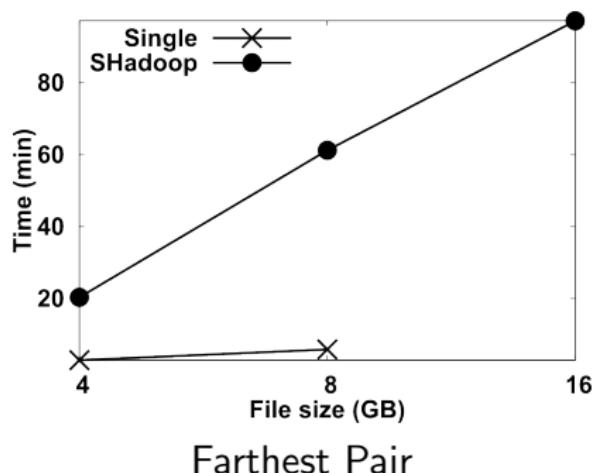


Uniform



Gaussian

Closest/Farthest pair



Agenda

1 Background

2 Computational Geometry Operations

- Union
- Skyline
- Convex Hull
- Farthest Pair
- Closest Pair

3 Experiments

4 Conclusions

Conclusions

- This paper introduced CG_Hadoop as a scalable and efficient MapReduce library.
- Focused on 5 fundamental computational geometry problems...
 - Polygon union, Skyline, Convex hull, Farthest and Closest Pairs.
- Provided versions for Apache Hadoop and SpatialHadoop systems.
- Distributed approach speed up performance.
- Spatial partitioning allows early pruning which make it even more efficient.
- Achieve up to 29x and 260x better performance.

Future ideas

- Working on more complex operations, for example motion patterns.
- Explore ports to new distributed platforms such as Spark or Simba.

Thank you!!!

Do you have any question?