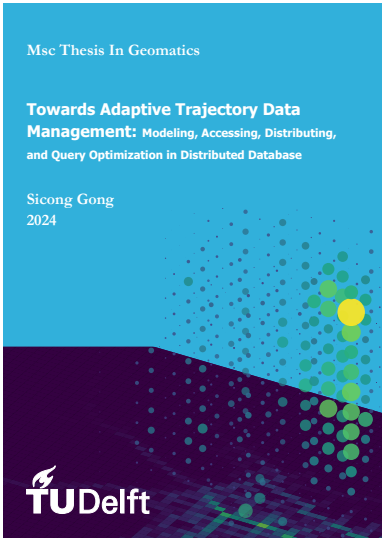
Towards Adaptive Trajectory Data Management:
Modelling, Accessing, Distributing, and Query Optimization in Distributed Database

Sicong, Gong (5711932)

1st Supervisor: Dr.ir. Martijn Meijers
2nd Supervisor: Drs. Wilko Quak
Co-reader: Dr. Ken Arroyo Ohori

June 17, 2024

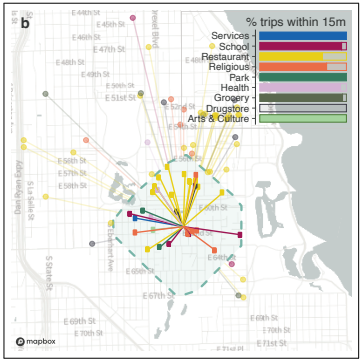
Table of Contents



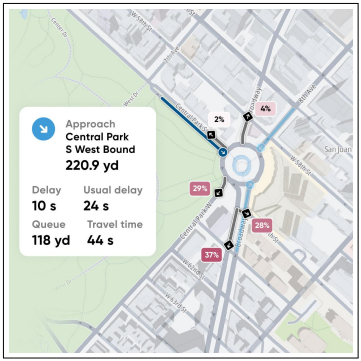
Introduction

Trajectory Applications

- **Urban planning → 15-minute city quantification.**
- **Traffic management → Traffic condition identification.**



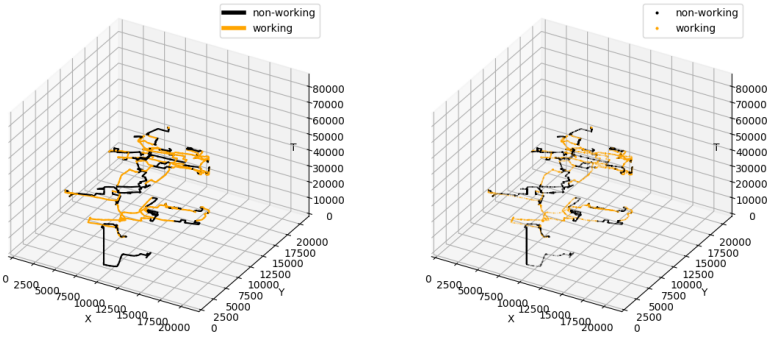
(a) 15-minute City Quantification, adopted from Abbiasov et al. (2024)



(b) Traffic Condition Identification, adopted from TomTom

Trajectory Definition

- **Humans' understanding → Continuous sequences.**
- **Data records acquired → Discrete points.**



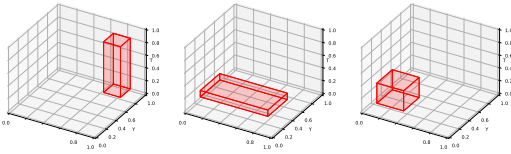
(a) Continuous Sequences

(b) Discrete Points

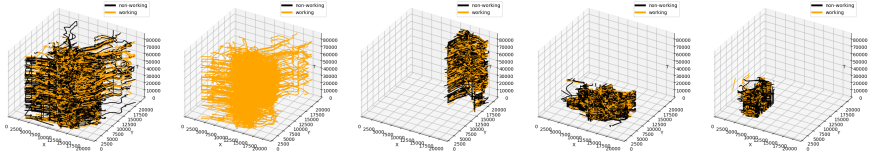
Trajectory Operations

TU Delft

- **Selection by ID + Selection by range.**



(a) t-4-c (Needle Used) (b) t-4-d (Slice Used) (c) t-4-e (Dice Used)

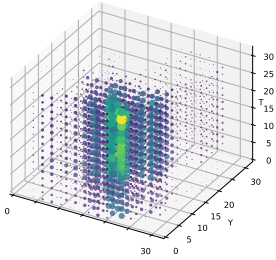


(d) t-4-a (ID Selection) (e) t-4-b (State Selection) (f) t-4-c (Needle Selection) (g) t-4-d (Slice Selection) (h) t-4-e (Dice Selection)

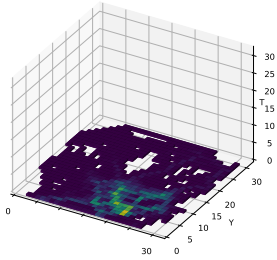
Trajectory Operations (Continued)

TU Delft

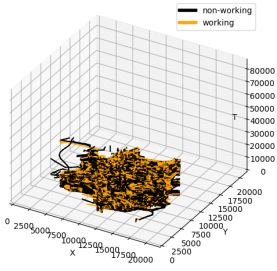
- **Aggregation + Projection + Simplification.**



(a) t-6-a (Aggregation)



(b) t-6-b (Projection)

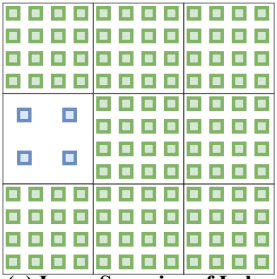


(c) t-6-c (Simplification)

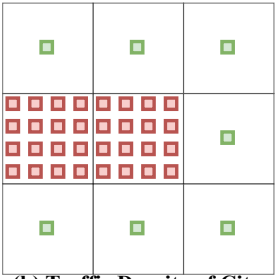
Trajectory Properties

TUDelft

- High frequency: Huge volume.
- High cardinality: Numerous entries.
- High dimensionality: Integration of space, time and semantics.
- High heterogeneity: Uneven distribution.



(a) Laser Scanning of Lake and Grass

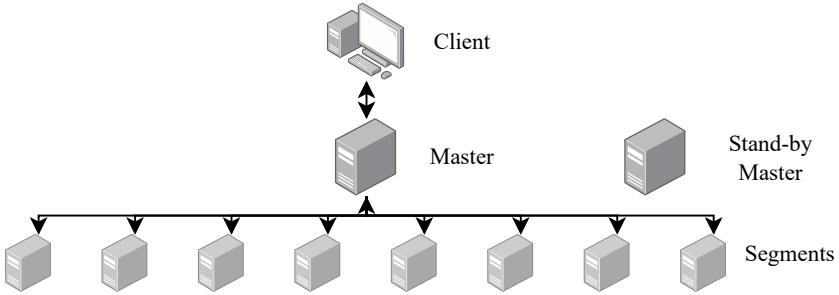


(b) Traffic Density of City and Rural Area

Distributed DBMS

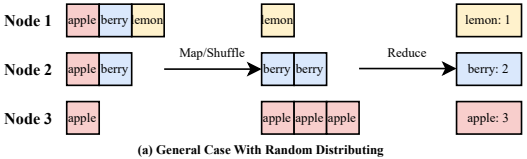
TUDelft

- **Scalability (Speed-up and Scale-up) and Localization (Cluster data and localize computations)**

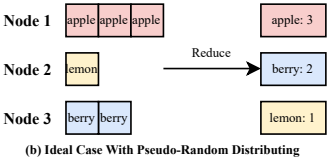


- **Experiment setting: 5 virtual machines (1 master, 4 nodes), each node has 2 segments.**

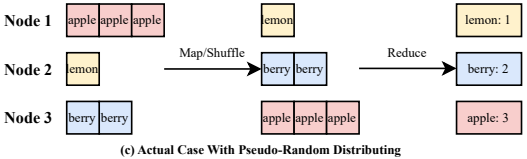
Map-Reduce Computation Model



(a) General Case With Random Distributing



(b) Ideal Case With Pseudo-Random Distributing



(c) Actual Case With Pseudo-Random Distributing

- **MapReduce partitions data into key-value pairs, processes them in parallel mappings and then aggregates the results through local computation in the reduce step.**

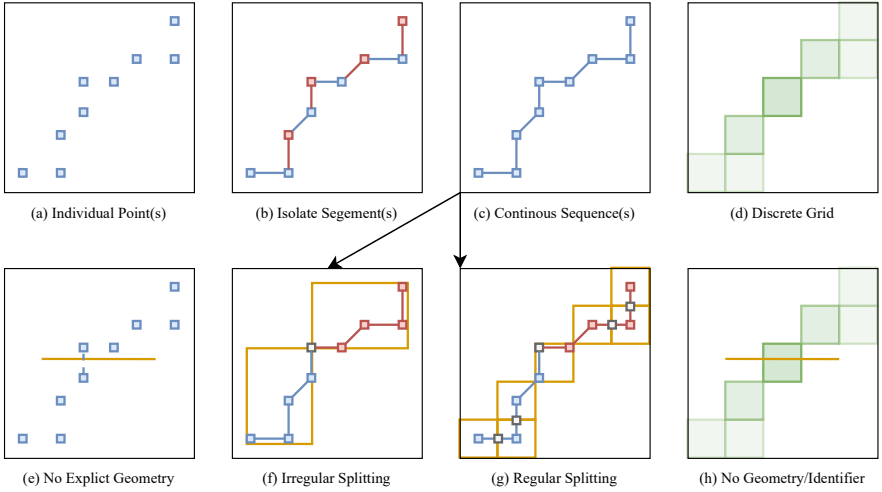
Distributed Products

TU Delft

Feature	Traditional Database	Hadoop	Spark	MPP Database
Volume	GB-TB	PB-EB	TB-PB	TB-PB
Robustness	High	High	High	Medium
Scalability	Low	High	High	High
Latency	Medium	High	Low	Very Low
Throughput	Low	High	High	Medium
Data Type	Structured	All	All	Structured

Q1: Modelling

Modelling Options

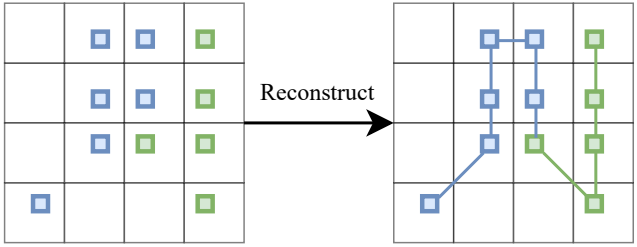


Reasons for Sequence-based Model

TU Delft

The sequence is better!

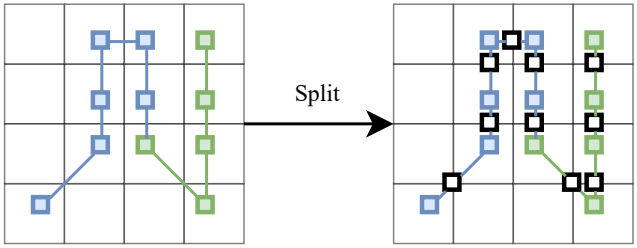
- **More supported operations.**
- **Smaller entries cardinality.**
- **Higher compression potential.**



(a) Individual Points (b) Entire Trajectory

Solutions for Unstructured Nature

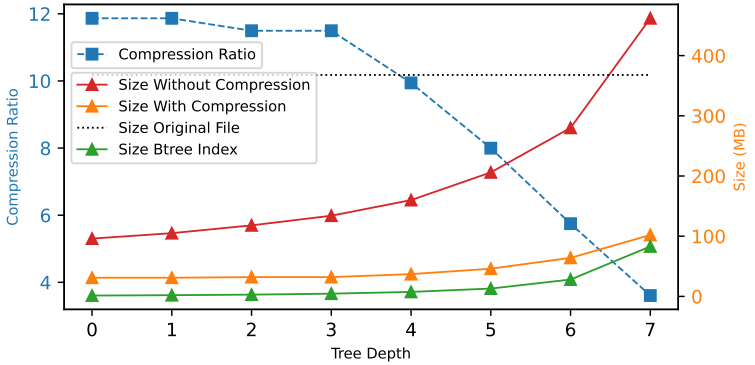
- Split by semantics.
- Split by spatio-temporal cube.



(b) Entire Trajectory (c) Splitted Sequences

Does the Above Methods Work? - Compression

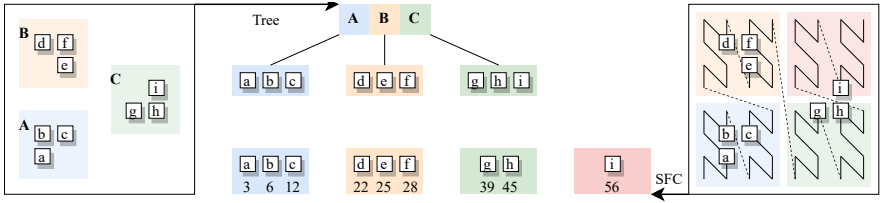
- Further subdivision of the space would decrease the compression ratio.



Q2: Accessing

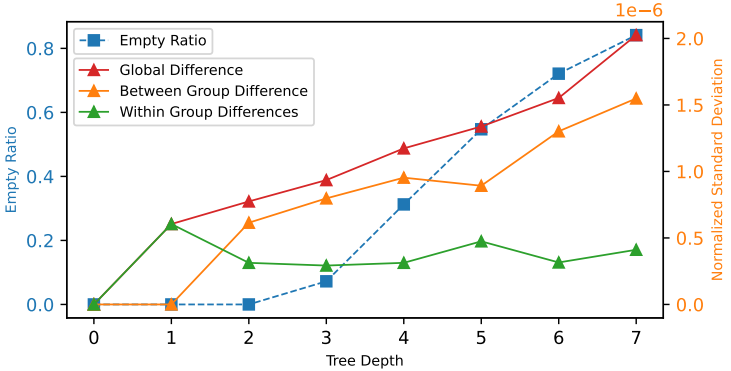
Accessing Methods

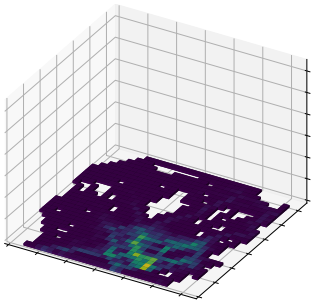
- R-tree → Adaptive but complex and costs more storage.
- Space filling curve → Rigid but simple and corresponds to the nature of modelling.



Uneven Distribution

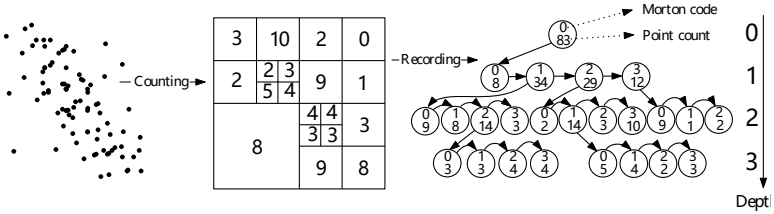
- Further subdivision of the space would increase the empty ratio.
- Further subdivision of the space would increase the global difference (unevenness).





HistSFC Solution

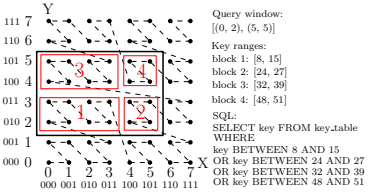
- **Histogram Tree: Adaptive octree.**
- **Space-filling-curve (Morton): Represent a record and use b-tree for indexing.**



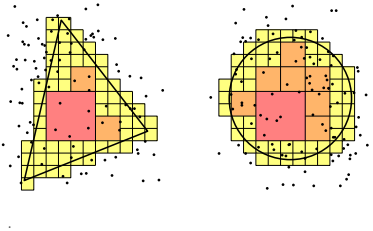
Adopted from Liu, 2022

Shape Querying

- **Recursively partitioning the extent of data according to SFC regions to match different query geometries, for selecting data in the table.**



(a) Executing a window query on a uniformly distributed 2D point set based on Morton encoding



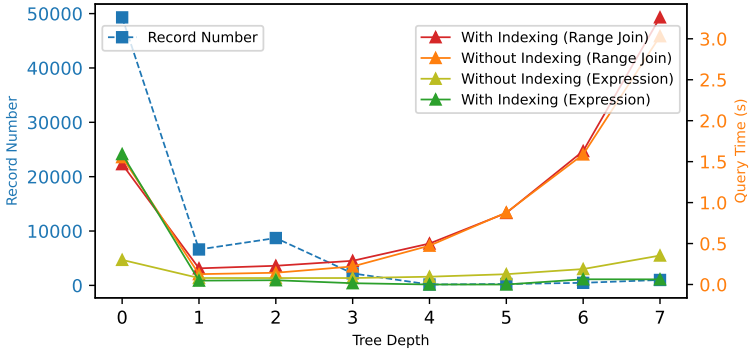
(b) Querying with a triangle and a circle: false positive points in boundary cells will be filtered out by a second filter

Adopted from Liu, 2022

Does the Above Methods Work? - Selection

TU Delft

- Selection performance is first increasing then decreasing with the further subdivision of the space.

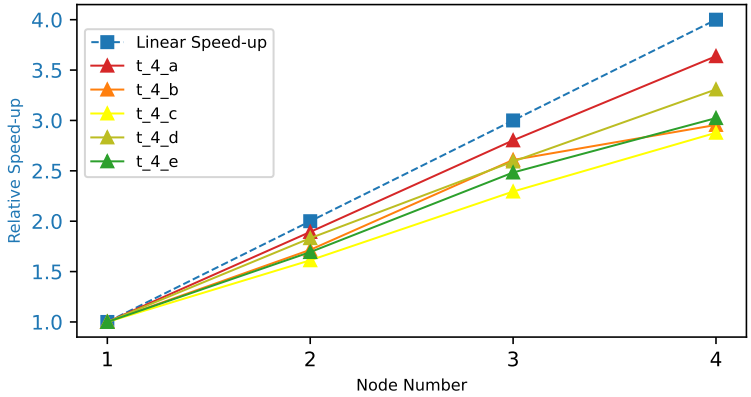


Q3: Distributing

Does the Distributed DBMS Work? - Speed-up

TU Delft

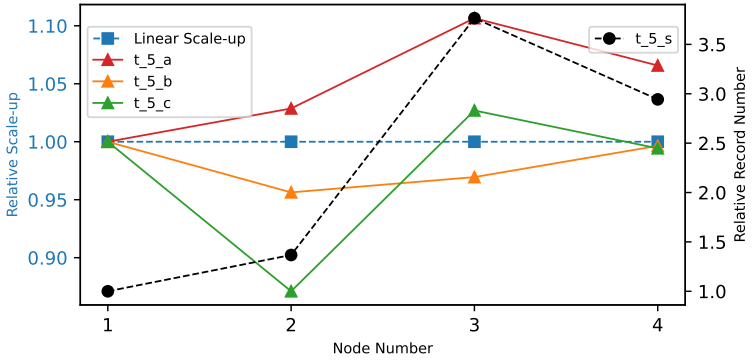
- **Five operations are designed to test the speed-up (same problem sizes with increasing resources) and the result is positive.**



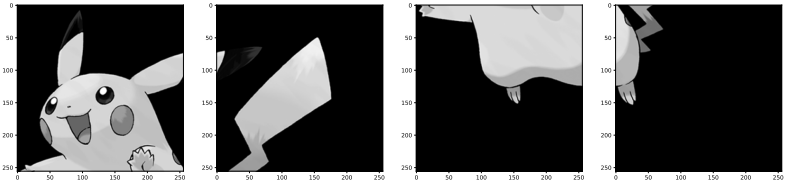
Does the Distributed DBMS Work? - Scale-up

TU Delft

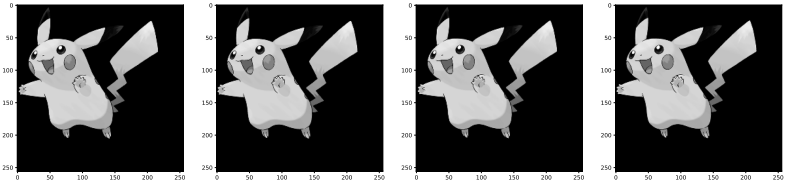
- Three operations are designed to test the scale-up (increasing problem sizes with increasing resources) and the result is positive.



Distributing Strategy



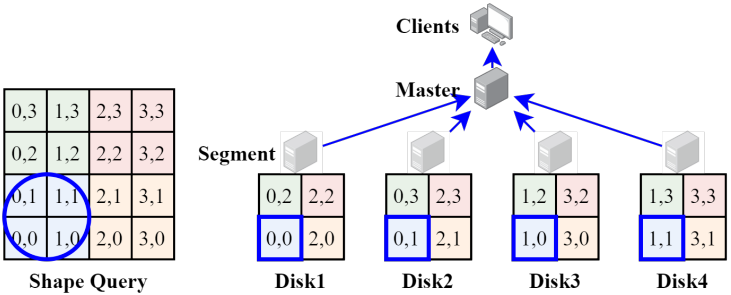
Block-based: Fold twice and split.



Sample-based: e.g. all the lower left pixels of 4 neighbours as a group.

Pseudo-Sampling Distributing

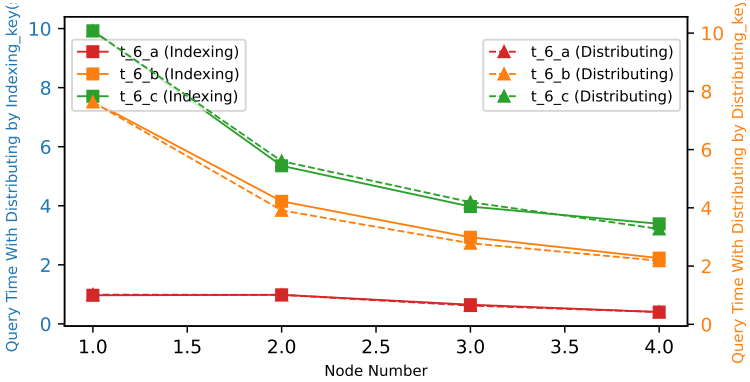
- Load-balancing: Block-based method leads to uneven data distribution.
- Localization: Random sampling leads to no locality being preserved.



Does the Above Methods Work? - Localization

TU Delft

- **Three operations (Aggregation, Projection and Simplification) are designed to test the localization but the result is negative.**



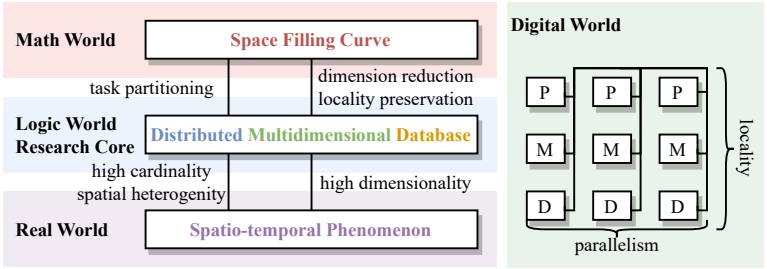
Conclusion

Conclusion

TU Delft

In conclusion, the above methods alleviate the difficulties mentioned.

- **Modelling: Reduce cardinality.**
- **Accessing: Reduce dimensionality and alleviate uneven distribution.**
- **Distributing: Use parallelism to speed up and scale up.**



Reflection

TU Delft

- **The main lesson learnt from this thesis is the need to adapt the data properties and platform features.**

$$\text{Distribution Awareness} \left(\begin{array}{l} \text{adaptive modelling (splitting)} \; + \\ \text{adaptive accessing (indexing)} \; + \\ \text{adaptive distributing (partitioning)} \; + \\ \text{adaptive querying (merging)} \end{array} \right) \times \text{Distributed Architecture} \quad (1)$$

Future Work

TU Delft

However, there are still some limitations that should be done in future work.

- **Realistic benchmarking: Not only in virtual machines.**
- **Workflow optimization: Adaptive splitting and range merging.**
- **Mathematical proofing: Not only by experiments.**

Thank You!

Reference

Liu, H. (2022). nD-PointCloud Data Management: continuous levels, adaptive histograms, and diverse query geometries. A+ BE| Architecture and the Built Environment, (12), 1-206.