Manage 4D historical AIS data by Space Filling Curve

Jinglan Li

Mentor #1: Martijn Meijers
Mentor #2: Haicheng Liu
Mentor #3: Ken Arroyo Ohori

TU Delft

Outline

- Introduction
- Related works
- Methodology
- Results & discussion
- Conclusions & future works

Outline

- Introduction
- Related works
- Methodology
- Results & discussion
- Conclusions & future works

Background

- Automatic identification system (AIS)

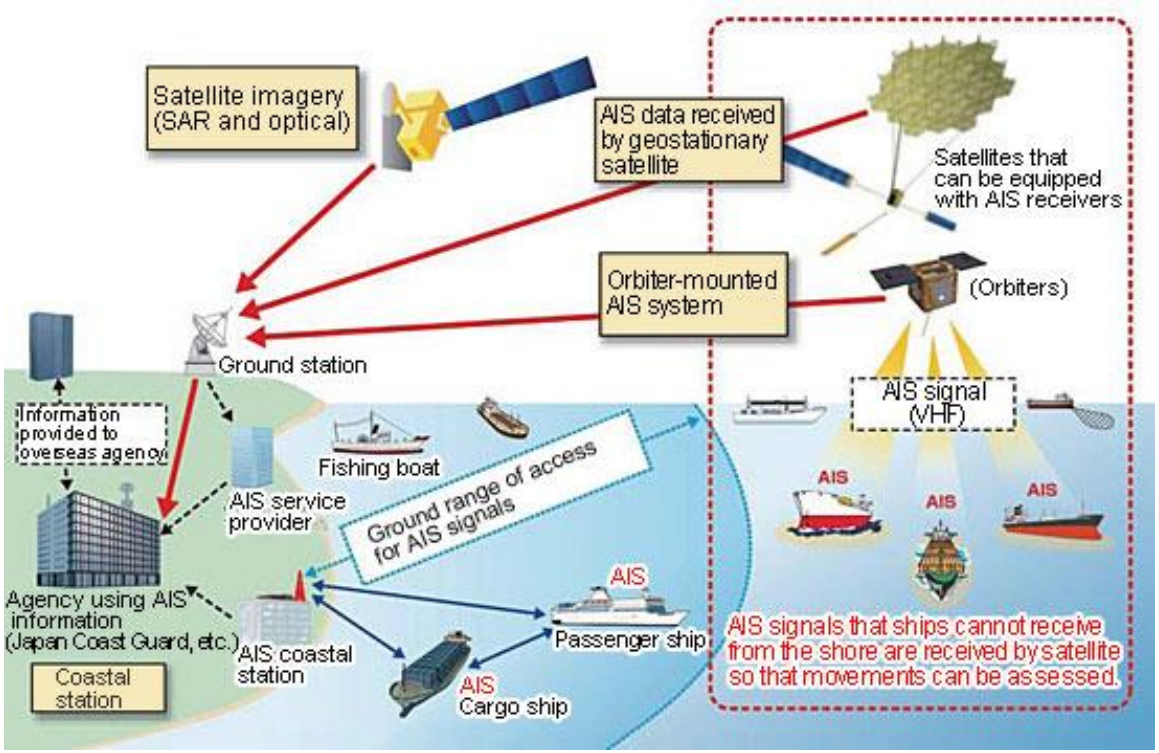


Figure 1 : Automatic identification system (AIS). The Source: https://www.cruiseshipportal.com/news/cruise-news-maritime-news/automatic-identification-systemsais-technology-explained/

Background

- Automatic identification system (AIS)
- AIS data
  - AIS data is encoded

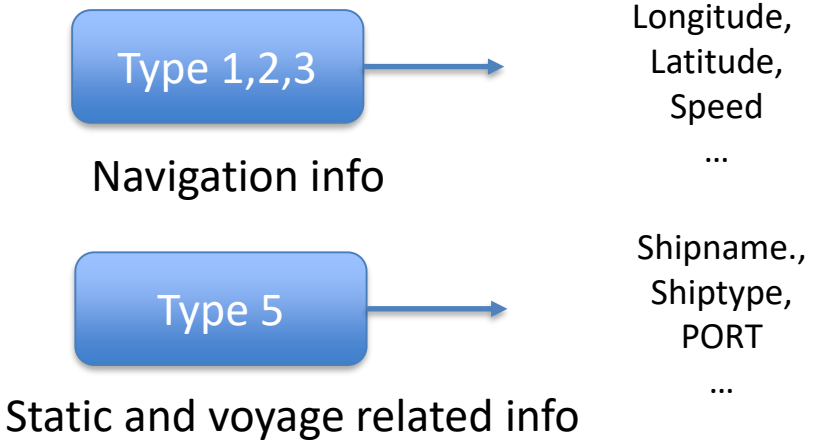
```
!AIVDM,1,1,,A,13aENUhP00PBwE2Mfg3PPgw`P@00,0*2E,2016-12-10 00:00:00.056,/10.0.201.3
!AIVDM,1,1,,B,1CAO2j7Oic0G<d8NiSgnvmrJ2H7l,0*1B,2016-12-10 00:00:00.084,/77.163.66.94
!AIVDM,1,1,,A,13aGs:PP00PCQwlMEB8=:gv00802,0*4C,2016-12-10 00:00:00.113,/81.243.244.33
!AIVDM,1,1,,B,133v1kPP00PD9I<MDRWD5Own2@02,0*3B,2016-12-10 00:00:00.130,/81.243.244.33
```

Figure 2 : the raw AIS data (in NMEA format)

Background

- Automatic identification system (AIS)
- AIS data
  - AIS data is encoded
  - AIS data has lots of types

There are 24 types of AIS information currently used:



Motivation

- The application of real-time AIS data
  - Detect anomalies motions of vessels
  - Make the smart port
  - Predict the trajectories of vessels
  - …
- The application of historical AIS data
  - Predict next position of vessels
  - Analyze navigation saturation
  - Forecast the development of the maritime industry
  - …

Motivation

- The application of real-time AIS data
- The application of historical AIS data
- The lack of efficiently managing AIS data
  - No existing benchmark database in maritime research area
  - 3D (longitude, latitude, time) data are mainly focused on

Research questions

- Main research question

How to efficiently manage 4D data (Longitude, Latitude, Time, MMSI) of vessels to do the efficient query by using Space Filling Curve in PostgreSQL?

Research questions

- Main research question
- Sub-questions
  - How to manage the 4D data is better to support the efficient query? Dealing with the integrated 4D data or dealing with the integrated 3D data first (3D + 1D) ?
  - How to scale data in each dimension properly to compute the SFC key?
  - Which SFC performs better in 4D data querying? Morton curve or Hilbert curve?
  - How about the BRIN index? From which aspects can I compare the indexing method?

Outline

- Introduction
- Related works
- Methodology
- Results & discussion
- Conclusions & future works

Related works

- Management of moving objects
- Managing and organizing data in database
- One-dimensional indexing technology

Management of moving objects

- The real-time positions of moving objects
  - Objects Spatio-temporal (MOST) data model
  - Relative space-based GIS data model

- The historical positions of moving objects
  - Model moving objects as abstract data type which could be integrated as attribute data types in database
  - Manage data using bit map indices

Related works

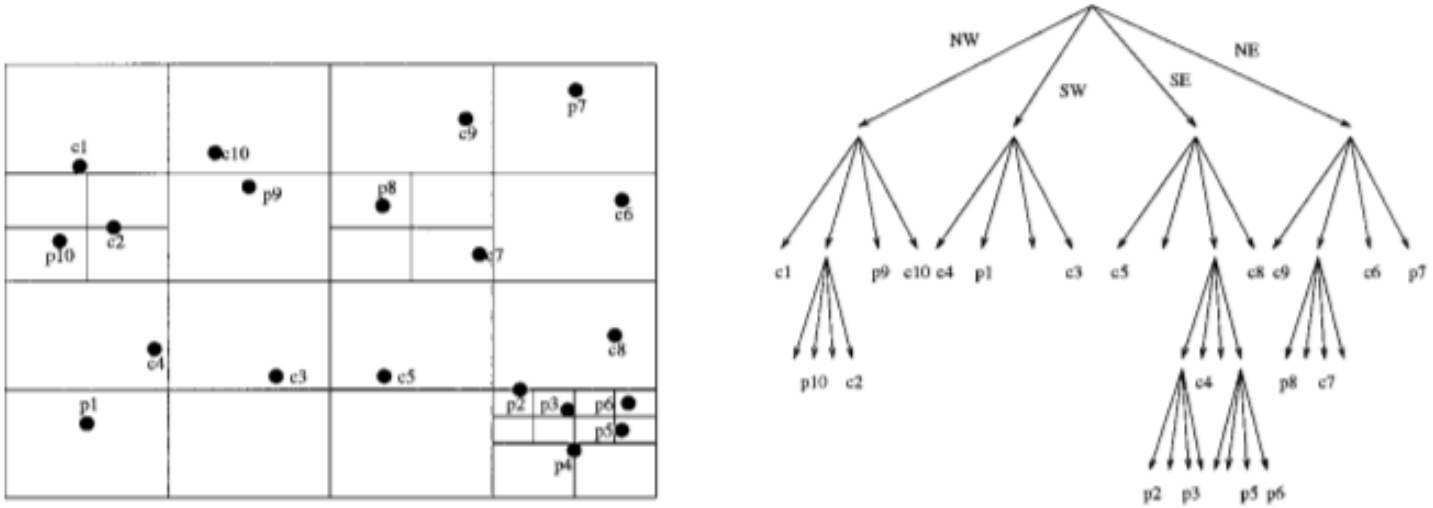
- Management of moving objects
- Managing and organizing data in database
- One-dimensional indexing technology

Managing and organizing data in database

- Lots of data structure support storage and retrieval of points in a multidimensional space
  - R-Tree
  - KD-Tree
  - Quadtree
  - Space Filling Curve

Quadtree

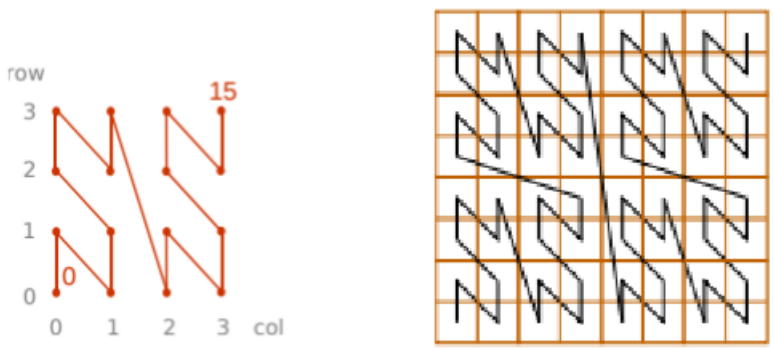
- Quadtrees are most often used to partition a two dimensional space by recursively subdividing it into four quadrants or regions.



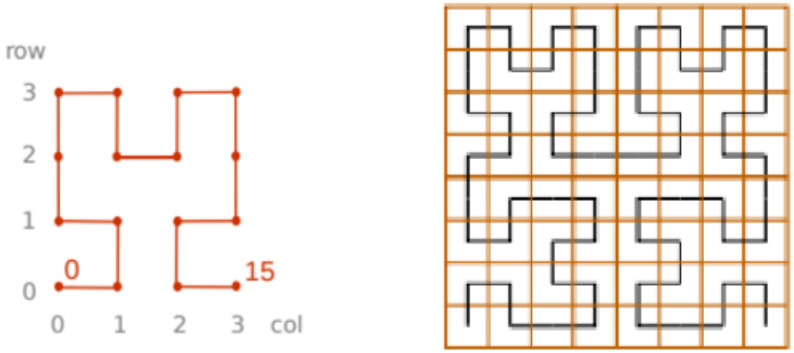
(Figure 3 : Region quadtree)

Space Filling curve (SFC)

- A SFC is a thread that goes through all the points in the space while visiting each point only one time.
- Morton curve, Hilbert curve…



(Figure 4 : Morton curve)



(Figure 5 : Hilbert curve)

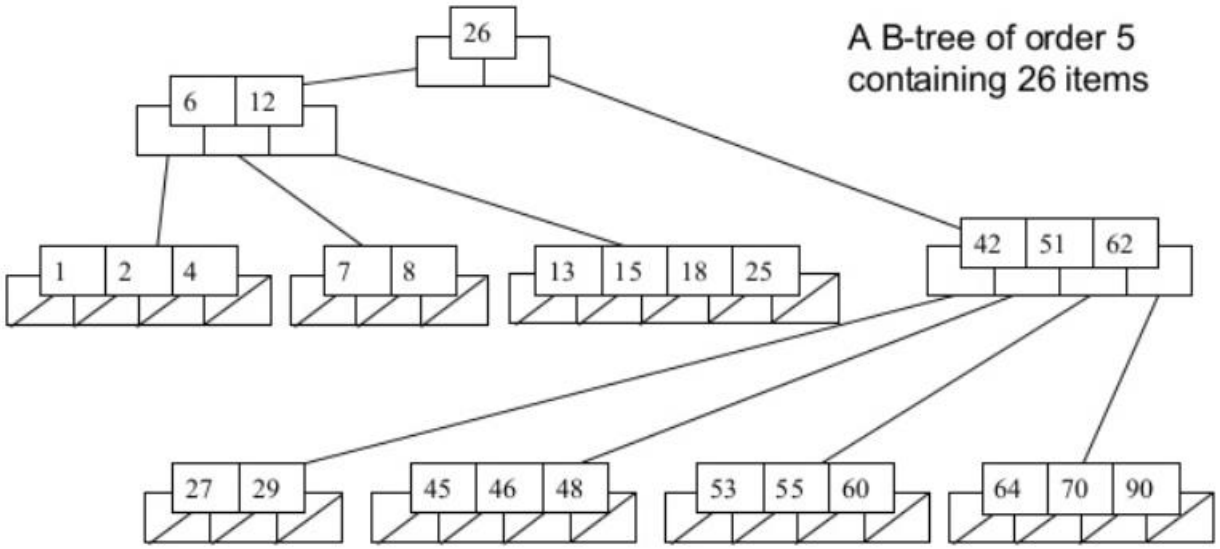
Related works

- Management of moving objects
- Managing and organizing data in database
- One-dimensional indexing technology

One-dimensional index

- B-Tree index

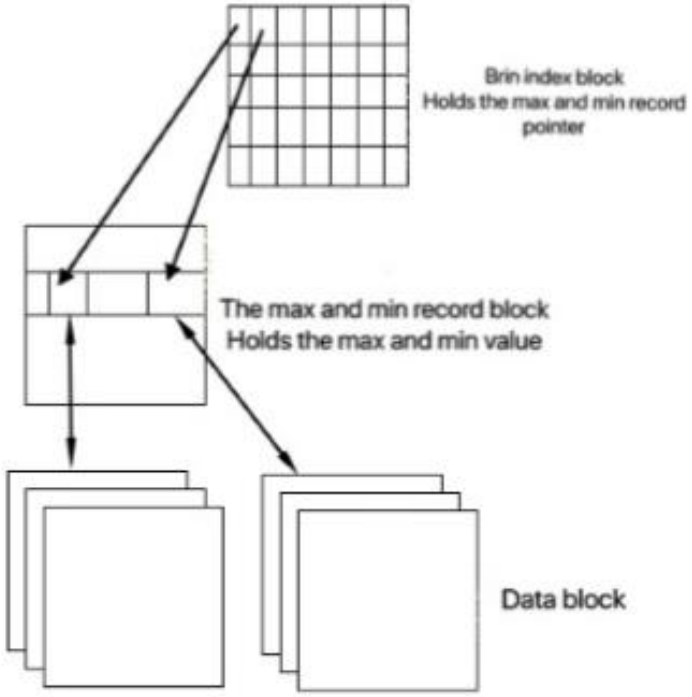
  It is the binary search tree and can store more than two keys per node and it is always balanced after insertion, deletion or updation.



(Figure 6 : B-Tree index)

One-dimensional index

- B-Tree index
- BRIN index
  - BRIN is the Block Range Indexes



(Figure 7 : BRIN index)

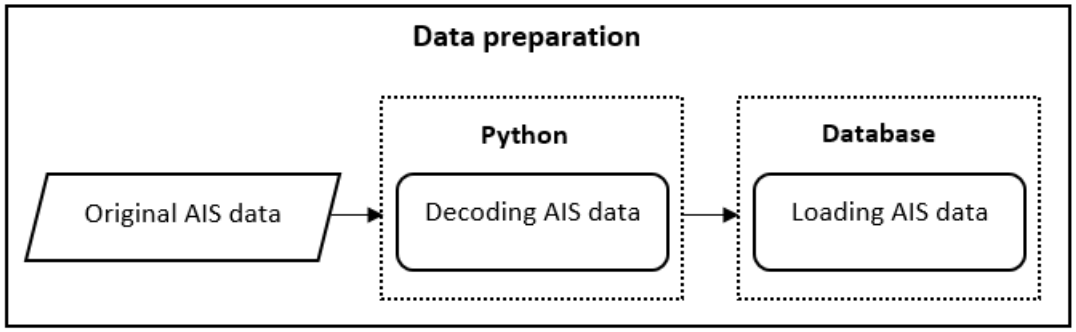
Outline

- Introduction
- Related works
- Methodology
- Results & discussion
- Conclusions & future works

Pipeline

Data preparation

- Decoding AIS data
  - Regarding with certain rules
- Loading AIS data
  - Load dynamic data (navigational info) into database
  - Extract Longitude, latitude, time, MMSI



(Figure 7 : overflow of data preparation)

Clustering and indexing

- The flow chart of clustering and indexing

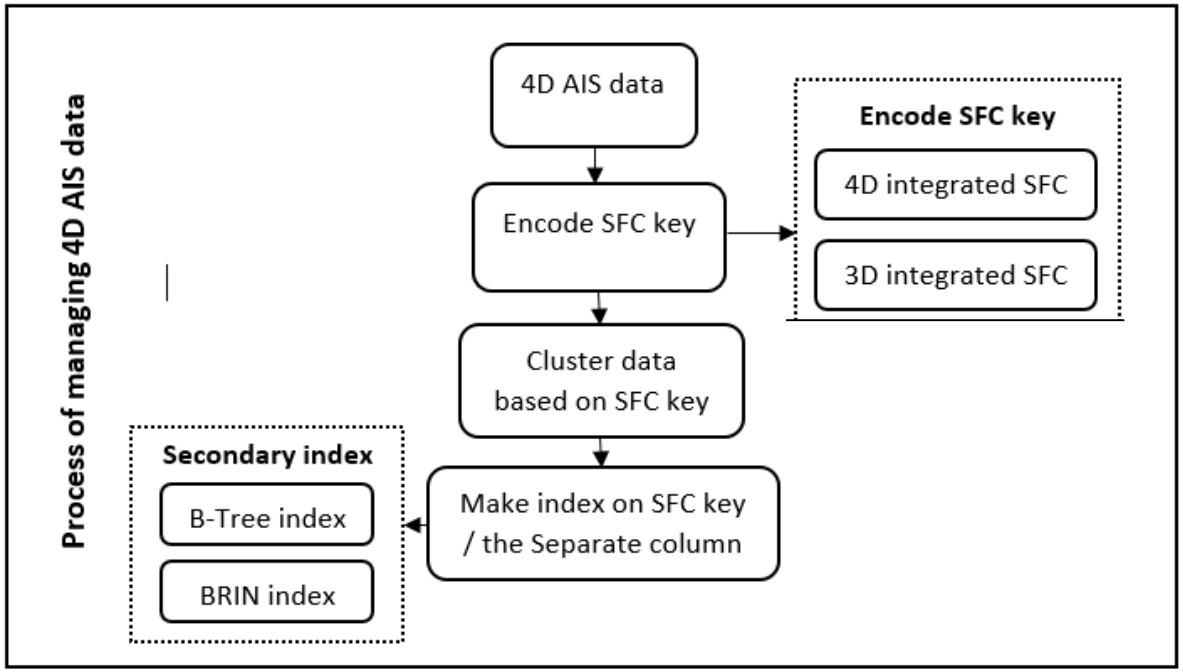
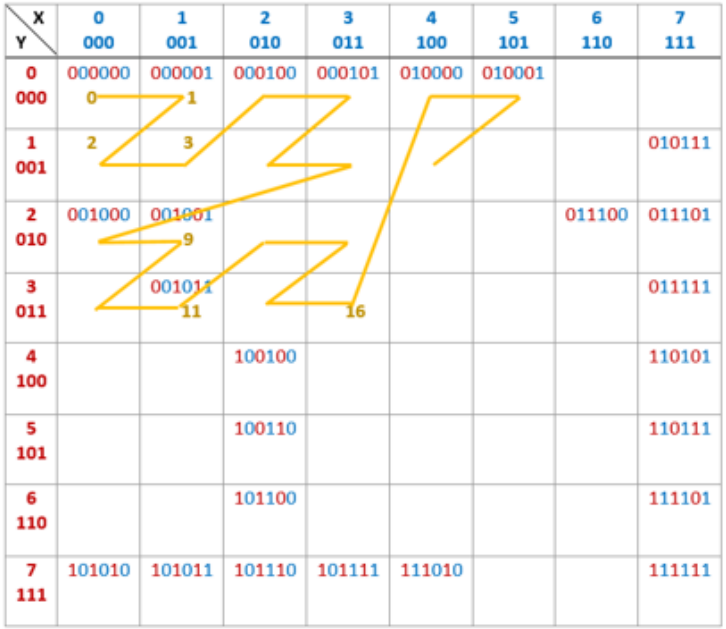


Figure 9 : flow chart of clustering and indexing

Clustering and indexing

- Space filling curve approach
  - Principle of SFC approach



Coordinate: (x, y) = (1, 3)
= (001, 011)
Interleaving the bits: 001011
= 11
The 11^{th} cell on the curve

(Figure 8 : principle of SFC approach (Morton curve))

Clustering and indexing

- Space filling curve approach
  - Principle of SFC approach
  - SFC key

Full resolution key:

the key that stores and retains the complete original information

Partial resolution key:

value zooms out in each dimension when calculating the partial resolution key for efficient storage

Clustering and indexing

- Managing 4D AIS data
  - 4D integrated SFC approach
  - 3D integrated SFC approach

Clustering and indexing

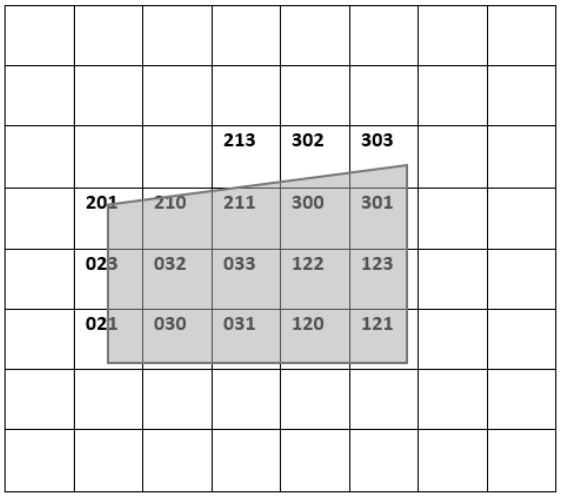
- Managing 4D AIS data
  - 4D integrated SFC approach
    - Scaling 4D data proper to calculate 4D SFC key and only **SFC key** will be left in database
    - Cluster data based on SFC key
    - Make index on SFC key

Clustering and indexing

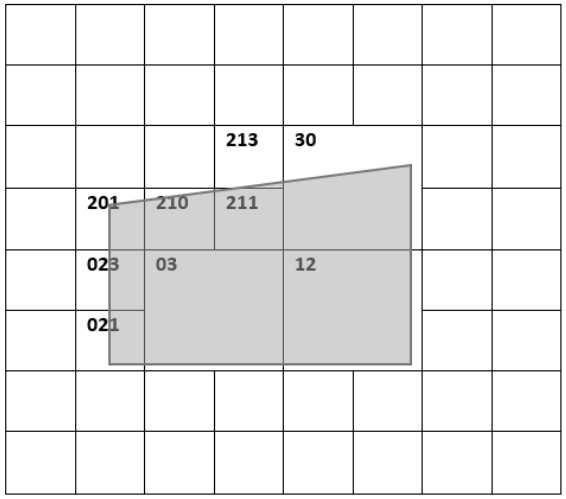
- Managing 4D AIS data
  - 4D integrated SFC approach
  - 3D integrated SFC approach
    - Scaling 3D data (Longitude, latitude, time) proper to calculate 3D SFC key. **SFC key** and **MMSI** will be left in database
    - Cluster data based on SFC key
    - Make index on SFC key and MMSI

Query test

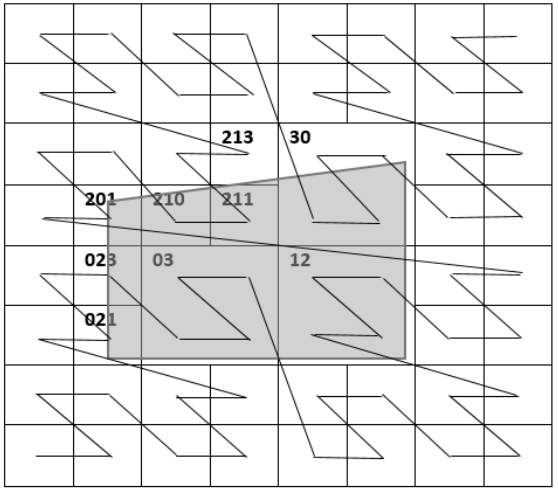
- Query SFC approach
  - The Quad-code has the special relationship with the Morton curve and Hilbert curve



(a) Query range with quadcode



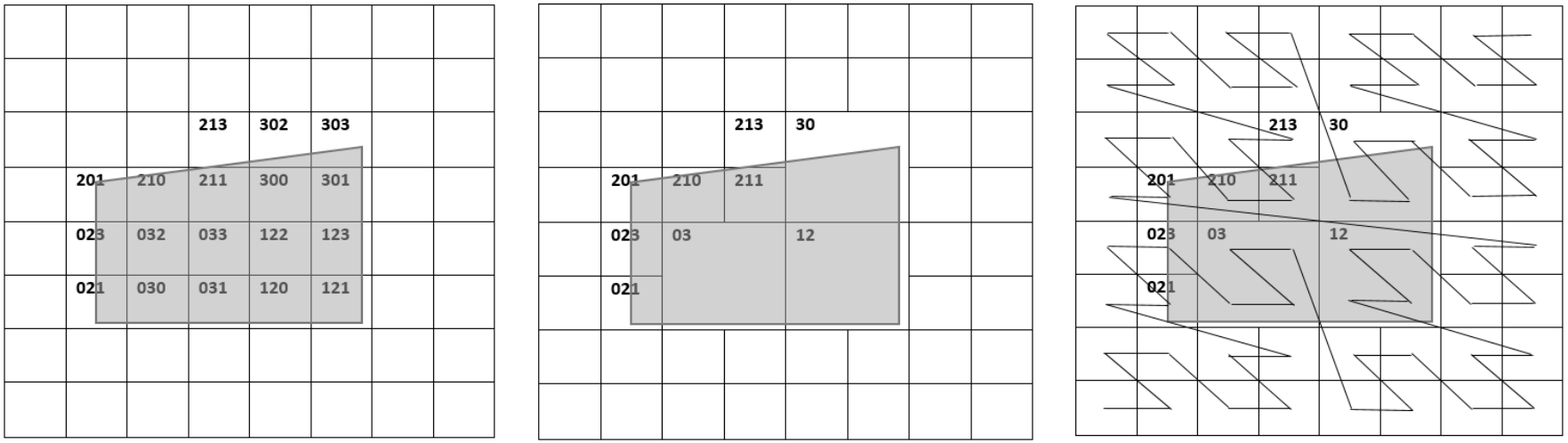
(b) merge consecutive range



(c) connect with SFC code

Query test

- Query SFC approach
  - The Quad-code has the special relationship with the Morton curve and Hilbert curve
  - Storage depth



(a) Query range with quadcode *(b) merge consecutive range* *(c) connect with SFC code*

(Figure 9 : the principle of query SFC)

Query test

- Query SFC approach
  - The Quad-code has the special relationship with the Morton curve and Hilbert curve
  - Storage depth
  - Query depth

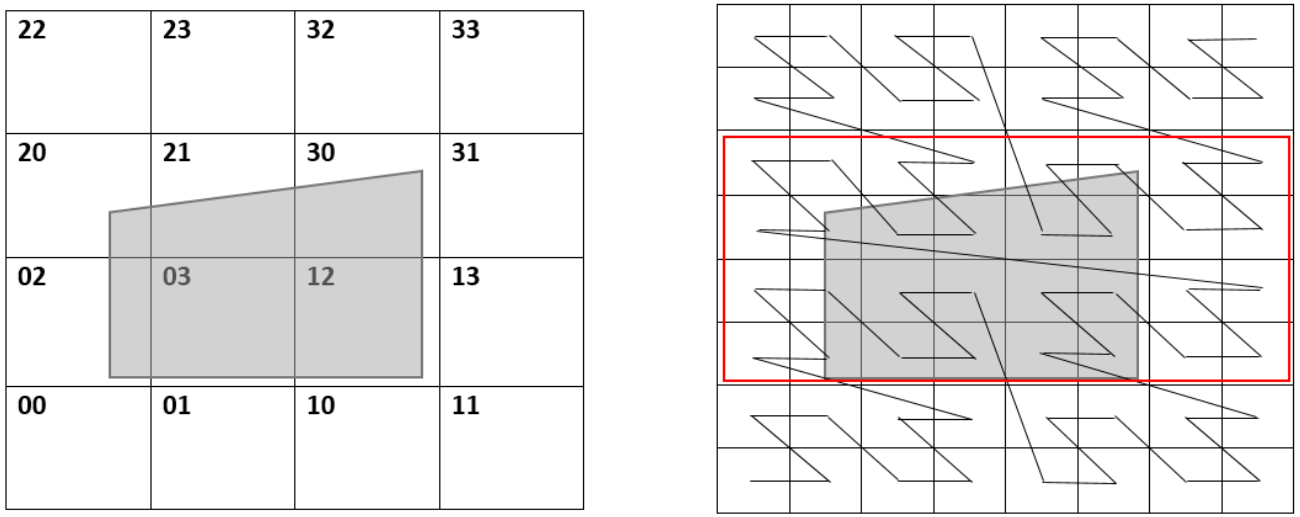


Figure 10: Query depth = 2 < 3

Query test

- Flow chart
  - Filtering step
  - Refinement step

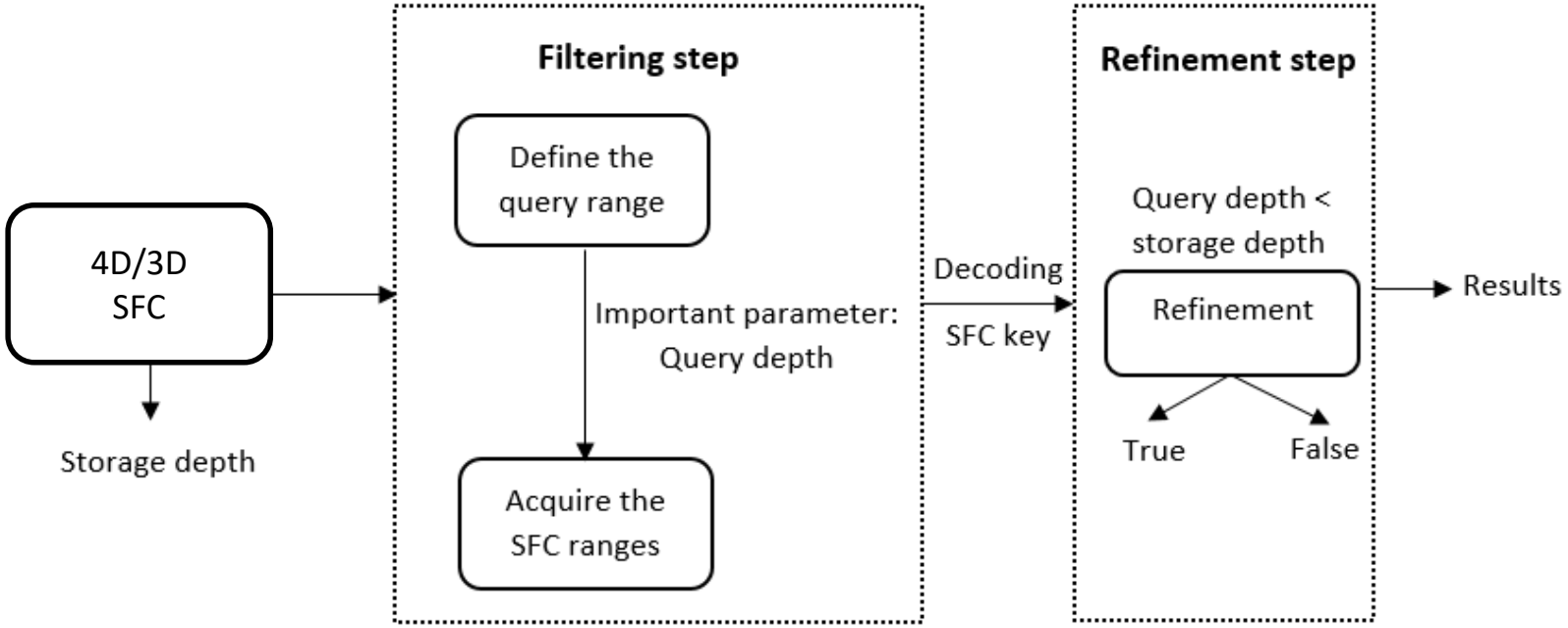


Figure 11 : the flow chart of query

Query test

- Flow chart
- Query type
  - Bounding box query

    The bounding box query is to ask all vessels in an certain range of an area in a period of time, and the period of time is 2mins.

Query test

- Flow chart
- Query type
  - Bounding box query
  - Trajectory query

    The trajectory query is to find the historical positions of vessels in the given time range.

Query test

- Flow chart
- Query type
- Test plan
  - 4D integrated approach
    - Indexes comparison
    - SFC comparison
    - Bounding box query
    - Trajectory query
  - 3D integrated approach
    - Bounding box query
    - Trajectory query

4D integrated approach

- Indexes comparison
  - B-Tree index and BRIN index
  - They will be compared in first query step : **filtering step**

4D integrated approach

- Indexes comparison
- SFC comparison
  - Morton curve and Hilbert curve

4D integrated approach

- Secondary indexes comparison
- SFC comparison
- Bounding box query
  - Make query box (4D hyperbox)
    - MMSI: whole range
    - Longitude, latitude, time (2mins): given range
  - Clustering and indexing using 4D SFC
  - Query 4D SFC

4D integrated approach

- Secondary indexes comparison
- SFC comparison
- Bounding box query
- Trajectory query
  - Make query box (4D hyperbox)
    - MMSI : given number
    - Longitude, latitude : whole range
    - Time : given range
  - Clustering and indexing using 4D SFC
  - Query 4D SFC

3D integrated approach

- Bounding box query
  - Make query box (3D hyperbox)
    - MMSI : Not in the hyperbox
    - Longitude, latitude, time : given range
  - Clustering and indexing using 3D SFC
  - Query 3D SFC

3D integrated approach

- Bounding box query
- Trajectory query
  - Filter table by MMSI (where mmsi = xxx)
  - Make query range (3D hyperbox)
    - MMSI : Not in the hyperbox
    - Longitude, latitude : whole range
    - Time : given range
  - Clustering and indexing using 3D SFC
  - Query 3D SFC

Benchmark

- Plain table

  Plain table is used to do the same query to test and query step by step.

  - Bounding box query

  select mmsi from orginal table where longitude >= lon1 and longitude < lon2 and latitude >= lat1 and latitude < lat2 and ts >= ts1 and ts < ts2;

  - Trajectory query

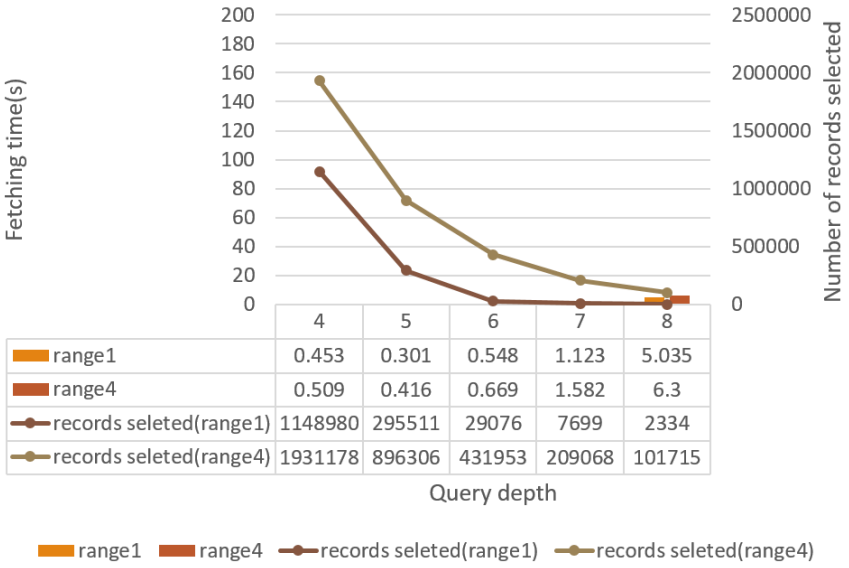
  select mmsi from orginal table where mmsi = S and ts >= ts1 and ts < ts2;

Outline

- Introduction
- Related works
- Methodology
- Results & discussion
- Conclusions & future works

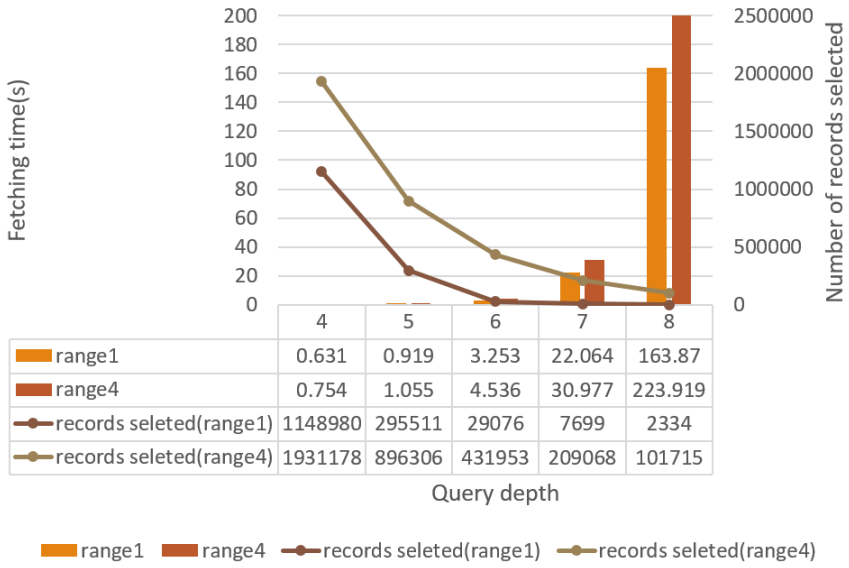
4D integrated approach

- Indexes comparison
  - BRIN index



	4	5	6	7	8
range1	0.453	0.301	0.548	1.123	5.035
range4	0.509	0.416	0.669	1.582	6.3
records seleted(range1)	1148980	295511	29076	7699	2334
records seleted(range4)	1931178	896306	431953	209068	101715

Pages per range = 32



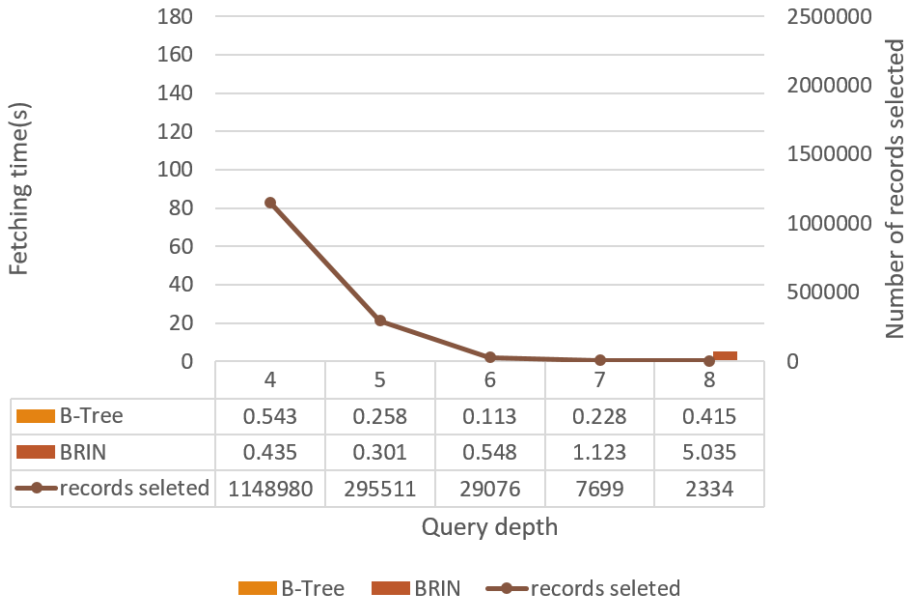
	4	5	6	7	8
range1	0.631	0.919	3.253	22.064	163.87
range4	0.754	1.055	4.536	30.977	223.919
records seleted(range1)	1148980	295511	29076	7699	2334
records seleted(range4)	1931178	896306	431953	209068	101715

pages per range = 64

Figure 12 the comparison regarding to the pages per range

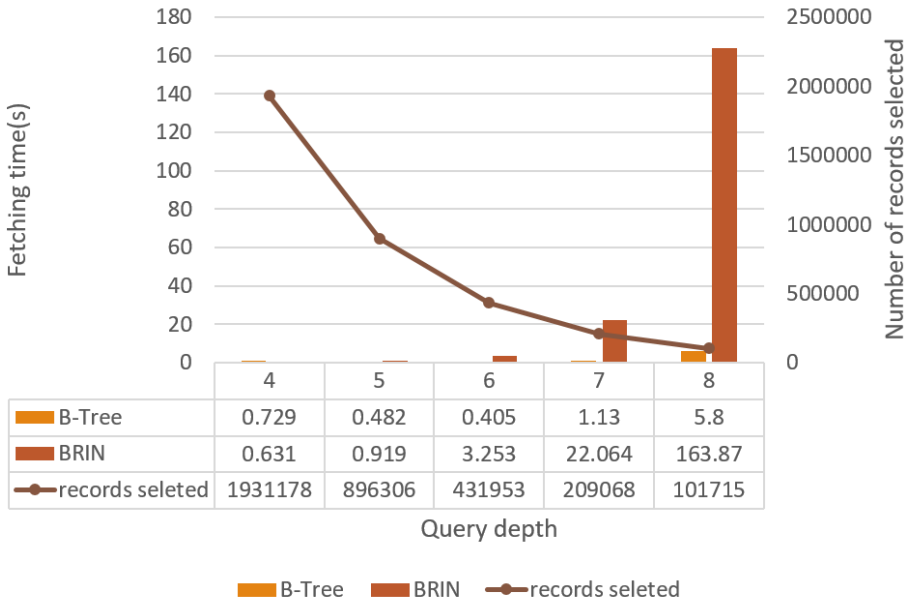
4D integrated approach

- Secondary indexes comparison
  - BRIN index
  - BRIN index VS BRIN index



	4	5	6	7	8
B-Tree	0.543	0.258	0.113	0.228	0.415
BRIN	0.435	0.301	0.548	1.123	5.035
records seleted	1148980	295511	29076	7699	2334

Query box 1



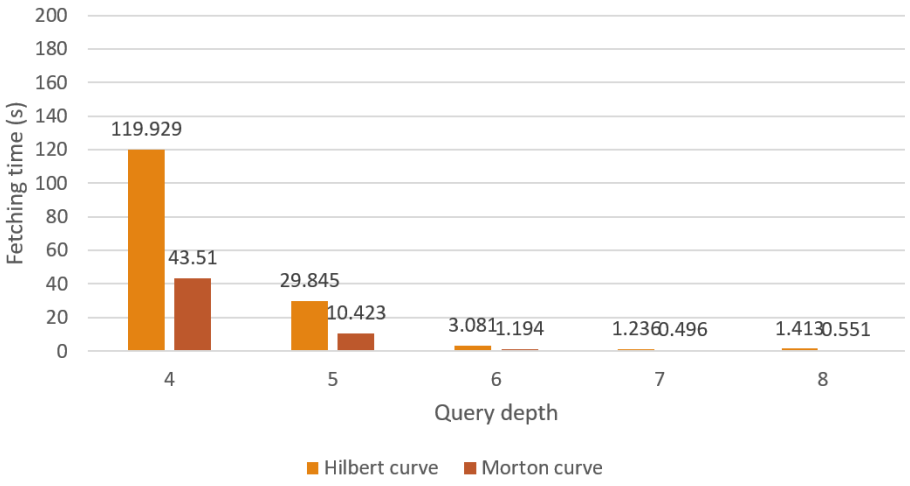
	4	5	6	7	8
B-Tree	0.729	0.482	0.405	1.13	5.8
BRIN	0.631	0.919	3.253	22.064	163.87
records seleted	1931178	896306	431953	209068	101715

Query box 2

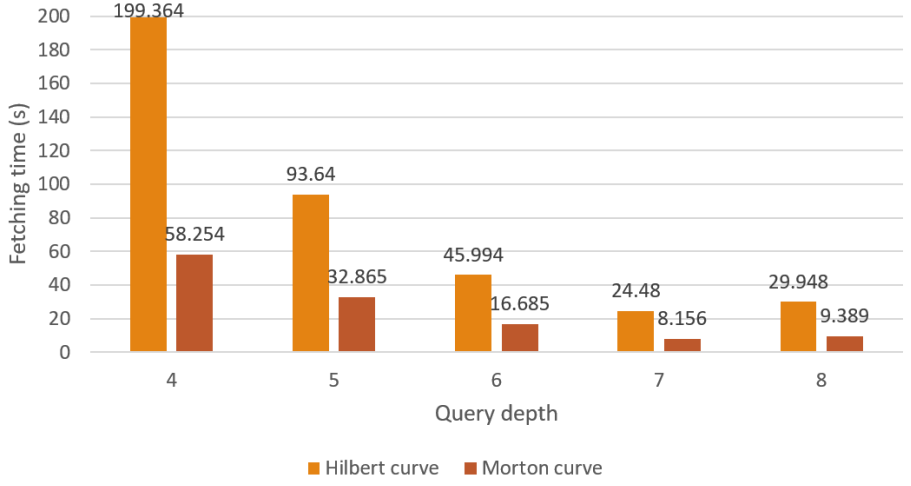
Figure 13 : Comparison between B-Tree index and BRIN index

4D integrated SFC approach

- SFC comparison
  - Morton curve and Hilbert curve
    - Fetching time



Query box 1

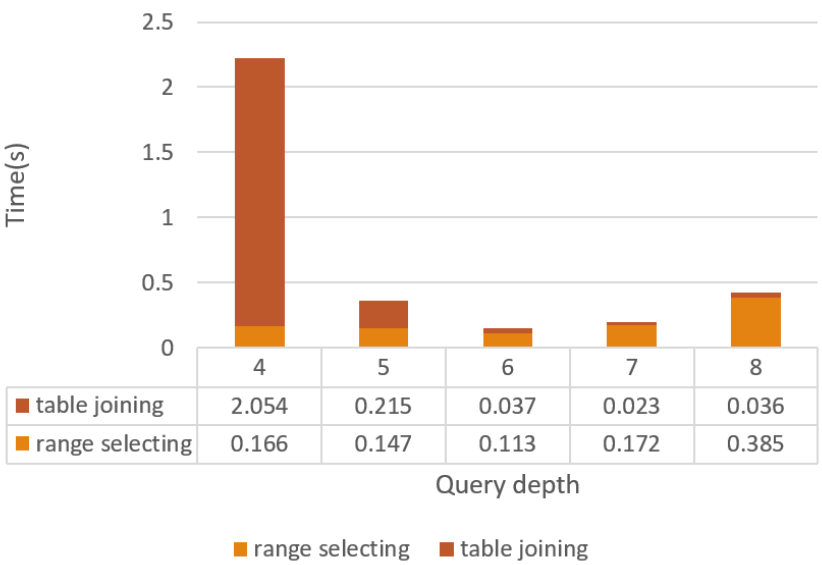


Query box 2

Figure 14 : Comparison between B-Tree index and BRIN index

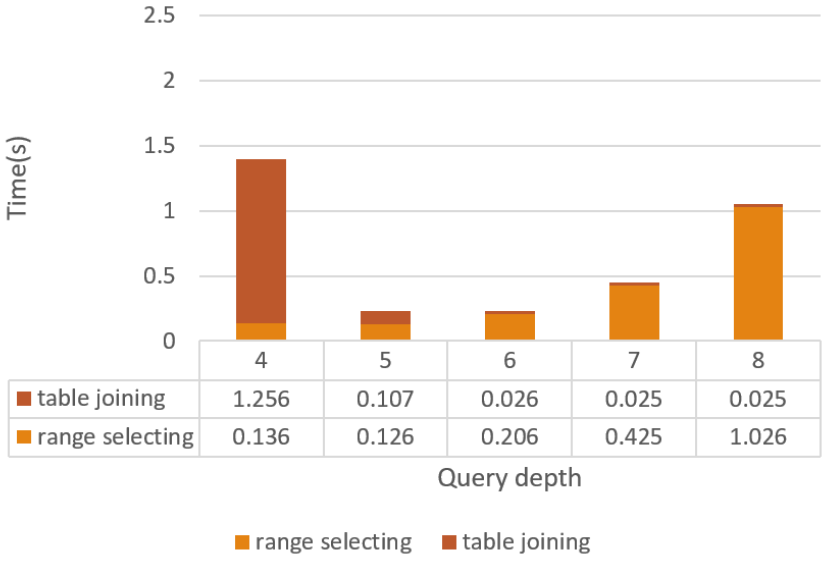
4D integrated SFC approach

- SFC comparison
  - Morton curve and Hilbert curve
    - Time needed for each step



	4	5	6	7	8
table joining	2.054	0.215	0.037	0.023	0.036
range selecting	0.166	0.147	0.113	0.172	0.385

Using Morton curve



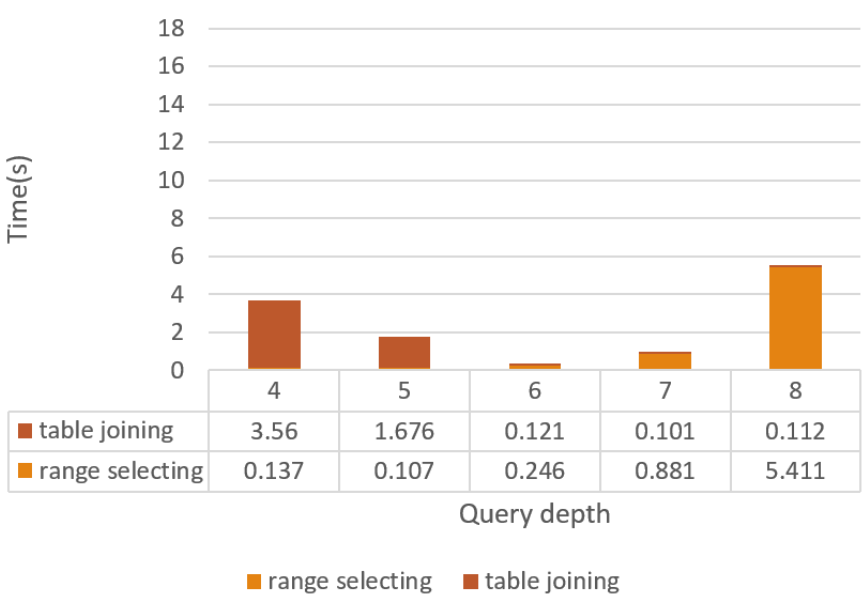
	4	5	6	7	8
table joining	1.256	0.107	0.026	0.025	0.025
range selecting	0.136	0.126	0.206	0.425	1.026

Using Hilbert curve

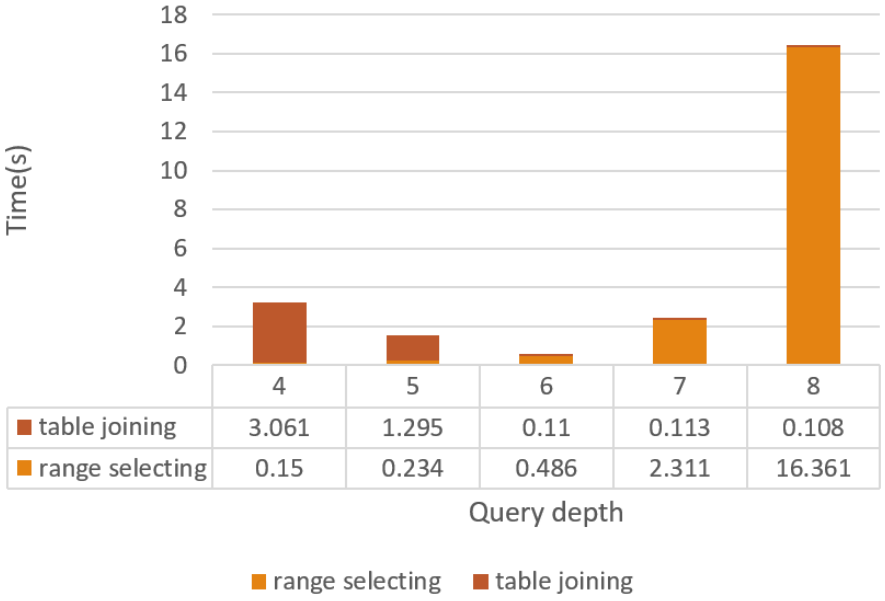
Figure 15 : Time needed for each step (query box 1)

4D integrated SFC approach

- SFC comparison
  - Morton curve and Hilbert curve
    - Time needed for each step



Using Morton curve



Using Hilbert curve

Figure 16 : Time needed for each step (query box 4)

4D integrated SFC approach

- SFC comparison
  - Morton curve and Hilbert curve
    - Locality

Curve	(max - min) value
Morton curve	12116959349677267343076078363237 12
Hilbert curve	289221564322095599394595202981386

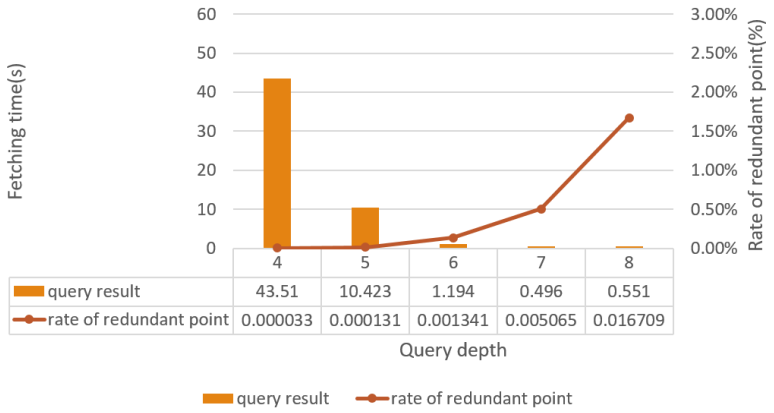
Table 1 : The SFC range after query (box 1)

Curve	(max - min) value
Morton curve	663351605602299167786941580690847624
Hilbert curve	209941383916602208768840196479418653

Table 2 : The SFC range after query (box 4)

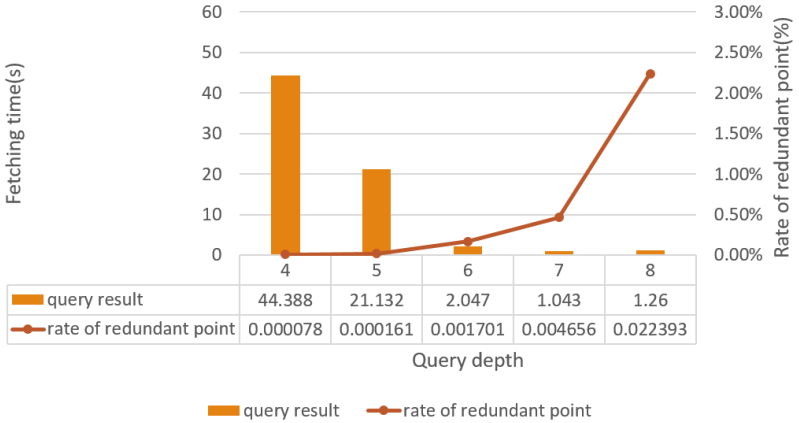
4D integrated SFC approach

- Bounding box query



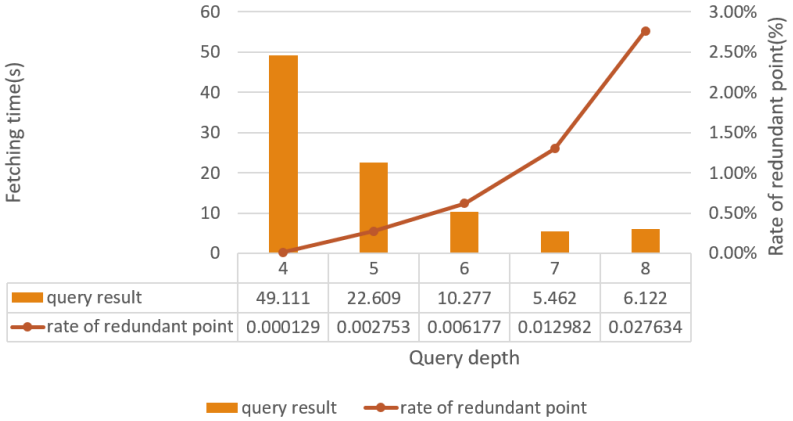
	4	5	6	7	8
query result	43.51	10.423	1.194	0.496	0.551
rate of redundant point	0.000033	0.000131	0.001341	0.005065	0.016709

query box 1



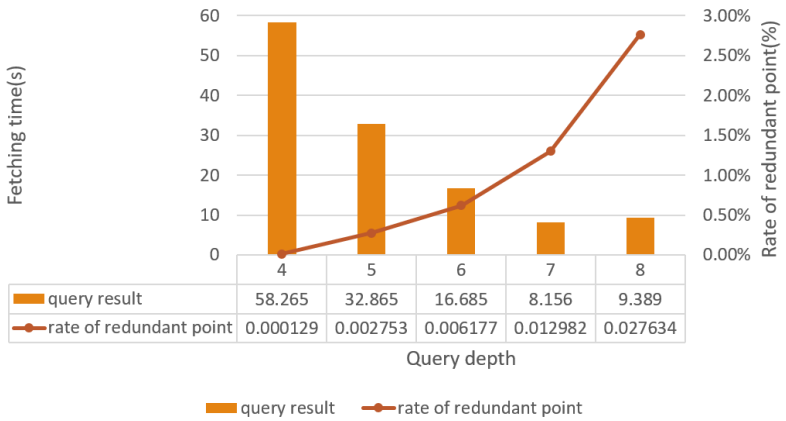
	4	5	6	7	8
query result	44.388	21.132	2.047	1.043	1.26
rate of redundant point	0.000078	0.000161	0.001701	0.004656	0.022393

query box 2



	4	5	6	7	8
query result	49.111	22.609	10.277	5.462	6.122
rate of redundant point	0.000129	0.002753	0.006177	0.012982	0.027634

query box 3



	4	5	6	7	8
query result	58.265	32.865	16.685	8.156	9.389
rate of redundant point	0.000129	0.002753	0.006177	0.012982	0.027634

query box 4

Figure 17 : Bounding box query using 4D SFC approach

4D integrated SFC approach

- Trajectory query

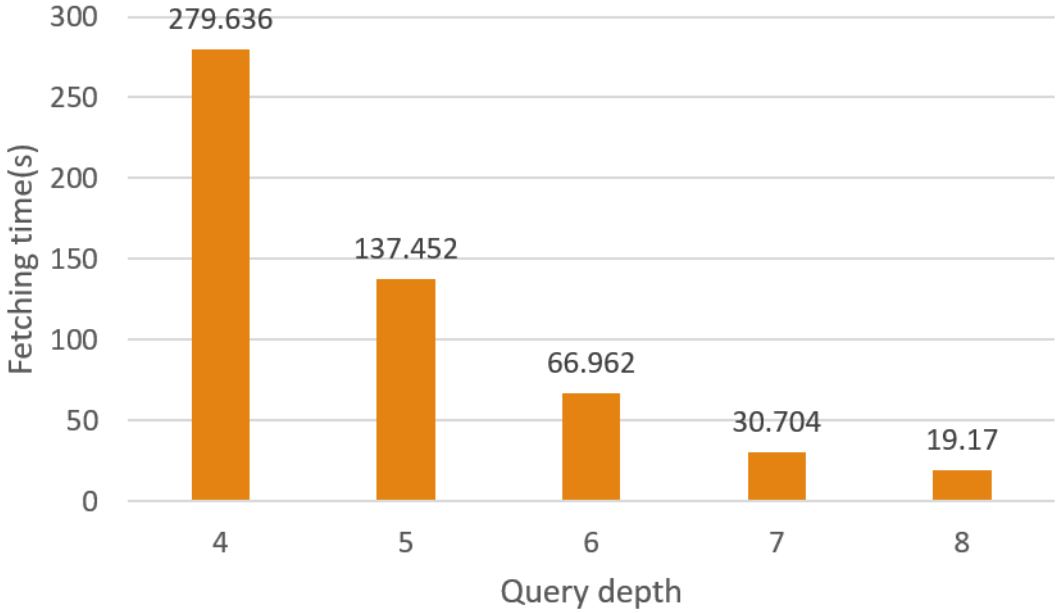
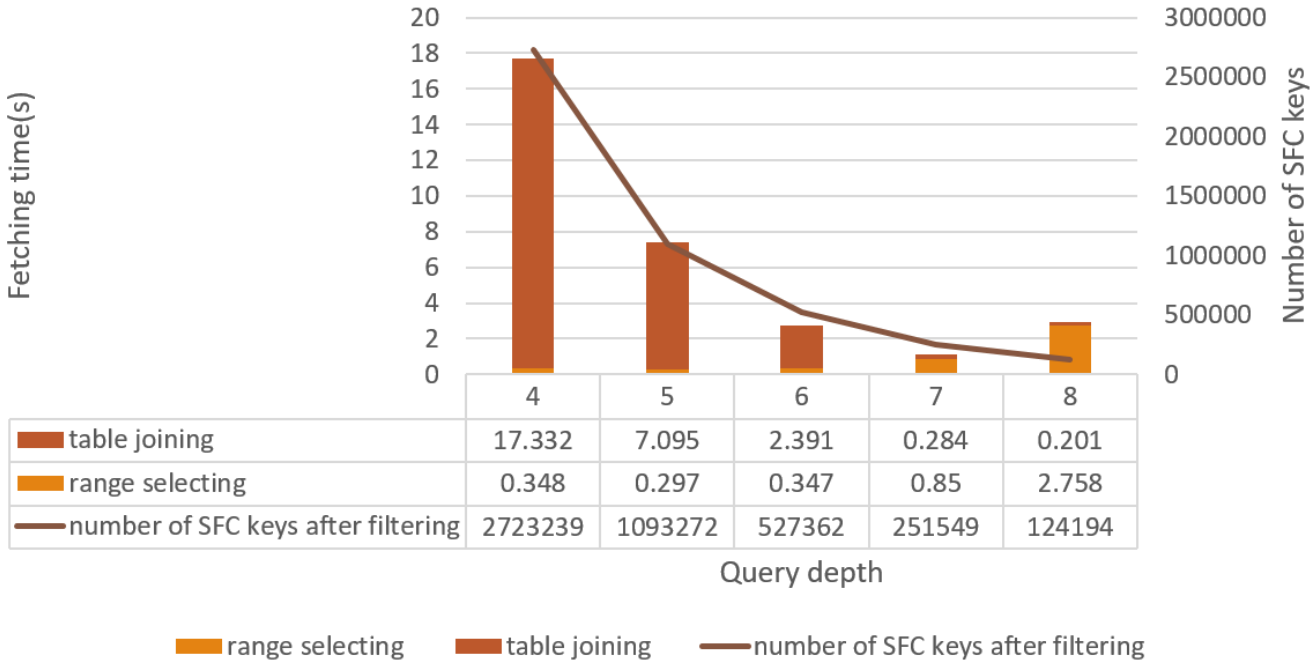


Figure 18 : Trajectory query using 4D integrated approach

4D integrated SFC approach

- Trajectory query

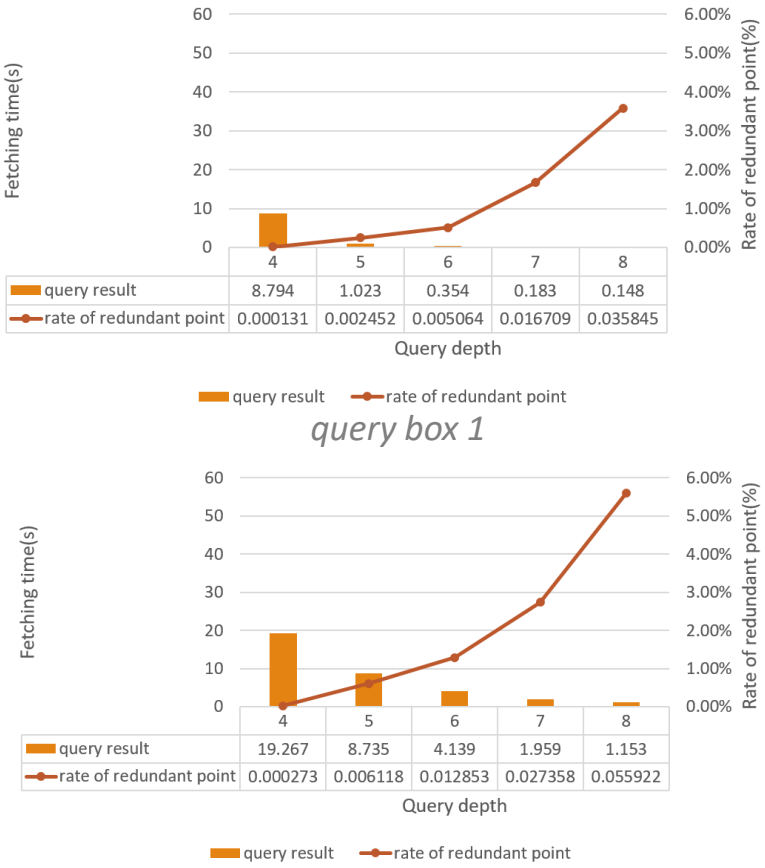


	4	5	6	7	8
table joining	17.332	7.095	2.391	0.284	0.201
range selecting	0.348	0.297	0.347	0.85	2.758
number of SFC keys after filtering	2723239	1093272	527362	251549	124194

Figure 19 : The time needed in each step

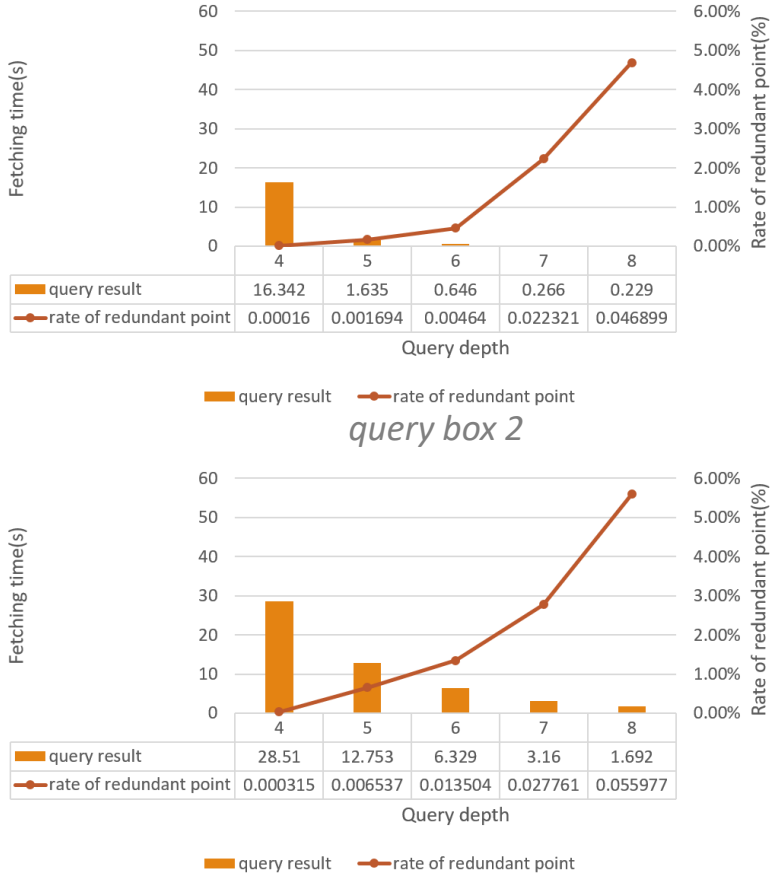
3D integrated SFC approach

- Bounding box query



	4	5	6	7	8
query result	8.794	1.023	0.354	0.183	0.148
rate of redundant point	0.000131	0.002452	0.005064	0.016709	0.035845

query box 1



	4	5	6	7	8
query result	16.342	1.635	0.646	0.266	0.229
rate of redundant point	0.00016	0.001694	0.00464	0.022321	0.046899

query box 2

	4	5	6	7	8
query result	19.267	8.735	4.139	1.959	1.153
rate of redundant point	0.000273	0.006118	0.012853	0.027358	0.055922

query box 3

	4	5	6	7	8
query result	28.51	12.753	6.329	3.16	1.692
rate of redundant point	0.000315	0.006537	0.013504	0.027761	0.055977

query box 4

Figure 20 : Bounding box query using 3D SFC approach

3D integrated SFC approach

- Bounding box query

Query depth	4D integrated approach	3D integrated approach
4	1148980	297461
5	295511	29077
6	29076	7700
7	7699	2334
8	2334	1088

Table 2 : The number of SFC keys after filtering (box1)

Query depth	4D integrated approach	3D integrated approach
4	1931178	903096
5	896306	435338
6	431965	210739
7	209068	102517
8	101715	50842

Table 3 : The number of SFC keys after filtering (box4)

3D integrated SFC approach

- Trajectory query

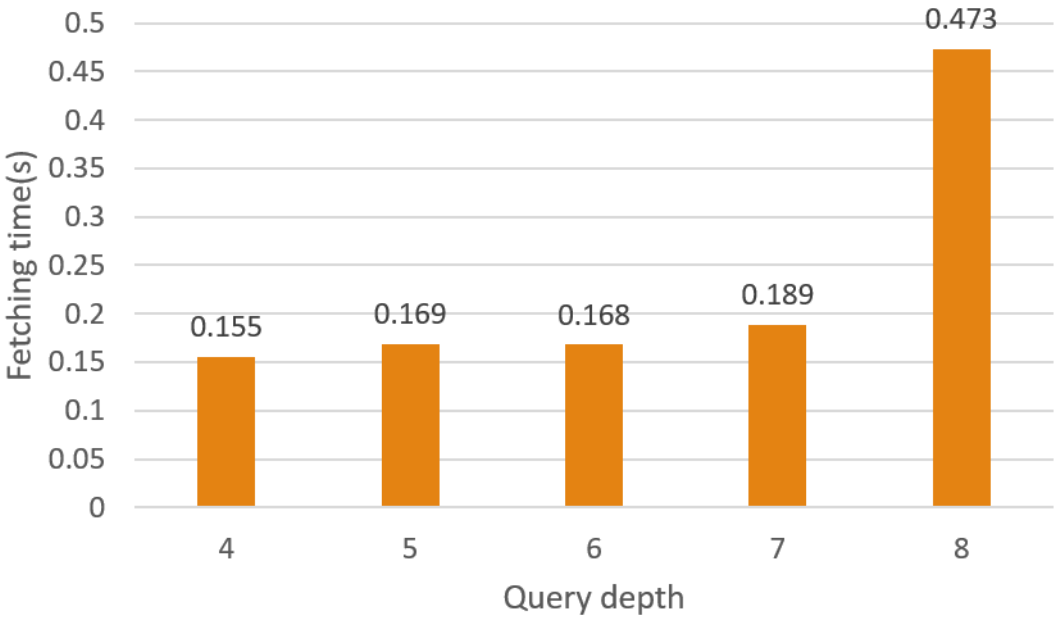


Figure 21 : Trajectory query using 3D SFC approach

Benchmark

- Bounding box query

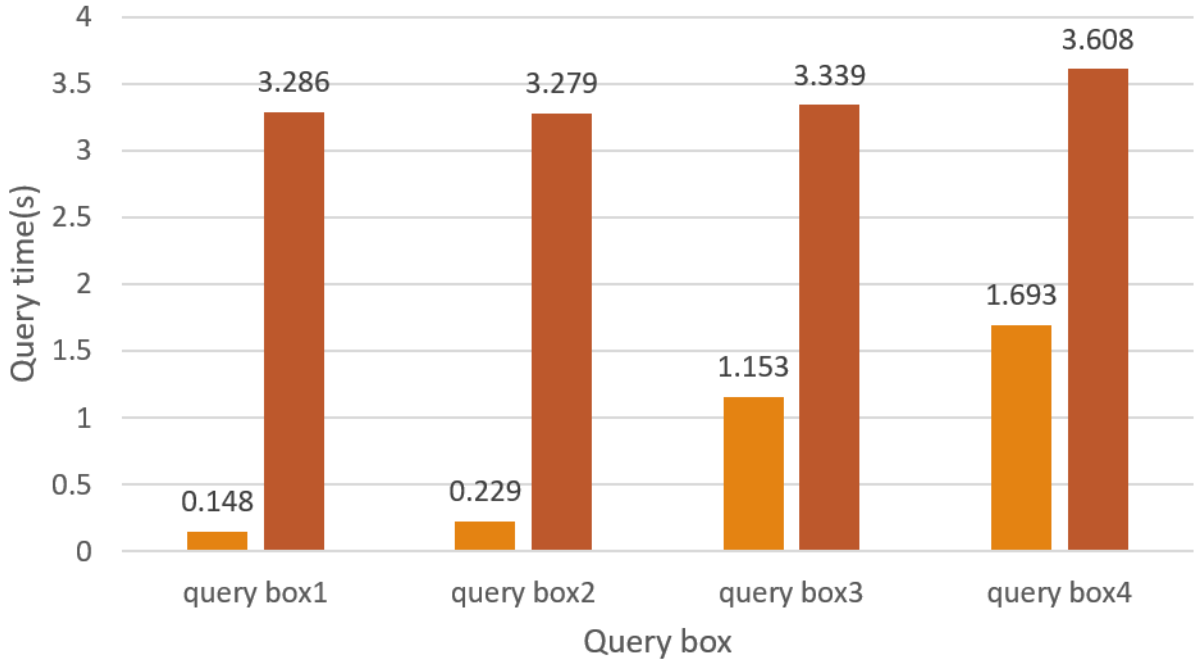


Figure 22 : Comparison between 3D approach and plain table

Benchmark

- Trajectory query

Query approach	Query time (s)
3D SFC approach	0.155
Plain table	3.560

Table 4 : query time using plain table in database

Outline

- Introduction
- Related works
- Methodology
- Results & discussion
- Conclusions & future works

Conclusion

- The answers of main questions:
  - How to efficiently manage 4D AIS data (Longitude, Latitude, Time, MMSI) of vessels to do the fast query by using Space Filling Curve in PostgreSQL?

    1) Choose a suitable Space Filling Curve
    2) Find a proper way to organize the data
    3) Select an efficient indexing method

Conclusion

- The answers of Sub-questions:
  - How to manage the 4D data is better to support the efficient query? Using 4D integrated approach or 3D integrated approach?

    The efficient management of the 4D data depends on the data and the query we used. For the 4D data (Longitude, latitude, time, MMSI) and the query (bounding box query and trajectory query) I used, I proposed the 4D approach and 3D approach.

    From the results, the 3D approach performs better regarding to the query I used. Less records were used to do the filtering and refining make the query more efficient.

Conclusion

- The answers of research questions:
  - How to manage the 4D data is better to support the efficient query?
  - How to scale data in each dimension properly to compute the SFC key?

    Whether for 4D approach or 3D approach, data in each dimension should have the roughly equal bit length.

    While, if the dimensions of data or the size of the datasets increase, scaling data to a small and fix range is also a good choice.

Conclusion

- The answers of research questions:
  - How to manage the 4D data is better to support the efficient query?
  - How to scale data in each dimension properly to compute the SFC key?
  - Which SFC performs better in 4D data querying? Morton curve or Hilbert curve?

    From the fetching time: Morton curve performs better because the decoding Hilbert key takes lots of time.

    From the locality: Hilbert curve performs better.

Conclusion

- The answers of research questions:
  - How to manage the 4D data is better to support the efficient query?
  - How to scale data in each dimension properly to compute the SFC key?
  - Which SFC performs better in 4D data querying? Morton curve or Hilbert curve?
  - How about the BRIN index as the secondary indexing method?

The BRIN index is a great index because the quite small size compared to the B-Tree index and it also can support the efficient query. While, the time for creating the BRIN index will be a bit long if the data is not ordered.

Future work

- The data may not only be limited to AIS data or the data I use. There are a lot of useful information in AIS data, which can be used as research objects.
- The geometry of the query range could be changed. What I used is the hyperbox, query boxes of other shapes can also be studied.

Thanks for your attention!

Questions?