Efficient Distributed Cross Match Queries in the Virtual Observatory

Individual Research Project

Pratyusha Doddapaneni

Outline

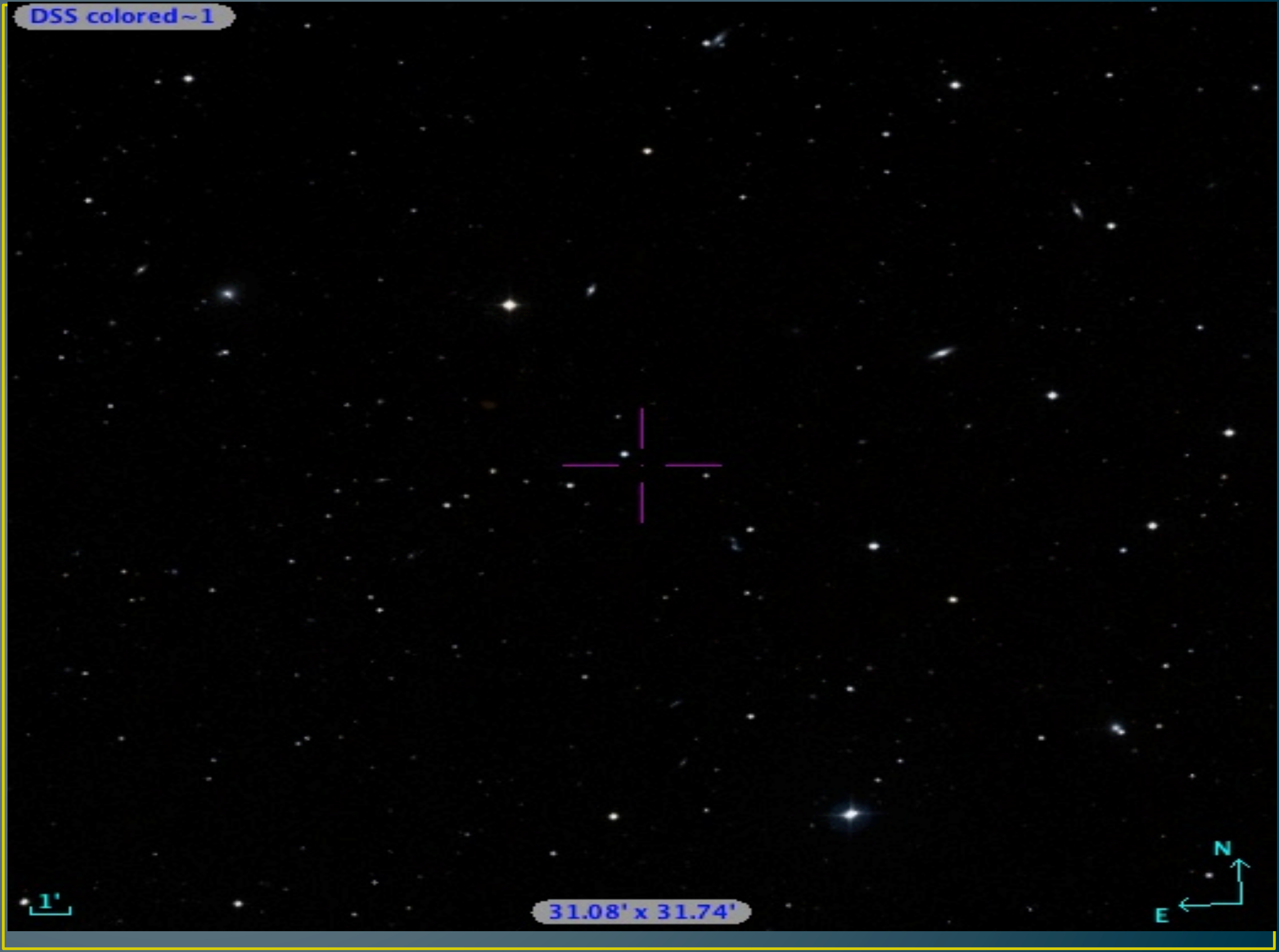
- **Background**
- Rationale
- Requirements / Environment
- Design
- Approaches
  - Implementation
  - Analysis
- Discussion
- Summary

Background

- Explosion in amount of data generated

- Problem ?
    - interoperable use of world's astronomical data resources

- Virtual Observatory
    - global electronic access to the astronomical data from the various astronomical databases
    - providing data analysis techniques via standards (TAP, VOTable …)


DSS colored~1
1'
31.08' x 31.74'
N
E

Outcome

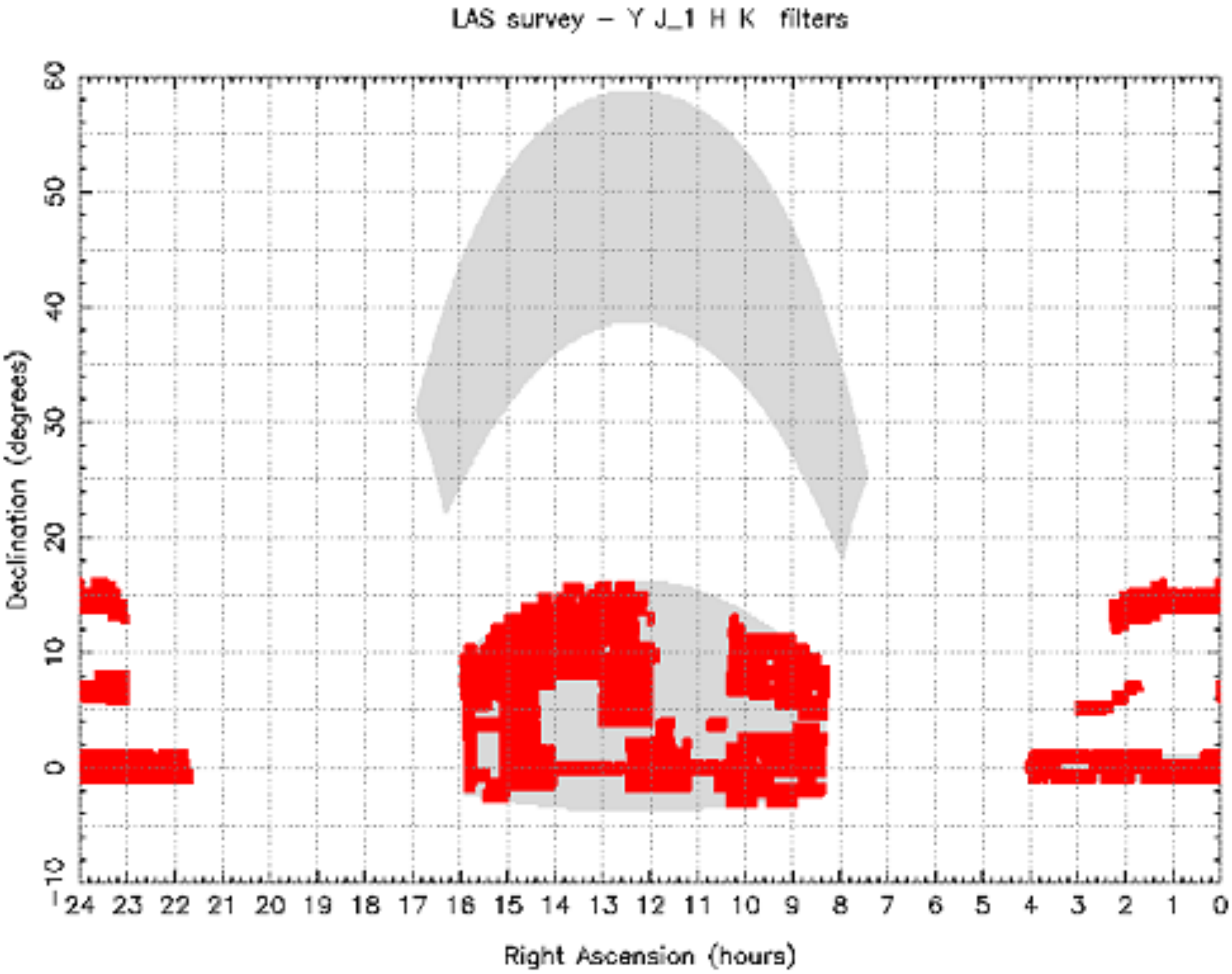
- Set of recommendations
- Drawn from the inferences
- Pertinent to the results and different metrics considered

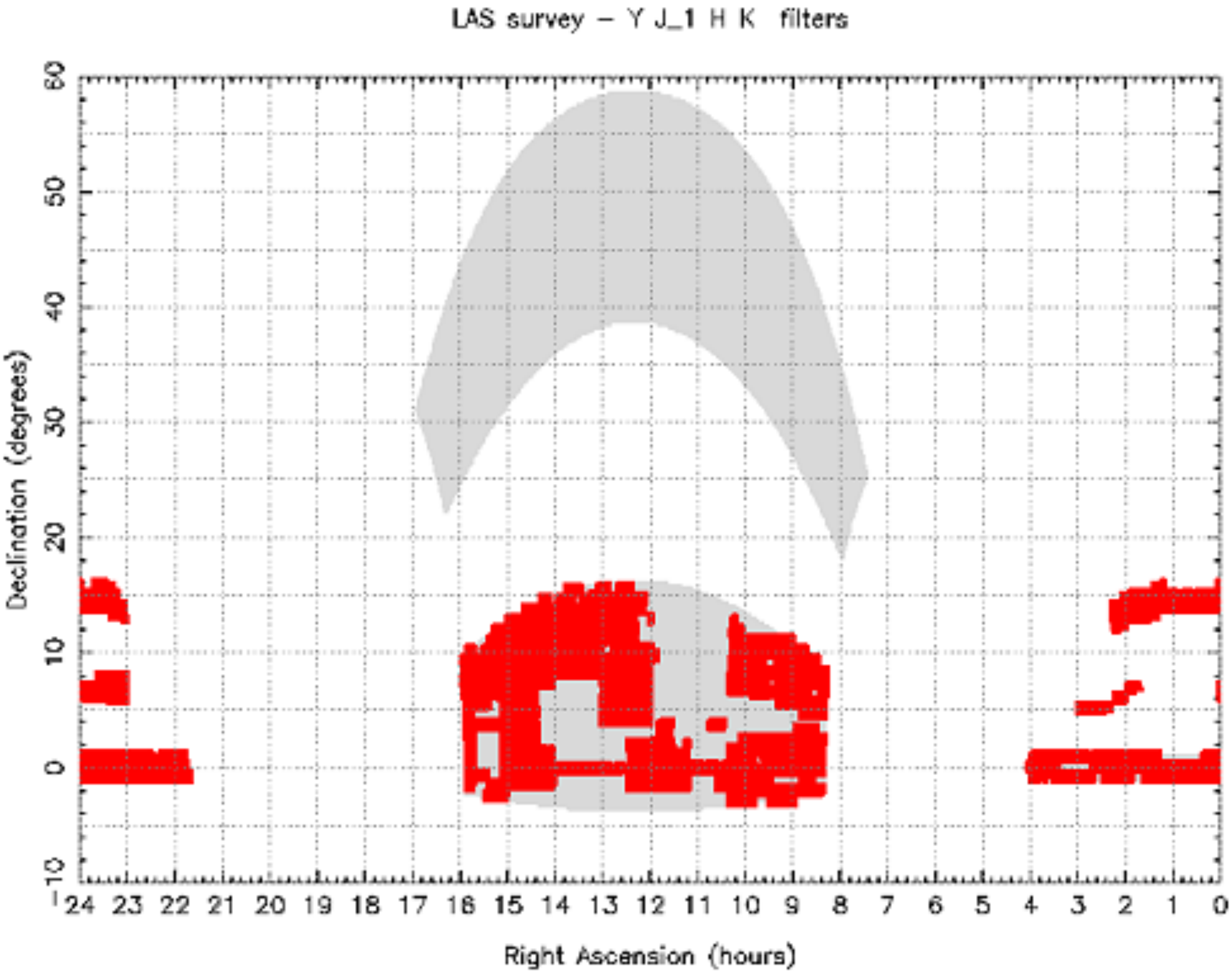
Outline

- Background
- Rationale
- **Requirements / Environment**
- Design
- Approaches
  - Implementation
  - Analysis
- Discussion
- Summary

Data Requirements

- Catalogues
  - UKIDSS DR5
  - SDSS Best DR7
- UKIDSS true near IR counterpart to SDSS
- LAS counterpart of UKIDSS used to perform cross match against SDSS


LAS survey – Y J_1 H K filters
Declination (degrees)
60
50
40
30
20
10
0
-10
Right Ascension (hours)
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0


LAS survey – Y J_1 H K filters
Declination (degrees)
60
50
40
30
20
10
0
-10
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
Right Ascension (hours)

Computing Environment

- Server machine set up at WFAU, Royal Observatory
  - Windows SQL Server 2008
  - Main memory: 16GB RAM
  - Disk space: Databases spread over 3 RAID5 arrays with 5.45 TB each
- Queries analysed based on elapsed time
  - Using the ‘cold’ database status

‘Cold’ database

- Mimicking the state of freshly restarted server
- Forms good basis of comparison of timings
- Achieved using MS-SQL DBCC

```
DBCC FREEPROCCACHE
DBCC DROPCLEANBUFFERS
DBCC FREESESSIONCACHE
DBCC FREESYSTEMCACHE
```

Outline

- Background
- Rationale
- Requirements / Environment
- **Design**
- Approaches
  - Implementation
  - Analysis
- Discussion
- Summary

Types of queries

- Self match queries (LAS vs. LAS)
    - Indexed
    - Indexed/Un-indexed

- Cross match queries (LAS vs. SDSS)
    - Indexed
    - Indexed/Un-indexed

Metrics for analysis

- With respect to
  - Elapsed times
  - Indexed and un-indexed attributes
  - Scalability
  - Impact of network times on query times

Outline

- Background
- Rationale
- Requirements / Environment
- Design
- **Approaches**
  - Implementation
  - Analysis
- Discussion
- Summary

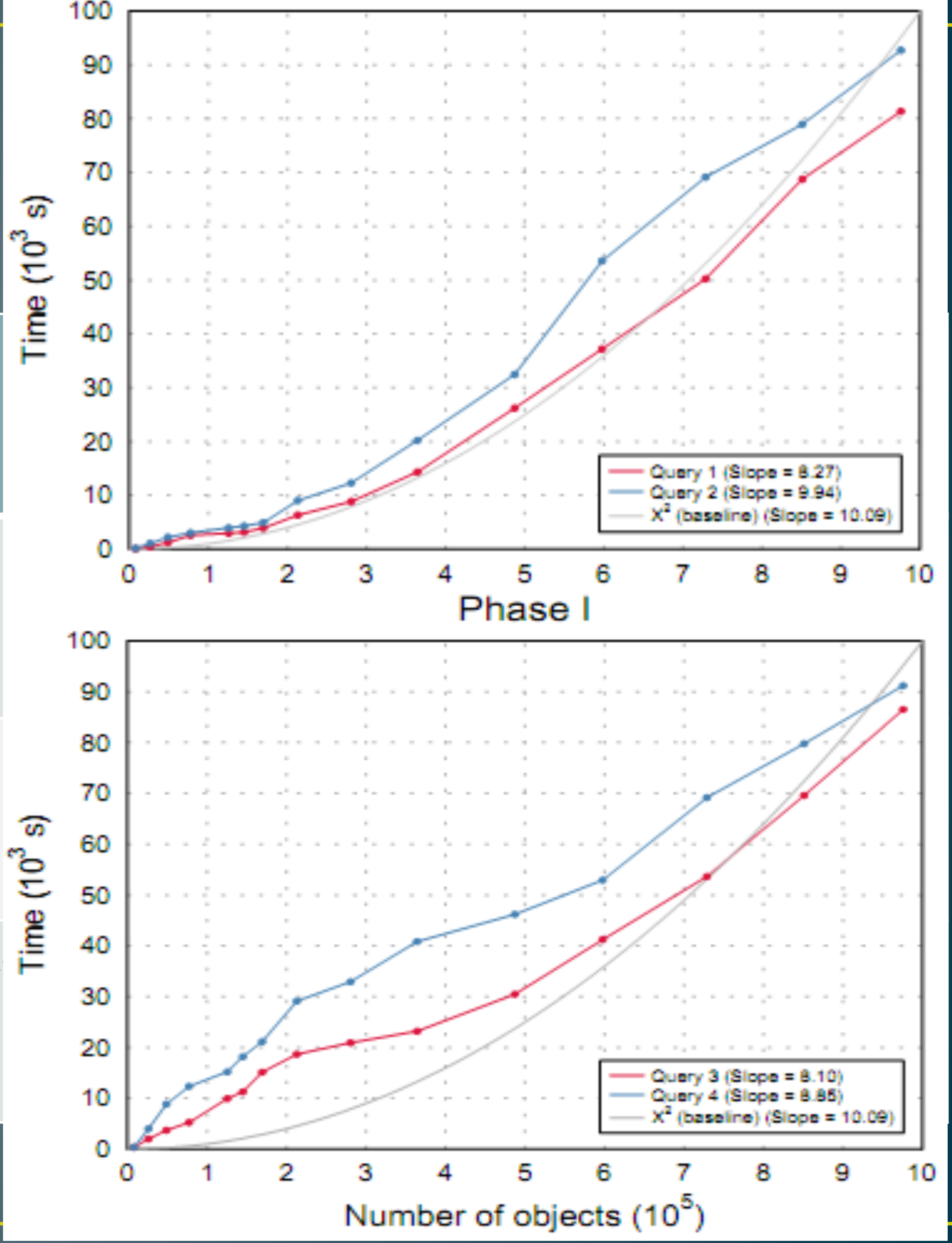
Database Based Approaches

Brute Force RA/Dec cut followed by GCD

- Great circle distance
  - shortest path between pair of objects
- Calculations of GCD but only over certain range of values restricted by RA & Dec values
- Advantages
  - Simple & easier to implement
- Disadvantages
  - complex formula
  - Cost ~ O(m*n)

Results

Criteria	
Query times	
Attributes	
Scalability	R



Pre-computed cross-neighbour tables

- Pre-computing join indexes
  - based on spatial proximities

- Extra costs ?
  - generated table ?
  - network costs ?
  - computation & storage costs ?
  - maintenance costs ?

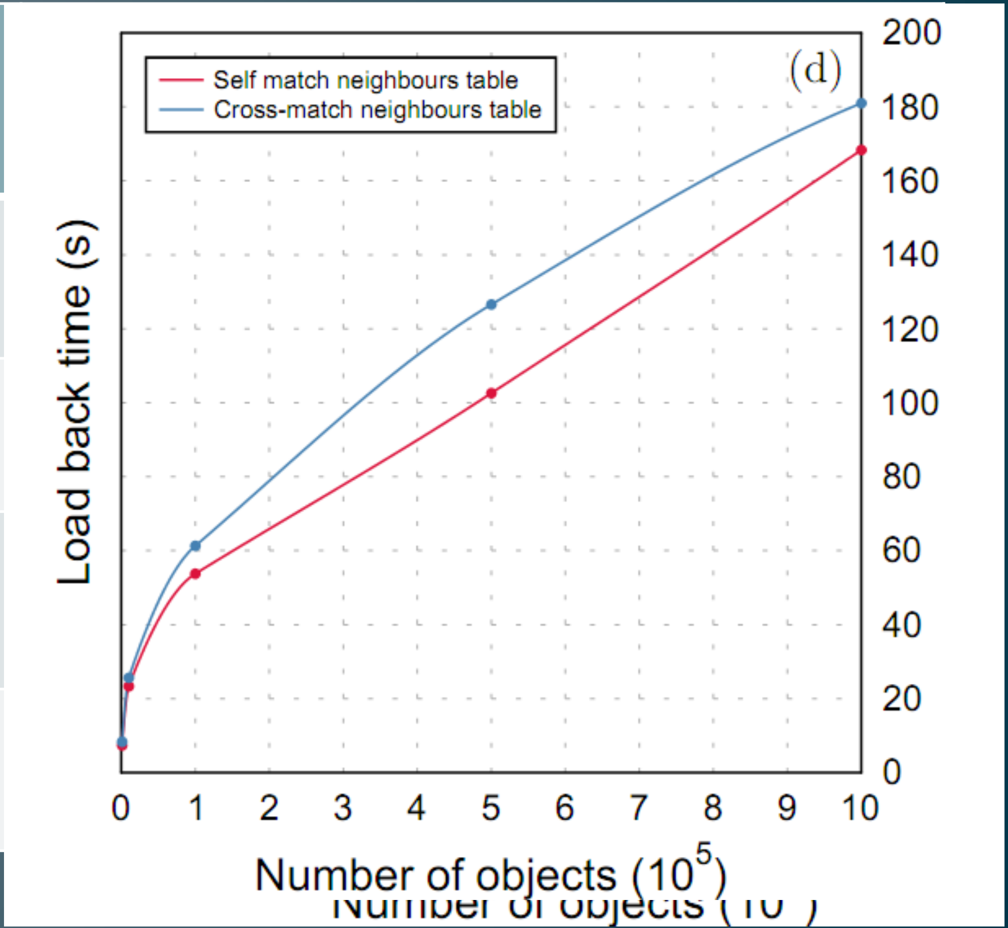
Pre-computed cross-neighbour tables

- Advantages
    - Faster query computation
    - Minimal computation overhead
- Disadvantages
    - Storage costs
    - Initial computation costs
        - Increases with increase in the size of the catalogues
    - Maintenance costs

Results

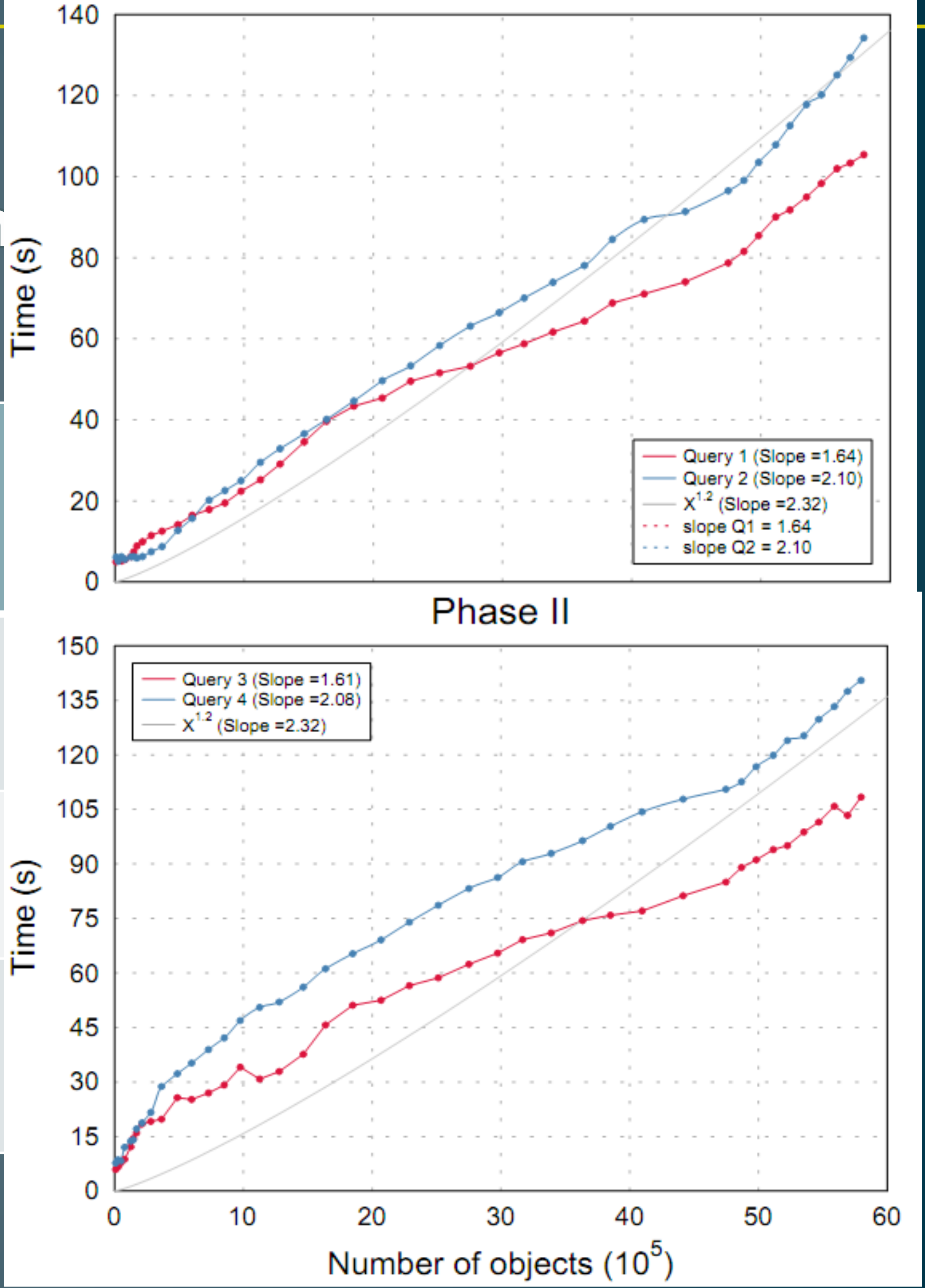
- Costs for generation of the neighbours table

Criteria
Extraction of tuples
Neighbours table construction
Memory footprint
Load back times



Results – Query an

Criteria
Query times
Attributes
Scalability

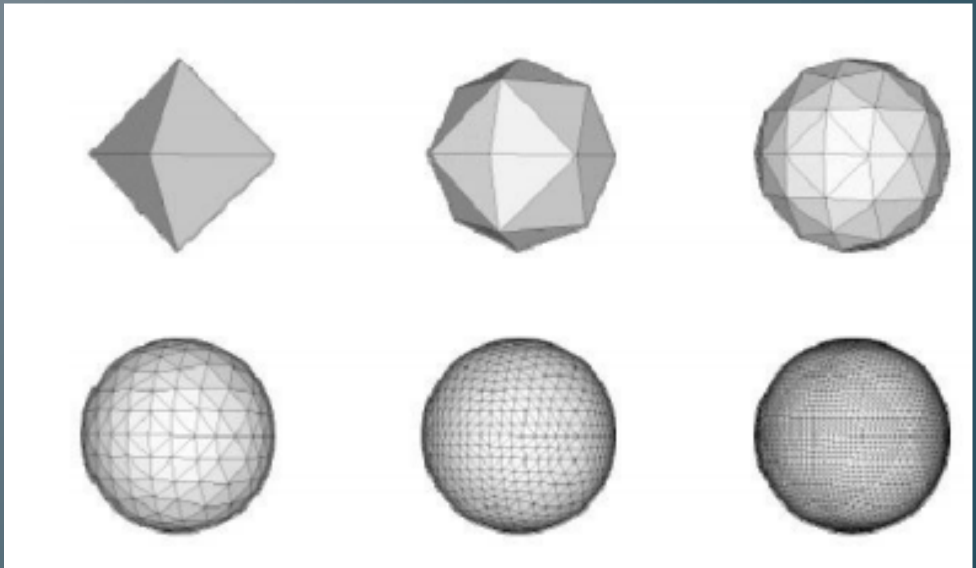


Localisation via ‘pixel code’ followed by GCD

- Encoding the rough position of each source entry
  - Partition the sphere into pixels of appropriate size
  - Localise the position of every source onto the pixel map
- Spatial Indexing Schemes
  - Hierarchical Triangular Mesh (HTM)
  - Hierarchical Equal Area isoLatitude Pixelisation (HEALPix)

Hierarchical Triangular Mesh

- Recursive decomposition of celestial spherical surface into similar sized & shaped triangles
  - Base trixels defined by projection of octahedron faces onto the surface of the sphere



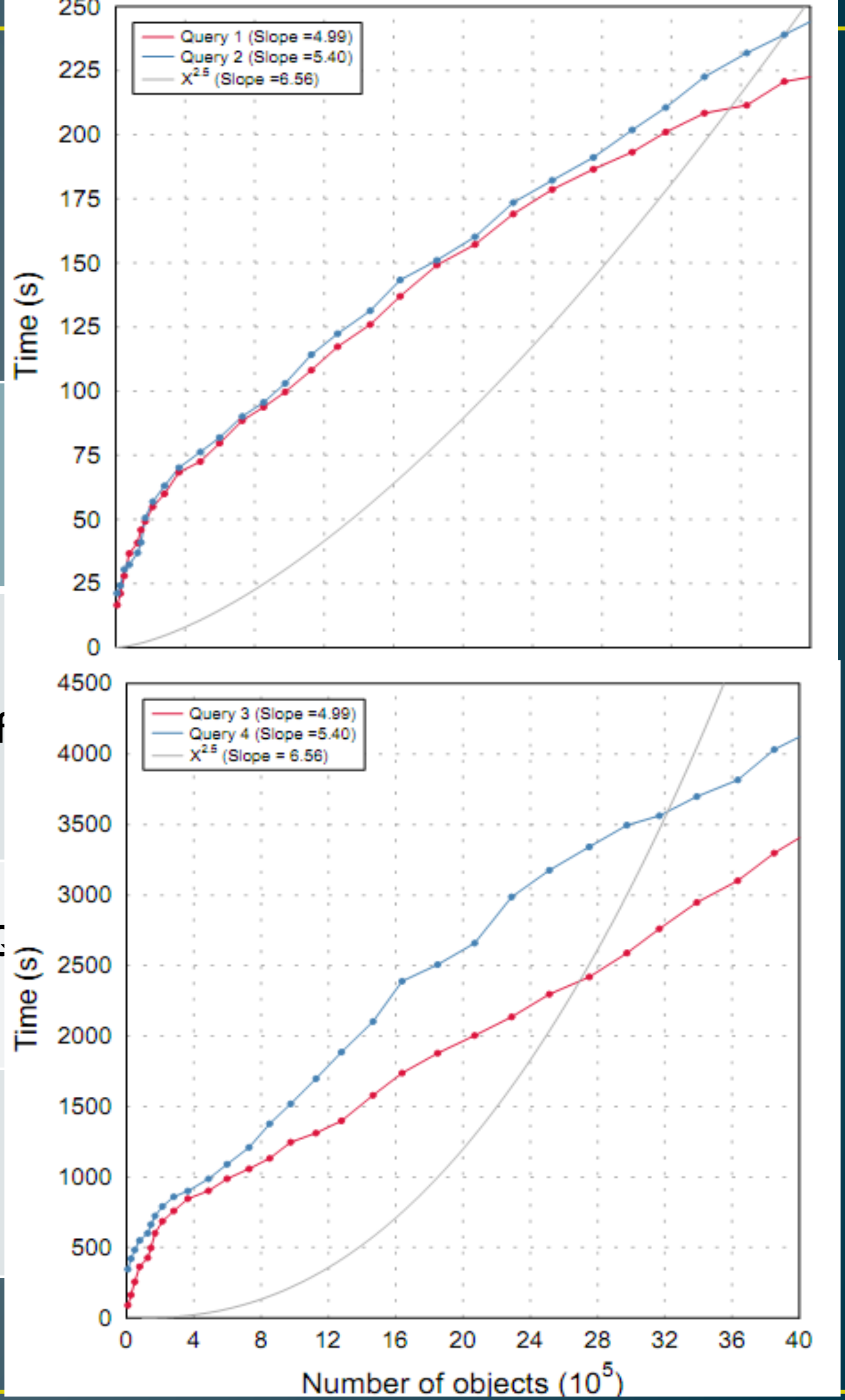
Hierarchical Triangular Mesh

- Advantages
  - Fast object matching
  - Acceptable performance

- Disadvantages
  - Deeper levels → losing accuracy in the data
  - Possibility of overshooting the recursive level & computationally expensive

Results

Criteria	
Query times	f
Attributes	Q
Scalability	

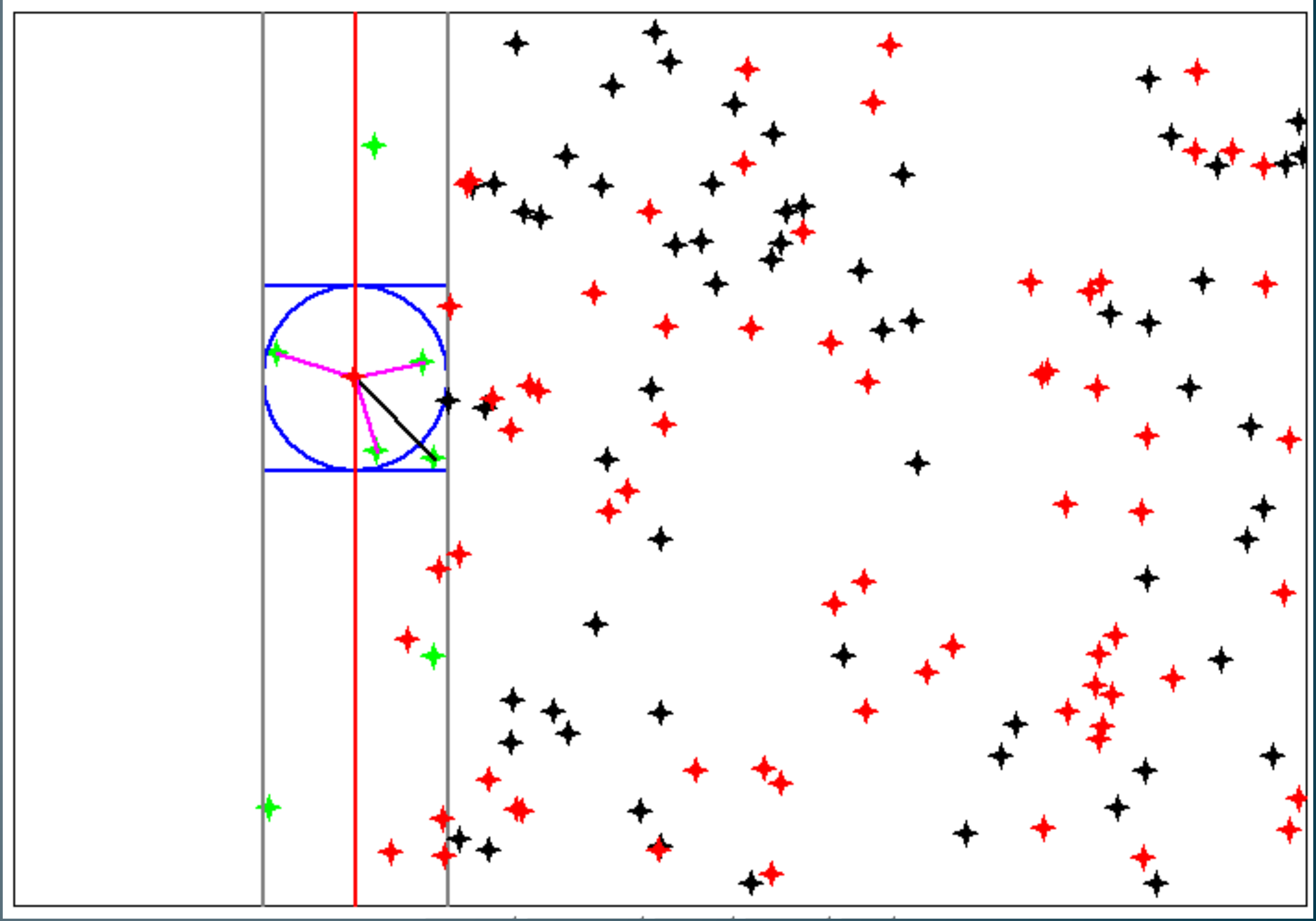


Middleware based option

- Plane Sweep Algorithm
  - Line sweep based on declination
  - Geometric operations
  - Complete once sweep is finished

- Pre-processing
  - Sorting the objects based on declination
  - Active list – small enough to reside in memory

Cross match between two collection of points

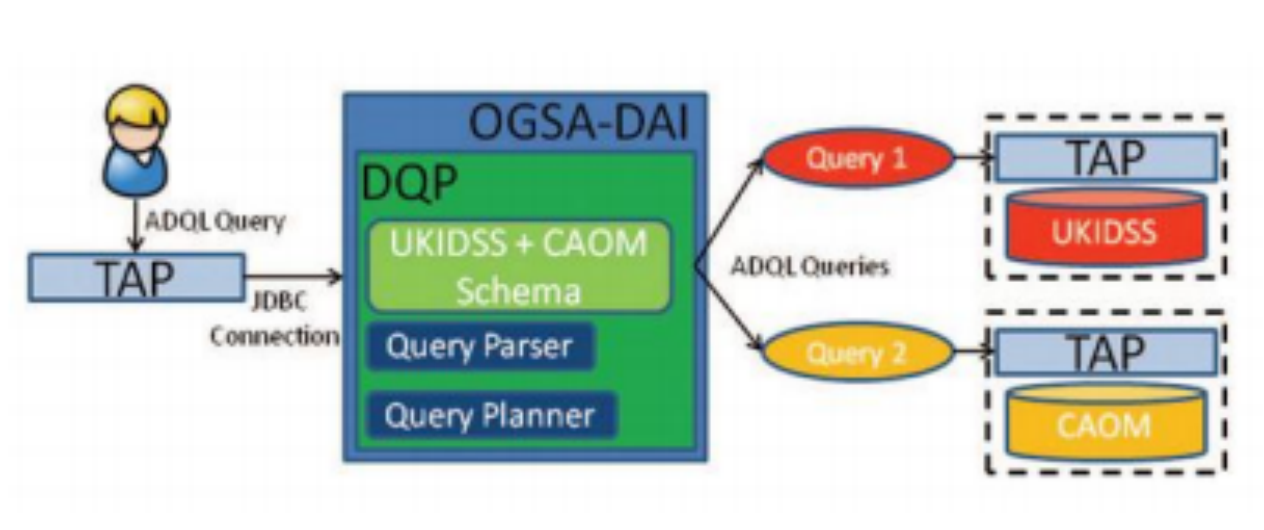


Plane Sweep Algorithm

- Advantages
    - Faster & efficient ~ O(N logM)
    - Low memory footprint
- Disadvantages
    - Inefficient - (large, small) catalogue pairs
    - Costs
        - Computational costs
        - I/O costs in data export & ingestion - plain text/binary files.

Middleware ?

- OGSA-DAI
  - middleware suite to perform queries between distributed catalogues using standards compliant services (TAP, VOTable etc)
  - data from multiple sources and of multiple types can be accessed, updated, combined, filtered, transformed and delivered.
- 3-layer service architecture for VO



Implementation

- Plane sweep implemented as activity in OGSA-DAI
  - Developed by EPCC

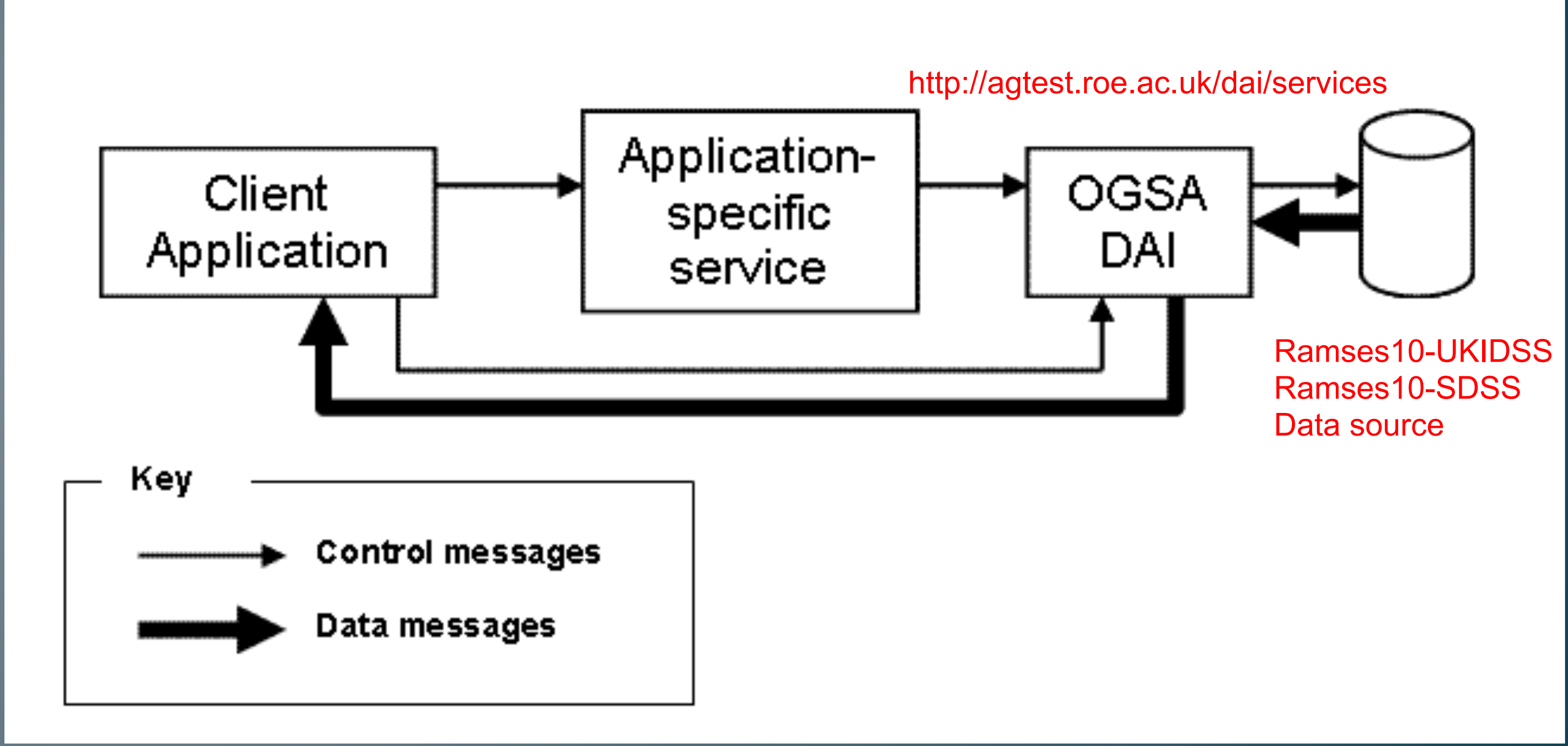
- Each kind of query -> separate client application

- Client application
  - Starts a tuple sweep activity
  - Streams tuples from participating tables wrt window size
  - Results streamed back from server to client

```
TupleSweepProduct product = new TupleSweepProduct();

product.addWindowSize(10/3600.0);

product.connectData1Input(query1.getDataOutput());
product.connectData2Input(query2.getDataOutput());
```



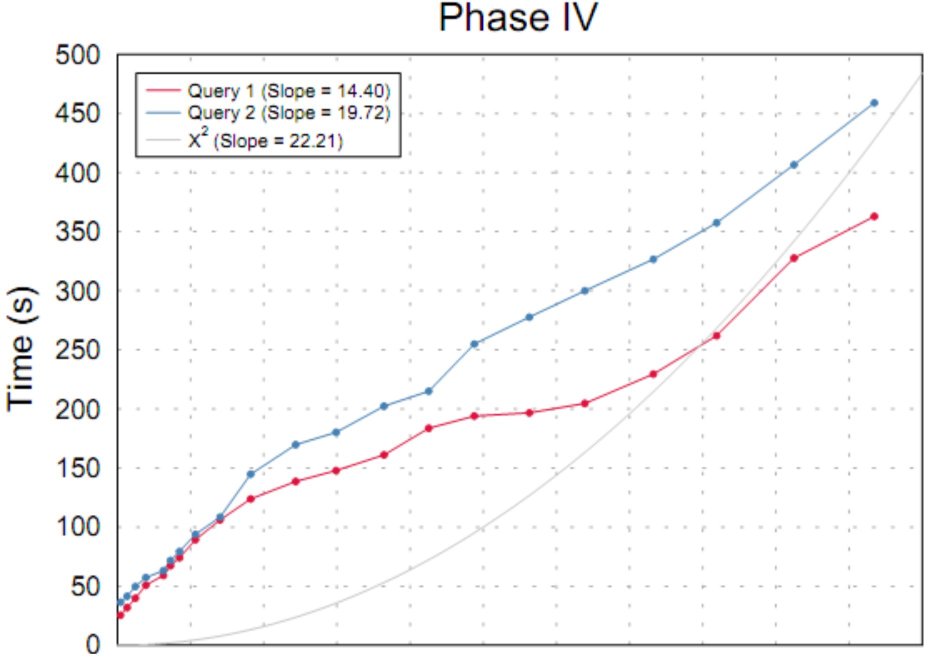
http://agtest.roe.ac.uk/dai/services

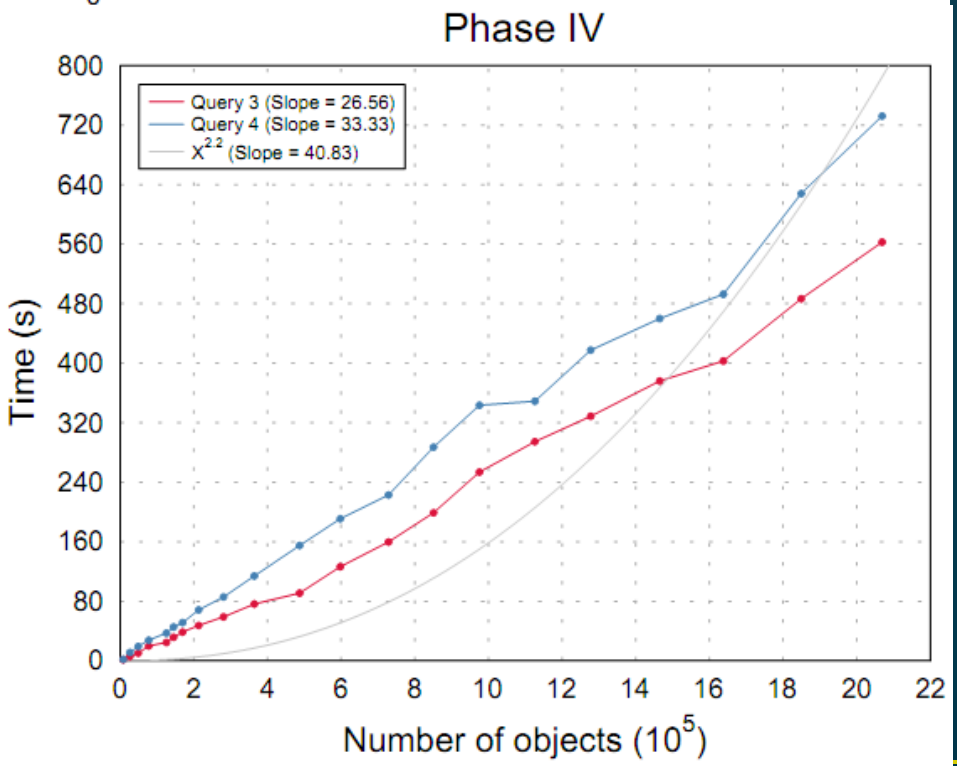
Ramses10-UKIDSS
Ramses10-SDSS
Data source

Projection criteria → Selection criteria → Workflow → Data source → Stream results

Results

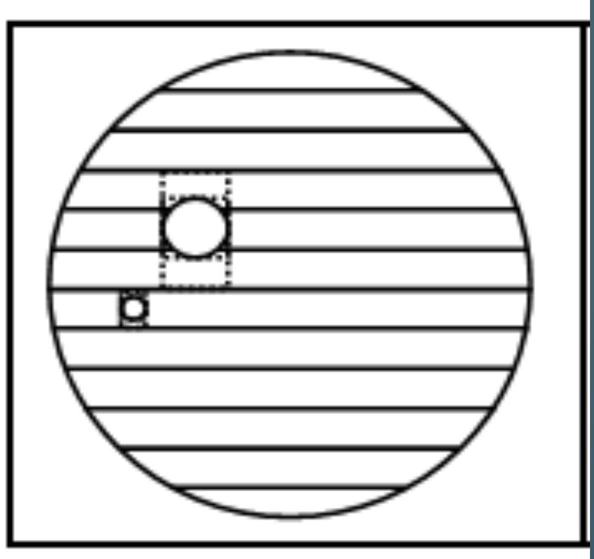
Criteria
Query times
Attributes
Scalability
* Network costs playe





Mixed option – Zones algorithm

- Bucketing the 2D spaces
- Celestial sphere division into zones
- Zone → declination cut on the sphere

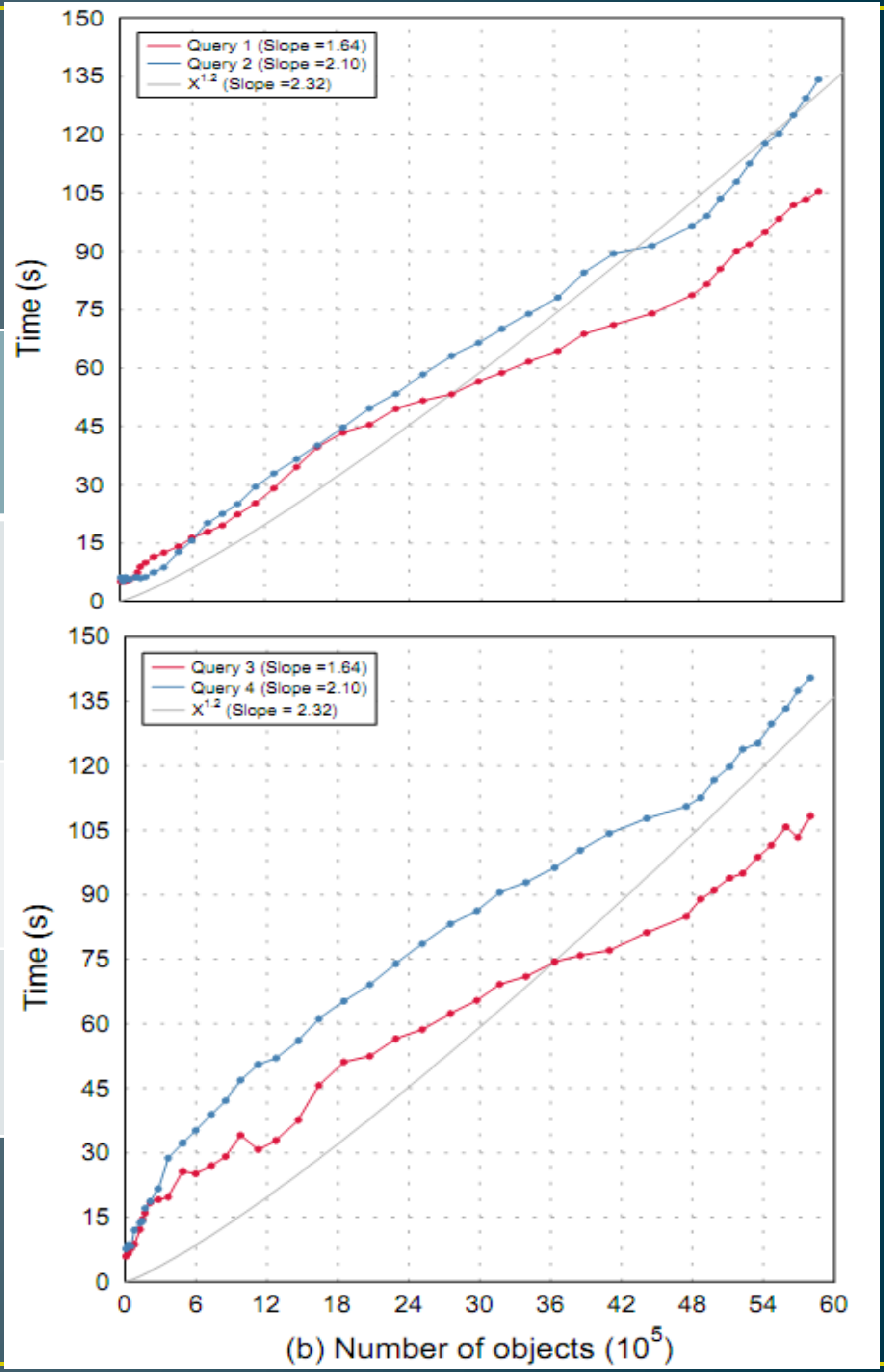


Mixed option – Zones algorithm

- Advantages
  - Good for point-near-point queries
  - Portable due to external implementation in SQL
  - Batch oriented spatial join queries – excellent performance
- Disadvantages
  - When the radius is 10 times more than the zone height, the algorithm tends to be inefficient

Results

Criteria
Query times
Attributes
Scalability



Results – additional costs

- Building the zone index table
- Building LAS/SDSS cross/self match table
- Addition of the clustered index
- Memory footprint of the generated table

  - Table construction times very high

Impact of Network related costs

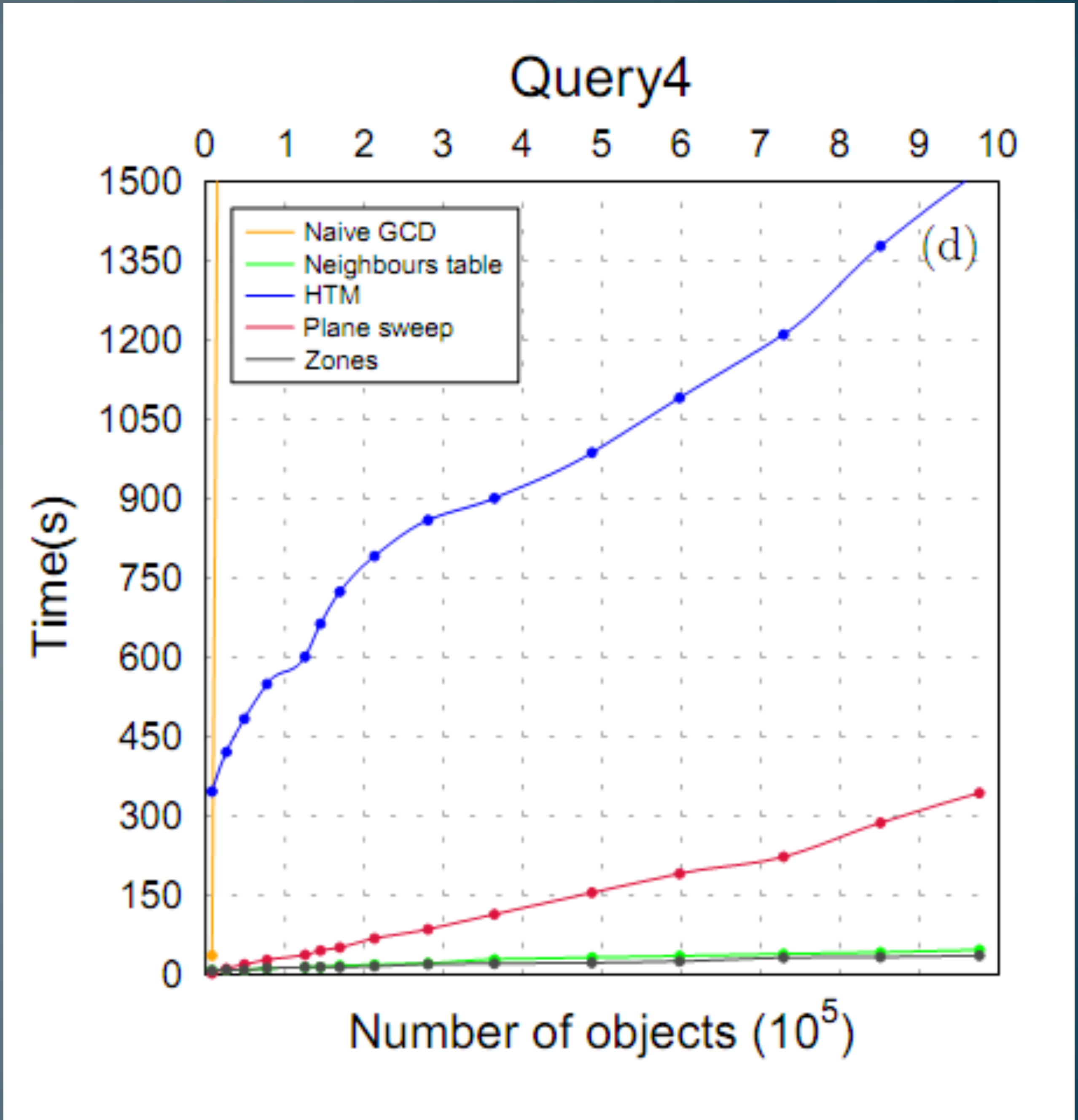
- Exact estimation – very complex
  - Network bandwidth
  - Topology
  - Network loads
  - Network environment

- Sample experiment
  - Query times differ with time and location

- Naïve estimates justify the magnified scalability factor in plane sweep approach

Outline

- Background
- Rationale
- Project Management
- Requirements / Environment
- Design
- Approaches
  - Implementation
  - Analysis
- **Discussion**
- Summary


Query4
0
1
2
3
4
5
6
7
8
9
10
1500
1350
1200
1050
900
750
600
450
300
150
0
Time(s)
Number of objects (10^5)
Naive GCD
Neighbours table
HTM
Plane sweep
Zones
(d)

Inferences

- Naïve GCD approach
  - Best approach for very small subsets of objects

- Pre-computed neighbours table
  - Depends on flexibility of end user
  - Steps - non automated
  - Burden of maintenance
  - High performance gains

Inferences

- HTM Approach
  - Concise judgement on level of recursion
  - Extremely fast cone searches (w.r.t cross match)
  - Rather diminished performance for range queries
- Plane sweep using OGSA-DAI
  - Major impact with respect to network related costs
  - Very low computation costs
  - Worthwhile in distributed environment.

Inferences

- Zones approach
  - Excellent performance
  - Automatable process of table construction
  - Higher times for construction
  - Regular maintenance (updates)
- Indexed vs. Un-indexed
  - In each case,
    - Queries with un-indexed attributed take up higher times as expected.

Future work

- Larger set of queries

- Comparison of physical I/O statistics
  - In depth network related costs estimation

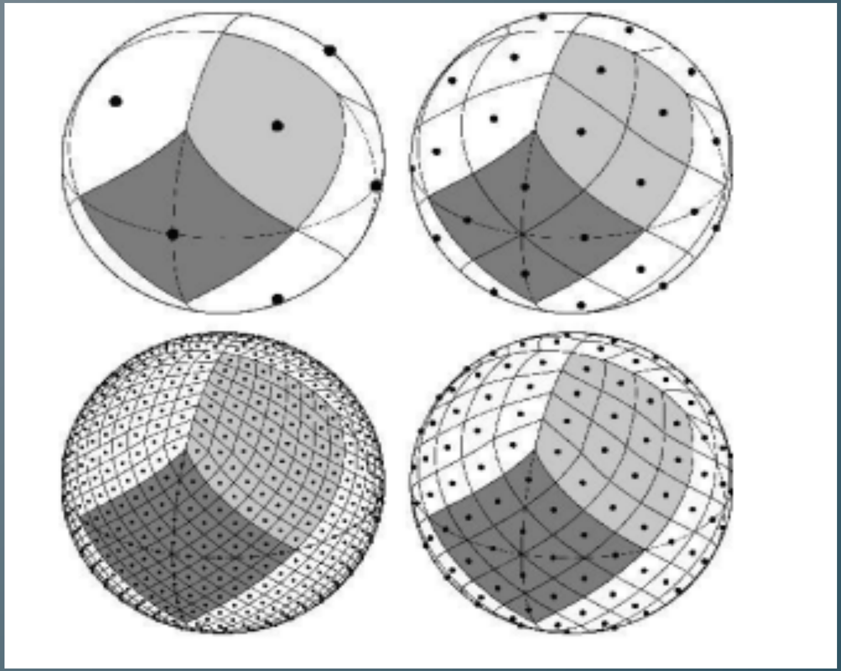
- Behaviour w.r.t different kinds of RDBMS

- Expand the approaches to HEALPix, advanced HTM indexing strategy etc.

Thank you

Hierarchical Equal Area isoLatitude Pixelisation (HEALPix)

- Mapping sphere to 12 diamond shaped entities
- Binary division of diamond entities into pixels/plane



Hierarchical Equal Area isoLatitude Pixelisation (HEALPix)

- Advantages
  - Operations are exact
  - The level of accuracy offered is very high
  - Offers fast numerical analysis & synthesis of distributed data
- Disadvantages
  - High complexity involved with implementation

Implementation ?

- HTM approach in IRP?

  - HTM approach has been well developed & documented
  - Integrates well with the SQL Server
  - Short time span available for the whole project
    - lot of effort on fairly less used & complex HEALPix approach

  - Scope for implementation ?
    - Based on time availability