Supporting Trajectory UDF Queries and Indexes on PostGIS

Pyoung Woo Yang* and Kwang Woo Nam**

*Turbo Soft Inc. and **Kunsan National University

kwnam@kunsan.ac.kr

Introduction

In this paper, we propose a system model for querying and indexing the GPS trajectory of moving objects on PostGIS/PostgreSQL.

We developed moving object data types including moving point, MDouble for GPS trajectories. Also, various moving objects UDFs are implemented for moving objects queries. For efficient query processing, r-tree index is extended for trajectory, and pre-materialization techniques are proposed for fast UDF processing.

Experimental results show that the pre-materialization techniques are about 1.2 times faster than naïve query processing using r-tree index.

Related Work

UDF(User Defined Function) approach have been used widely for developing advanced applications[1]. But, trajectory DBMS is not popular because of performance optimization [2].

Trajectories DBMSs

- DOMINO[3]
- HERMES[4]
- Trajectory Functions of PostGIS[5]

Trajectory UDF Queries by Examples

Creation of Trajectory Table

```
CREATE TABLE taxi
( taxi_id int, taxi_number char(20),
  taxi_model char(20), taxi_driver char(20) );

SELECT addtrajectorycolumn('public', 'taxi', 'traj', 4326, 'MOVINGPOINT', 2, 150);
```

Appending/Updating Trajectories

Inserting Moving Objects

```
insert into taxi values(1, '57NU2001', 'Optima', 'hongkd7');
insert into taxi values(2, '57NU2002', 'SonataYF', 'hongkd7');
```

Appending and Updating GPS Trjaectories

```
UPDATE taxi
SET traj = append(traj, 'MPOINT( 100 100 5000, 150 150
5001)') WHERE taxi_id = 1;

UPDATE taxi
SET traj = remove(traj, 'PERIOD( 5001, 5003)' )
WHERE taxi_id = 1;
```

Retrieving Trajectories

Spatial and Temporal Slicing

```
SELECT slice( traj, TIMESTAMP '2011-02-20 17:13:07', TIMESTAMP '2011-02-20 17:26:07')
FROM taxi
WHERE tj_OVERLAP( slice(traj, GeomFromText('POLYGON(15000 18000, 30000 30000, 15000 18000)')),
        'PERIOD( 5003, 5009 )');
```

Trajectory Predicates

```
SELECT taxi_id, tj_enter( traj, st_geometry('BOX(17 6, 29 31)'::box2d)
FROM taxi
WHERE tj_enter( traj, GeomFromText('POLYGON(15000 18000, 30000 30000,
        15000 18000)'), 'PERIOD( 5003, 5006 )');
```

Trajectory Functions

```
SELECT taxi_id, taxi_number, TJ_Traj(traj)
FROM taxi
WHERE TJ_Passes(traj, TJ_BOX(116.35, 39.93, 116.22, 40.14,
      PERIODS('2008-02-02 13:30:44', '2008-02-02 15:54:46')));

SELECT count(*)
FROM taxi
WHERE TJ_Inside(traj, TJ_BOX(116.35, 39.93, 116.22, 40.14,
      PERIODS('2008-02-02 13:30:44', '2008-02-02 15:54:46')));

SELECT taxi_id, taxi_number
FROM taxi
WHERE TJ_Cross(traj, taxi_id, TJ_BOX(116.35, 39.93, 116.22, 40.14,
      PERIODS('2008-02-02 13:30:44', '2008-02-02 15:54:46')));
```

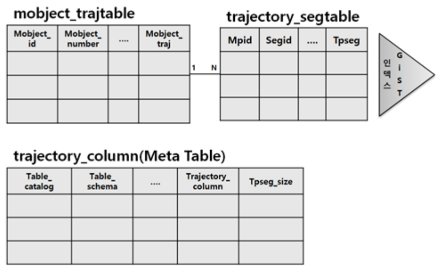
MDouble Functions

```
SELECT taxi_id, m_distance(traj, GeomFromText('Point( 50 50 )') ),
       m_mindistance(traj, GeomFromText('Point( 50 50 )') ),
       m_maxdistance(traj, GeomFromText('Point( 50 50 )') )
FROM taxi;

SELECT taxi_id, slice( traj, TIMESTAMP '2011-02-20 17:13:07', TIMESTAMP '2011-02-20 17:26:07' )
FROM taxi
WHERE m_mindistance( slice( traj, TIMESTAMP '2011-02-20 17:13:07', TIMESTAMP '2011-02-20 17:26:07'),
      GeomFromText('Point( 50 50 )') < 50;
```

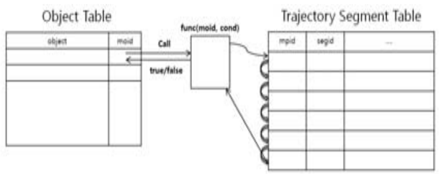
Trajectory Data Model on PostGIS

Trajectories are split into trajectory segment tables. And, metadata for trajectory objects are stored in a trajectory_column table.

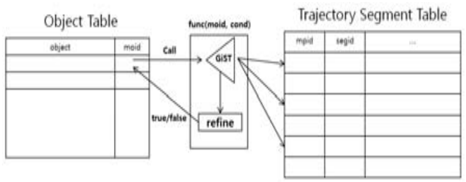


Performance Optimization by Query Materialization

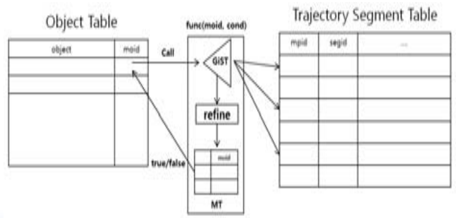
Naïve Approach : No Optimization



GiST Index Extension Approach for Trajectories

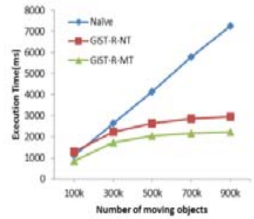


Query Materialization Approach

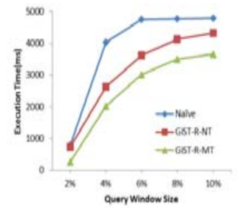


Performance Evaluation

- Time by number of trajectories



- Time by query window size



Conclusion and References

PostTrajectory : We developed a new system supporting trajectory queries on PostGIS using UDFs.

[1] C. Ordonez, and C. Garcia-Alvarado, "A data mining system based on SQL queries and UDFs for relational databases," in Proceedings of the 20th ACM Conference on Information and Knowledge Management, 2011, pp. 2521-2524.

[2] O. Wolfson, B. Xu, S. Chamberlain, and L. Jiang, "Moving objects databases: Issues and solutions," in Proceedings of the 10th International Conference on Scientific and Statistical Database Management, 1998, pp. 111–22.

[3] O. Wolfson, A.P. Sistla, B. Xu, S.J. Zhou, S. Chamberlain, "DOMINO: databases for moving objects tracking," in Proceedings of the SIGMOD International Conference on Management of Data, 1999, pp. 547–549.

[4] N. Pelekis, Y. Theodoridis, S. Vosinakis, T. Panayiotopoulos, "Hermes - A Framework for Location-Based Data Management," in Proceedings of EDBT, 2006, pp. 1130-1134.

[5] https://postgis.net/docs/reference.html#Temporal

ACKNOWLEDGEMENT

This research was supported by a Grant (14NSIP-B080144-01) from National Land Space Information Research Program funded by Ministry of Land, Infrastructure and Transport of Korean government.

PostTrajectory Project

http://github.com/awarematics/posttrajectory