Experimental Setup:

Choice of the application: In order to test our replacement model for mobile environment, we conduct experiments in a controlled, simulated environment. The choice of our representative application is that of a mobile navigation system where a user is moving in a car from one location to another, following a predetermined path. It is a client server environment where the client (the user) requests information from the server through the navigation system present in the vehicle. The location of the server(s) is unknown to the client, but it is assumed that the client will have some wireless connectivity via which it will communicate with the nearest server(s) along the way. During the journey, the user keeps asking queries pertaining to his need and/or interests. The queries range from finding McDonalds that is closest to the current location of the vehicle, to finding any gas station along the way where the gas price is the lowest. The user asks queries intermittently, switching between think time and ask time. The on board system has a limited amount of memory that can be used as a cache to store most frequently requested data. Note that the data requested by the user could be of spatial nature as well, such as maps, location information etc.

Other apps: fleet tracking, traffic management.

What is needed for performing an experiment? For conducting experiments in a simulated environment the following things are required. 1. Workload: Datasets and queries are together referred to as the workload. We can use either real datasets obtained from different sources like CITE, or artificially generate spatiotemporal data using a synthetic dataset generator. For our application real data sets are difficult to obtain and even those that can be obtained are too complex to work with. The complexity arises not only because of the huge size of datasets to be stored and manipulated, but also because of the complex structures of spatial objects. Also, in real datasets, parameters are more or less fixed depending on the real world behavior and it is extremely difficult to customize them. With the synthetic dataset generator it is easy to control various parameters related to the data and queries such as desirable cardinality, statistical distribution of various temporal and geometric features, etc.

For generating datasets, we use the existing synthetic dataset generator GSTD [PT00] for this purpose. GSTD generates sets of moving points/rectangular data that follow extended set of distributions (random/gaussian/skewed). Extensive sets of domain specific queries are created that are location aware, location dependent or simply non-location related. Queries to be executed are then picked up from these sets in a manner determined by parameters like the expected selectivity of the chosen queries and overlap rate. The specifications of these parameters are included in the query execution guidelines to be discussed in later sections.

2. Modeling the behavior of the moving objects: For our application we use a two dimensional space called the workspace. The workspace represents the area on which the mobile object moves from one location to another. Using GSTD we generate sets of static points and regions to emulate real life objects like rivers, buildings, highways, hotels, gas stations etc. These static objects are collectively called the infrastructure [?]. The motion of the mobile objects in the 2D space constrained by the infrastructure. With the help of GSTD we create trajectories to model the behavior of the mobile objects. Two points on the trajectory are chosen to represent the starting point and the destination for the mobile unit. A variety of parameters affect the behavior of the moving object. Prominent among them are the initial statistical distribution of the points and regions in the workspace, generation of trajectories, selection of source and destination points on the trajectory, direction of motion and speed of the vehicle. These parameters can be controlled during the simulation to create a realistic scenario.

3. Query execution guidelines:

The important factors that affect the entire experiment are the query execution quidelines. These are, the frequency of query issuance, that is the think time and the ask time. Selectivity of chosen queries, overlap rate and the type of queries asked (Location related, Location aware, general non location related queries.

---

output: a set of queries input: database tables, hotels, restaurants trajectory of moving object location of hotels and restaurants: infrasturcture

query config parameters type: LA, LD, NRL overlap rate frequency of issuance: think time ... direction of motion

hotel db

capacity range 0-1000 people tables 0-100 state NY, MD city zipcode enter NY and MD zipcodes (range)

Procedure:

1.GSTD – generates infrastructure .. hotels and thier location (unix-csv) 2.Random number generaotor generates names of hotels, number of hotels and their order. (win)

1 and 2 are merged and fed into the database using sqlldr. (win)

4.GSTD generates path of the moving object along with the time (x,y,t)

5.QueryGenerator.java generates queries from 3 input files LAqueryset, LDqueryset and NLRqueryset. the order, frequency and execution guidelines are capured in the properties.properties file.

6.QueryGenerator.java also contains an auxiliary module 'translate' to convert LD queries into LA queries. 'translate' takes the LA queries and the trajectory of the moving object, and replaces relative time/location references into absolute ones. It requires a file called TimeLoc (t,x,y) for mapping the coordinates of the moving object. We can extract this info from the database and store in the file TimeLoc.txt

final output file is GeneratedQueryLoad.txt that contains queries that are going to be issued by the moving object over its course of journey. time information is fed from the properties file.

Metrics :

What are you trying to prove ? what are the metrics used ?

Why GSTD ?

By providing the number of moving objects as an input to GSTD, it can generate a set of trajectories, one for each moving object. The motion of each object is simulated by specifying a number of control parameters. The direction, distribution can be specified. The algorithm generates a set of tuples of the form ¡x,y,t¿ where (x,y) gives the location of the object on a 2D plane at time t. The trajectory of the moving object is creates by joining consecutive points by linear interpolation. The algorithm also provides ways to create static rectangular regions of different sizes with different statistical distributions. These objects (collectively called infrastructure) can be view as bridges, hotels, buildings, road networks etc. to give a realistic view of the real-world landscape. The motion of the moving object is constrained by these infrastructure objects, just as in a real-world situation.

Advantages of using GSTD

Other alternatives

In [YW03] Wolfson and Yin introduce a method to generate synthetic spatio-temporal information called pseudo trajectories of moving objects. They obtain real speed variations by actually driving through the streets of Chicago noting the speed at regular intervals, and then superimpose these speed patterns on a randomly selected route to generate the trajectory of a moving object. Though simple and elegant, this method seems to be too restrictive and localized and may not be result in datasets that are truly representative of the real trajectories generated by moving objects.

With the City Simulator application [cit], it is possible to simulate the motion of upto 1 million objects moving in a city, driving on the streets, walking along the sideways and even walking up and down the floors of a building. It can even simulate traffic conditions on the road. The customizable parameters are number of objects in the experiment, entry and exit probabilities of these objects in the buildings, up/down movement of object in buildings and the scatter probability. However, since it is not possible to control individual speed or direction of a moving object, this method seems to be far less suitable to model mobile navigation application like ours. Secondly, although most of the parameters are configurable at the start of the experiment, they change when a certain threshold on the number of people in the building is reached. After that we do not have control over these parameters. This application is primarily designed for the evaluation of database algorithms for indexing as in LOCUS [MK02].

In [SM01], Salgio and Moreira come up with an interesting model to synthetically generate datasets. They contend that real-world objects do not move in a completely random fashion. The surrounding environment affects their actions and their behaviors are guided by some goals. On similar lines, they argue that generating completely random datasets and trajectories would not be a representative of real-world moving objects. As a modeling scenario, they use the example of fishing ships that go in the direction of the most attractive shoal of fishes while trying to avoid storm areas. Fishes are themselves attracted by plankton areas. Ships are moving points; planktons and storms are moving areas with fixed centers but variable shapes while shoals are moving regions. Although the approach is novel, the applicability of this model seems to

be limited to a few restricted scenarios where attraction and repulsion between various objects can be well defined and is meaningful. Also, in its current state, the model cannot be applied to road networks where most of the infrastructure like buildings, roads, rivers etc., is static.

1 Experimental Settings

For our experiments we need two things:
Dataset
Query workload

We assume that all objects move in a 2D space within the area defined by the user. Controllable parameters of the dataset are:

- Type of object: Point object.
- Cardinality of moving object: Since our main emphasis is on semantic caching, we select a single moving object for this purpose.
- Speed of moving object: We can specify a range for minimum and maximum speed.
- Direction of motion: We can chose the direction of motion (north/south/east/west/random).
- Initial position of the object.
- Number of snapshots: Number of observations that can be made during the run of the experiment.
- Frequency of updates: The frequency at which the observation regarding the moving object (current-location,current-time) is recorded.

Each value is calculated using a random generator.

Just for the sake of simplicity, we consider a single relation with atleast the following attributes defined: (Location-x, Location-y, Hotel-Name/School-Name, Hotel-ID/School-ID)
Query workload selection parameters are:

- Type of queries: Rectangular queries (Select and Project operations only)
- Location Dependence: Query predicates involving the current location of the moving object.
- Temporal/Prediction queries (Optional): Queries involving time as one of the predicates
- Selectivity of results: Number of tuples present in the result. We can place a max cap on the number of tuples fetched, depending on the size of the cache.
- Overlap rate: If there is no overlap between the queries (and hence their corresponding results), we would not derive any benefits from caching. We define the overlap rate to be the rate at which the queries overlap each other. We could set the value, for example 25%, meaning, on an average every fourth query overlaps with one or more previous queries.
- Think time: The time between two consecutive queries issued by the user. We set this value in a range from a minute to five minutes.

References

[cit]

[MK02] J. Myllymaki and J. Kaufman. Locus: A testbed for dynamic spatial indexing, 2002.

[PT00] D. Pfoser and Y. Theodoridis. Generating semantics-based trajectories of moving objects, 2000.

[SM01] Jean-Marc Saglio and Jose Moreira. Oporto: A realistic scenario generator for moving objects. *GeoInformatica*, 5(1):71–93, 2001.

[YW03] Huabei Yin and Ouri Wolfson. Accuracy and resource consumption in tracking moving objects: Symposium on spatio-temporal databases, santorini island, greece, 2003, 2003.