Data Management issues
In
Location Dependent Services
Next few slides...

- **Location Based Services**
- Challenges
- Why Caching?
- Terminologies
- Existing Caching Techniques
- Drawbacks
- Proposed system
Location- Dependent Information Services (LDIS)

Provide services based on the geographical location of users.
Application Opportunities

- Mapping, navigation, and directions applications
- Workforce-tracking and management applications
- Entertainment and gaming applications
- “Finder” applications - find services in the local environment
- Weather applications
- Reminder applications to prompt users when they reach particular locations
- Information services
  - stock quotes
  - nearest cash ATM
Enabling Technologies

- Wireless Networks
- Mobile Devices
- Positioning Technologies
- Database Systems
Next few slides

- Location Based Services
- **Challenges**
- Why Caching?
- Existing Caching Techniques
- Drawbacks
- Proposed system
What makes LDIS Challenging?

- User Movement
- Mobile Environment Constraints
- Spatial Property of Queries
User Mobility

- Continuous querying leads to high workload
- Results may be invalid due to high movement
## Mobile Environment Constraints

<table>
<thead>
<tr>
<th>Characteristic of Wireless Communication</th>
<th>Characteristic of Mobile Devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>✷ Frequent Disconnections</td>
<td>✷ Limited memory</td>
</tr>
<tr>
<td>✷ Expensive</td>
<td>✷ Limited computational power</td>
</tr>
<tr>
<td>✷ Relatively unreliable</td>
<td>✷ Small screen</td>
</tr>
<tr>
<td>✷ High bandwidth Variability</td>
<td>✷ Limited battery Life</td>
</tr>
<tr>
<td>✷ Scarcity of Bandwidth</td>
<td>✷ Variability in resources</td>
</tr>
<tr>
<td></td>
<td>✷ Frequent location updates</td>
</tr>
</tbody>
</table>
Location Dependent Data (LD)

- The class of data whose value is functionally dependent on location.
- Location determines the current value of the data.

Examples: City code

Location Independent Data (LID)

- The class of data whose value is functionally independent of location.
- Value of the location does not determine the value of the data.

Example: name, account number
Location Dependent Queries

- Data objects are associated with location
- Query predicates based on location
- Query result depends on location and time

E.g., Find the cheapest hotel here .... (Planar) Point query

E.g., Find all hotels within 5 miles..... Window query

E.g., Find the nearest hotel to me ...(K-)nearest-neighbor search
Query Types

- Mobile clients, querying static objects
- Stationary clients, querying moving objects
- Mobile clients, querying mobile objects
Differ from Temporal Queries

- Highly Dynamic
- Multiple answer sets depending on location/time
- The mobility of clients
- Location of client may be part of the query itself
- May depend on direction of movement
- The mobility of data objects
- Locations of data objects change (e.g., cars)
Next few slides

- Location Based Services
- Challenges
- Why Caching?
- Terminologies
- Existing Caching Techniques
- Drawbacks
- Proposed system
Why Caching?

- Improves data accessibility
- Minimizes access costs.
Cache Management

- Cache Prefetching
- Cache Invalidation
- Cache Replacement
Cache Prefetching

- The Data required is predicted based on the user demands and client’s movements.
- Cache repository will consist of data required for high update intensive transactions.
Cache inconsistency can occur due to temporal and location-dependent updates.

- Temporal-dependent invalidation
- Location-dependent invalidation
Cache Replacement...

✧ Process

✧ - Flush out cache chunks
✧ - Write back to the server or drop
✧ - Increase in communication overhead

✧ Objective

■ Provide a high hit rate
■ Perform as few replacements as possible.
Existing Cache Replacement Strategies

- Euclidean
- Manhattan
- Farthest Away Replacement (FAR)
- Probability Area Inverse Distance (PAID)
Drawbacks

- Rely on the temporal locality of client’s access pattern.
- Evict only the most unlikely data item from the cache.
Farthest Away Replacement (FAR)

M--very large
Probability Area Inverse Distance (PAID)

The Cost function $c_{i,j}$ for data value $j$ of item $i$ is defined as:

$$c_{i,j} = \frac{P_i \times A(v'_{i,j})}{D(v'_{i,j})}$$

Where

- $P_i$: Access probability of item $i$
- $A(v'_{i,j})$: Area of the attached valid scope $v'_{i,j}$
- $D(v'_{i,j})$: Distance between the current position and valid scope $v'_{i,j}$
DrawBacks of PAID

- Does not consider the randomness of user movement
- Does not consider data updates
- Does not give the user an option to choose his object of interest
Next few slides

- Location Based Services
- Challenges
- Why Caching?
- Terminologies
- Existing Caching Techniques
- Drawbacks
- Proposed system
Weighted Angular Distance Based Cache Replacement Strategy

- Temporal locality,
- Location of data,
- Location of the client
- Direction of the client’s movement
- User Preference

- Remove a set of data items and replace them with a corresponding number of data items which are more likely to be used.
Terminologies and Assumptions

**Geometric Model**
- 2D or 3-D coordinates,
- e.g., the latitude/ longitude pair
- Compatibility across heterogeneous systems

**Symbolic Model**
- Logical, real-world entities
- Bus stop number, train station name
- Coarser location granularity
- Suitable for LBS applications due to semantics

**Service Area** : the geographical area covered by the mobile system
Terminologies

Data instance
location-dependent queries return different results, each result is called a data instance

Valid scope
the area within which a data instance remains valid
Can be seen as an implicit property associated with a query result.

Spatial Scope
the area where the query result remains valid.

Temporal Scope
the period of time when the query result remains valid.
Contextual Attributes Scope for context-aware applications
e.g., temperature range
Valid Scope
The valid scope is defined as the region within which the item value is valid.

For Data Item i

<table>
<thead>
<tr>
<th>Item Value</th>
<th>Valid Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>OV(_1)</td>
<td>VS(_1)</td>
</tr>
<tr>
<td>OV(_2)</td>
<td>VS(_2)</td>
</tr>
<tr>
<td>OV(_3)</td>
<td>VS(_3)</td>
</tr>
<tr>
<td>OV(_4)</td>
<td>VS(_4)</td>
</tr>
<tr>
<td>OV(_5)</td>
<td>VS(_5)</td>
</tr>
<tr>
<td>OV(_6)</td>
<td>VS(_6)</td>
</tr>
</tbody>
</table>
... Valid Scope

- Refers to the geometric area of the valid scope of a data value.
- Indicator of the access probabilities for different data values.
- Larger the valid scope area of the data, higher the probability of access
Scope Distribution

It is the set of valid scopes for all items values of a data item.

For Data Item i, scope distribution $SD_i$:

$SD_i = \{VS_1, VS_2, VS_3, VS_4, VS_5, VS_6\}$
Voronoi Diagrams ...

The Voronoi diagram of a point set is the collection of regions that are closer to one point than to any other point.
...Voronoi Diagrams

- Area of valid Scope is generated based on the Voronoi diagram.

- A Voronoi diagram divides a space into disjoint polygons

- The nearest neighbor of any point inside a polygon is the generator of the polygon.
Voronoi Polygons

A set of limited number of points, called generator points, in the Euclidean plane
Associate all locations in the plane to their closest generator(s).
The set of locations assigned to each generator forms a region called Voronoi polygon or Voronoi cell, of that generator.

The set of Voronoi polygons associated with all the generators is called the Voronoi diagram with respect to the generators set.
The Voronoi polygons of a Voronoi diagram are collectively exhaustive because every location in the plane is associated with at least one generator.
The polygons are also mutually exclusive except for their boundaries.
The boundaries of the polygons, called Voronoi edges, are the set of locations that can be assigned to more than one generator. The Voronoi polygons that share the same edges are called adjacent polygons and their generators are called adjacent generators.
Voronoi

The Voronoi polygon and Voronoi diagram can be formally defined as:

\[ P = \{ p_1, ..., p_n \} \subset \mathbb{R}^2, \quad 2 < n < \infty, \quad p_i \neq p_j \quad \text{for} \quad i \neq j, i, j \in \tilde{I}_n = \{1, ..., n\}. \]

The region given by

\[ VP(p_i) = \{ p \mid d(p, p_i) \leq d(p, p_j) \} \quad \text{for} \quad j \neq i, j \in I_n \]

where \( d(p; p_i) \) specifies the minimum distance between \( p \) and \( p_i \) is called the \textit{Voronoi Polygon} associated with \( p_i \), and the set given by:

\[ VD(P) = \{ VP(p_1), ..., VP(p_n) \} \]
Properties

- Property 1:
  - The Voronoi diagram of a point set $P$, $V D(P)$, is unique.

- Property 2:
  - The nearest generator point of $pi$ (e.g., $pj$) is among the generator points whose Voronoi polygons share similar Voronoi edges with $V P(pi)$. 
Weighted Angular Distance Based Cache Replacement Strategy

- Temporal locality,
- Location of data,
- Location of the client
- Direction of the client’s movement
- User Preference

- Remove a set of data items and replace them with a corresponding number of data items which are more likely to be used.
Cost Function

\[ c_{i,j} = \frac{P_i A_{i,j}}{\alpha_d D(V_{i,j}) + \alpha_\theta \theta(V_{i,j})} \times \frac{1}{v} \]

- \( P \) - access probability,
- \( A \) - valid scope area,
- \( D \) - beeline distance
- \( \theta \) - angle of bearing,
- \( \alpha_d \) and \( \alpha_\theta \) - beeline and bearing coefficients
- \( v \) denotes the supplement variable.
Access Probability

- Probability of a particular data item being queried again within the same valid scope.
- Exponential aging method. (Two parameters are maintained for each data item i)
  - $(P_i)$ Running probability
  - $(t_{last, i})$ Time of the last access to the item

\[
P_i = \frac{\alpha}{(t_{current} - t_{last, i})} + (1 - \alpha)P_i
\]

- $t_{current}$ The current system time
- $\alpha$ a constant factor to weight the importance of the most recent access in the probability estimate
Angle of Bearing

The angular distance (in degrees), $\theta$ between the client’s current direction and direction vector of a particular valid scope.

$$c_{i,j} = \frac{P_i.A_{i,j}}{\alpha_d.D(V_{i,j}) + \alpha_\theta.\theta(V_{i,j})} \times \frac{1}{v}$$
Clients' current direction

Direction Vector

Valid scope area

$\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$, $\mathbf{v}_4$, $\mathbf{v}_5$, $\mathbf{v}_6$

$\mathbf{d}_1$, $\mathbf{d}_2$, $\mathbf{d}_3$, $\mathbf{d}_4$, $\mathbf{d}_5$, $\mathbf{d}_6$
The linear distance of a valid scope from the client’s current location. In a location-dependent data service,

\[ P_i = \frac{\alpha}{(t_{\text{current}} - t_{\text{last},i})} + (1 - \alpha)P_i \]
In location-dependent data services, the beeline distance may gain precedence over the angle of bearing or vice-versa depending on the need and type of service.

\[ c_{i,j} = \frac{P_i A_{i,j}}{\alpha_d D(V_{i,j}) + \alpha_\theta \theta(V_{i,j})} \times \frac{1}{v} \]
Supplement Variable

- Relevance of LDD changes dynamically
- V is a measure of data relevance
- Can signify any scenario being just a numerical value.
Flow of the proposed cache replacement policy

Service Provider DB

Access Probability
Valid Scope Area
Data Relevance

α₀ and αₜ values

Mobile Client

CACHE REPLACEMENT POLICY

{P, A, V, D, θ} data item

{Cost} data item

Rank Generation Algorithm

Rank List

Evicted Data Items

Selected Data Items

Cost
data item

Data Relevance

Access Probability
Valid Scope Area
Data Relevance

Service Provider DB
Algorithm

- **Input**: Rank list, list of current items in the cache
- **Output**: List of items in the cache
- **Procedure**:
  1. loop $i = 0$ to all items in the cache
     - Determine lowest ranked item in the cache, $C_{lowest}$
     - Compare $C_{lowest}$ with the highest ranked item in the rank list, $R_{highest}$.
     - **If** $\text{Rank}(C_{lowest}) < \text{Rank}(R_{highest})$ then
       - Replace $C_{lowest}$ with $R_{highest}$
       - Remove $R_{highest}$
     - **If** $\text{Rank}(C_{lowest}) \geq \text{Rank}(R_{highest})$ then
       - Exit
  - End loop
Scope Distributions

Scope distribution S1

Scope distribution S2
Effect of changing Query Interval

**Scope Distribution 1**

- Cache Hit Ratio vs. Query Interval
- LRU, FAR, PAID, WIDAAP

**Scope Distribution 2**

- Cache Hit Ratio vs. Query Interval
- LRU, FAR, PAID, WIDAAP
Effect of changing Moving Interval

Scope Distribution 1

Scope Distribution 2

Cache Hit Ratio

LRU
FAR
PAID
WIDAAP

Moving Interval

Cache Hit Ratio

LRU
FAR
PAID
WIDAAP
Effect of Cache Size

Scope Distribution 1

Scope Distribution 2
Thank You
Cache Replacement

- Cache Replacement Decisions

1. Distance between a client’s current location and the location of each cached data object
2. Movement direction of mobile clients and the object’s updates.

- The most common replacement policies are:
  - Extensions of traditional replacement techniques
  - Key-based techniques use priority factors
  - Function-based techniques use time factor