CareDB: A Context and Preference-Aware Location-Based Database

Justin J. Levandoski **Mohamed F. Mokbel** **Mohamed E. Khalefa**

Department of Computer Science and Engineering - University of Minnesota

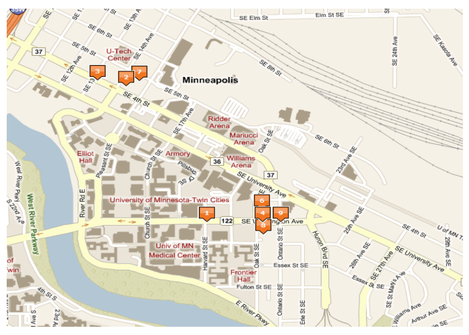
Overview

Need For Context and Preference in Databases

Consider a mobile location-based business finding application

"Find me a restaurant for dinner"

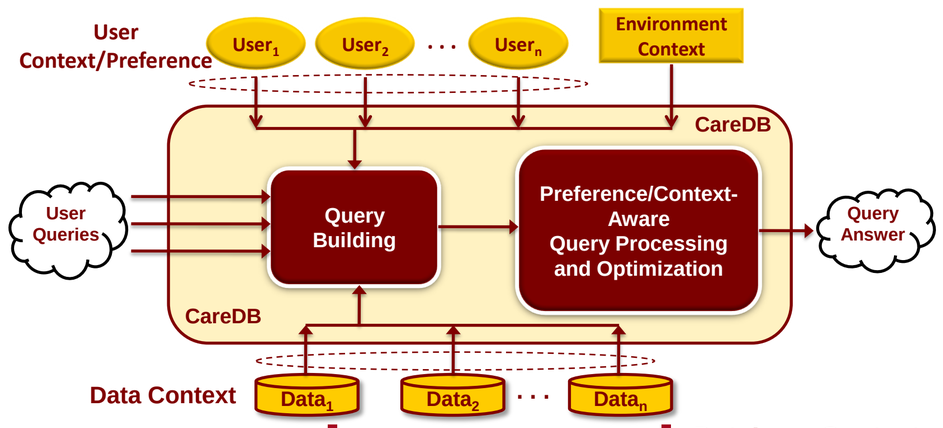
- **Existing applications return the K closest restaurants**
- **Consider five closest restaurants for dinner**
  - **Restaurant 1**
    - **Hour and a half wait**
  - **Restaurant 2**
    - **Does not meet my diet**
  - **Restaurant 3**
    - **Too expensive**
  - **Restaurant 4**
    - **Closed for remodeling**
  - **Restaurant 5**
    - **30 minute drive due to heavy traffic**



The five restaurants are NOT useful as database systems are detached from:

1. Personal preferences (dietary restrictions, budget)
2. Extra contextual data (time of day, traffic, waiting times)

CareDB Architecture



User Context/Preference
Traffic
User Status
User Salary
Budget
……

Environmental Context
Traffic
Weather
Road Network
Transportation
……

Database Context
Restaurant waiting time
Restaurant location
Restaurant rating
Restaurant price
……

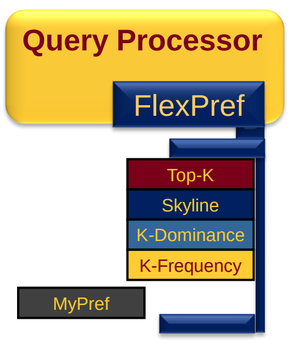
CareDB Functionality

Extensible Preference Query Processor

Extensible preference query processing support provided through FlexPref extensible preference query processing framework

1. **Define *two macros* and *three functions* in separate file outside CareDB/FlexPref**

2. Compile into CareDB/FlexPref using command:
`DefinePreference MyPref with MyPref.c`




```
#define DefaultScore
Default score assigned to each object
#define IsTransitive
Whether preference function is transitive or not
```

```
PairwiseCompare(Object P, Object Q)
INPUT:  Two objects P and Q
ACTION: Update the score of P
RETURN: 1 if Q can never be a preferred object
       -1 if P can never be a preferred object
        0 otherwise
```

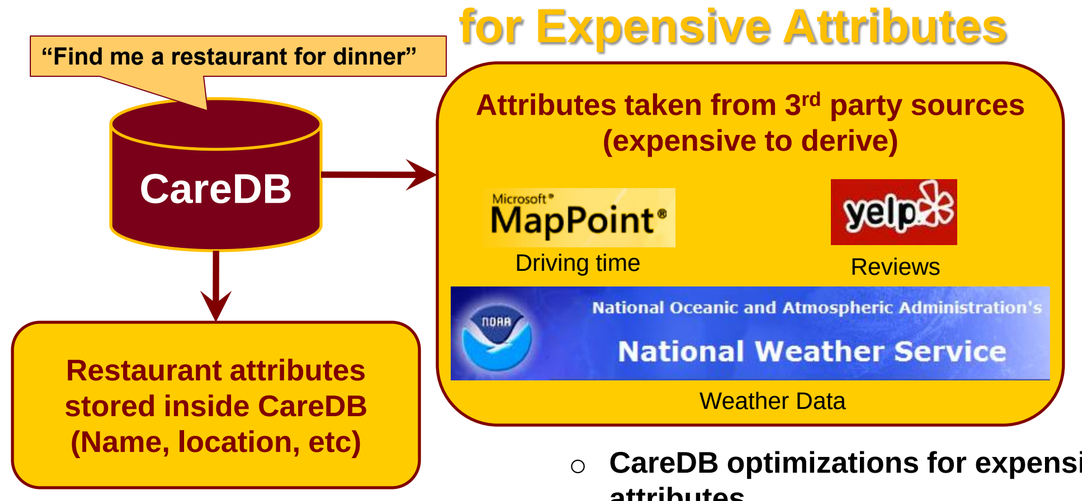
```
IsPreferredObject(Object P, PreferenceSet S)
INPUT:  A data object P and a set of preferred objects S
RETURN: True if P is a preferred object and can be added to S
        False otherwise
```

```
AddPreferredToSet(Object P, PreferenceSet S)
INPUT:  A data object P and a set of preferred objects S
ACTION: Add P to S and remove or rearrange objects from S
```

Writing Queries

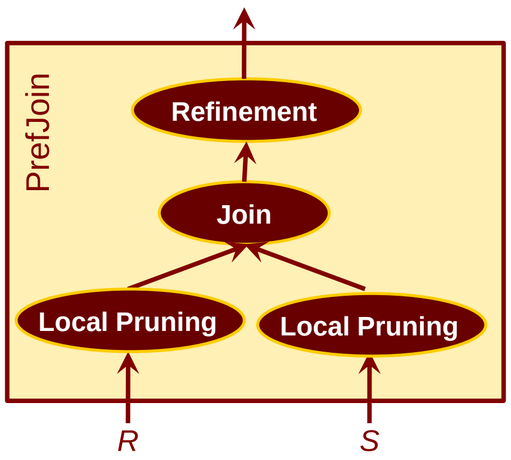
```
SELECT *
FROM Restaurants R
WHERE [Where_clause]
PREFERENCE OF [Attribute/Objectives]
WITH PREF METHOD MyPref
```

Efficient Query Processing for Expensive Attributes



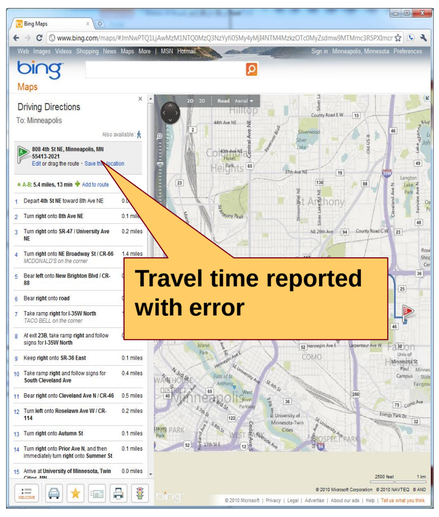
- **Cost model changes**
  - **Local data cheap to process relative to third-party contextual data**
  - **Must optimize to request least amount of data from third party**
- **CareDB optimizations for expensive attributes**
  - **Perform preference evaluation over local (cheap) attributes**
  - **Prune objects guaranteed to never be preference answers**
  - **Minimize subsequent expensive attribute requests for correct answer**

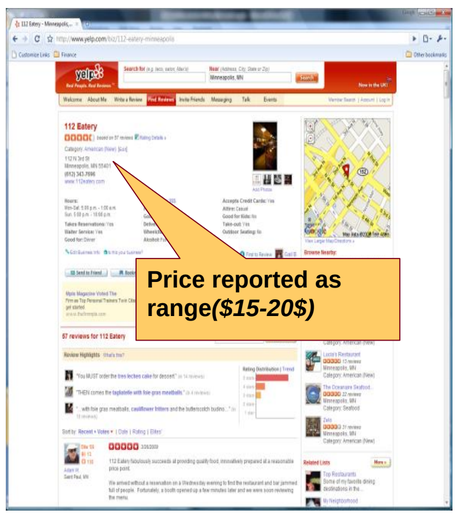
Efficient Generic Preference Join



- **Naïve join approach**
  - **Join all data**
  - **Then perform preference evaluation**
- **CareDB Join**
  - **Prune tuples from base relation that can never be preference answers**
  - **Refine join result to form correct preference answer**
  - **Generic to many preference methods**
  - **More efficient than naïve approach**

Preference Query Processing for Uncertain Data

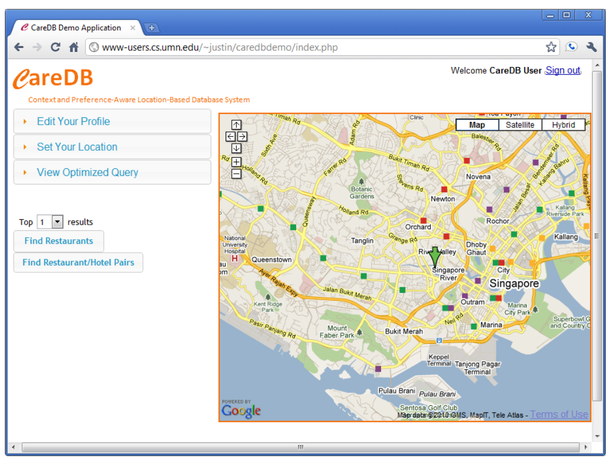




- **Uncertainty inherent in many contextual data sources**
  - **Restaurant prices reported as a range**
  - **Travel time reported with error**
- **CareDB supports preference query processing over uncertain data reported as range**
  - **Efficient two-phase filter/refine algorithm**
  - **Objects reported with probability of being preference answer**

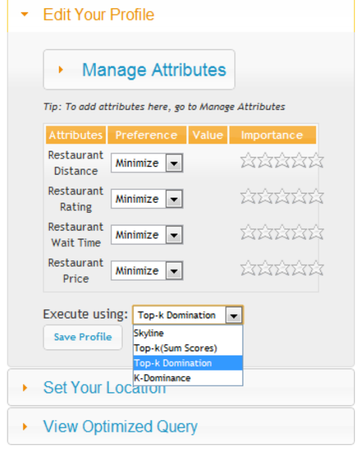
CareDB in Action

Environment



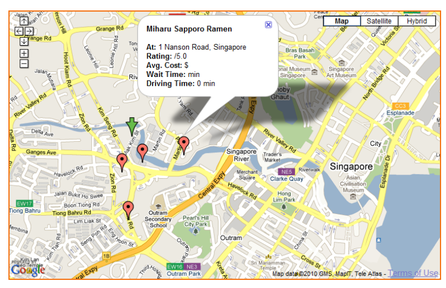
- **Mobile (iPhone) and web-based restaurant and hotel finder application implemented in Google Maps with real Singapore restaurant and hotel data (e.g., reviews)**
- **Data stored in underlying PostgreSQL database.**
- **CareDB implemented inside PostgreSQL engine.**

User Preference Profiles

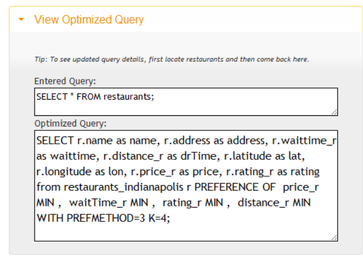


- **Users can log in and edit their preference profiles explicitly using attributes for restaurants and hotels**
- **Some contextual attributes are uncertain (reported as range), while others require expensive third-party access**

Querying CareDB



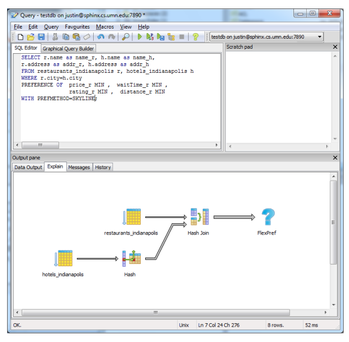
- **Users press "find restaurant/hotel" button.**
- **Answers displayed on Google Maps.**
- **Answers generated by CareDB executing in PostgreSQL**



- **Users can view SQL query extended with preference syntax that was built and executed in CareDB**

Extensible Preference Definition

- **Using the CareDB extensible preference query processing framework, attendees can add preference methods to CareDB and instantly use the method in preference queries**



- **Users can also issue ad-hoc queries in CareDB, and view query plans, using graphical client connected to CareDB**

*This work is supported in part by the National Science Foundation under Grants IIS-0811998, IIS-0811935, CNS-0708604, IIS-0952977 and by a Microsoft Research Gift