Towards a Location Dependent Benchmark: - Position Paper for NSF Workshop-

Margaret H. Dunham, Ayşe Yasemin Seydim *
Department of Computer Science and Engineering
Southern Methodist University
Dallas, TX 75275-0122
mhd(yasemin)@engr.smu.edu

1 Introduction

Benchmarking has long been a powerful tool for evaluation and comparison of computer systems. The benchmarking of database systems is recognized as extremely difficult due to the breadth of application domains, types of database systems, and hardware architectural alternatives involved. One of the first database benchmarks was the Wisconsin Benchmark which targeted relational query systems [1]. There have been many successors to this approach which improved on it. Currently accepted transaction processing benchmarks are based on determining the transaction per second processing rate for sets of predefined queries [2]. These are loosely based on debit and credit banking applications.

From our view point, the primary purpose of any database benchmarking tool is to determine reasonable ways to compare the performance of different implementations of the same set of queries. However, the queries are usually assumed to be domain specific. Typically the benchmark consists of three features:

1. **Queries:** There are typically simplistic versions of real life queries which could be executed.
2. **Data:** Data may be artificial data which is representative of the data to be used in that domain.
3. **Execution Guidelines:** The execution guidelines indicate specifically how the benchmark is to be executed, what performance metrics are to be used, and how these metrics are to be generated. These guidelines should be applicable for evaluating a real implementation, a testbed, a prototype, or even a simulation of a proposed implementation.

Existing benchmarking tools are simply not adequate for a mobile computing environment. The data and query and execution guidelines are not clear or applicable to a mobile computing environment. One reason is the difference in applications involved. In actuality, we are not sure any one really knows what mobile computing applications would be realistic to use as a model "typical application". We certainly don't think that debit credit banking applications are reasonable. The execution guidelines should include information about the architecture and connectivity involved. For example there should be some guidelines about disconnection rates.

*This material is based upon work supported by the National Science Foundation under Grant No. IIS-9979458

However, the major reason that we feel existing benchmarking tools are inadequate is because the mobility aspect is ignored completely. If a query is requested from a Mobile Unit (MU), how should the MU be moving?

When benchmarking location dependent queries, traditional benchmarks are even less suitable. Location Dependent Queries are quite different from traditional centralized and distributed queries. Certainly, debitcredit type queries are not applicable.

Due to the difference from traditional queries and the future importance of location dependent queries, we think that there is much need for benchmarks designed specifically for location dependent queries. A location dependent query should consist of four components:

1. **Queries:** Queries should be targeted to location dependent types of queries with results dependent on location of MU.

2. **Data:** Location dependent data which is typical of location dependent queries should be used.

3. **MU Behavior:** This includes the movement and connectivity of the MU.

4. **Execution Guidelines:** These are similar to traditional benchmarking guidelines, but should be targeted to the wireless environment with queries requested from an MU and executed at a fixed host server.

We briefly investigate some of the crucial benchmarking issues below.

Queries. There are three basic query types to be included [3]: a Location Aware Query (LAQ), a Location Dependent Query (LDQ), and a traditional query. An LAQ is one which includes any location related attributes in its predicates. An LDQ is one where the results depend on the MU location. Once the MU location is bound to a location, the LDQ can be viewed as an LAQ. Finally, a traditional query is one with no location attributes. The benchmark queries should be a mix of these three types. Included in them are the spatial operators to be used.

Data. It seems that the most cited types of location dependent queries involve hotel or restaurant information. Thus it would make sense to use this data as the benchmark data. A schema for the data would include: name of hotel/restaurant, address, city, state, zipcode, country, latitude/longitude, and other descriptive information. Here it is important to have different granularities of location included as we envision that different realizations of location dependent services would support different granularities.

MU Behavior. The MU Behavior is perhaps the most novel portion of the LDQ Benchmark. We envision that it will consist of at least two compoments. The first part contains the connectivity information. We should be able to model sleepers (MUs not connected often) and workaholics (those connected often). We also need to be able to model unplanned disconnections. The second portion of the behavior is the MU movement. Here we must be able to model the different ways in which MUs can move. There are three dimensions to the movement modeling: speed, direction, and movement pattern. We think that the following movement patterns will suffice:

1. **One Way:** The MU moves from point a to point b at a uniform speed.

2. **Round Trip:** The MU moves from point a to point b at a uniform speed, waits a certain time at point b, then moves from point b to point a at a uniform speed.

3. **Random:** The MU moves in a set of one way segments. There is a waiting time between the segment activations. The entire set of segments is the movement pattern. The ending point of segment i is the starting point of segment $i + 1$.

This movement assumes that points a and b are randomly generated in a two-dimensional space which is used as the benchmarking area.

Metrics. Another interesting issue to be examined are the metrics to be used. Simple transaction response time and throughput measures are not enough. Some sort of quality of service metric is needed. Here the quality is similar to that of precision and recall used in traditional Information Retrieval Systems. Imagine the response to a query "What are the closest hotels?" as being a set of hotels. Different implementations of the query may obtain different results. This could be based on the how or when the MU is bound to a location. It may be based on how "closest" is calculated and what location granularities are used.

2 Summary

We have argued the need for a benchmark targeting the location dependent query processing domain. We have also presented our preliminary views concerning the query, data, and MU Behavior components of such a benchmark.

References

[1] , D. Bitton, D. J. DeWitt, and C. Turbyfill, "Benchmarking Database Systems, A Systematic Approach", *Proceedings of the Ninth International Conference on Very Large Data Bases*, 1983.

[2] Jim Gray (Editor), *The Benchmark Handbook for Database and Transaction Processing Systems, Second Edition*, Morgan Kaufmann, 1993.

[3] Ayşe Y. Seydim, Margaret H. Dunham, and Vijay, Kumar, "Location Dependent Qeury Processing," *Proceedings of the Second ACM International Workshop on Data Engineering for Wireless and Mobile Access (MobiDE)*, 2001, pp 47-53.