Project Proposal for Lahar Backward Processing

Eugenia Gabrielova

ABSTRACT

Lahar is a system capable of processing event queries on Markovian (imprecise) streams. Backward query processing might save time by ignoring parts of a stream that can't satisfy a given query. This project will require reversing NFA event queries, tracking start and end times for event query matches, and implementing an optimization algorithm to help Lahar decide between forward and backward processing. Evaluation will determine what queries are most adaptable to backward processing, and how effective it is in practice.

1. INTRODUCTION

Markovian streams represent uncertain data streams produced from probabilistic models. Though many unique types of Markovian streams exist, some of the most common include location streams from RFID or GPS.

A Markovian stream representing a student's activities over the course of the afternoon will include a probability distribution for various locations at each time step - in class, at the gym, in their aapartment, at a café. The stream will also contain correlations between timesteps. The student is more likely to be in class at 10:01 if he is already in class at 10:00, but less likely to attend class if he is still in his apartment at 10:00 (perhaps asleep). This information may become exceptionally useful if it is necessary to retrace the student's steps, in the case of a lost item or a high risk situation on campus.

1.1 Motivation

Lahar is a data warehousing system that was created to analyze Markovian streams by processing real-time event queries and returning a set of likely outcomes. No system exists that can efficiently store and accurately query large amounts of uncertain data. Lahar offers significant technological advancement in a number of applications, including smart homes, theft detection, and hospital care.

1.2 Improving Query Processing

Recall the activities of the computer science student adventuring across campus. It might be useful to use Lahar to learn more about the location of the student throughout the day.

- How many times did Amanda go back to her room between classes?
- How many students went to the café on Monday evening?
- How long did Albert spend at the gym?

Recall the first example: *"How many times did Amanda go back to her room between classes?"* This information might be important to Amanda because she wants to make time for a jog or a lunch date, or maybe some time in the library. This information could be important to other people as well, such as Amanda's roommate. This query could be answered by counting instances in Amanda's daily stream where she *likely* passed between her dorm and the Computer Science building.

Lahar has the ability to process challenging queries, but the magnitude of the data it must analyze is immense. The quantity of deterministic streams in one Markovian stream can be exponential. Any optimization that narrows the field of possible results for a query, or enables Lahar to reach a conclusion more quickly, could improve the speed of the system. When indexes are available, it is possible to save time by looking only at the relevant portion of a stream. However, it is not realistic to assume that an index will always exist.

The queries that Lahar analyzes can be described in terms of predicates, such as "before-10-am" and "this-is-the-gym". If a predicate is rarely satisfiable in the data (suppose there is only one gym on campus), then it has high selectivity. The selectivity of a predicate is low if it is satisfied often in the data.

In Lahar, an event query that starts with a set of low-selectivity predicates may result in a number of partial query matches. If the end of the query contains any number of high-selectivity predicates, many of those partial matches will become dead ends. By processing a query in reverse - that is, by satisfying more unusual predicates first - Lahar will generate fewer dead end results for that query.

2. RESEARCH PLAN

To achieve this improved performance, the focus of this project is to improve Lahar's approach to event queries. In particular, the goal is to optimize how the Lahar system treats forward and backward processing, and to see if this capability improves performance on real data.

2.1 Tasks to Complete

Step 1: Determine how to reverse an NFA event query. The main task here is to develop an algorithm that, given an NFA *a*, can create an NFA *a'* such that processing *a'* backwards and *a* forwards on a Markovian stream produces equivalent event query matches. It is important to note that it is not sufficient to simply turn arrows around.

Step 2: Implement backward processing in Lahar. This should be able to process a reversed NFA, and be able to track both start and end times for each event query match (rather than just beginning timestamps).

Step 3: Implement a forward versus backward processing query optimizer for Lahar, based on a particular query and a particular stream. If indexes are available, the two processes may perform very similarly. If no indexes are available, forward and backward processing could have extraordinarily different performance.

2.2 Evaluation

In order to evaluate the effectiveness of optimized forward and backward processing in Lahar, the project will culminate with an analysis based on real-world data in the RFID domain, obtained from data collected throughout the Paul Allen Center.

- Types of queries in which backward processing makes a difference
- When backward processing makes a difference, how significant it is
- Difference in Lahar performance between using forward processing only, versus both backward and forward processing.
- Practicality of using backward processing with real-life NFA queries.

2.3 Timeline

- Friday, July 17: Deadline for an initial algorithm for backward stream processing
- Friday, July 24: Final draft of mid-point paper due. By this point, major issues anticipated should be identified, along with a concrete plan for experimentation. Implementation of reverse NFA should be complete by this point, as well as an initial attempt at an optimizer.
- Wednesday, August 5: Implementation of optimizer complete. Plan for testing should be ready to go, if not already started.
- August 9 to 14: Crunch time, and a practice talk.
- Monday, August 15th: Estimated completion date, final paper due, and a final talk.

2.4 Resources

In the event that there is time to develop follow-up algorithms, it might be helpful to have access to resources relating to algorithm analysis. Two textbooks in particular, Algorithm Design (Kleinberg and Tardos) and CLR - are exceptionally informative and available through WorldCat, if not here in the CSE department. These texts may also be useful for evaluation. Many valuable resources are available online (Java documentation, for example).

3. CONCLUSIONS

The ultimate goal of this project is to improve Lahar's speed, and to optimize the manner in which it handles diverse event queries. This could improve the performance of Lahar in various applications in the RFID domain.

4. ACKNOWLEDGMENTS

I would like to thank Julie Letchner and Dr. Magdalena Balazinska for their guidance and mentorship in the Database group at the University of Washington this summer. Thanks to the DREU program providing the opportunity for this research experience.

5. REFERENCES

[1] J. Letchner, C. Ré, M. Balazinska, and M. Philipose. LaharOLAP Demonstration: Warehousing Markovian Streams. In *35th International Conference on Very Large Databases*, 2009.

[2] J. Letchner, C. Ré, M. Balazinska, and M. Philipose. Access methods for markovian streams. In *25th International Conference on Data Engineering*, 2009.

[3] J. Letchner, C. Ré, M. Balazinska, and M. Philipose. LaharOLAP: Supporting OLAP queries on Markovian streams. Technical Report #CSE-09-03-03, University of Washington, March 2009.