CS525s – 2006 Elke A. Rundensteiner

Final Examination

Date: 7th December 2006
Allocated Time: Complete Class Period (in class exam)
Maximum Points: 50

STUDENT NAME: _____

Please place your signature below to confirm that you have completed this examination on your own without help from others, and using only material explicitly permitted for this examination.

SIGNATURE: _____

General Instructions:

- If a question seems vague, make reasonable assumptions, state those and answer the question under those assumptions. *Make sure to state any assumptions you make! Make sure to show your work!*

PROBLEMS:	MAX	YOUR SCORE
problem 1 (out-of-order-processing)	14	
problem 2 (sensor databases):	12	
problem 3 (query optimization):	12	
problem 4 (monotonic/query algebra):	12	
total:	50	

Problem 1: Out-of-order Stream Processing [14pts]

Most stream processing work assumes that all tuples arrive in monotonically increasing order of their timestamps.

- Why is this assumption made, i.e., which aspects of a stream processing system are simplified by this assumption? What happens if this assumption were made and relied upon in the system, yet in reality the tuples would be unexpectantly arriving out-of-order? (question by Andy)
- Describe three alternative methods of dealing with out-of-order tuples. Compare and contrast the three methods in terms of their pros and cons. Which method would you select ? (question by Andy)
- The Window-ID concept was introduced in the "Semantics And Evaluation Techniques For Window Aggregates In Data Streams" paper for groupby-with-aggregate queries. What are the uses of that method? Consider usage of the Window-ID concept when data arrives in order versus when data does not arrive in order. (question by Joshua)
- Discuss the application of this Window-ID method to join processing. Describe what, if anything, you would adjust about the join processing algorithms (say an MJOIN) to utilize the Window-ID method? Discuss the pros and cons of your solution. (question by Joshua)

Problem 2: Sensor Network Databases [12pts]

For the "TinyDB: an acquisitional query processing system for sensor networks" paper. TinyDB introduces a variety of different query rewriting and optimization techniques, in particular, exemplary aggregate pushdown, predicate re-ordering, and event query batching rewrites (Chapters 4.2 and 4.3). (derived from Andy's question).

- Explain the main ideas of these three optimizations, respectively.
- Which particular problems specific to the sensor network system are being tackled by each of these three optimizations, respectively?
- TinyDB seems to be focussing on processing of aggregation queries via their semantic routing trees. What about join queries? Sketch some challenges that may arise when trying to process joins inside sensor networks, and outline steps required towards solving these problems. (derived from Andy's question)

Problem 3: Adaptive Ordering of Pipelined Stream Filters [12pts]

In the paper on "Adaptive ordering of pipelined stream filters",

- What is the purpose of the greedy invariant in the A-greedy algorithm, and how does it work? List one concrete scenario (different from the paper) to show that greedy invariant is violated. What can you infer from it? (question by Mingzhu)
- Explain how and when the reoptimizer updates the view matrix. How complex in time is that update? (question by Mingzhu)
- Briefly sketch how query plan migration would work to achieve this plan reordering of operators (once a new optimal order has been identified by A-greedy optimizer).

Problem 4: Stream Algebra Query Operators [12pts]

Consider the following properties of query operators:
(a) stateless and non-blocking
(b) small bounded state and non-blocking
(c) unbounded-state and non-blocking
(d) stateless and blocking
(e) small bounded state and blocking
(f) unbounded-state and blocking

- Give an example of an (algebra) query operator for each of the above properties, if one exists. If not, indicate so as well.
- What does the notion of monotonic mean for stream queries? Give an example of a query operator that is monotonic, and one that is not. (derived from Sheng's question)
- What, if any, is the relationship between the concepts of "monotonic" and each of the 6 types listed above? For each of the query operators you picked to answer subquestion (4.1) above, indicate if it is monotonic or not.

.

Done with Exam Yet ? I hope it's not too too long ! ! ! But you guys had such good example exam questions ...