UNIVERSITÄT DES SAARLANDES
MAX-PLANCK-INSTITUT INFORMATIK
Dr.-Ing. Ralf Schenkel
Dr.-Ing. Marc Spaniol

Web Dynamics (SS 10)
Assignment 5

Handout on: June 17, 2010

Due on: June 24, 2010

Exercise 5.1: SOPT and POPT time-travel index lists

Build an inverted index for time-travel queries for the terms "time" and "Web" of the first ten slides of chapter 5 of the lecture (5-1 to 5-10). Assume that each slide $5 - x$ corresponds to a version of the same document that starts at time point x and is replaced at time point $x + 1$ by the next slide. Explain how an SOPT and a POPT index would look like, and what the space overhead of POPT and the performance overhead of SPOT are. Now assume that each slide $5 - x$ represents a different document that starts at time point x and lives for x time points. How does that change the picture?

Exercise 5.2: Coalescing

Apply the algorithm for approximate temporal coalescing on the sequence of scores from Figure 1, using $\epsilon = 1$.

Exercise 5.3: Time-travel interval queries

An important class of time-travel queries are interval queries of the form `"soccer world cup"@[01-JUN-2006,31-JUL-2006]`. Discuss how the scoring model introduced in the lecture for point-in-time queries can be adapted to interval queries, and if the SOPT inverted index and query processing technique (slide 5.12) can still be applied.

Exercise 5.4: Partition selection for interval queries

Consider again time-travel interval queries. A naive way to process them on a temporally partitioned index list would be to read all partitions that overlap with the query interval and process all entries from these partitions. However, sometimes it is possible to get the same (or at least almost the same) result with fewer partitions. Figure 2 shows an example for such a case. Which partitions can be left out here during processing?

Give a formal definition of partitions that can be left out during query processing (hint: start with quantifying the contribution of a partition to the set of results). Extend this definition to the case where retrieving only $\alpha\%$ of the results is acceptable. Give a formal definition of the partition selection problem. How could that be implemented without reading the partitions first, for example by using concise summaries of the versions in a partition (for example Bloom filters or KMV synopses)?

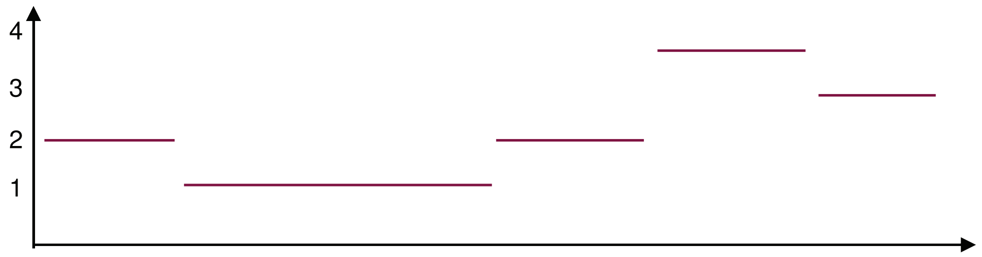

Figure 1: simple sequence of scores

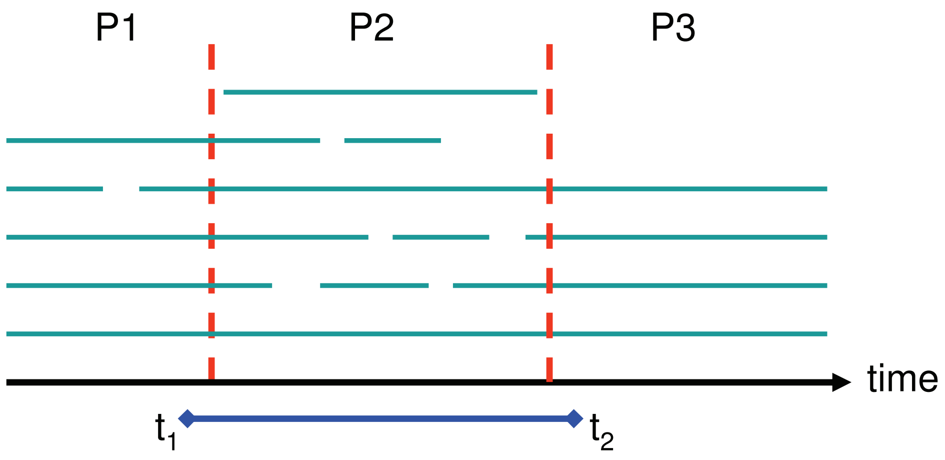

Figure 2: partitioned index list with three partitions