

Architecture and Implementation of Database Systems

Exercise Sheet 9

Marcos Antonio Vaz Salles

ETH Zurich

Exercise 1 – Inverted List Programming Exercise

Implement an inverted keyword list using a relational database, e.g. Oracle (or some other of your choice). Use the following schema:

```
INVERTED_LIST (KEYWORD, DOCUMENT_ID, TERM_FREQUENCY)
```

```
DOCUMENT (DOCUMENT_ID, PATH, LASTMODIFIED)
```

Offer a high-level keyword search interface in which users may input a keyword query Q and obtain as a response an iterator of documents I_D . Your interface should translate Q to SQL over the above schema. Assume that Q is of the form:

$$Q = k_1 \text{ [AND|OR] } k_2 \text{ [AND|OR] } \dots k_n$$

One example query would be $Q_1 = \text{jens dittrich architecture}$. The absence of a boolean operator indicates that AND semantics are expected. This means that only documents that contain all the three keywords should be returned in the response to Q_1 . The documents returned should be sorted according to the sum of the term frequencies for the keywords.

Load your inverted list with a test text collection and run queries against it. What indexes could you create on the above schema to speed-up query processing? How would you organize the schema physically? What is the performance when compared to a traditional inverted list implementation (e.g. Google Desktop)? What are possible explanations for the performance difference?

Challenge: Use as test data the current snapshot of the English Wikipedia! (http://en.wikipedia.org/wiki/Wikipedia:Database_download) Parse the XML files for the wiki using a SAX parser to obtain the pages and their respective text. Insert all the documents extracted into your inverted list.

Exercise 2 – Strict 2PL

Show how the following transaction histories would be handled by a scheduler based on strict 2PL. Add information to the transaction histories regarding:

- which locks are acquired (and when);
- whether any deadlock is incurred at any point;
- what is the waits-for graph for each deadlock situation;
- which transactions are aborted in the event of a deadlock;
- which transactions succeed.

(a) R1(A) R2(A) W1(A) W2(A) C1 C2

(b) W3(A) R1(A) W1(Z) R2(B) W2(Y) W3(B) C1 C2 C3