T–106.5241 Distributed Databases

Exam, May 10th, 2010

Write the following clearly on top of every paper you submit: "T–106.5241, May 10th, 2010", your full name, student ID and study programme, and the **total number of papers** you submit.

1. (6p) Explain the following concepts briefly:
   (a) horizontal partitioning
   (b) one-phase commit protocol
   (c) weak mutual consistency of replicas
   (d) semijoin
   (e) hash partitioning
   (f) page server (also known as: data server with page shipping)

2. (a) (4p) Describe the two-phase commit protocol that was used on the course (i.e., the version that is based on the presumed abort property). What messages are transferred between the different sites? What log records are produced? When is the log forced to disk? You can assume that the transaction commits successfully and that none of the sites crash.
   (b) (2p) In the two-phase commit protocol, there exists a situation where a cohort (=participant) may need to wait for a long time before it can either commit or abort its subtransaction. What is this situation?

3.(a–c) (3p) Are the following combinations possible under the quorum consensus protocol? n = number of replicas, p = read quorum, q = write quorum.
   (a) $n = 15, p = 1, q = 1$
   (b) $n = 15, p = 4, q = 12$
   (c) $n = 15, p = 8, q = 7$
   (d) (1p) In the parts (a) to (c) that were possible, how many of the sites containing replicas can crash, so that the quorum consensus protocol can still continue without needing to wait for the crashed sites?
   (e) (2p) Give an example of one-item deadlock in a replicated database.

4. How should the following queries be parallelized in a parallel database system? The values of attribute A are evenly distributed in the range [0..1000]. The relations $r(\underline{A}, B)$ and $s(\underline{A}, C)$ are range-partitioned (with the same ranges) to all processors. The final result of the query should be given to a single processor.
   (a) (1p) **select count(distinct** A**) from** r.
   (b) (2p) **select distinct** B **from** r **where** $A > 500$.
   (c) (1p) **select count(*) from** r, s **where** $r.A = s.A$ **and** $B < 20$ **and** $C < 50$.
   (d) (2p) **select distinct** C **from** r, s **where** $r.A = s.A$ **and** $B < 20$.

5. (a) (4p) Site s_1 of a distributed database crashes. At the time of the crash, the contents of the log on disk are the following:
   101: $\langle$*begin-checkpoint*$\rangle$
   102: $\langle$*transaction-table*, $\{\}\rangle$
   103: $\langle$*page-table*, $\{\}\rangle$
   104: $\langle$*end-checkpoint*$\rangle$
   105: $\langle T_1, s_0, T, B\rangle$
   106: $\langle T_1, W, p, i, x, \ldots\rangle$
   107: $\langle T_1', s_0, T', B\rangle$
   108: $\langle T_1', W, q, j, y, \ldots\rangle$
   109: $\langle T_1, \{s_0, s_1, s_2\}, P\rangle$
   110: $\langle S, B\rangle$
   111: $\langle S, W, r, k, z, \ldots\rangle$
   112: $\langle S, \{s_1, s_2\}, P\rangle$
   Restart recovery is performed using the ARIES algorithm. What are the contents of the reconstructed transaction table at the end of the analysis phase? What happens to the transactions T_1, T_1' ja S in the undo phase? Does the recovery algorithm need to acquire any locks in this situation?
   (b) (2p) In a page server (also known as data server with page shipping), what happens when the server s notices that a client c has crashed? What does c need to do when it later restarts?