INM370 – Advanced Databases
Tutorial 8 – Replicated Databases
Model Answers

This week's tutorial is based on the topic of replicated database systems.

Question 1. *Compare and contrast* ***eager*** *with* ***lazy*** *replication.*

We examined protocols for updating data that worked on the basis that all updates are carried out as part of the "enclosing", global transaction. This was necessary because a distributed transaction accesses different fragments on different sites; in other words, the updates are immediately applied at every site. Atomicity is ensured by using an atomic commitment protocol, e.g. 2PC (two-phase commit) protocol. The immediate propagation of updates in a replicated database is called eager or synchronous update propagation. Eager update propagation ensures that all copies are updated within the "enclosing", global transaction and voting at the end ensures the atomicity.

An alternative mechanism to eager replication is called lazy or asynchronous update propagation. With this mechanism, the target database is updated after the source database has been modified. The delay in regaining consistency may range from a few seconds to several hours or even days. However, the data eventually converges to the same value at all sites (eventual consistency). Although not all applications can cope with such a delay, it appears to be a practical compromise between data integrity and availability that may be more appropriate for organizations that are able to work with replicas that do not necessarily have to be always synchronized and current.

For further details, see relevant week 8 lecture slides (e.g. slide 13), and Section 26.2.4 from the Connolly and Begg book (6th Ed).

Question 2. *Describe different ways of implementing a* ***replication architecture****.*

There are two main architecture for replication: *kernel-based* and *middleware-based*. The latter can be implemented in 3 different ways:

- Kernel-based replication;
- Middleware-based replication;
    - Centralised MW-based replication
    - Replicated middleware-based replication;
    - Decentralized middleware-based replication.

For further details, see relevant week 8 lecture slides (e.g. slides 4-8), and Sections 26.2 to 26.2.2 from the Connolly and Begg book (6th Ed).

Question 3. *Discuss the functionality required of a replication server.*

At its basic level, we expect a distributed data replication service to be capable of copying data from one database to another, synchronously or asynchronously. However, there are many other functions that need to be provided, including:

- *Scalability* - The service should be able to replicate data across several computers, distributing the load among them; it should be able to replicate both small and large volumes of data.
- *Mapping and transformation* - The service should be able to handle replication across heterogeneous DBMSs and platforms. As was noted in Section 24.1.3 of the Connolly & Begg textbook, this may involve mapping and transforming the data from one data model into a different data model (e.g. relational to object-relational), or the data in one data type to a corresponding data type in another DBMS.
- *Object replication* - It should be possible to replicate objects other than data. For example, some systems allow indexes and stored procedures (or triggers) to be replicated.
- *Specification of replication schema* - The system should provide a mechanism to allow a privileged user to specify the data and objects to be replicated.
- *Subscription mechanism* - The system should provide a mechanism to allow a privileged user to subscribe to the data and objects available for replication.
- *Initialization mechanism* - The system should provide a mechanism to allow for the initialization of a target replica.
- *Easy administration* - It should be easy for the DBA to administer the system and to check the status and monitor the performance of the replication system components.

For further details, see relevant week 8 lecture slides (e.g. slides 9-10), and the relevant latter part of the Section 26.2.2 from the Connolly and Begg book (6th Ed).

Question 4. *List and discuss the main* ***replication schemes****.*

Based on the properties of i) update propagation – ***when*** and ii) update location – ***where***, the chosen replication scheme can be one of the following:

- Eager, primary copy;
- Eager, update anywhere;
- Lazy, primary copy;
- Lazy, update anywhere.
    - Note: Conflict detection performed based on version comparison, not value

comparison: Version vectors deal with versions, NOT values!

- An example of conflict detection is given in the dedicated sub-section – “Conflict detection using version vectors” - on p900 of the Connolly and Begg book, and the Fig. 26.14.

Please see relevant week 8 lecture slides – from 24 until the end, and the Section 26.3 from the Connolly and Begg book (6th Ed) for detailed discussion of the available replication scheme.