Running Head: OPTIMISTIC VS. PESSIMISTIC CONCURRENCY

Concurrency Control Types: Optimistic vs. Pessimistic Concurrency

Jered McClure

Walden University

Concurrency Control Types: Optimistic vs. Pessimistic Concurrency

The primary benefit of optimistic concurrency for a database warehouse is in its ability to allow multiple database connections on the same data. This means users or applications can access the warehouse in a timely manner without worrying about waiting for another person or application’s transaction to complete. However, the downside to this is that if a person or process must make a change to data, and their data change conflicts with another user or process’ change, then they must redo their entire transaction.

In OLTP, optimistic concurrency is essential to user access management. If a user makes a change to a piece of data, but leaves before committing the change or closing their session, their connection will simply time out without causing any sufferance to other users of the system. However, if their changes conflict with another user’s change, their transaction will likely need to be redone from the start, or at the very least resubmitted to the system.

On the other hand, pessimistic concurrency essentially assumes that there will be a conflict between users or applications trying to access data in the warehouse. Therefore, when a user or application accesses the warehouse, the data being transacted upon is locked so that no other person or process can access said data. This ensures that all data is consistent when the transaction is complete. That being said, this type of data locking can increase transaction time and processing overhead. Since all transactions are assumed to be in conflict, even simple select statements which cause no changes can lock data so that other users or applications cannot access data.

For an OLTP connection, pessimistic concurrency means users are guaranteed to have exclusive access to a transaction for the duration of their transaction. This means that they will not have to resubmit their system request. The downside to this is that if they would like to access a specific piece of data and another user already has that file locked, then they must wait until the previous transaction is complete. In some applications, such waiting can actually cause a drain on system resources since a connection will need to be maintained to monitor the locking sate of the data.

A great place to use optimistic concurrency for OLTP is in a business web portal. As stated above, if a user accesses a web form stored in the system’s database, but leaves before completing their change or closing their session, the transaction will not affect other users of the web portal who are trying to access the same form. In contrast, pessimistic concurrency would be called for in a loan decision application. A user accessing data on a loan can be assured that any changes that they make to the loan are consistent with what the system currently has stored.

In each concurrency control type, the primary consideration is the user/application. Customers affect concurrency in that they expect a system that is timely and does not cause undue frustration via multiple attempts to submit data to the system. Business users, on the other hand, expect accurate and concise data that is consistent with what is currently stored in the system. Computer applications are in a different arena altogether. Depending on the requirement of the application, optimistic or pessimistic concurrency can be allowed, so long as the application is built to handle each scenario accordingly.

Reference

Coronel, C., Morris, S., & Rob, P. (2012). *Database Systems: Design, Implementation, and Management* (10th ed.). Boston, MA, USA: Cengage Learning.

IBM. (2011, September 13). *Pessimistic vs. Optimistic Concurrency Control*. Retrieved January 8, 2013, from IBM Publib: http://publib.boulder.ibm.com/infocenter/soliddb/v6r3/index.jsp?topic=/com.ibm.swg.im.soliddb.sql.doc/doc/pessimistic.vs.optimistic.concurrency.control.html