

Mulgara - Feature #37

Lock-splitting required for MulgaraTransactionManager

12/04/2006 05:15 AM - Andrae Muys -

Status:	Closed	Start date:	
Priority:	High	Due date:	
Assignee:	Andrae Muys -	% Done:	0%
Category:	Mulgara	Estimated time:	0.00 hour
Target version:			
Resolution:	fixed		
Description While documenting the new transaction work I noticed that the read-only and read-write paths through the Manager share the same lock - the manager just uses synchronized methods (ie. is a monitor) . As read-write transactions can block for a substantial duration during a commit, while the block-cache is flushed to disk, they should not hold the same lock as the read-only methods require to initiate read-transactions while this is happening. The solution is to introduce two locks independent of the manager's monitor lock. 1. Covering alterations to the transaction metadata held in the manager, all of which are simple O(1) operations. 2. Covering alterations to the write-lock/writing-session metadata, which must be held over a commit(). Deadlock is not possible as the 2nd lock is only obtained by sessions wishing to obtain the write-lock. 			

History

#1 - 05/07/2007 04:25 AM - Andrae Muys -

Resolved in /trunk in revision 265

#2 - 10/10/2007 06:05 AM - Andrae Muys -

This did end up introducing a bug - which was logged and fixed as mgr-70.