Cache Coherency in Oracle Parallel Server

B.Klots(bklots@us.oracle.com)
Oracle Corporation

Oracle Parallel Server (OPS) is a relational database system running on a shared disk cluster. OPS allows a concurrent direct access of multiple users from multiple instances to all the data in the database. Shared disk architecture employed by Oracle has a number of advantages:

- High availability: when one node fails then other nodes can proceed uninterrupted, with all the data still available to them.
- Aggregate CPU and memory resources
- Enhanced throughput
- Load balancing

These advantages do not come for free. The challenge of this architecture (as of any clustered or distributed architecture) is to provide data coherency for the independent users of the system. The way to do that is to use locking. Oracle uses multiple level locking: row locks on transaction levels, instance locks within instances, and global locks among the instances. The latter are specific to Oracle Parallel Server.

In a nutshell the cache coherency protocol for Oracle Parallel Server is as follows. If a unit of data is being used at an instance and these data are requested at another instance, a conflict may occur. Global locks are used to resolve these conflicts. Before accessing the data unit instance acquires a lock on it. Another instance which wants to access same data unit asks for another lock on the data. This request can be either compatible or incompatible with the lock held by the first instance. If it is compatible (e.g. both instances want to read current data) the lock is granted to the requester and it proceeds with the operation. If the request is incompatible (e.g. the first instance writes and another instance wants also to modify the data) then the requester blocks. First instance is signaled with a request/order to finish its processing and flush the data to the shared disk storage. When it does so, it also releases the lock. Now the requester can be granted the requested lock, it reads the current copy of the block from the disk and proceeds. The global lock operations and all the communication involved in that are performed by Distributed Lock Manager (DLM).

In the talk we analyze two basic problems associated with the architecture of a locking scheme:

- Locking granularity. The trade off here ia as following: the more data is covered by a single lock, the more substantial is the amortization of locking performance overhead. On the other hand, increased granularity reduces the concurrency and consequently the global throughput of the system. A large granule size may lead to false conflicts. The proper optimization is a very challenging problem here.
- Dynamic versus static binding of data to locks. There are two basic schemes of matching data and locks in OPS: hash locking is a static locking scheme and fine grain locking is a dynamic locking scheme. These two schemes represent a special set of trade offs.. While hash locking is a good scheme for partitioned or DSS types of loads, the fine grain locking presents a good choice for OLTP.

Proceedings of the 22nd VLDB Conference
Mumbai(Bombay), India, 1996