Andrew Dawson
1220796
Lab 6 Write Up

SimpleDB is a relational database management system. That is recoverable and capable of running in parallel. The basic parts of SimpleDB are the BufferPool, the operators, locking/transaction support, logging/recovery support and support for running in parallel. The BufferPool is simply an in memory cache that holds recently used pages. The BufferPool is used by all the operators to get pages so that used pages are cached. The operators consist of Scanning, Joining, Filtering, Aggregating, Inserting and Deleting. The operators are all implemented as Iterators. The operators take some input (another operator) and produce output tuples from their child iterators. Chaining these operators together is what creates a query in SimpleDB. SimpleDB is capable of running transactions via its support for locking. Through implementing strict two phase locking simpleDB provides ACID properties for transactions. The locking is all taken care of by the LockManager class in conjunction with the BufferPool. In order for a transaction to operate on a page it must acquire the correct lock in BufferPool before it can operate on that page. When a transaction commits or aborts it releases all its locks - allowing other transactions to make progress. SimpleDB is also a recoverable system - if it crashes in the middle of execution it can recovery via an append only log that is flushed to disk. This log is used both to rollback aborted transactions as well as to restore the state of the database in the case of a crash. Lastly, SimpleDB also provides support for running in parallel via a Worker/Master relationship. The master node (Server class) receives a query and generates a non parallel query plan. Then it optimizes the query for parallel execution by replacing the operators with parallelized operators. Then the data (stored in Heap File) is partitioned across many worker nodes. Each of these worker nodes receives a query plan and a subsection of the data. The subsection of the data is stored locally on each worker's machine. The workers execute their query and send the results back to the Server.