RocksDB

Description:

The rocksdb library provides a persistent key value store. Keys and values are arbitrary byte arrays. The keys are ordered within the key value store according to a user-specified comparator function.

RocksDB borrows significant code from the open source [leveldb](https://code.google.com/p/leveldb/) project as well as significant ideas from [Apache HBase](http://hbase.apache.org/). The initial code was forked from open source leveldb 1.5. It also builds upon code and ideas that were developed at Facebook before RocksDB.

It works on 3 basic things:

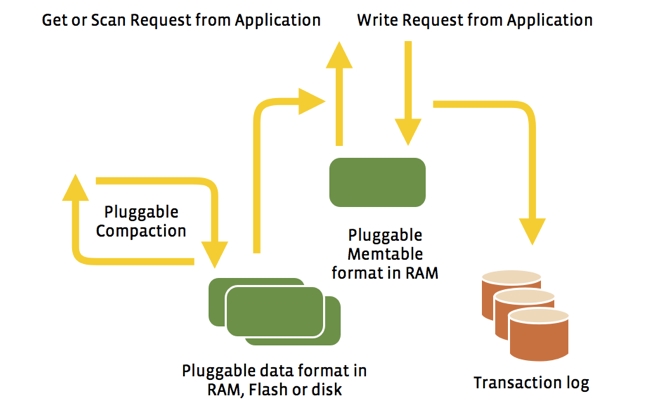
1. *memtable*

The *memtable* is an in-memory data structure - new writes are inserted into the *memtable* and are optionally written to the *logfile*.

2. *sstfile*

The data in an *sstfile* is sorted to facilitate easy lookup of keys.

3.*logfile*. The *logfile* is a sequentially-written file on storage. When the *memtable* fills up, it is flushed to a *sstfile* on storage and the corresponding *logfile* can be safely deleted.



**Note:**

The following flow chart shows the execution process of header files starting from builder.cc .

To keep the diagram simple repetitions are avoided. Once if the sequence of files is shown then if later on same file is used the sequence is omitted to keep the diagram understandable.