LevelDB-Study

Team_Cache Analysis

Made by Subin Hong, Seungwon Ha

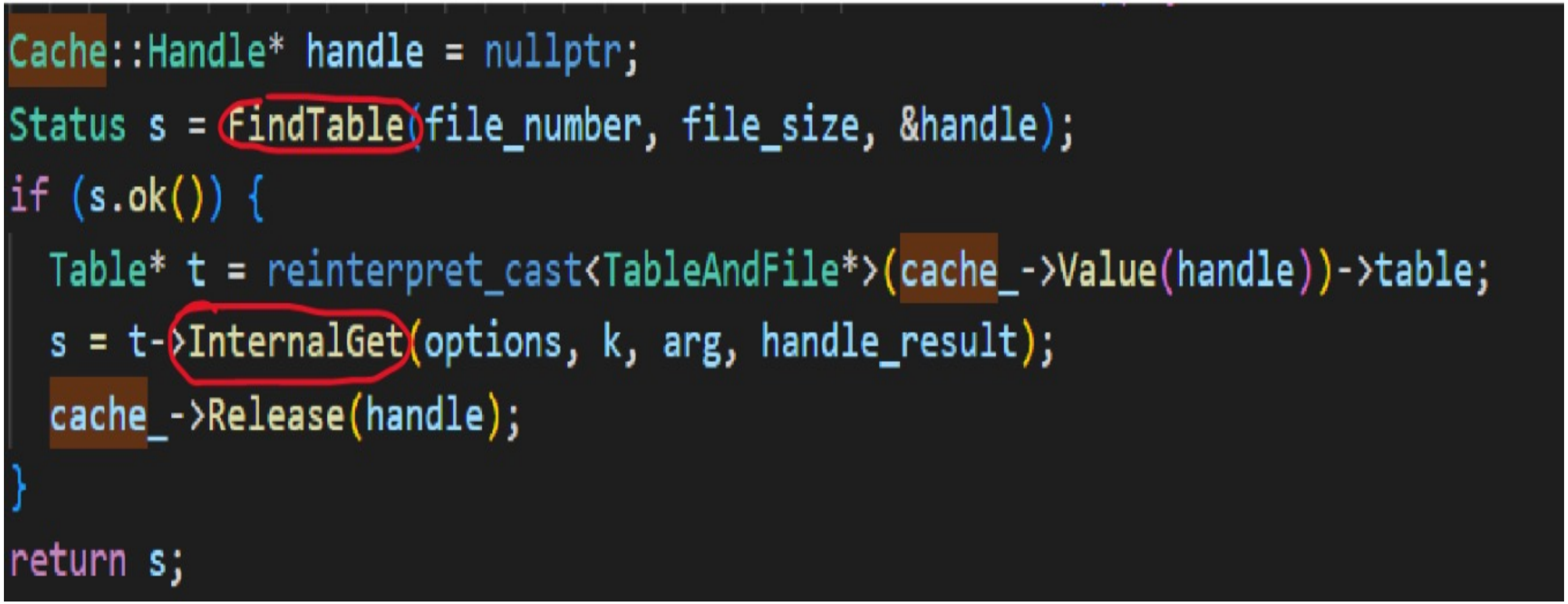
E-Mail: zed6740@dankook.ac.kr, 12gktmddnjs@naver.com

Contents

- ▪ Cache flow analysis

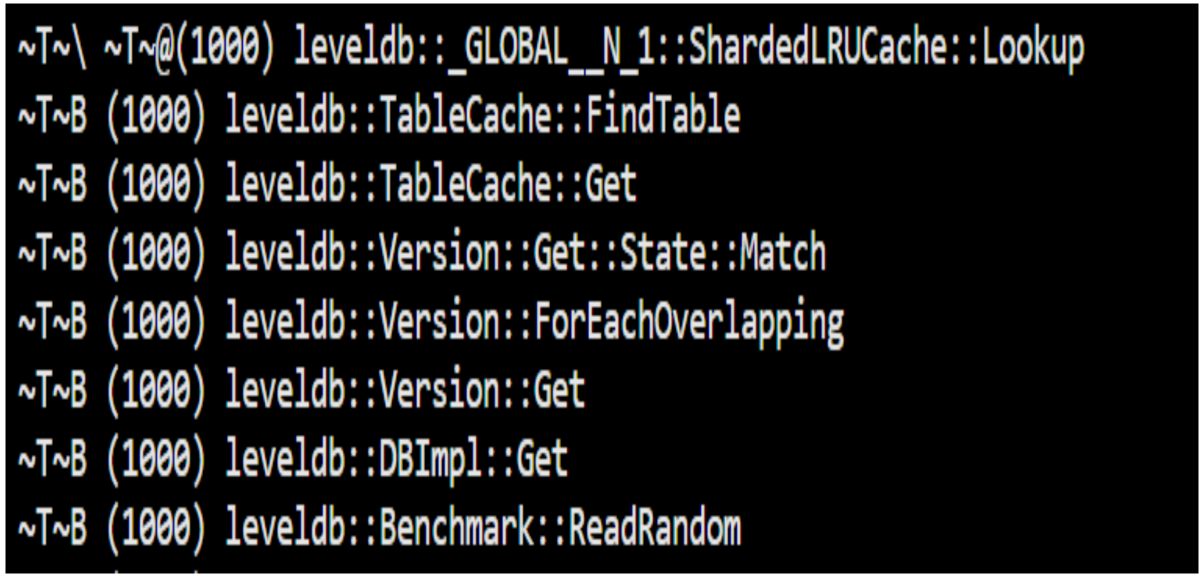
- Specific code analysis

Cache flow analysis

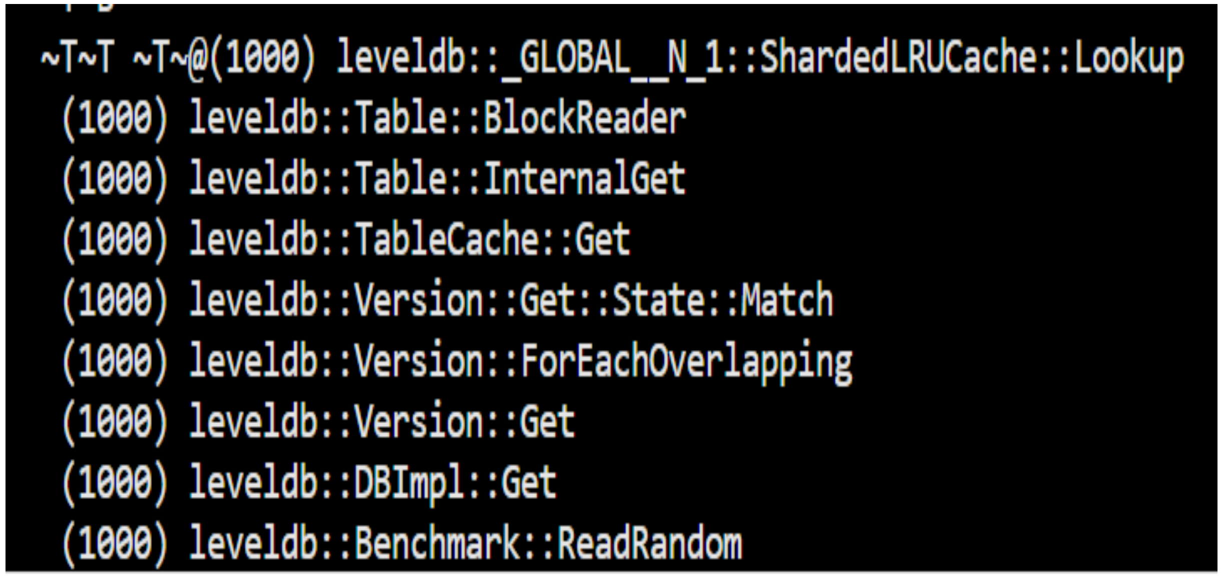


```
Cache::Handle* handle = nullptr;
Status s = FindTable(file_number, file_size, &handle);
if (s.ok()) {
  Table* t = reinterpret_cast<TableAndFile*>(cache_->Value(handle))->table;
  s = t->InternalGet(options, k, arg, handle_result);
  cache_->Release(handle);
}
return s;
```

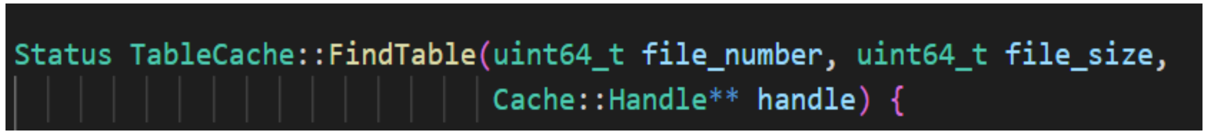
Cache flow analysis



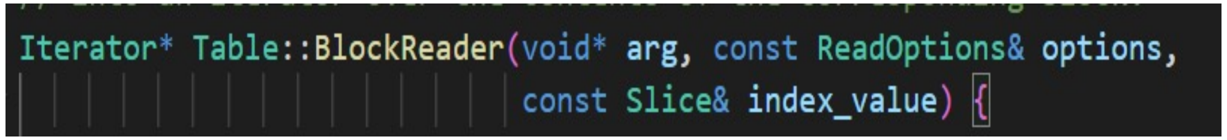
```
~T~\ ~T~@(1000) leveldb::_GLOBAL__N_1::ShardedLRUCache::Lookup
~T~B (1000) leveldb::TableCache::FindTable
~T~B (1000) leveldb::TableCache::Get
~T~B (1000) leveldb::Version::Get::State::Match
~T~B (1000) leveldb::Version::ForEachOverlapping
~T~B (1000) leveldb::Version::Get
~T~B (1000) leveldb::DBImpl::Get
~T~B (1000) leveldb::Benchmark::ReadRandom
```



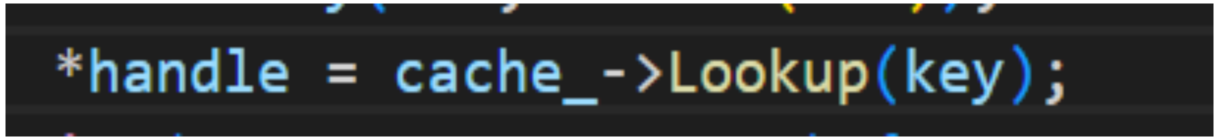
```
~T~T ~T~@(1000) leveldb::_GLOBAL__N_1::ShardedLRUCache::Lookup
 (1000) leveldb::Table::BlockReader
 (1000) leveldb::Table::InternalGet
 (1000) leveldb::TableCache::Get
 (1000) leveldb::Version::Get::State::Match
 (1000) leveldb::Version::ForEachOverlapping
 (1000) leveldb::Version::Get
 (1000) leveldb::DBImpl::Get
 (1000) leveldb::Benchmark::ReadRandom
```



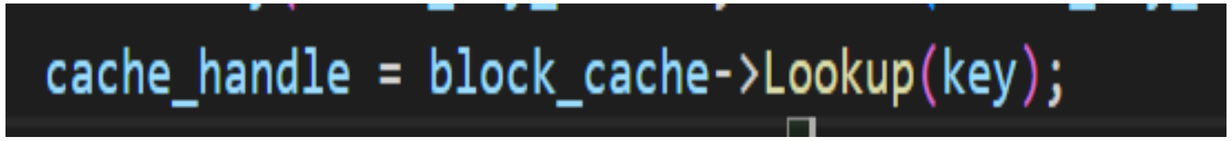
```
Status TableCache::FindTable(uint64_t file_number, uint64_t file_size,
                             Cache::Handle** handle) {
```



```
Iterator* Table::BlockReader(void* arg, const ReadOptions& options,
                             const Slice& index_value) {
```

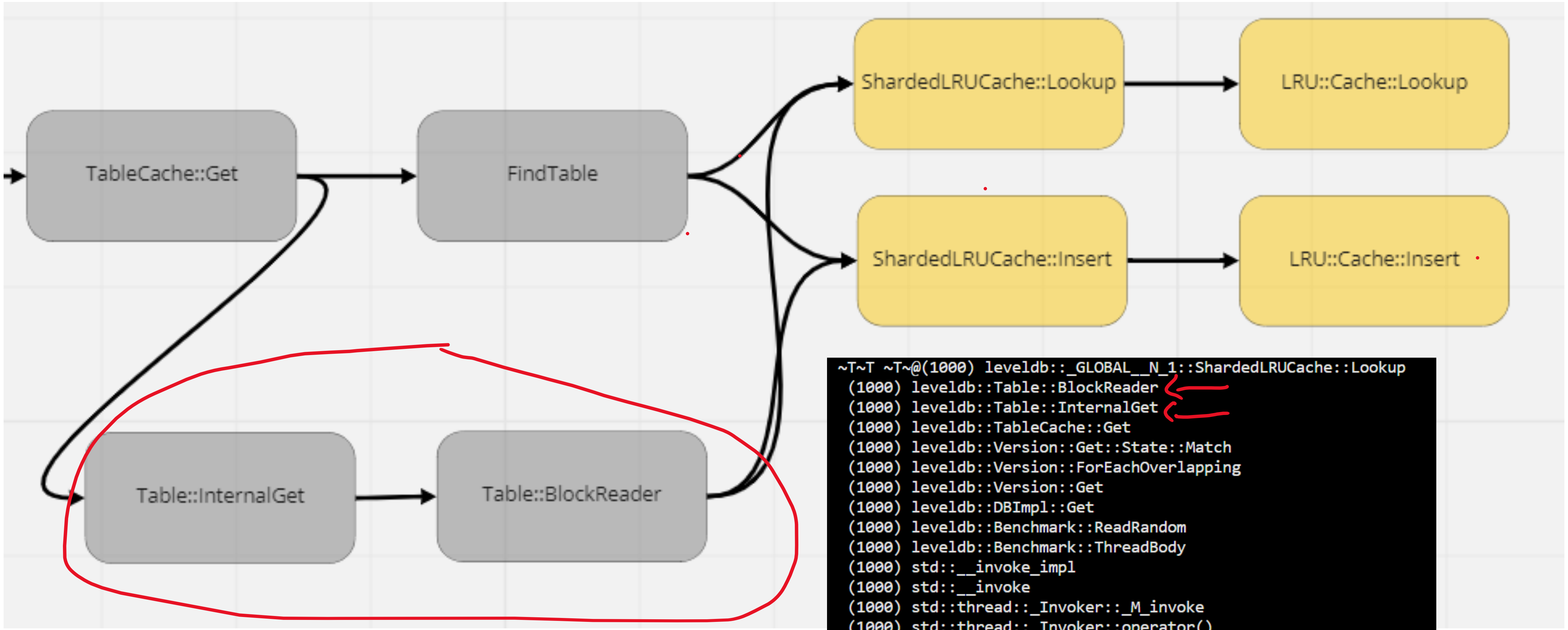


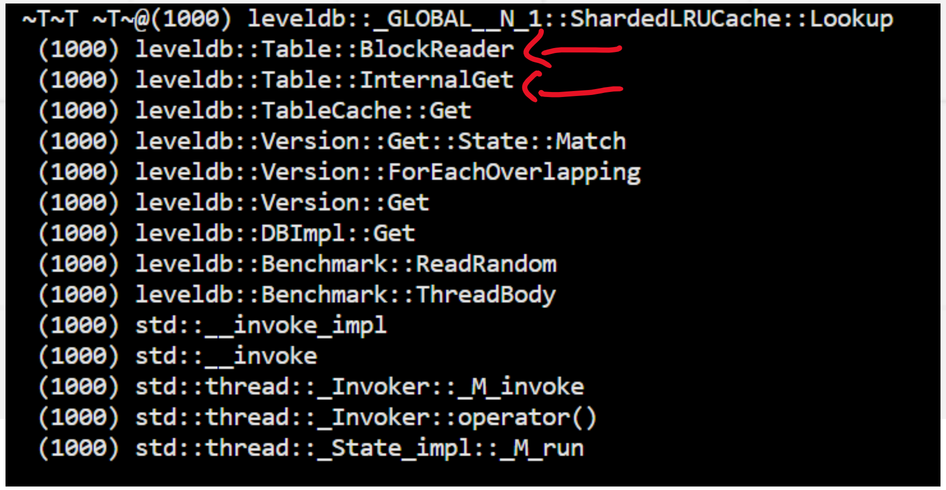
```
  *handle = cache_->Lookup(key);
```



```
cache_handle = block_cache->Lookup(key);
```

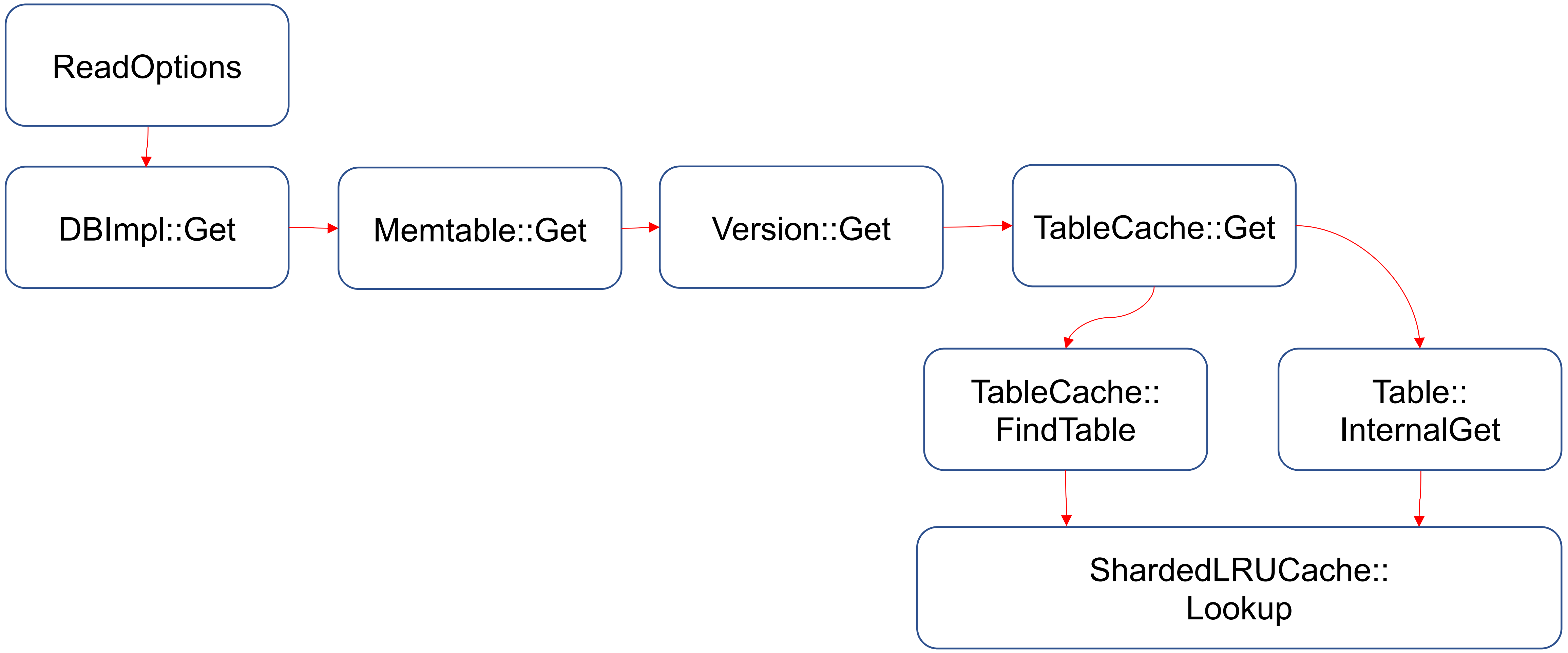
Cache flow analysis



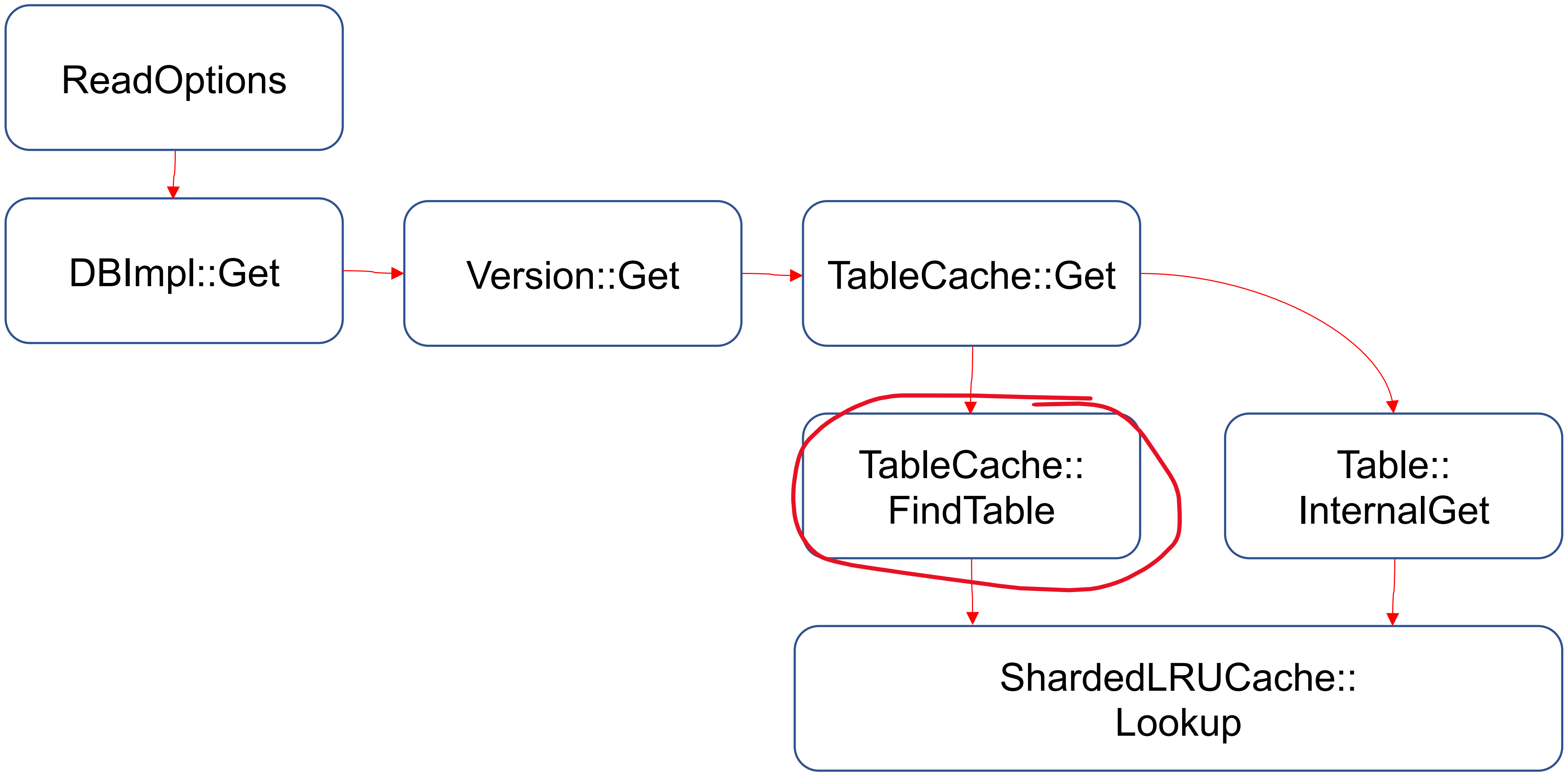


```
~T~T ~T~@(1000) leveldb::_GLOBAL__N_1::ShardedLRUCache::Lookup
  (1000) leveldb::Table::BlockReader
  (1000) leveldb::Table::InternalGet
  (1000) leveldb::TableCache::Get
  (1000) leveldb::Version::Get::State::Match
  (1000) leveldb::Version::ForEachOverlapping
  (1000) leveldb::Version::Get
  (1000) leveldb::DBImpl::Get
  (1000) leveldb::Benchmark::ReadRandom
  (1000) leveldb::Benchmark::ThreadBody
  (1000) std::__invoke_impl
  (1000) std::__invoke
  (1000) std::thread::_Invoker::_M_invoke
  (1000) std::thread::_Invoker::operator()
  (1000) std::thread::_State_impl::_M_run
```

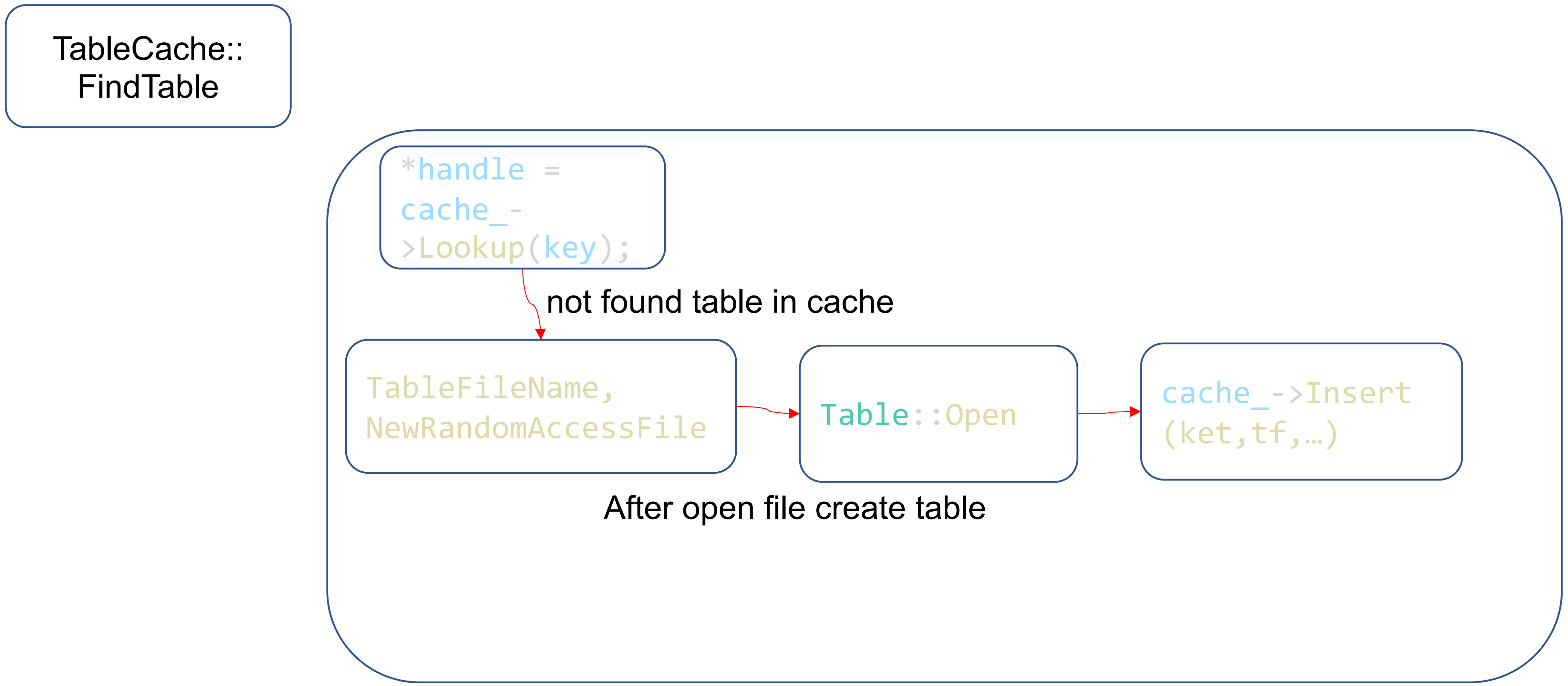
Cache flow analysis



Cache flow analysis



Cache flow analysis



```
*handle = cache_->Lookup(key);
```

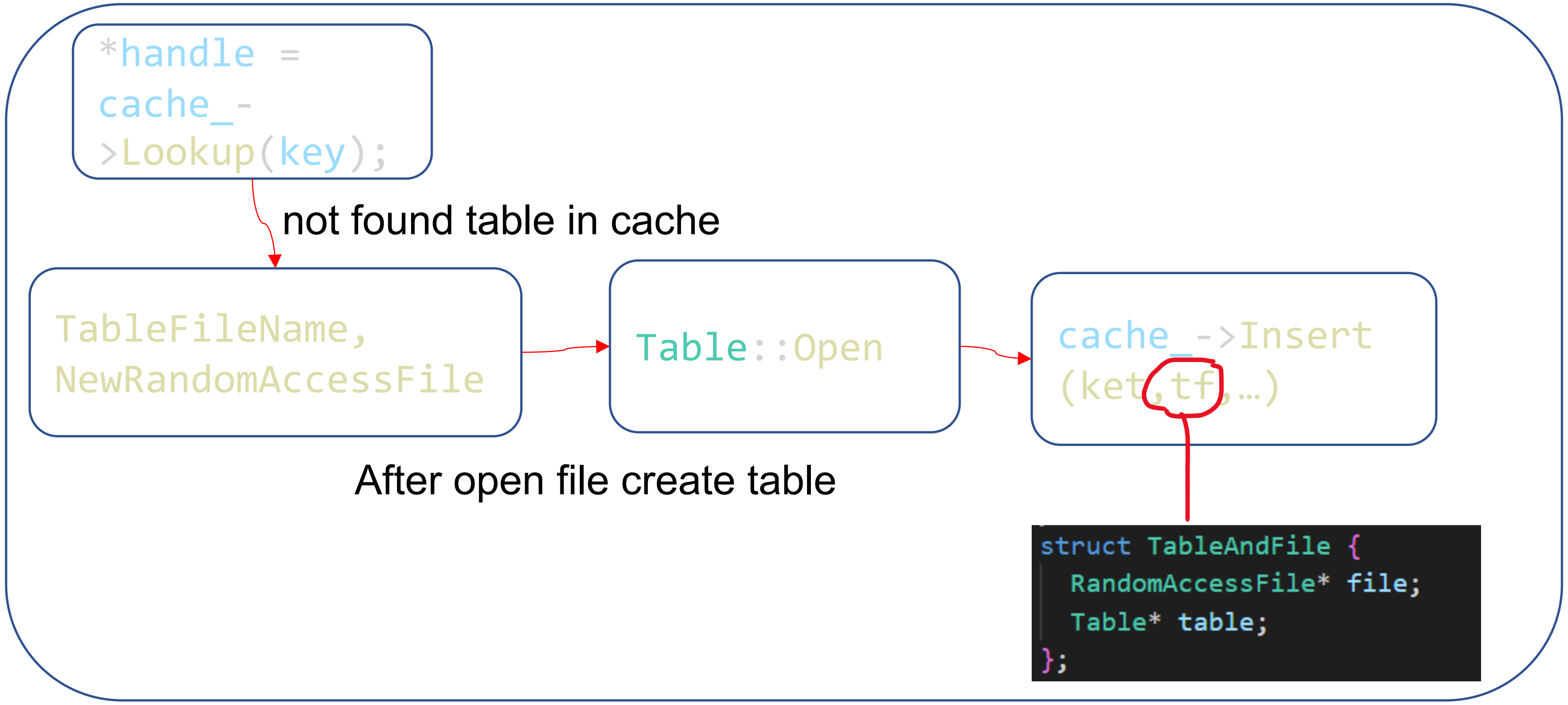
```
TableFileName, NewRandomAccessFile
```

```
Table::Open
```

```
cache_->Insert(ket,tf,…)
```

Cache flow analysis

TableCache::
FindTable



```
*handle = cache_->Lookup(key);
```

not found table in cache

```
TableFileName,
NewRandomAccessFile
```

```
Table::Open
```

```
cache_->Insert
(ket,tf,…)
```

After open file create table

```
struct TableAndFile {
  RandomAccessFile* file;
  Table* table;
};
```

Cache flow analysis

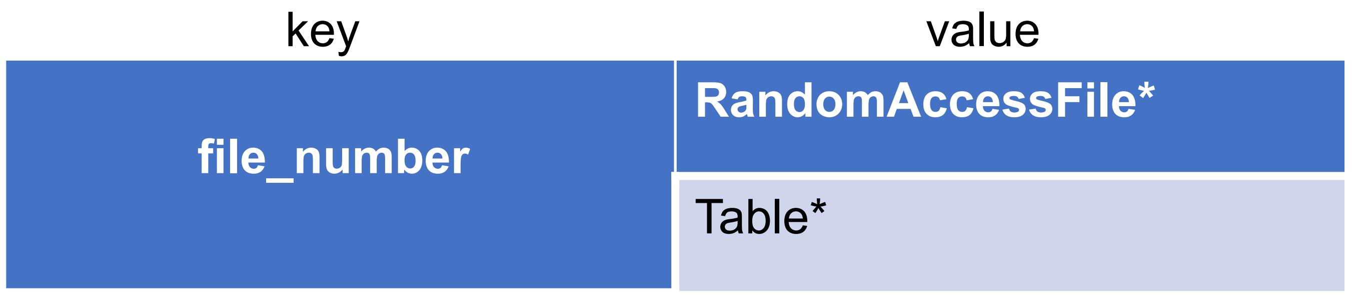
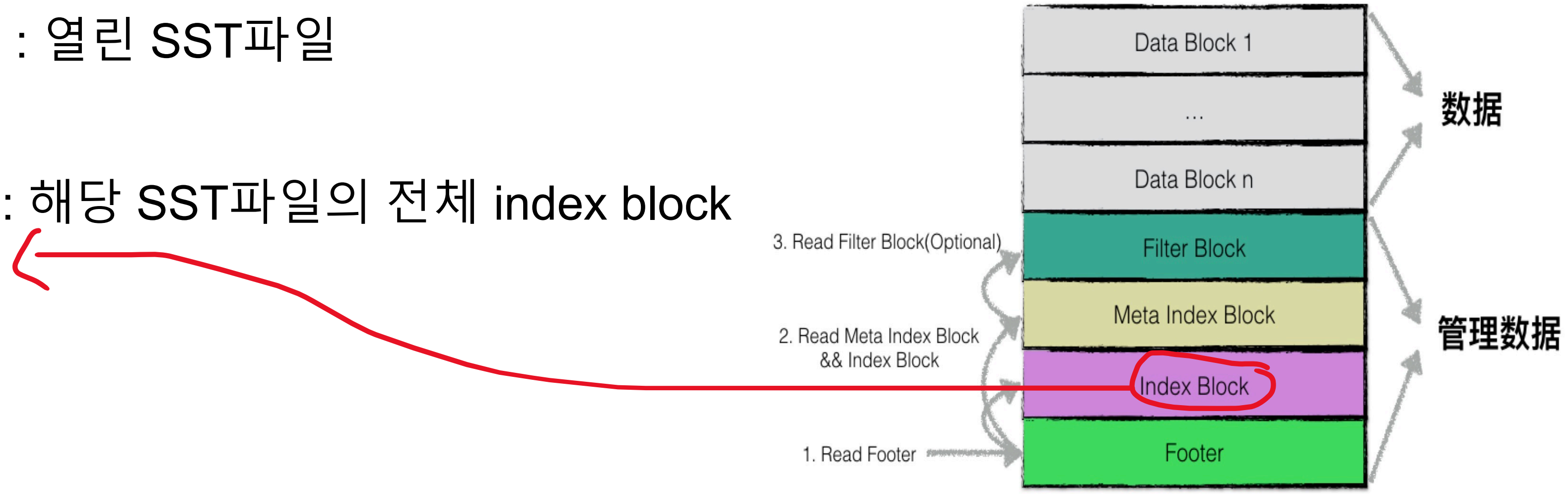


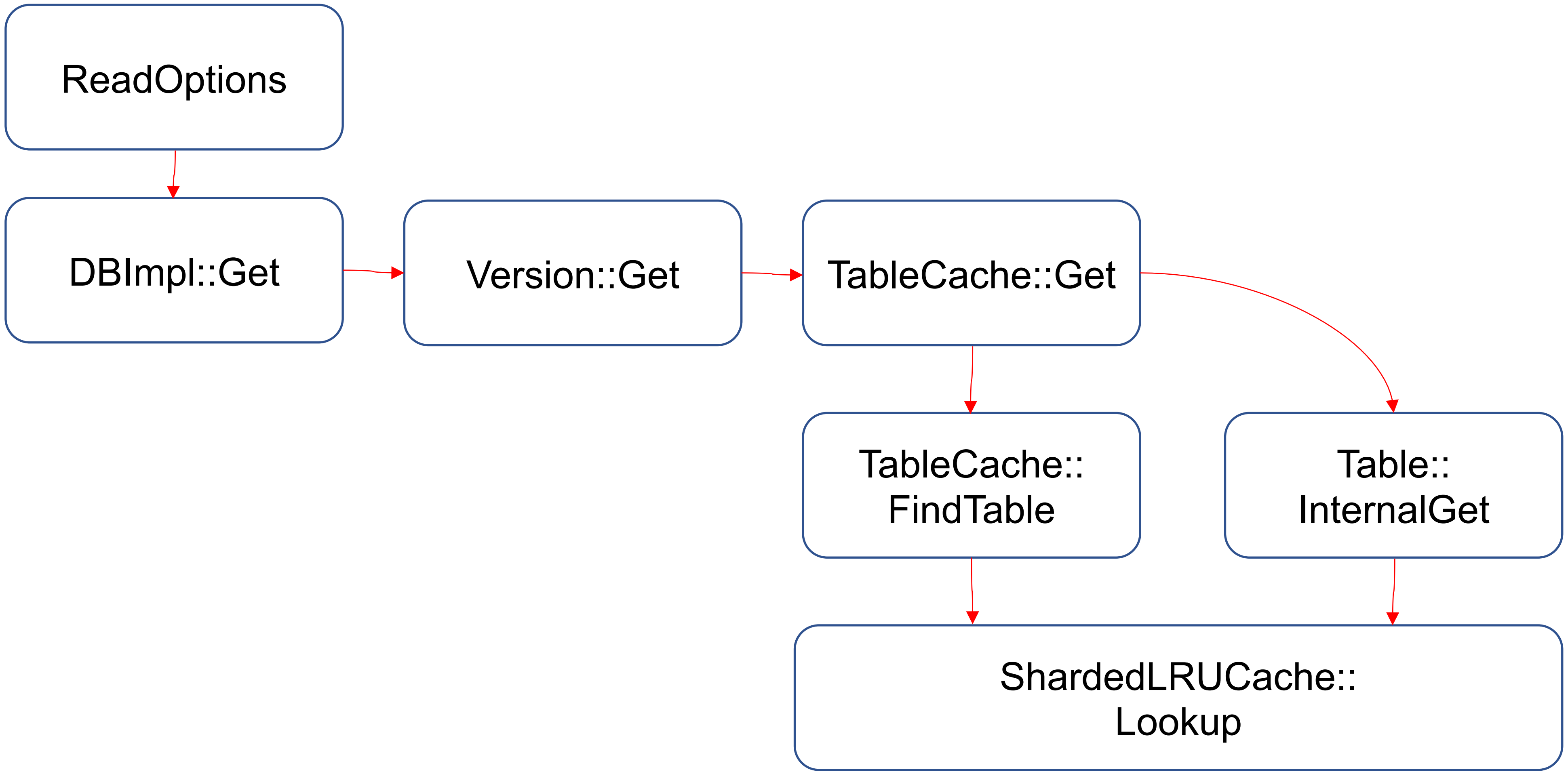
Table Cache Structure

RandomAccessFile* : 열린 SST파일

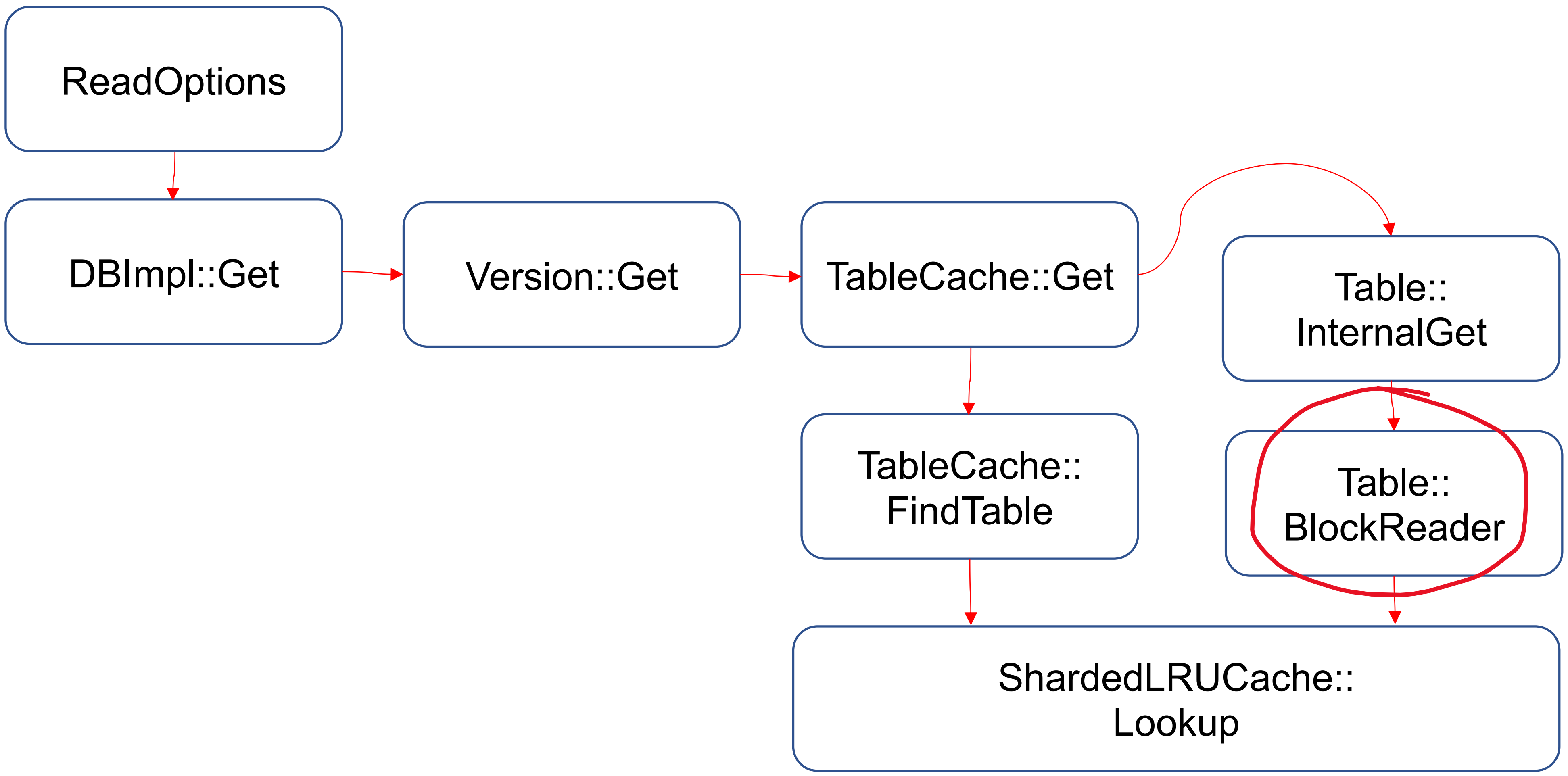
Table* : 해당 SST파일의 전체 index block



Cache flow analysis



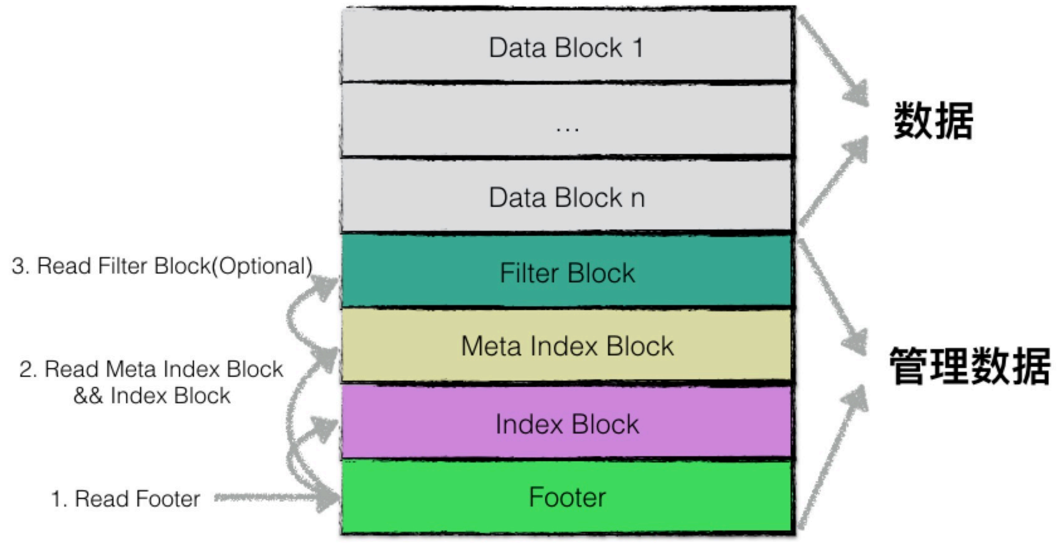
Cache flow analysis



Cache flow analysis

cache_id + block_offset	Block Data (Data Block)
cache_id + block_offset	Block Data
cache_id + block_offset	Block Data

Block Cache Structure

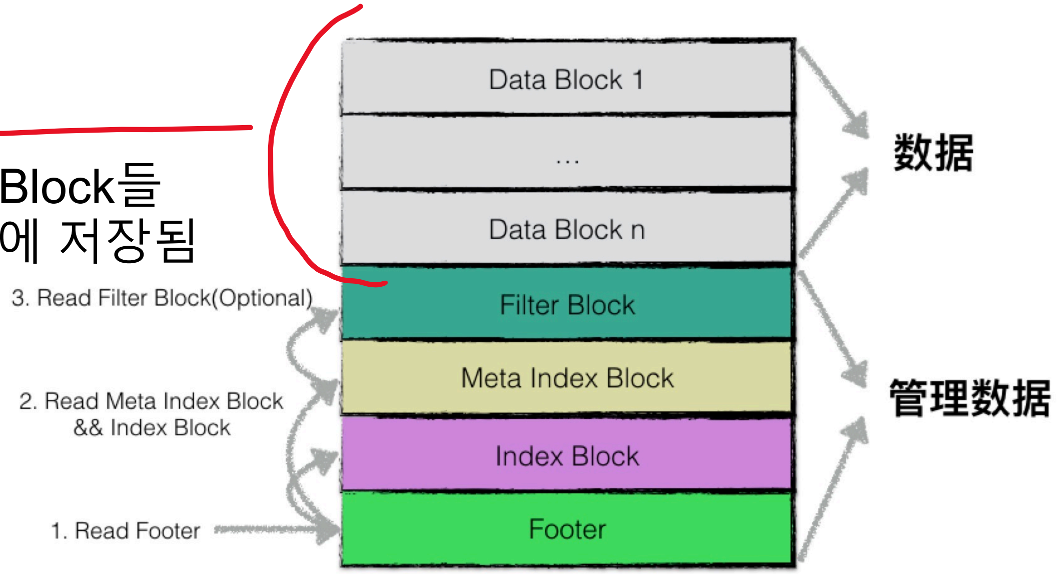


Cache flow analysis

cache_id + block_offset	Block Data (Data Block)
cache_id + block_offset	Block Data
cache_id + block_offset	Block Data

Block Cache Structure

열린 sst파일의 DataBlock들
은 전역 BlockCache에 저장됨

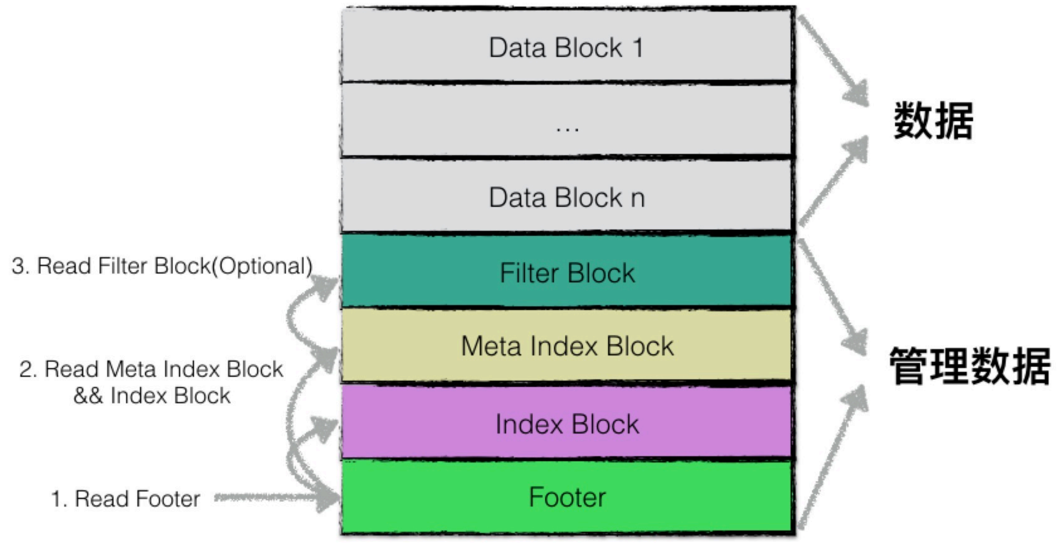


Cache flow analysis

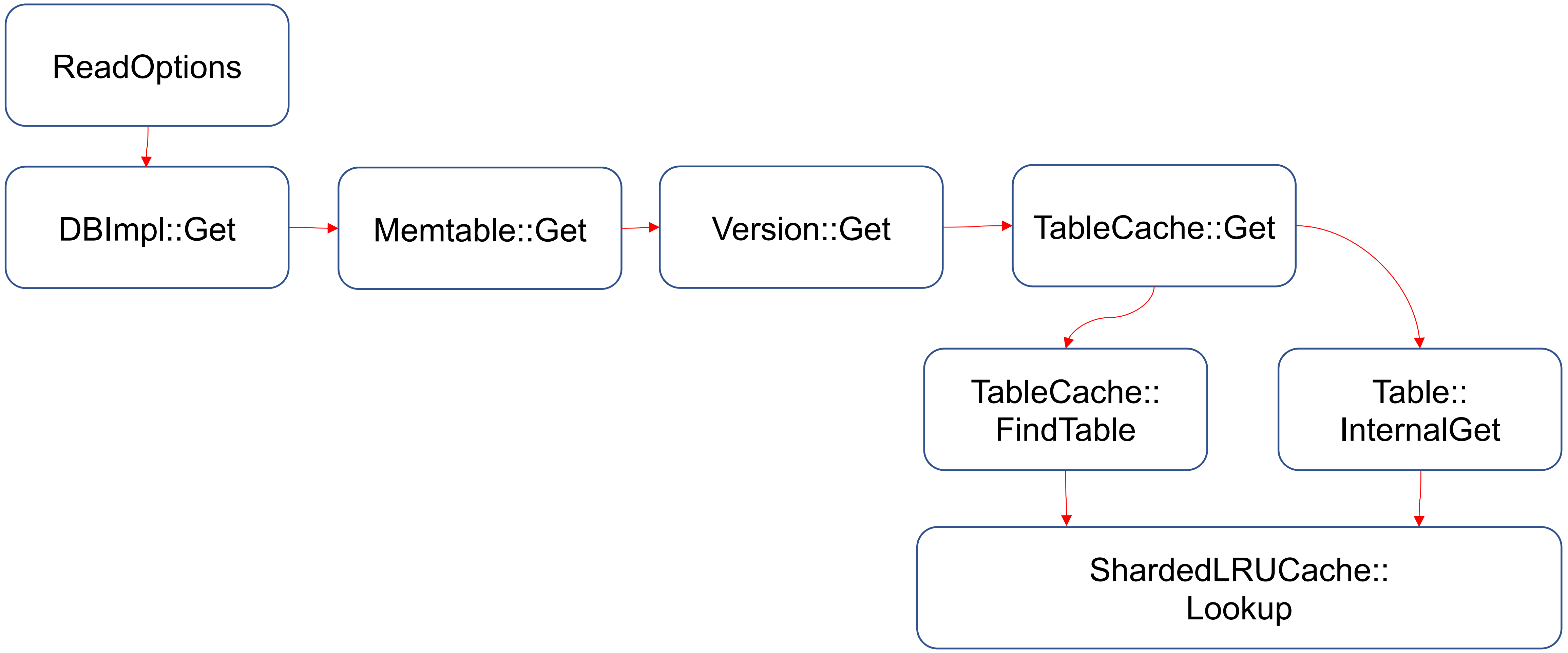
cache_id + block_offset	Block Data (Data Block)
cache_id + block_offset	Block Data
cache_id + block_offset	Block Data

Block Cache Structure

다른 sst파일의 Data Block offset이
동일할 수 있으므로 구별을 위해,
각 sst파일에 고유한 cache_id를 조
합하여 key를 구성함



Cache flow analysis


ReadOptions
DBImpl::Get
Memtable::Get
Version::Get
TableCache::Get
TableCache:: FindTable
Table:: InternalGet
ShardedLRUCache:: Lookup