

# Team Compaction

좌우꾸와썸

E-Mail : erosbryant@dankook.ac.kr

박서영

E-Mail : lilianapsy@naver.com

발표: 강상우

E-Mail : aarom416@naver.com

# Contents

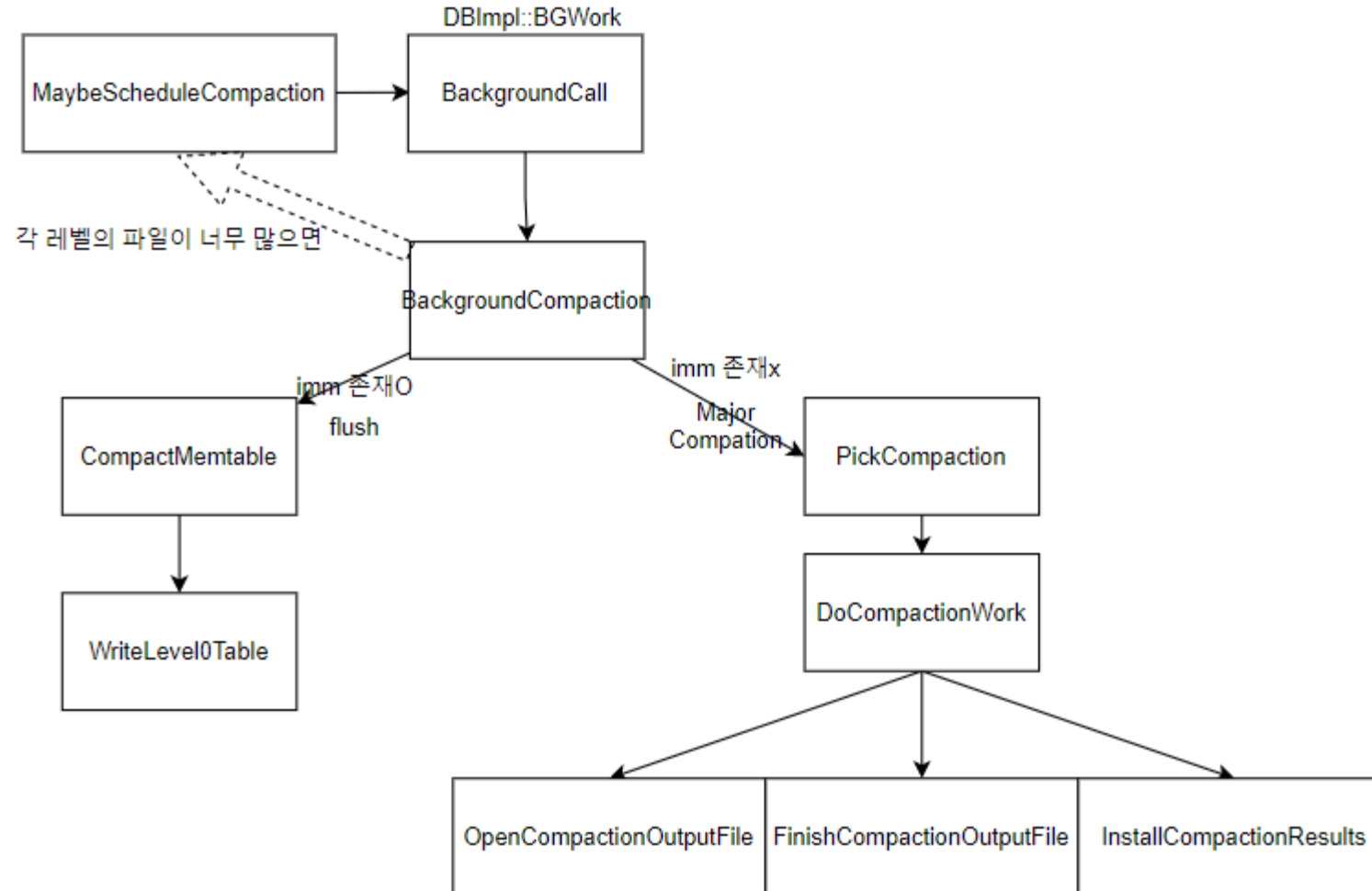
- Overall Compaction Code Flow
  - MaybeScheduleCompaction()
  - BackgroundCompaction()
  - DoCompactionWork()

# Setup Environment

- Use ufttrace record, tui
- Benchmark : fillrandom
- Num = 10000

```
erosbryant@ErosBryant-computer:~/workspace/sangwoo/leveldb_debug/build$ ufttrace record ./db_bench --benchmarks="fillrandom" --num=100000
LevelDB:    version 1.23
Date:       Sun Aug  7 00:22:40 2022
CPU:        20 * 12th Gen Intel(R) Core(TM) i7-12700K
CPUCache:   25600 KB
Keys:       16 bytes each
Values:     100 bytes each (50 bytes after compression)
Entries:    100000
RawSize:    11.1 MB (estimated)
FileSize:   6.3 MB (estimated)
WARNING: Optimization is disabled: benchmarks unnecessarily slow
WARNING: Assertions are enabled; benchmarks unnecessarily slow
-----
fillrandom  :      59.371 micros/op;      1.9 MB/s
```

# Overall Code Flow



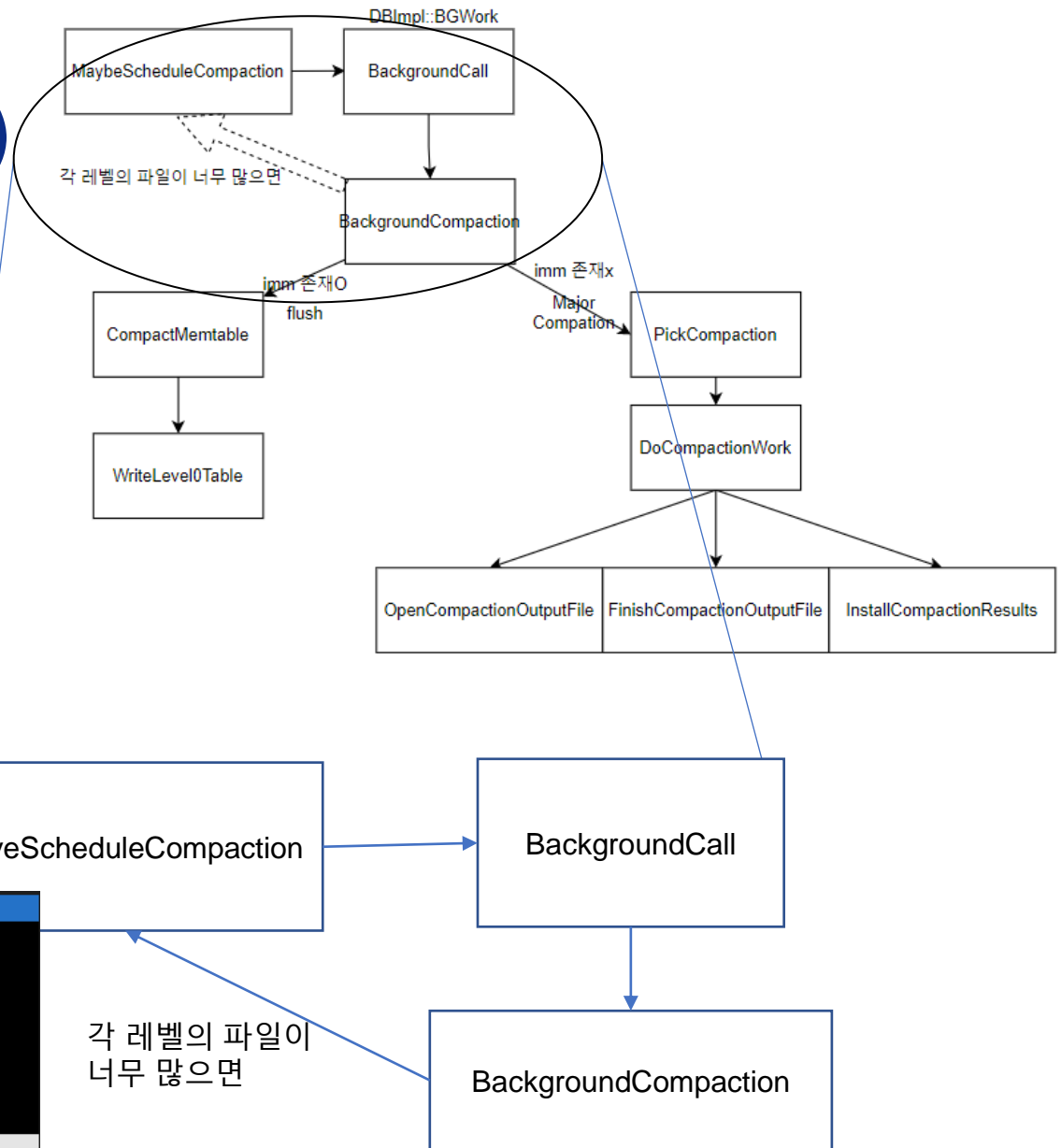
# Overall Code Flow: MaybeScheduleCompaction()

```
void DBImpl::MaybeScheduleCompaction() {
    mutex_.AssertHeld();
    if (background_compaction_scheduled_) {
        // Already scheduled
    } else if (shutting_down_.load(std::memory_order_acquire)) {
        // DB is being deleted; no more background compactions
    } else if (!bg_error_.ok()) {
        // Already got an error; no more changes
    } else if (imm_ == nullptr && manual_compaction_ == nullptr &&
```

## TOTAL TIME : FUNCTION

===== Call Graph =====

```
3.703 s : (8) leveldb::DBImpl::BackgroundCall
5.337 us : ▶(8) leveldb::MutexLock::MutexLock
:
1.008 us : ▶(8) std::atomic::load
:
0.201 us : -(8) leveldb::Status::ok
:
3.703 s : ▶(8) leveldb::DBImpl::BackgroundCompaction
:
3.681 us : ▶(8) leveldb::DBImpl::MaybeScheduleCompaction
:
8.705 us : ▶(8) leveldb::port::CondVar::SignalAll
:
3.755 us : ▶(8) leveldb::MutexLock::~~MutexLock
```



# Overall Code Flow: BackgroundCompaction()

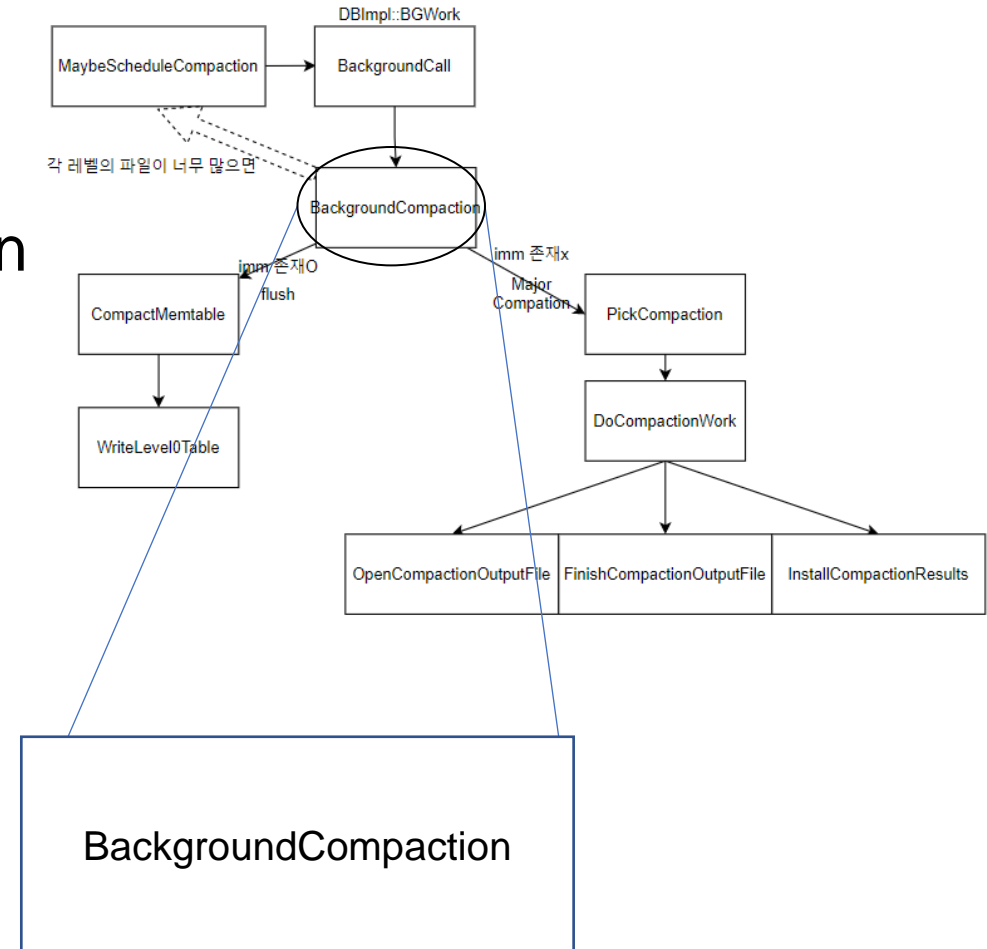
## Manual compaction vs Automatic compaction

### Manual compaction

- Set key range
- Debugging
- Not normally used

### Automatic compaction

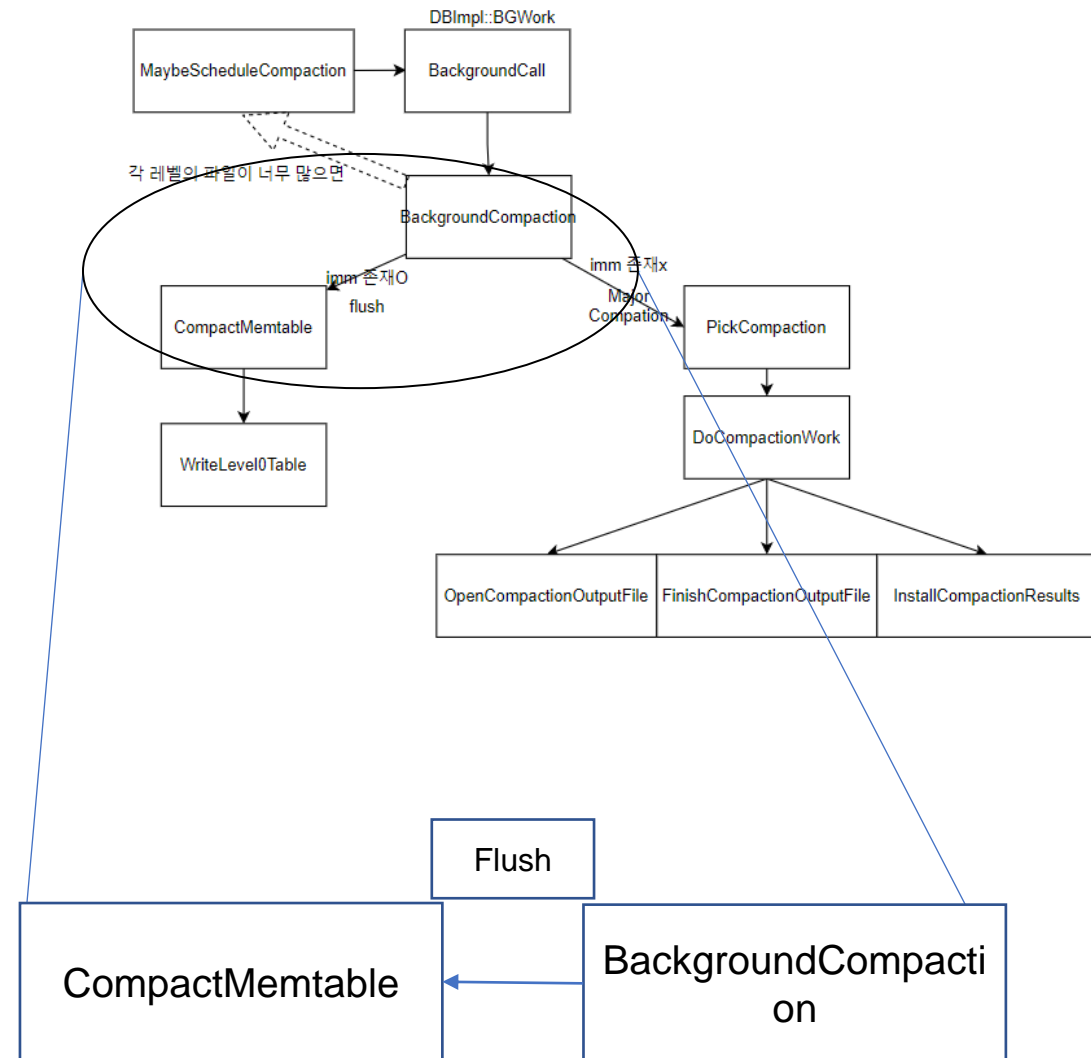
- Mainly used



# Overall Code Flow: BackgroundCompaction()

## CompactMemtable

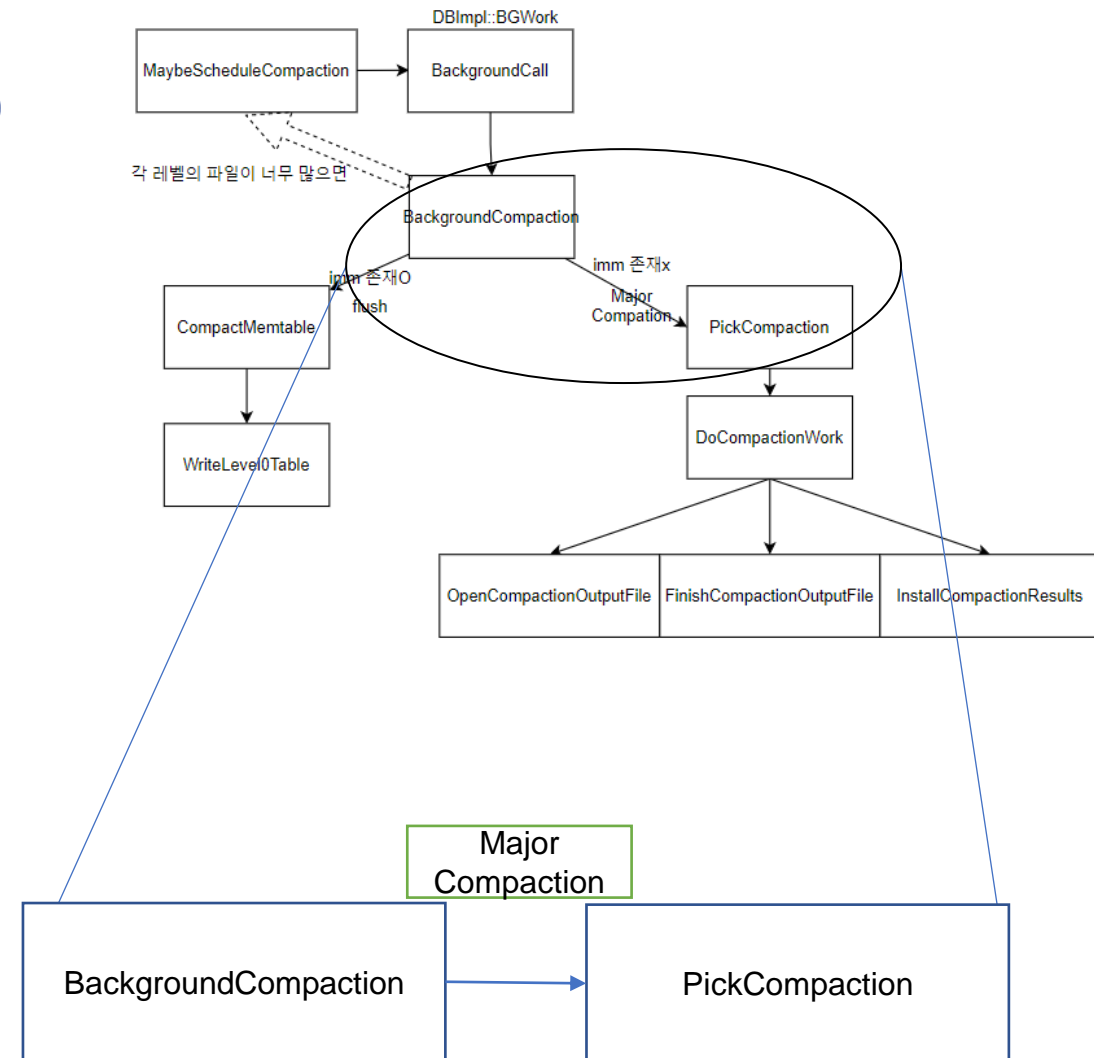
- Disk to flush
- Modify version
- Delete old files



# Overall Code Flow: BackgroundCompaction()

## PickCompaction

- Select the appropriate level



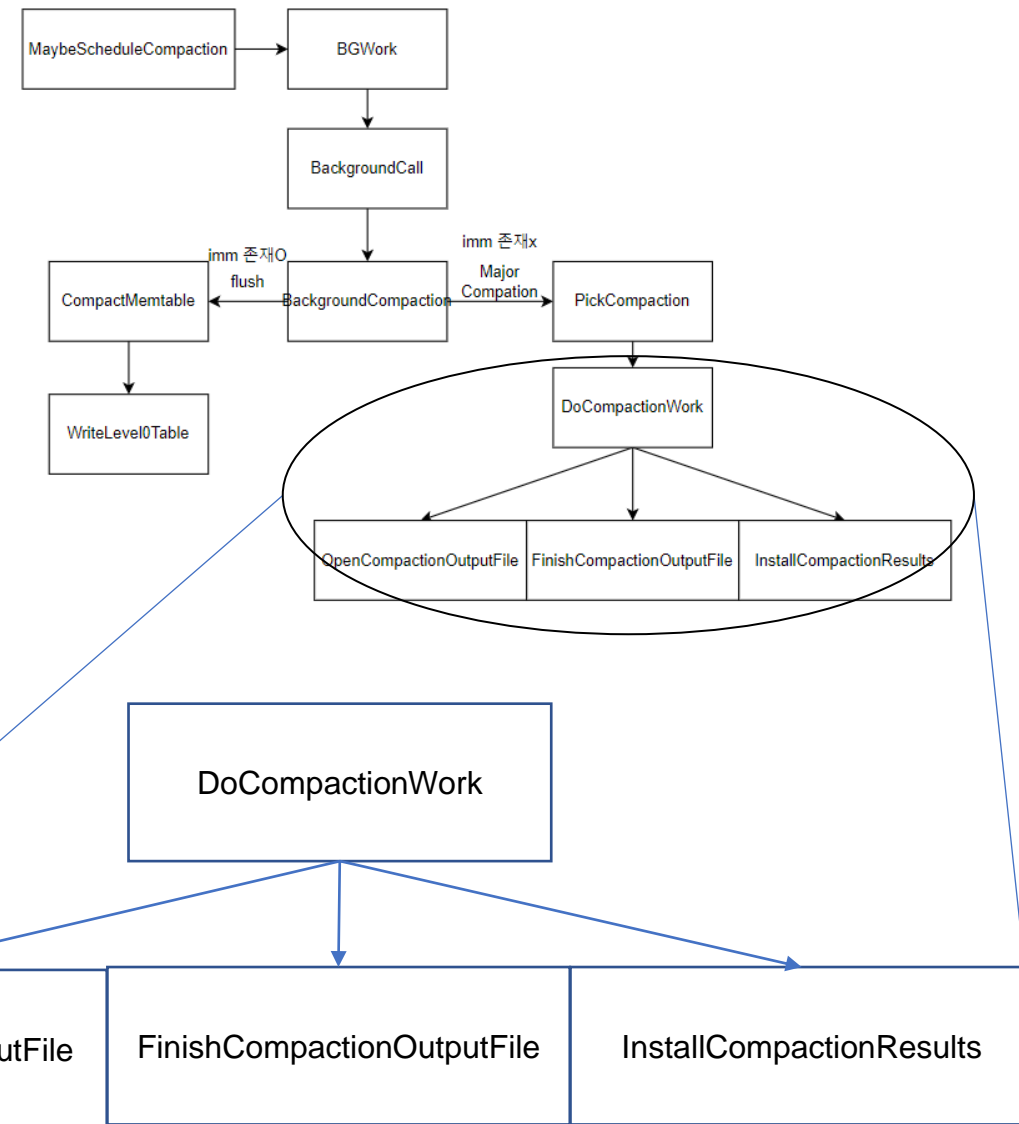


# Overall Code Flow: DoCompactionWork()

## DoCompactionWork

- Overall responsible for major compaction

```
22.605 us : (4) leveldb::PosixLogger::Logv
22.605 us : (4) leveldb::Log
22.605 us : (4) leveldb::DBImpl::DoCompactionWork
22.605 us : (4) leveldb::DBImpl::BackgroundCompaction
22.605 us : (4) leveldb::DBImpl::BackgroundCall
22.605 us : (4) leveldb::DBImpl::BGWork
22.605 us : (4) leveldb::_GLOBAL__N_1::PosixEnv::BackgroundThreadMain
22.605 us : (4) leveldb::_GLOBAL__N_1::PosixEnv::BackgroundThreadEntryPoint
22.605 us : (4) std::_invoke_impl
22.605 us : (4) std::_invoke
22.605 us : (4) std::thread::Invoker::M_invoke
22.605 us : (4) std::thread::Invoker::operator()
22.605 us : (4) std::thread::State_impl::M_run
```

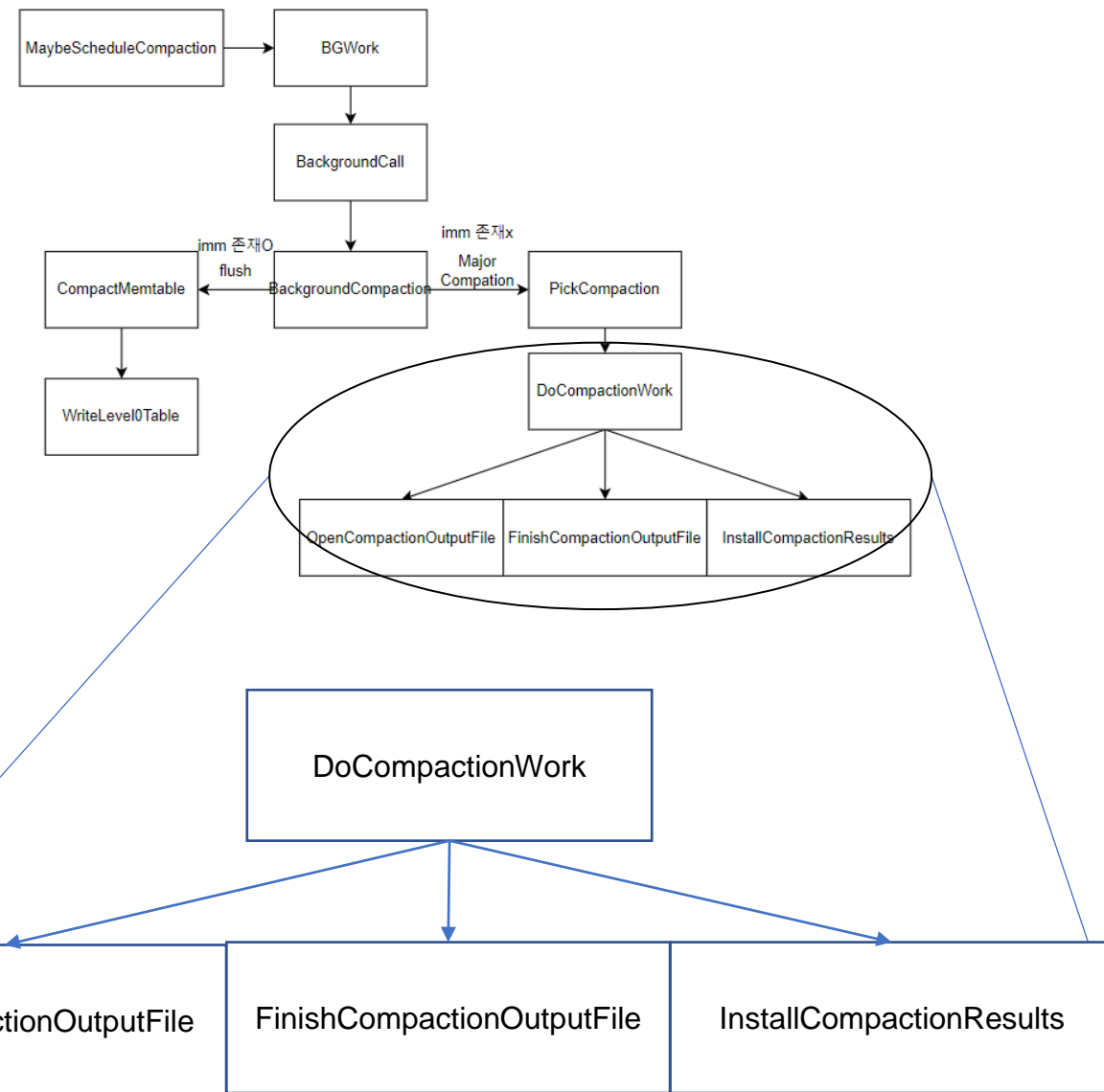


# Overall Code Flow: DoCompactionWork()

## OpenCompactionOutputFile

- If new sstable necessary
- Create new sstable
- Write sstable

```
// Make the output file
std::string fname = TableFileName(dbname_, file_number);
Status s = env_->NewWritableFile(fname, &compact->outfile);
if (s.ok()) {
    compact->builder = new TableBuilder(options_, compact->outfile);
}
return s;
}
```

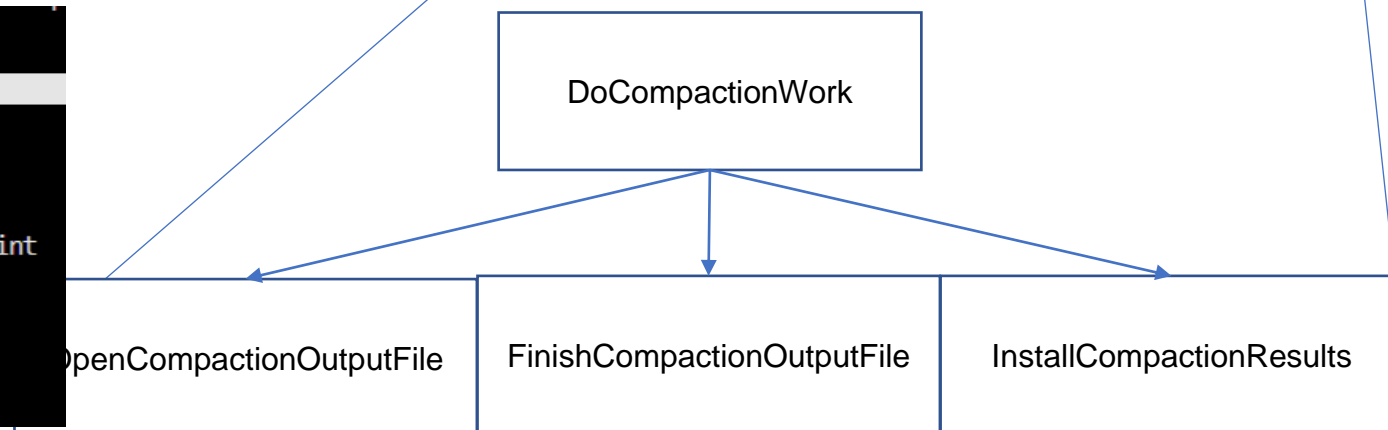
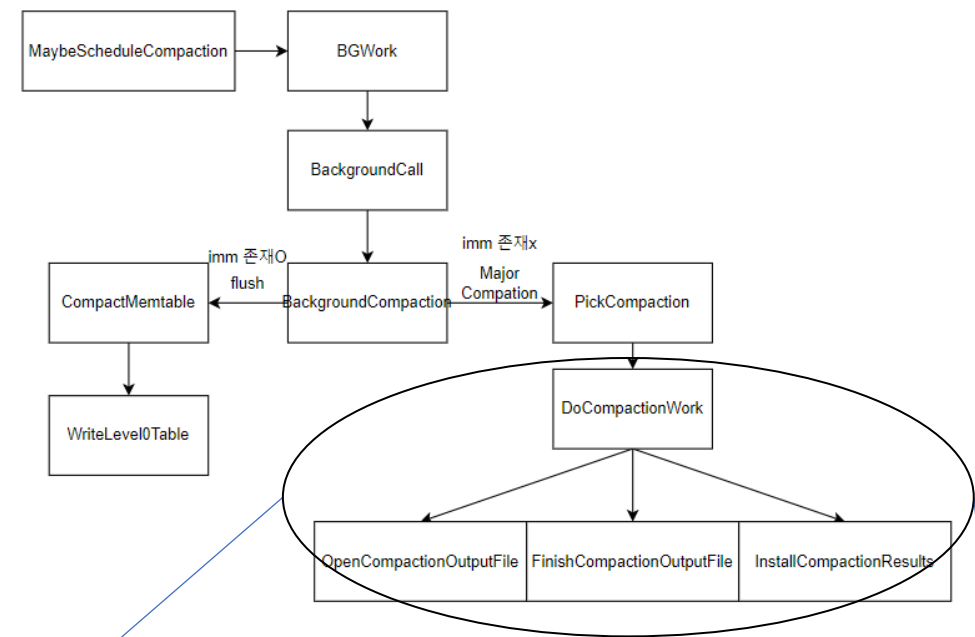


# Overall Code Flow: DoCompactionWork()

## FinishCompactionOutputFile

- Literal error check
- Sstable error check
- Usable check

```
===== Back-trace =====
35.562 ms : (4) leveldb::DBImpl::FinishCompactionOutputFile
35.562 ms : (4) leveldb::DBImpl::DoCompactionWork
35.562 ms : (4) leveldb::DBImpl::BackgroundCompaction
35.562 ms : (4) leveldb::DBImpl::BackgroundCall
35.562 ms : (4) leveldb::DBImpl::BGWork
35.562 ms : (4) leveldb::GLOBAL_N 1::PosixEnv::BackgroundThreadMain
35.562 ms : (4) leveldb::GLOBAL_N 1::PosixEnv::BackgroundThreadEntryPoint
35.562 ms : (4) std::__invoke_impl
35.562 ms : (4) std::__invoke
35.562 ms : (4) std::thread::_Invoker::_M_invoke
35.562 ms : (4) std::thread::_Invoker::operator()
35.562 ms : (4) std::thread::State_impl::_M_run
```

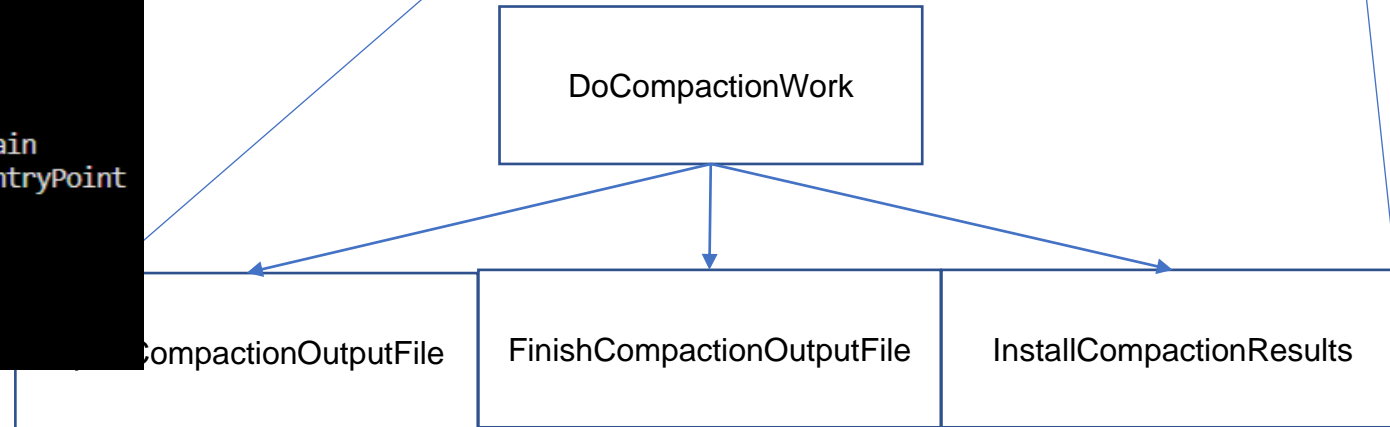
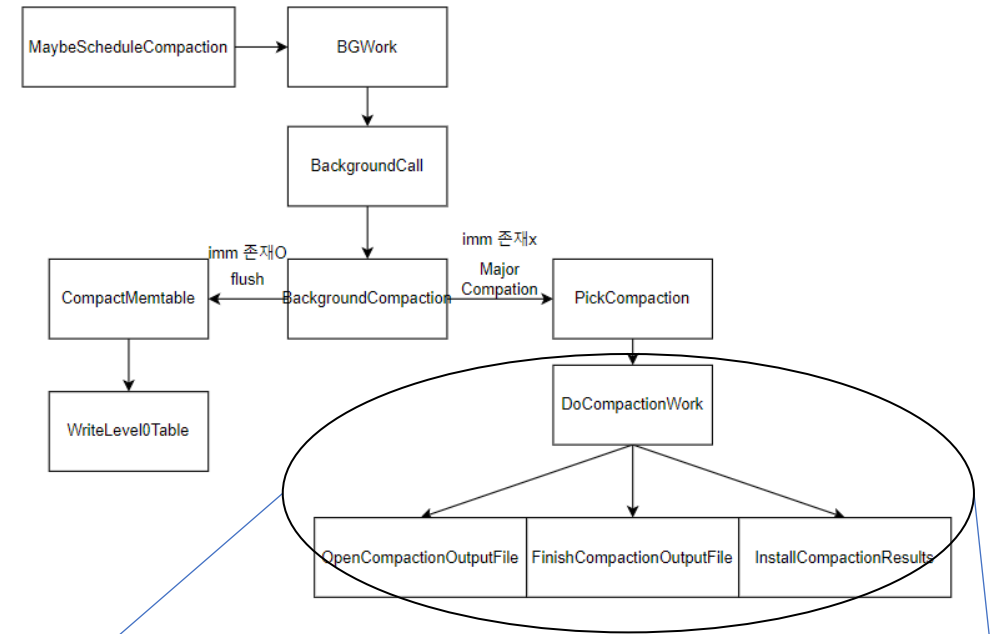


# Overall Code Flow: DoCompactionWork()

## InstallCompactionResults

- Level, Level+1 add compaction output
- Version update

```
6.432 us : (2) leveldb::DBImpl::InstallCompactionResults
6.432 us : (2) leveldb::DBImpl::DoCompactionWork
6.432 us : (2) leveldb::DBImpl::BackgroundCompaction
6.432 us : (2) leveldb::DBImpl::BackgroundCall
6.432 us : (2) leveldb::DBImpl::BGWork
6.432 us : (2) leveldb::GLOBAL_N_1::PosixEnv::BackgroundThreadMain
6.432 us : (2) leveldb::GLOBAL_N_1::PosixEnv::BackgroundThreadEntryPoint
6.432 us : (2) std::__invoke_impl
6.432 us : (2) std::__invoke
6.432 us : (2) std::thread::_Invoker::_M_invoke
6.432 us : (2) std::thread::_Invoker::operator()
6.432 us : (2) std::thread::_State_impl::_M_run
```



# Thank you