

# Redo log/InnoDB log file size Optimization

**목표:** 이 강의에서는 Redo Log 와 Log Buffer 를 이용한 최적화 방법에 대해서 배워봅니다.

**포인트:** Redo Log 개념과 트랜잭션에서 어떻게 사용되는지 그리고 이를 이용한 최적화 방법에 대해서 이해하는게 중요합니다.

## ▼ Understanding and Operation of the Redo Log

### Redo Log 란?

- 데이터 변경 내역을 로그에 저장해서 Durability 를 올리기 위해 사용된다.
- 즉 커밋했지만 아직 디스크에 반영되지 않는 데이터가 있는 상황에서 MySQL 이 죽은 경우에 복구할 때 사용한다.
- `innodb_flush_log_at_trx_commit` 값에 따라서 플러쉬 시점이 달라진다.

### 트랜잭션 처리와 Redo Log 기록 과정:

1. 트랜잭션이 데이터를 변경하면, 해당 변경 내용은 먼저 Log Buffer에 기록됨.
2. 트랜잭션이 커밋을 요청하면, MySQL은 먼저 Log Buffer 에 있는 모든 관련 로그 기록을 Redo Log 파일로 플러시
3. Log Buffer의 내용이 Redo Log 파일로 플러시된 후, 트랜잭션이 성공적으로 커밋됨. 이로써 데이터 변경 사항이 안정적으로 저장된 것으로 간주.

### `innodb_flush_log_at_trx_commit` 값에 따른 동작 과정:

- 값이 0이면 1초에 한번씩 플러쉬됨. 그래서 Durability 를 엄격하게 보장하지 않음.
- 값이 1이면 매 트랜잭션마다 디스크에 플러쉬됨. 엄격한 Durability 를 보장함.
- 값이 2이면 매 트랜잭션마다 커밋되면 운영체제의 메모리 버퍼에 기록되고 동기화는 1초마다 유지됨. MySQL 이 종료되더라도 운영체제가 정상적으로 작동한다면 Durability 를 보장. 그렇지 않더라도 1초간의 유실이 생길 수 있음.

## 체크포인트란? :

- 체크포인트는 데이터베이스가 Redo Log의 내용을 정기적으로 디스크의 실제 데이터 파일로 플러시하는 과정.

## ▼ Optimization Methods:

### Redo Log 파일 사이즈를 증가시키는 것:

- Redo Log 파일 사이즈만큼 가득 차게되면 Redo Log 의 변경된 데이터 내역들이 디스크 데이터에 반영이되는 체크포인트 과정이 발생한다.
- 만약 Redo Log 의 사이즈가 작다면 이런 체크포인트 과정이 자주 발생해서 Disk Write 작업이 많이 발생할 수 있다.
- Write Intensive 한 어플리케이션이라면 Redo Log File 사이즈를 증가시키는 것이 좋다.
- MySQL 8.0.30 부터는 `innodb_redo_log_capacity` 로 설정할 수 있고, 이전 버전에서는 `innodb_log_file_size` 와 `innodb_log_files_group` 를 사용해야함.
- 적절한 Redo Log 파일 사이즈를 측정하기 위해서는 Peak Traffic 동안 Redo Log 파일이 쌓이는 사이즈를 측정해서 이를 바탕으로 설정하는 것을 권장한다.

### Redo Log 가 특정 시간동안 얼마나 쌓였는지 아는 방법:

#### Secret #2: InnoDB Redo Log - Recommendations

During peak traffic time, you can get an estimation of the required amount for the Redo Log Capacity by running the query below (all in one single line):

```
MySQL > SELECT VARIABLE_VALUE from performance_schema.global_status
WHERE VARIABLE_NAME='Innodb_redo_log_current_lsn' INTO @a;SELECT sleep(60)
INTO @garb ;SELECT VARIABLE_VALUE FROM performance_schema.global_status
WHERE VARIABLE_NAME='Innodb_redo_log_current_lsn' INTO @b;select
format_bytes(abs(@a - @b)) per_min, format_bytes(abs(@a - @b)*60) per_hour;
+-----+-----+
| per_min | per_hour |
+-----+-----+
| 21.18 MiB | 1.24 GiB |
+-----+-----+
```

### Log Buffer 의 사이즈를 올리는 것:

- 트랜잭션의 변경 내용은 Log Buffer 에 쌓이게되고 이후 커밋 요청을 하면 Redo Log 로 플러쉬 된다.
- 그러나 큰 트랜잭션으로 인해서 Log Buffer 의 크기보다 더 많은 내용이 쌓이게 된다면 미리 디스크에 플러쉬하는 과정이 발생한다.
- 그래서 Bulk Insert 와 같은 큰 트랜잭션 처리 작업을 하게 된다면 Log Buffer 의 크기를 올려서 디스크에 플러쉬 하는 과정을 줄일 수 있다.
- `innodb_log_buffer_size` 로 Log Buffer 의 크기를 제어할 수 있다.
- `innodb_log_waits` 상태 변수 값을 통해서 미리 디스크에 플러쉬 한 과정이 발생했는지 알 수 있다.