

# Scalable Lock Manager

---

## Concurrent Programming

### Programming Project #4

Final due date: Dec 23, 2017 (HARD DEADLINE)

## 1 OBJECTIVE

MariaDB 10.2에서 기본으로 사용하는 InnoDB storage engine의 Lock Manager에는 몇가지 성능 병목점이 존재한다. InnoDB의 Lock Manager는 가장 먼저 병목으로 지목되는 곳으로서 이 부분을 scalable하게 만들어 MariaDB의 성능을 개선하는 것을 본 프로젝트의 최종 목표로 한다.

## 2 DETAILS

DBMS에서는 Lock Manager를 통해 동시에 수행되는 transaction의 충돌을 방지한다. Lock Manager는 internal data structure를 통해 수행중인 transaction이 획득한 lock을 관리하고,

추가적인 lock 획득 요청에 대해 허용하거나 block시킨다. 또한 deadlock checking을 통해 transaction간에 deadlock이 발생하지 않도록 관리한다.

InnoDB의 Lock Manager에서는 다양한 lock mode 중 이번 과제에서는 Shared/Exclusive 두가지 mode의 lock mode를 고려하여 과제를 수행하고 record lock만을 관심의 대상으로 한다. Transaction이 특정 record에 read operation을 수행하려 하면 Lock Manager에 Shared lock의 획득을 요청하며, write operation을 수행하려 하면 Exclusive lock의 획득을 요청한다. 한 record가 Shared lock으로 잠겨 있는 상태에서 다른 transaction은 동시에 해당 record에 대한 Shared lock을 획득할 수 있으나, Exclusive lock은 획득할 수 없다. 한 record가 Exclusive lock으로 잠겨 있는 상태에서는 다른 transaction이 해당 record에 대한 Shared/Exclusive lock을 획득할 수 없다.

이번 프로젝트는 2개의 Milestone으로 나누어 진행하며, 각 Milestone에 따른 제출기한이 정해져 있다.

(관련된 코드는 교수님께서 수업시간에 설명해주실 예정입니다)

### 3 MILESTONE 1

Milestone 1에서는 read-only transaction만을 수행할 경우에 발생하는 성능 병목 현상을 해결해야 한다. Read-only transaction은 값을 읽으려는 record에 대해 Shared lock만을 요청한다. 모든 transaction이 read-only인 경우 어떠한 record의 값을 동시에 읽어도 아무 문제가 없지만, InnoDB의 Lock Manager는 global mutex를 통해 lock을 획득하는 과정을 보호하기 때문에 성능 병목이 발생한다.

첫 번째 milestone에서는 read-only transaction만을 수행하는 경우 Lock Manager를 scalable하게 만들어서 read-only workload에 대한 성능을 향상시키는 것을 목표로 한다.

(due date: Dec 6, 2017)

## 4 MILESTONE 2

Milestone 2에서는 read/write transaction을 수행할 경우에 발생하는 성능 병목 현상을 해결해야 한다. InnoDB의 Lock Manager는 transaction이 어떠한 record에 대해 lock을 요청할 때, 해당 lock이 deadlock을 발생시킬 수 있는지를 판단(deadlock checking)하여 deadlock 상황을 방지한다. Lock Manager는 deadlock checking을 수행할 때 global mutex를 사용함으로 인해 성능 병목을 유발하게 된다. 두 번째 milestone에서는 Lock Manager의 deadlock checking을 scalable하게 만들어 read-write workload 에서도 MariaDB의 성능을 향상시킬 수 있도록 한다.

(due date: Dec 23, 2017)

## 5 TEST PROTOCOL

MariaDB의 성능 측정은 sysbench를 활용한다.

Milestone 1의 경우 read-only workload만을 수행하여 성능을 측정하고 perf 를 통해 병목이 되는 곳이 어디인지 그리고 잘 해결되었는지 확인한다.

Milestone 2의 경우 read/write가 혼합된 workload를 수행하여 성능 및 병목이 제거되었는지 확인한다.

## 6 SUBMISSION

과제 제출은 GitLab으로 하며, project4 디렉토리를 생성하여 과제를 제출한다.

(build 결과물이 GitLab에 올라가지 않도록 주의)

보고서는 GitLab의 Wiki를 통해 제출한다. project4 라는 이름의 page를 생성하여 작성한다.