Database Project (Fall 2023)

Homework #1 (Due date: Oct 4)

Student ID 2020315798

Name Choi Jin Woo

1. Write the system setup of your environment. Fill in the blank below.

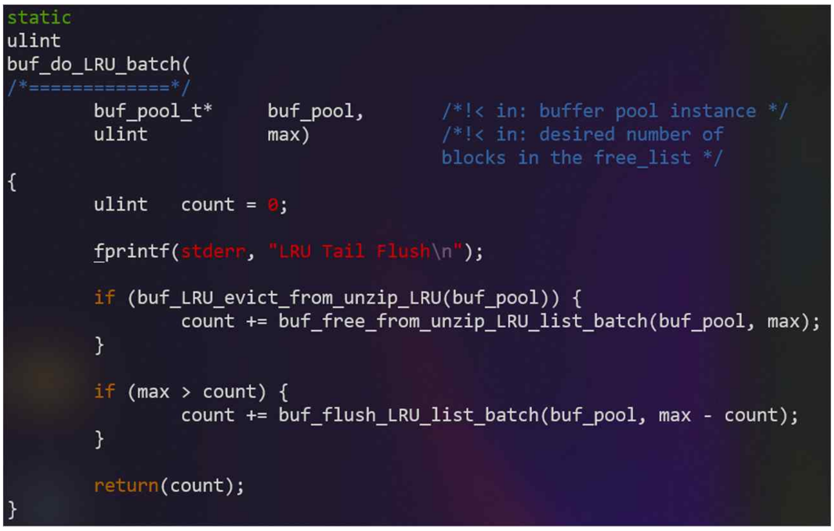
Type	Specification
OS	Ubuntu 18.04
CPU	Intel(R) Core(TM) i5-8265u CPU @ 1.60GHz (2 cores)
Memory (RAM)	6 GB
Kernel	Linux version 5.4.0-84-generic
Data Device (Optional)	
Log Device (Optional)	

2. Write the benchmark setup of your environment. Fill in the blank below.

Type	Specification
DB Size	1GB
Buffer Pool Size	100M, 300M, 500M
Benchmark Tool	tpcc-mysql
Runtime	1200s
Connections	8

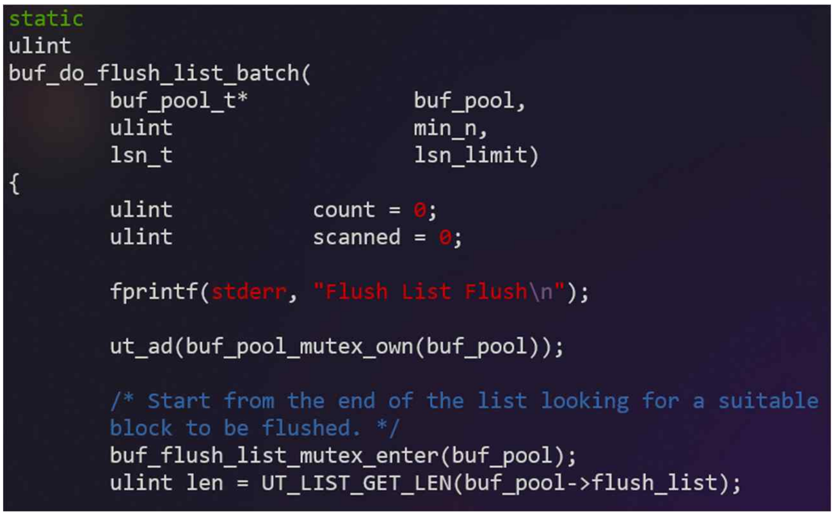
3. Describe the part of the MySQL source code where you added the code to print the flush types, with a screenshot.

LRU list flush:



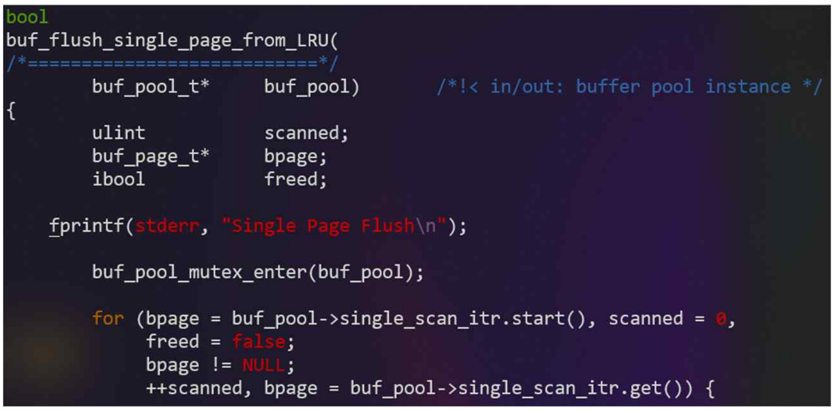
```
static
ulint
buf_do_LRU_batch(
/*=============*/
        buf_pool_t*     buf_pool,       /*!< in: buffer pool instance */
        ulint           max)            /*!< in: desired number of
                                        blocks in the free_list */
{
        ulint   count = 0;

        fprintf(stderr, "LRU Tail Flush\n");

        if (buf_LRU_evict_from_unzip_LRU(buf_pool)) {
                count += buf_free_from_unzip_LRU_list_batch(buf_pool, max);
        }

        if (max > count) {
                count += buf_flush_LRU_list_batch(buf_pool, max - count);
        }

        return(count);
}
```

Flush list flush:



```
static
ulint
buf_do_flush_list_batch(
        buf_pool_t*             buf_pool,
        ulint                   min_n,
        lsn_t                   lsn_limit)
{
        ulint           count = 0;
        ulint           scanned = 0;

        fprintf(stderr, "Flush List Flush\n");

        ut_ad(buf_pool_mutex_own(buf_pool));

        /* Start from the end of the list looking for a suitable
        block to be flushed. */
        buf_flush_list_mutex_enter(buf_pool);
        ulint len = UT_LIST_GET_LEN(buf_pool->flush_list);
```

Single Page flush:



```
bool
buf_flush_single_page_from_LRU(
/*===========================*/
        buf_pool_t*     buf_pool)       /*!< in/out: buffer pool instance */
{
        ulint           scanned;
        buf_page_t*     bpage;
        ibool           freed;

    fprintf(stderr, "Single Page Flush\n");

        buf_pool_mutex_enter(buf_pool);

        for (bpage = buf_pool->single_scan_itr.start(), scanned = 0,
             freed = false;
             bpage != NULL;
             ++scanned, bpage = buf_pool->single_scan_itr.get()) {
```

4. Assume that you have a database with a size of 10 warehouses. Varying the buffer size to 10%, 30%, 50% of database size, calculate the ratio of flush types. You should show the ratio of each flush types and explain about the result.

Buffer Size	10%	30%	50%
LRU List flush (%)	5.98% (3059)	42.33% (3458)	46.46% (3631)
Flush List flush (%)	7.26% (3712)	44.56% (3640)	51.72% (4043)
Single Page flush (%)	86.76% (44380)	13.11% (1071)	1.82% (142)

Explanation)

As the buffer size increases, there is a tendency for the LRU List flush and Flush List flush ratios to increase. This is because a larger buffer can cache more data, helping to defer disk flushing operations. With an increase in buffer size, there is a decrease in the ratio of Single Page flush. This is because a larger buffer holds more data in memory, reducing the need to flush individual pages to disk.

5. It is commonly known that single-page flushes cause performance degradation. Describe the reason for this.

It is commonly known that single-page flush causes performance degradation for InnoDB in MySQL. This is because of inefficient disk utilization, disk I/O overhead, and increased write amplification.

First, there is an inefficiency of disk utilization. Writing individual pages to the disk may be inefficient in case of disk space utilization. As disk blocks are normally written in large chunks, writing small pages individually may result in wasted space on the disk, which can also cause a fragmentation.

Second, there will be a disk I/O overhead. As dist I/O is relatively slow compared to the memory operations, frequent disk I/O operations can lead to I/O contention and increased latency.

Last, frequent single-page flush can lead to increased write amplification. This is a ratio of data written to disk compared to the memory. High write amplification can wear out storage faster and diminish its lifespan.

6. Explain the checkpoint, and why is it necessary? Also, what happens when the checkpoint occurs in the MySQL? Explain with the three main lists in the MySQL.

What is the checkpoint)

Checkpoint in MySQL is a process that helps maintain the consistency and durability of the database.

Why is it necessary)

Checkpoint ensures that data changes are safely and efficiently written to the disk. It ensures atomicity, consistency, isolation, and durability of database transactions.

What happens when the checkpoint occurs)

1. Free list

   During checkpoint, InnoDB identifies pages on the Free list which are not needed anymore and reusable. These pages are put to the list of pages which will be flushed to disk as part of the checkpoint.

2. LRU list

   During checkpoint, InnoDB ensures dirty pages, which have been modified, are written to disk. This is necessary to maintain data consistency and persist changes to data.

3. Flush list

   During a checkpoint, InnoDB scans the Flush list to identify and write pages that are ready to be flushed to disk. Also, the pages in the Flush list are normally sorted in the order of their importance to be flushed.