InnoDB
Change Buffering

Davi Arnaut

Davi Arnaut
@davi

MySQL Internals development at LinkedIn

Worked at MySQL 2007-2011

Designed and built Twitter MySQL

Long time Open Source contributor: Apache, Linux kernel, etc.

Overview of InnoDB Change Buffering

The high-level idea

Consists of buffering modifications (insert, delete and purge operations) to non-unique secondary indexes.

Modifications to secondary indexes usually happen in relatively random (primary key) order, potentially causing a lot of random disk I/O operations.

Instead of performing these random I/O operations necessary to read secondary index pages, modifications are cached in a special data structure named the **change buffer**.

Enabling change buffering

System variable innodb_change_buffering

- inserts
- deletes
- purges
- changes (inserts and delete-mark)
- all (default)
- none

```
SET GLOBAL innodb_change_buffering = “…”
SET GLOBAL innodb_change_buffer_max_size = 25;
```

The implementation

A modification is cached when the relevant secondary index leaf page necessary to perform the operation is not in the buffer pool.

When an operation on a secondary index page is buffered, an entry is set on the change buffer bitmap to indicate that changes are pending for that page.

Buffered changes are merged when relevant secondary index pages are read from disk, or periodically and in batches by a background thread.

Change Buffer

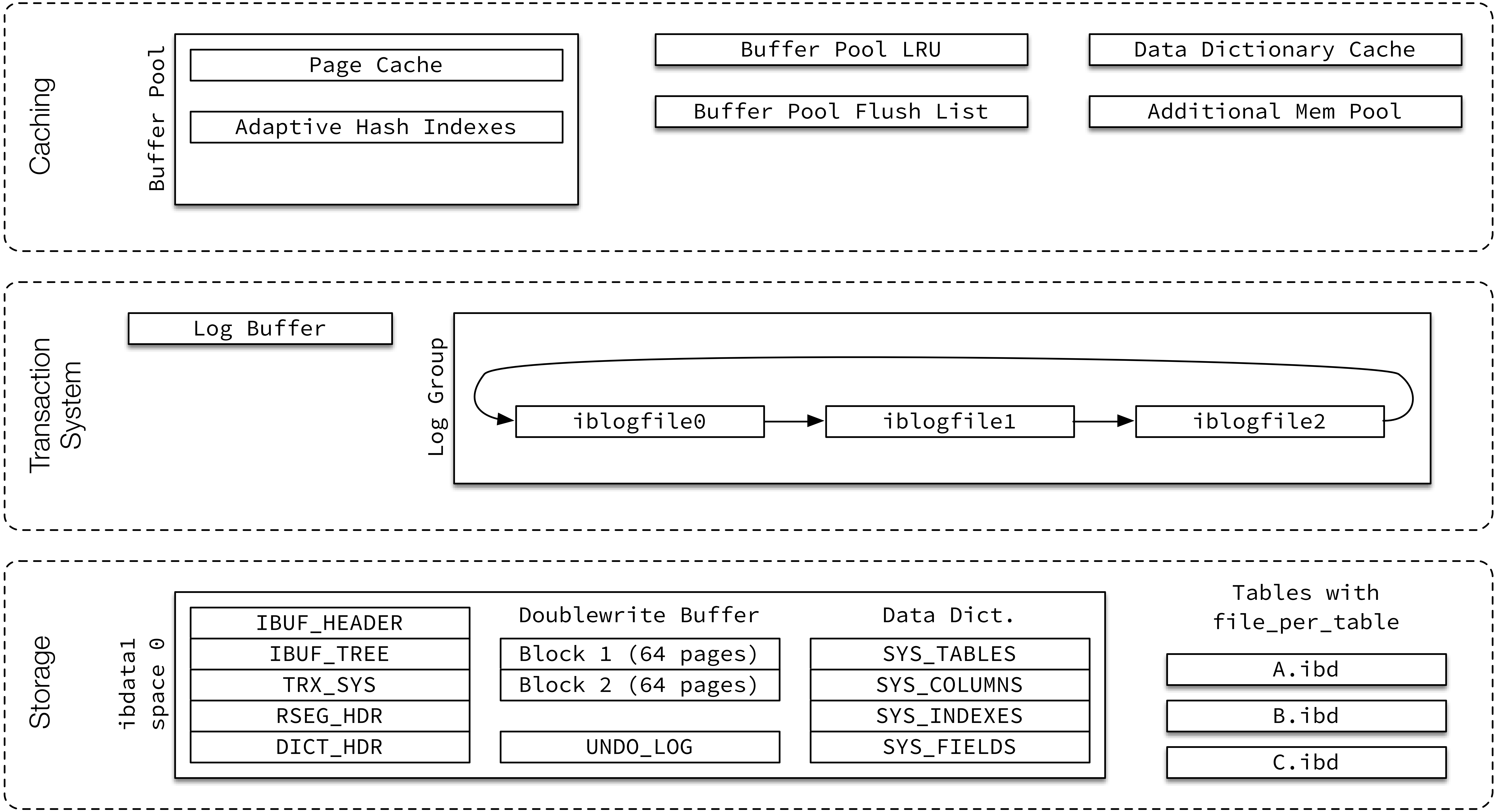
The change buffer, where modifications are cached, is a special table/index stored in the system tablespace. The number of the root page of the change buffer index is 4.

The clustering key is roughly a space ID and page number, which is the location of where the modification would have been made.

Whenever a secondary page index page is read, the change buffer bitmap is checked for pending merges. Otherwise, the change buffer index is randomly traversed for merges.

Physical Structures

High-level Overview



Caching

Buffer Pool

Page Cache

Adaptive Hash Indexes

Buffer Pool LRU

Buffer Pool Flush List

Data Dictionary Cache

Additional Mem Pool

Transaction System

Log Buffer

Log Group

iblogfile0

iblogfile1

iblogfile2

Storage

ibdata1 space 0

IBUF_HEADER

IBUF_TREE

TRX_SYS

RSEG_HDR

DICT_HDR

Doublewrite Buffer

Block 1 (64 pages)

Block 2 (64 pages)

UNDO_LOG

Data Dict.

SYS_TABLES

SYS_COLUMNS

SYS_INDEXES

SYS_FIELDS

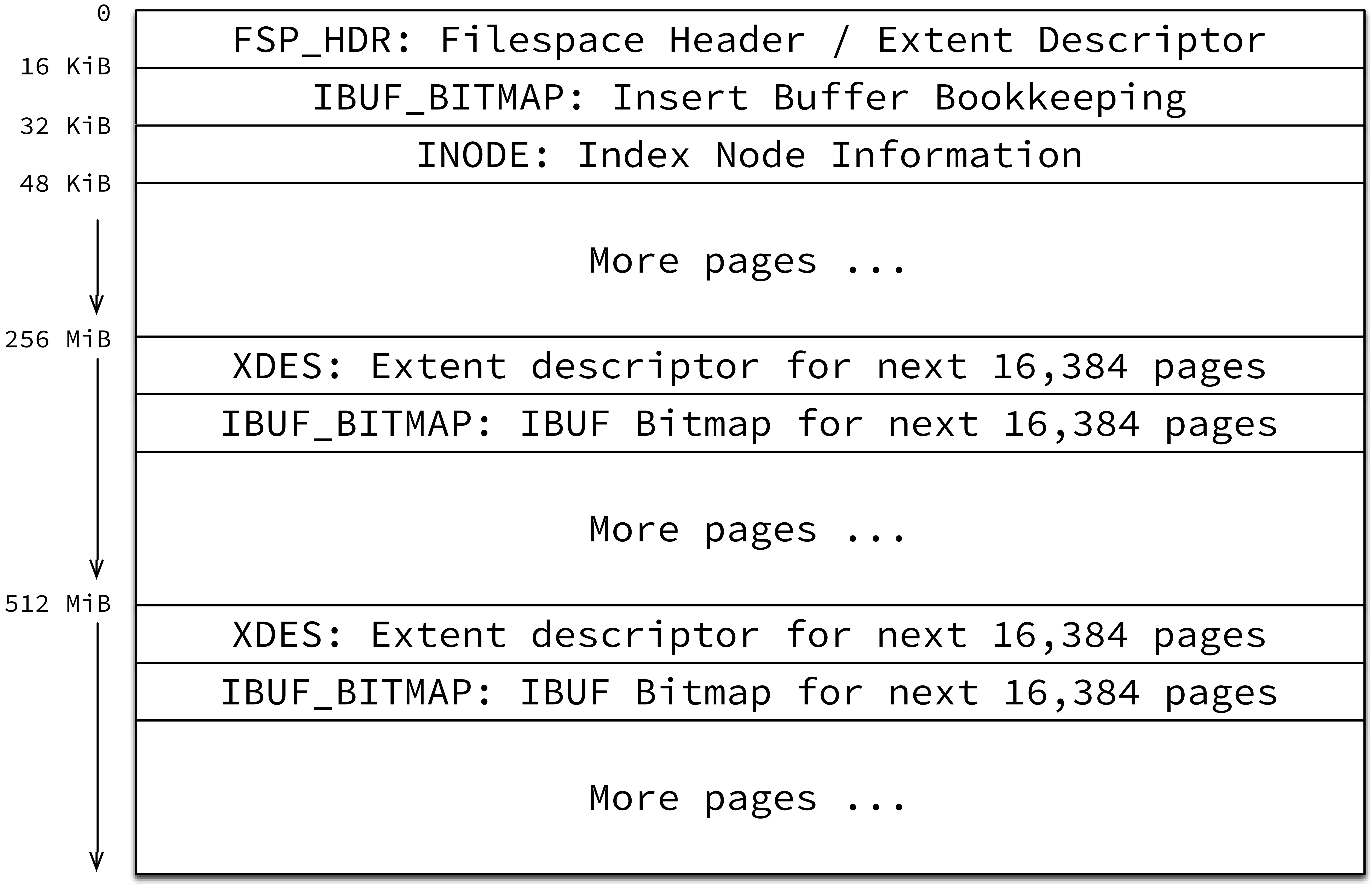
Tables with file_per_table

A.ibd

B.ibd

C.ibd

Space File Overview



0

FSP_HDR: Filespace Header / Extent Descriptor

16 KiB

IBUF_BITMAP: Insert Buffer Bookkeeping

32 KiB

INODE: Index Node Information

48 KiB

More pages ...

256 MiB

XDES: Extent descriptor for next 16,384 pages

IBUF_BITMAP: IBUF Bitmap for next 16,384 pages

More pages ...

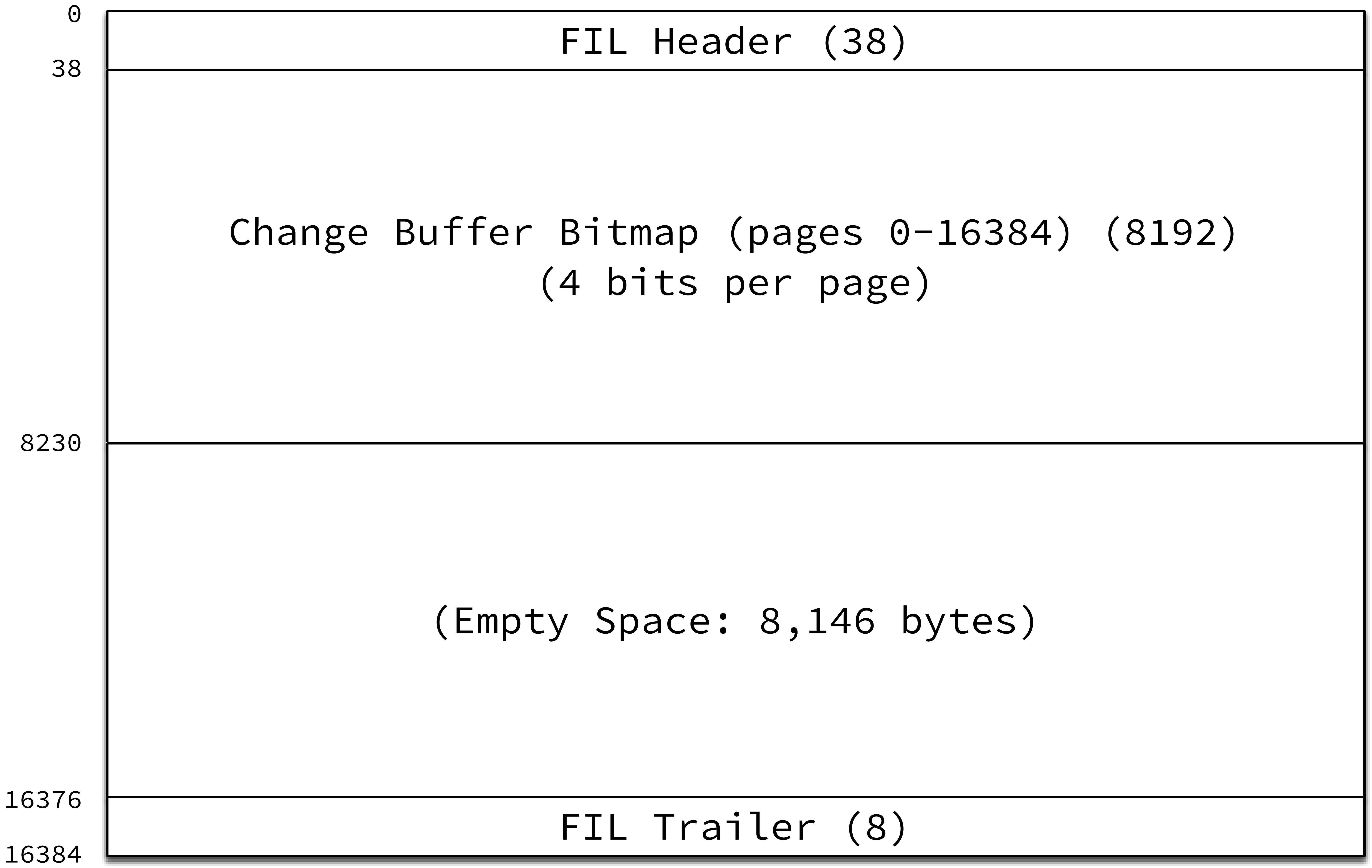
512 MiB

XDES: Extent descriptor for next 16,384 pages

IBUF_BITMAP: IBUF Bitmap for next 16,384 pages

More pages ...

IBUF_BITMAP Overview



0

FIL Header (38)

38

Change Buffer Bitmap (pages 0-16384) (8192)
(4 bits per page)

8230

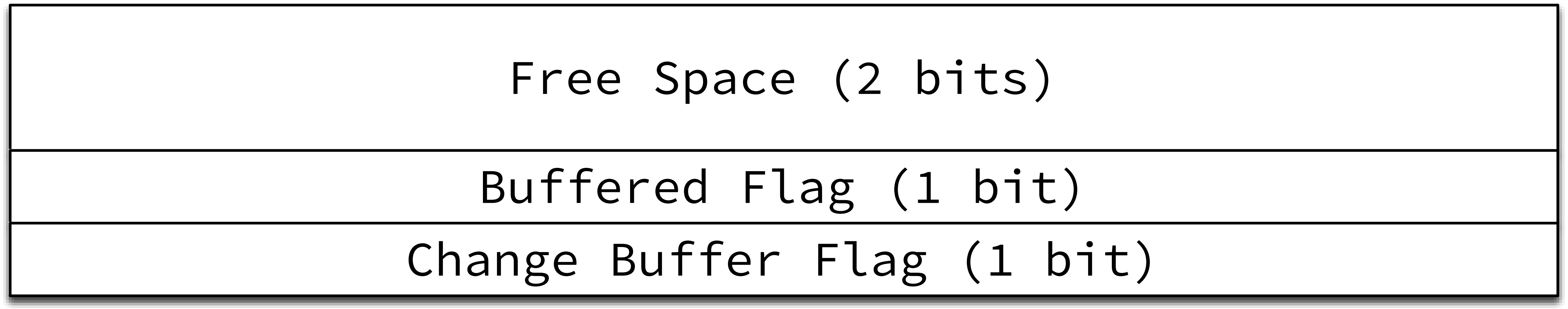
(Empty Space: 8,146 bytes)

16376

FIL Trailer (8)

16384

IBUF_BITMAP Page Entry

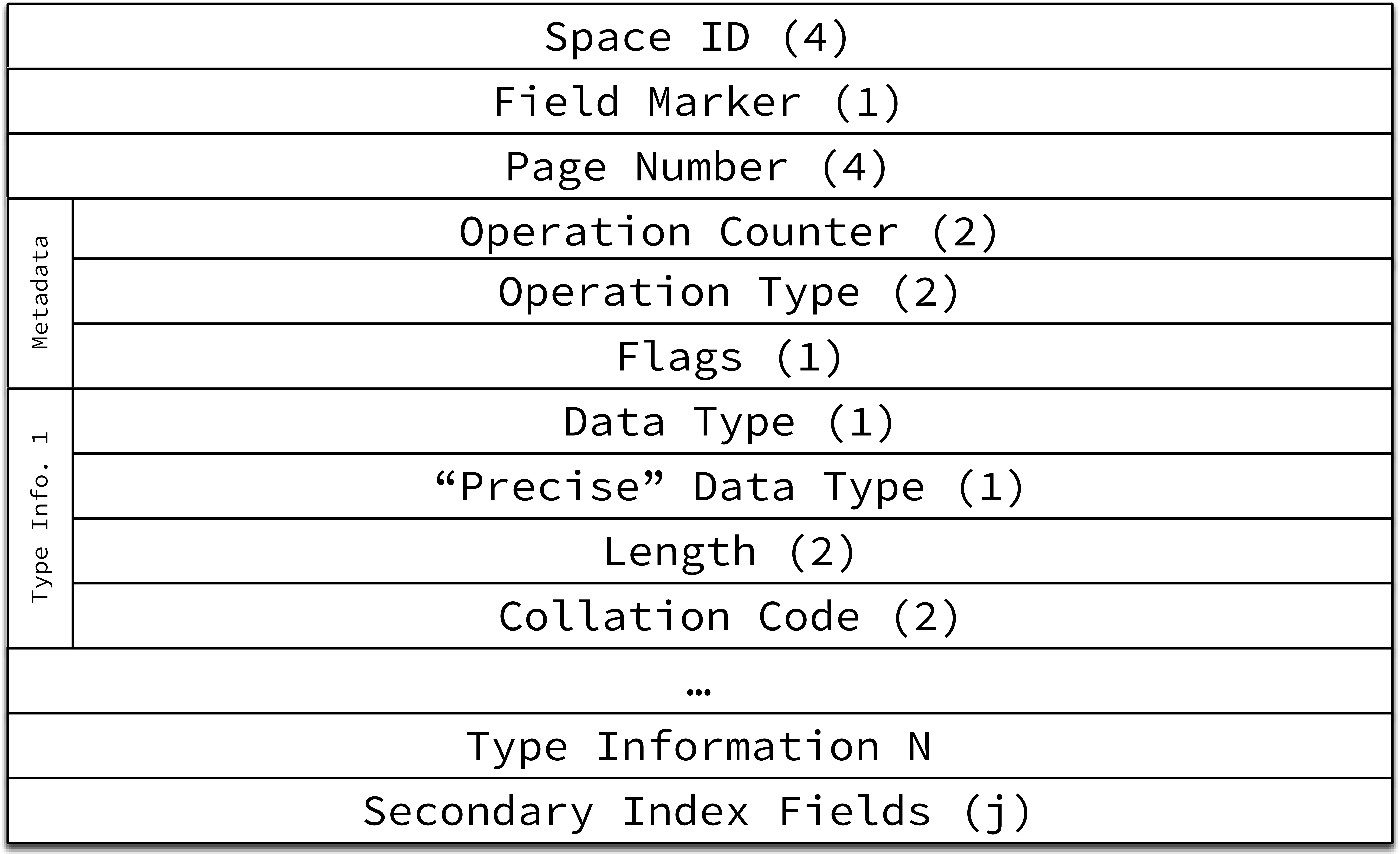


Free Space (2 bits)

Buffered Flag (1 bit)

Change Buffer Flag (1 bit)

Record Format - Change Buffer - Leaf Pages



	Space ID (4)
	Field Marker (1)
	Page Number (4)
Metadata	Operation Counter (2)
	Operation Type (2)
	Flags (1)
Type Info. 1	Data Type (1)
	“Precise” Data Type (1)
	Length (2)
	Collation Code (2)
	…
	Type Information N
	Secondary Index Fields (j)

Problems with Change Buffering

XtraBackup Bug#1366065

Exporting tables is inefficient when backup contains a large (and unrelated) change buffer

Q & A