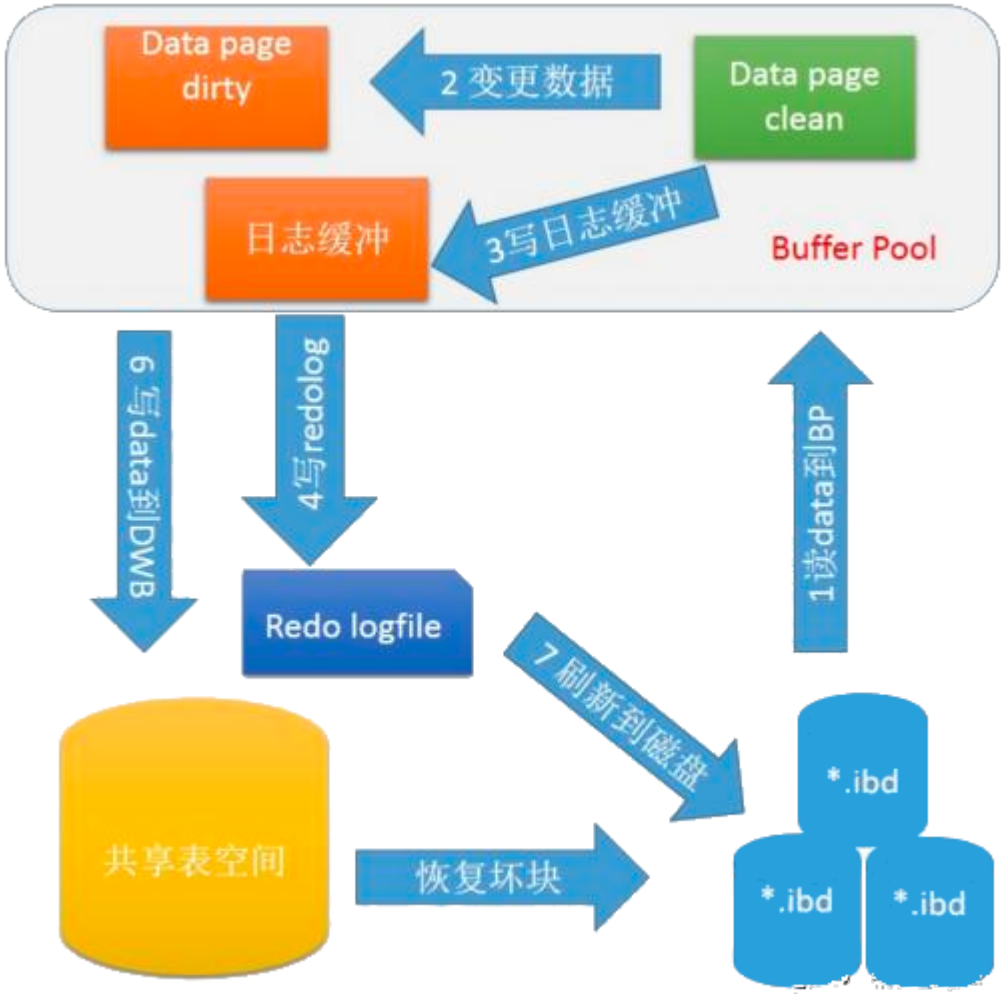
double write

Partial page write 问题

数据库，OS和磁盘读写的基本单位是块，也可以称之为(page size)block size。
InnoDb的page size 为16K,而OS的块则一般为4K;IO块则更小，linux内核要求IO block size<=OS block size，因此一次InnoDB的page刷到磁盘上需要4次OS写文件操作,而如果这四次操作的过程中发生了系统掉电或者奔溃, 只写入了一部分数据,如此就会造成数据的不完整。

如果发生写失效，MySQL可以根据redo log进行恢复。这是一个办法，但是必须清楚地认识到，redo log中记录的是对页的物理修改，如偏移量800，写’aaaa’记录。如果这个页本身已经发生了损坏，再对其进行重做是没有意义的。MySQL在恢复的过程中检查page的checksum，checksum就是检查page的最后事务号，发生partial page write问题时，page已经损坏，找不到该page中的事务号。在InnoDB看来，这样的数据页是无法通过checksum验证的，就无法恢复。即时我们强制让其通过验证，也无法从崩溃中恢复，因为当前InnoDB存在的一些日志类型，有些是逻辑操作，并不能做到幂等。

double write 执行流程



> mysql将内存中的数据刷到磁盘(刷脏)的过程如下:
>
> 1. 使用内存复制将脏数据复制到内存中的double write buffer(2M),
> 2. 通过double write buffer再分2次，每次写入1MB到共享表空间，然后立即调用fsync函数，同步到磁盘上。避免缓冲带来的问题，在这个过程中，doublewrite是顺序写。
> 3. 完成doublewrite写入后，在将double write buffer写入各个表空间文件，这时是离散写入。

double write 性能

在共享表空间上的双重写缓冲区实际上也是一个文件，写DWB会导致系统有更多的fsync操作，而硬盘的fsync性能，所以它会降低mysql的整体性能。但是并不会降低到原来的50%。这主要是因为：

1）double write是一个连接的存储空间，所以硬盘在写数据的时候是顺序写，而不是随机写，这样性能更高。

2）将数据从双写缓冲区写入到真正的segment中的时候，系统会自动合并连接空间刷新的方式，每次可以刷新多个页面；

如果页面大小是16k，那么就有128个页面（1M）需要写，但是128个页面写入到共享表空间是1次IO完成，则doublewrite写入是1 + 128次。其中128次是写数据文件表空间。

doublewrite写入是顺序的，性能开销转化为量，通常5%-25%的性能影响。

如何恢复

写缓冲区失败

如果是写双写缓冲区本身失败，那么这些数据不会被写入磁盘，InnoDB此时会从磁盘加载原始数据，然后通过InnoDB的事务日志来计算出正确的数据，重新写入到双写缓冲区。

写磁盘失败

- 读取页面尾部的校验和，如果校验和不匹配则证明该页面已经损坏
- 如果页面损坏则从double write buffer中找到该页的一个副本，复制到表空间，应用redo log进行恢复。