但是参数 slave_parallel_workers 设置过小。当然设置的工作线程个数应该结合服务器的配置和当前服务器的负载等因素综合考虑，在 5.1 节中，我们会看到线程是 CPU 调度的最小单位。

- Waiting for Slave Worker queue

由于工作线程的任务队列已满，所以协调线程处于等待状态。这种情况前面介绍过，是由于一个事务包含了过多的 Event，并且工作线程应用 Event 的速度赶不上协调线程分配 Event 的速度，导致积压的 Event 超过了 16384 个。

另外，实际上还有一种等待状态，也就是“Waiting for Slave Workers to free pending events”，它由所谓的 big event 造成。源码中将 big event 描述为“event size is greater than slave_pending_jobs_size_max but less than slave_max_allowed_packet”。个人认为其出现的可能性不大，了解即可，可以在 append_item_to_jobs 函数中找到答案。

下面对日志中的输出进行详细解释，如表 4-1 所示。

表 4-1

指　标	解　释
seconds elapsed	整个分配过程消耗的时间，单位为秒，超过 120 秒会出现这个警告日志
events assigned	本工作线程分配的 Event 数量
worker queues filled over overrun level	本工作线程任务队列中 Event 的个数大于源码变量 mts_slave_worker_queue_len_max 的 90%。当前硬编码为大于 14746
waited due a Worker queue full	本工作线程任务队列已满的次数。当前硬编码为大于 16384
waited due the total size	big event 出现的次数
waited at clock conflicts	由于不能并行回放，协调线程等待的时间，单位为纳秒
waited (count) when Workers occupied	由于没有空闲的工作线程而等待的次数
waited when Workers occupied	由于没有空闲的工作线程而等待的时间

这个日志基本覆盖了前面我们讨论的全部可能性。我们再看看案例中的日志，waited at clock conflicts=91895169800 大约为 91 秒。在 120 秒的等待中大约有 91 秒是因为不能并行回放而造成的，很明显应该考虑是否有大事务的存在。

4.1.4　并行回放判定一例

图 4-2 是主库使用 WRITESET 方式生成的 binary log 片段，主要观察 last commit 和 seq number，我们通过这种分析来熟悉分配工作线程的过程。