```
When a cpu is offline, blk_mq_hctx_notify_dead() is called once for each
hctx for the offline cpu.

While blk_mq_hctx_notify_dead() is used to splice all ctx->rq_lists[type]
to hctx->dispatch, it never checks whether the ctx is already mapped to the
hctx.

For example, on a VM (with nvme) of 4 cpu, to offline cpu 2 out of the
4 cpu (0-3), blk_mq_hctx_notify_dead() is called once for each io queue
hctx:

1st: blk_mq_ctx->cpu = 2 for blk_mq_hw_ctx->queue_num = 3
2nd: blk_mq_ctx->cpu = 2 for blk_mq_hw_ctx->queue_num = 2
3rd: blk_mq_ctx->cpu = 2 for blk_mq_hw_ctx->queue_num = 1
4th: blk_mq_ctx->cpu = 2 for blk_mq_hw_ctx->queue_num = 0

Although blk_mq_ctx->cpu = 2 is only mapped to blk_mq_hw_ctx->queue_num = 2
in this case, its ctx->rq_lists[type] will however be moved to
blk_mq_hw_ctx->queue_num = 3 during the 1st call of
blk_mq_hctx_notify_dead().

This patch would return and go ahead to next call of
blk_mq_hctx_notify_dead() if ctx is not mapped to hctx.
```