碩論相關：

* 如果所有的write request在write buffer hit，那代表當min benefit block>current block benefit時，也要將min benefit block踢掉(將write buffer中min benefit block踢掉，不理會該block的benefit是否比current block來的大)
  + 產生問題(A)：會發生很多block放入write buffer又立刻被踢掉的狀況
    - 假設當下write buffer內min benefit block是block 5，benefit 30，current block 6，benefit=25
    - 此時會將block 5踢掉(因為是min)，將block 6寫入，但其實block 5更有存在價值
    - 問題A連帶的問題：因為將block 5踢掉，block 6寫入，此時會造成block 6變成min，下個request到來時立刻就會被踢掉，造成很低的duration
    - 問題A連帶的問題：而不該被踢掉的min block不久後有機會再度被存取，因此有可能再度被寫入write buffer，然後過一段時間再度被踢掉，此時會出現多個相同block number的duration，該拿哪個當作duration value?
  + 實際狀況：write buffer會再還有一定空間的時候就開始踢資料，因此實際上不會有以上狀況，min block會是block 6
* 思考問題
  + Q：為什麼所有write request都要放入write buffer?
  + Answer：分成兩種狀況(write buffer is full, write buffer not full)
    - Write buffer is not full
      * Write hit🡪Overwrite
      * Write miss🡪Write to a new block
      * 如果write buffer還有空間，data一定會寫入write buffer，因為寫入write buffer一定比寫入SSD快
    - Write buffer is full
      * Write hit🡪overwrite
      * Write miss🡪select victim block and flush to SSD
      * 雖然在write buffer滿了的時候，從write buffer踢下去和直接寫入SSD看似花的時間差不多，但其實選擇寫入write buffer(踢一個block下去)有更多好處：
        + 延長SSD壽命(write-intensive data只須寫回SSD一次，不會反覆寫入消耗SSD的寫入次數)
        + 對於非write-intensive data，若寫入write buffer，因為少量寫入的資料，通常是循序存取，因此很可能之後會將整個block寫滿，之後踢到SSD時可減少GC的時間，且【少量寫入】不代表【少量讀取】，如果該block是大量讀取，則放在write buffer可大幅改善讀取的效能
        + 反之，如果將少量寫入的資料寫進SSD，則會遇到以下問題：

不能保證後續request在FTL轉換後，會寫入同個block，因此有可能GC的時間不會減少

如果該block是read-intensive，會造成每次讀取都要到硬碟抓資料，效能會很慢很多

* + 結論：所有的write request都放入write buffer對效能來說會比較好。
  + Q：為什麼只看write hit ratio、total hit ratio不夠(不準)?
  + Write hit ratio
    - 高或低都沒影響，反正就是不存在，就寫進write buffer，存在，就overwrite
  + Read hit ratio
    - 整個trace中一定包含cold data, hot data，只有當下已經在write buffer內的data被read時，read hit ratio才會提高，但不太可能整個一直反覆對write buffer內的資料讀取。
    - 因為trace中一定有少量讀取的request(cold data)，當存取到那類型的request(cold data)，不論write buffer內部是怎麼擺放，read hit ratio一定會下降，原因如下：
      * 假設運氣好，write buffer內存在cold data，當read request(cold data)來能夠read hit，但這筆data就佔掉了一個page或者是多個page的空間，讓hot data無法放入(變成這個cold data會讓之後進來的hot data miss)，因此之後一定要將這個block踢掉。
      * 反過來說，如果write buffer內都是hot data，那read request(cold data)來時，就會miss，得特地從SSD讀上來
  + 因此只看hit ratio來當成判斷依據，是不準確的，重點在於能否在request來的時候，確保write buffer內有該筆資料(也就是需要同時考量recency & frequency)
  + 還需要考量的因素
    - Flush block count
      * 這個因素，代表cold data是否集中在某些block，越是集中，踢下去的block就能夠越少，也有越多空間可以給其他hot data使用。
      * 因此，在維持同樣hit ratio的情況下，有可能會有不同的flush block count，cold data越是集中在某些block，則flush block count越小，反之，就越大。
* Disksim 相關
  + Syssim其實就是子系統，也就是request進入disksim前，會先進入syssim