Final project report

Team 9

金子荃 104062333

陳志祥 104062330

翁挺傑 104062120

1. 題目:  
   2018 [VLDB’18] Contention-Aware Lock Scheduling for Transactional Database
2. 作者:
3. Boyu Tian
4. Jiamin Huang
5. Barzan Mozafari
6. Grant Schoenebeck
7. 論文研究要點:

Lock Management在現今的資料庫系統中扮演著重要的腳色。最直觀的方式是採用FIFO，意即先索取的transaction越早得到lock。然而，對於資料庫而言，他效率取決於資源的分配，因此FIFO在大多數情況下無法得到最好的效能。故該篇論文的作者想透過不同的演算法，實作出多樣的優先權判定方式，將lock給予transaction，比如MLF、MBLF、LDSF與DDF等。

1. 實作摘要:

對於所有演算法，皆需要以一個Priority Queue，儲存所有向某個Object的lock的transaction，而其優先權判定則是各個演算法所定義。除此之外，因notifyAll將會把所有transaction呼喚醒，所以我們只將queue的第一個transaction取出，其餘將會繼續沉睡。

1. FIFO:

當transaction向某個Object索取其lock，將會把它存進queue當中，並依照排入時間，依序給予其lock。

1. VATS:

同樣是以時間作為優先權的判斷基準，但與FIFO不同，VATS所著重的是「transaction的年齡」，也就是從進入系統的時間點。因此越早進入系統的transaction將會有更高的優先權得到lock。

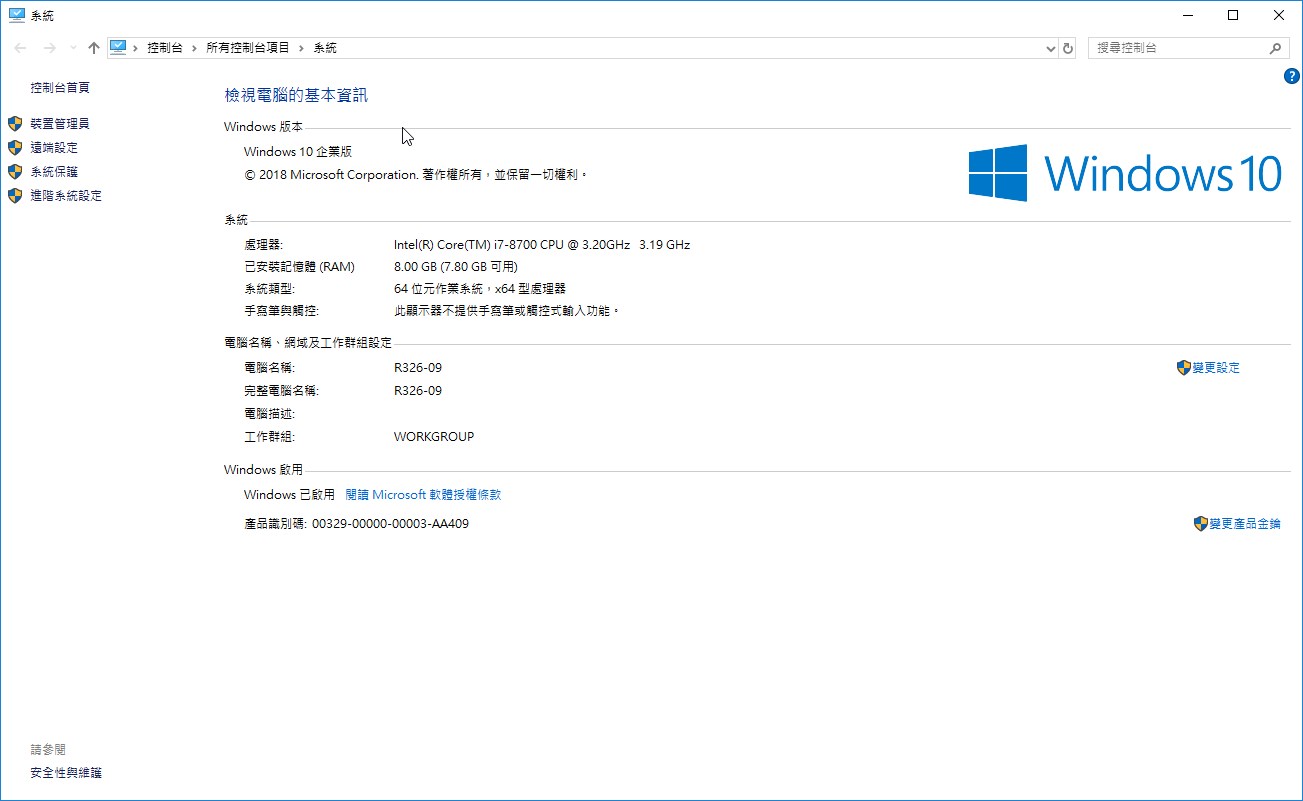
1. MLF:

以擁有的lock數為判斷基準，當一個transaction得到任一個object的lock，他的優先權就會逐步上升。因此，我們以一個Map<Long, Integer>為儲存每個transaction所擁有的lock數。

1. MBLF:

不同於MLF，只在乎一個transaction拿了多少lock；MBLF的優先權判定是依據一個transaction所有lock中，有多少會影響到其他transaction的運行。因此，若有一個transaction欲索取object的lock，但此時需要等待，那我們將會把目前擁有object lock的transaction，他的阻擋數加一。但需要注意的是，一旦阻擋數加一，除非該transaction釋放lock，否則事後將不會因那個object而再次增加阻擋數。

1. LDSF:
2. DDF:
3. 實驗數據:
4. Environment:



1. Test Bench(Micro):
2. RW\_TX\_RATE = 0.5
3. RTE = 10
4. Conflict Rate = 0.001
5. Experiments plot:
6. RW\_TX\_RATE
7. RTE
8. Conflict Rate
9. 分析結果:
10. 實作的困難:
11. 結論: