

2021 NYCU OS HW3 report

Question	Answer
<p>Q1.</p> <p>Briefly describe your design for the sorting algorithm, merge function, the thread management.</p> <p>Also, describing the number of sort threads and number of merge threads in the Multi-thread program.</p>	<p>我把 partition_num、thread_num 用 define 定義，之後要改的時候改一個地方就好比較方便。</p> <p>我的 partition_num 要是 2 的次方。</p> <ol style="list-style-type: none">1. 把陣列切成 partition_num 個，切的方式是紀錄每個 partition 的起始位置和結束位置，每個 partition 彼此互斥。2. 因為 partition 合併後會需要繼續合併，所以也把合併後的位置紀錄，一直紀錄直到只剩一個 partition。3. 對每個 partition 做 bubble sort，這邊可以用 multithread 加速。4. 把每個 sort 好的 partition 兩兩 merge 起來，一直重複 merge 直到剩下一個 partition，這邊也可以用 multithread 加速。 <p>multithread 加速的方法：把 thread 要做的那個 function 的參數填在二維陣列中，根據 thread num 決定填在二維陣列的哪一個 row，再建立 thread_num 個 thread，一次傳一個 row 的參數進去，在 main thread 中 join 這 thread_num 個 thread。</p> <p>我本來是在 for 填好一個 thread 要做的那個 function 的參數就建立一次 thread，但好像會因為 main thread 繼續執行而把參數覆寫，create 出的 thread 拿到的參數有時候就會出錯，所以我才換成填在二維陣列中。</p>

<p>Q2.</p> <p>Show the fastest time acceleration between single-thread and multi-thread. (Take screenshots of the time between single-thread and multi-thread)</p>	<p>Multithread: 4 threads, 8 partitions</p> <pre>time ./0816147_MT_best < test/input/input2.txt > test/output/output2_MT_best.txt real 1m52.141s user 7m20.468s sys 0m0.108s diff test/answer/answer2.txt test/output/output2_MT_best.txt</pre> <p>Single thread: 1 thread, 8 partitions</p> <pre>time ./0816147_ST < test/input/input2.txt > test/output/output2_ST.txt real 7m17.223s user 7m14.970s sys 0m0.225s diff test/answer/answer2.txt test/output/output2_ST.txt</pre> <p>437/112 = 3.9，將近 4 倍的加速，與預期滿接近的。</p>
<p>Q3.</p> <p>You need a brief description of the best multi-threads and worst multi-threads methods.</p> <p>The content includes the number of threads used and the way of partitioning, comparing the difference in time, and taking the screenshot between two multi-thread results.</p>	<p>Multithread best: 4 threads, 8 partitions</p> <pre>time ./0816147_MT_best < test/input/input2.txt > test/output/output2_MT_best.txt real 1m52.141s user 7m20.468s sys 0m0.108s diff test/answer/answer2.txt test/output/output2_MT_best.txt</pre> <p>Multithread worst: 4 threads, 4 partitions</p> <pre>time ./0816147_MT_worst < test/input/input2.txt > test/output/output2_MT_worst.txt real 3m48.834s user 14m51.685s sys 0m0.231s diff test/answer/answer2.txt test/output/output2_MT_worst.txt</pre> <p>228/117=1.948，在這個 case，2 倍的 partition 有將近 2 倍的加速。</p> <p>我 partition 的方式一樣，只是切的數量不一樣而已，程式碼也只有改最前面 define partition_num 的值而已。</p>
<p>Q4.</p> <p>What did you learn from doing hw3?</p>	<p>學到了如何使用 thread 和 mutex lock，本來只有上課知道概念而已，實際操作後就更熟悉了。</p> <p>且體會到 main thread 和 create 出來的 thread 可能會因為 context switch 而有非預期的結果，且這種 bug 不是每次都會出現，所以讓我非常難 debug。</p> <p>過程中也為了不要讓他因為 context switch 而有問題所以用到 mutex lock，但有個地方感覺不該有問題卻一直找不到錯，所以最後改用二維陣列放參數，也就不需要 mutex lock 了。</p> <p>所以這次讓我真正意識到 multithread 在寫的時候真的要非常小心，不然會 debug 到天荒地老都找不出問題。</p>