Assignment 2 – Quicksort

วิเคราะห์ปัญหา

การทำ quicksort ประกอบไปด้วยการเลือก pivot แล้วทำการ partition ข้อมูลให้ทางซ้ายของ pivot น้อยกว่า pivot และให้ข้อมูลทางขวาของ pivot มากกว่า pivot แล้วทำการ recursion ลงไปทำงานในส่วนที่เล็กลงทั้งสองส่วนไปเรื่อย ๆ จนครบ และในได้ใช้ Hoare’s partition ซึ่งใช้ partition สองตัวขยับเข้าหากันเนื่องจากสามารถทำงานได้เร็วกว่าแบบใช้ pivot เดียว

จากนั้นจึงได้ทำการเรียงลำดับขั้นตอนการทำงานทั้งหมดได้ดังนี้ โดยให้ n เป็นจำนวนข้อมูลและ p เป็นจำนวน process

1. Read input file ใช้เวลาเป็น O(n) เป็นการทำงานแบบ sequential
2. Communication to process เป็น overhead ในการทำงานแบบ parallel
3. Computation ใช้เวลาเป็น O(nlogn/p) เป็นการทำงานแบบ parallel
4. Communication from process เป็น overhead ในการทำงานแบบ parallel
5. Merging from process ใช้เวลาเป็น O(np2) เพื่อรวม array ขนาด n ตัว จำนวน p array เป็น array เดียว เป็นการทำงานแบบ sequential
6. Write output file ใช้เวลาเป็น O(n) เป็นการทำงานแบบ sequential

พบว่าส่วนที่เป็น sequential ของงานนี้คือข้อ 1, 5, 6 และส่วนที่เป็น parallel ของงานนี้คือข้อ 4 ส่วนการส่งข้อมูลไปกลับ process อื่น ๆ นั้นพบว่าใช้เวลาน้อยมาก จึงตัดออก

จะได้ว่า และ

สร้าง model

ดังนั้น เมื่อใช้ Amdahl’s law จะได้ว่า

speedup

และ efficiency

สำหรับ analytical model จะต้องหาเวลาทั้งหมดที่ใช้ในการทำงานแบบ sequential และเวลาที่ใช้ในการทำงานทั้งหมดในการทำงานแบบ parallel

ในการทำงานแบบ sequential โปรแกรมจะต้องอ่านข้อมูล ใช้เวลา O(n) ทำการคำนวณ ใช้เวลา O(nlogn) และเขียนข้อมูลไปยังไฟล์ ใช้เวลา O(n) ดังนั้นจะได้ว่า

สำหรับในการทำงานแบบ parallel โปรแกรมยะทำงานตาม 7 ขั้นตอนข้างต้น ดังนั้นจะได้ว่า

เนื่องจากเวลาที่ใช้ในการส่งและรับข้อมูลระหว่าง process น้อยมาก จึงตัดออก

ดังนั้น เมื่อใช้ analytical model จะได้ว่า

speedup

และ efficiency

ซึ่งเมื่อนำมาสร้างกราฟเทียบกับ Ideal case และผลจากการทำงานจริงแล้วจะได้ดังนี้

|  |  |  |  |
| --- | --- | --- | --- |
| Type | Processor | Speedup | Efficiency |
| Ideal | 1 | 1.000000 | 1.000000 |
|  | 2 | 2.000000 | 1.000000 |
|  | 4 | 4.000000 | 1.000000 |
|  | 8 | 8.000000 | 1.000000 |
|  | 16 | 16.000000 | 1.000000 |
| Amdahl's law | 1 | 1.000000 | 1.000000 |
|  | 2 | 1.488620 | 0.744309 |
|  | 4 | 1.221250 | 0.305313 |
|  | 8 | 1.037800 | 0.129725 |
|  | 16 | 1.005210 | 0.062825 |
| Analytical | 1 | 0.961437 | 0.961437 |
|  | 2 | 1.427450 | 0.713726 |
|  | 4 | 1.050510 | 0.262627 |
|  | 8 | 0.362028 | 0.045254 |
|  | 16 | 0.096100 | 0.006006 |
| Actual | 1 | 1.000000 | 1.000000 |
|  | 2 | 1.403245 | 0.701623 |
|  | 4 | 1.522008 | 0.380502 |
|  | 8 | 1.754877 | 0.219360 |
|  | 16 | 1.575206 | 0.098450 |

จะพบว่า โมเดลที่สร้างมานั้น underestimate เพราะเมื่อลองคำนวณด้วยโปรแกรมจริงแล้วสามารถ speedup ได้มากกว่าโมเดล เนื่องจากมีการ optimize ส่วนต่าง ๆ ของโปรแกรม เช่น การอ่านและเขียนไฟล์เป็น string ผ่าน buffer แล้วนำมาแปลงเป็นตัวเลขด้วยฟังก์ชั่นแปลงที่เขียนเอง การใช้ openMP ในการคำนวณ quicksort ทำให้สามารถวนลูปได้เร็วขึ้น การเรียกใช้ openMP จากลูปนอกสุดเพื่อลดการสร้างและทำลาย thread เป็นต้น