Homework Report

ข้อที่ 1 Intro to Parallel Programming

วิธีการที่ใช้

ใช้วิธีการ Quadratic Sieve Algorithm โดยมีหลักการคือ ถ้าเราสามารถ หา X, Y โดยที่ X != ±Y และทำให้ $X^2 \equiv Y^2 \ (mod\ n)$ เมื่อ n คือ ตัวเลขที่เราต้องการแยกตัวประกอบ เราจะได้ว่า gcd(n, Y - X) และ gcd(n, Y+X) จะเป็นตัวประกอบของ โดยจากวิธีของ [1] ได้เสนอการ ประมาณค่า X, Y ไว้ดังนี้ ให้ $f(x) = Ax^2 + 2Bx + C\ mod(n)$ และคำนวณหา A, B, C โดย

$$A = D^2 \text{ เมื่อ } D \text{ คือจำนวนเฉพาะที่ไม่เป็นตัวประกอบของ } f(x)$$

$h_0 \equiv \ n^{\frac{D-3}{4}} \text{ (mod D)}, h_1 \equiv \ n^{\frac{D+1}{4}} \ (mod\ D), h_1^2 \equiv \ nn^{\frac{D-1}{2}} \ (mod\ D)\ , h_2 \equiv (2h_1)^{-1}(\frac{n-h_1^2}{D}) \ (mod\ D)$

$$B \equiv \ h_1 + h_2 D \ (mod\ D)$$

$$C = \frac{B^2 - A}{n}$$

แล้วจึงนำชุด ของ f(x) ที่มีตัวประกอบจำนวนเฉพาะอยู่ในเซ็ตของ Factor Base มาหา X^2, Y^2 ที่เป็นไปได้โดยใช้ Gaussian Elimination ใน GF(2) เมื่อได้ค่า X^2, Y^2 แล้วก็สามารถหา ตัวประกอบได้จาก gcd(n, Y - X)

จากขั้นตอนข้างต้น ในขั้นตอนหาชุดของ f(x) นั้นสามารถทำแบบ parallel ได้ เพราะแต่ละ f(x) นั้นไม่ขึ้นต่อกันและใช้ข้อมูลคือ เซ็ตของ Factor Base, n และ D ซึ่ง Factor Base และ n ไม่เปลี่ยนแปลงตลอดการทำงานอยู่แล้ว จึงออกแบบให้มี process หนึ่ง(Master) หาค่า D และส่งให้ process ที่เหลือ(Slave) หาค่าของ f(x) ที่ตรงตามเงื่อนไขจาก D ที่ ได้รับและส่งค่ากลับให้ Master เป็นคนเก็บ จนกระทั่งหาชุดของ f(x) ได้ตามจำนวนที่ต้องการแล้ว Slave ทั้งหมดจะหยุดทำงานเ หลือแต่ Master ที่นำชุดของ f(x) มาทำ Gaussian Elimination พร้อมสรุปค่าตัวประกอบที่หาได้ และจบการทำงาน

ผลลัพธ์

Test case	Number to factorize
T20	18567078082619935259
T30	350243405507562291174415825999
T40	5705979550618670446308578858542675373983
T45	732197471686198597184965476425281169401188191

ตารางแสดงรายละเอียดของแต่ละ Test case

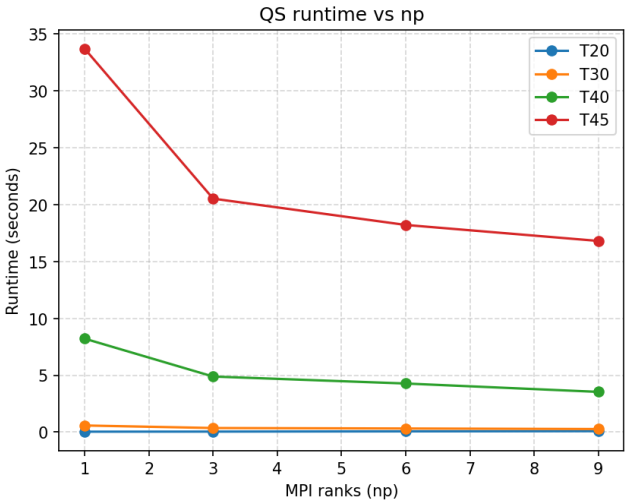
np	T20	T30	T40	T45
1	0.038	0.583	8.214	33.701
3	0.045	0.359	4.886	20.529

6	0.066	0.318	4.273	18.216
9	0.083	0.271	3.538	16.81

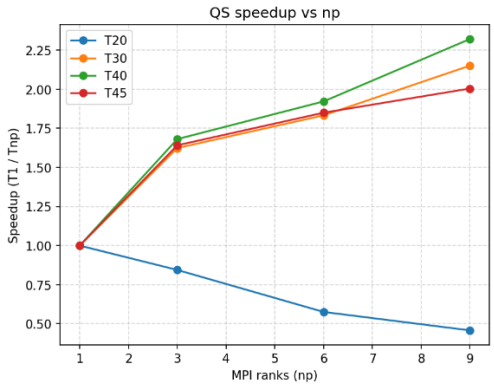
ตารางแสดงจำนวน **process** กับเวลาที่ใช้ในหน่วยวินาทีในแต่ละ **Test case**

np	T20	T30	T40	T45
1	1	1	1	1
3	0.844	1.624	1.681	1.642
6	0.576	1.833	1.922	1.850
9	0.458	2.151	2.322	2.005

ตารางแสดงจำนวน **process** กับจำนวนเท่าที่เร็วกว่าเมื่อเทียบกับการใช้ 1 **process**



กราฟแสดงจำนวน **process** กับเวลาที่ใช้ในหน่วยวินาทีในแต่ละ **Test case**



กราฟแสดงจำนวน **process** กับจำนวนเท่าที่เร็วกว่าเมื่อเทียบกับการใช้ 1 **process**

วิเคราะห์ผลลัพธ์

จาก Amdahl's law ได้ว่า

$$S = \frac{1}{1 - P + \frac{P}{N}}$$

เมื่อ *S* คือ *Speed up*

P คือ สัดส่วนที่ *parallel*

N คือ จำนวน *process* ที่ ทำ *parallel*

จัดรูปใหม่เป็น

$$P = \frac{1 - \frac{1}{S}}{1 - \frac{1}{n}}$$

เมื่อแทน S และ N จากผลลัพธ์ที่ได้ จะได้ P ดังตาราง

N	T20	T30	T40	T45
1	0	0	0	0
3	-0.276	0.576	0.608	0.586
6	-0.884	0.545	0.576	0.551
9	-1.332	0.602	0.640	0.564

ตารางแสดง ค่า P ที่คำนวณได้จากผลลัพธ์

พบว่า นอกจาก ที่ T20 P มีค่าติดลบ เป็นผลมาจาก เลขที่ใช้คำนวณมีขนาดเล็กเกินไปจน ทำให้เวลาที่ speed up น้อยกว่า over head ที่เพิ่มขึ้นของ การ parallel แล้ว test case ที่เหลือ จะมี parallel part เฉลี่ยที่ 0.583 จึงสรุปว่า โปรแกรมนี้จะมี serial part เฉลี่ย = 1 - 0.583 = 0.417

ข้อที่ 2 **Copy on Write**