เสริมเนื้อหาบทที่ 8 การคำนวณแบบขนาน (Parallel Computing)ด้วยบอร์ด Pi3/Pi4

เนื่องด้วยจากกฎของ Amdahl สามารถประยุกต์กับการคำนวณค่า Speedup ของการคูณเมทริกซ์แบบขนานนี้ได้จึงได้เลือกเพิ่มเนื้อหาในส่วน กฎของ Amdahl โดยมีเนื้อหาดังนี้

กฎของ Amdahl(Amdahl's law)

เป็นสูตรที่ทำให้ทฤษฎีเพิ่มความเร็วในความล่าช้าของการดำเนินการของงานที่ได้รับการแก้ไขเป็นภาระงานที่สามารถคาดหวังของระบบที่มีทรัพยากรที่ดีขึ้น โดยตั้งชื่อตามนักวิทยาศาสตร์คอมพิวเตอร์ Gene Amdahl และถูกนำเสนอในการประชุม AFIPS Spring Joint Computer Conference ในปี 1967 ซึ่งกฎของ Amdahl มักใช้ในการคำนวณแบบคู่ขนานเพื่อหาความเร็วตามทฤษฎีเมื่อใช้โปรเซสเซอร์หลายตัว

คำนิยาม (Definition)

คำจำกัดความ: “ การปรับปรุงที่ได้รับในประสิทธิภาพของระบบเนื่องจากการสับเปลี่ยนของส่วนประกอบอย่างใดอย่างหนึ่งนั้นถูก จำกัด ด้วยส่วนของเวลาที่ใช้ส่วนประกอบ”

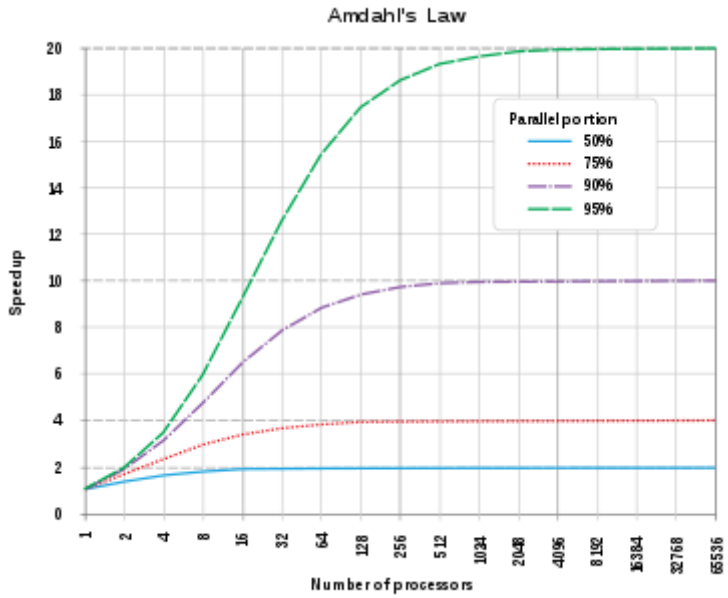
$$S_{\text{latency}}(s) = \frac{1}{(1-p) + \frac{p}{s}}$$

- **S latency** เป็น Speedup ตามทฤษฎีของการปฏิบัติงานทั้งหมด
- **s** คือ Speedup ของส่วนหนึ่งของงานที่ได้รับประโยชน์จากทรัพยากรระบบที่ได้รับการปรับปรุง
- **p** คือสัดส่วนของเวลาดำเนินการที่ส่วนที่ได้รับประโยชน์จากทรัพยากรที่ปรับปรุงแล้วซึ่งเดิมครอบครองอยู่

นอกจากนี้

$$\begin{cases} S_{\text{latency}}(s) \leq \dfrac{1}{1-p} \\ \lim\limits_{s \to \infty} S_{\text{latency}}(s) = \dfrac{1}{1-p}. \end{cases}$$

แสดงให้เห็นว่า Speedup ตามทฤษฎีของการดำเนินงานทั้งหมดเพิ่มขึ้นด้วยการปรับปรุงทรัพยากรของระบบและโดยไม่คำนึงถึงขนาดของการปรับปรุง Speedup ตามทฤษฎีมักจะถูกจำกัดโดยส่วนของงานที่ไม่ได้รับประโยชน์จากการปรับปรุง



ที่มา : [Amdahl's law - Wikipedia]

การเพิ่มความเร็วตามทฤษฎีของเวลาแฝงของการดำเนินการโปรแกรมตามฟังก์ชันของจำนวนโปรเซสเซอร์ที่ดำเนินการ ตามกฎของ Amdahl การเร่งความเร็วจะถูกจำกัดโดยส่วนอนุกรมของโปรแกรม ตัวอย่างเช่น หาก 95% ของโปรแกรมสามารถขนานกันได้ ความเร็วสูงสุดตามทฤษฎีโดยใช้การคำนวณแบบขนานจะเท่ากับ 20 เท่า

ที่มาของสูตร $S_{\text{latency}}(s) = \dfrac{1}{(1-p) + \frac{p}{s}}$ **:**

งานที่ดำเนินการโดยระบบซึ่งมีการปรับปรุงทรัพยากรเมื่อเปรียบเทียบกับระบบเริ่มต้นที่คล้ายคลึงกันสามารถแบ่งออกเป็นสองส่วน :

- ส่วนที่ไม่ได้รับประโยชน์จากการปรับปรุงทรัพยากรของระบบ
- ส่วนที่ได้รับประโยชน์จากการปรับปรุงทรัพยากรของระบบ

เวลาการดำเนินการของงานทั้งหมดก่อนการปรับปรุงทรัพยากรของระบบจะแสดงเป็น **T** รวมถึงเวลาดำเนินการของส่วนที่จะไม่ได้รับประโยชน์จากการปรับปรุงทรัพยากรและเวลาดำเนินการในส่วนที่จะได้รับประโยชน์ เศษส่วนของเวลาดำเนินการของงานที่จะได้รับประโยชน์จากการปรับปรุงทรัพยากรแสดงโดย **p** ส่วนที่เกี่ยวกับส่วนที่ไม่ได้รับประโยชน์จากสิ่งนั้นก็คือ **1-p** จากนั้น :

$$T = (1 - p)T + pT.$$

การดำเนินการในส่วนที่ได้รับประโยชน์จากการปรับปรุงทรัพยากรที่เร่งโดยปัจจัย **s** หลังจากการปรับปรุงทรัพยากร ดังนั้น เวลาดำเนินการของส่วนที่ไม่ได้ประโยชน์จากมันยังคงเท่าเดิม ในขณะที่ส่วนที่ได้ประโยชน์จะกลายเป็น : $\frac{p}{s}T.$

เวลาดำเนินการตามทฤษฎี **T(s)** ของงานทั้งหมดหลังจากการปรับปรุงทรัพยากรแล้ว :

$$T(s) = (1 - p)T + \frac{p}{s}T.$$

กฎของ Amdahl ให้ความเร็วทางทฤษฎีในความหน่วงแฝงของการดำเนินการงานทั้งหมดที่ปริมาณงานคงที่(**W**) ซึ่งส่งผลให้ :

$$S_{\text{latency}}(s) = \frac{TW}{T(s)W} = \frac{T}{T(s)} = \frac{1}{1 - p + \frac{p}{s}}.$$

การประยุกต์ใช้กับ Speedup

Speedup คือการถูกจำกัดโดยเวลาทั้งหมดที่จำเป็นสำหรับส่วนต่อเนื่อง (serial) ของโปรแกรม สำหรับการคำนวณ 10 ชั่วโมง หากเราสามารถทำการการคำนวณ 9 ชั่วโมงแบบขนานและ 1 ชั่วโมงไม่ใช่แบบขนานกันได้ ความเร็วสูงสุดของเราจะถูกจำกัดให้เร็วขึ้น 10 เท่า หากคอมพิวเตอร์เร็วขึ้น ตัวเร่งความเร็วก็จะเท่าเดิม

ถ้าใช้โปรเซสเซอร์ n ตัว Speedup เป็น :

$$Speedup_n = \frac{T_1}{T_n}$$

ที่ T_1 คือเวลาดำเนินการในแกนเดียว, T_n คือเวลาดำเนินการบน n cores, Speedup ควรจะ >1

Speedup Efficiency คือ

$$Efficiency_n = \frac{Speedup_n}{n}$$

ซึ่ง Amdahl's Law บอกว่า

$$Speedup_n = \frac{T_1}{T_n} = \frac{1}{\frac{F_{parallel}}{n} + F_{sequential}} = \frac{1}{\frac{F_{parallel}}{n} + (1 - F_{parallel})}$$

$F_{parallel} / n$: เศษส่วนนี้สามารถลดลงได้โดยการใช้โปรเซสเซอร์หลายตัว, $F_{sequential}$: ไม่สามารถลดลงได้

สามารถหา F parallel ได้โดยใช้ Amdahl's Law เมื่อรู้ speedup และ จำนวนของโปรเซสเซอร์

Amdahl's Law บอกว่า **:**

$$S = \frac{T_1}{T_n} = \frac{1}{\frac{F}{n} + (1-F)} \Rightarrow \frac{1}{S} = \frac{F}{n} + (1-F) = 1 + \frac{F - nF}{n} \Rightarrow \frac{1}{S} - 1 = F\frac{(1-n)}{n}$$

$$F = \frac{\frac{1}{S} - 1}{\frac{1-n}{n}} = \frac{\frac{T_n}{T_1} - 1}{\frac{1-n}{n}} = \frac{\frac{T_n - T_1}{T_1}}{\frac{1-n}{n}} = \frac{\frac{T_1 - T_n}{T_1}}{\frac{n-1}{n}} = \frac{n(T_1 - T_n)}{T_1(n-1)} = \frac{n}{(n-1)}\frac{T_1 - T_n}{T_1} = \frac{n}{(n-1)}\left(1 - \frac{1}{Speedup}\right)$$

หากค่า (n,S) มีหลายค่า สามารถใช้ค่าเฉลี่ย (โดยความจริงแล้วเป็นกำลังสองที่น้อยที่สุด) :

$$F_i = \frac{n_i}{(n_i - 1)}\frac{T_1 - T_{n_i}}{T_1}, i = 2..N$$

$$\bar{F} = \frac{\sum_{i=2}^{N} F_i}{N-1}$$

ถ้า i = 1, $T_{ni} = T_1$

กฎของ Amdahl สามารถให้ความเร็วสูงสุดได้

เศษส่วนดังต่อไปนี้กำหนดขอบเขตบนว่าจะได้รับประโยชน์มากหรือน้อยเท่าใดจากการเพิ่มตัวประมวลผลมากขึ้น :

$$\max Speedup = \lim_{n\to\infty} Speedup = \frac{1}{F_{sequential}} = \frac{1}{1 - F_{parallel}}$$

$F_{parallel}$	maxSpeedup
0.00	1.00
0.10	1.11
0.20	1.25
0.30	1.43
0.40	1.67
0.50	2.00
0.60	2.50
0.70	3.33
0.80	5.00
0.90	10.00
0.95	20.00
0.99	100.00

ตัวอย่างเช่น หากโปรแกรมต้องใช้เวลา 20 ชั่วโมงจึงจะเสร็จสมบูรณ์โดยใช้เธรดเดียว แต่ส่วนหนึ่งชั่วโมงของโปรแกรมไม่สามารถขนานกันได้ ดังนั้นเวลาดำเนินการที่เหลือเพียง 19 ชั่วโมง (p = 0.95) เท่านั้นที่จะสามารถขนานกันได้ จำนวนเธรดที่อุทิศให้กับการดำเนินการแบบขนานของโปรแกรมนี้ เวลาดำเนินการขั้นต่ำต้องไม่น้อยกว่า 1 ชั่วโมง ดังนั้นการเร่งความเร็วตามทฤษฎีจึงจำกัดไว้ที่ 20 เท่าของประสิทธิภาพของเธรดเดียว (1 / (1 – p) = 20)

การเพิ่มประสิทธิภาพส่วนต่อเนื่องของโปรแกรมคู่ขนาน

ถ้าส่วนที่ไม่ขนานกันถูกปรับให้เหมาะสมโดยปัจจัยของ O

$$T(O,s) = (1-p)\frac{T}{O} + \frac{p}{s}T.$$

เป็นไปตามกฎของ Amdahl ที่ว่าการเร่งความเร็วเนื่องจากการขนานกันนั้นถูกกำหนดโดย

$$S_{\text{latency}}(O,s) = \frac{T(O)}{T(O,s)} = \frac{(1-p)\frac{1}{O} + p}{\frac{1-p}{O} + \frac{p}{s}}.$$

เมื่อ s = 1,จะมี $S_{\text{latency}}(O,s) = 1$ ซึ่งหมายความว่าการเร่งความเร็วจะถูกวัดตามเวลาดำเนินการหลังจากปรับชิ้นส่วนที่ไม่เป็นแบบขนาน

เมื่อ s = ∞

$$S_{\text{latency}}(O,\infty) = \frac{T(O)}{T(O,s)} = \frac{(1-p)\frac{1}{O} + p}{\frac{1-p}{O} + \frac{p}{s}} = 1 + \frac{p}{1-p}O.$$

ถ้า 1 – p = 0.4 , O = 2 และ s = 5 แล้ว

$$S_{\text{latency}}(O,s) = \frac{T(O)}{T(O,s)} = \frac{0.4\frac{1}{2} + 0.6}{\frac{0.4}{2} + \frac{0.6}{5}} = 2.5.$$

การแปลงชิ้นส่วนที่ต่อเนื่องกันของโปรแกรมคู่ขนานให้เป็นแบบขนาน

พิจารณากรณีที่ส่วนที่ไม่ขนานกันลดลงด้วยค่า O′ และส่วนที่ขนานกันได้จะเพิ่มขึ้นตามลำดับ แล้ว

$$T'(O',s) = \frac{1-p}{O'}T + \left(1 - \frac{1-p}{O'}\right)\frac{T}{s}.$$

มันเป็นไปตามกฎของ Amdahl ที่ว่าการเร่งความเร็วเนื่องจากการขนานกันนั้นถูกกำหนดโดย

$$S'_{\text{latency}}(O',s) = \frac{T'(O')}{T'(O',s)} = \frac{1}{\frac{1-p}{O'} + \left(1 - \frac{1-p}{O'}\right)\frac{1}{s}}.$$

ที่มาข้างต้นสอดคล้องกับการวิเคราะห์ของ Jakob Jenkov เกี่ยวกับเวลาดำเนินการกับการแลกเปลี่ยนเพื่อเร่งความเร็ว

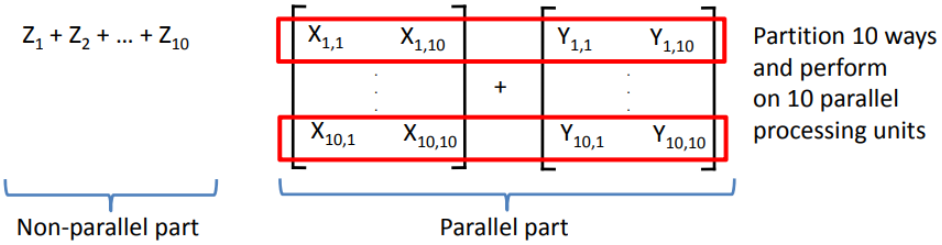
ตัวอย่าง(Examples)

สมมุติหาก 30% ของเวลาดำเนินการอาจเป็นเรื่องของการเพิ่มความเร็ว p จะเป็น 0.3; ถ้าต้องการปรับปรุงทำให้ส่วนที่ได้รับผลกระทบเร็วขึ้นสองเท่า s จะเป็น 2 กฎของ Amdahl ระบุว่าการเร่งความเร็วโดยรวมของการใช้การปรับปรุงจะเป็นดังนี้:

$\mathbf{S}_{\text{latency}} = 1 / 1 - p + (p / s) = 1 / 1 – 0.3 + (0.3/2) = 1.18$

สมมติว่าเราได้รับงานต่อเนื่องซึ่งแบ่งออกเป็นสี่ส่วนต่อเนื่องกัน ซึ่งมีเปอร์เซ็นต์ของเวลาในการดำเนินการคือ p 1 = 0.11 , p 2 = 0.18 , p 3 = 0.23 และ p 4 = 0.48 ตามลำดับ จากนั้นเราจะบอกว่าส่วนที่ 1 ไม่เร่งความเร็ว ดังนั้น s1 = 1ในขณะที่ส่วนที่ 2 เร่งขึ้น 5 เท่า ดังนั้น s2 = 5 ส่วนที่ 3 เร่งขึ้น 20 ครั้ง ดังนั้น s3 = 20 , และส่วนที่ 4 คือการเร่งความเร็วขึ้น 1.6 เท่า ดังนั้น s4 = 1.6 โดยใช้กฎของ Amdahl การเร่งความเร็วโดยรวมคือ

$\mathbf{S}_{\text{latency}} = 1 / (p1/s1) + (p2/s2) + (p3/s3) + (p4/s4) = 1 / (0.11/1) + (0.18/5) + (0.23/20) + (0.48/1.6) = 2.19$



ที่มา : Microsoft PowerPoint - 16LecSu12TLP.pptx (berkeley.edu)

พิจารณา การรวมตัวแปร scalar 10 ตัวและ 10 x 10 เมทริกซ์ (ผลรวมเมทริกซ์) บนโปรเซสเซอร์ 10 ตัว

$\mathbf{S}_{\text{latency}} = 1 / 1 - p + (p / s) = 1 / 1 - 0.909 + (0.909 / 10) = 5.5$

ถ้าเป็นโปรเซสเซอร์ 100 ตัว

$\mathbf{S}_{\text{latency}} = 1 / 1 - p + (p / s) = 1 / 1 - 0.909 + (0.909 / 100) = 10.0$

ถ้าเป็น 100 x 100 เมทริกซ์ บนโปรเซสเซอร์ 10 ตัว

$\mathbf{S}_{\text{latency}} = 1 / 1 - p + (p / s) = 1 / 1 - 0.999 + (0.999 / 10) = 9.9$

ถ้าเป็น 100 x 100 เมทริกซ์ บนโปรเซสเซอร์ 100 ตัว

$\mathbf{S}_{\text{latency}} = 1 / 1 - p + (p / s) = 1 / 1 - 0.999 + (0.999 / 100) = 91.0$