

การใช้คอมพิวเตอร์เพื่อช่วยในการวิเคราะห์ความเกี่ยวพันข้อมูล

วิเชียร เปรมชัยสวัสดิ์
ผู้ช่วยศาสตราจารย์
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น

บทคัดย่อ

การทำงานของเครื่องคอมพิวเตอร์ที่ใช้หลายโปรเซสเซอร์ทำงานร่วมกัน เพื่อเพิ่มความเร็วในการประมวลผลนั้น จะสามารถกระทำได้อย่างถูกต้องต่อเมื่อทราบความเกี่ยวพันของข้อมูลเสียก่อน โปรแกรมเพื่อแสดงความเกี่ยวพันของข้อมูลจึงได้ถูกพัฒนาขึ้นบนเครื่องไมโครคอมพิวเตอร์ เพื่อใช้เป็นพื้นฐานในการศึกษาวิธีการประมวลผลแบบขนาน และการสร้างตัวแปลภาษาแบบอัตโนมัติ โดยมีโปรแกรมที่เขียนขึ้นแบบอนุกรมเป็นข้อมูลเข้า

Computer - Aid Data Dependency Analysis

W. Premchalswadi
Department of Computer Engineering
Faculty of Engineering
Khon Kaen University

Abstract

The processing of computers using multiprocessor to increase the speed of processing can be performed only when data dependence in that program is known. Program for analyzing data dependence is developed on a microcomputer. This will be used to study parallel processing and construction of an automatic parallel compiler that uses a sequential program as input.

บทนำ

ปัจจุบันได้เริ่มมีการนำเครื่องคอมพิวเตอร์ที่มีการประมวลผลโดยใช้หลายโปรเซสเซอร์เข้ามาใช้ เพื่อเพิ่มความเร็วในการประมวลผลกันมากขึ้นและมีแนวโน้มว่าจะเป็นที่ใช้กันแพร่หลายในอนาคต ดังนั้น การศึกษาถึงการทำงานของเครื่องคอมพิวเตอร์เหล่านี้จึงเป็นสิ่งจำเป็น เพื่อเป็นการเพิ่มประสิทธิภาพในการใช้งาน ตลอดจนทราบถึงข้อจำกัดต่าง ๆ

การทำงานของเครื่องคอมพิวเตอร์ที่มีหลายโปรเซสเซอร์นั้นเปรียบได้กับการทำงานร่วมกันของคนหลายคนเป็นทีม ซึ่งจะต้องมีการวางแผนและการประสานงานเป็นอย่างดีเพื่อให้การทำงานนั้นลุล่วงไปอย่างมีประสิทธิภาพ การทำงานเป็นทีมนี้มิได้เป็นหลักประกันว่างานนั้นจะเสร็จสิ้นได้อย่างรวดเร็ว ในทางตรงกันข้ามอาจกลับทำให้ล่าช้ากว่าเดิมได้เช่นเดียวกัน ถ้าขาดการวางแผนประสานงานที่มีประสิทธิภาพ การทำงานของเครื่องคอมพิวเตอร์แบบหลายโปรเซสเซอร์ก็เช่นเดียวกัน จะต้องมีการสั่งการให้แต่ละโปรเซสเซอร์ทำงานอย่างมีจังหวะและสอดคล้องกัน จึงจะสามารถเพิ่มความเร็วในการประมวลผลได้

การสั่งให้คอมพิวเตอร์แบบหลายโปรเซสเซอร์ทำงานนี้ ก็ทำโดยใช้โปรแกรมเช่นเดียวกับเครื่องคอมพิวเตอร์แบบโปรเซสเซอร์เดียวซึ่งมีด้วยกันสองวิธีคือ วิธีแรกผู้เขียนโปรแกรมจะต้องเป็นผู้กำหนดว่าเมื่อใดจะให้ใช้หลายโปรเซสเซอร์ทำงานร่วมกัน และเมื่อใดจะให้โปรเซสเซอร์ตัวเดียวทำงาน วิธีที่สอง การสั่งการเป็นหน้าที่ของตัวแปลภาษา (compiler) ที่จะทำหน้าที่ในการแปลงโปรแกรมที่เขียนเพื่อใช้กับเครื่องคอมพิวเตอร์แบบโปรเซสเซอร์เดียว ให้ใช้กับเครื่องคอมพิวเตอร์หลายโปรเซสเซอร์โดยอัตโนมัติ

ในแบบแรกนั้นผู้เขียนโปรแกรมจะต้องมีความรู้ความสามารถเป็นอย่างดี ทั้งในเนื้อหาของงานที่ทำและเทคนิคการเขียนโปรแกรมแบบขนาน ซึ่งเป็นการยากทีเดียวที่จะหาผู้ที่มีความเชี่ยวชาญทั้งสองด้านในบุคคลเดียวกัน ส่วนในวิธีที่สองนั้นผู้ใช้สามารถเขียนโปรแกรมด้วยวิธีการธรรมดา ไม่มีความจำเป็นต้องเรียนรู้หลักการใหม่ๆ ที่ใช้กับการประมวลผลแบบขนาน หน้าที่เหล่านี้ได้ปล่อยให้เป็นการของตัวแปลภาษา การที่เครื่องคอมพิวเตอร์ จะสามารถทำงานอย่างขนานได้นั้นสิ่งจำเป็นเบื้องต้นที่จะต้องทราบก็คือ ความเกี่ยวพันข้อมูลในตัวโปรแกรมนั้นกล่าวคือ ถ้าชุดของคำสั่งใดที่ไม่มีความเกี่ยวพันข้อมูลซึ่งกันและกันในขณะใดขณะหนึ่ง ก็สามารถนำมาประมวลผลแบบขนาน โดยใช้หลายโปรเซสเซอร์ประมวลผลคำสั่งเหล่านั้นในเวลาเดียวกันได้

การศึกษาเกี่ยวกับการเกี่ยวพันข้อมูลจึงกล่าวได้ว่า เป็นเงื่อนไขพื้นฐานที่จะทำให้การประมวลผลอย่างขนานกระทำได้หรือไม่อย่างไร และยังเป็นพื้นฐานในการสร้างตัวแปลภาษาแบบอัตโนมัติที่ใช้กับเครื่องคอมพิวเตอร์ชนิดหลายโปรเซสเซอร์ได้อีกด้วย โปรแกรมที่ทำการวิเคราะห์ความเกี่ยวพันข้อมูลนี้จึงได้ถูกพัฒนาขึ้น โดยในเบื้องต้นนี้ได้กำหนดขอบเขตของการพัฒนาอยู่ที่ตัวแปรที่เป็นแบบสเกลาร์ในเบสิกบล็อกเท่านั้น ซึ่งมุ่งหวังว่าจะเป็นประโยชน์ต่อการเรียนการสอนและการวิจัยทางด้านการประมวลผลแบบขนานต่อไป

การวิเคราะห์ความเกี่ยวพันข้อมูล

คำสั่ง (Statement) T จะมีความเกี่ยวพันกับคำสั่ง S เมื่อปรากฏว่า ในขณะที่ขณะหนึ่งค่าของ S' ของ S และ T' ของ T และตำแหน่งในหน่วยความจำ M ทำให้เกิด

1. ทั้ง S' และ T' อ้างอิงถึงตำแหน่งที่ M และการอ้างอิงอย่างน้อยหนึ่งครั้งที่เป็นการเขียนข้อมูล (write)
2. ในโปรแกรมที่เขียนแบบอนุกรมนั้น S' ต้องถูกทำงานก่อน T'
3. ในการทำงาน ณ เวลาเดียวกัน ตำแหน่งที่หน่วยความจำ M จะต้องไม่ถูกเขียนลงไประหว่างเวลาที่ S' ทำงานเสร็จสิ้นและเวลาที่ T' เริ่มทำงาน

ความเกี่ยวพันข้อมูลที่กล่าวมาข้างต้นนี้สามารถแบ่งออกได้เป็น 3 ประเภทด้วยกันดังนี้

- 2.1 โฟลว์ ดีเพนเดนซ์ (Flow dependence) เป็นการที่มีการกำหนดค่าของตัวแปรในคำสั่งใดคำสั่งหนึ่ง แล้วมีการใช้ตัวแปรดังกล่าวในคำสั่งอื่นๆ กล่าวคือ S' เขียนข้อมูลที่ M แล้ว T' อ่าน
- 2.2 เอาท์พุท ดีเพนเดนซ์ (Output dependence) เป็นการที่มีการกำหนดค่าของตัวแปรในคำสั่งหนึ่ง แล้วมีการกำหนดค่าของตัวแปรนั้นในคำสั่งอื่นซ้ำอีกที กล่าวคือ S' เขียนข้อมูลที่ M และหลังจากนั้น T' เขียนข้อมูลอีกครั้งหนึ่ง
- 2.3 แอนตี้ ดีเพนเดนซ์ (Anti dependence) เป็นการที่มีการใช้ค่าตัวแปร ในคำสั่งหนึ่งแล้วภายหลังมีการกำหนดค่าใหม่ให้กับตัวแปรตัวนั้นอีก กล่าวคือ S' อ่านข้อมูลที่ M แล้ว T' เขียน จากความสัมพันธ์ทั้งสามชนิดนี้เราสามารถนำมาเขียนในรูปของตัวแปรเข้าและตัวแปรออก (Input and Output variables) เพื่อแสดงความสัมพันธ์ที่จะเกิดจากคู่ของตัวแปร (X, Y) ได้ดังนี้

โฟลว์ดีเพนเดนซ์ ถ้า $x \in \text{OUT}(S)$ และ $y \in \text{IN}(T)$;

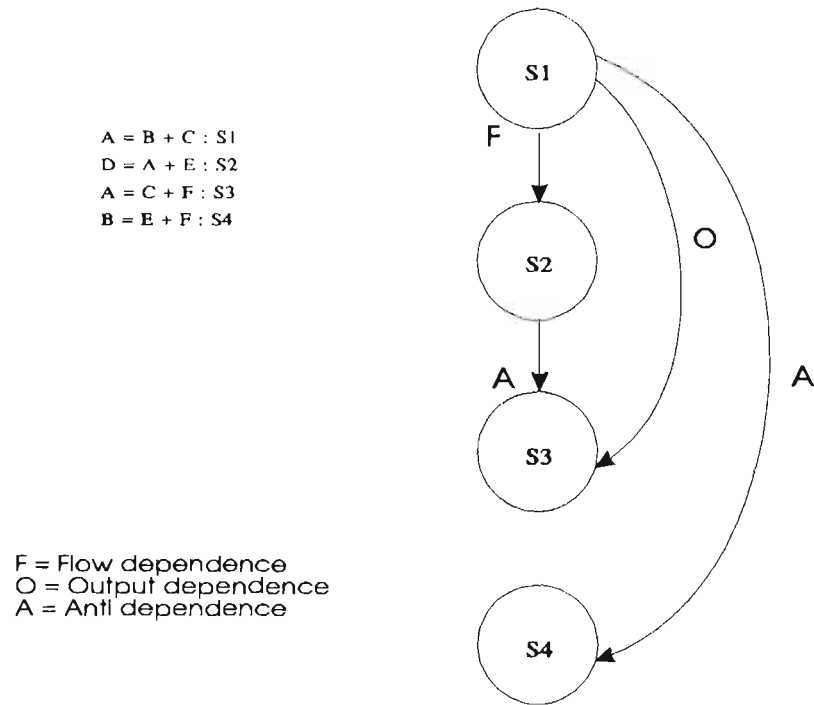
เอาท์พุทดีเพนเดนซ์ ถ้า $x \in \text{OUT}(S)$ และ $y \in \text{OUT}(T)$;

แอนตี้ดีเพนเดนซ์ ถ้า $x \in \text{IN}(S)$ และ $y \in \text{OUT}(T)$.

โดยที่ IN และ OUT คือชุดของตัวแปรเข้าและออกตามลำดับ

ดังนั้น T จะขึ้นต่อ S ถ้าอย่างน้อยมีคู่ของตัวแปร (X, Y) ที่ทำให้เกิดความเกี่ยวพันไม่ว่าชนิดใดก็ตามของ T กับ S เช่น T เป็นโฟลว์ดีเพนเดนซ์กับ S ถ้ามีตัวแปรอย่างน้อยหนึ่งตัวในตัวแปรเข้า y ของ T ที่ทำให้ (X, Y) เกิดความเกี่ยวพันของ T กับ S โดยที่ $x \in \text{OUT}(S)$

ความเกี่ยวพันของข้อมูลทั้ง 3 แบบที่กล่าวมาแล้วสามารถแสดงได้ดังรูปที่ 1



รูปที่ 1 ตัวอย่างแสดงความสัมพันธ์ข้อมูลแบบต่าง ๆ

จากตัวอย่างในรูปที่ 1 คำสั่งที่ 2 (S2) มีการใช้ค่าของ A ที่ได้มีการกำหนดไว้ใน คำสั่งที่ 1 (S1) ดังนั้นจึงกล่าวได้ว่า S1 มีโฟลว์ ดีเพนเดนซ์กับ S2 ซึ่งสามารถเขียนสั้นๆ ได้ดังนี้ $S1 \delta S2$ ส่วน S3 ได้มีการกำหนดค่าของ ตัวแปร A อีกครั้ง ซึ่งก่อนหน้านี้ได้มีการกำหนดค่าของตัวแปร A ไปใน S1 จึงกล่าวได้ว่า S1 มีเอ๊าท์พุท ดีเพนเดนซ์กับ S3 และเขียนสั้นๆ ได้ดังนี้ $S1 \delta_o S3$ และในส่วนของ S1 และ S4 นั้น จะเห็นได้ว่ามีการใช้ค่าของตัวแปร B ใน S1 แล้วมีการกำหนดค่าให้กับตัวแปร B ใน S4 จึงกล่าวได้ว่า S1 มี แอนตี้ ดีเพนเดนซ์กับ S4 ซึ่งเขียนได้ดังนี้ $S1 \delta_A S4$.

นอกจากความสัมพันธ์ของข้อมูลที่กล่าวมาแล้ว โฟลว์คอนโทรลก็เป็นสิ่งสำคัญ อีกอย่างหนึ่งที่ต้องนำมาพิจารณาในการหาความสัมพันธ์ของข้อมูล เช่น

```

T1 : A = B+C
      if (X >= 0) then
T2 : A = 0
      else
T3 : D = A
      endif

```

จะพบว่ามีความสัมพันธ์ดังนี้

T1 δ_o T2

และ **T2 δ T3**

แต่จะเห็นว่า T2 δ T3 จะไม่เป็นจริง เนื่องจากถึงแม้ว่าจะมีการกำหนดค่าให้ A ใน T2 และมีการใช้ค่า A ใน T3 และ T3 ปรากฏอยู่หลัง T2 ในโปรแกรมก็ตาม ทั้งนี้เนื่องจาก T2 และ T3 อยู่กันคนละกิ่งของคำสั่ง IF ซึ่งทำให้ค่าของ A ที่เกิดขึ้นใน T3 จะไม่มีทอมมาจาก T2 ได้เลย

เนื่องจากทิศทางการทำงานในโปรแกรมนั้น จะไม่สามารถทราบล่วงหน้าได้จนกว่าจะมีการประมวลผลจริงเท่านั้น ซึ่งจะขึ้นอยู่กับเงื่อนไขของ IF ในขณะนั้น ดังนั้นความสัมพันธ์ของข้อมูลจึงมีใช้เป็นเพียงสิ่งเดียวที่จะบ่งชี้ถึงการสื่อสารข้อมูลหรือ ความขัดแย้งของหน่วยความจำ เช่น

```

T1 : A = B+C
C1 : if (X >= 0) then
T2 : A = A+2
      endif
T3 : D = A*2.1

```

จากการวิเคราะห์ความสัมพันธ์ของข้อมูลพบว่า

T1 δ T3

และ **T2 δ T3**

ถึงแม้ว่า T3 ที่จริงแล้วจะรับค่าของ A จาก T1 หรือ T2 อันใดอันหนึ่งเท่านั้นขึ้นอยู่กับค่าของ X ความสัมพันธ์เช่นนี้เกิดขึ้นเนื่องจากการใช้คำสั่งเงื่อนไข IF ทำให้ความสัมพันธ์มีขึ้นอย่างมีเงื่อนไขเช่นเดียวกัน เราเรียกความสัมพันธ์เช่นนี้ว่าเป็น คอนโทรลดีเพนเดนซ์ (control dependence) ซึ่งเขียนแทนได้ด้วย

T1 δ^c T2

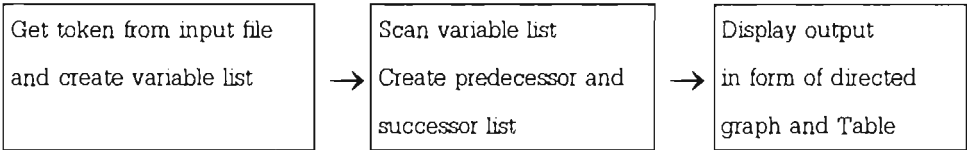
ความสัมพันธ์ของ คอนโทรล ดีเพนเดนซ์ เป็นส่วนที่ต้องนำมาพิจารณาควบคู่กับการหาความ เกี่ยวพันธ์ข้อมูลอย่างอื่นๆ ที่ได้กล่าวมาแล้ว เพื่อที่จะทำให้สามารถจัดเรียงลำดับคำสั่งในโปรแกรม ใหม่ได้

ความเกี่ยวพันธ์ข้อมูลเหล่านี้จะต้องถูกวิเคราะห์ด้วยตัวแปลภาษาโดยอัตโนมัติ เพื่อใช้เป็นข้อมูล ในการสร้างโค้ดสั่งให้เครื่องคอมพิวเตอร์ทำงานอย่างขนานต่อไป

โปรแกรมและการทดสอบ

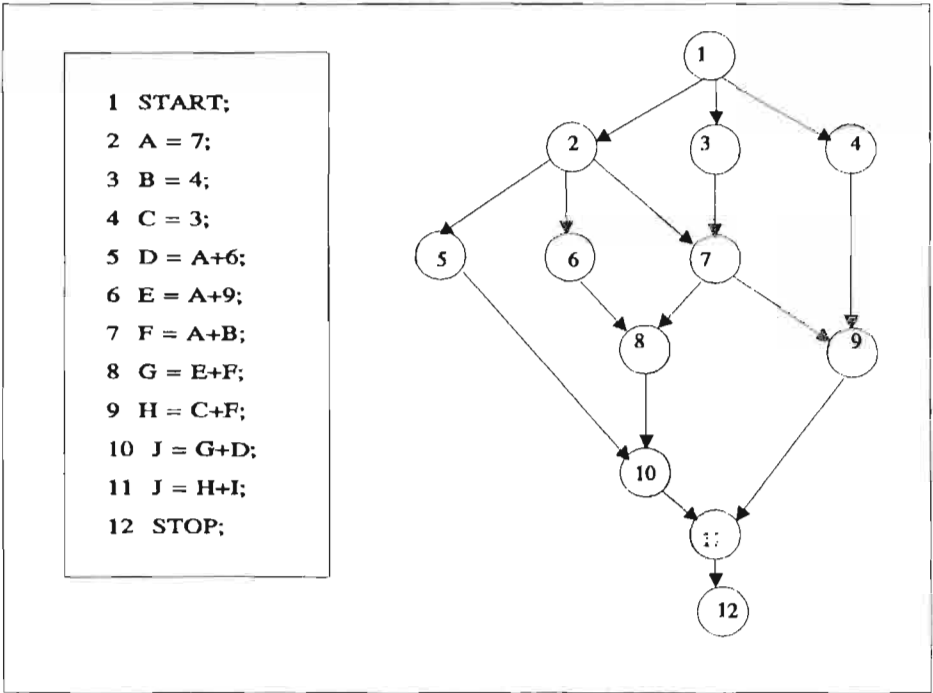
โปรแกรมวิเคราะห์ความเกี่ยวพันธ์ข้อมูลนี้ได้ถูกพัฒนาขึ้น โดยอาศัยหลักการในข้อ 2 โดยมี โปรแกรมที่เขียนทั่วไป ในลักษณะที่คล้ายกับภาษาฟอร์แทรน แต่ได้มีการเพิ่มบางส่วนเข้าไปเพื่อให้ช่วยต่อ การศึกษาในขั้นต้นนี้ โดยให้มีคำสั่ง START และ STOP เป็นคำสั่งเริ่มแรกและสุดท้ายของโปรแกรมตาม ลำดับ และแต่ละคำสั่งจะแยกกันด้วยเครื่องหมาย semicolon (;) เป็นข้อมูลเข้า(อินพุท) ส่วนข้อมูลออก (เอาท์พุท) สามารถแสดงได้เป็นสองลักษณะคือ ตารางแสดงถึงคำสั่งที่อยู่ก่อนและหลังคำสั่งนั้นๆ ที่มีความ เกี่ยวพันธ์ข้อมูลกันและเพื่อให้ช่วยต่อความเข้าใจของผู้ใช้ ความเกี่ยวพันธ์ของแต่ละคำสั่งก็สามารถแสดง ออกมาในรูปของกราฟได้อีกด้วย

ขั้นตอนการทำงานของโปรแกรมวิเคราะห์ความเกี่ยวพันธ์ข้อมูลนี้ได้แสดงไว้ในรูปที่ 3



รูปที่ 3 ขั้นตอนของโปรแกรมการวิเคราะห์ความเกี่ยวพันธ์ข้อมูล

รูปที่ 4 แสดงผลลัพธ์ของการวิเคราะห์ความเกี่ยวพันธ์ข้อมูล โดยที่ทางด้านซ้ายมือ คือ อิน พุทโปรแกรม ส่วนทางด้านขวามือคือผลที่เกิดจากการวิเคราะห์ข้อมูลของโปรแกรมนี้อยู่ที่วงกลมแต่ละวงจะ แทนคำสั่งแต่ละคำสั่งและตัวเลขที่อยู่ภายในวงกลมจะแสดงเลขที่บรรทัดที่อยู่ของคำสั่งนั้น เพื่อใช้ในการอ้างอิง และเส้นตรงที่เชื่อมต่อระหว่างวงกลมแต่ละวงจะบอกให้ทราบว่า คำสั่งมีความเกี่ยวพันธ์ข้อมูลกัน ตาราง ที่ 1 และ 2 แสดงคำสั่งที่อยู่ก่อนหน้าและหลังของแต่ละคำสั่งตามลำดับ



รูปที่ 4 ผลลัพธ์จากการวิเคราะห์ความสัมพันธ์ข้อมูล

ตารางที่ 1
แสดงคำสั่งที่อยู่ก่อนหน้า

task no.	Predecessor task
2	1
3	1
4	1
5	2
6	2
7	2,3
8	6,7
9	4,7
10	5,8
11	9,10

ตารางที่ 2
แสดงคำสั่งที่อยู่หลัง

task no.	Successor task
2	5,6,7
3	7
4	9
5	10
6	8
7	8,9
8	10
9	11
10	11
11	12

สรุปและข้อเสนอแนะ

โปรแกรมที่ถูกพัฒนาขึ้นนี้สามารถทำการวิเคราะห์ความเกี่ยวพันข้อมูลและแสดงผลลัพธ์ออกมาในรูปของกราฟ และตารางแสดงคำสั่งที่อยู่ก่อนและหลังคำสั่งแต่ละคำสั่งได้อย่างถูกต้อง ซึ่งจะเป็นประโยชน์ในการเรียนการสอนเกี่ยวกับการประมวลผลอย่างขนาน แต่ก็มีข้อจำกัดอยู่ที่เป็นการวิเคราะห์ความเกี่ยวพันข้อมูลแบบ สเกลล่า ในแบบลิสบล็อกใดบล็อกหนึ่งเท่านั้น ซึ่งต่อไปจะได้ขยายความสามารถให้รวมถึงข้อมูลแบบอาร์เรย์และการวิเคราะห์โปรแกรมที่มีลูปรวมอยู่ด้วยต่อไป ขาวสารที่ได้จากการวิเคราะห์ความเกี่ยวพันข้อมูลนี้ สามารถที่จะนำไปใช้ในการกำหนดการแบ่งงาน (Job scheduling) ให้แต่ละโปรเซสเซอร์ ให้มีประสิทธิภาพสูงสุดต่อไปอีกด้วย

บรรณานุกรม

- 1) D.A. Padua and M.J. Wolfe; **"Advanced Compiler optimizations for Supercomputers"** Communication of the ACM, Volume 29, No.12, Dec. 1986, pp 1184-1201
- 2) H. Kasahara.; **"Parallel Processing Technology"** Corona Publishing, 1991
- 3) U. Banerjee, **"Dependence Analysis for Supercomputing"**, Kluwer Academic, 1988