

ความเกี่ยวพันของข้อมูล

นุชรี เปรมชัยสวัสดิ์

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น

บทคัดย่อ

วิธีการประมวลผลอย่างขนานเป็นหนทางหนึ่งที่จะได้มาซึ่งความเร็วของการคำนวณอย่างสูง แต่อย่างไรก็ตามประสิทธิภาพของการทำงานอย่างขนานนี้ขึ้นอยู่กับความขนานที่มีอยู่ในแต่ละโปรแกรมซึ่งสามารถค้นหาได้ โดยการหาความสัมพันธ์ของข้อมูล ดังนั้นความรู้เกี่ยวกับความสัมพันธ์ของข้อมูลจึงเป็นสิ่งจำเป็นเพื่อนำมาเพิ่มประสิทธิภาพหรือสร้างตัวแปรภาษาอย่างอัตโนมัติเพื่อใช้ในการประมวลผลอย่างขนาน บทความนี้ได้กล่าวถึงรูปแบบต่าง ๆ ของความสัมพันธ์ของข้อมูลที่เกิดขึ้นในแต่ละโปรแกรม

ABSTRACT

Parallel processing is an approach to meet high speed computing power. However, it's performance is depended on parallelism having in each program which can be detected by finding data-dependence. To improve performance or building an automatic parallel compiler, knowledge of data dependence is required. This paper describes how data dependence occurred in a program.

1. บทนำ

คอมพิวเตอร์ซึ่งมีขีดความสามารถสูง ได้เข้ามามีส่วนสำคัญและจำเป็นอย่างมากในงานที่ต้องคำนวณ โดยเฉพาะอย่างยิ่งในงานทางด้านวิทยาศาสตร์และวิศวกรรมศาสตร์ คอมพิวเตอร์ขนาดใหญ่ได้ถูกนำมาใช้ในงานเหล่านี้ แต่อย่างไรก็ตาม ก็ยังมีความต้องการคอมพิวเตอร์ที่มีความเร็วสูงขึ้นกว่านี้ไปเรื่อย ๆ อย่างไม่มีที่จบสิ้น เพื่อสนองต่อความต้องการดังกล่าว ได้มีแนวความคิดที่จะนำมาใช้ในการเพิ่มความเร็วให้แก่ เครื่องคอมพิวเตอร์ ซึ่งสามารถแบ่งได้เป็นสองแนวความคิดใหญ่ ๆ ด้วยกันคือ การเพิ่มความเร็วของวงจรไฟฟ้า และการเพิ่มจำนวนคำสั่งที่ คอมพิวเตอร์สามารถ กระทำ ได้ โดยเทคนิค ทาง ไพพ์ไลน์ (pipeline) หรือ การทำงาน แบบ อย่าง ขนาน (parallelism) แนวความคิดแรกมีขีดจำกัดของความเร็วสูงสุดที่ความเร็วของแสง ดังนั้นแนวความคิดของการทำ งานอย่างขนาน จึงเป็นเพียงหนทางเดียวที่จะเพิ่มความเร็วของการคำนวณได้ ตามต้องการเสมือนว่าไม่มีขีด จำกัด ถ้าการแลกเปลี่ยนข้อมูลระหว่างโปรเซสเซอร์และหน่วยความจำไม่ทำให้เกิดปัญหาคอขวดขึ้น แนวความคิดนี้ได้เป็นที่ยอมรับของนักออกแบบคอมพิวเตอร์ในปัจจุบันเป็นอย่างมาก ซึ่งจะเห็นได้จากการเสนอคอมพิวเตอร์ รุ่นใหม่ ๆ ที่ใช้แนวความคิดนี้ออกสู่ท้องตลาด กล่าวคือ การใช้โปรเซสเซอร์ หลายตัวให้ทำงานพร้อมกันอย่าง ขนาน

การที่จะสามารถทำให้คอมพิวเตอร์สามารถทำงานอย่างขนานได้ สิ่งจำเป็นเบื้องต้นก็คือ จะต้องทราบถึงความเกี่ยวพันของข้อมูลในโปรแกรมที่จะถูกทำการประมวลผล อย่างขนานนี้เสียก่อน ความเกี่ยวพันของข้อมูลจะเป็นตัวบ่งชี้ว่า ข้อมูลเหล่านี้จะสามารถนำมาทำการประมวลผลอย่างขนานได้หรือไม่ มากน้อยเพียงใด ความเกี่ยวพันของข้อมูลนี้สามารถแบ่งได้เป็น 3 ประเภทด้วยกันคือ

1. โฟลว์ดีเพนเดนซ์ (Flow dependence)
2. เอาท์พุทดีเพนเดนซ์ (Output dependence)
3. แอนตี้ดีเพนเดนซ์ (Anti dependence)

ความเกี่ยวพันทั้ง 3 ประเภทนี้ ได้อธิบายโดยการยกตัวอย่างให้เห็นต่อไปนี้

2. โฟลว์ดีเพนเดนซ์

$$T1 : A = B + C$$

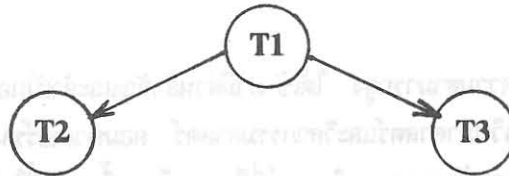
$$T2 : D = A + 2$$

$$T3 : E = A * 3$$

กระทงความ T1 และ T2 ไม่สามารถทำงานในเวลาเดียวกันหรือทำงานอย่างขนานกันได้ เนื่องจาก T2 ต้องใช้ค่า A ซึ่งเกิดจากการคำนวณใน T1 เราเรียกความสัมพันธ์เช่นนี้ว่า โฟลว์ดีเพนเดนซ์ กล่าวคือ T1 มีโฟลว์ดีเพนเดนซ์กับ T2 ซึ่งเขียนแทนได้ด้วย

$$T1 \delta T2$$

และสามารถแสดงความเกี่ยวพันของตัวอย่างข้างบนได้ด้วยภาพดังนี้



จากภาพข้างบนพบว่า T2 และ T3 ไม่มีความเกี่ยวพันกันเลย ดังนั้นเราสามารถที่จะประมวลผลกระทงความ T2 และ T3 ได้ในเวลาเดียวกัน

3. เอาท์พุทดีเพนเดนซ์

$$T1 : A = B + C$$

$$T2 : D = A + 2$$

$$T3 : A = E + F$$

กระทงความ T3 มีการกำหนดค่าใหม่ให้กับ A หลังจาก T1 ได้มีการกำหนดค่าให้กับ A แล้ว ถ้า T1 ถูกประมวลผลหลัง T3 จะทำให้ค่าที่ได้จากการประมวลผลผิดพลาดขึ้นได้ ดังนั้น จึงเป็นเงื่อนไขว่า T1 ต้องทำก่อน T3 เราเรียกความสัมพันธ์ในลักษณะเช่นนี้ว่า เอาท์พุทดีเพนเดนซ์ ซึ่งเขียนแทนได้ด้วย

$$T1 \delta^o T3$$

4. แอนตี้ดีเพนเดนซ์

$$T1 : A = B + C$$

$$T2 : B \leftarrow D/2$$

ในกรณีนี้ กระบวนการ T1 ต้องใช้ค่าของ B ก่อนที่จะมีการกำหนดค่าใหม่ให้แก่ B ใน T2 ลักษณะการเกี่ยวพันของ T1 และ T2 เช่นนี้เรียกว่าความเกี่ยวพันแบบแอนตี้ ดีเพนเดนซ์ ซึ่งเขียนแทนได้ด้วย

$$T1 \quad \bar{\delta} \quad T2$$

เพื่อความเข้าใจของยกตัวอย่างการแสดงความเกี่ยวพันทั้งสามแบบดังนี้

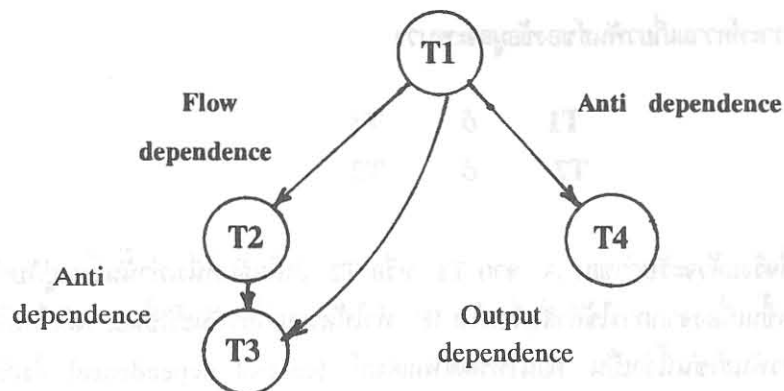
$$T1 : A = B + C$$

$$T2 : D = A + E$$

$$T3 : A = C + F$$

$$T4 : B = E + F$$

จากตัวอย่างนี้เราสามารถแสดงความเกี่ยวพันของข้อมูลทั้งหมดได้ด้วยภาพดังนี้



5. โฟลว์คอนโทรล

นอกจากความเกี่ยวพันของข้อมูลที่กล่าวมาแล้ว โฟลว์คอนโทรลก็เป็นสิ่งสำคัญอีกอย่างหนึ่งที่จะต้องนำมาพิจารณาในการแสดงความเกี่ยวพันของข้อมูล เช่น

$$T1 : A = B + C$$

if (X >= 0) then

$$T2 : A = 0$$

else

$$T3 : D = A$$

endif

จะพบว่ามี ความเกี่ยวพันดังนี้

T1 δ^0 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 จะไม่มีทางมาจาก T2 ได้เลย

เนื่องจากทิศทางการทำงานในโปรแกรมนั้น จะไม่สามารถทราบล่วงหน้าได้จนกว่าจะมีการประมวลจริงเท่านั้น ซึ่งจะขึ้นอยู่กับเงื่อนไขของ IF ในขณะนั้น ดังนั้นความสัมพันธ์ของข้อมูลจึงมิใช่เป็นเพียงสิ่งเดียวที่จะบ่งชี้ถึงการสื่อสารข้อมูลหรือ ความขัดแย้งของหน่วยความจำ เช่น

T1 : A = B + C

C1 : if (X >= 0) then

T2 : A = A + 2

endif

T3 : D = A * 2.1

จากการวิเคราะห์ความเกี่ยวพันของข้อมูลจะพบว่า

T1 δ T3
T2 δ T3

และ

ถึงแม้ว่า T3 ที่จริงแล้วจะรับค่าของ A จาก T1 หรือ T2 อันใดอันหนึ่งเท่านั้นขึ้นอยู่กับค่าของ X ความเกี่ยวพันเช่นนี้เกิดขึ้นเนื่องจากการใช้คำสั่งเงื่อนไข IF ทำให้ความเกี่ยวพันมีขึ้นอย่างมีเงื่อนไขเช่นเดียวกัน เราเรียกความเกี่ยวพันเช่นนี้ว่าเป็น คอนโทรลดีเพนเดนซ์ (control dependence) ซึ่งเขียนแทนได้ด้วย

C1 δ^c T2

ความสัมพันธ์ของ คอนโทรลดีเพนเดนซ์ เป็นส่วนที่ต้องนำมาพิจารณาควบคู่กับการหาความเกี่ยวพันของข้อมูลอย่างอื่นๆ ที่ได้กล่าวมาแล้ว เพื่อที่จะทำให้สามารถจัดเรียงลำดับกระบวนงานในโปรแกรมใหม่ได้

6. สรุป

บทความนี้ได้กล่าวถึงความเกี่ยวพันของข้อมูลในรูปแบบต่าง ๆ เพื่อให้ทราบถึงเงื่อนไขเบื้องต้นในการที่จะนำข้อมูลเหล่านี้ไปประมวลผลอย่างขนานต่อไป ซึ่งจะเห็นได้ว่าความสามารถในการประมวลผลอย่างขนานจะขึ้นอยู่กับชนิดของความเกี่ยวพันที่มีอยู่ในแต่ละโปรแกรม ซึ่งไม่เท่ากัน ประสิทธิภาพของการทำงานอย่างขนานก็จะต่างกันด้วย ถึงแม้ว่าจะใช้เครื่องคอมพิวเตอร์ในการประมวลผลเครื่องเดียวกันก็ตาม

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