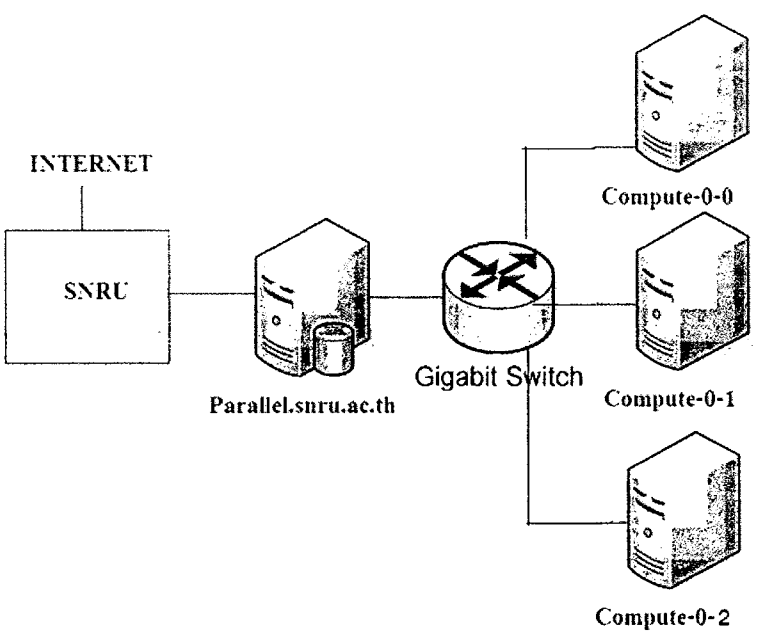
บทที่ 4 ผลการวิจัย

ผลที่ได้จากการดำเนินงานวิจัยนี้ทำให้ได้ระบบการคำนวณแบบขนานของมหาวิทยาลัย ได้งานวิจัยอื่น ๆ ที่ใช้ระบบคำนวณแบบขนานนี้ และนอกจากนี้ยังได้นำไปถ่ายทอดสู่การเรียนการสอนในรายวิชาทางคณิตศาสตร์ประยุกต์แก่นักศึกษาสาขาวิชาคณิตศาสตร์และสาขาวิชาวิทยาการคอมพิวเตอร์ที่เรียนวิชาการวิเคราะห์เชิงตัวเลข

4.1 ระบบการคำนวณแบบขนาน Parallel-SNRU

จากการดำเนินการวิจัยตามแผนการดำเนินงานตลอดโครงการวิจัย ทำให้ได้ระบบสำหรับระบบการคำนวณแบบขนานแบบพีซีคลัสเตอร์ ซึ่งระบบการคำนวณแบบขนานสำหรับงานวิจัยทางการคำนวณทางวิทยาศาสตร์นี้ได้ติดตั้งระบบไว้ที่ห้องคอมพิวเตอร์แม่ข่าย ศูนย์คอมพิวเตอร์ของมหาวิทยาลัยราชภัฏสกลนคร มีโครงสร้างของระบบดังแสดงในภาพที่ 4.1



รูปที่ 4.1 แสดงภาพโครงสร้างของระบบการคำนวณแบบขนาน

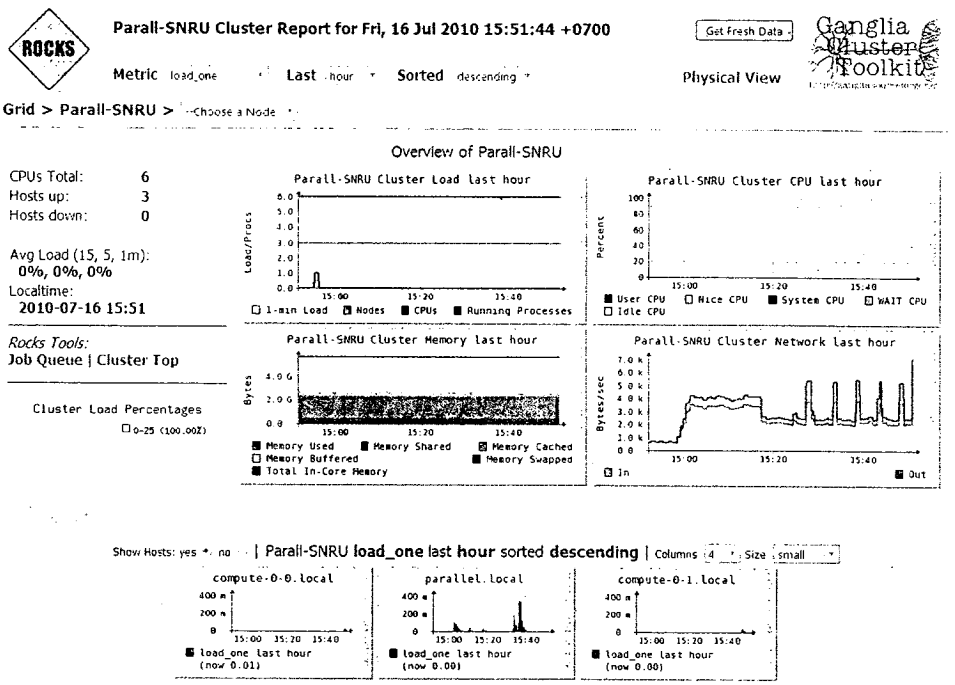
ซึ่งประกอบด้วย

- เครื่องคอมพิวเตอร์ 4 เครื่อง (8 CPUs)
- Gigabit Switch
- สายเชื่อมต่อระบบ
- หมายเลข IP=202.29.24.55 หรือ parallel.snru.ac.th

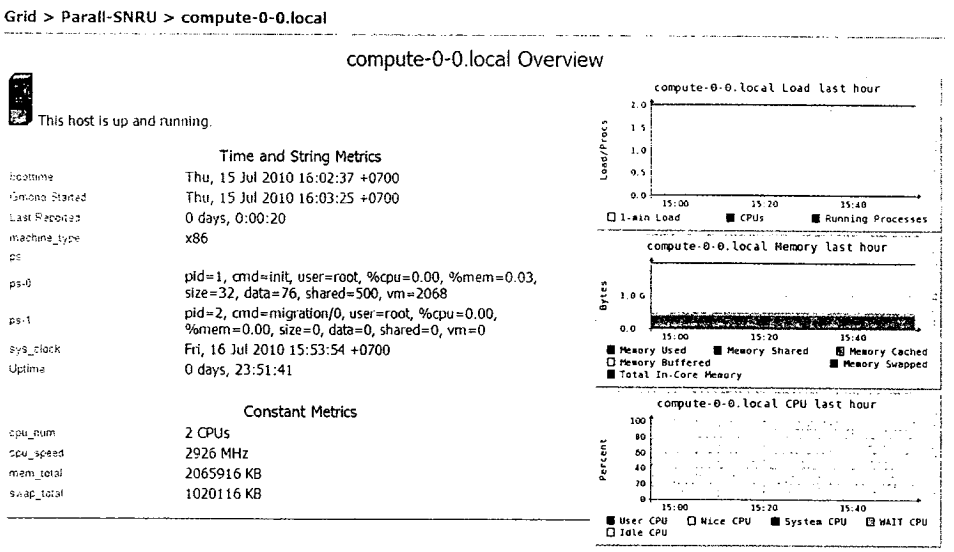
โดยสามารถใช้ได้จริง 6 หน่วยประมวลผล (CPUs) เนื่องจากเครื่อง Front-end ใช้ไป 2 หน่วยประมวลผล ทำให้เหลือเพียง 6 หน่วยประมวลของเครื่องลูก ซึ่งก็ถือมีหน่วยประมวลว่าพอเพียง ไม่มาก ไม่น้อย จนเกินไปสำหรับใช้ประมวลผลงานวิจัยอื่นๆด้านการคำนวณ หรือ ด้านการเรียนการสอนทางด้านคณิตศาสตร์ประยุกต์

ส่วนปัญหาอุปสรรคที่พบในระหว่างการดำเนินการโครงการวิจัย ได้แก่ระบบไฟฟ้าของมหาวิทยาลัยที่ขัดข้องบ่อยครั้งมาก และปัญหาของระบบเน็ตเวิร์กของมหาวิทยาลัยที่เครื่องแม่ข่ายไม่สามารถใช้งานได้หลายครั้งทำให้ผู้วิจัยไม่สามารถดูแลและสั่งการทำงานของระบบการคำนวณแบบขนานจากภายนอกได้ แต่ก็ได้รับช่วยเหลือและความอนุเคราะห์ในการแก้ปัญหาเป็นอย่างดีจากทางเจ้าหน้าที่ที่ศูนย์คอมพิวเตอร์

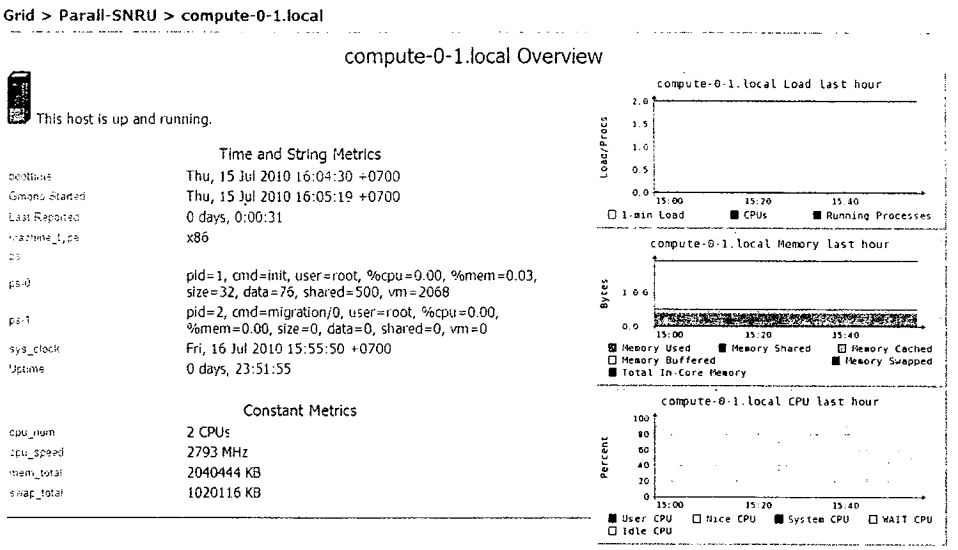
สำหรับการตรวจสอบสถานะการทำงานของระบบ สามารถดูรายละเอียดได้ที่ http://parallel.snru.ac.th ดังแสดงในรูปด้านล่างนี้ โดยจะแสดงสถานะการทำงานของเครื่องอาทิ เช่น การใช้งาน CPU การใช้งานหน่วยความจำ การใช้งาน Disk และ Load Average เป็นต้นดังแสดงในภาพต่อไปนี้



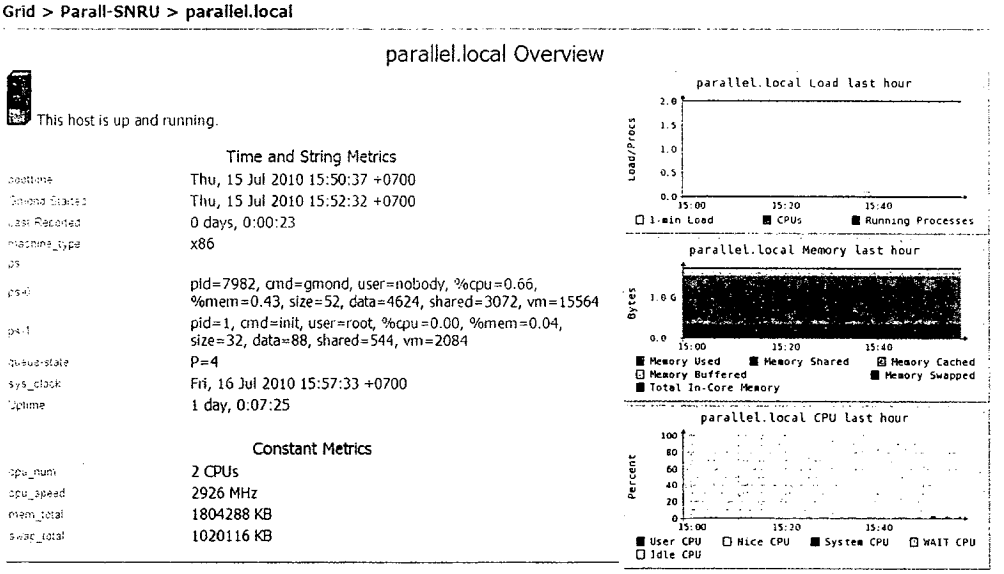
รูปที่ 4.2 ระบบการตรวจสอบสถานะทั้งหมดของระบบ



รูปที่ 4.3 แสดงสถานะบนเครื่อง compute-0-0



รูปที่ 4.4 แสดงสถานะบนเครื่อง compute-0-1



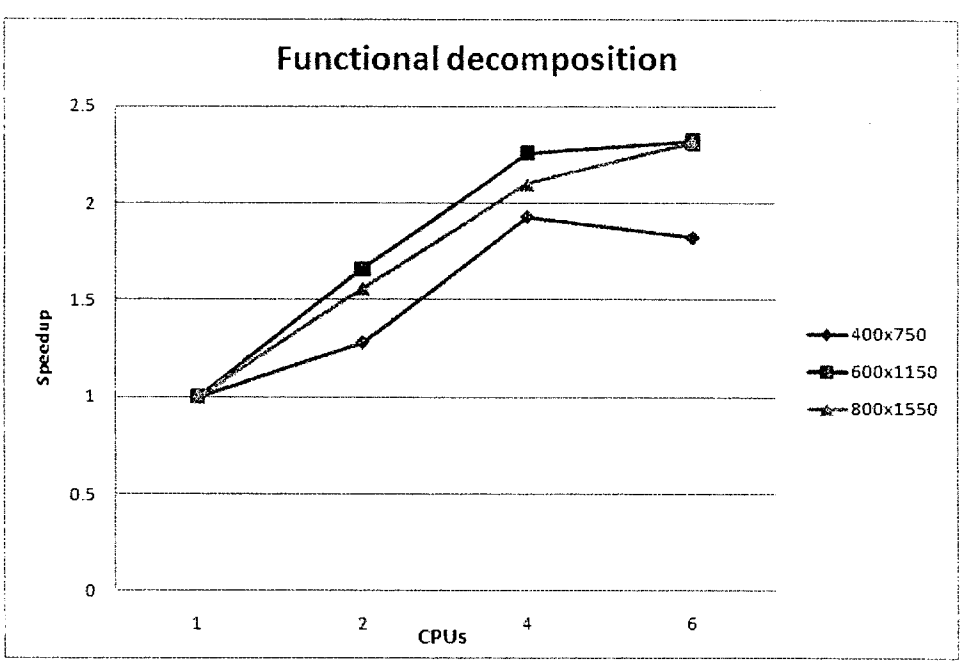
รูปที่ 4.5 แสดงสถานะบนเครื่อง parallel (Front-end)

4.2 ความเสถียรภาพของระบบ

ในการทดสอบความเสถียรภาพของระบบโดยผู้วิจัยได้ทดสอบระบบด้วยโปรแกรมการคำนวณแบบขนานจากงานวิจัยเรื่องพลศาสตร์ของร่องรอยการไหลแบบปั่นป่วนในของไหลที่เป็นชั้นๆ (TURBULENT WAKE DYNAMICS IN LINEAR STRATIFIED FLUID) โดยแบ่งการทดสอบโปรแกรมสองแบบคือแบบ functional decomposition และแบบ domain decomposition และแบ่งขนาดปัญหาทดสอบเป็น 3 ขนาดคือ 400x750 600x1150 และ 800x1550 แล้วเปรียบเทียบเวลาในการประมวลผล เพื่อนำไปหาค่า Speedup ของระบบตามขนาดของปัญหาที่แตกต่างกันดังแสดงในตารางที่ 4.1 และ 4.2

CPUs	size=400x750	speedup	Size=600x1150	speedup	size=800x1550	speedup
1	3,094.24	-	7,863.26	-	13,245.32	-
2	2,417.06	1.28	4,724.35	1.66	8,451.28	1.56
4	1,599.32	1.93	3,472.41	2.26	6,305.41	2.10
6	1,691.23	1.82	3,387.50	2.32	5,713.22	2.31

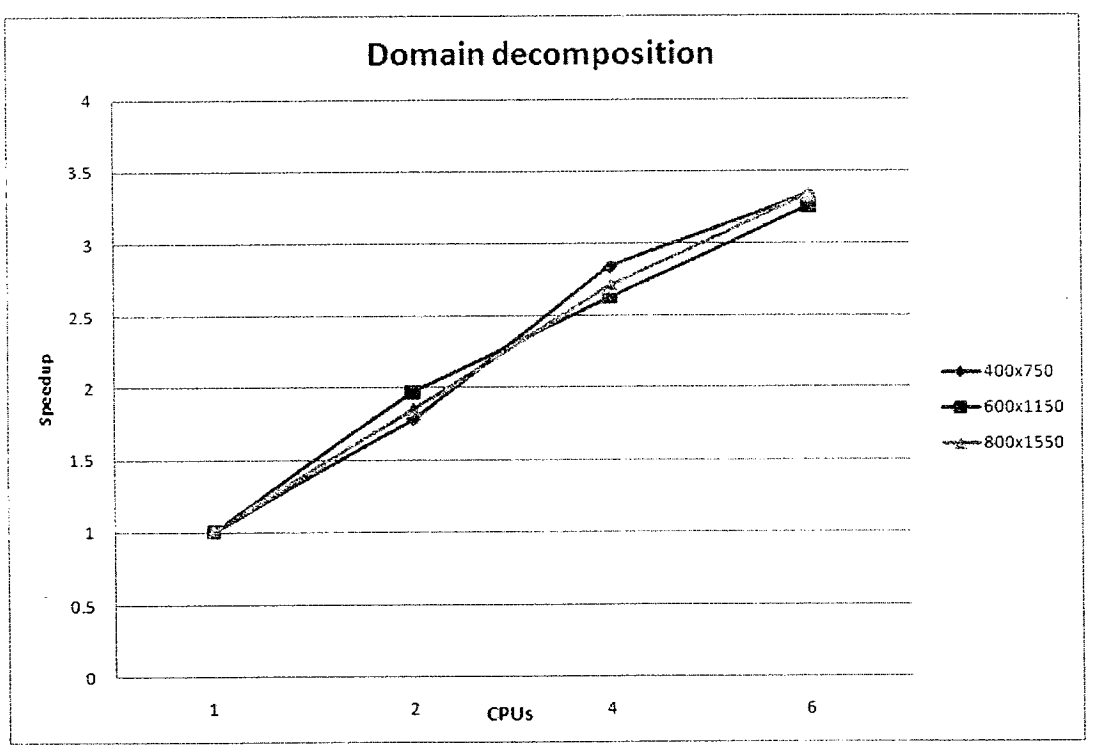
ตารางที่ 4.1 ผลการประมวลผลบนระบบการคำนวณ Parallel-SNRU ตามขนาดของปัญหาแบบ functional decomposition



รูปที่ 4.6 แสดงค่า Speedup เทียบจำนวนหน่วยประมวลผลของ Functional decomposition ตามขนาดของปัญหา

CPUs	size=400x750	speedup	size=600x1150	speedup	size=800x1550	speedup
1	3,094.24	-	7,863.26	-	13,245.32	-
2	1,731.52	1.78	3,987.32	1.97	7,157.26	1.85
4	1,089.07	2.84	2,995.12	2.63	4,892.57	2.71
6	924.20	3.34	2,414.45	3.26	3,971.34	3.34

ตารางที่ 4.2 ผลการประมวลผลบนระบบการคำนวณ Parallel-SNRU ตามขนาดของปัญหาแบบ domain decomposition



รูปที่ 4.7 แสดงค่า Speedup เทียบจำนวนหน่วยประมวลผลของ Domain decomposition ตามขนาดของปัญหา

จากผลที่ได้จะเห็นได้ว่าการคำนวณแบบ Domain decomposition สำหรับปัญหานี้ได้ผล Speedup ที่ดีกว่าแบบ Functional decomposition โดยการแบ่งแบบแรกจำนวนหน่วยประมวลผลที่เหมาะสมอยู่ที่ 4

หน่วยประมวลผล ส่วนแบบที่สองจะเห็นว่ายิ่งเพิ่มจำนวนหน่วยประมวลผล ค่า Speedup ก็จะยิ่งเพิ่มขึ้นตามไปด้วย แต่เนื่องจากข้อจำกัดด้านจำนวนหน่วยประมวลผลของระบบการคำนวณนี้ จึงสามารถแสดงผลได้สูงสุดเพียงแค่ 6 หน่วยประมวลผลเท่านั้น

4.3 งานวิจัยอื่นที่ได้ใช้การประมวลผลจากงานวิจัยนี้

ในส่วนนี้หลังจากที่ได้ทดสอบระบบเรียบร้อยแล้ว ผู้วิจัยยังได้ใช้ระบบการคำนวณนี้ในการประมวลความถูกต้องของการคำนวณทางคณิตศาสตร์ของงานวิจัยใหม่ ชื่อเรื่อง A New Algorithm of Numerical Integration by 3-Points Parabolic Regression Technique ในวารสารนานาชาติ Applied Mathematical Sciences, Vol. 7, 2013, no. 16, 765 - 772 โดยมีผู้ร่วมวิจัยนี้คือ S. Muangchan (ผู้วิจัย), P. Prasertsang, P. Mukwachi และ S. Sompong ดังแสดงในภาคผนวก ข. นอกจากนี้ยังมีงานวิจัยของอาจารย์ศรีจันทร์ ทานะขันธ์ ที่ได้ประมวลผลการคำนวณบนระบบนี้ ชื่อเรื่อง A New Algorithm of Modified Bisection Method for Nonlinear Equation, S. Tanakan, Applied Mathematical Sciences, Vol. 7, 2013, no. 123, 6107 - 6114.

โดยในขณะนี้ผู้วิจัยกำลังดำเนินการวิจัยทางคณิตศาสตร์เกี่ยวกับการหาผลเฉลยทั้งหมดของสมการไม่เชิงเส้นบนช่วง ซึ่งโดยทั่วไปแล้วไม่ว่าจะเป็นวิธีแบ่งช่วงครึ่ง (Bisection method) หรือวิธีของนิวตัน (Newton method) จะได้ผลเฉลยเพียงผลเฉลยเดียวเท่านั้นในการรันโปรแกรมหรือประมวลผลโปรแกรมหนึ่งครั้ง ซึ่งยากต่อการทำให้ได้ผลเฉลยทั้งหมดที่มีอยู่ โดยการดำเนินงานวิจัยเพื่อหาผลเฉลยทั้งหมดของสมการไม่เชิงเส้นนี้กำลังอยู่ระหว่างดำเนินการคิดขั้นตอนวิธีคำนวณแบบใหม่ ซึ่งคาดว่าหลังจากนั้นจะได้นำมาประยุกต์เขียนโปรแกรมสำหรับประมวลผลบนระบบคำนวณแบบขนานของงานวิจัยนี้ต่อไป

4.4 การถ่ายทอดสู่การเรียนการสอน

การถ่ายทอดความรู้สู่การเรียนการสอน ในภาคการศึกษาที่ 1/2553 และ1/2554 ผู้วิจัยได้ใช้ระบบการคำนวณเพื่อเป็นกรณีศึกษาให้แก่นักศึกษาสาขาวิชาคณิตศาสตร์และนักศึกษาสาขาวิชาวิทยาการคอมพิวเตอร์ เป็นส่วนหนึ่งของการเรียนการสอนรายวิชาการวิเคราะห์เชิงตัวเลข ในหัวข้อต่อไปนี้

4.4.1 การอินทิเกรตเชิงตัวเลข

ในกรณีศึกษานี้นักศึกษาจะได้เรียนรู้เกี่ยวกับการอินทิเกรต ซึ่งถ้าหากฟังก์ชันไม่อยู่ในรูปแบบธรรมดาทั่วไปแล้ว การอินทิเกรตจะยากและซับซ้อน ดังนั้นแทนที่จะหาค่าของอินทิกรัลโดยตรง ในการคำนวณจริงสามารถใช้การประมาณค่าแทนได้ดังนี้

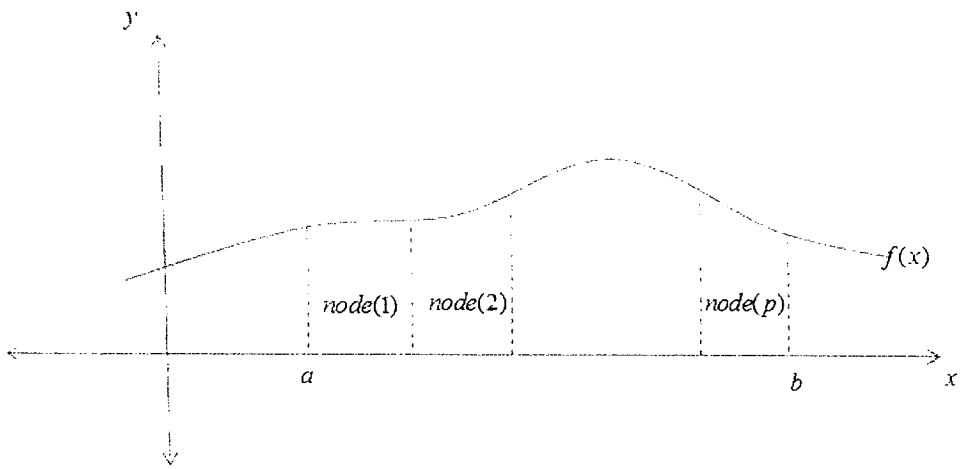
สมมติให้ $a < b$ และพิจารณาให้ $a = c_0 < c_1 < \cdots < c_{p-2} < c_{p-1} = b$ แล้ว

$$\int_a^b f(x)dx = \sum_{k=1}^{p} \int_{c_{k-1}}^{c_k} f(x)dx$$

เมื่อ p แทนจำนวนหน่วยประมวลผล (CPUs) ซึ่งในแต่ละหน่วยประมวลผลจะคำนวณหาค่าต่อไปนี้

$$\int_{c_{k-1}}^{c_k} f(x)dx \approx \sum_{j=1}^{n} w_j f(x_j)$$

เมื่อ $x_j \in [c_{k-1}, c_k]$ และ $w_j = |c_k - c_{k-1}|$ ดังแสดงในภาพด้านล่างนี้ ซึ่งเมื่อทุกหน่วยประมวลผลคำนวณค่าเสร็จสิ้นก็จะส่งคืนค่ากลับมารวมกันซึ่งจะเป็นค่าประมาณของค่าของการอินทิเกรต โดยที่ $n \to \infty$ แล้ว $\int_{c_{k-1}}^{c_k} f(x)dx \to \sum_{j=1}^{n} w_j f(x_j)$ นั่นเอง



รูปที่ 4.8 แสดงแบ่งงานให้หน่วยประมวลผล

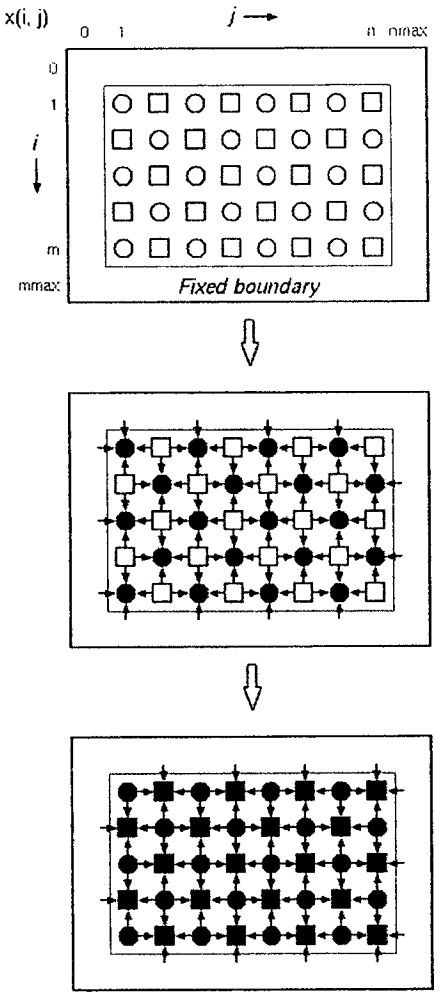
โดยนักศึกษาจะได้เรียนรู้การเขียนโปรแกรมและเปรียบเทียบผล ความถูกต้องของค่าที่ได้กับการอินทิเกรตแบบปกติ

4.4.2 วิธี SOR แบบแดง-ดำ (Red-Black SOR Method)

โดยปกติแล้วในชั้นเรียนนักศึกษาจะได้เรียนเนื้อหาของวิธี SOR แบบปกติ ซึ่งการประมวลผลจะเป็นแบบตามลำดับขั้นตอน (Sequential) แต่การคำนวณจะช้าหากระบบสมการมีขนาดใหญ่ ดังนั้นวิธี SOR แบบแดง-ดำ (Red-Black SOR Method) จึงถูกนำมาใช้งานโดยพัฒนามาจากวิธี SOR แบบดั้งเดิม เริ่มต้นสมาชิกในแต่ละตำแหน่งของเมตริกซ์จะถูกสมมติให้มีสีเป็นสีแดงแดงทั้งหรือดำทั้งหมดก่อน จากนั้นคำนวณตามพิกัดตำแหน่งดังนี้

- ถ้า $i+j$ เป็นเลขคู่ แล้ว $x(i,j)$ เป็นสีแดง
- ถ้า $i+j$ เป็นเลขคี่ แล้ว $x(i,j)$ เป็นสีดำ

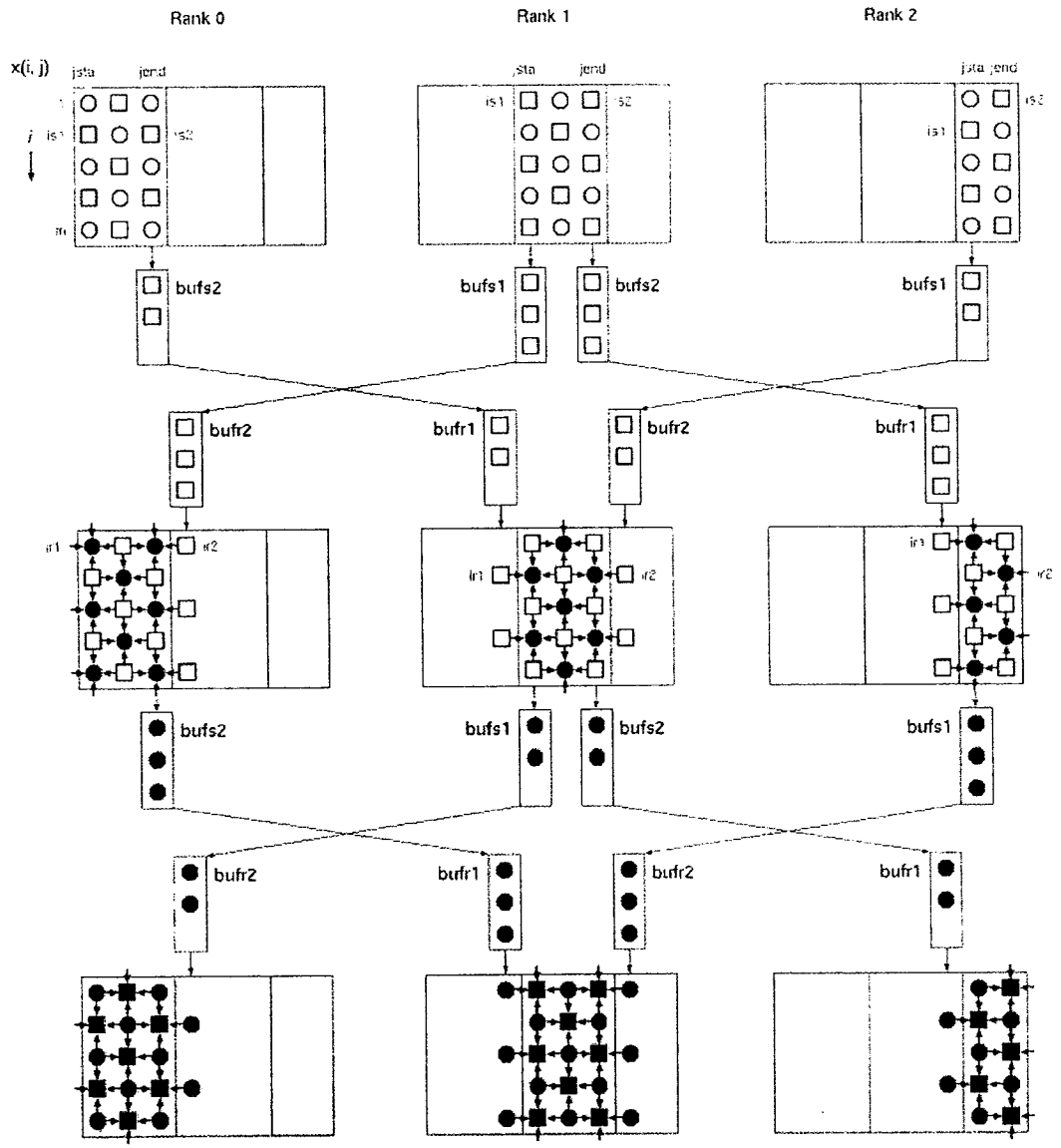
  ดังแสดงในภาพด้านล่างนี้ โดยสีทึบแทนสีดำหมายถึงข้อมูลได้อัพเดจแล้วในรอบของการคำนวณซ้ำ ส่วนสีโปร่งแทนสีแดงหมายถึงข้อมูลของตำแหน่งนี้ยังไม่ได้อัพเดจในรอบการคำนวณซ้ำนี้



รูปที่ 4.9 แสดงการทำงานของการอัพเดจข้อมูลในรอบการคำนวณ

ซึ่งหากระบบสมการมีขนาดใหญ่จะเสียเวลาเป็นอย่างมากในการประมวลผลในแต่ละรอบการคำนวณ ดังนั้นวิธี SOR แบบแดง-ดำ (Red-Black SOR Method) จะแบ่งการคำนวณไปยังหน่วยประมวลผลต่าง ๆ ในระบบ เมื่อประมวลผลเสร็จแล้วก็จะส่งข้อมูลของตำแหน่งที่เกี่ยวข้องเท่านั้นให้กับหน่วย

ประมวลผลอื่น ๆ หลังจากนั้นจึงจะเริ่มการประมวลผลในรอบการคำนวณซ้ำต่อไปดังภาพต่อไปนี้



รูปที่ 4.10 แสดงการแบ่งงานหน่วยประมวลผลในรอบการคำนวณ

ซึ่งนักศึกษาจะได้เข้าไปเขียนโปรแกรมการคำนวณเพื่อทดสอบขั้นตอนวิธีแล้วเปรียบเทียบผลกับวิธีแบบปกติ เพื่อให้เกิดทักษะการเรียนรู้แบบใหม่เพื่อให้สามารถนำไปประยุกต์ใช้แก้ปัญหาทางการคำนวณต่อไป