บทที่ 4

ผลการวิจัย

4.1 เครื่องที่ใช้ทำการวิจัย

เครื่องที่ใช้ทำการวิจัยประกอบด้วยคอมพิวเตอร์คลัสเตอร์ 9 โหนดของภาควิชาคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์ ซึ่งมีรายละเอียดฮาร์ดแวร์ดังนี้ ตัวประมวลผลอินเทลซีออน (Intel Xeon) 2.6 GHz สองตัว หน่วยความจำ 2 GB และฮาร์ดิสก์ขนาด 40 GB แต่ละโหนดเชื่อมโยงหากันผ่านนิคความเร็ว 1 กิ๊กกะบิทและใช้ระบบปฏิบัติการร๊อค (Rock) 4.2.1 ในส่วนของระบบวีซีซีพีมีรายละเอียดดังนี้ โหนดจำนวน 8 โหนดจะถูกติดตั้งเครื่องเสมือนของคีมูทั้งแบบที่ไม่รองรับเช็คพอยต์กับแบบที่รองรับเช็คพอยต์ เครื่องเสมือนแต่ละเครื่องถูกกำหนดให้มีหน่วยความจำขนาด 512 MB และดิสก์อิมเมจ 300 MB (ขยายได้ถึง 4 GB) ในส่วนของตัวประมวลผลใช้เวอร์ชวลโพรเซสเซอร์ที่คีมูจัดเตรียมไว้ให้และใช้ระบบปฏิบัติการเกสเป็น แดมสมอลลินุกซ์ (Dammed Small Linux) 3.4.1

ในส่วนของเครือข่ายเสมือนผู้วิจัยติดตั้งวีดีอีสวิตซ์บนทุก ๆ โหนดของคลัสเตอร์ โดยมีแปดตัวเชื่อมต่อกับไฮเปอร์ไวเซอร์และหนึ่งตัวบนโหนดที่เก้าสำหรับให้โหนดทั้งแปดเชื่อมต่อมา ซึ่งจากการทดลองทำให้ผู้วิจัยทราบว่าเครือข่ายเสมือนของโพรโตไทป์นี้มีข้อกำจัดบางประการดังจะแสดงให้เห็นต่อไป

ผู้วิจัยเลือกใช้เคอร์เนลของแนสกับโอเพนเอ็มพีไอ (OpenMPI) 1.2.3 เพื่อทำการทดลอง โดยข้อมูลไบนารี่ (binary) ของเบนช์มาร์กจะถูกแจกจ่ายไปบนดิสก์อิมเมจของทุก ๆ เครื่องเสมือน สำหรับผลการทดลองที่นำเสนอในบทนี้เป็นค่าเฉลี่ยของการทดลอง 10 ครั้ง ส่วนข้อมูลผลการทดลองทั้งหมดถูกจัดไว้ในภาคผนวก

4.2 โอเวอร์เฮด

ในการทดลองชุดที่หนึ่งเป็นการสำรวจโอเวอร์เฮดของวีซีซีพีโดยการเปรียบเทียบเวลาการประมวลผลเบนช์มาร์กของแนสกับคีมูและไม่มีการทำเช็คพอยต์/รีโคเวอร์รี สมมติฐานของ

งานวิจัยนี้คือ โอเวอร์เฮดของระบบวีซีซีพีไม่ควรส่งผลกระทบต่อสมรรถนะของระบบมากอย่างมีนัยสำคัญ

ตารางที่ 4.1

โอเวอร์เฮดของวีซีซีพีบนแนสเคอร์เนลอีพี, ซีจี, ไอเอส, และเอ็มจี

Kernel EP	*2 nodes*	*4 nodes*	8 nodes
VCCP cluster	57.78	28.71	14.63
QEMU cluster	57.41	28.7	14.59
Overheads	0.37	0.01	0.04
Overheads %	0.64%	0.03%	0.27%

Kernel CG	*2 nodes*	*4 nodes*	8 nodes
VCCP cluster	91.02	128.68	110.45
QEMU cluster	90.37	124.57	109.74
Overheads	0.65	4.11	0.71
Overheads %	0.72%	3.30%	0.65%

Kernel IS	*2 nodes*	*4 nodes*	8 nodes
VCCP cluster	2.48	1.73	1.46
QEMU cluster	2.45	1.69	1.42
Overheads	0.03	0.04	0.04
Overheads %	1.22%	2.37%	2.82%

Kernel MG	*2 nodes*	*4 nodes*	8 nodes
VCCP cluster	7.88	17.05	21.29
QEMU cluster	7.7	16.87	21.17
Overheads	0.18	0.18	0.12
Overheads %	2.34%	1.07%	0.57%

Kernel EP	*2 nodes*	*4 nodes*	8 nodes
VCCP cluster	57.78	28.71	14.63
REAL cluster	51.39	25.73	12.92
Overheads	6.39	2.98	1.71
Overheads %	12.43%	11.58%	13.24%

จากตารางที่ 4.1 แสดงให้เห็นว่าโอเวอร์เฮดของวีซีซีพีจากการรันแนสเคอร์เนลมีน้อยมาก ทำให้เวลาที่ใช้ทดสอบออกมาในลักษณะแบบเดียวกับคีมูคลัสเตอร์โดยโอเวอร์เฮดที่มากที่สุดคือ 3.3% และน้อยที่สุดคือ 0.03% ซึ่งค่อนข้างต่ำตรงตามสมมติฐานที่คาดหวัง และผลการทดลองนี้ยังแสดงให้เห็นถึงข้อจำกัดของเครือข่ายวีดีอีบนโปรแกรมประยุกต์ที่มีการติดต่อสื่อสารกันในระหว่างทำงานอย่างเข้มข้นแบบเอ็มจีและซีจีอันเนื่องมาจากปัญหาสองประการดังนี้

1. การแก่งแย่งทรัพยากรระหว่างวีเอ็มและวีดีอีสวิตซ์ในขณะการประมวลผลเคอร์เนลที่ต้องใช้การติดต่อสื่อสารอย่างเข้มข้น ดังเห็นได้จากวีดีอีสวิตซ์มีการรายงานเกี่ยวกับ “temporarily unavailable socket resources” เป็นระยะ ปัญหานี้อาจเกิดขึ้นมาจากวีเอ็มหรือวีดีอีมีการส่งเฟรมไปสู่ซ็อคเก็ตเร็วเกินไปจนทำให้เน็ตเวิร์คบัฟเฟอร์ (network buffer) โอเวอร์โหลดไปจนถึงการบริโภคซีพียูไทม์ (CPU time) ที่มากเกินไปของวีเอ็มทำให้วีดีอีไม่ได้รับซีพียูไทม์ที่เพียงพอสำหรับการระบายเฟรมข้อมูลบนเครือข่าย

2. เกิดคอขวด (bottleneck) ขึ้นที่วีดีอีสวิตซ์กลางในขณะการประมวลผลโปรแกรมประยุกต์ที่ต้องใช้การติดต่อสื่อสารอย่างเข้มข้น จึงเป็นผลให้เวลาที่ใช้ประมวลผลของซีจีและเอ็มจีไม่ลดลงเมื่อจำนวนคอมพิวต์โหนดเพิ่มสูงขึ้น ถึงแม้ว่าไอเอสจะมีการติดต่อสื่อสารแต่ปริมาณเฟรมที่ส่งออกไปยังคงอยู่ในระดับที่เครือข่ายรองรับไหว

เพื่อยืนยันให้เห็นว่าปัญหาเกิดที่เครือข่ายวีดีอีผู้วิจัยจึงแสดงผลเปรียบเทียบของระบบวีซีซีพีกับระบบจริงไว้ในตารางที่ 4.1 ซึ่งเห็นได้ชัดเจนว่าในอีพีซึ่งไม่มีการติดต่อสื่อสารระหว่างประมวลผลคลัสเตอร์เสมือนให้ประสิทธิภาพน้อยกว่าเครื่องจริงประมาณ 12 % อันเนื่องมาจากเวอร์ชวลไลเซชันโอเวอร์เฮด

4.3 สมรรถนะการเช็คพอยต์

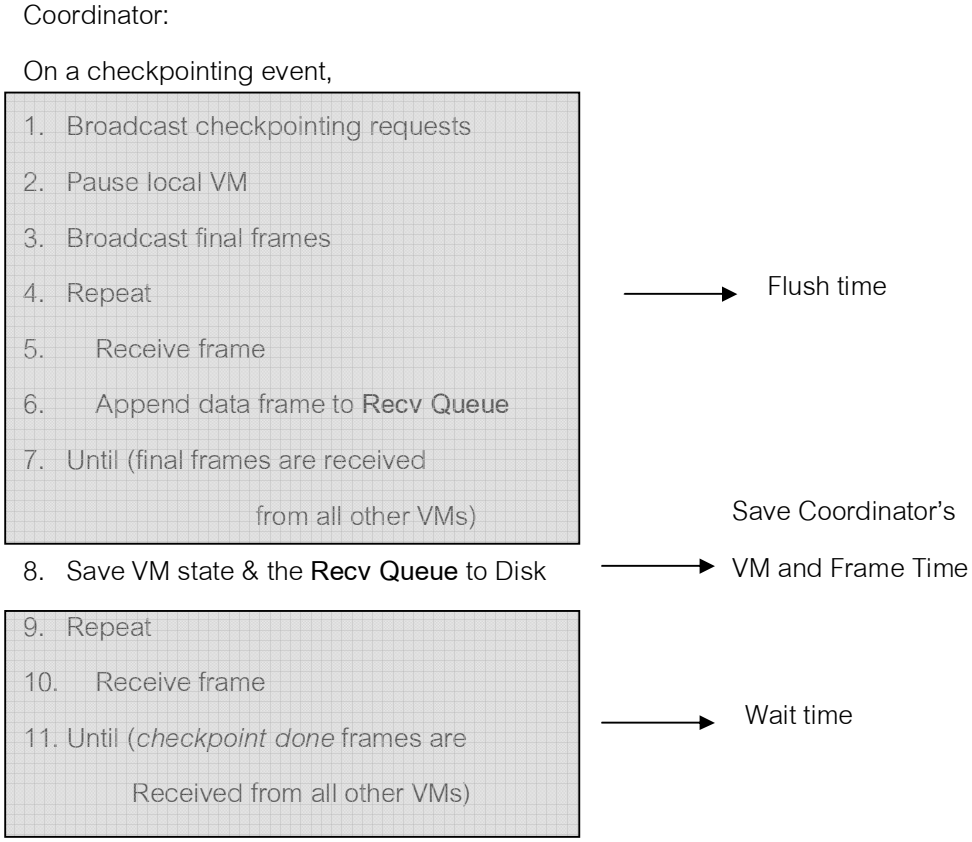
การทดลองนี้ผู้วิจัยสั่งให้ทำโคออดิเนตเช็คพอยต์ในขณะรันเบนช์มาร์กบนคลัสเตอร์เสมือน 2, 4, และ 8 โหนดโดยแต่ละกลุ่มจะทำการทดลอง 10 ครั้ง ผลการทำลองนี้จำแนกเวลาออกเป็นส่วนย่อย ๆ ตามลำดับขั้นตอนเช็คพอยต์บนโคออดิเนเตอร์ ดังนั้นพวกเราจึงนิยามเวลาเช็คพอยต์ (checkpointing time) ดังต่อไปนี้

Checkpointing time = Flush Time + Save VM + Save Frames + Wait Time

โดยเวลาชะล้าง (flush time) แสดงถึงช่วงเวลาตั้งแต่โคออดิเนเตอร์เผยแพร่เช็คพอยต์ติ้งรีเควสไปจนกระทั่งมันได้รับไฟนอลเฟรมจากทุก ๆ เครื่องเสมือน บันทึกวีเอ็ม (save vm) แสดงถึงเวลาทั้งหมดที่โคออดิเนเตอร์ใช้บันทึกสเตทของวีเอ็มลงสู่เช็คพอยต์ไฟล์ บันทึกเฟรม (save frames) แสดงถึงเวลาทั้งหมดที่โคออดิเนเตอร์ใช้บันทึกเฟรมบนรีซีฟคิวลงสู่เช็คพอยต์ไฟล์ สุดท้ายเวลารอคอย (wait time) แสดงถึงเวลาทั้งหมดที่โคออดิเนเตอร์ต้องรอคอยหลังจากบันทึกรีซีฟคิวเสร็จ

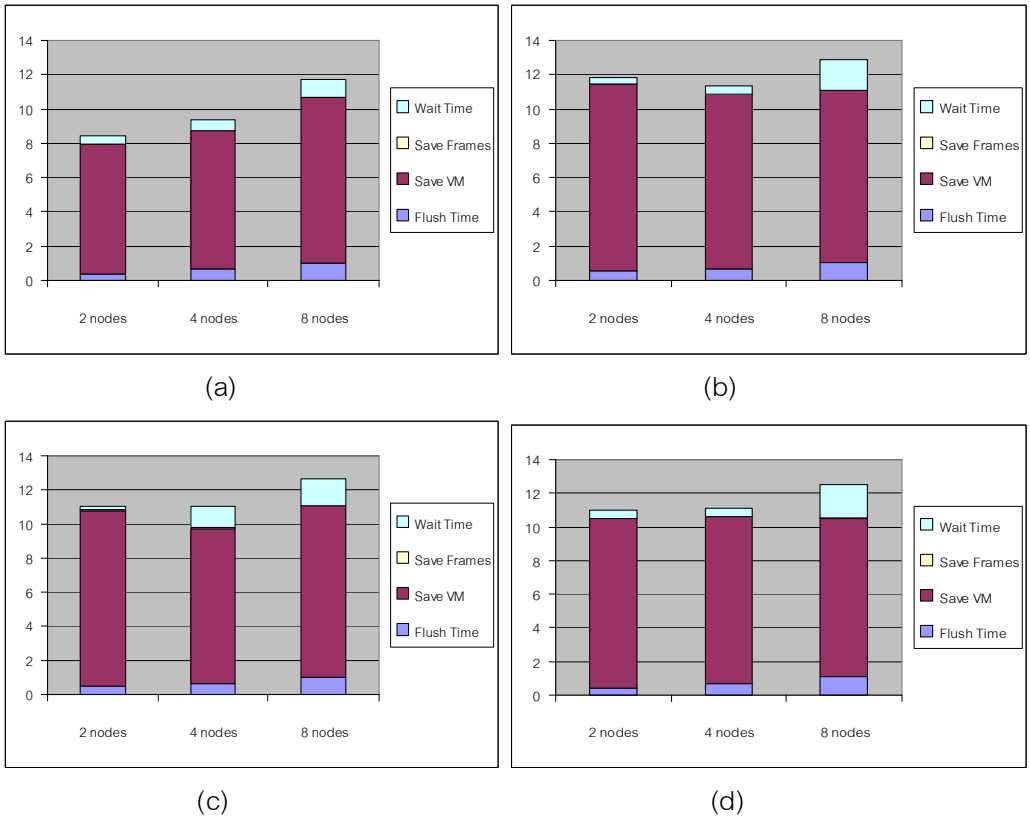
จนกระทั่งได้รับเช็คพอยต์ดันจากทุก ๆ เครื่องเสมือนและรีซูมการประมวลผล ซึ่งสามารถสรุปได้ดังภาพที่ 4.1 และผลการทดลองแสดงในภาพที่ 4.2

ภาพที่ 4.1
ภาพแสดงเวลาเช็คพอยต์ที่สัมพันธ์กับโพรโทคอล



ภาพที่ 4.2 แสดงผลการทดสอบสมรรถนะด้วยแนสโดยแกน x บอกจำนวนโหนดที่ใช้ ส่วนแกน y แสดงเวลาที่ใช้เช็คพอยต์มีหน่วยเป็นวินาที ผลการทดลองแสดงให้เห็นว่าเวลาชะล้างช่องสื่อสารเพิ่มสูงขึ้นเมื่อจำนวนโหนดที่เกี่ยวข้องเพิ่มขึ้น เนื่องมาจากปริมาณการติดต่อสื่อสารที่เพิ่มสูงขึ้นตามจำนวนโหนด

ภาพที่ 4.2
สมรรถนะการเช็คพอยต์ของ (a) เคอร์เนลอีพี (b) เคอร์เนลซีจี
(c) เคอร์เนลไอเอส และ (d) เคอร์เนลเอ็มจี

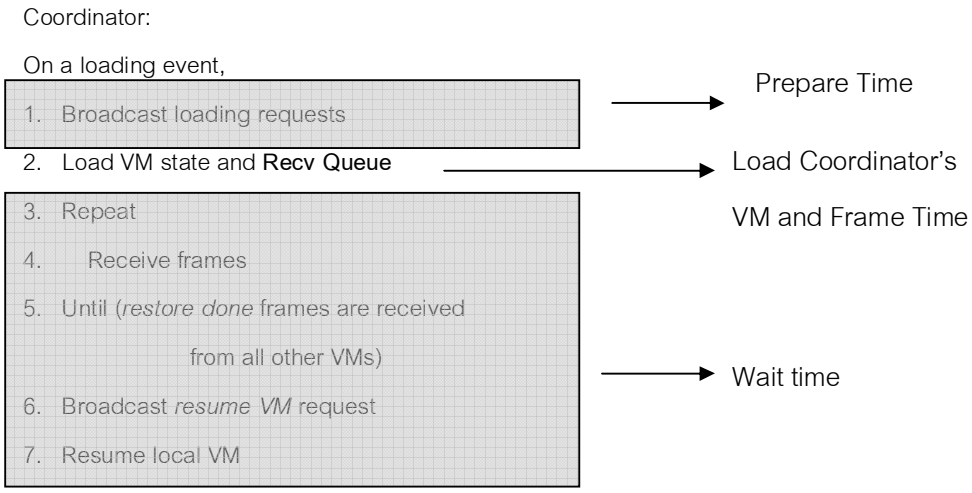


ในทุก ๆ การทดลองแสดงให้เห็นว่าเวลาส่วนใหญ่ของการเช็คพอยต์ถูกใช้ไปกับการบันทึกสเตทของเครื่องเสมือนตั้งแต่ 7.54 ถึง 10.9 วินาที คิดเป็นเปอร์เซ็นต์คือ 75 ถึง 82 เปอร์เซ็นต์ ปัจจัยสำคัญที่ส่งผลต่อเวลาบันทึกสเตทคือ ขนาดของหน่วยความจำหลักของเครื่องเสมือนและขนาดสแน๊ปช๊อตของดิสก์อิมเมจ

ในส่วนของการบันทึกเฟรมข้อมูลนั้นแทบจะไม่มีนัยสำคัญเมื่อเปรียบเทียบกับขนาดของเวลาที่ใช้บันทึกสเตทของวีเอ็ม เนื่องมาจากปริมาณเฟรมไม่มากนักถึงแม้ว่าจะเป็นโปรแกรมประยุกต์ที่มีการสื่อสารเข้มข้นอย่างเอ็มจีก็ใช้เวลาบันทึกเฟรมมากที่สุดเพียง 0.09 วินาทีเท่านั้น

ในส่วนเวลารอคอยนั้นจะสูงขึ้นตามจำนวนโหนดในคลัสเตอร์ ที่เพิ่มขึ้นนี้ เนื่องมาจากมีบางโหนดในคลัสเตอร์เสมือนทำการบันทึกสเตทของวีเอ็มช้ากวาโหนดอื่นดังแสดงไว้ในภาคผนวก ง

ภาพที่ 4.3
ภาพแสดงเวลารีโคเวอร์รีที่สัมพันธ์กับโพรโทคอล



4.4 สมรรถนะการรีโคเวอร์รี

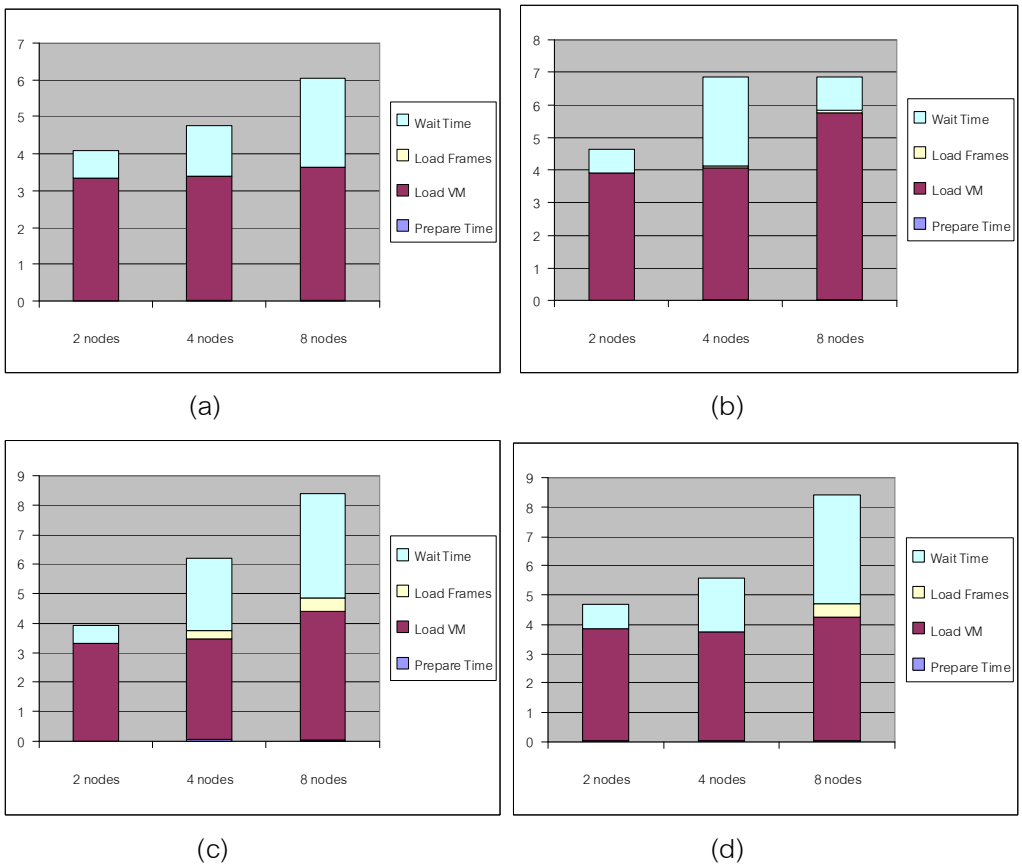
การทดลองนี้ผู้วิจัยสั่งให้คลัสเตอร์เสมือนทำรีโคเวอร์รีเพื่อกู้ระบบขึ้นมาในตำแหน่งเช็คพอยต์ล่าสุดโดยทดสอบบนคลัสเตอร์เสมือนขนาด 2, 4, และ 8 โหนดเป็นจำนวน 10 ครั้งในแต่ละกลุ่ม และเสนอผลการทดลองโดยจำแนกเวลาออกเป็นส่วนย่อย ๆ ตามลำดับขั้นตอนรีโคเวอร์รีบนโคออดิเนเตอร์ ดังนั้นพวกเราจึงนิยามเวลารีโคเวอร์รี (recovery time) ดังต่อไปนี้

Recovery time = Prepare Time + Load VM + Load Frames + Wait Time

โดยเวลาเตรียมพร้อม (prepare time) เป็นเวลาที่โคออดิเนเตอร์เผยแพร่โหลดเฟรมให้กับโหนดอื่น ๆ ไปจนถึงเวลาที่โคออดิเนเตอร์พร้อมโหลดเช็คพอยต์ โหลดวีเอ็ม (load vm) แสดงถึงเวลาทั้งหมดที่โคออดิเนเตอร์ใช้โหลดสเตทของวีเอ็มจากเช็คพอยต์ไฟล์ โหลดเฟรม (load frames)

แสดงถึงเวลาทั้งหมดที่โคออดิเนเตอร์ใช้โหลดเฟรมจากเช็คพอยต์ไฟล์ลงสู่รีซีฟคิว สุดท้ายเวลารอคอย (wait time) แสดงถึงเวลาทั้งหมดที่โคออดิเนเตอร์ต้องรอคอยหลังจากโหลดข้อมูลลงคิวเสร็จจนกระทั่งได้รับรีโคเวอร์รีดันจากทุก ๆ เครื่องเสมือนและรีซูมการประมวลผล ซึ่งสามารถสรุปได้ดังภาพที่ 4.3

ภาพที่ 4.4
สมรรถนะการรีโคเวอร์รีของ (a) เคอร์เนลอีพี (b) เคอร์เนลซีจี
(c) เคอร์เนลไอเอส และ (d) เคอร์เนลเอ็มจี



ภาพที่ 4.4 แสดงผลการทดสอบสมรรถนะด้วยแนสโดยแกน x บอกจำนวนโหนดที่ใช้ ส่วนแกน y แสดงเวลาที่ใช้รีโคเวอร์รีมีหน่วยเป็นวินาที ผลการทดลองแสดงให้เห็นว่าเวลาเตรียมพร้อมมีลักษณะคงที่ เนื่องมาจากในขั้นตอนนี้เป็นเพียงการส่งโหลดเฟรมไปแจ้งโหนดต่าง ๆ ในระบบเท่านั้น ในส่วนของเวลารีโคเวอร์รีและอ่านเฟรมเป็นไปในทำนองเดียวกับการทดลอง

เช็คพอยต์แต่ในส่วนของเวลารอคอยกลับเพิ่มสูงขึ้นอย่างมากจนใกล้เคียงกับเวลารีโคเวอร์รี ซึ่งตรงนี้ก็เกิดมาจากความแตกต่างของเวลารีโคเวอร์รีของโหนดที่เร็วที่สุดกับช้าที่สุดเช่นเดียวกัน โดยข้อมูลการทดลองวัดเวลาหนึ่งครั้งของทั้งแปดโหนดถูกแสดงไว้ในภาคผนวก ง

4.5 การสรุปผลและงานในอนาคต

งานวิจัยนี้เสนอแนวทางใหม่สำหรับการออกแบบและสร้างการทนทานต่อความผิดพร่องให้กับระบบเอชพีซี เป้าหมายสูงสุงผู้วิจัยคือ การสร้างระบบที่มีความโปร่งใสสูงเพียงพอสำหรับซ่อนกลไกการเช็คพอยต์/รีโคเวอร์รีออกจากระบบปฏิบัติการและโปรแกรมประยุกต์ สำหรับผลลัพธ์ที่ได้จากงานวิจัยนี้คือ สถาปัตยกรรมของคลัสเตอร์เสมือนและระบบวีซีซีพีที่ปฏิบัติตามโพรโทคอลที่เสนอ ผู้วิจัยได้อธิบายความถูกต้องและข้อจำกัดบางประการของงานวิจัยชิ้นนี้ซึ่งเกิดมาจากคลัอกสกูร์ว

ผู้วิจัยทำการพัฒนาระบบโพรโตไทป์ขึ้นโดยใช้คีมูเป็นฐานการพัฒนาและทำการทดลองด้วยเคอร์เนลสี่ตัวของแนส ผลการทดลองโดยไม่มีการทำเช็คพอยต์หรือรีโคเวอร์รีแสดงให้เห็นว่ามีโอเวอร์เฮดต่ำเมื่อเปรียบเทียบเวลาประมวลผลของวีซีซีพีคลัสเตอร์กับคีมูคลัสเตอร์ รวมทั้งยังแสดงให้เห็นอีกว่าเวลาการประมวลผลแนสเคอร์เนลบางตัวไม่ดีนักเมื่อเพิ่มจำนวนวีเอ็มโหนด อันเนื่องมาจากเครือข่ายวีดีอีของโพรโตไทป์นี้ยังมีความสามารถไม่เพียงพอในการจัดส่งเฟรมข้อมูลจำนวนมากของโปรแกรมประยุกต์แบบขนาดและไม่ได้เกิดมาจากวีดีอีแต่อย่างใด ในส่วนสมรรถนะการเช็คพอยต์มีแนวโน้มขึ้นอยู่กับจำนวนวีเอ็มโหนดและความไม่สอดคล้องกันของการบันทึกเสตทของวีเอ็มในแต่ละเครื่อง ซึ่งเวลาที่ใช้ในการบันทึกสเตทนั้นมีขนาดใหญ่เมื่อเปรียบเทียบกับเวลาในส่วนอื่นแต่อย่างไรก็ตามเวลานี้ควรขึ้นอยู่กับขนาดของหน่วยความจำของวีเอ็มและดิสก์อิมเมจไม่ใช่จำนวนโหนดในระบบ ในส่วนสมรรถนะการรีโคเวอร์รีมีลักษณะคล้ายกับการเช็คพอยต์เว้นแต่ว่าเวลาในการรอคอยมีผลกระทบต่อเวลาโดยรวม อันเนื่องมาจากเวลาที่โหนดต่าง ๆ ใช้โหลดเสตทมีความแตกต่างกันมากจนทำให้โคออดิเนเตอร์ต้องรอคอยเป็นเวลานาน

สำหรับงานในอนาคตผู้วิจัยวางแผนปรับปรุงเครือข่ายของคลัสเตอร์เสมือนโดยการสร้างการเชื่อมโยงโดยตรงระหว่างวีเอ็มแทนการใช้วีดีอีสวิตช์และสวิตช์กลาง ค้นหาความเป็นไปได้ในการทำเวอร์ชวลไอ/โอ (virtual I/O) กับวีซีซีพี ยิ่งไปกว่านั้นค้นหากลไกที่ช่วยย่นระยะเวลาที่ใช้บันทึกและโหลดสเตทของเครื่องเสมือน