

บทที่ 3

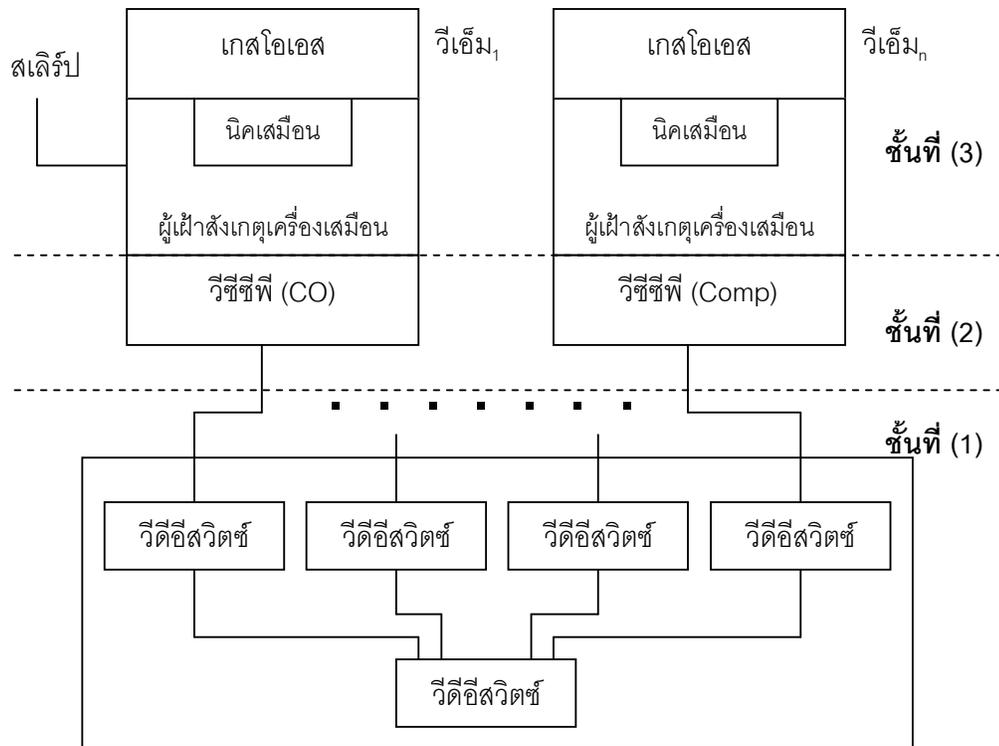
วิธีการวิจัย

3.1 สถาปัตยกรรมของคลัสเตอร์เสมือน

สำหรับคลัสเตอร์เสมือนต้นแบบที่ใช้ในการทดลองผู้วิจัยนิยามว่า คลัสเตอร์เสมือนประกอบด้วยซอฟต์แวร์ 3 ชั้นคือ 1) ชั้นเครือข่ายเสมือน 2) ชั้นวีซีซีพี และ 3) ชั้นเครื่องเสมือน ดังภาพที่ 3.1 สำหรับโพรโตไทป์ในชั้นที่หนึ่งนี้เป็นการเชื่อมโยงกันของวีดีอีสวิทช์ (VDE switch) ในทอพอโลยี (topology) แบบดวงดาว (star) โดยวีดีอีสวิทช์โพรเซสจะถูกสั่งให้ทำงานบนโฮสคอมพิวเตอร์ของคลัสเตอร์เสมือนและสวิทช์เหล่านั้นจะถูกสั่งให้เชื่อมโยงไปที่สวิทช์กลางบนคอมพิวเตอร์อีกเครื่องหนึ่ง ด้วยการออกแบบนี้เฟรมที่ถูกส่งออกมาจะต้องเดินทางไปยังสวิทช์กลางก่อนจึงจะถูกจัดส่งไปยังเครื่องปลายทาง เมื่อพิจารณากระบวนการทำงานในระบบเครือข่ายทำให้พบว่าเฟรมข้อมูลจากเครื่องเสมือนจะถูกส่งไปยังวีดีอีสวิทช์บนโฮสผ่านทางแทป (TAP) ซึ่งเป็นไฟล์เดสคริปเตอร์ (file descriptor) แบบลำดับ ต่อมาวีดีอีสวิทช์บนเครื่องโฮสจะอ่านเฟรมออกมาจากแทปตามลำดับที่ถูกส่งออกมาและส่งต่อไปยังสวิทช์กลางผ่านทางวีดีอีปลั๊ก (VDE plug) ด้วยเอสเอสเอช (ssh) โพรโทคอลที่มีการรับประกันเฟรมข้อมูลไม่สูญหายหรือผิดลำดับ เมื่อสวิทช์กลางได้รับเฟรมทางวีดีอีปลั๊กแล้วเฟรมดังกล่าวจะถูกส่งไปยังเครื่องเสมือนปลายทางทันทีด้วยเส้นทางที่กำหนดไว้ (ด้วยสตาร์ทอพอโลยีเครือข่ายนี้ไม่มีมัลติพาท (multipath)) ด้วยกระบวนการทำงานนี้ทำให้การส่งเฟรมระหว่างเครื่องเสมือนสองคู่ใด ๆ ให้การรับประกันว่าเป็นแบบเข้าก่อนออกก่อนและเชื่อถือได้

ชั้นซอฟต์แวร์ลำดับที่สองเป็นที่อยู่ของวีซีซีพีซอฟต์แวร์ซึ่งมีหน้าที่มอนิเตอร์ (monitor) อีเทอร์เน็ตเฟรมทั้งหมดที่เข้าและจากเครื่องเสมือน ไปจนถึงการทำหน้าที่จัดส่งเฟรมประสานงานสำหรับการเซ็คพอยต์/รีโคเวอรี่ ด้วยหน้าที่ดังกล่าววีซีซีพีจึงถูกจัดวางไว้ในตำแหน่งระหว่างไฮเปอร์ไวเซอร์และวีดีอีสวิทช์ ผู้วิจัยตั้งสมมติฐานว่าทุก ๆ วีซีซีพีรู้เลขที่อยู่ไอพีของทุก ๆ เครื่องเสมือนบนเครือข่ายวีดีอี ในงานวิจัยนี้วีซีซีพีถูกจำแนกออกเป็นสองชนิดคือ โคออดิเนเตอร์ (CO) และคอมพิวเตอร์ (Comp) ที่เซดโหนดของคลัสเตอร์เสมือนจะติดตั้งโคออดิเนเตอร์สำหรับทำหน้าที่เริ่มต้นและดูแลการทำโคออดิเนตเซ็คพอยต์หรืออีกนัยหนึ่งวีซีซีพีคอมพิวเตอร์ต้องปฏิบัติตามคำสั่งจากวีซีซีพีโคออดิเนเตอร์ในระหว่างการเซ็คพอยต์/รีโคเวอรี่

ภาพที่ 3.1
ชั้นซอฟต์แวร์ในคลัสเตอร์เสมือน



ชั้นซอฟต์แวร์ที่สามนิยามเซตของเครื่องเสมือน โดยแต่ละเครื่องมีไฮเปอร์ไวเซอร์ที่ทำหน้าที่จำลองฮาร์ดแวร์ให้แก่แกสโอเอสและโปรแกรมประยุกต์ เครื่องเสมือนเครื่องหนึ่งจะถูกกำหนดให้เป็นเซตโหนดของคลัสเตอร์ ส่วนเครื่องที่เหลือเป็นคอมพิวต์โหนดซึ่งแต่ละโหนดต้องมี นิค (NIC: Network Interface Card) อันหนึ่งที่ใช้โต้ตอบกับวีซีซีพีและเครือข่ายเสมือน สำหรับเซตโหนดต้องการนิกอีกอันหนึ่งเพื่อเชื่อมโยงสู่อินเทอร์เน็ตผ่านสลิร์ป (slirp) โพรโทคอล

3.2 โพรโทคอล

ผู้วิจัยออกแบบให้รีซีฟคิว (receive queue) ถูกสร้างขึ้นไว้ในวีซีซีพี โดยกำหนดให้เฟรมจากเครือข่ายที่ดีจะต้องถูกส่งไปที่ท้ายคิวและถูกอ่านโดยเครื่องเสมือนจากทางหัวคิว ดังนั้นถึงแม้ว่าจะมีเฟรมจากเครือข่ายมาในขณะที่เครื่องเสมือนจะไม่พร้อมประมวลผลเฟรมข้อมูลนั้นก็จะไม่สูญหายและผิดลำดับ และเพื่อให้โปรโทคอลทำงานประสบความสำเร็จผู้วิจัยจึงกำหนดสมมติฐานว่าขณะที่โปรโทคอลทำงานเครื่องเสมือนและเครือข่ายเสมือนต้องสามารถปฏิบัติงานได้ถูกต้อง โปรแกรมประยุกต์ต้องไม่พึ่งพาเวลาจริงของนาฬิกา (real time clock) ในการทำงานและสุดท้ายถ้าหากมีความผิดพลาดใด ๆ เกิดขึ้น คลัสเตอร์เสมือนต้องรีโคเวอรี่จากเช็คพอยต์ล่าสุด

ภาพที่ 3.2
เช็คพอยต์โปรโทคอล

Coordinator:	Compute:
On a checkpointing event,	Upon receiving a checkpoint request,
1. Broadcast checkpointing requests	1. Pause local VM
2. Pause local VM	2. Broadcast final frames
3. Broadcast final frames	3. Repeat
4. Repeat	4. Receive frame
5. Receive frame	5. Append data frame to Recv Queue
6. Append data frame to Recv Queue	6. Until (final frames are received
7. Until (final frames are received	from all other VMs)
from all other VMs)	7. Save VM state & the Recv Queue to
8. Save VM state & the Recv Queue to	Disk
Disk	8. Send <i>checkpoint done</i> frame to
9. Repeat	Coordinator
10. Receive frame	9. Wait until receiving a resume frame
11. Until (<i>checkpoint done</i> frames are	10. Resume VM

3.2.1 เซ็คพอยต์ไฟรโทคอล

ภาพที่ 3.2 แสดงเซ็คพอยต์ไฟรโทคอลของโคออดิเนเตอร์ (CO) และคอมพิวต์ (Comp) ซึ่งกระบวนการเซ็คพอยต์จะเริ่มต้นโดยโคออดิเนเตอร์ (CO) เผยแพร่เซ็คพอยต์ตั้งรีเคส (checkpointing request) ไปสู่ทุกโหนดในคลัสเตอร์เสมือน เมื่อได้รับรีเคสแล้ววีซีพีจะสั่งให้ไฮเปอร์ไวเซอร์หยุดการประมวลผลเพื่อสงวนสแตทของวีเอ็มและหยุดการส่งเฟรมออกไปสู่เครือข่าย ในส่วนของโคออดิเนเตอร์หลังจากเผยแพร่เซ็คพอยต์ตั้งรีเคสแล้วก็จะทำแบบเดียวกันต่อมาทุก ๆ โหนดจะประสานงานกันชะล้างเฟรมที่ตกค้างในเครือข่ายออกไปโดยการส่งเฟรมเฟรม (final frame) ไปยังทุกช่องสื่อสารและรอคอยเฟรมเฟรมของโหนดอื่นด้วยคุณสมบัติเข้าก่อนออกก่อนของเครือข่ายที่ดีจึงให้การรับประกันว่าเฟรมที่กำลังเดินทางจะต้องถูกรับโดยวีเอ็มปลายทางก่อนที่เฟรมเฟรมจะมาถึง โดยตลอดกระบวนการนี้ถ้าหากมีเฟรมข้อมูลเข้ามาถึงวีเอ็มเฟรมดังกล่าวจะถูกจับเก็บลงที่ท้ายรีซีฟคิว

หลังจากการชะล้างช่องสื่อสารเสร็จสมบูรณ์สแตทของคลัสเตอร์เสมือนจะเป็นคอนซิสเทนต์โกลบอลสแตทและพร้อมสำหรับการเซ็คพอยต์ด้วยสองขั้นตอนต่อไปนี้ ขั้นตอนหนึ่งไฮเปอร์ไวเซอร์ทำการบันทึกสแตทของวีเอ็มและข้อมูลที่อยู่ภายในรีซีฟคิวลงสู่ดิสก์ ขั้นตอนที่สองทำการบันทึกสแตทของดิสก์อิมเมจ (disk image) ด้วยระบบไฟล์แบบสแตก (stack file system)

เมื่อเซ็คพอยต์ถูกจัดเก็บลงสู่บันทึกข้อมูลที่มีเสถียรภาพแล้วคอมพิวต์โหนดจะส่งเซ็คพอยต์ตัน (checkpoint done) เพื่อแจ้งโคออดิเนเตอร์ว่ากระบวนการทำโลคอลเซ็คพอยต์ของมันเสร็จเรียบร้อย เมื่อโคออดิเนเตอร์ได้รับเซ็คพอยต์ตันครบทุกโหนดแล้ววีซีพีจะสั่งให้ไฮเปอร์ไวเซอร์เผยแพร่ออกไปเพื่อสั่งให้ทุกโหนดทำงานตามปกติ

3.2.2 รีโควเรอรีไฟรโทคอล

ภาพที่ 3.3 แสดงรีโควเรอรีไฟรโทคอลสำหรับโคออดิเนเตอร์และคอมพิวต์โหนดตามลำดับ การรีโควเรอรีจะดำเนินการโดยโคออดิเนเตอร์เผยแพร่โหลดตั้งรีเคส (loading request) ไปสู่โหนดภายในคลัสเตอร์เสมือน หลังจากคอมพิวต์โหนดได้รับรีเคสและหลังจากโคออดิเนเตอร์เผยแพร่รีเคสเสร็จกระบวนการประมวลผลของมันจะถูกหยุดลงและทำการอ่านเซ็คพอยต์ในส่วนของวีเอ็มกลับมา จากนั้นรีซีฟคิวจะถูกเคลียร์แล้วถูกเติมด้วยเฟรมในเซ็คพอยต์ที่ท้ายคิว เมื่อการอ่านเซ็คพอยต์เสร็จสมบูรณ์คอมพิวต์โหนดจะส่งรีโสตร์ตัน (restore done) ไปสู่โคออดิเนเตอร์ เมื่อโคออดิเนเตอร์ได้รับรีโสตร์ตันมาจากคอมพิวต์โหนดทั้งหมดแล้วมันจะเผยแพร่

ริซูมวีเอ็ม (resume VM) ไปสู่โคออดิเนเตอร์เพื่อดำเนินการประมวลผลต่อจากเช็คพอยต์ที่อ่านขึ้นมา

ภาพที่ 3.3
รีโคเวอริโพรโทคอล

Coordinator:	Compute:
On a loading event,	Upon receiving a <i>loading</i> request,
1. Broadcast loading requests	1. Load VM and Recv Queue
2. Load VM state and Recv Queue	2. Send <i>restore done</i> frame to Coordinator
3. Repeat	3. Wait until receiving a <i>resume VM</i> frame
4. Receive frames	4. Resume VM
5. Until (<i>restore done</i> frames are received from all other VMs)	
6. Broadcast <i>resume VM</i> request	
7. Resume local VM	

3.3 การสร้างระบบวิชีซีพี

ผู้วิจัยได้ทำการดัดแปลงคิมูซอฟต์แวร์เพื่อเพิ่มฟังก์ชันการทำงานของวิชีซีพีให้กับฟังก์ชันการจัดการไอ/โอ (handle I/O) ของคิมู เมื่อพิจารณาจากภาพที่ 3.1 จะพบว่าการดัดแปลงนั้นถูกกระทำบนไฮเปอร์ไวเซอร์ในชั้นที่ 3 ส่วนชั้นที่ 2 เป็นชั้นที่ถูกสร้างเพิ่มเข้าไปจากระบบเดิม

การดัดแปลงจำนวนหนึ่งถูกปฏิบัติที่ไฟล์หลัก (main file) ของไฮเปอร์ไวเซอร์, vl.c ลำดับที่หนึ่งทำการดัดแปลง main() function เพื่อรับตำแหน่งและชื่อของเช็คพอยต์ ลำดับต่อมาดัดแปลง main_loop_wait() function ซึ่งมีหน้าที่รับผิดชอบสำหรับการส่งผ่านข้อมูลนำเข้าจากไฮเปอร์ไวเซอร์ไปสู่เครื่องเสมือนเพื่อแทรกส่วนติดต่อกับระบบวิชีซีพีเข้าไป เพื่อให้สามารถตรวจจับและส่งผ่านข้อมูลจากระบบคิวไปสู่เครื่องเสมือนก่อนการรับข้อมูลต่าง ๆ จากเครือข่ายผู้วิจัยได้ดัดแปลง tap_send() function เพื่อตรวจจับเฟรมพิเศษที่ถูกใช้เพื่อควบคุมการสร้างเช็คพอยต์/รีโคเวอริและเพื่อจัดเก็บเฟรมนำเข้าสู่ระบบคิวในระหว่างที่เครื่องเสมือนยังไม่ริเริ่มการทำงาน

ไฟล์ของคีมูอีกอันหนึ่งที่ถูกดัดแปลงคือ monitor.c โดยการเพิ่มส่วนต่อประสานกับผู้ใช้แบบบรรทัดคำสั่ง (command line user interface) ชุดหนึ่งให้กับไฮเปอร์ไวเซอร์เพื่อให้ผู้ใช้สามารถสั่งคัลด์เตอร์เสมือนให้ปฏิบัติการทำโคออดิเนตเช็คพอยต์ผ่านส่วนควบคุมของคีมู (QEMU console) ในส่วนของ monitor.c เองก็มีการเรียกใช้คำสั่งสำหรับสร้างเช็คพอยต์และรีโคเวอรี่ซึ่งถูกสร้างแยกไว้ในไฟล์อื่นคือ vccp_read_config.c และ tap-queue.c

สุดท้ายผู้วิจัยสร้างฟังก์ชันสร้างเช็คพอยต์และรีโคเวอรี่ในไฟล์ vccp_read_config.c และสร้างระบบคิวใน tap-queue.c ตามลำดับ ส่วนการส่งเฟรมพิเศษเพื่อควบคุมการทำงานในระบบเช่น เช็คพอยต์รีเคส รีซูมวีเอ็ม และโหลดเฟรมจะทำผ่าน qemu_send_packet() function

3.4 ความถูกต้องและข้อจำกัด

ในหัวข้อนี้จะอธิบายถึงความถูกต้องและข้อจำกัดของโพรโทคอลการเช็คพอยต์และรีโคเวอรี่ในระดับไฮเปอร์ไวเซอร์ที่ถูกเสนอในงานวิจัยนี้ เพื่อเป็นการยืนยันว่ากระบวนการใด ๆ ของโพรโทคอลจะไม่ไปขัดขวางตรรกะการคำนวณ (computation logic) ของเกสโอเอสไปจนถึงโปรแกรมประยุกต์ใด ๆ บนคัลด์เตอร์เสมือน หรืออีกนัยหนึ่งเพื่อยืนยันว่าคัลด์เตอร์เสมือนยังคงผลิตผลลัพธ์ที่ถูกต้องเหมือนกับการทำงานในสภาพแวดล้อมที่ไม่มีเช็คพอยต์รีโคเวอรี่โพรโทคอล โดยการพิสูจน์จะกล่าวถึงเรื่องของการติดต่อสื่อสารและคล็อกสกินที่เกิดจากโพรโทคอล

3.4.1 การติดต่อสื่อสารแบบไฟโฟและเชื่อถือได้

ในประเด็นเรื่องการติดต่อสื่อสารผู้วิจัยจะแสดงให้เห็นว่าการทำเช็คพอยต์รีโคเวอรี่ไม่ได้ทำลายคุณลักษณะเข้าก่อนออกก่อนและเชื่อถือได้ของเครือข่ายวีดีโอ

ทฤษฎีบท 1: โพรโทคอลที่เสนอไม่ได้เป็นสาเหตุของเฟรมสูญหายและทำลายคุณสมบัติเข้าก่อนออกก่อน

พิสูจน์: กรณีเฟรมสูญหายสามารถจำแนกออกเป็นกรณีต่าง ๆ ได้ดังนี้

1. โพรโทคอลสามารถเป็นต้นเหตุการสูญเสยเฟรมที่ถูกส่งออกมาจากเครื่องเสมือนก่อนการเช็คพอยต์และรีโคเวอรี่เสตทเครื่องเสมือนและเฟรมข้อมูล แต่อย่างไรก็ตามเหตุการณ์นี้ไม่เป็นจริงเพราะก่อนการบันทึกเสตทและเฟรมข้อมูล (1) ขั้นตอนวิธีการเช็คพอยต์มีการบันทึก

ทุก ๆ เฟรมข้อมูลที่กำลังเดินทางอยู่ลงในรีซีฟคิวและ (2) ด้วยคุณลักษณะการสื่อสารแบบเข้าก่อนออกก่อนและเชื่อถือได้ทำให้เฟรมถูกบันทึกลงในระบบคิวตามลำดับอย่างถูกต้องก่อนที่จะได้รับไพนอลเฟรม ในทำนองเดียวกันในการรีโคเวอรี่นั้นเฟรมข้อมูลต่าง ๆ ที่ถูกระงับจากเครือข่ายจะถูกอ่านกลับมาไว้ในคิวตามลำดับที่ถูกส่งมาเพื่อป้อนให้กับเครื่องเสมือนภายหลังจากที่ชุมการทำงาน ดังนั้นจึงสรุปได้ว่าขั้นตอนวิธีนี้ไม่ทำให้เกิดเฟรมสูญหายก่อนการเช็คพอยต์และรีโคเวอรี่เสตทเครื่องเสมือนและเฟรมข้อมูล

2. โพรโทคอลสามารถเป็นสาเหตุของการสูญเสียเฟรมในระหว่างการชะล้างช่องสื่อสารเพื่อสร้างคอนซิสเทนต์โกลบอลเสตท เพราะว่าก่อนที่ไพนอลเฟรมที่มีหน้าที่ชะล้างช่องสื่อสารจะถูกส่งออกไปเครื่องเสมือนทั้งหมดจะถูกหยุดการทำงานเพื่อปิดกั้นไม่ให้ส่งเฟรมข้อมูลออกมาอีก และด้วยคุณสมบัติเข้าก่อนออกก่อนและเชื่อถือได้ของเครือข่ายจึงรับประกันว่าไพนอลเฟรมจะถูกรับเป็นเฟรมสุดท้ายภายหลังจากเฟรมที่ค้างอยู่ในเครือข่ายถูกบันทึกลงบนรีซีฟคิว ดังนั้นในขั้นตอนการชะล้างเฟรมจึงไม่ทำให้เกิดเฟรมสูญหาย

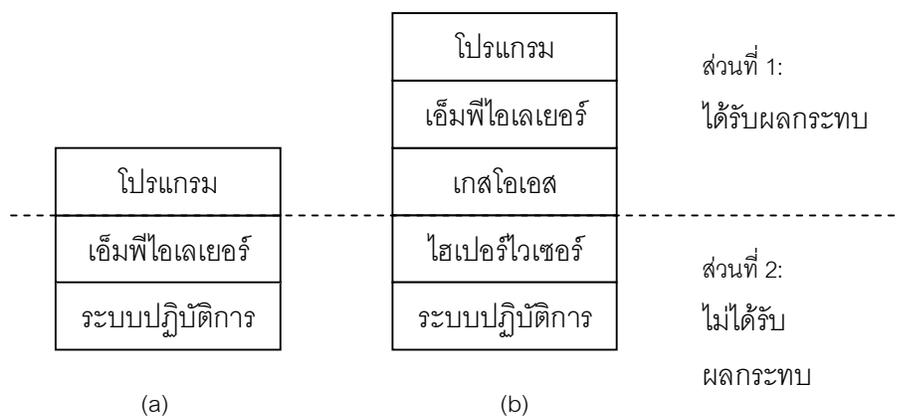
3. โพรโทคอลเป็นสาเหตุที่ทำให้เฟรมข้อมูลที่ถูกส่งออกมาภายหลังจากการบันทึกและรีโคเวอรี่เสตทเครื่องเสมือนสูญหาย ภายหลังจากเหตุการณ์ทั้งสองเครื่องเสมือนทั้งหมดควรริเริ่มการทำงานและเริ่มต้นส่งเฟรมข้อมูลไปสู่เครื่องเสมือนอื่น แต่เนื่องจากรีเริ่มเฟรมอาจส่งไปถึงเครื่องเสมือนคนละเวลาดังนั้นจึงสามารถเกิดเหตุการณ์ซึ่งเครื่องเสมือนส่งเฟรมไปหาเครื่องเสมือนที่ยังไม่ริเริ่มการทำงานซึ่งอาจทำให้เฟรมสูญหาย แต่อย่างไรก็ตามสถานการณ์นี้ไม่สามารถเกิดขึ้นได้เพราะขั้นตอนวิธีนี้บังคับให้เครื่องเสมือนใส่เฟรมข้อมูลลงในรีซีฟคิวเสมอในระหว่างที่รอคอยรีเริ่มเฟรม ดังนั้นท้ายที่สุดแล้วเครื่องเสมือนจะได้รับเฟรมดังกล่าวจากคิวภายหลังการริเริ่ม

4. กรณีเครือข่ายแบบเข้าก่อนออกก่อนขั้นตอนวิธีที่เสนอนี้ให้การรับประกันว่าเฟรมข้อมูลจะถูกส่งออกไปตามลำดับอย่างถูกต้องทั้งก่อนและหลังการสร้างเช็คพอยต์และรีโคเวอรี่ เพราะว่า (1) ขั้นตอนวิธีนี้จะใส่เฟรมข้อมูลไว้ที่ท้ายคิวเสมอในกรณีที่เครื่องเสมือนไม่พร้อมทำงาน (2) เมื่อเครื่องเสมือนพร้อมทำงานเฟรมจะถูกอ่านขึ้นมาจากตำแหน่งหัวคิว (3) ถ้ามีเฟรมข้อมูลเข้ามาในขณะที่คิวมีข้อมูลค้างอยู่เฟรมข้อมูลนั้นจะถูกใส่ลงท้ายคิวเพื่อส่งวนลำดับ

ภาพที่ 3.4

แผนภาพผลกระทบของนาฬิกาในซอฟต์แวร์ชั้นต่าง ๆ เมื่อมีการหยุดการประมวลผล

(a) ชั้นซอฟต์แวร์บนระบบปกติ (b) ชั้นซอฟต์แวร์บนเครื่องเสมือน



3.4.2 คล็อกสกีวและไทม์เอาต์

เมื่อพิจารณาโปรโตคอลของงานวิจัยนี้ทำให้พบว่าการหยุดการประมวลผลของเครื่องเสมือนมีโอกาสทำให้เกิดคล็อกสกีวได้ เนื่องจากเครื่องเสมือนแต่ละเครื่องมีนาฬิกาเป็นของตัวเองซึ่งหยุดเดินขณะเซ็คพอยต์และกลับมาเดินอีกครั้งเมื่อรีซุม เมื่อพิจารณาจากภาพที่ 3.4 ผู้วิจัยพบว่าประเด็นนี้เป็นสิ่งใหม่ที่ไม่เกิดในโปรแกรมประยุกต์บนระบบปกติ เพราะว่าบนระบบปกติการหยุดการประมวลผลจะส่งผลในระดับของโปรแกรมประยุกต์เท่านั้น ไม่ส่งผลกระทบต่อตัวระบบปฏิบัติการแบบในเครื่องเสมือน

ทฤษฎีบท 2: เซ็คพอยต์/รีโคเวอรี่โปรโตคอลสามารถเป็นสาเหตุให้เกิดคล็อกสกีวท่ามกลางเครื่องเสมือนต่าง ๆ

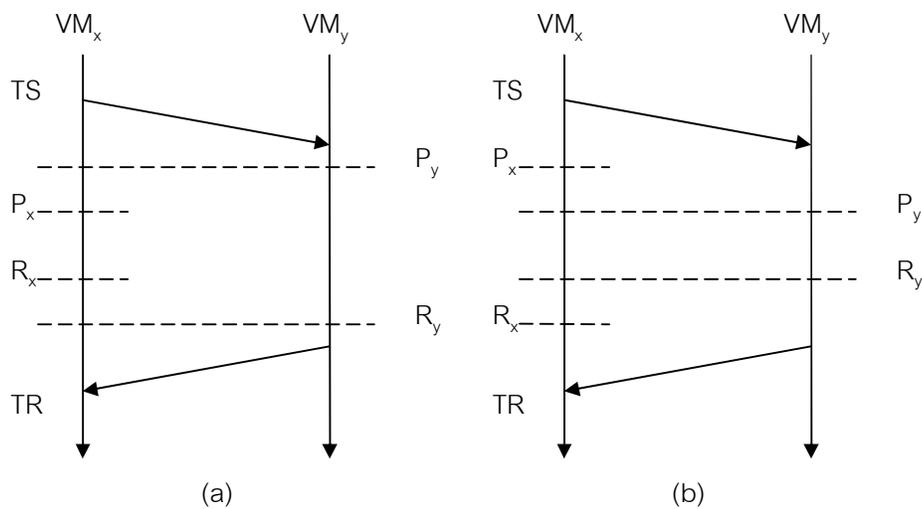
การพิสูจน์: จากโปรโตคอลพบว่าเซ็คพอยต์เฟรมของการเซ็คพอยต์และรีซุมเฟรมของทั้งสองวิธีสามารถเดินทางไปถึงเครื่องเสมือนปลายทางได้คนละเวลา ซึ่งขึ้นอยู่กับสถานะของเครือข่ายในขณะนั้น ดังนั้นจึงเป็นสาเหตุให้เครื่องเสมือนมีการหยุดและรีซุมคนละเวลาเป็นเหตุให้เกิดคล็อกสกีวระหว่างโหนดต่าง ๆ ของคลัสเตอร์เสมือน

ทฤษฎีบท 3: เมื่อพิจารณาจากฝั่งของผู้ส่งข้อมูลพบว่าเซิร์ฟเวอร์/รีโคโนเวอร์รีโคโนคอลสามารถมีผลกระทบต่อเวลาเดินทางครบรอบ (RTT: Round Trip Time) ของผู้ส่งและอาจเป็นผลให้เกิดการหมดเวลาการติดต่อสื่อสาร

การพิสูจน์: สมมติให้มีเครื่องเสมือนสองเครื่องคือ VM_x และ VM_y กำลังติดต่อสื่อสารกันภายในคลัสเตอร์เสมือน ในตอนเริ่มต้นนาฬิกาของทั้งสองเครื่องตรงกันและมีอัตราการเดินของเวลาเท่ากัน (progression time)

ภาพที่ 3.5

ผลกระทบของโคโนคอลที่มีต่อเวลาครบรอบ



เมื่อ VM_x ส่งรีควีสไปให้ VM_y จะต้องมีการรอคอยการตอบกลับเป็นระยะเวลาหนึ่ง ซึ่งในระหว่างนี้หากมีการสร้างเซิร์ฟเวอร์จะก่อให้เกิดคล็อกสกรูว์ขึ้น (ทฤษฎีบท 2) กำหนดให้ $T\Delta$ คือ ผลต่างเวลาของเครื่องเสมือนทั้งสอง ผู้วิจัยจะแสดงให้เห็นว่ามีสถานการณ์ที่ $T\Delta$ มีผลกระทบต่อ RTT ที่รายงานออกมาจาก VM_x และอาจเป็นสาเหตุที่ทำให้เกิดไทม์เอาท์ของการติดต่อสื่อสาร (communication timeout) ผู้วิจัยยังแสดงให้เห็นอีกว่า $T\Delta$, RTT, และจำนวนเวลาที่แท้จริง (RTT_{real}) ที่วิเอ็มทั้งสองใช้ไปกับการส่งรีควีสและตอบกลับมีความเกี่ยวข้องกัน ข้อสังเกต RTT_{real} สามารถถูกพิจารณาว่าเป็น RTT เมื่อวิเอ็มทั้งสองทำงานตามปกติโดยไม่มีการสร้างเซิร์ฟ

พอยต์หรือรีโคเวอรี่ ภาพที่ 3.5 แสดงสถานการณ์ที่การเช็คพอยต์และรีโคเวอรี่มีผลกระทบต่อ RTT กำหนดให้ TS เป็นเวลาที่รีเคสถูกส่งออกจาก VM_x, TR เป็นเวลาที่เรสปอนถูกรับที่ VM_x, P_x เป็นเวลาที่ VM_x ถูกหยุด, P_y เป็นเวลาที่ VM_y ถูกหยุด, R_x เป็นเวลาที่ VM_x ถูกริซุม, และ R_y เป็นเวลาที่ VM_y ถูกริซุม จาก (a), พวกเราสามารถสรุปได้ดังนี้

$$RTT = RTT_{real} + T\Delta \quad (1)$$

$$RTT_{real} = (TR - R_y) + (P_y - TS) \quad (2)$$

$$T\Delta = (R_y - R_x) + (P_x - P_y) \quad (3)$$

ดังนั้นถ้า $T\Delta$ เป็นค่าบวกแสดงว่า RTT ของเครื่องผู้ส่งควรมีขนาดมากกว่า RTT_{real} ของการสื่อสารที่เกิดขึ้น หรืออีกนัยหนึ่งคือ ถ้า $T\Delta$ เป็นค่าลบจะทำให้ RTT มีค่าน้อยกว่า RTT_{real} ซึ่ง RTT มีค่าเป็นบวกในรูป (a) และค่าเป็นลบในภาพที่ (b) ตามลำดับ รวมทั้งมันควรมีค่าเป็น 0 เมื่อ $(R_y - R_x) + (P_x - P_y)$ เป็น 0 นั่นเป็นผลให้การหยุดและริซุมการทำงานไม่ก่อให้เกิดคล็อกสกริวระหว่างเครื่องเสมือนสองเครื่อง

3.4.3 ข้อจำกัดและวิธีการแก้ไข

จากหัวข้อที่ผ่านมาแสดงให้เห็นว่าโปรโตคอลที่ใช้บนวีซีซีพีในวิทยานิพนธ์ฉบับนี้มีข้อจำกัดบางประการในด้านการนำไปใช้ดังนี้

1. จากทฤษฎีบท 2 แสดงให้เห็นว่าวีซีซีพีอาจไม่เหมาะสมสำหรับโปรแกรมประยุกต์ที่ตรรกะการคำนวณสามารถ (computational logic) เปลี่ยนแปลงได้เมื่อเกิดคล็อกสกริว
2. จากทฤษฎีบท 3 แสดงให้เห็นว่าวีซีซีพีอาจไม่เหมาะสมสำหรับโปรแกรมประยุกต์ที่ต้องการตอบสนองอย่างรวดเร็ว เพราะว่าโปรแกรมชนิดนี้มักมีค่าการหมดเวลาการสื่อสารต่ำจนไม่สามารถทนทานต่อเวลาเดินทางครบรอบที่เพิ่มขึ้นได้

มีหลายเทคนิคที่เราสามารถใช้เพื่อหลีกเลี่ยงข้อจำกัดที่หนึ่ง ตัวอย่างเช่น ใช้นาฬิกาตรรกะ (logical clock) ของ Lamport สำหรับการคำนวณแทนการใช้เวลาของนาฬิกาของเครื่องเสมือนซึ่งเป็นการแก้ไขในระดับโปรแกรมประยุกต์ อีกวิธีหนึ่งคือ ใช้โปรโตคอลเวลาบนเครือข่าย (NTP: Network Time Protocol) เพื่อประสานเวลาของเครื่องเสมือนทั้งหมดให้ตรงกันในระดับ

ระบบปฏิบัติการ หรือการใช้เทคนิคประสานเวลาบางอย่างก่อนการรีซุมเครื่องเสมือนภายหลังการ
สร้างเซ็คพอยต์หรือรีโคเวอรี่ในระดับของไฮเปอร์ไวเซอร์

ในการหลีกเลี่ยงการเกิดโทรมเอนท์ของการติดต่อสื่อสารอาจจำเป็นต้องปรับค่าการหมด
เวลาสื่อสารของโปรแกรมประยุกต์และระบบปฏิบัติการ ให้มีค่าสูงกว่าค่าที่ถูกใช้ในสภาพแวดล้อม
ที่ไม่มีการเซ็คพอยต์อย่างน้อยที่สุด $T\Delta$