บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

1. อภิปรายและสรุปผลการวิจัย

การประเมินประสิทธิภาพของการเช็คพอยน์เวอร์ชวลแมชชีนระหว่างวิธีการทำเช็คพอยน์แบบหยุดการทำงาน แบบไลฟ์เช็คพอยน์ และแบบไลฟ์เช็คพอยน์ที่ใช้เทรด สามารถแยกอภิปรายผลการทดลองเป็น 2 ประเด็นคือ ประเด็นเกี่ยวกับเวลาในการทำงานของแอพพลิเคชันและเช็คพอยน์โอเวอร์เฮด และประเด็นเกี่ยวกับเวลาที่ใช้ในการเช็คพอยน์และขนาดของเช็คพอยน์ไฟล์

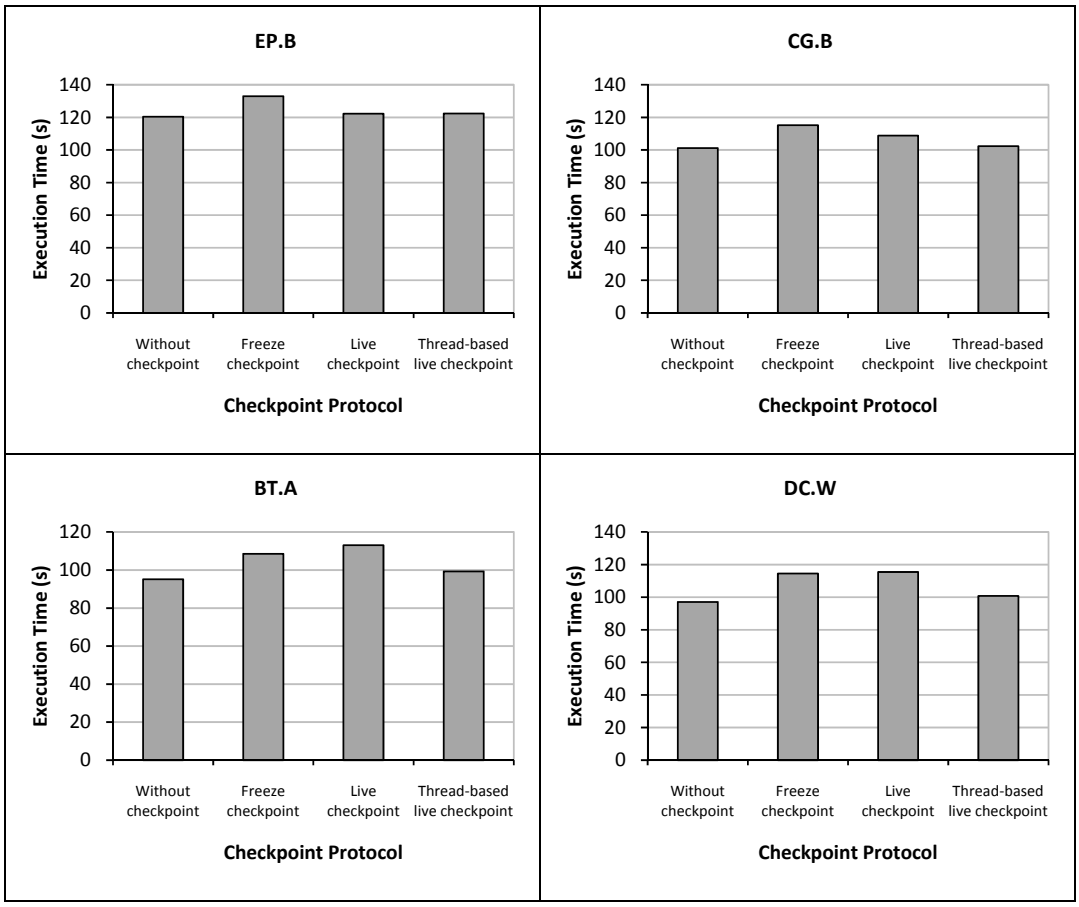
เวลาในการทำงานของแอพพลิเคชัน (execution time) คือเวลาการทำงานของเบนช์มาร์กที่มีการเช็คพอยน์เวอร์ชวลแมชชีนด้วยวิธีการแบบหยุดการทำงาน (freeze checkpoint) วิธีไลฟ์เช็คพอยน์ปกติ (live checkpoint) และวิธีการที่ใช้เทรดในไลฟ์เช็คพอยน์ (thread-based live checkpoint) ส่วนเช็คพอยน์โอเวอร์เฮด (checkpoint overhead) คือเวลาการทำงานของแอพพลิเคชันที่เพิ่มขึ้นเนื่องจากการเช็คพอยน์ โดยเปรียบเทียบกับเวลาการทำงานของแอพพลิเคชันที่ไม่มีการเช็คพอยน์ เวลาที่ใช้ในการเช็คพอยน์ (checkpoint latency) คือเวลาที่ใช้ในการบันทึกสถานะของเวอร์ชวลแมชชีนลงในเช็คพอยน์ไฟล์ ซึ่งขนาดของเช็คพอยน์ไฟล์จะขึ้นอยู่กับปริมาณข้อมูลที่จะต้องคัดลอก

1.1 เวลาในการทำงานของแอพพลิเคชันและเช็คพอยน์โอเวอร์เฮด

จากภาพที่ 5.1 พบว่าเวลาการทำงานของแอพพลิเคชันที่มีการเช็คพอยน์เวอร์ชวลแมชชีนด้วยวิธีการที่ใช้เทรดในไลฟ์เช็คพอยน์ใช้เวลาน้อยกว่าการเช็คพอยน์ด้วยวิธีการเดิมหรือการเช็คพอยน์แบบหยุดการทำงาน และการเช็คพอยน์ด้วยวิธีไลฟ์เช็คพอยน์ปกติ อย่างมีนัยสำคัญ ($p<.05$) ในทุกเบนช์มาร์กที่ใช้ในการทดลองยกเว้น EP คลาส B ซึ่งแอพพลิเคชันใช้เวลาทำงานใกล้เคียงกันระหว่างการเช็คพอยน์ด้วยวิธีไลฟ์เช็คพอยน์ที่ใช้เทรดและวิธีไลฟ์เช็คพอยน์ปกติ เนื่องจาก EP เป็นเบนช์มาร์กที่ต้องการใช้หน่วยความจำและเขียนข้อมูลลงหน่วยความจำน้อยกว่าเบนช์มาร์กอื่น ทำให้ขนาดข้อมูลในหน่วยความจำที่จะต้องคัดลอกลงสู่เช็คพอยน์ไฟล์มีจำนวนน้อย (ประมาณ 170MB จากตารางที่ 3.4) การเช็คพอยน์จึงเสร็จสิ้นได้อย่างรวดเร็วโดยมีโอเวอร์เฮดเพียงเล็กน้อย (1.96s ในไลฟ์เช็คพอยน์ที่ใช้เทรด และ 1.86s ในไลฟ์เช็คพอยน์ปกติ ดังภาพที่ 5.2)

ภาพที่ 5.1

เวลาในการทำงานของเบนช์มาร์กที่มีการเช็คพอยน์และไม่มีการเช็คพอยน์



การเช็คพอยน์ด้วยวิธีการที่ใช้เทรดในไลฟ์เช็คพอยน์ทำให้แอพพลิเคชันสามารถทำงานเสร็จได้เร็วกว่าการเช็คพอยน์ด้วยวิธีการอื่น เพราะเทรดจะช่วยทำงานเช็คพอยน์ส่วนใหญ่แทนเวอร์ชวลแมชชีน ทำให้แอพพลิเคชันสามารถทำงานได้มากขึ้นในขณะที่มีการเช็คพอยน์

ในกรณีของแอพพลิเคชันที่ต้องการใช้หน่วยความจำมาก (CG) การเช็คพอยน์แบบไลฟ์เช็คพอยน์จะทำให้แอพพลิเคชันทำงานเสร็จได้เร็วกว่าการเช็คพอยน์ด้วยวิธีหยุดการทำงาน เพราะแอพพลิเคชันสามารถทำงานได้ในขณะที่มีการคัดลอกหน่วยความจำ และการใช้เทรดในวิธีการของไลฟ์เช็คพอยน์จะทำให้แอพพลิเคชันทำงานเสร็จได้เร็วขึ้น เพราะเทรดช่วยทำการคัดลอกหน่วยความจำแทนเวอร์ชวลแมชชีน ดังนั้นในช่วงเวลาที่มีการคัดลอกหน่วยความจำนี้

แอพพลิเคชันจึงสามารถทำงานได้มากกว่าวิธีการของไลฟ์เช็คพอยน์ปกติ ซึ่งเวอร์ชวลแมชชีนจะต้องคัดลอกหน่วยความจำสลับกับยอมให้แอพพลิเคชันทำงาน

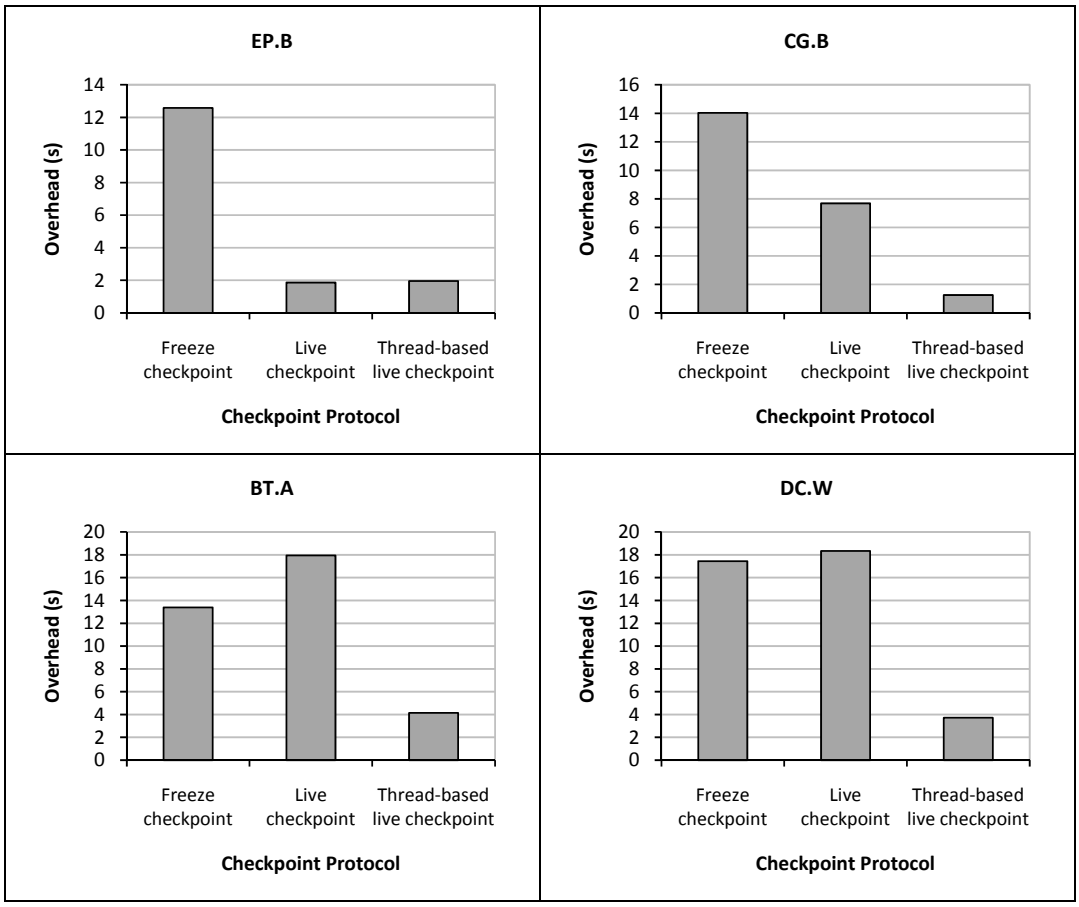
ปัญหาของการเช็คพอยน์ด้วยวิธีการของไลฟ์เช็คพอยน์ปกติเห็นได้ชัดในกรณีที่แอพพลิเคชันมีการเขียนข้อมูลลงหน่วยความจำมาก ซึ่งจะทำให้ข้อมูลที่จะต้องคัดลอกเพิ่มขึ้นตามไปด้วย และจะส่งผลให้เวลาการทำงานของแอพพลิเคชันเพิ่มขึ้นเรื่อยๆเนื่องจากแอพพลิเคชันถูกขัดจังหวะมากเกินไป ดังที่พบในการทำงานของเบนช์มาร์ก BT

ส่วนในกรณีของแอพพลิเคชันที่เขียนไฟล์ขนาดใหญ่ (DC) เวลาการทำงานของแอพพลิเคชันที่มีการเช็คพอยน์ด้วยวิธีการที่ใช้เทรดในไลฟ์เช็คพอยน์ก็น้อยกว่าการเช็คพอยน์ด้วยวิธีการอื่น ถึงแม้ว่าเวลาที่แอพพลิเคชันต้องหยุดทำงานอาจเพิ่มขึ้นเนื่องจากเวลาในการบันทึกสถานะของดิสก์เพิ่มขึ้น แต่เวลาทั้งหมดที่แอพพลิเคชันถูกขัดจังหวะก็ยังคงน้อยกว่าวิธีการอื่นเพราะการใช้เทรดทำให้การคัดลอกข้อมูลจากหน่วยความจำเสร็จสิ้นอย่างรวดเร็ว ส่วนการเช็คพอยน์ด้วยวิธีการของไลฟ์เช็คพอยน์ปกติ แอพพลิเคชันถูกขัดจังหวะค่อนข้างมาก เนื่องจากมีการเขียนหน่วยความจำมากตามการเขียนไฟล์ ทำให้ใช้เวลาในการทำงานใกล้เคียงกับการเช็คพอยน์ด้วยวิธีหยุดการทำงาน

เวลาการทำงานของแอพพลิเคชันจะสอดคล้องกับโอเวอร์เฮดของการเช็คพอยน์ โดยแอพพลิเคชันต้องใช้เวลาในการทำงานเพิ่มขึ้นเนื่องจากมีการเช็คพอยน์เวอร์ชวลแมชชีน ดังนั้นเช็คพอยน์โอเวอร์เฮดจึงเป็นเวลาการทำงานของเวอร์ชวลแมชชีนที่เกี่ยวข้องกับการเช็คพอยน์ ซึ่งโดยทั่วไปคือการเซ็ตอัพสถานะไมเกรชัน การคัดลอกหน่วยความจำ และการบันทึกสถานะเวอร์ชวลแมชชีนและสถานะของดิสก์อิมเมจ สำหรับวิธีการที่ใช้เทรดในไลฟ์เช็คพอยน์ การคัดลอกหน่วยความจำซึ่งเป็นขั้นตอนที่ใช้เวลามากที่สุด (มากกว่า 90% ของเวลาเช็คพอยน์ในเบนช์มาร์ก EP, CG, และ BT และมากกว่า 70% ในเบนช์มาร์ก DC) จะไม่ถูกรวมอยู่ในเช็คพอยน์โอเวอร์เฮดเนื่องจากเป็นการทำงานของเทรด แต่จะมีการทำงานอื่น คือการตรวจสอบสถานะการทำงานของเทรดเป็นระยะๆ เริ่มต้นเมื่อเทรดทำการคัดลอกหน่วยความจำ และสิ้นสุดเมื่อเทรดจบการทำงาน อย่างไรก็ตาม เวลาที่เวอร์ชวลแมชชีนใช้ในการตรวจสอบสถานะการทำงานของเทรดนี้เป็นโอเวอร์เฮดที่น้อยมาก (5μs-13μs) นอกจากนี้ โอเวอร์เฮดที่เฉพาะเจาะจงสำหรับวิธีการที่ใช้เทรดอีกอย่างหนึ่งคือโอเวอร์เฮดที่เกี่ยวข้องกับการเซ็ตอัพเทรดและการทำซิงโครไนเซชันเมื่อเวอร์ชวลแมชชีนต้องการเขียนข้อมูลลงในหน่วยความจำในช่วงเวลาที่มีการเช็คพอยน์ ซึ่งโอเวอร์เฮดทั้งหมดที่เกี่ยวข้องกับการทำงานของเทรดเป็นโอเวอร์เฮดที่น้อยเช่นเดียวกัน (1.76s)

ภาพที่ 5.2

โอเวอร์เฮดของแต่ละวิธีการเช็คพอยน์



ภาพที่ 5.2 แสดงเช็คพอยน์โอเวอร์เฮดของวิธีการเช็คพอยน์แต่ละแบบ พบว่าการเช็คพอยน์ด้วยวิธีการที่ใช้เทรดในไลฟ์เช็คพอยน์มีโอเวอร์เฮดเพียงเล็กน้อยเมื่อเทียบกับวิธีการเช็คพอยน์แบบอื่น เพราะเวอร์ชวลแมชชีนไม่ต้องทำการคัดลอกหน่วยความจำสำหรับการเช็คพอยน์ ทำให้แอพพลิเคชันสามารถทำงานต่อไปได้โดยไม่ถูกขัดจังหวะเป็นเวลานาน และเวลาทั้งหมดที่ต้องหยุดเวอร์ชวลแมชชีนมีเพียงเล็กน้อยเท่านั้น (ดาวน์ไทม์ที่มากที่สุดคือ 1.27s ในเบนช์มาร์ก DC)

การคัดลอกหน่วยความจำเป็นขั้นตอนที่ใช้เวลามากที่สุดในการทำเช็คพอยน์ ซึ่งระยะเวลาที่ใช้จะขึ้นอยู่กับขนาดของหน่วยความจำและการเขียนข้อมูลลงหน่วยความจำของแต่ละแอพพลิเคชัน จากภาพที่ 5.2 โอเวอร์เฮดของการเช็คพอยน์ด้วยวิธีไลฟ์เช็คพอยน์ปกติและวิธีไลฟ์

เช็คพอยน์ที่ใช้เทรดของเบนช์มาร์ก EP มีขนาดใกล้เคียงกัน เพราะ EP มีการใช้หน่วยความจำน้อย และถึงแม้ว่าวิธีการไลฟ์เช็คพอยน์จะมีการขัดจังหวะเวอร์ชวลแมชชีนมากกว่าแบบเทรด แต่การเช็คพอยน์แบบเทรดต้องใช้เวลาในการเซ็ตอัพการใช้เทรดในการทำไมเกรชัน ได้แก่ การสร้างเทรด และการจัดเตรียมโครงสร้างข้อมูลเพื่อทำให้การเข้าถึงข้อมูลร่วมกันของเทรดและเวอร์ชวลแมชชีนเป็นไปอย่างถูกต้อง

เมื่อแอพพลิเคชันใช้หน่วยความจำมากขึ้น โอเวอร์เฮดของวิธีไลฟ์เช็คพอยน์ปกติก็จะเพิ่มขึ้นดังที่แสดงให้เห็นในเบนช์มาร์ก CG, BT และ DC โดยที่โอเวอร์เฮดในเบนช์มาร์ก BT เพิ่มขึ้นอย่างเห็นได้ชัดเนื่องจาก BT เป็นแอพพลิเคชันที่มีการเขียนหน่วยความจำมาก ทำให้เวอร์ชวลแมชชีนต้องคัดลอกข้อมูลลงสู่เช็คพอยน์ไฟล์ต่อไปเรื่อยๆ เป็นผลให้แอพพลิเคชันใช้เวลาทำงานเพิ่มมากขึ้น ซึ่งแตกต่างจากวิธีการที่ใช้เทรดในไลฟ์เช็คพอยน์ที่มีโอเวอร์เฮดค่อนข้างคงที่ ถึงแม้ว่าแอพพลิเคชันจะใช้หน่วยความจำเพิ่มขึ้นหรือมีการเขียนหน่วยความจำมากก็ตาม เนื่องจากเวอร์ชวลแมชชีนสามารถทำงานได้โดยเกือบจะไม่ถูกหยุดในขณะที่เทรดคัดลอกหน่วยความจำ ดังนั้น โดยทั่วไปแล้ว การคัดลอกหน่วยความจำจึงมีผลกระทบต่อโอเวอร์เฮดน้อยมากในวิธีการที่ใช้เทรด แต่อย่างไรก็ตาม ค่าโอเวอร์เฮดอาจเพิ่มขึ้นเล็กน้อยถ้าจำนวนรอบของการคัดลอกหน่วยความจำเท่ากับ 200 รอบ เพราะเวลาที่ใช้ในการคัดลอกหน่วยความจำที่เหลือในรอบสุดท้ายอาจเพิ่มขึ้น

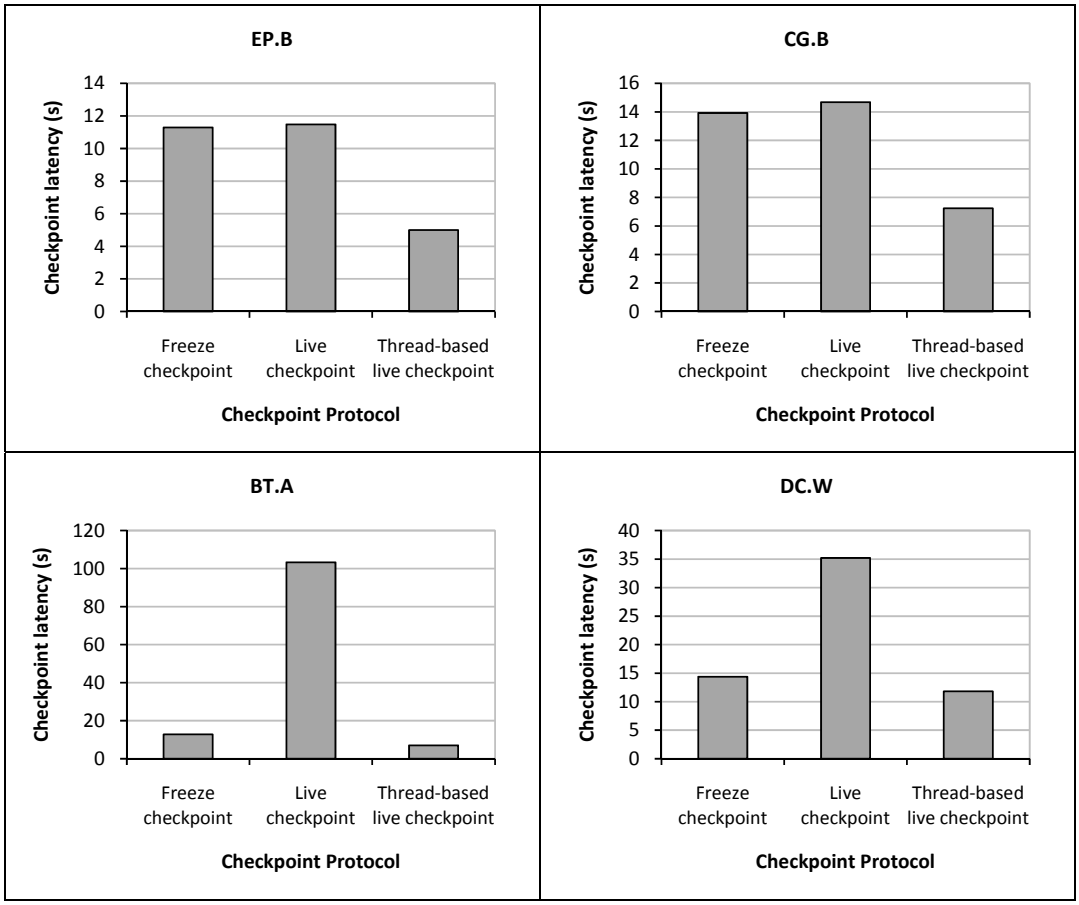
โอเวอร์เฮดของการเช็คพอยน์ด้วยวิธีการที่ใช้เทรดคือเวลาที่เวอร์ชวลแมชชีนจะต้องหยุดเพื่อตรวจสอบสถานะของเทรดเป็นระยะๆรวมกับเวลาในการเซ็ตค่าเริ่มต้นสถานะไมเกรชันและเวลาบันทึกสถานะในขั้นตอนสุดท้าย ซึ่งมีค่าโอเวอร์เฮดที่เพิ่มขึ้นในการทำงานของโปรแกรมเบนช์มาร์กน้อยกว่าวิธีการอื่นมาก (1.25s-4.14s) ในขณะที่โอเวอร์เฮดของวิธีการไลฟ์เช็คพอยน์ปกติคือเวลาที่ใช้ในการเซ็ตค่าเริ่มต้นสถานะไมเกรชัน เวลาที่ใช้ในการคัดลอกหน่วยความจำและเวลาบันทึกสถานะในขั้นตอนสุดท้าย ซึ่งทำให้โอเวอร์เฮดเพิ่มขึ้นได้ถ้าขนาดของหน่วยความจำที่จะต้องคัดลอกเพิ่มขึ้น (1.86s-18.34s) ส่วนโอเวอร์เฮดของวิธีการแบบหยุดการทำงานคือระยะเวลาที่เวอร์ชวลแมชชีนจะต้องหยุดเพื่อทำเช็คพอยน์ (12.58s-17.44s) ซึ่งมีค่าค่อนข้างคงที่ในการเช็คพอยน์แต่ละครั้ง เพราะในขณะที่ทำเช็คพอยน์ ขนาดของหน่วยความจำที่จะต้องคัดลอกไม่ได้ถูกเปลี่ยนแปลง เวลาที่ใช้ในการคัดลอกหน่วยความจำจึงขึ้นอยู่กับขนาดของหน่วยความจำที่แต่ละแอพพลิเคชันต้องการใช้

1.2 เวลาที่ใช้ในการเช็คพอยน์และขนาดของเช็คพอยน์ไฟล์

เวลาที่ใช้ในการเช็คพอยน์ส่วนใหญ่คือเวลาที่ใช้ในการคัดลอกหน่วยความจำ ดังนั้นความต้องการใช้หน่วยความจำและการเขียนหน่วยความจำของแอพพลิเคชันจึงมีผลกระทบอย่างมากต่อเวลาทั้งหมดที่ใช้ในการเช็คพอยน์ สำหรับวิธีการที่ใช้เทรดในไลฟ์เช็คพอยน์ การหยุดคัดลอกหน่วยความจำของเทรดสามารถเกิดขึ้นได้ใน 2 กรณีคือ 1) เมื่อขนาดหน่วยความจำที่เหลือมีน้อยเพียงพอ (เงื่อนไขเดิม) และ 2) เมื่อคัดลอกหน่วยความจำถึง 200 รอบ

ภาพที่ 5.3

เวลาในการเช็คพอยน์เวอร์ชวลแมชชีน



จากภาพที่ 5.3 การเช็คพอยน์ด้วยวิธีการที่ใช้เทรดในไลฟ์เช็คพอยน์ใช้เวลาในการเช็คพอยน์น้อยกว่าการเช็คพอยน์ด้วยวิธีการอื่นในทุกเบนช์มาร์กที่ใช้ในการทดลอง โดยใช้เวลา

เพียง 5.00s-11.81s ในขณะที่วิธีการแบบหยุดการทำงานใช้เวลา 11.30s-14.39s และวิธีการไลฟ์เช็คพอยน์ปกติใช้เวลา 11.48s-103.33s การเช็คพอยน์ด้วยวิธีการที่ใช้เทรดในไลฟ์เช็คพอยน์ใช้เวลาน้อยเนื่องจากเทรดสามารถทำการคัดลอกหน่วยความจำได้ต่อเนื่องอย่างรวดเร็วโดยที่ไม่มีการตั้งเวลา (100ms) ถึงแม้ว่าแอพพลิเคชันจะมีการเขียนหน่วยความจำในปริมาณมาก (อย่างเช่นเบนช์มาร์ก BT) การคัดลอกหน่วยความจำของเทรดก็ยังคงสามารถสิ้นสุดได้อย่างรวดเร็ว เนื่องจากแต่ละรอบของการคัดลอกหน่วยความจำจะมีจำนวนเพจที่ถูกเปลี่ยนแปลงน้อยกว่า ทำให้การหยุดคัดลอกหน่วยความจำตามเงื่อนไขเกิดขึ้นได้เร็วกว่า ซึ่งส่วนใหญ่เป็นการหยุดคัดลอกในกรณีที่ไม่ถึง 200 รอบ (90%) นอกจากนี้ เทรดยังทำงานได้อย่างอิสระบนคอร์อื่นที่ไม่ใช่คอร์เดียวกับเวอร์ชวลแมชชีน ส่วนวิธีไลฟ์เช็คพอยน์ปกติ การคัดลอกหน่วยความจำจะต้องทำสลับกับการทำงานของแอพพลิเคชัน ดังนั้นถ้าแอพพลิเคชันมีการเปลี่ยนแปลงข้อมูลในหน่วยความจำมาก ก็จะทำให้การเช็คพอยน์ใช้เวลามากขึ้น ซึ่งกรณีที่แย่ที่สุดคือการเช็คพอยน์ที่ใช้เวลามากจนเกือบจะเท่ากับเวลาที่แอพพลิเคชันทำงานเสร็จ (ดังที่พบในเบนช์มาร์ก BT การเช็คพอยน์ใช้เวลา 103.33s แอพพลิเคชันใช้เวลาทำงาน 113.09s)

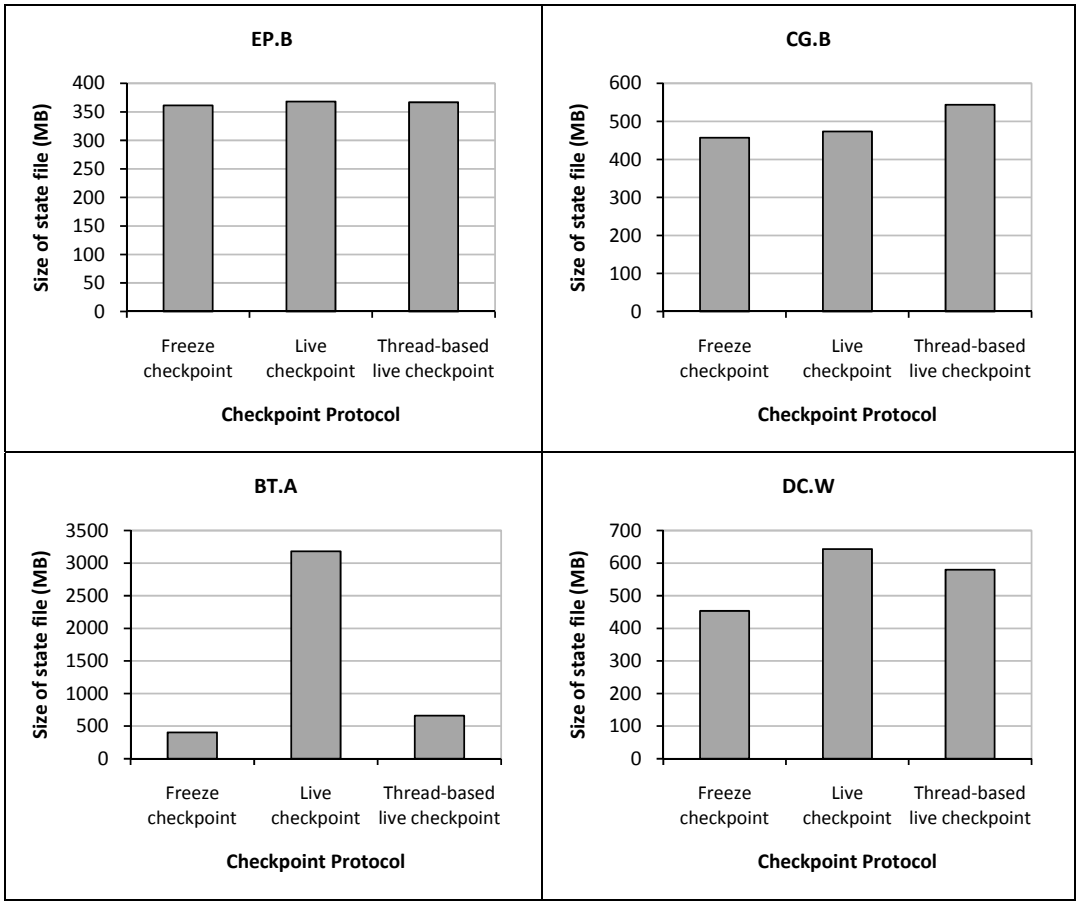
โดยปกติ เวลาที่ใช้ในการเช็คพอยน์จะประกอบด้วยเวลาเซ็ตอัพสถานะไมเกรชัน เวลาคัดลอกหน่วยความจำ เวลาบันทึกสถานะของเวอร์ชวลแมชชีน และเวลาบันทึกสถานะของดิสก์อิมเมจ อย่างไรก็ตาม สำหรับวิธีไลฟ์เช็คพอยน์ปกติ เวลาที่ใช้ในการเช็คพอยน์จะรวมถึงเวลาที่แอพพลิเคชันทำงานในขณะที่ทำการคัดลอกหน่วยความจำด้วย ส่วนวิธีหยุดการทำงาน ถึงแม้ว่าแอพพลิเคชันจะไม่ได้ทำงานในขณะที่ทำเช็คพอยน์ แต่การคัดลอกหน่วยความจำจำเป็นที่จะต้องหยุดเป็นระยะทุกๆ 100ms ตามอัลกอริทึมของเควีเอ็ม ซึ่งช่วงเวลาที่เวอร์ชวลแมชชีนหยุดทำงานจะไม่มีการเปลี่ยนแปลงข้อมูลในหน่วยความจำ ทำให้เวลาในการเช็คพอยน์ค่อนข้างคงที่และน้อยกว่าวิธีไลฟ์เช็คพอยน์ปกติในเบนช์มาร์ก CG, BT และ DC ส่วนเบนช์มาร์ก EP เป็นแอพพลิเคชันที่ต้องการใช้และมีการเขียนหน่วยความจำน้อย การเช็คพอยน์จึงสามารถสิ้นสุดได้เร็ว ทำให้ระยะเวลาที่ทำการเช็คพอยน์ของวิธีหยุดการทำงานและวิธีไลฟ์เช็คพอยน์ปกติมีค่าใกล้เคียงกัน

การตั้งเวลาเพื่อทำการคัดลอกหน่วยความจำซ้ำๆ ทุกๆ 100ms ของวิธีหยุดการทำงาน ทำให้การเช็คพอยน์ใช้เวลาเพิ่มขึ้น ในขณะที่การคัดลอกหน่วยความจำของวิธีการที่ใช้เทรดในไลฟ์เช็คพอยน์สามารถทำได้โดยไม่ต้องหยุดรอเวลาระหว่างที่คัดลอกข้อมูลในแต่ละรอบ ถึงแม้ว่าข้อมูลในหน่วยความจำจะถูกเปลี่ยนแปลงโดยเวอร์ชวลแมชชีนซึ่งทำให้เทรดต้องคัดลอกข้อมูลมากกว่าก็ตาม การเช็คพอยน์ด้วยวิธีการที่ใช้เทรดก็ยังสามารถทำได้เร็วกว่าวิธีหยุดการทำงานในเบนช์มาร์ก EP, CG และ BT ส่วนเบนช์มาร์ก DC เป็นแอพพลิเคชันที่มีการเขียนข้อมูล

จำนวนมากลงไฟล์ (2.4GB) ทำให้มีการเขียนข้อมูลในหน่วยความจำมากขึ้นและการบันทึกสถานะของดิสก์อิมเมจใช้เวลาเพิ่มขึ้น (.73s-1.04s เมื่อเทียบกับเบนช์มาร์กอื่น) นอกจากนั้น ในกรณีของการคัดลอกหน่วยความจำที่สิ้นสุดเมื่อถึง 200 รอบ หน่วยความจำที่เหลือจะมีมากขึ้น (เพิ่มขึ้น 33.98MB) เวลาที่จะต้องคัดลอกจึงมากกว่ากรณีของการคัดลอกที่สิ้นสุดก่อน 200 รอบ (เพิ่มขึ้นประมาณ .30s) เบนช์มาร์กนี้จึงทำให้เวลาในการเช็คพอยน์ของวิธีการที่ใช้เทรดในไลฟ์เช็คพอยน์มีค่ามากและใกล้เคียงกับวิธีการแบบหยุดการทำงาน

ภาพที่ 5.4

ขนาดของไฟล์สถานะเวอร์ชวลแมชชีน



สำหรับวิธีหยุดการทำงาน ขนาดของไฟล์สถานะจะได้รับผลกระทบจากขนาดของหน่วยความจำที่แอพพลิเคชันต้องการใช้ ถ้าแอพพลิเคชันต้องการใช้หน่วยความจำเพิ่มขึ้น ไฟล์สถานะก็จะมีขนาดใหญ่ขึ้น

ส่วนการเช็คพอยน์ด้วยวิธีการของไลฟ์เช็คพอยน์ เวลาที่ใช้ในการเช็คพอยน์จะมีผลกระทบต่อขนาดของไฟล์สถานะด้วย แอพพลิเคชันที่ทำให้การเช็คพอยน์ใช้เวลามากขึ้น ไฟล์สถานะก็จะมีขนาดเพิ่มขึ้นตามไปด้วย ซึ่งโดยปกติ ขนาดของไฟล์สถานะจะใหญ่กว่าการเช็คพอยน์แบบหยุดการทำงาน เนื่องจากหน่วยความจำถูกเปลี่ยนแปลงได้ในขณะที่ทำเช็คพอยน์

สำหรับวิธีการที่ใช้เทรด ขนาดของไฟล์สถานะจะเพิ่มขึ้นตามเวลาที่ใช้ในการคัดลอกหน่วยความจำได้เร็วกว่าวิธีไลฟ์เช็คพอยน์ปกติ เนื่องจากเทรดทำการคัดลอกหน่วยความจำตลอดเวลา ในขณะที่วิธีการของไลฟ์เช็คพอยน์ปกติ การคัดลอกหน่วยความจำจะถูกขัดจังหวะเป็นระยะทุกๆ 100ms อย่างไรก็ตาม ขนาดของหน่วยความจำที่จะต้องคัดลอกในวิธีไลฟ์เช็คพอยน์ปกติสามารถเปลี่ยนแปลงได้มากตามพฤติกรรมการทำงานของแอพพลิเคชัน (531.60MB-3473.69MB) ในขณะที่วิธีการใช้เทรดในไลฟ์เช็คพอยน์ ขนาดของหน่วยความจำที่จะต้องคัดลอกมีปริมาณน้อยกว่า (523.53MB-655.52MB) เพราะช่วงเวลาคัดลอกหน่วยความจำของเทรดสามารถสิ้นสุดได้เร็วกว่า นอกจากนี้ ขนาดของไฟล์สถานะจะมีผลต่อเวลาที่ใช้ในการรีสตาร์ทเวอร์ชวลแมชชีน โดยไฟล์สถานะของวิธีการที่ใช้เทรดมีขนาดใหญ่กว่าวิธีหยุดการทำงานเล็กน้อย (5.45MB-258.03MB) ซึ่งทำให้เวลาในการรีสตาร์ทมากกว่าเพียงเล็กน้อย (.26s-2.35s)

จากภาพที่ 5.4 ถึงแม้ว่าวิธีการที่ใช้เทรดในไลฟ์เช็คพอยน์จะทำให้เช็คพอยน์ไฟล์มีขนาดใหญ่กว่าวิธีการแบบหยุดการทำงาน แต่ก็ยังใช้เวลาในการเช็คพอยน์น้อยกว่า นอกจากนั้นขนาดของเช็คพอยน์ไฟล์ก็ยังไม่ใหญ่มากเกินไป (366.75MB-660.86MB) โดยมีขนาดใกล้เคียงกับขนาดเช็คพอยน์ไฟล์ของวิธีการแบบหยุดการทำงาน (361.30MB-457.05MB)

ในกรณีของเบนช์มาร์ก BT การเช็คพอยน์ด้วยวิธีการไลฟ์เช็คพอยน์ปกติจะสร้างไฟล์สถานะที่มีขนาดใหญ่กว่าการเช็คพอยน์ด้วยวิธีการอื่นอย่างเห็นได้ชัด ซึ่งเกิดจากพฤติกรรมการทำงานของเบนช์มาร์ก BT ที่เขียนข้อมูลลงหน่วยความจำเป็นปริมาณมาก ทำให้มีการคัดลอก dirty page จำนวนมากลงสู่เช็คพอยน์ไฟล์ และทำให้การเช็คพอยน์ใช้เวลานาน ส่วนกรณีของเบนช์มาร์ก DC ซึ่งมีการเขียนหน่วยความจำค่อนข้างมาก เช็คพอยน์ไฟล์จึงมีขนาดค่อนข้างใหญ่ สำหรับวิธีการของไลฟ์เช็คพอยน์ การใช้เทรดจะช่วยให้การคัดลอกหน่วยความจำสิ้นสุดได้เร็ว (170 รอบ) ในขณะที่วิธีการไลฟ์เช็คพอยน์ปกติจะทำให้การคัดลอกหน่วยความจำสิ้นสุดได้ช้ากว่า

(245 รอบ) ดังนั้น ขนาดไฟล์สถานะของวิธีการไลฟ์เช็คพอยน์ปกติจึงค่อนข้างใหญ่กว่าไฟล์สถานะของวิธีการอื่นสำหรับเบนช์มาร์ก DC

1.3 สรุปความสามารถของการเช็คพอยน์ด้วยวิธีการที่ใช้เทรดในไลฟ์เช็คพอยน์ ดังนี้

- แอพพลิเคชันสามารถทำงานได้ตามปกติในขณะที่มีการเช็คพอยน์ โดยเกือบจะไม่ถูกขัดจังหวะจากการคัดลอกหน่วยความจำ ทำให้โอเวอร์เฮดน้อยกว่าวิธีการอื่น
- เทรดทำการคัดลอกหน่วยความจำได้ต่อเนื่องอย่างรวดเร็วและไม่มีการตั้งเวลา ทำให้ใช้เวลาในการเช็คพอยน์น้อยกว่าวิธีการอื่น ถึงแม้ว่าข้อมูลที่จะต้องคัดลอกอาจมีมากกว่า
- วิธีการนี้ไม่ทำให้การเช็คพอยน์ใช้เวลานานมากเกินไปเหมือนกับวิธีการไลฟ์เช็คพอยน์ปกติ รวมทั้งเช็คพอยน์ไฟล์มีขนาดไม่ใหญ่มากเกินไปเมื่อเทียบกับวิธีการแบบหยุดการทำงาน

2. ข้อเสนอแนะ

2.1 การซิงโครไนซ์ระหว่างเวอร์ชวลแมชชีนและเทรดสำหรับการเขียนและการคัดลอกหน่วยความจำสามารถทำให้เป็น atomic operation ได้ โดยเมื่อโอเปอเรชันการเขียนหน่วยความจำและการเซ็ต dirty bit ของเวอร์ชวลแมชชีน และโอเปอเรชันการรีเซ็ต dirty bit และการคัดลอกหน่วยความจำของเทรดเป็น atomic แล้ว จะทำให้แน่ใจว่าการเปลี่ยนแปลง dirty bit ทำได้อย่างถูกต้องและโอเปอเรชันเป็นไปตามลำดับ

2.2 การบันทึกสถานะของเวอร์ชวลแมชชีนลงในไฟล์สามารถกำหนดให้เขียนลงบนดิสก์ตัวอื่นได้ เพื่อให้การเขียนข้อมูลลงดิสก์ของเวอร์ชวลแมชชีนและการเขียนเช็คพอยน์ไฟล์ของเทรดทำได้รวดเร็วมากขึ้น