

ซิสเต็มส์เวอร์ชวลแมชชีนช่วยให้ฮาร์ดแวร์แพลตฟอร์มของโฮสเครื่องหนึ่งสามารถสนับสนุนแกสโอเอสหลายระบบได้ในเวลาเดียวกัน ด้วยเทคโนโลยีเวอร์ชวลแมชชีน ผู้ใช้สามารถรันระบบปฏิบัติการที่แตกต่างกันได้บนฮาร์ดแวร์เดียวกัน เครื่องมือที่สำคัญอย่างหนึ่งของเวอร์ชวลแมชชีนมอนิเตอร์คือไลฟ์ไมเกรชัน ซึ่งเป็นความสามารถในการย้ายเวอร์ชวลแมชชีนจากเครื่องหนึ่งไปอีกร่างหนึ่งได้โดยเกือบจะไม่ต้องหยุดเวอร์ชวลแมชชีน ไลฟ์ไมเกรชันในเควีเอ็ม (KVM) สามารถใช้ในการทำเช็คพอยน์เวอร์ชวลแมชชีนได้โดยการเก็บสถานะของเวอร์ชวลแมชชีนลงในไฟล์ปกติ โปรโตคอลในการทำเช็คพอยน์นี้จำเป็นที่จะต้องหยุดเวอร์ชวลแมชชีนในระหว่างที่ทำเช็คพอยน์ ซึ่งอาจมีผลต่อประสิทธิภาพการทำงานของแอปพลิเคชัน ในการทดลองเบื้องต้นพบว่าการแก้ไขโปรโตคอลการเช็คพอยน์เดิมให้ทำงานโดยที่ไม่ต้องหยุดเวอร์ชวลแมชชีนสามารถช่วยปรับปรุงประสิทธิภาพการทำงานของแอปพลิเคชันได้ อย่างไรก็ตาม ข้อเสียที่สำคัญของวิธีการนี้คือในกรณีที่แอปพลิเคชันมีความต้องการใช้หน่วยความจำและการเขียนหน่วยความจำมาก จะทำให้การเช็คพอยน์ใช้เวลามากขึ้นและประสิทธิภาพการทำงานของแอปพลิเคชันลดลง งานวิจัยนี้จึงเสนอวิธีการไลฟ์เช็คพอยน์แบบเทรด ซึ่งหมายถึงการเช็คพอยน์เวอร์ชวลแมชชีนโดยใช้เทรดในเทคนิคไลฟ์ไมเกรชัน ด้วยวิธีการนี้ เทรดจะทำเช็คพอยน์ในขณะที่เวอร์ชวลแมชชีนกำลังทำงาน ทำให้เวอร์ชวลแมชชีนโปรเซสสามารถทำงานต่อไปได้โดยไม่ต้องทำเช็คพอยน์จนกว่าจะถึงเวลาที่ต้องหยุดเพื่อบันทึกสถานะ การอิมพลีเมนต์วิธีการนี้ทำในเควีเอ็ม และการประเมินประสิทธิภาพใช้โปรแกรมจาก NPB ผลการทดลองพบว่าโปรโตคอลไลฟ์เช็คพอยน์แบบเทรดสามารถแก้ปัญหาของกลไกการเช็คพอยน์ที่ใช้ไลฟ์ไมเกรชันโดยตรงได้ ซึ่งวิธีการนี้จะทำให้โอเวอร์เฮดลดลง แอปพลิเคชันทำงานเสร็จเร็วขึ้น ในขณะที่เวลาที่เครื่องหยุดทำงานยังคงน้อยมากเช่นเดิม

A system virtual machine (VM) provides a single-host hardware platform that supports multiple guest operating system environments concurrently. By using VM technology, users could run different operating systems on the same hardware. A key feature of virtual machine monitor is live migration, which is the ability to transport a VM from one host to another without noticeable guest downtime. KVM live migration supports the VM checkpointing capability by storing the VM state in a regular file. Its checkpointing protocol requires a VM to be stopped during the entire process of checkpointing that may affect the application performance. In the preliminary study, a slightly modified version of live-migration checkpointing protocol that allows a VM to continue its execution during the checkpoint time can improve the application performance. However, an important drawback to this implementation is that the delayed checkpoint latency induced by large memory requirement and high memory access can degrade performance. In this study, a novel thread-based live checkpointing is proposed. This model leverages live migration mechanism introducing a thread, which is responsible for the majority of the checkpoint activities. This enables a VM process to continue normal execution without performing checkpoint operations until it reaches the final stage of checkpointing. This approach is implemented in KVM and its performance evaluations are conducted using the NAS parallel benchmarks. Experiments show that thread-based live checkpoint protocol can lift out the restrictions inherited in the live-migration-based checkpointing mechanism. A notable improvement of execution time and lower overhead can be found whereas its downtime is negligible.