

## Abstract

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.

**Keywords:** Checkpoint, Virtual Machine, Live Checkpointing, Live Migration