Abstract

As large-scale computing clusters and supercomputers continue to grow, the mean time between failures becomes smaller and relevant for applications with ordinary execution times. Thus, fault tolerance mechanisms become essential to avoid losing the entire progress of an applications in the event of a fault. Checkpoint/Restore is a commonly applied fault tolerance mechanism based on saving the entire state of a process to stable storage, allowing to restore it from that state after a failure.

Due to the continuous growth of large-scale computing systems, load-balancing strategies based on virtualization techniques are also increasingly interesting in the field of high-performance computing. Container virtualization is a promising virtualization technique for high-performance computing environments, because it minimizes performance penalties of traditional hypervisor-based virtualization. Hence, this work presents the design of an application-transparent checkpointing mechanism that benefits from the potential use of containers. The designed mechanism uses a coordinated checkpointing protocol to save not only the current state of an MPI job, but the entire underlying containers, to stable storage, and is thereby inherently solving the problem of residual dependencies. The checkpointing mechanism has been integrated into ParaStation MPI, an open-source MPI implementation. Containerization is realized by means of LinuX Containers, and container checkpoints are taken with CRIU. Coordinating all processes of an MPI job and checkpointing the underlying containers is integrated into a complete checkpointing framework for containerized environments.