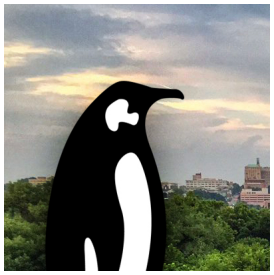

Contribution ID: **207** Type: **not specified**

pkernfs: Persisting guest memory and kernel/device state safely across kexec

Tuesday, 14 November 2023 11:30 (40 minutes)

Hypervisor live update is a mechanism to support updating a hypervisor in a way that has limited impact to running virtual machines. This is done by pausing/serialising running VMs, kexec-ing into a new kernel, starting new VMM processes and then deserialising/resuming the VMs so that they continue running from where they were. So far, all public approaches with KVM neglected device assignment which introduces a new dimension of problems. This session will highlight the additional problem space that device assignment brings and discuss potential solutions.

To support hypervisor live update with device assignment Linux needs new memory management capabilities. In addition to the ability to preserve guest memory and state across kexec, it needs to be able to persist and re-hydrate kernel and device state such as IOMMU page tables so that DMA can keep running during kexec. In this session we explore these requirements and a proposed solution: pkernfs. This is a new in-memory persistent file system which can store guest memory, userspace memory and kernel/device memory for IOMMU page tables.

We also explore other requirements around improving the security posture of guest memory and how pkernfs will solve these, such as by integrating with gmem [1] and keeping guest memory out of the kernel's direct map. By moving the guest memory into reserved DRAM it also avoids the struct page overhead for guest memory and do huge/gigating allocations similar to what DMEMFS [2] was aiming to achieve.

I will give a short demo of hypervisor live update with PCI device assignment to illustrate what is being solved.

There will be a request for reviews and feedback on the RFC which has been posted to lkml.

The QEMU side of the live update is done largely via Steven Sistare's QEMU live update patch set [3], with additional changes to support live update with PCI device passthrough; the focus of this session is on the kernel memory management side.

[1] https://lore.kernel.org/lkml/20230718234512.1690985-1-seanjc@google.com/T/
[2] https://lore.kernel.org/kvm/cover.1607332046.git.yuleixzhang@tencent.com/
[3] https://lore.kernel.org/qemu-devel/1658851843-236870-1-git-send-email-steven.sistare@oracle.com/

Primary authors: GRAF, Alexander; GOWANS, James (Amazon EC2)

Presenters: GRAF, Alexander; GOWANS, James (Amazon EC2)

Session Classification: KVM MC

Track Classification: LPC Microconference: KVM MC