Preserving State for Fast Hypervisor Update

Pasha Tatashin, Google / Microsoft

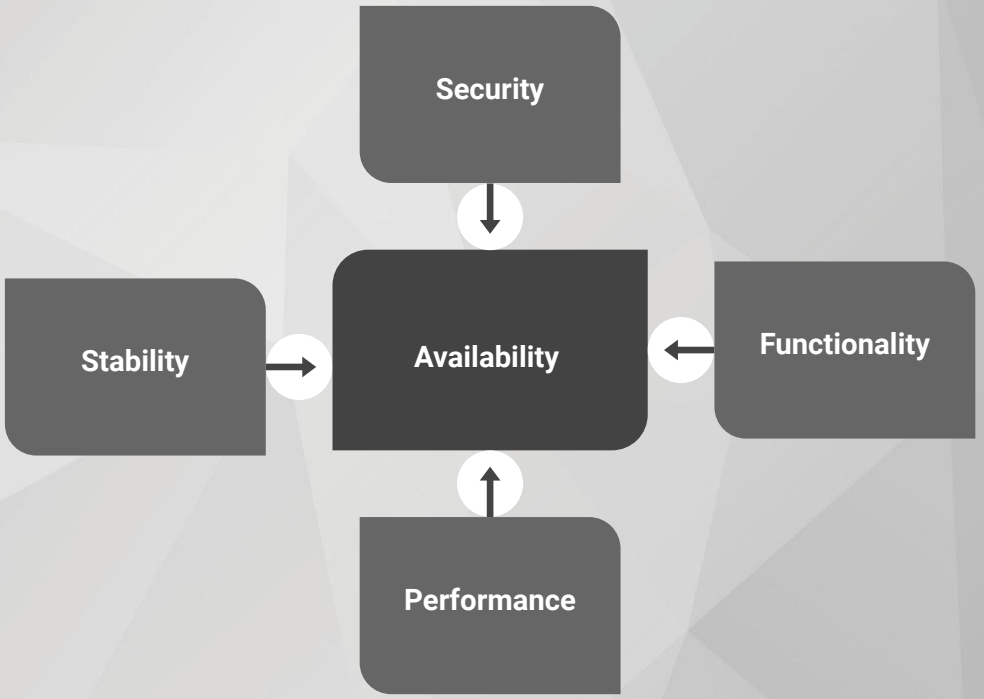
LINUX
PLUMBERS
CONFERENCE › September 20-24, 2021

Outline

- Fast Hypervisor Update Introduction
- Hot-patching and Live Migration vs. Fast Local Update
- Fast Local Update
- Recent Publications
- Fast Local Update: Building Blocks
- Fast Local Update: Demo Cloud Hypervisor
- Fast Local Update: Data for QEMU
- Discussion topics:
  - Emulated PMEM Interface
  - Preserving device state across reboot
  - Support for Virtual Functions
- Future: Multi-root update

Fast Hypervisor Update Introduction

- Cloud relies on virtualization
- Hypervisors must be provisioned with minimal disruption for the tenants
- Shutting down VMs is not an option, therefore a fast traditional reboot is not a solution



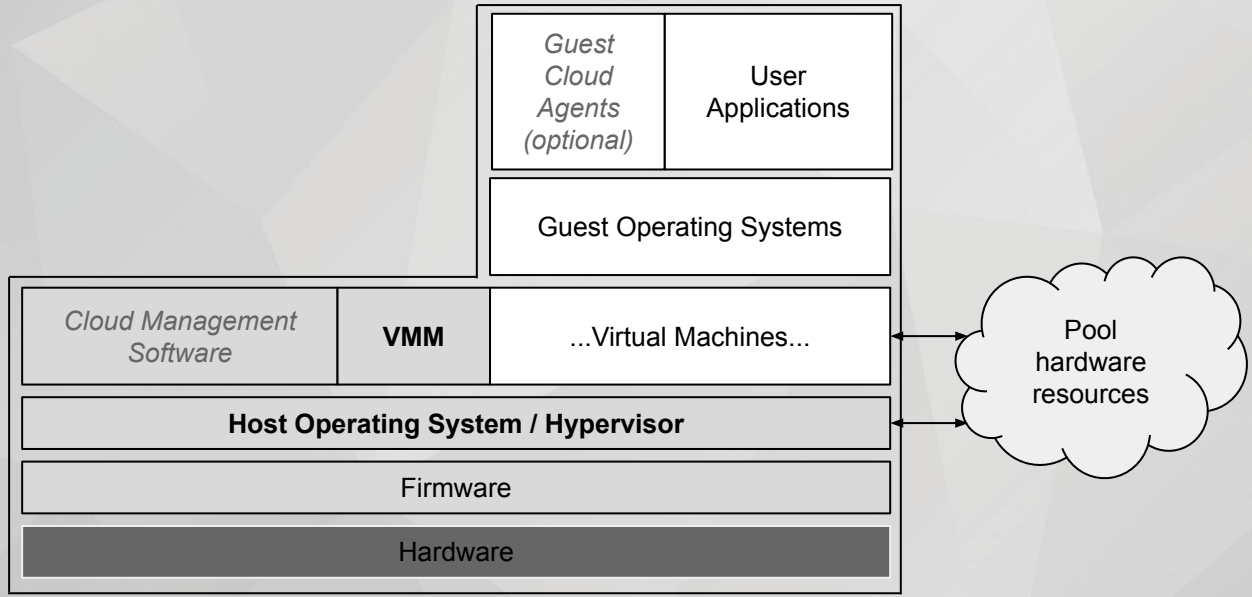
Hot-patching and Live Migration vs. Fast Local Update

- Hot Patching - Excellent for quick security fixes, not an update solution.
  - Extra development expenses
  - Maintenance matrix complexity
  - Limited inlined functions and macros support
  - Limited data structures modification
  - Not a software update
  - Slowdown
  - Extra testing
  - Kernel rejuvenation

- Live Migration - Excellent for hardware provisioning, and load balancing.
  - Additional network traffic
  - Keeping device state challenges
  - Virtual machine utilization dependence
  - Extra hardware availability

Fast Local Update

- Being able to update all parts of virtualization stack without terminating virtual machines locally on the host
- The software parts may include: VMM, Driver Modules, Kernel, and Firmware



Recent Publications

- Bagdi, H., Kugve, R., and Gopalan, K. (2017). Hyperfresh: Live refresh of hypervisors using nested virtualization.
- Zhang, X., Zheng, X., Wang, Z., Li, Q., Fu, J., Zhang, Y., and Shen, Y. (2019). Fast and scalable VMM live upgrade in large cloud infrastructure.
- Sistare, Steven (2020). Qemu live update. In KVM Forum 2020 Lyon, France.
- Zeng, Jason (2019). Seamless cloud system upgrade with vmm fast restart. In KVM Forum 2019 Lyon, France.
- Zeng, Jason (2020). Device keepalive state for local live migration and vmm fast restart. In KVM Forum 2020 Lyon, France.
- Russinovich, Mark, et al. "Virtual machine preserving host updates for zero day patching in public cloud." Proceedings of the Sixteenth European Conference on Computer Systems. 2021.
- Pacheco, Dino Lopez, et al. "Hy-FiX: Fast In-place Upgrades of KVM Hypervisors." IEEE transactions on cloud computing (2021)
- Tatashin, Pavel, and William Moloney. "In-Place VM Migration for Fast Update of Hypervisor Stack." Intelligent Computing. Springer, Cham, 2022
- Tatashin, Pavel, and William Moloney. "Seamless Update of a Hypervisor Stack." Intelligent Computing. Springer, Cham, 2022

Fast Local Update: Building Blocks

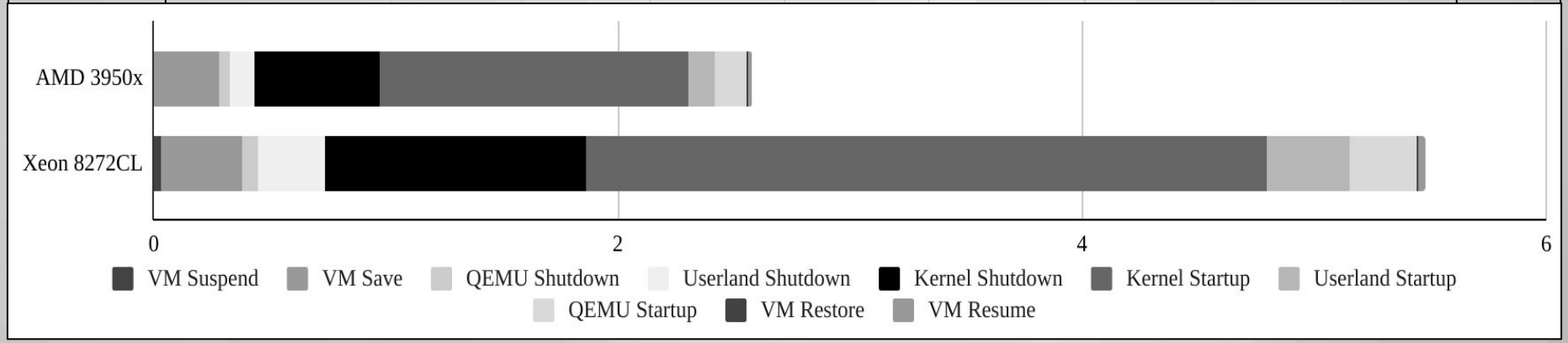
- Kexec
  - Fast reboot
  - Skip firmware reset
  - Enables memory state preservation
- Local Live Migration
  - Migration to a file, or snapshotting
  - Preserves the VM state across reboot, so hypervisor can be rebooted
- Emulated PMEM + DAX + EXT4
  - Allows using VM memory as a file on a RAM based filesystem without pagechache
  - Enables skipping memory copy during local live migration
  - Enables preserving VM state across reboot

Fast Hypervisor Update: Demo Cloud Hypervisor

- Link to the demo
  https://youtu.be/fRLYCeyQLcI

Fast Hypervisor Update: Data for QEMU

Machine	VM Suspend	VM Save	QEMU Shutdown	Userland Shutdown	Kernel Shutdown	Kernel Startup	Userland Startup	QEMU Startup	VM Restore	VM Resume	Total
AMD 3950x	0.0021	0.282	0.042	0.1051	0.5405	1.3324	0.1166	0.1368	0.0021	0.0157	**2.5753**
Xeon 8272CL	0.0297	0.3537	0.0656	0.2857	1.1296	2.9323	0.3571	0.2898	0.0045	0.0309	**5.4789**



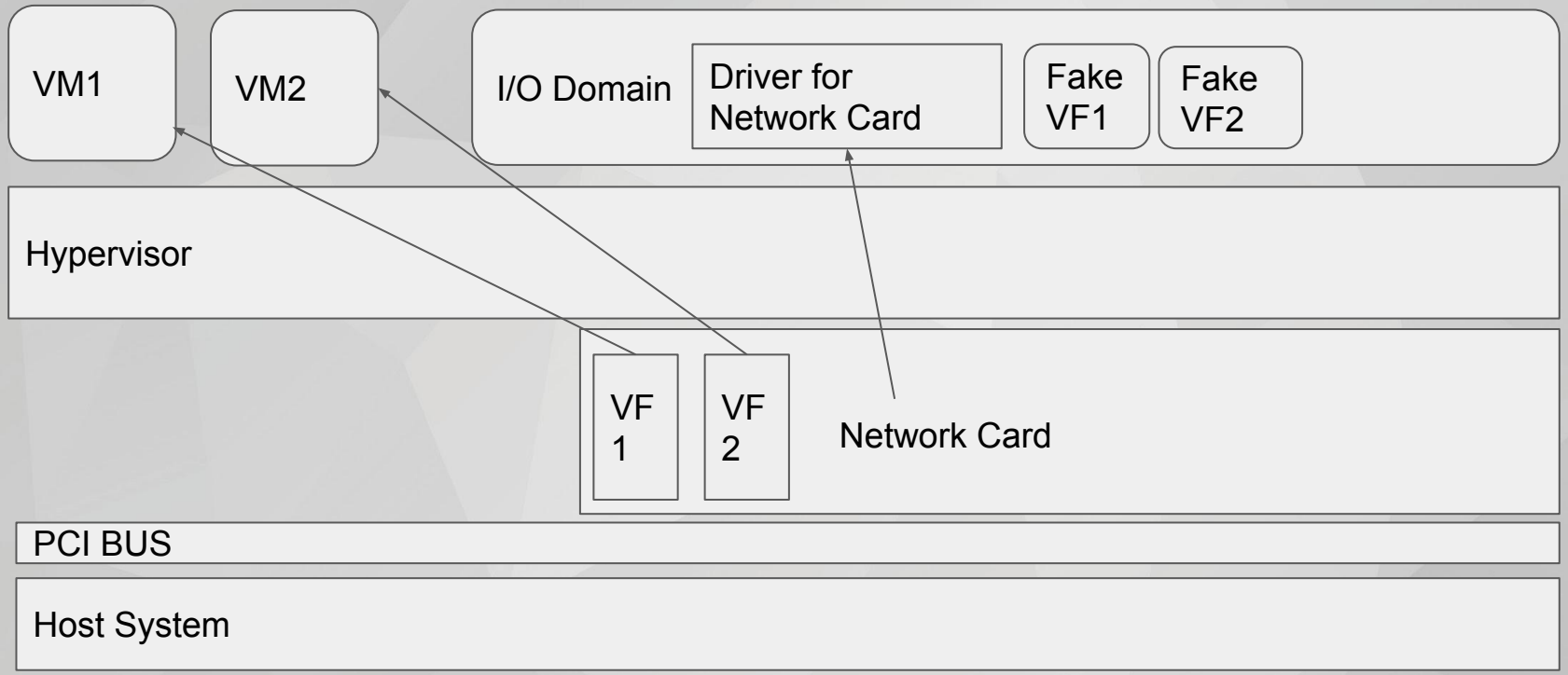
Discussion topics: Emulated PMEM Interface

- Emulated PMEM can be created in three ways
  - UEFI marks part of regular memory as *Type 7*
  - Add *pmem* node under root node in the device tree
  - Kernel parameter memmap=nn[KMG]!ss[KMG]
- The admin must know the intimate knowledge about the physical memory layout.
- Not portable
- “Legacy” PMEM?

Discussion topics: Preserving Device State Across Reboot

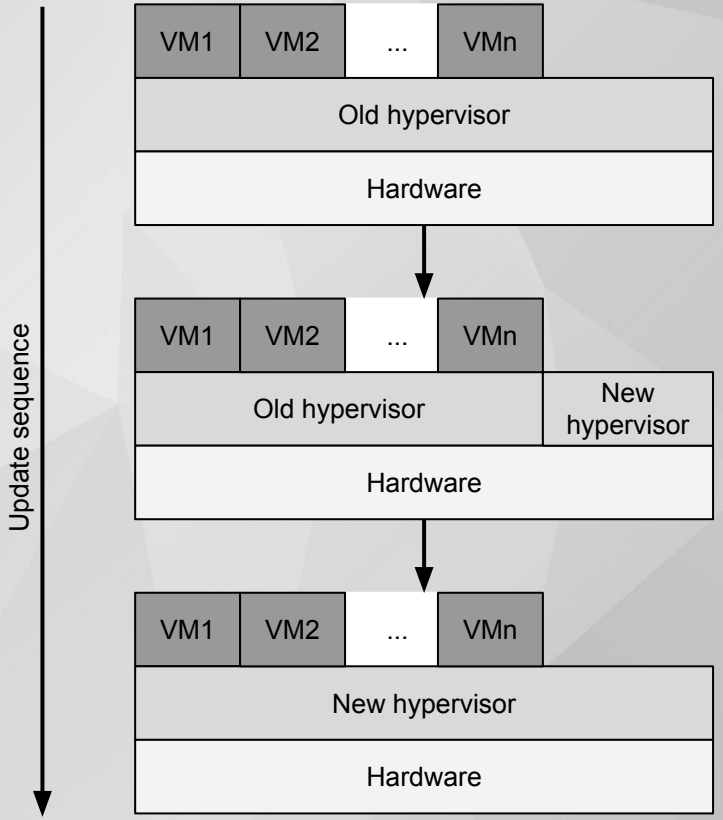
- For Intel IOMMU based device the preserved state includes saving the following data across reboot:
-
    - VMX posted interrupts **tables** and soft data structures
    - IOMMU **page tables** and soft data structures
    - Interrupt Remappings soft data structures
    - PCI soft data structures
    - VFIO PCI soft data structures
- The soft data structures are relatively small in size and can be portably passed from one kernel to another, but interrupt tables, and page tables which are configured for the devices must stay in place.
- Can we use the same memory where the rest of VM memory is preserved to create those data structures?

Discussion topics: Support for Virtual Functions



Future: Multi-root update

- Dual boot can be achieved with a modified kexec call that starts new kernel on offlined CPUs and with specifically configured memmap parameter without shutting down the current kernel.
- Pass the VM handling from one hypervisor to another on the fly by using local live migration, and PMEM.
- Pass the passthrough device states by passing the preserved soft data structures, and also the device state that was also allocated on PMEM with VMs.



Questions

Thank you