

Live Updates @ Akamai KVM Forum, September, 2024

Michael Galaxy

mgalaxy@akamai.com

<https://www.linkedin.com/in/mrgalaxy/>

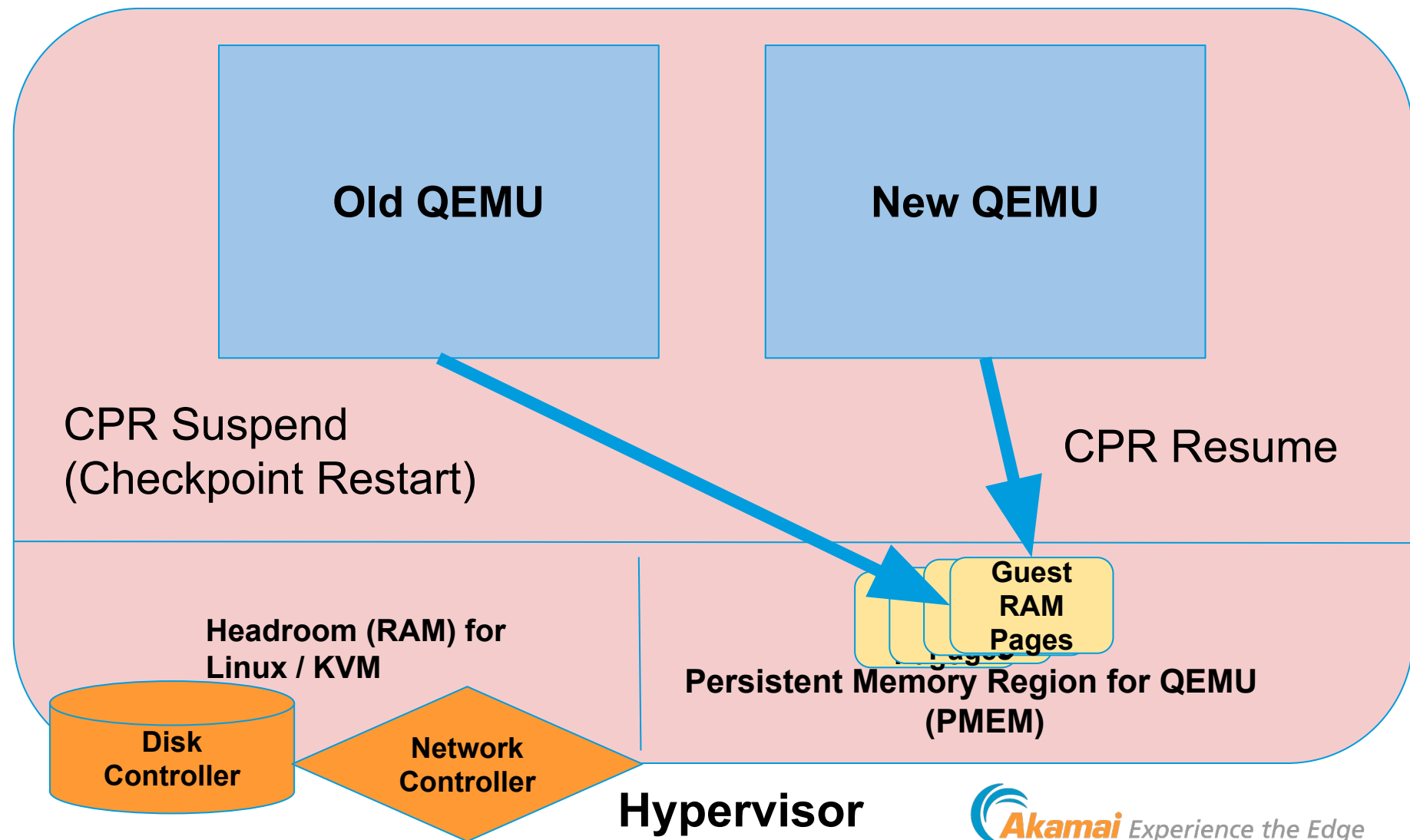


What are we doing with QEMU @ Akamai?

- Akamai known for CDN and Security businesses
- Cloud development for several years
 - ◆ internal-only cloud platform
 - ◆ Roughly 15K kvm guests
- Purchased **Linode** in 2022
 - ◆ > 500,000 kvm guests
- Difficulty finding live migration target hosts
 - ◆ Major supply-chain issues during COVID
Severe capacity constraints. Difficulty purchasing hardware.
 - ◆ Difficulty finding live migration target hosts: e.g. CPU flags. Lack of space. Security restrictions. Customer restrictions.
- We want to update our software / kernels MUCH faster:
 - ◆ Highly-secure infrastructure:
 - Aggressive Encryption, Attestation, Audits, of CDN and Cloud businesses at all levels of software.
 - Delicate balance between software rollouts and maintaining secure systems
 - ◆ Hosts are slow to be updated without Live Migration
- Reducing our “spare” / “stranded” capacity is desired.

Workflow

1. Store guest device state (without RAM)
2. Preserve cgroup hierarchy & QEMU command line
3. Shutdown QEMU
4. Cleanly sync and unmount all filesystems
5. kexec into new new kernel (maybe quiesce devices)
6. Enter new kernel
7. Re-initialize all devices, udev, and re-mount all filesystems
8. Reconnect and mount persistent memory over DAX filesystem
9. Reconstruct cgroup hierarchy & vhost / virtio filedescriptors (bypassing libvirt)
10. Bring QEMU back online and re-attach memory (CPR)



Live Update Configuration

First, Configure PMEM: (Persistent Memory: www.pmem.io)

- ◆ Already available **now in Linux**. On top of PMEM: DAX + EXT4:
 - `/path/to/pmem => /dev/pmem0/path/to/pmem/libvirt/qemu/domain/pc.ram`
- ◆ Guest-level:
 - `qemu-system-x86_64 -object { "qom-type":"memory-backend-file",
"id":"pc.ram",
"mem-path" : "/path/to/pmem/libvirt/qemu/domain/pc.ram",
"share":true,
"size":xxxx,
"host-nodes":[0]
}`

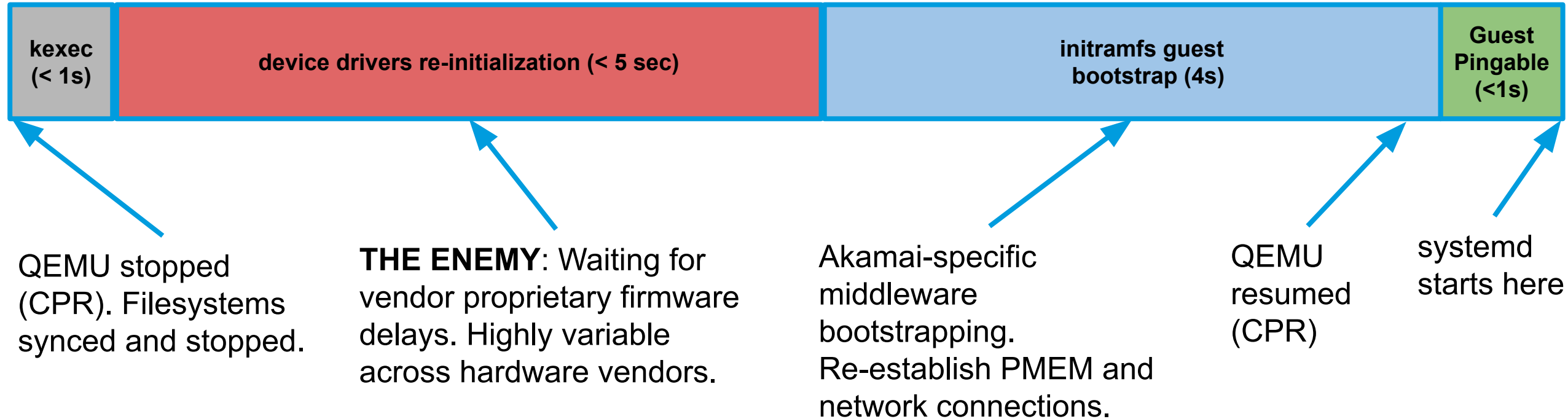
Next, Configure CPR (Steve Sistare @ Oracle) officially merged in QEMU v9:

1. `migrate-set-capabilities: { "capability": "x-ignore-shared" }`
2. `migrate-set-parameters: { "mode": "cpr-reboot" }` # do not exec(), just exit and restore later)
3. `migrate: { "uri": "file:/path/to/qemu_save_file" }`

Then: `kexec` to new kernel (and potentially new userland)

Finally: Bring QEMU back to life with the same configuration.

Performance Reality-Check









1. “Trivial” Live updates (simple hardware): Potentially under 5 seconds
2. Complex hardware without middleware optimizations: ~ 10 seconds
3. Device drivers can balloon to many seconds
 - a. =====> Still fast enough to keep TCP Connections alive!

Challenges

1. Persistent memory headaches:
 - a. Hard-coded persistent memory regions
vmlinux memmap=XXXXXX
System memory overhead is now permanent. Cannot be changed with VM restart.
 - b. NUMA node migration no longer allowed. Must be overcome at cloud level.
 - c. NUMA and PCI Memory alignment
 - d. Swap is impossible
2. Controlling the kernel command line and loaded drivers
3. Loop testing (perpetually running VMs)
4. systemd overheads / bypassing
 - a. Booting out of initramfs
 - b. recreating cgroup hierarchies
5. Standard kexec-based issues
 - a. Needing to re-initialize all devices from scratch
 - b. Vendor firmware bugs..... poor responses / resolution from the vendor
6. Time-keeping a surprise: Time stops inside the guest during the live update
7. **[PATCH v3 0/4] multiple memory backend support for CPR Live Updates**
 - a. Live-migrating PMEM guests (upgrading and downgrading)
 - b. NUMA-support for PMEM guests

Multiple Persistent Memory Options

- 1) PMEM (Persistent Memory): No kernel changes required. What we're using now.
- 2) PKRAM (Oracle): <https://lkml.org/lkml/2023/4/26/1008>
 - a) Incorporates a tmpfs-based solution. Does *not* require DAX (another layer of indirection)
- 3) pkernfs (Amazon): <https://lkml.org/lkml/2024/2/6/916>
 - a) Still requires reservation at boot time.
- 4) PRMEM (Microsoft): <https://lkml.org/lkml/2023/10/16/1492>
 - a) Accessed via modified ramdisk + DAX support => Ramdisks are not swappable
 - b) shrinking ramdisk footprint unclear, but has API for growing reserved memory amount

LKML Option	Merged?	Resizable?	NUMA-compatible?	Swap-compatible?
PMEM		X		X
PKRAM (Oracle)	X			
PRMEM (Microsoft)	X		X	X
pkernfs (Amazon)	X	X	not sure / probably	X

Thanks!

Questions?

mgalaxy@akamai.com