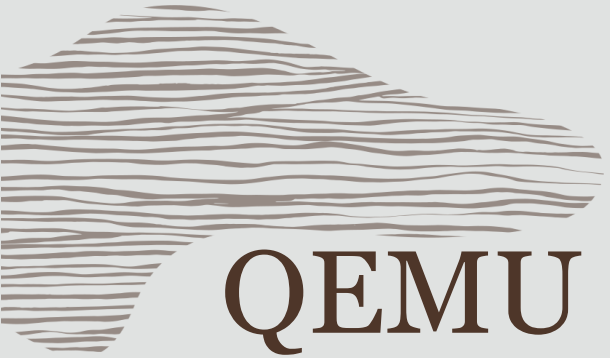
ORACLE



QEMU Live Update
KVM Summit 2020

Steve Sistare
Software Architect
Oracle Corporation
October 30, 2020

What is Live Update?

- Method to update QEMU to new version, while keeping the guest alive. Minimal impact:
  - Guest pause ~100 msec
  - Transparent to guest clients, no loss of connections.
  - Supports SRIOV w/o guest cooperation
- Enable critical bug fixes, security mitigations. New features.
- Less costly than Live Migration
  - Reduced resource consumption – memory, network, hosts
  - No unbounded, reduced performance phase.
  - Suitable for large local storage

Design Elements

- Old qemu process exec's new qemu binary
- Guest memory preserved in-place in RAM
- External descriptors preserved
  - Serial console, qemu monitor, vnc, pty, vhost
- vfio/sriov descriptors preserved
- KVM descriptor (instance) *not* preserved; destroy and recreate
- QEMU device state serialized and saved to file

qemu monitor interfaces: cprsave, cprload

- CPR = CheckPoint and Restart
- QMP and HMP

Preserving guest memory

- **Proposed extension:** madvise(MADV_DOEXEC, addr, len)
  - Mappings at addr preserved across exec, at same VA.
    - for MAP_ANON anonymous memory
  - exec'd binary must opt-in via ELF note
- Implementation
  - madvise sets VM_EXEC_KEEP flag on vma's
  - exec copies marked vma'a from old mm to new mm (like fork)
    - iff target binary contains ELF note
- See lkml: [RFC PATCH 0/5] madvise MADV_DOEXEC

Preserving guest memory (2)

- VFIO DMA mappings remain valid.
  - DMA activity continues while guest is paused.
  - iova → va → pfn safe throughout.
- Save addr, len in environ, use after exec

```
setenv QEMU_ADDR_pc.ram  140611724247040
setenv QEMU_LEN_pc.ram   1073741824
```

- Attach mem to new KVM instance via KVM_SET_USER_MEMORY_REGION
  - O(npages) cost : 1 msec / GB
  - Constant cost with kernel patch

    4b442955 KVM: x86: avoid unnecessary rmap walks when creating/moving slots (5.8)

VFIO device state

- PCI BAR and config regions
  - vfio-device fd
    post-exec mmap BAR, read config
- DMA mappings
  - iommu group fd: /dev/vfio/<group>
    container fd: /dev/vfio/vfio
- Interrupt state
  - eventfd's: err_irq, req_irq, msix_irq
  - msix_table, msix_pba in vmstate file
- Clear FD_CLOEXEC, save fd's in environ.
- Attach to new KVM and kvm-vfio
  KVM_CREATE_DEVICE KVM_IRQFD
  KVM_SET_GSI_ROUTING KVM_IOEVENTFD

```
setenv QEMU_FD_/dev/vfio/65                 15
setenv QEMU_FD_vfio_container_65        24
setenv QEMU_FD_0000:3a:10.0                 26
setenv QEMU_FD_0000:3a:10.0_err_0     27
setenv QEMU_FD_0000:3a:10.0_req_0     28
setenv QEMU_FD_0000:3a:10.0_interrupt_0             162
setenv QEMU_FD_0000:3a:10.0_kvm_interrupt_0  163
setenv QEMU_FD_0000:3a:10.0_interrupt_1             164
setenv QEMU_FD_0000:3a:10.0_kvm_interrupt_1    165
```

- Tested qemu, irqchip, and posted interrupt delivery across update.

QEMU Device State

- Leverage vmstate framework from live migration
  - select handlers based on operation
- Serialize to ordinary file. Not a socket, not qcow snapshot.
- No block device snapshot

  → must not modify block device between save/restore
- Small, less than 1 MB
- Fast

New interfaces:
cprsave <file> <mode>

- Pause guest VCPUs
- Save qemu device state to file
- Call MADV_DOEXEC on RAM segments
  - Guest main memory, video ram, etc.
- Preserve VFIO descriptors and state
- Preserve other descriptors
- Destroy KVM instance

- exec new qemu binary with same argv
  - Optionally call /usr/bin/qemu-exec
  - Site-specific exec trampoline to modify argv or the execution environment.
- qemu starts
  - Create new KVM instance
  - Reuse RAM
  - Reuse vfio and other descriptors
  - Attach vfio fd's to new KVM instance

New interfaces
cprload <file>

In pre-launch state

- **send device-add commands (if any)**

Send cprload

- **Load device state from file**
  - **Includes vfio MSIX state**
- **Continue VCPUs**

Guest pause time: 100 msec

- **optimize?**

Example 1

Window 1

```
# qemu-system-x86_64 ...
QEMU 4.2.0 monitor - type 'help' ...
(qemu) info status
VM status: running


(qemu) cprsave vm1.savevm restart
QEMU 4.2.1 monitor - type 'help'

(qemu) info status
VM status: paused (prelaunch)

(qemu) cprload vm1.savevm

(qemu) info status
VM status: running
```

Window 2

```
# yum update qemu
```

Demo Video

Demo Video

Legacy Live Update

- Update legacy qemu process to latest version
  - Inject code into legacy qemu process to perform cprsave: vmsave.so
    - Access qemu data structures and globals
      - eg ram_list, savevm_state, chardevs, vhost_devices
      - dlopen does *not* resolve them, must get addresses via symbol lookup.
    - Delete some vmstate handlers, register new ones (eg vfio)
    - Call MADV_DOEXEC on guest memory. Find devices, preserve fd's
- Hot patch a monitor function to dlopen vmsave.so, call entry point
  - write patch to /proc/*pid*/mem
  - Call the monitor function via monitor socket
- Send cprload to update qemu
- vmsave.so has binary dependency on qemu data structures and variables
  - Build vmsave-*ver*.so per legacy version
  - Indexed by qemu's gcc build-id

Kernel Live Update

- cprsave *file* reboot; kexec boot; cprload *file*
- Update host kernel while pausing guest (longer pause)
- Guest ram in /dev/shm, preserved across kexec reboot
  - shm PFNs copied to free pages, linked together. Head passed across kexec.
  - pages removed from free list early in boot.
  - shm inode re-created after reboot, pages added to file address_space
  - parallelized for speed
  - see PKRAM kernel patches
- Supports SRIOV via guest suspend to ram
  - requires guest agent (eg qemu-ga)
  - guest-suspend-ram; cprsave; kexec; cprload; system_wakeup
  - guest drivers flush requests and re-initialize → no device state to save/restore

Future Work

- **Merge with Intel work to preserve SRIOV across reboot? Eliminates guest agent.**
  - **Jason Zheng,** Device Keepalive State for Local Live Migration and VMM Fast Restart
- **Alternative to MADV_DOEXEC**
- **Revise QEMU patches**

References

[1] QEMU Live Update patches
Steve Sistare, Mark Kanda, Maran Wilson
https://lore.kernel.org/qemu-devel/1596122076-341293-1-git-send-email-steven.sistare@oracle.com

[2] MADV_DOEXEC kernel patches
Anthony Yznaga, Steve Sistare
https://lore.kernel.org/lkml/1595869887-23307-1-git-send-email-anthony.yznaga@oracle.com/

[3] PKRAM kernel patches
Anthony Yznaga
https://lore.kernel.org/lkml/1588812129-8596-1-git-send-email-anthony.yznaga@oracle.com

[4] KVM_SET_USER_MEMORY_REGION kernel patches
Anthony Yznaga
https://lore.kernel.org/patchwork/cover/1251714/