redhat.

virtio-mem: Paravirtualized Memory

KVM Forum 2018, Edinburgh, Scotland

David Hildenbrand
Software Engineer
26. October 2018

AGENDA

- Memory Hot(un)plug and Ballooning *
  - Basics
  - Main Differences
  - (Selected) Issues
- Hyper-V and XEN
  - Paravirtualized Memory Hotplug
- virtio-mem
  - Design Goals
  - Idea + Details
  - It's not "ballooning"
  - Planned Steps + Current State

* virtio-balloon

Memory Hot(un)plug and Ballooning

Basics

- **Memory Hotplug**
  - Add completely new memory to a system (e.g. plug a DIMM)
- **Memory Hotunplug**
  - Remove memory completely from a system (e.g. unplug a DIMM)
  - Whole applicable memory (e.g. DIMM) has to be evacuated first
- **Balloon Inflation**
  - Allocate some memory in the guest and tell the hypervisor about it
  - Used by some people for memory unplug
- **Balloon Deflation**
  - Free previously allocated memory in the guest after telling the hypervisor

Memory Hot(un)plug and Ballooning

Main Differences

	Memory Hot(un)plug	Ballooning
Environments	physical/virtual	virtual
Granularity	DIMM, e.g. >= 128 MB on x86 Linux	Page, e.g. 4 KB
Architecture dependent	Yes (HW interface)	No (except page size)
Likelihood of unplug/inflation succeeding	Small*/Medium	High

* e.g. Linux requires memory to be onlined MOVABLE

Memory Hot(un)plug

(Selected) Issues

- Some architectures **don't support memory hotplug**
  - e.g. s390x only has "standby memory"
- Some architectures **don't support memory hotplug notification**
  - e.g. ARM64 requires manual memory probing in the guest
- Unplugging of memory in Linux requires **MOVABLE zone**
  - … and still might fail if one single page can't be moved
  - … and there are many issues to that (zone imbalance …)
- **Different limits** (ACPI slots, MMAPs, KVM memory slots, minimum DIMM size …)
  - Limited flexibility for hot(un)plug granularity

Ballooning (virtio-balloon)

(Selected) Issues

- **Broken* by design**
  - Guest can (and does!) reuse inflated memory and fake balloon stats
  - Hypervisor cannot reject any inflation/deflation request
- Based on **4 KB pages**
  - Huge pages/different page size in the hypervisor?
- **Not NUMA aware**
- Used for **different use cases**
- **“Real” memory hotplug requires other technologies**

* esp. for memory hotunplug

Architecture Dependent Memory Hotplug
+
Ballooning for Memory Unplug?

… we can do better …

Hyper-V and XEN

Paravirtualized Memory Hotplug

- **Paravirtualized interface** to plug/indicate new memory
  - Ballooning to unplug (and replug) memory
- **Unplugged memory is protected** in the hypervisor
  - e.g. writing is forbidden under Hyper-V
- Based on **4 KB pages**
- **Not NUMA aware**
- Reboot handling
  - XEN: Balloon has to be reinflated
    - *What if the balloon driver doesn't start / starts too late?*
  - Hyper-V: e820 map is fixed up - "Memory reorganized"
    - *Problematic in QEMU (e.g. migration, NUMA, DIMMs ...)*

virtio-mem

virtio-mem

Design Goals

- **Unified memory hot(un)plug for all architectures**
  - avoid mixing technologies (e.g. ACPI and virtio)
- **Manage size changes completely inside QEMU**
  - Don’t require e.g. plugged DIMMs on command line
  - Simplify migration
- Provide a safe way to **detect malicious guests**
  - Unplugged memory should not be reused by mistake
- Support **different page sizes**/huge pages
- Support **NUMA**
- **…**

virtio-mem

What we want

"Try to add ***x*** **MB** to **node** ***y***"

"Try to remove ***x*** **MB** from **node** ***y***"

"How much memory was actually added/removed?"

What we have

Messing with DIMMs and ballooning in the guest and the hypervisor

What virtio-mem provides

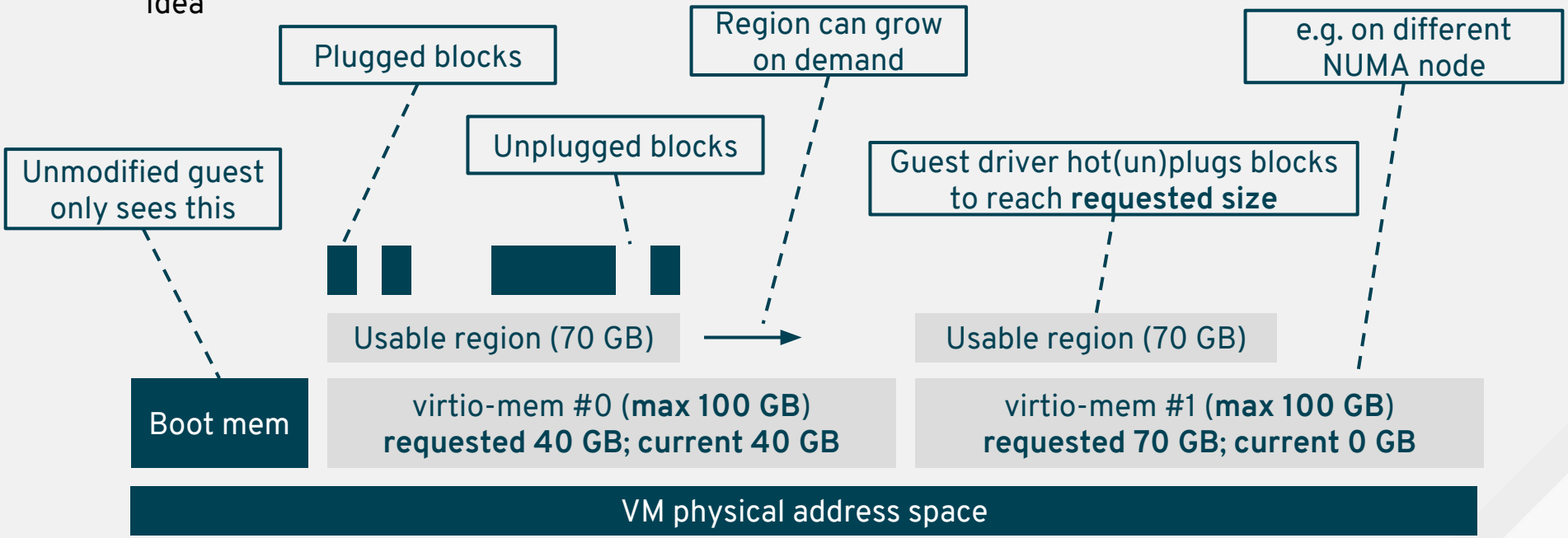
"New virtio-mem **device** ***z*** with **maximum size** ***x'*** on **node** ***y***"

"Set requested size of virtio-mem **device** ***z*** to ***x*** **MB**"

"Get current size of virtio-mem **device** ***z***"

virtio-mem

Idea



virtio-mem

Details

- **“current size”**
  - We always know (and track) how much memory the guest has plugged
- **“requested size”**
  - Indicates requests to add/remove memory
- **“block size”**
  - Hot(un)plug granularity. Configurable.
- **“max region size”**
  - Reserved memory region. 1 mmap in QEMU.
- **“usable region size”**
  - Actual region size the guest can use for plug/unplug.
  - Can grow with “requested size” up to “max region size”.
- **Plug block in QEMU**
  - track state in bitmap (+ unprotect)
- **Unplug block in QEMU**
  - track state in bitmap + madvise(DONTNEED) (+ protect using userfaultfd WP)

It’s not “ballooning”

virtio-mem

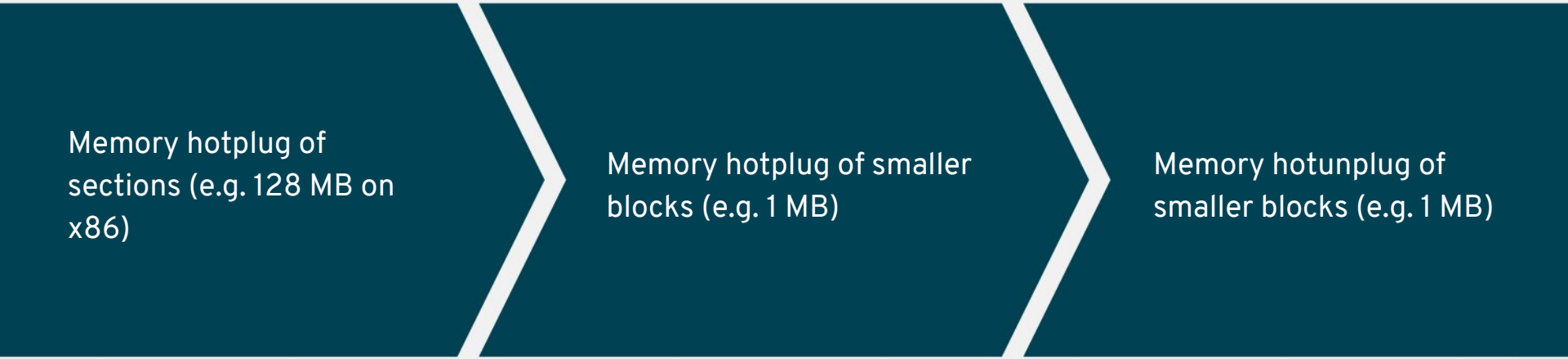
It’s not “ballooning”

- virtio-mem works on (configurable) **bigger blocks (e.g. 1 MB)**
  - Not pages like balloon drivers
- Device only works on **assigned memory region (plug/unplug)**
  - Not on all system memory / DIMMs
  - NUMA aware even for guests without NUMA support
- We can **detect malicious guests**
  - During boot, only unmodified boot memory will be used
  - We can protect unplugged memory (e.g. read-only)
- **Makes life easier* in QEMU**
  - We can resize the memory region e.g. on reboots
  - Protection of memory can be controlled by device

* life is never easy in QEMU

virtio-mem

Planned steps for Linux Guests



“Rome wasn’t built in one day” … probably not in one year either

virtio-mem

Planned steps for Windows Guests

… it’s difficult ...

virtio-mem

Current State

- **QEMU side**
  - Done: Initial prototype and virtio protocol
  - In progress: Allow virtio devices to be memory devices
  - TBD: “real” resizable memory regions (mmap hackery)
  - TBD: protect unplugged memory (e.g. via userfaultfd WP)
  - TBD: migration/dump should not access unplugged memory
- **KVM side**
  - TBD: atomically resizable memory regions
- **Linux driver side**
  - In progress: Adding/removing memory from a device driver
  - TBD: Fake onlining/offlining of e.g. 1 MB blocks in a certain memory range
  - TBD: Hinder kdump from accessing unplugged blocks

redhat.

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHat

youtube.com/user/RedHatVideos

Backup Slides

Ballooning (virtio-balloon)

Use Cases

- Collaborative memory management
  - Inflate/Deflate: Move free memory between VMs
  - e.g. “auto-ballooning”, strong memory overcommitment
  - -> Free page hinting
- Free up memory in caches
  - Inflate: Memory pressure triggers clearing of guest page cache
  - -> virtio-pmem / virtio-fs
- Memory hot(un)plug
  - Inflate/Deflate: Add/remove memory to/from a VM
  - Memory hotplug limited by balloon size
  - -> virtio-mem

Memory Hotplug and Ballooning

Why not combine both?

Use e.g. ACPI memory hotplug for adding memory and ballooning for removing memory

- How to handle reboots with an inflated balloon?
  - Require to reinflate the balloon?
  - Resize? What to resize? Which DIMMs to drop? What about migration?
- How to detect malicious guests?
  - Remember, virtio-balloon is broken by design
- NUMA aware memory unplug?
  - Remember, ballooning and NUMA is difficult (e.g. OS without NUMA)
- What about architectures without proper memory hotplug interfaces? Architectures without support for DIMMs?