KVM GUEST FREE PAGE HINTING

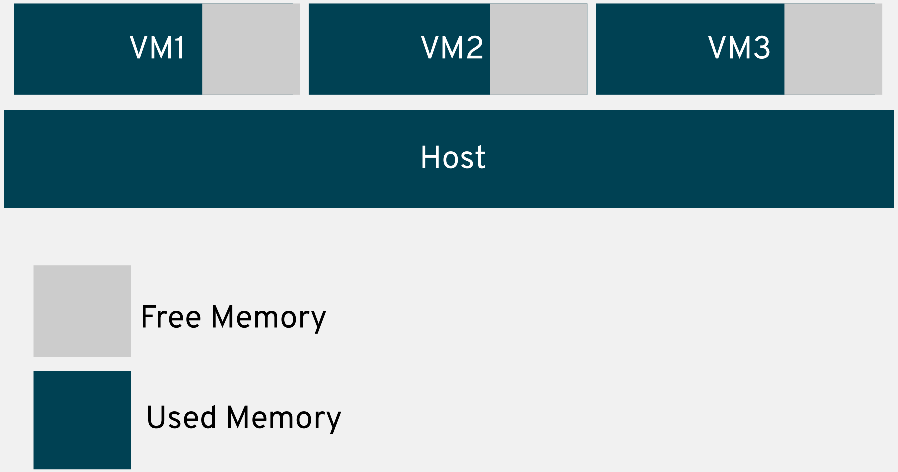
Nitesh Narayan Lal
Software Engineer
October 26th 2018

AGENDA

- Issue
- Objective
- Implementation

ISSUE

Inability of the guest to report the free memory back to the host even if the host is running out of memory.



VM1

VM2

VM3

Host

Free Memory

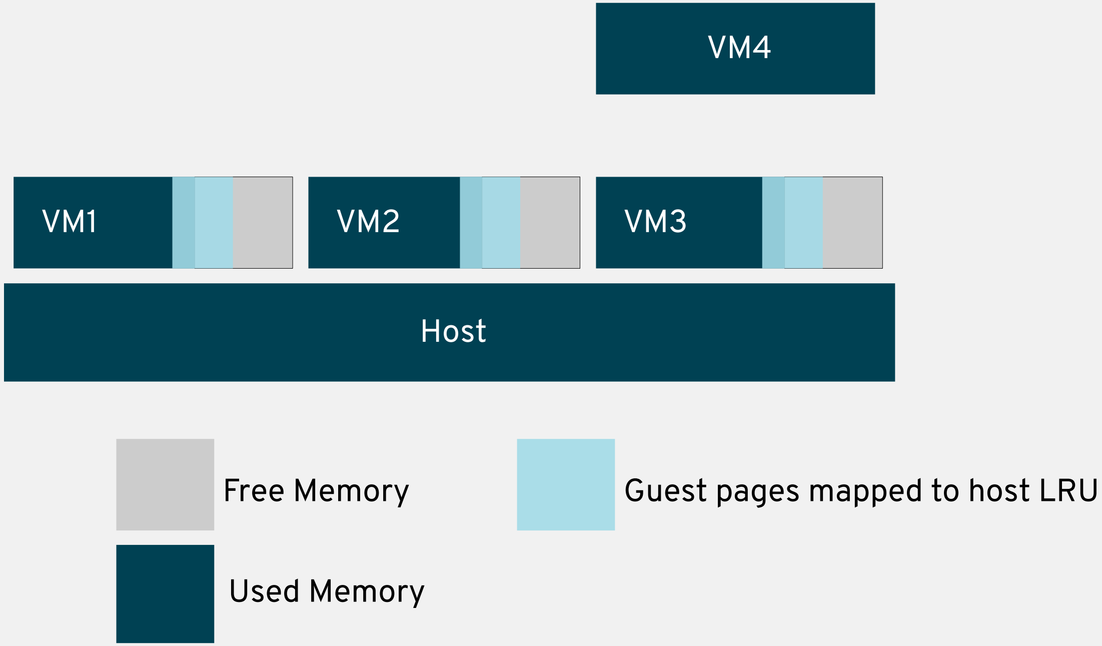
Used Memory

Host wants to launch another guest...

- Is it possible?
  - YES

Photo by pixabay.com from Pexels

How?



VM4

VM1 VM2 VM3

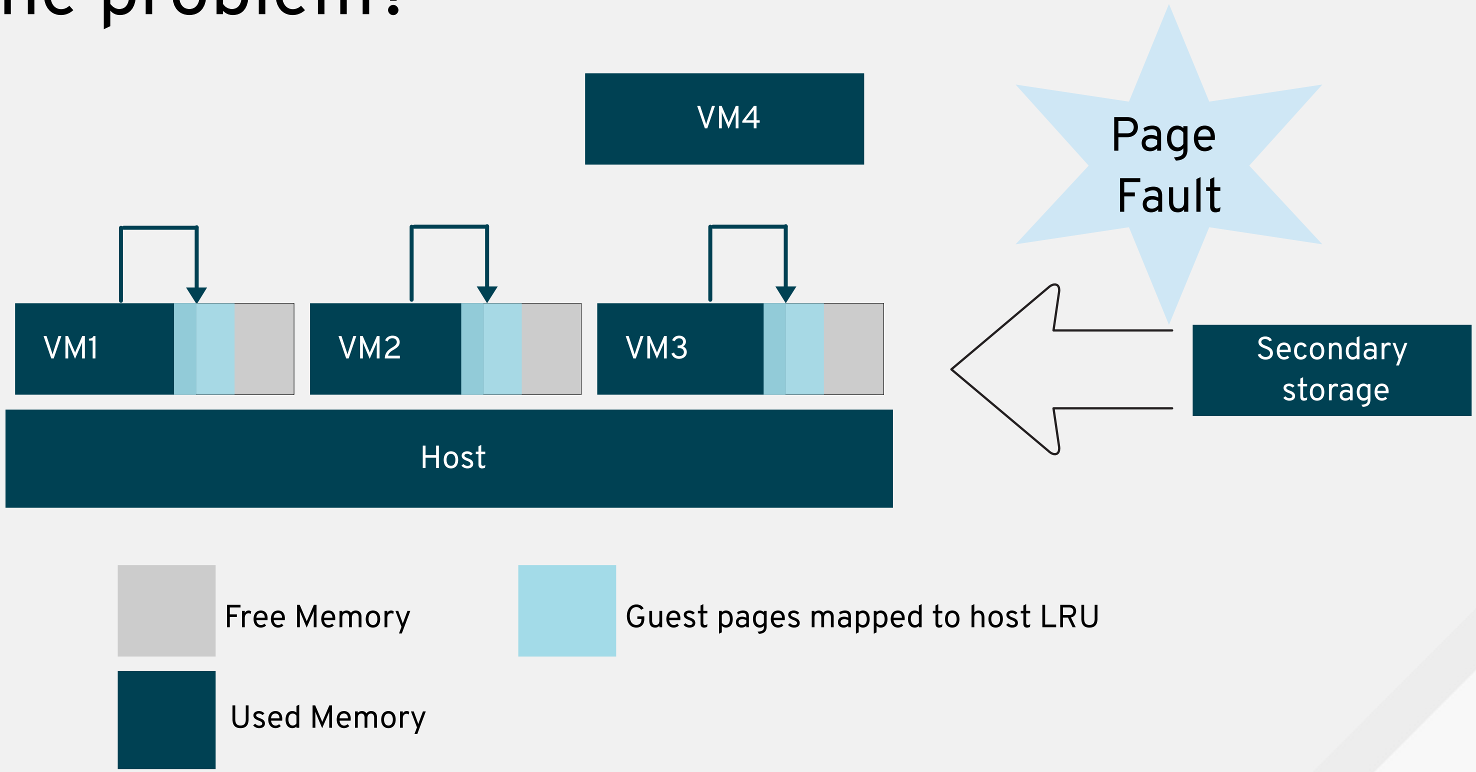
Host

Free Memory

Guest pages mapped to host LRU

Used Memory

What’s the problem?



Issues?

vCPU Stall

Photo by Artem Bali from Pexels

IO is expensive

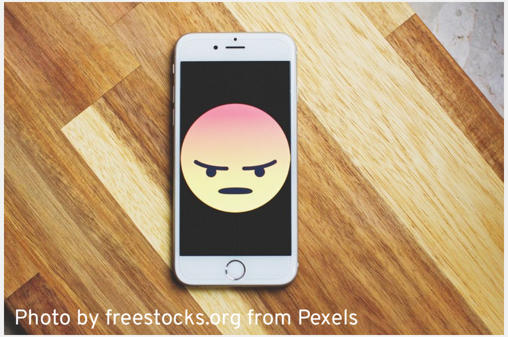


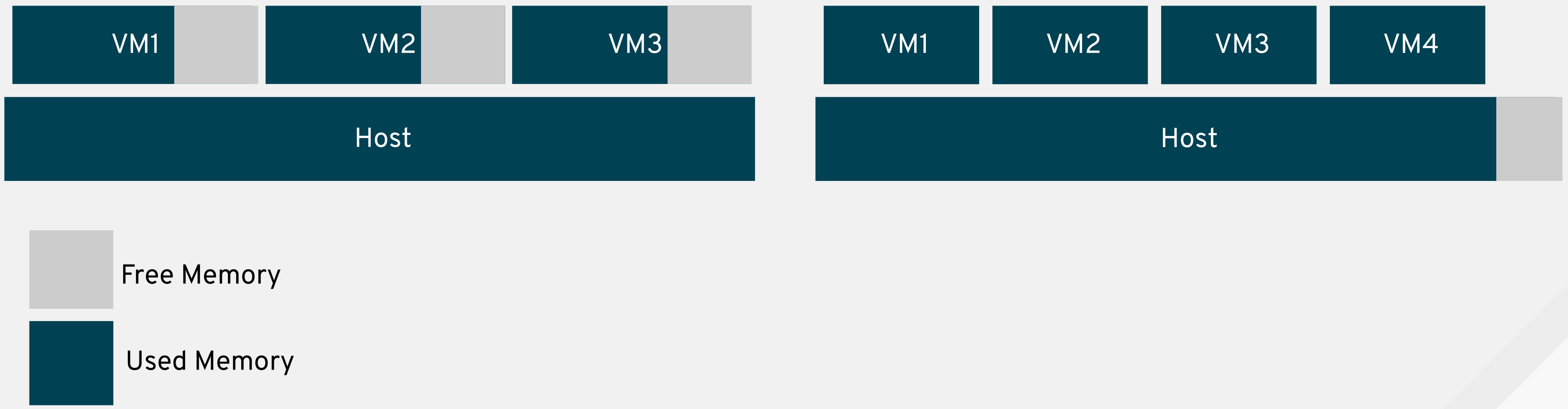
Photo by freestocks.org from Pexels

IO on free pages???

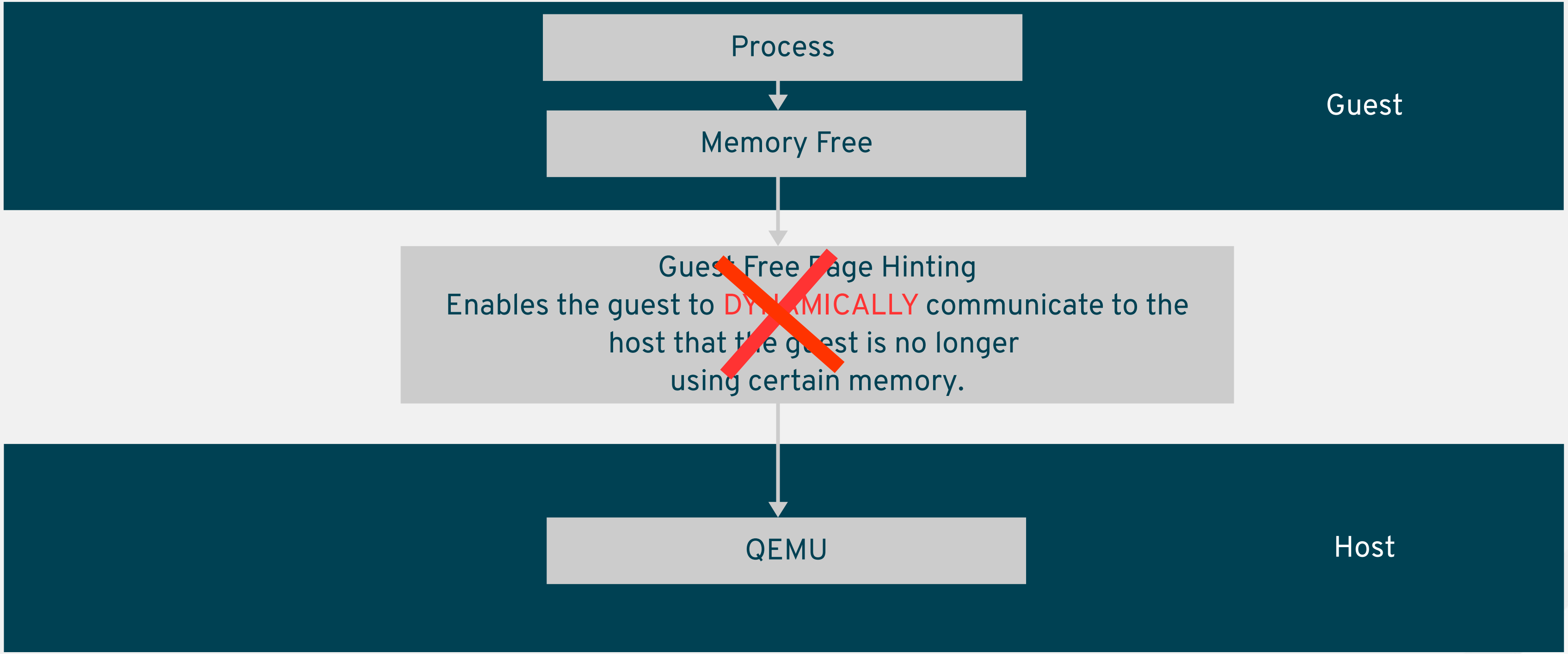
Why don’t you discard its data?

OBJECTIVE

Enhancement in the kernel to fit more virtual machines/process on each system, by removing unused memory from virtual machines.



CURRENT SITUATION



Process

Memory Free

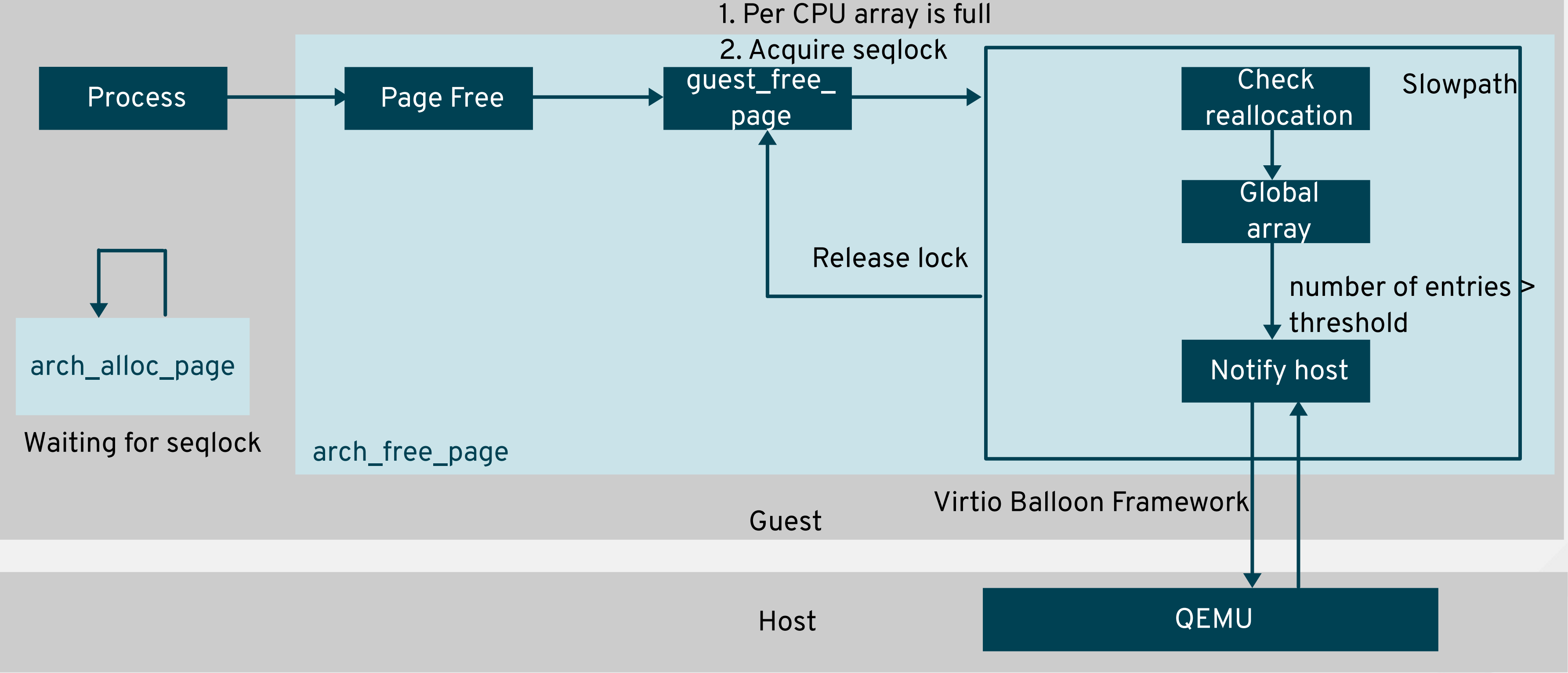
Guest

Guest Free Page Hinting
Enables the guest to DYNAMICALLY communicate to the host that the guest is no longer using certain memory.

QEMU

Host

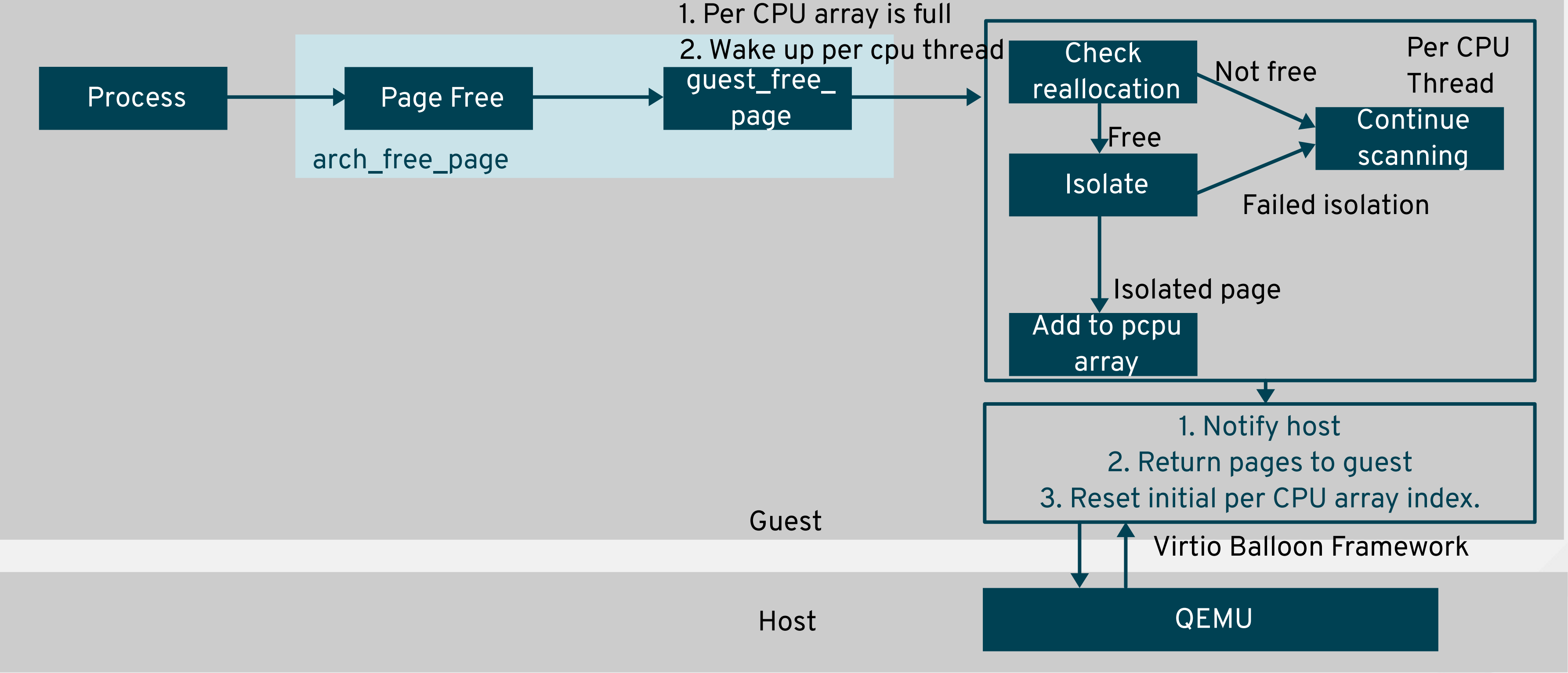
INITIAL IMPLEMENTATION



ISSUES

- Blocks all allocations until the host frees the captured free pages.
    - To avoid reallocation of page which we are sending to the host.
- Blocks arch_free_page() & arch_alloc_page flow.

CURRENT IMPLEMENTATION



BENEFITS

- Still reports the free memory dynamically without human intervention.
- Zone lock is acquired only for the page to be scanned.
  - Other zones are still free to do allocations.
- Doesn’t block the arch_free_page() or arch_alloc_page() flow.

CHALLENGES

- Failing to capture good number of freed pages due to limited array size.
- Failing to isolate a good number of pages.
  - Probably because they are still in Per-CPU Page Frame Cache list and not in buddy.
- Find a good test-case that's real-world and easy to observe in development.
- Analyze and reduce the per CPU thread overhead.

LIMITATIONS

- We depend on buddy free list for freeing a page.
  - If a page is not present in buddy free list we can not isolate.
- Isolation failures under high memory pressure on guest.

FUTURE OPTIMIZATIONS

- Kicking host only with a larger set of isolated pages.
- Sorting per CPU entries based on zone number to avoid repetitive locks on the same zone.
- Making the threshold condition configurable based on the user-case/requirement.

REFERENCES

- Last posted upstream patch-set: https://www.spinics.net/lists/kvm/msg170113.html
- Development Linux repository: https://github.com/niteshnarayanlal/linux-hinting
- Development QEMU repository: https://github.com/niteshnarayanlal/qemu

redhat.

THANK YOU

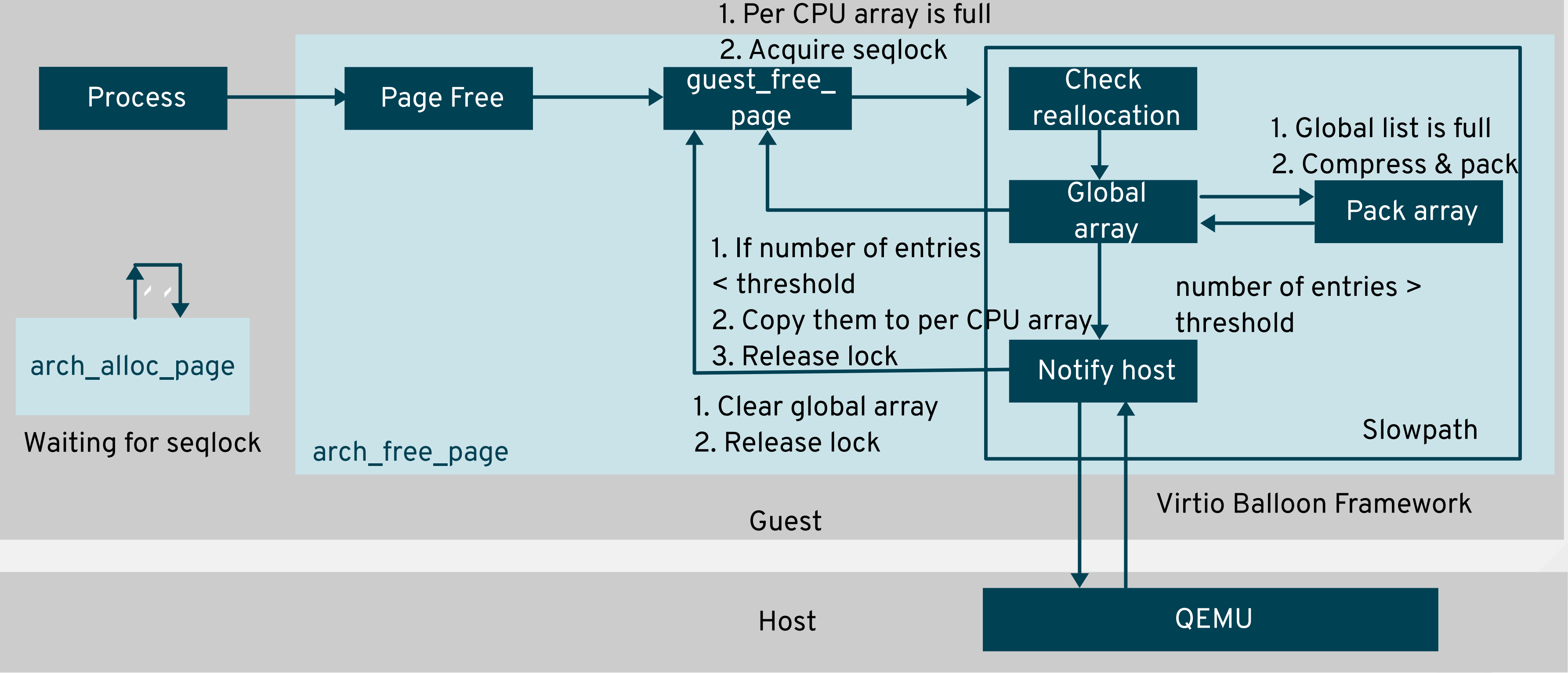
- plus.google.com/+RedHat
- linkedin.com/company/red-hat
- youtube.com/user/RedHatVideos
- facebook.com/redhatinc
- twitter.com/RedHat

Backup Slide

How is this different from “Virtio-balloon: support free page reporting”?

- Similarity?
- Differences?
  - Implementation
  - Use-case
- Will “Guest free page hinting” approach work for migration use case?
  - Can we compare?

INITIAL IMPLEMENTATION



Process
Page Free
guest_free_ page
1. Per CPU array is full
2. Acquire seqlock
Check reallocation
Global array
1. Global list is full
2. Compress & pack
Pack array
1. If number of entries < threshold
2. Copy them to per CPU array
3. Release lock
number of entries > threshold
Notify host
1. Clear global array
2. Release lock
Slowpath
arch_alloc_page
Waiting for seqlock
arch_free_page
Virtio Balloon Framework
Guest
Host
QEMU