HyperAlloc: Efficient VM Memory De/Inflation via Hypervisor-Shared Page-Frame Allocators

The provisioning of the *right* amount of DRAM to virtual machines (VMs) is still a major challenge and cost driver in virtualization settings. Many VMs run applications with highly volatile memory demands, which either leads to massive overprovisioning in low-demand phases or poor QoS in high-demand phases. *Memory hotplugging* and *ballooning* have become established techniques (in Linux/KVM available via *virtio-mem* and *virtio-balloon*) to dynamically de/inflate the physical memory of a VM in a cooperative manner, by having the guests give back unused memory to the hypervisor. However, current VM deflation techniques are either not DMA-safe, preventing the passthrough of important devices like GPUs or NICs, or are not flexible or fast enough to cope with the frequently changing demands of the guest.

We present HyperAlloc, a DMA-safe and extremely efficient mechanism for virtual machine de/inflation. The core idea is to provide the hypervisor direct access to the guest's page-frame allocator, greatly reducing the communication overhead. HyperAlloc can shrink virtual machines 362 times faster than *virtio-balloon* and 10 times faster than *virtio-mem* while having no measurable impact on the guest's performance. HyperAlloc's *automatic reclamation* provides for better memory elasticity by reducing the average memory footprint of a clang compilation by 17 percent compared to *virtio-balloon*'s free-page reporting while, again, having no measurable impact on the guest's performance.

Publication