Memory Overcommit in Containerized Environments

T.J. Alumbaugh (talumbau@google.com)

Yuanchu Xie (yuanchu@google.com)

LSF/MM/BPF 2023

Goal: Optimize memory in overcommitted *containerized* environments

Containers could be virtual machines, K8s containers, applications with memcgs

Clients Use Cases

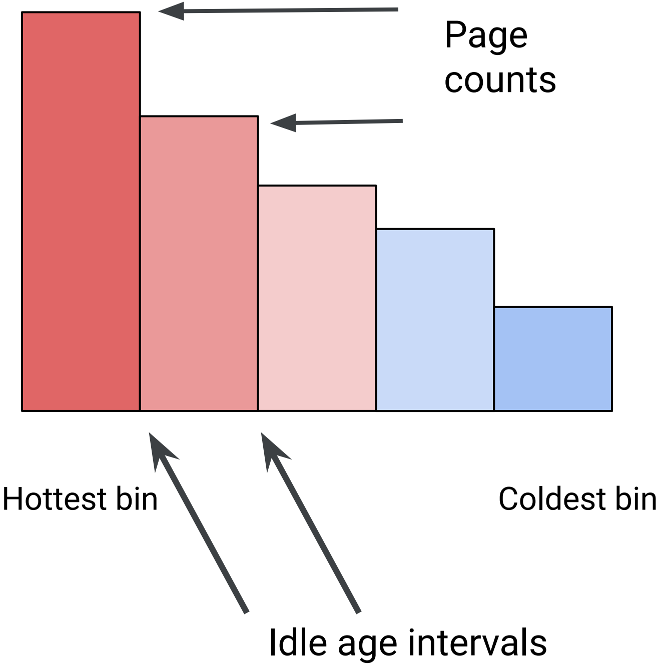
- Virtualized OS on desktops/tablets for device flexibility
- Isolated execution environments for security

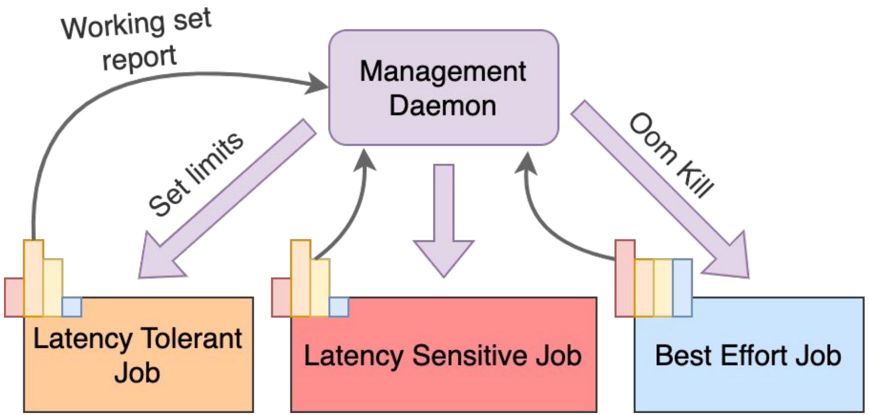
Datacenter servers

- SLO for different availability tiers
- Proactive reclaim
- Demotion/promotion between tiers

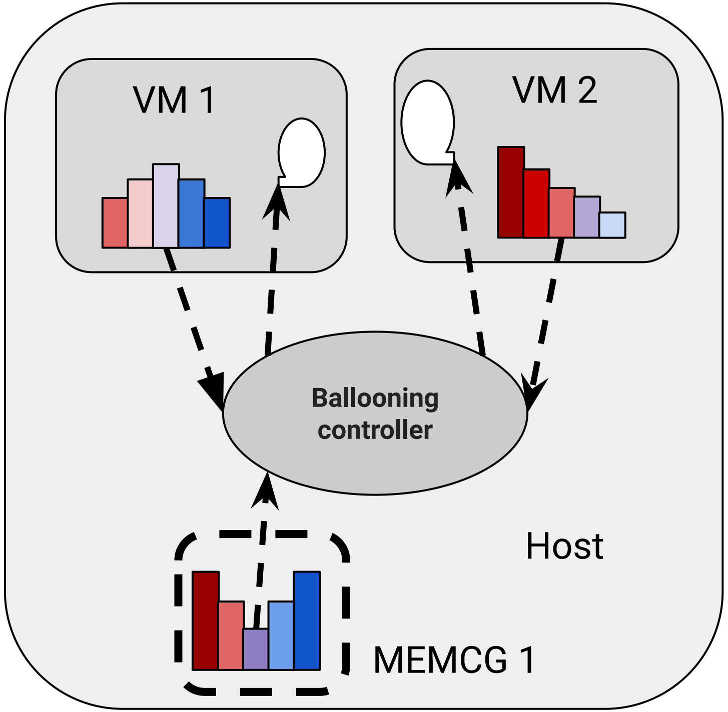
Working Set as a Histogram

- Working set is a binning of pages, by time, or just coldness.
- We ***collect WS in the guest/memcg hierarchy*** for a better estimate of memory utilization inside containers
- Generated on-demand from reclaim activity
- We use the balloon device ***send WS to the host***, which enables the host to make balloon size decisions for each guest





Datacenter Use Case

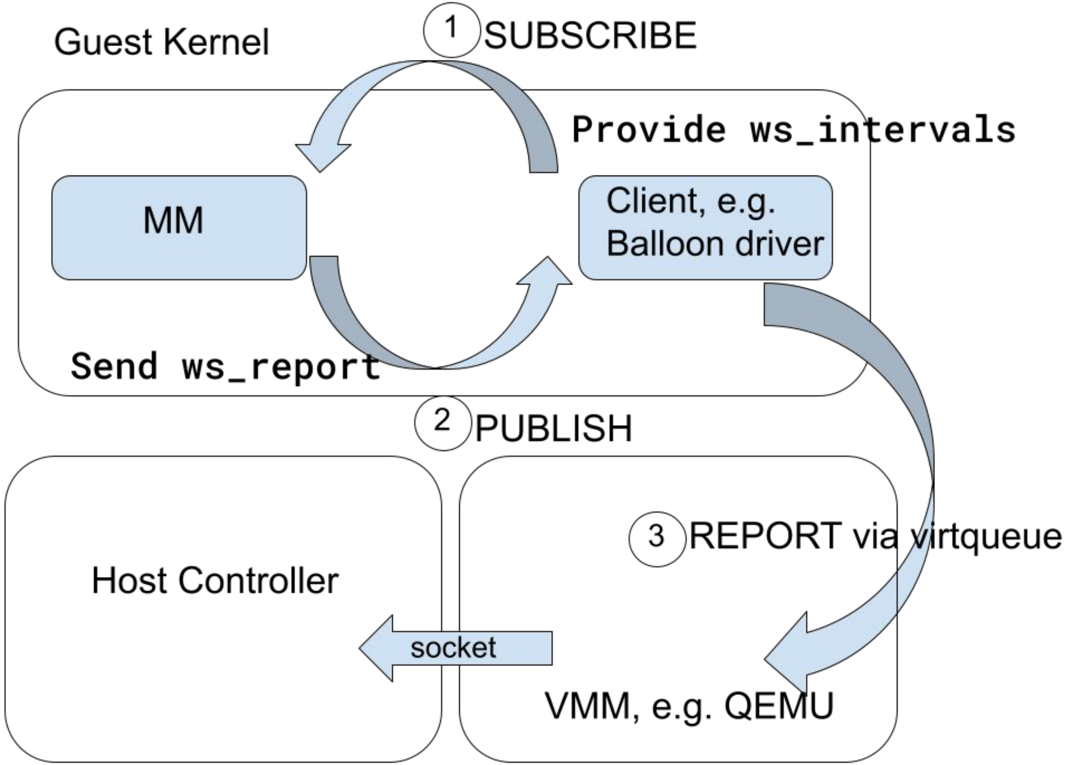


Client Use Case

Getting WS Reports from VMs: WS Reporting

Working Set report notification

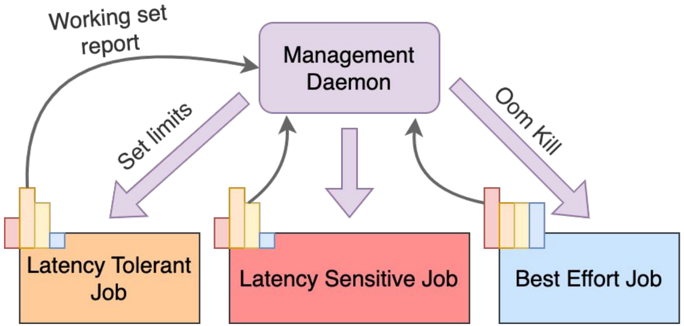
- Clients ***subscribe*** by providing intervals and a WS “receiver” object
- During background reclaim (or on demand) the kernel generates the report and ***publishes*** to the receiver (i.e. the balloon driver)
- The driver ***reports*** the Working Set histogram to the VMM via a virtqueue

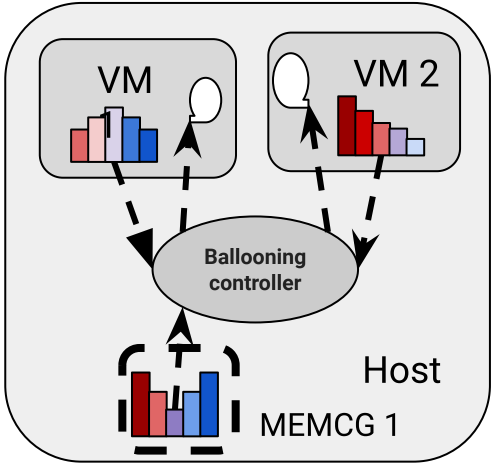


Host controller responsibilities

A host controller receives signals and gives control inputs to the system:

- Receives (and/or queries for) Working Set reports
- Must implement a ***policy*** for memory adjustments.
- Has some notion of ***fairness,*** even if it is implicit.
- Sets memcg limits/balloon size as needed to maintain SLAs
- Can use historical data (past executions, changes in working set, etc) to guide its policy decisions





Code + Additional Resources

- Kernel patch + Balloon Driver patch RFC: **linux-mm@**

- Balloon Device:
  - QEMU implementation RFC: **qemu-devel@**
  - Crosvm implementation: **github.com/google/crosvm**

- VIRTIO Spec Additions: See **virtio-comment@, virtio-dev@**