Binder Enhancements in Oreo

Linux Plumbers Android Microconference
September, 2017

Todd Kjos <tkjos@google.com>

Binder Features added for Oreo

- Multiple Binder Domains
- Scatter-Gather
- Fine-Grained Locking
- RT Priority Inheritence
- Binder Allocator: Security Bugfix
- Binder Allocator: Lazy Free via Shrinker

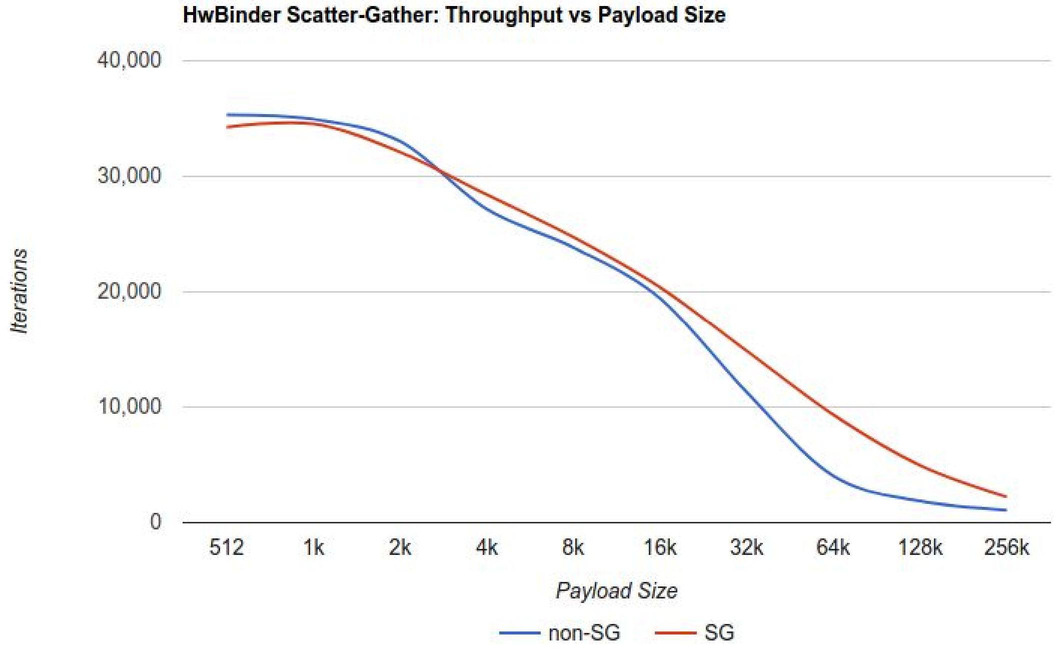
Multiple Binder Domains

- Each domain has its own:
  - Device node (/dev/binder, /dev/hwbinder, …)
  - ServiceManager (service registration and discovery)
- Domains are isolated from each other
  - **binder**: (aka “framework binder”) communication between non-vendor processes
  - **hwbinder**: communication between non-vendor processes and vendor processes (HALs) and between vendor processes that implement HIDL interfaces
  - **vndbinder**: communication between vendor processes that implement AIDL interfaces
- creation of domains are controlled at compile time by CONFIG_ANDROID_BINDER_DEVICES Kconfig option. The three domains listed above are the default and are all required for Oreo

Scatter-Gather

- Normal pattern is to copy data 3 times
  - Serialize into parcel in the calling process
  - Kernel copy to target process
  - Unserialize in the target process
- With scatter-gather, this is reduced to only the kernel copy to target process
  - Currently enabled for HIDL interfaces (hwbinder) only

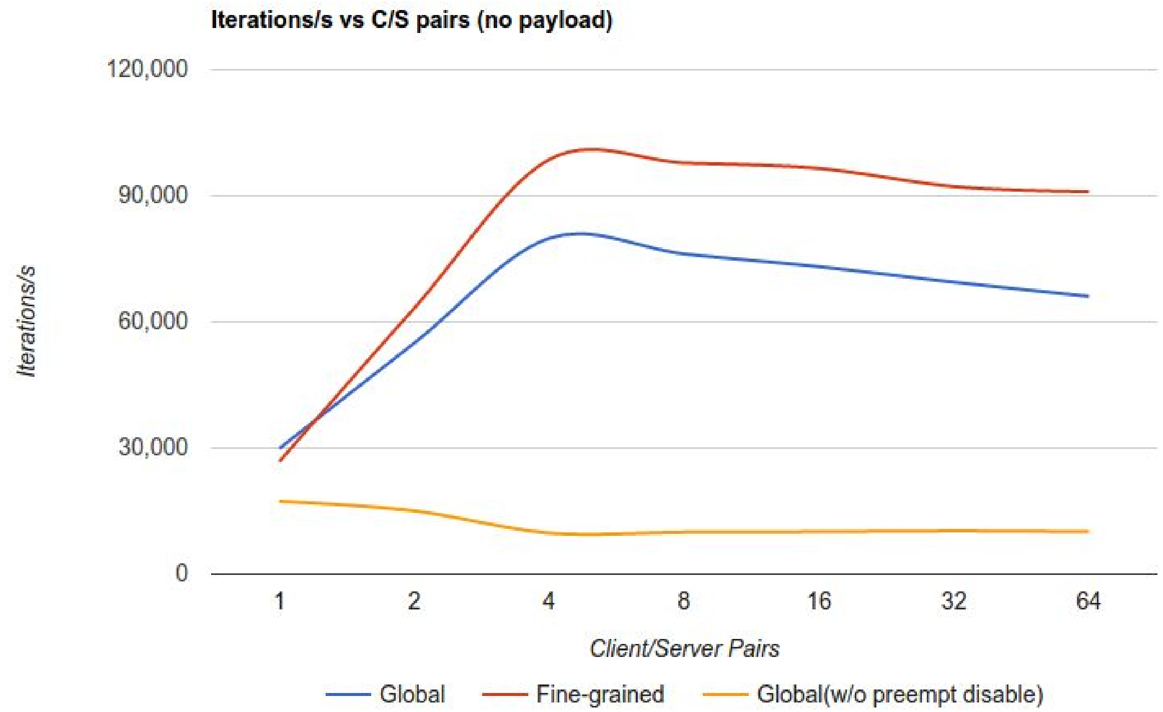
Scatter-Gather Performance



Fine-Grained Locking

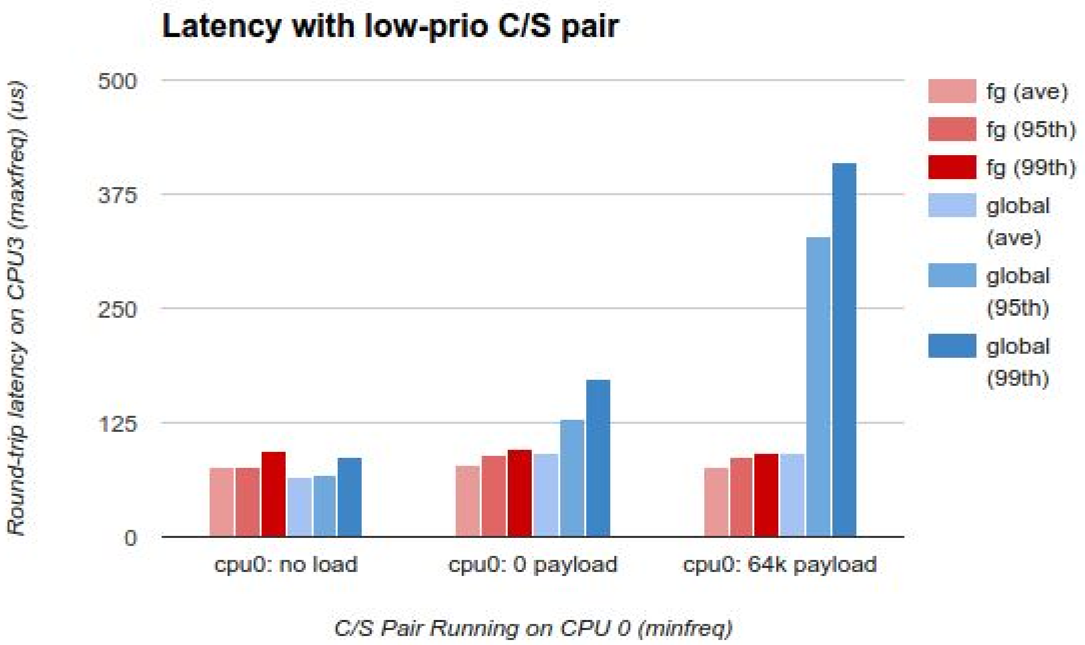
- Used to have single global mutex to protect binder driver state
- Change motivated by priority-inversion cases causing long 95th/99th percentile latencies
  - Contention wasn't really the issue
  - Low-prio task preempted while holding mutex block high-prio tasks
  - Results in long delays inducing in dropped-frames etc
- Since 2015 (Nexus 6p/5x), worked around this by disabling preemption when mutex is held
  - Preemption re-enabled for user data copies, allocations etc
  - It was a hacky, non-upstreamable solution -- but effective
  - upstream binder driver was out-of-date vs what was being shipped
- Moved to fine-grained locking via spinlocks and per-process mutex (instead of global)

Fine-Grained Locking Performance



Fine-Grained Locking Performance

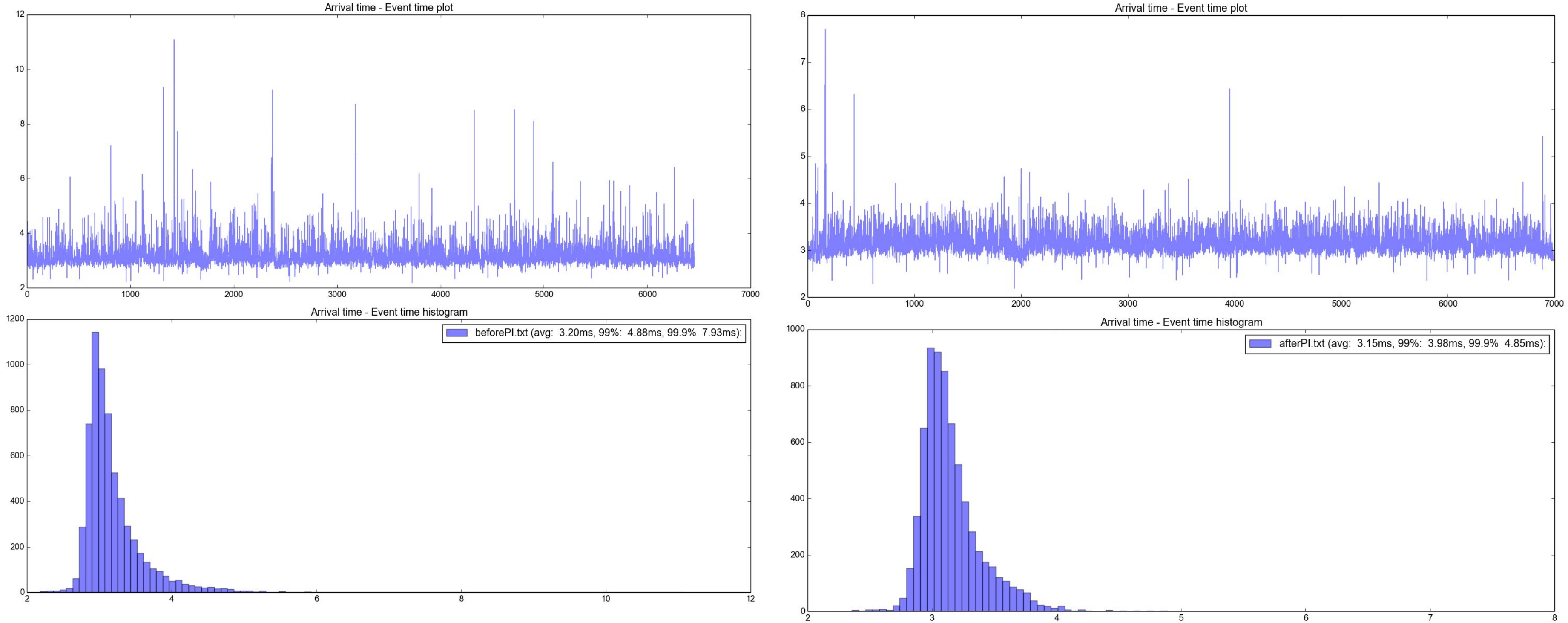
Low-priority load running on cpu0. Measure latency of C/S pair on cpu3



RT Priority Inheritance

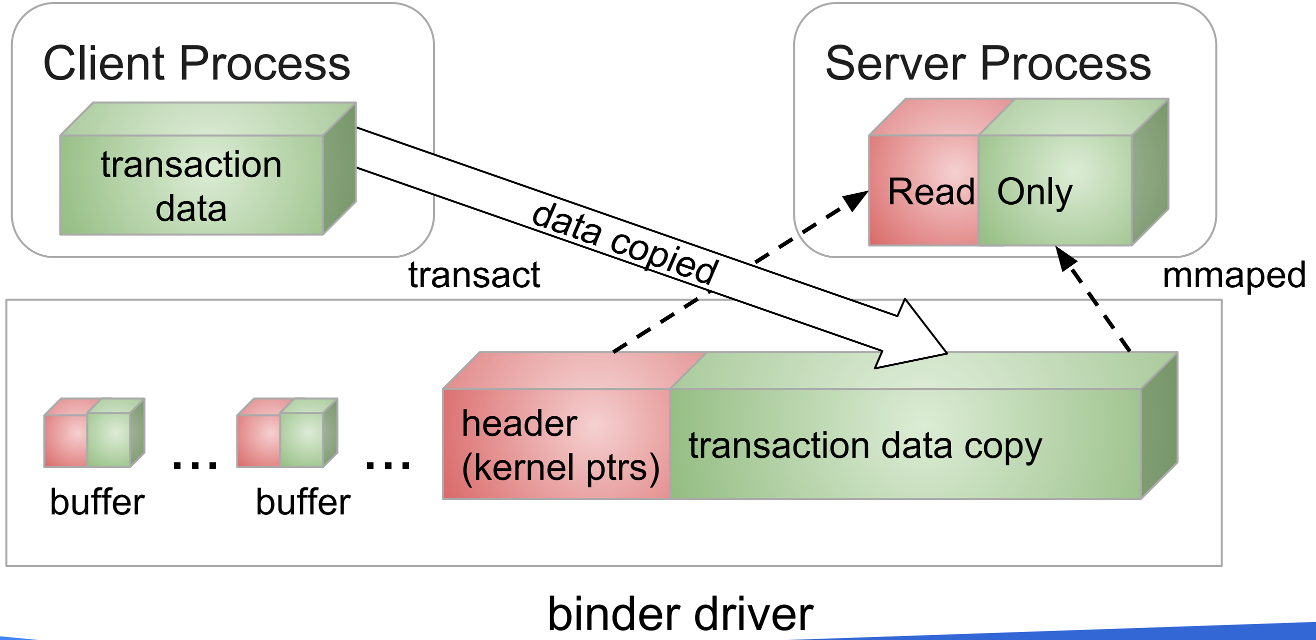
- binder already had *nice* priority inheritence
- Not sufficient with more Android processes running at real-time priority (especially with Treble's *binderized* HALs)
- Binder thread serving an RT client is promoted to appropriate RT sched class + prio
- RT Priority Inheritance can be enabled on a node-by-node basis
    - Currently enabled for *hwbinder*, disabled for framework binder

RT Priority Inheritance Performance



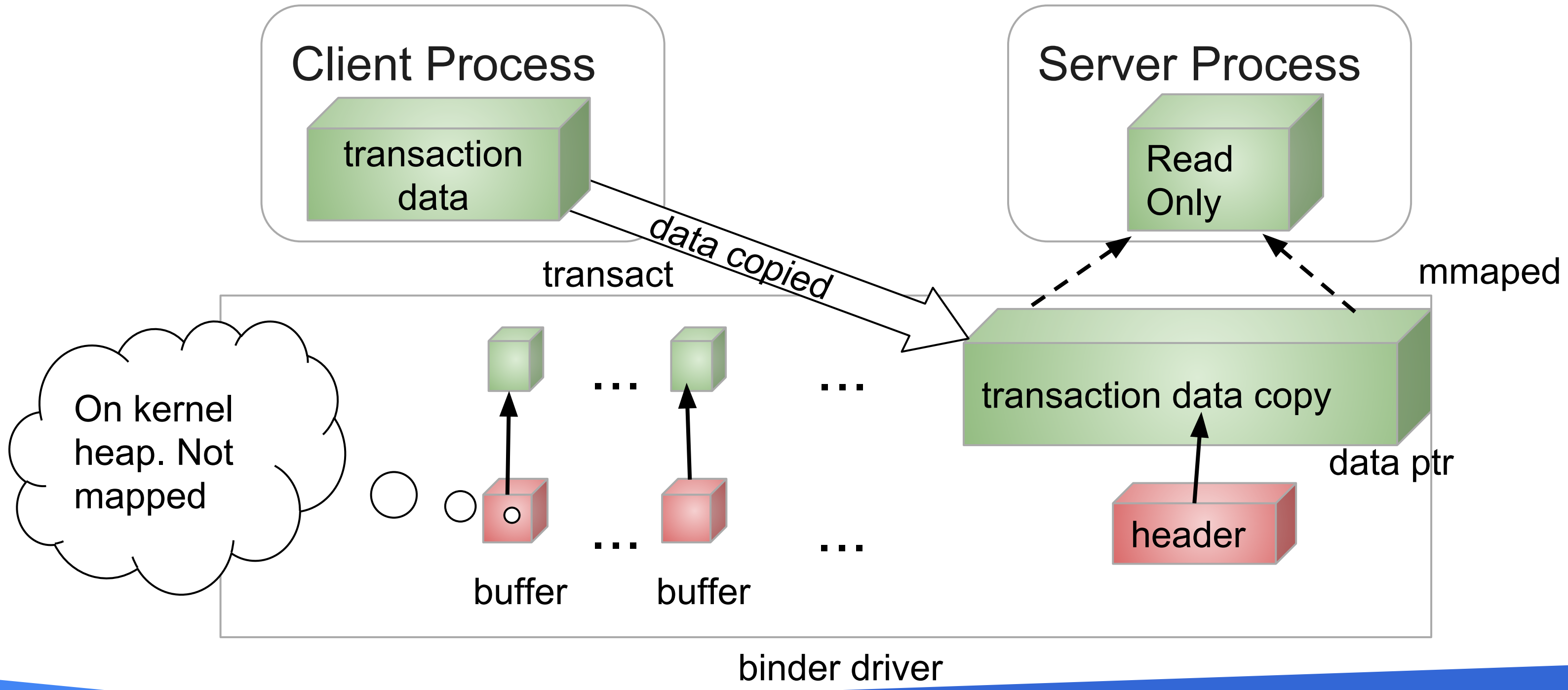
Binder Allocator: Security Bugfix

- Transaction header (containing kernel ptrs) mapped read-only in target user space



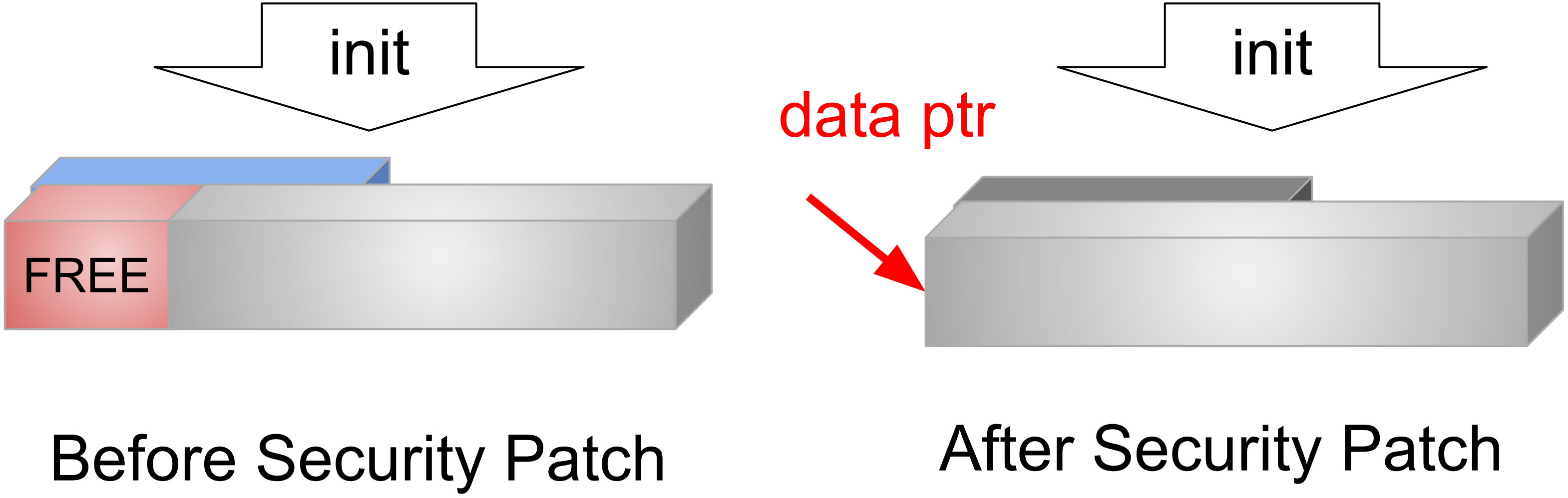
Binder Allocator: Security Bugfix (continued)

- Move buffer header out of shared area -- no longer visible to userspace

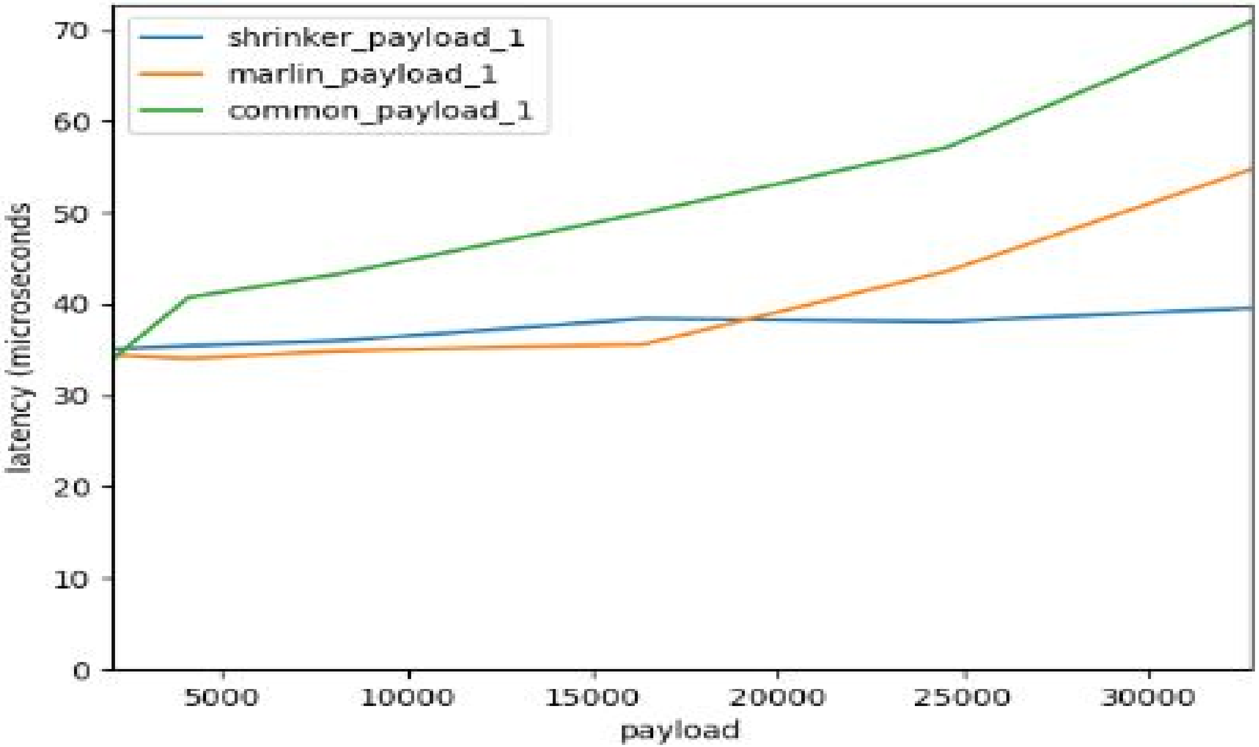


Binder Allocator: Lazy Free via Shrinker

- Problem: Since buffer header is no longer in the mmap'd space, it is freed when the last transaction is complete. Many more allocs/frees
- Solution: Use Linux shrinker to free pages



Binder Allocator: Lazy Free via Shrinker Performance



Q & A