

Compatibility of 4KB Applications on Large Page-Sized Systems

Hardware-Accelerated Per-Process Page Size (P3S) on Native 16KB Linux Kernels

Kalesh Singh, Kelvin Zhang, Frederick Mayle, Jordan Huus
Android Systems & Kernel Team, Google

Linux Plumbers Conference (LPC 2026)

The 16KB Imperative & The Solution Space

Why 16KB Native Kernel?

- **Faster App Startup Latency:** Noticeably quicker cold-start times
- **Less UI Frame Freeze:** Smoother frame rendering and fewer dropped frames.
- **Fewer Low-Memory Kills (LMKs):** Improved background retention.
- **Reduced Reclaim Overhead:** Lower CPU power consumption during memory pressure.

The 4KB Alignment Barrier:

- ELF Rule: `p_vaddr % p_align == p_offset % p_align` (`p_align = 0x1000`)
- Placing code (rx) and data (rw-) in same 16KB page triggers W^X violations or SIGSEGV

Approach	Mechanism	Compatibility	Performance Overhead	Key Drawbacks
Play Store Mandate	Force rebuild with 16KB alignment	Incomplete	0% (Native)	Decades of abandonware/SDKs remain broken.
Bionic Linker Compat	Userspace copy & segment remapping	~70–80%	Minimal memory overhead (uses anon memory for ELF mappings)	Only solve load-time compatibility issues; not runtime issues.
Hypervisor / MicroVM	Run 4KB guest OS in pKVM	100%	High CPU & RAM	Heavy RAM footprint, boot latency, GPU IPC hurdles.
Hardware P3S (Our Work)	Asymmetric MMU translation (TG0=4KB)	~100%	~0% (Native)	Subpage physical slicing in kernel MM.

Core Philosophy & ARMv8-A MMU Dual Regimes

The Core Axiom of P3S:

- Prioritize functional correctness over optimizations for legacy 4KB software.
- Ensure zero overhead for native 16KB software.

ARMv8-A Asymmetric Translation Regimes:

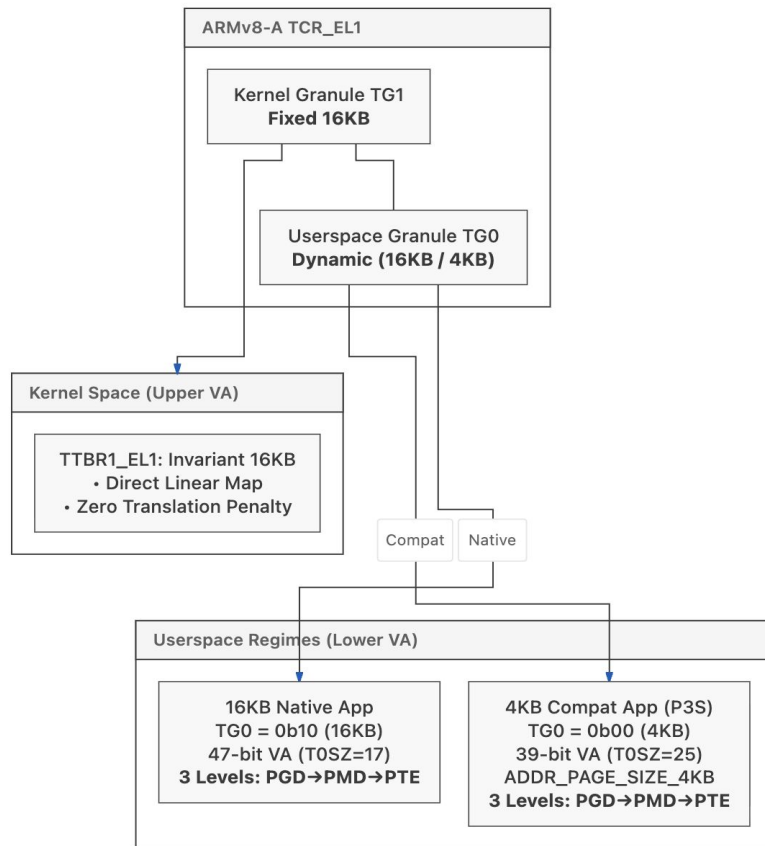
- Upper VA (TTBR1_EL1): Kernel space. Translation granule TG1 statically pinned to 16KB. Zero overhead for kernel code, slab allocators, and drivers.
- Lower VA (TTBR0_EL1): Userspace. Translation granule TG0 dynamically toggled: 16KB (0b10) for native apps; 4KB (0b00) for compat apps.

Process Activation & Personality:

- Flag: ADDR_PAGE_SIZE_4KB assigned during sys_execve()
- Children strictly inherit parent's translation geometry across fork()/clone()

Unified 3-Level Hardware Page Table Folding:

- 16KB Native: 47-bit VA (T0SZ=17) -> 3 Levels: PGD -> PMD -> PTE.
- 4KB Compat: 39-bit VA (T0SZ=25, 512GB) -> 3 Levels: PGD -> PMD -> PTE.
- Result: Zero branch divergence in generic kernel page table walkers



Kernel Decoupling & Header Redirection Architecture

Decoupling Compile-Time Constants:

- Linux historically treats PAGE_SIZE as build-time macro (CONFIG_ARM64_16K_PAGES).
- P3S introduces dynamic geometry tracking per address space: mm_struct->page_shift (12 vs. 14).
- Linear Map Invariance: Direct map macros remain pinned to PAGE_SHIFT_KERNEL (14).

Scoped task_struct->p3s_remote_mm:

- Dedicated pointer in task_struct initialized to NULL on fork.
- Remote-MM operations (ptrace, /proc/pid/mem, OOM, MGLRU lookaround) use RAII guard P3S_CONTEXT_REMOTE_MM.
- Automatic restoration on scope exit via DEFINE_CLASS destructor.

Build Integrity & Test Coverage:

- Runtime assertions (p3s_assert_mm) verify MM geometry.
- Guarded by scripts/check-p3s-include.sh.
- Boot toggle via p3s=on|off command-line parameter for CI testing.

Header Redirection & asm/p3s.h:

```
/* arch/arm64/include/asm/p3s.h */
static __always_inline struct mm_struct
*p3s_current_mm(void)
{
    if (likely(current))
        return unlikely(current->p3s_remote_mm) ?
            current->p3s_remote_mm : current->mm;
    return NULL;
}

/* include/linux/p3s_user_pages.h */
#define PAGE_SHIFT    mm_pte_shift(p3s_current_mm())
#define PAGE_SIZE     (1UL << PAGE_SHIFT)

/* Scoped RAII remote MM guard (linux/cleanup.h) */
#define P3S_CONTEXT_REMOTE_MM(remote_mm) \
    CLASS(p3s_remote_mm, \
        __UNIQUE_ID(p3s_rmm))(remote_mm)
```

Subpage Slicing & VMA Architecture (vm_slice_off)

VMA Subpage Slice Tracking (vm_slice_off):

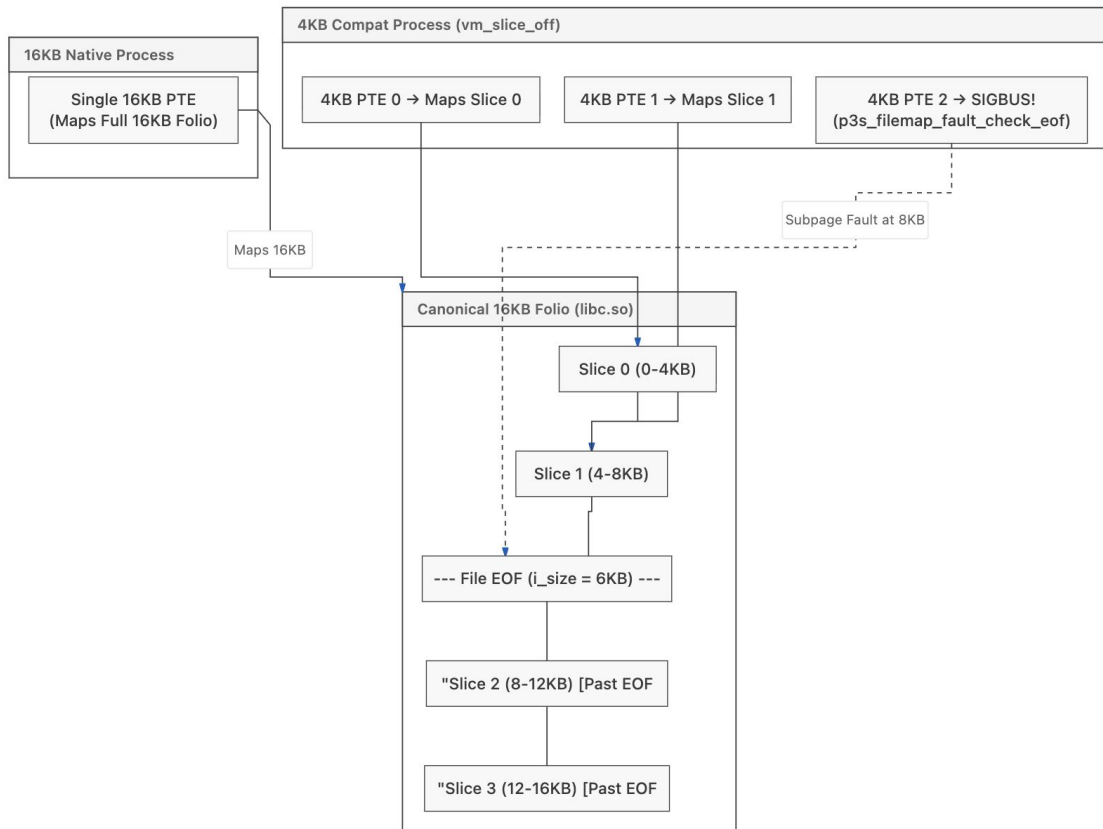
- Adds `vm_slice_off` to struct `vm_area_struct` to track 4KB slice offset within host 16KB folio.
- Fully preserved across VMA split, expansion, merge, and `mremap` (enforcing 16KB slice alignment on moves).
- VMA interval tree is indexed by host folio span

Filemap Integration & EOF Protection:

- Page cache canonically maintains all file data as 16KB folios (zero cache aliasing).
- Native 16KB processes map 1x 16KB PTE; 4KB compat processes map 4x 4KB PTEs to Slices 0..3.
- EOF Protection: `p3s_filemap_fault_check_eof()` detects subpage faults past `i_size` within final 16KB folio and raises `SIGBUS`.
- `vma_folio_slice_bounds()` clamps batch mapping to folio, VMA, and page table boundaries.

Anonymous Faulting & Eager-Copy CoW:

- Allocates 16KB host folio on first anon fault and eager-populates adjacent slices.
- Eager CoW copies full 16KB folio upon write fault, preventing slice fragmentation and fault storms.



Subsystem Integration: Deliberate Policy & Gating

Disable or bypass subsystems that introduce multi-granule complexity or locking hazards; adapt subsystems essential for memory reclamation, process management, and security.

Subsystem	Policy for 4KB Compat	Architectural Rationale & Implementation
THP & hugetlbfs	Gated / Rejected	PMD collapse conflicts with 3-level folded walker geometry; hugetlb operates only on native 16KB.
KSM (Samepage Merging)	Gated / Disabled	Deduplicating subpage slices across mismatched folio hosts is disallowed.
userfaultfd	Non-Anon Gated	Shmem/hugetlb uffd gated to prevent folio lock deadlocks; anonymous uffd fully supported (4KB slice copy/move).
contpte & Fast GUP	Bypassed / Locked	contpte coalescing bypassed; lockless fast GUP gated to lock to prevent race conditions during subpage walks.
ARM64 MTE	Adapted	Clears Memory Tagging Extension tags at granular 4KB slice boundaries.
Swap & Migration	Adapted	Preserves 4KB subpage slice offsets across swap entries and page migration.
/proc Interfaces	Adapted	maps, smaps, and pagemap accurately scaled to MM page size.

Android Userspace: Bifurcated Zygote & Self-Healing

The Fork-Without-Exec Dilemma:

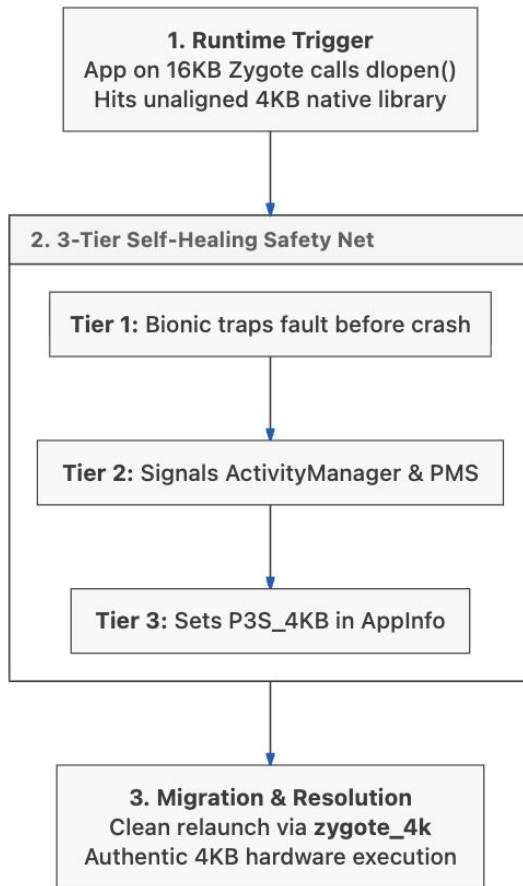
- Android apps do not call `execve()` - they fork from Zygote to share preloaded ART runtime classes and achieve sub-100ms warm starts.
- MMU Constraint: Translation granule (TCR_EL1.TG0) is permanently bound at `execve()`; cannot mutate running process granule.

The Dedicated `zygote_4k` Service:

- Launched via `execve()` with `ADDR_PAGE_SIZE_4KB`, operating with authentic `TG0 = 4KB`.
- On-Demand Lifecycle: Starts when first 4KB app launches; tears down when idle to recover RAM.
- Secondary zygotes: `app_zygote_4k` (isolated processes) and `webview_zygote_4k` (Chromium).

3-Tier Dynamic Self-Healing Loop:

- Tier 1 (Linker Trap): Bionic dynamic linker intercepts 4KB segment misalignment during `dlopen()` (dynamic plugins, feature splits)
- Tier 2 (Framework Signal): Linker traps condition and notifies ActivityManager
- Tier 3: Sets `P3S_4KB` in `AppInfo` (PackageManager flags package; app cleanly relaunches via `zygote_4k`)



Mixed-Mode Challenges

Real Hardware: PSCI Identity Map Mismatch

- **Symptom:** Level-1 Translation Fault (ESR 0x96000005) in `__cpu_replace_ttbr1()` during CPU idle exit
- **Root Cause:** Kernel identity map (`idmap_pg_dir`) is compiled in native 16KB format. When waking up from PSCI suspend from a 4KB process, `cpu_do_resume()` restored `TG0 = 4KB`. Activating the 16KB identity map under `TG0=4KB` caused an immediate hardware translation fault.
- **Fix:** Enforce `TG0 = 16KB` before activating `idmap_pg_dir` during CPU idle resume.

Subsystem Challenge	Mixed-Mode Dynamics	Current Implementation & Upstream Path
Reverse Mapping (rmap)	Walking 16KB folios referenced by mixed 16KB and 4KB PTEs	<code>page_vma_mapped()</code> matches 4KB subpage PTEs in 16KB folios; tracking subpage mapcounts.
Dirty Tracking & Writeback	4KB PTE dirtying vs. 16KB folio writeback	Aggregates subpage dirty state; evaluating subpage-granular writeback to reduce flash write amplification.
Compaction & Migration	Moving a 16KB folio with up to four 4KB PTEs across PIDs	Batched multi-slice atomic unmap; <code>migrate_page()</code> preserves <code>vm_slice_off</code> .
Direct I/O & Fast GUP	DMA pinning of individual subpages	Subpage slice pin tracking; lockless fast GUP gated to avoid race conditions.

Validation Status, Upstream Roadmap & Q&A

Platform Tier	Target / Branch	Current Status	Test Scope & Verification Coverage
Linux Kernel	android17-6.18-p3s (topic:p3s_v1)	Patches Posted	TCR switching, 3-level folded walkers, vm_slice_off, p3s_remote_mm RAIL context.
Virtual CI	aosp_cf_arm64_phone_p3s	Automated CI	16KB kernel + 4KB rootfs boot, smoke, stress, and reboot test suites.
MM Selftests	tools/testing/selftests/mm	Adapted & Passing	Memory management selftests adapted for 4KB compat page sizes.
Framework	Android 17 Platform	In Progress	PMS scanning, ProcessList routing, on-demand zygote_4k lifecycle.
Apps	Top Games	Completed	Top Games functional in P3S mode.

Summary & Takeaways:

- **Zero Native Overhead:** Native 16KB apps retain full performance gains (faster startup, less frame freeze, fewer LMKs)
- **100% Ecosystem Compatibility:** Legacy 4KB software executes seamlessly via authentic ARMv8-A MMU translation (TG0 = 4KB)

Q&A & Discussion:

- **Gerrit Topic:** https://r.android.com/q/topic:p3s_v1