

The 0.5 Second App Launch: Unpacking Zygotote & Copy-on-Write

The first pillar of Android's high-performance architecture is in how its processes are born.



Esmaeel Moradi
@esmaeel-moradi700

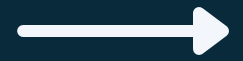


The 0.5 Second App Launch: Unpacking Zygoter & Copy-on-Write

The first pillar of Android's high-performance architecture is in how its processes are born.



Esmaeiel Moradi
@esmaeiel-moradi700



The Problem and the Parent Process

Challenge:

Launching a new Java process is SLOW.
It requires loading the ART (Android Runtime) and thousands of framework classes.

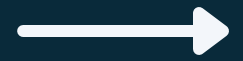
Solution:

The Zygote Process

- **Zygote is the initial Java process started at boot.**
- **It pre-loads all common, heavy resources and framework code into its RAM.**
- **Zygote acts as the "Mother Process," ready to clone itself instantly.**



Esmaeeil Moradi
@esmaeeil-moradi700



fork:

The Instant Clone

When a New App (App A) is Requested:

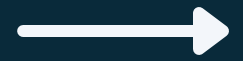
1. Zygote receives a request from AMS.
2. It executes the Linux `fork()` system call.
3. The Kernel creates a new process (App A) that is an exact replica of Zygote—including all the pre-loaded code.

The Key:

At this stage, App A and Zygote share the same physical memory.



Esmaeel Moradi
@esmaeeil-moradi700



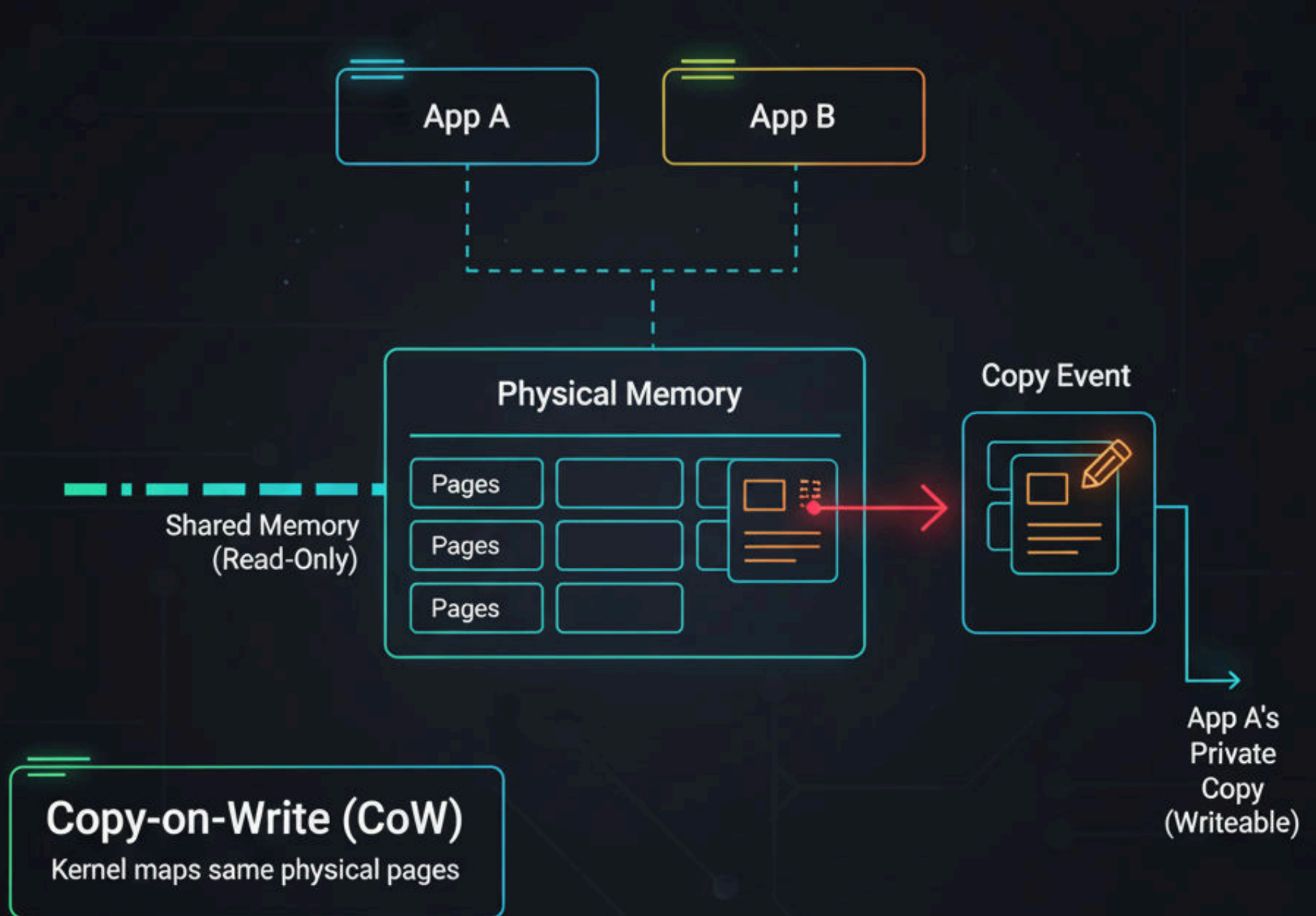
The Magic: Copy-on-Write (CoW)

How Memory Sharing Works:

- **CoW Rule:** The Kernel does not copy the memory pages. It maps the same physical pages to both processes' Virtual Address Spaces.
- **When is the Copy Made?** Only when one of the processes (say App A) tries to **WRITE** to a shared page.
- **Result:** App A gets all the core framework code for free and only needs unique memory for its own data.



Esmaeel Moradi
@esmaeeil-moradi700



Esmaeel Moradi
@esmaeeil-moradi700



Performance & Memory Trade-off

Zygote + CoW = High-Performance Android

- 1. Speed: Achieves Warm Start times in milliseconds, significantly improving User Experience.**
- 2. Efficiency: Maximizes RAM usage by sharing the heaviest libraries across 100s of processes.**



Esmaeeil Moradi
@esmaeeil-moradi700

✦ How do these processes talk to each other securely?

We dive into the Binder IPC and its Single-Copy mechanism.
(Follow the thread!)

Share your expertise below! 📵



Esmaeeil Moradi
@esmaeeil-moradi700