



Contribution ID: 303

Type: not specified

Bootloader-Based S2D(suspend to disk) Mechanism on mobile

In an increasing number of scenarios, the use of S2D (Suspend to Disk) functionality is required or expected on mobile. For example, when a mobile is running low on battery, it can use the S2D suspend process to save the user's current state and then enter a power-down mode. Once the battery level is sufficient again, the device can quickly resume the previous state through the S2D resume process. Additionally, some mobile devices may aim to support multiple operating systems, and S2D can help quickly restore the last state of a system when switching between them.

The Linux kernel already supports S2D, but there are still some limitations and impacts when used in real-world scenarios. For example, storage-related drivers need to be loaded early during the Linux kernel boot process in order to restore the suspend image from storage. However, in practice, due to system design constraints, storage drivers may be loaded relatively late, making it impossible to restore the suspend image. Another, restoring a suspended image using the Linux kernel requires booting the kernel to a specific stage where the environment is sufficient to perform the restore. Additionally, the image must be restored page by page, which may result in failing to meet system boot KPI requirements.

To address these issues, we designed a bootloader-side restore mechanism for the suspended image. The basic idea is the Linux kernel completes the system suspend and saves the image to storage during the S2D process. Then, upon a cold reboot, the bootloader is responsible for restoring the S2D image. During this process, the image data can be loaded in blocks rather than page by page, thereby maximizing system boot KPI performance and avoiding issues related to delayed loading of storage drivers.

Primary author: FAN, Tengfei

Presenter: FAN, Tengfei

Session Classification: Android MC

Track Classification: Android MC