



SD SWAP Note



Outline

- Preface
- Android default solution
- MTK SD-SWAP solution

Outline

- **Preface**
- Android default solution
- MTK SD-SWAP solution

Preface

- Android ICS開始支援multiple-storage。
- MTK亦跟進支援, 增加了內置SD及外置SD。
- 但Multiple Storage本身具備著使用上的限制, 故MTK為因應這些限制而提出了數個解決方案。
- 本文件專注於其中的SD-SWAP solution, 其他的solution請參閱DMS。

Outline

- Preface
- **Android default solution**
- MTK SD-SWAP solution

Android原生Multi-Storage-- 行為

- 在ICS版, android 本身已支援multi-storage. (GB版本不支援)
 - 行為 in default MTK mode
 - /mnt/sdcard -> 內置sd
 - /mnt/sdcard2 -> 外部sd (如果有外部sd)
 - 對於APP在空間上的使用可以分兩部分來看
 - Code的部分 (通常都<10M)
 - 大部分都安裝在/data下. 如果該APP支援app2sd的話, APP2SD就能幫我們把APP安裝在storage_list.xml中指定為“Primary”的storage之上
 - 在執行時會去存取的資料
 - android 內建APP (camera, gallery, music...) ,或是market上的APP, 仍然只能存取特定的storage, 一般而言會有兩種存取方式
 1. 透過 getExternalStorageDirectory()
 2. 有些APP甚至不透過getExternalStorageDirectory(), 而將路徑寫死在 /mnt/sdcard
 - APP所存取的資料可能很大, 例如, 導航軟體的圖資(> 500M)

Android原生Multi-Storage --限制

- 雖然Android支援multi-storage, i.e./mnt下可以放複數個dir並掛載多個storage, 但對於大部分的APP而言, 還是只會去某一個特定的storage存取data. (目前Android沒有一個公定的api可以知道其他storage mount在哪個目錄)
- 這也代表, 即便Android提供了Multi-Storage, 但大部分的APP仍然只會存取到一個storage, i.e. /mnt/sdcard底下掛載的裝置。所以APP執行時能存取的data的空間就會與/mnt/sdcard的容量有很大的關聯。
- MTK的/mnt/sdcard底下掛載eMMC Storage, 約莫2GB的空間, 空間不大很容易用完。
- 反之/mnt/sdcard2掛載SD Card, 空間由user插入的card size決定, 一般而言皆大於2GB。外置T卡可用空間大許多, 對於使用上較為方便。
- 為此MTK提出了“ **SWAP**”, 將/mnt/sdcard改掛載user插入的外置T卡, 而內置卡改掛載於/mnt/sdcard2
 - 無SWAP
 - /mnt/sdcard → internal SD
 - /mnt/sdcard2 → external SD
 - 有SWAP
 - /mnt/sdcard → external SD
 - /mnt/sdcard2 → internal SD

Outline

- Preface
- Android default solution
- **MTK SD-SWAP solution**

MTK的Multi-Storage 方案

- 在android的架構下, 做了些修改, 目前MTK針對multi-storage有底下方案
 - [方案1] MTK ICS official MP
 - [方案2] 外部sd優先(GB時期提出的方案, MTK繼續沿用於ICS)
 - [方案3] 完全無內置sd (S/H皆採類似此方案)
- 此份文件主要探討的是[方案2], i.e.MTK內部常稱呼的“ **SD SWAP**” 方案
- 要注意的是, 此方案行為較複雜, 同時存在著一些限制, 為了造成使用上的誤會因此盡量詳細描述, 還請仔細閱讀以避免自身及客戶使用上的問題, 也請確定客戶能夠接受所有的使用限制。

Outline

- Preface
- Android default solution
- MTK SD-SWAP solution
 - Spec
 - Limitation
 - Hot to enable it?
 - Related Commands
 - Misc

Outline

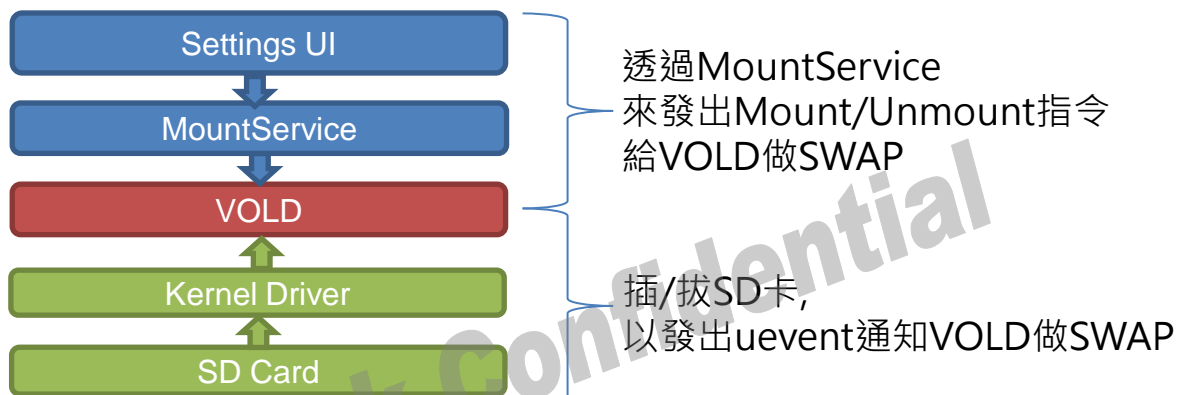
- Preface
- Android default solution
- MTK SD-SWAP solution
 - **Spec**
 - Limitation
 - Hot to enable it?
 - Related Commands
 - Misc

[方案2]外部sd優先-- Spec

- 定義“ SWAP” : SWAP發生, 即代表SD mount point交換, 反之則是沒發生SD SWAP
 - /mnt/sdcard : 掛載External SD
 - /mnt/sdcard2 : 掛載Internal SD
- 這個feature option是必須在compile time決定是否enable, UI/command line不會有任何可以讓user選擇去動態enable SD SWAP feature option的方式。
- 欲使用SWAP, 請先確定手機支援兩張卡, 其中一張是Phone Storage(內置卡), 而另一張則是外置T卡。
- 支援動態SWAP。
- APP僅允許存放於/data或者是外置卡
 - 外置SD卡沒有掛載時, 只允許應用安裝在/data, 不允許利用APP2SD安裝在Phone-Storage
- 保留 “設置/存儲/默認存儲位置” 選項

[方案2]外部sd優先 – Dynamic SWAP(1/2)

- 為增加使用上的彈性及便利性, 此方案支援動態SWAP SD卡。
- 動態SWAP不是指“可以透過UI or Command line讓原本不支援SWAP的手機變成可支援”。
- 動態SWAP是指使用者插/拔/Mount/Unmount外置卡過後, 內/外置卡的mount point會產生相對應的SWAP行為。
- 對SD SWAP solution而言, VOLD daemon會負責執行SWAP的動作, 在我們的平台上, VOLD只會在兩種狀況下執行SD SWAP:
 - 使用者透過Settings UI去mount/unmount外置卡
 - 使用者去熱插拔SD卡
 - 在上述兩個情況裡, SD mount path會跟著變化(可見次頁整理)



[方案2]外部sd優先 – Dynamic SWAP(2/2)

- 歸納在開啟SD SWAP solution的手機上, 會出現的各種情況：

行為	行為發生前的 /mnt/sdcard	行為發生前的 /mnt/sdcard2	行為發生後的 /mnt/sdcard	行為發生後的 /mnt/sdcard2	APP2SD
沒插SD卡開機	無掛載	無掛載	掛載內置卡	無掛載	No
插SD卡開機	無掛載	無掛載	掛載外置卡	掛載內置卡	Yes
UI點選unmount 外置卡	掛載外置卡	掛載內置卡	掛載內置卡	無掛載	No
UI點選mount 外置卡	掛載內置卡	無掛載	掛載外置卡	掛載內置卡	Yes
拔出外置卡	掛載外置卡	掛載內置卡	掛載內置卡	無掛載	No
插入外置卡	掛載內置卡	無掛載	掛載外置卡	掛載內置卡	Yes

Outline

- Preface
- Android default solution
- MTK SD-SWAP solution
 - Spec
 - **Limitation**
 - Hot to enable it?
 - Related Commands
 - Misc

[方案2]外部sd優先--限制(1/3)

- 這個feature最重要的一個改變就在於SWAP之後, " /mnt/sdcard" 掛載的裝置是**可被移除**的外置SD Card。以下的討論皆針對此限制加以探討。
- [限制一] **SD SWAP會影響到所有會去access /mnt/sdcard or /mnt/sdcard2的APP或tool**
 - 有的APP/TOOL owner認為外置卡一定是掛載在/mnt/sdcard2, 因此在code裡寫死路徑, 固定去存取/mnt/sdcard2, 但在SD-SWAP solution裡, 如果SWAP發生的話, External-SD其實會被掛載到/mnt/sdcard; 而若SWAP沒發生的狀況下, /mnt/sdcard掛載又變回 內置卡。
 - 因此請盡量避免用fixed string, 改呼叫function去做判斷。
 - 判斷是否為外置卡的Sample code如下:
 - Volume屬性isRemovable為true (表示可插拔)
 - 路徑不是 "mnt/usbotg" (此為外接usb)
 - 同理可推得判斷內置卡的方式

```
StorageVolume[] volumes = StorageManager.getVolumeList();
for (int i = 0; i < volumes.length; i++) {
    if (volumes[i].isRemovable() && !volumes[i].getPath().equals("/mnt/usbotg")) {
        //to do
    }
}
```

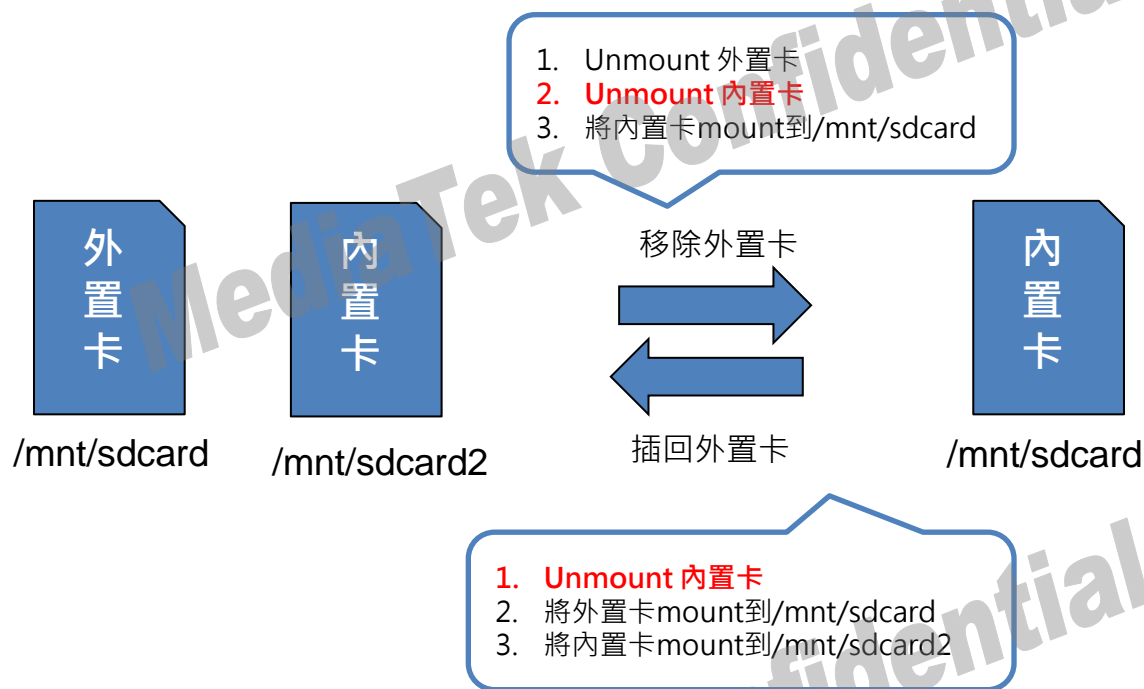

[方案2] 外部sd優先 -- 限制(2/3)

- [限制二] 沒有插入外置T卡時, 無法進行APP2SD
 - 首先必須澄清一點的是, 沒插入外置T卡不是代表不能install APK, 而是代表不能實行APP2SD。MTK原始方案中APP2SD是移動到內置卡。
 - Android原生設計是只允許“ APP2SD移動APP到Primary Storage”。且一台手機上只會有一個Primary Storage。
 - 有同仁問到“為何不讓移除外置卡後, APP2SD改變成移動到內置卡?”。因為若是我們允許APP2SD可以移動到外置SD(/mnt/sdcard)以及內置SD(/mnt/sdcard2), 這會導致APP使用上的混淆。
 - 舉例來說, 若允許APP2SD也能移動APP到內置卡, 今天發生SWAP以後, 因為Primary Storage在此方案是“外部SD優先”, Primary Storage變成了外置卡, Android會找不到原本已經安裝在內置卡的APP, 如此會造成使用者混淆。
 - 為此於此方案中, 我們僅允許在插入外置卡的情況下實行APP2SD。
 - 有插入外置T卡且SWAP enabled:
 - 可以install /data
 - 可以APP2SD (移到/mnt/sdcard)
 - 沒插入外置T卡或unmount外置T卡, 且SWAP enabled:
 - 可以install /data
 - 不可以APP2SD

[方案2] 外部sd優先 -- 限制(3/3)

■ [限制三] 動態SWAP的過程中會kill process

- 由於要達成動態SWAP mount point, 過程中一定會發生 **unmount Phone-Storage** 。
- 在Android行未設計裡, 這會導致當前正在access Phone-Storage的processes被強迫中止。會有無法避免的使用者體驗瑕疵。



[方案2] 外部sd優先－優點/缺點/限制整理

■ 優點

- 以外部sd為主 (掛載到 /mnt/sdcard), 在空間運用上有擴充性 (若空間不足, 使用者可以換購較大容量之外部SD)

■ 缺點

- 由於是以外部sd為主, 所以內置sd的用處不大 (只能在無外部sd時, 去儲存多媒體檔案) → 因此建議內置sd的空間不需太大, 以避免浪費emmc空間

■ 限制

- SD SWAP會影響到所有會去access /mnt/sdcard or /mnt/sdcard2的APP或tool
- 沒有插入外置T卡時, 無法進行APP2SD。
- 動態SWAP的過程中會kill process。
- 因為有這些限制, 而且改動原生Google design具相當風險, 因此我們以不改動原生設計的前提下去實作swap feature, 請務必確定客戶可以接受這些限制, 感謝！

Outline

- Preface
- Android default solution
- MTK SD-SWAP solution
 - Spec
 - Limitation
 - **Hot to enable it?**
 - Related Commands
 - Misc

[方案2] 外部sd優先 -- How to enable ?

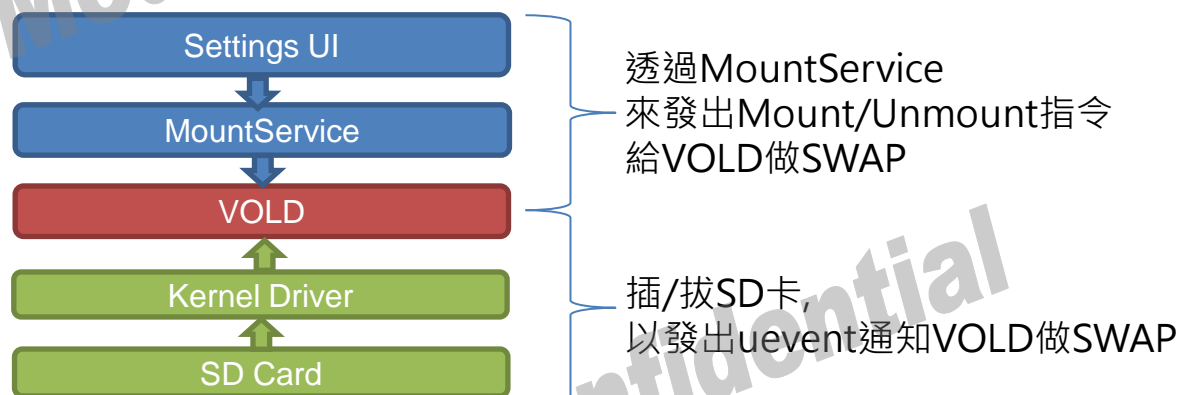
- 該方案因為影響storage行為甚鉅, 因此default disable, 欲使用請依照以下步驟開啟。
- 有加入的branches :
 - ALPS.ICS
 - ALPS.ICS.MP
 - ALPS.ICS.TDVIP.SP
 - ALPS.ICS2.6577.SP
- 如何Enable該方案
 - 該方案支援
 - GB2
 - ICS (Ready on [1012/06](#))
 - 請確定javaoption.pm有” MTK_2SDCARD_SWAP” 這個定義
 - 以及mediatek/config/common/ProjectConfig.mk有
 - MTK_2SDCARD_SWAP=no
 - 開啟 : In alps\mediatek\config\\${PROJECT}\ProjectConfig.mk
 - 有**MTK_2SDCARD_SWAP=yes**
 - 有加**MTK_2SDCARD_SWAP**於**AUTO_ADD_GLOBAL_DEFINE_BY_NAME**

Outline

- Preface
- Android default solution
- MTK SD-SWAP solution
 - Spec
 - Limitation
 - Hot to enable it?
 - **Related Commands**
 - Misc

[方案2] 外部sd優先 – 相關實驗用指令

- 如同前面所言, VOLD會接收來自MountService或uevent的訊息, 以做出SWAP動作, 這裡介紹幾個adb shell指令可以模擬這兩種指令。
 - 模擬MountService發出指令
 - adb shell vdc volume mount /mnt/sdcard2 swap → mount外置卡, 且必須做swap
 - adb shell vdc volume unmount /mnt/sdcard force_and_swap → unmount外置卡, 且必須做swap
 - 模擬熱插拔卡
 - adb shell "echo add > /sys/devices/platform/mtk-sd.1/mmc_host/mmc1/mmc1:\${NUM}/block/mmcblk1/uevent" → 模擬插入SD卡
 - adb shell "echo remove > /sys/devices/platform/mtk-sd.1/mmc_host/mmc1/mmc1:\${NUM}/block/mmcblk1/uevent" → 模擬拔出SD卡
 - **\${NUM}**有可能變動, 請自行填入您手機上看到的數值。



Outline

- Preface
- Android default solution
- MTK SD-SWAP solution
 - Spec
 - Limitation
 - Hot to enable it?
 - Related Commands
 - **Misc**

MEDIATEK

www.mediatek.com

