

Android 架構、開發工具與虛擬平台

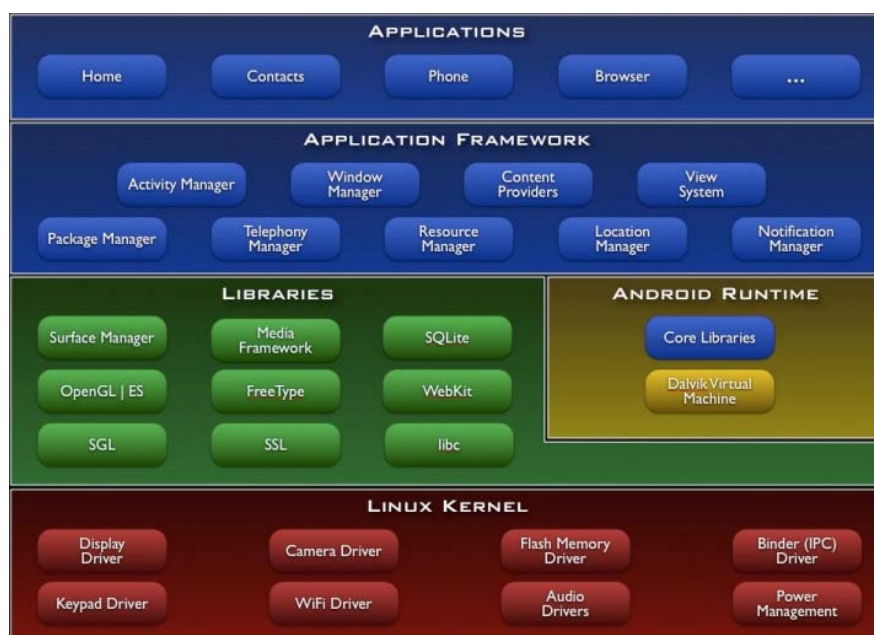
廖彧宏 洪士灝

引言

Google 於 2008 年正式公開的手機作業系統 Android，大幅度降低廠商發展智慧型手機系統的複雜度。由於使用 Android 並不需要支付 Google 軟體授權費用，許多手機廠商均對 Android 抱持支持的態度。台灣業界長期代工設計手機，硬體方面已有相當基礎，Android 的出現，正是台灣業界的契機。Android 採用開放原始碼的技術與策略，讓全世界的開發人員都有機會深入了解並且改善其內部，使得 Android 成為熱門的研究平台，因此本文簡介 Android 平台的架構以及 Android 軟體開發工具中的虛擬平台，以拋磚引玉，希望國內學界業界在開發 Android 應用的同時，能夠更進一步研發與改進 Android 平台，讓台灣擁有世界一流的技術。

Android 平台架構淺說

對Android平台有興趣的開發者，首先應該對其框架(Framework)有基本的認識。基本上，如圖表 1所示由上而下，平台可以劃分為：作業系統、系統函式庫和應用程式等三個層級。最下層的作業系統源自於Linux核心，是個支援許多處理器與周邊裝置的開放原始碼作業系統。中層的系統函式庫又可以分為兩個部分：(1) 以 C/C++ 實作基本的系統(libc)、加解密運算(SSL)、向量圖渲染引擎(SGL)、3D引擎(OpenGL)、字型函式庫(FreeType)、瀏覽器核心(WebKit)、2D/3D顯示渲染引擎(Surface Manager)、多媒體框架(Media Framework)和嵌入式資料庫(SQLite)等函式庫，這如圖表 1的綠色區塊所示；(2) 圖表 1的黃色的區塊，以Dalvik VM (DVM)所建構的Android應用程式執行的環境，使上層的應用程式可以跨平台的執行。支援手機應用所需的各類使用者直接相關的應用程式層，也分為兩個層次：能夠在DVM上運行的函式庫，統稱應用框架(Application Framework)，以及Google和開發者所提供的應用程式，或者是使用者購買與下載取得的軟體。



圖表 1

Google 在設計 Android 平台時採用了將系統拆解成對應特定領域模組的分層結構設計概念，其目的是讓不同領域的協力研發合作廠商，得以專心發展屬於自己特定領域的系統，以因應近來手機功能越益複雜化的趨勢。今日的手機已經被視為一個豐富的操作平台，除了基本的通訊功能，還逐漸被賦予更多元的應用，如播放多媒體的需求、全球定位系統(GPS)的使用、傳統網頁的瀏覽等等。因此，為了平衡多元化操作造成系統的複雜性、應用程式開發的便利性、執行效能與耗電的問題。Google 除了上述所提到的有條理的分層之外，還將整個 Android 架構，也做了許多的細部重整。其中最著名的例子就是一套名為 Bionic 的 C 函式庫。Google 忽略部分標準 C 函式庫中不常用的功能，例如地區化(locales)和寬字元(wide char)的支援，不予以實作。在 Android 框架下，這些被忽略的部分，都可以被 DVM 所隱藏不致於產生問題；而藉由精簡而有效率底層系統，降低 Android 系統執行時對記憶體和處理器計算資源的消耗，以實現嵌入式系統輕量、快速的需求。另外，配合分層的概念，開發人員可以輕鬆的利用 Java 的知識，開發跨平台的應用程式，不必理會系統底層的差異性。

Android 軟體開發概況

為了加速 Android 平台軟體的開發，Google 於 2008 年九月首先釋出第一版的 SDK 1.0 Release 1 供開發人員使用，之後，則規律的釋出新版本的 SDK。今年九月，Google 公布了最新版本的 Android 1.6 SDK，相較於前一個版號，SDK 1.6 針對螢幕解析度、CDMA 和 WiMAX 等新的通信協定、手勢(gesture)、以及語音閱讀等功能提供更多的支援。Android 框架提供了一種開發模型，除了框架本身提供的 API (Application Programming Interface)，Google 針對開發週期

需要的協助，發展一系列的工​​具協助開發人員，這組工​​具與框架統稱為 Android 軟體開發套件(Software Development Kit, SDK) 。這些協力工​​具及其摘要說明如表格 1，工​​具的組成，含蓋應用程式編譯與安裝套件的製作，與整合開發工​​具(Integrated Development Environment, IDE)整合的橋接介面，介面測試與除錯工​​具等等。其中，大多數的工​​具都和虛擬平台有關，因此我們接下來特別針對虛擬平台的架構，以及虛擬平台的使用加以介紹。

表格 1

aapt	封裝應用程式的工​​具
adb	如同 gdb ，用以橋接 Android 系統的除錯工​​具
aidl	產生 IPC 介面
AVDs	產生不同設定的虛擬平台
ddms	IDE 工​​具和 Android 平台的中介軟體
dx	將 Java 的 Byte Code 翻譯成 Dalvik VM 的執行格式
Draw 9-Patch	NinePatch 的視覺化編輯工​​具
Emulator	Android 的虛擬平台
Hierarchy Viewer	協助最佳化應用程式的介面
mksdcard	產生虛擬平台使用的 SD 卡的影像檔(Image File)
Monkey	產生虛擬平台上，使用者亂數行為的事件
sqlite3	存取 Android 平台上資料庫的介面
Traceview	分析程式執行的工​​具
zipalign	針對封裝的進行邊際對齊(Alignment)的最佳化工​​具

Android 與虛擬平台

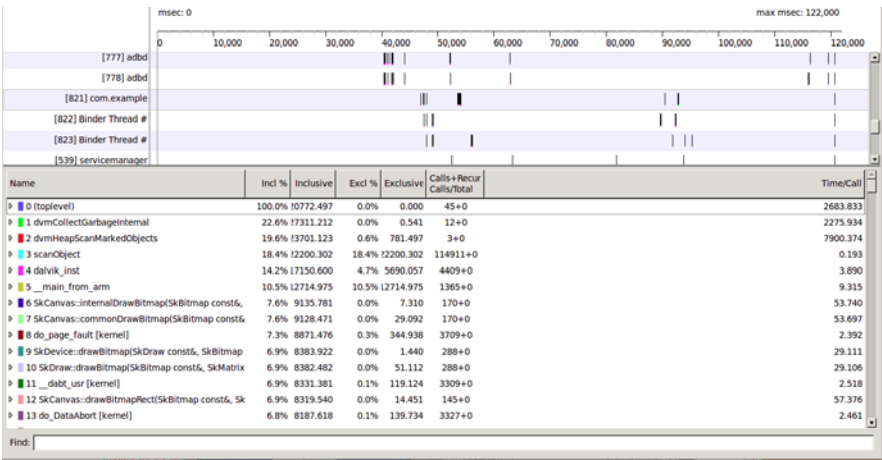
雖然 Android 平台為一個從上至下的開放式平台，但是對於專注於開發應用軟體的人員而言，這些功能強大的 API 如何與底層硬體溝通並不是開發人員需要關注的議題。因此 Google 在 Android 框架中，設計了一個以 ARM 指令集為基礎代號 Goldfish 的系統整合晶片(SoC)，提供一個平台獨立的底層硬體結構，架構出一個虛擬平台(Virtual Platform)，供應用程式的開發人員模擬執行開發的應用程式。

Google 設計的虛擬平台使用開放原始碼的 QEMU 作為核心的技術。QEMU 的開發理念強調異質平台執行的相容性以及高效能的模擬(Emulation)。因此，QEMU 使用眾多的技巧， 包含動態執行碼轉譯(Dynamic Binary Translation)，提升跨平台模擬執行的效率。然而，當虛擬平台在進行模擬時，雖然能夠快速的驗證發展中的程式是否能正確的執行，對於嵌入式系統的開發者而言，我們需要虛擬平台提供更多的資訊，更進一步驗證系統的效能是否符合規格或是分析系統效能的瓶頸所在。因此，虛擬平台若能準確預測開發中軟體在開發中硬體平台的執行效能，甚至估計執行時的能耗(Power Consumption)，將更能加速系統層級軟硬體系統整合設計(System-Level Hardware/Software Co-design)，甚至支援多核心(Multicore)系統的開發，產生更

有效率的手持裝置程式。

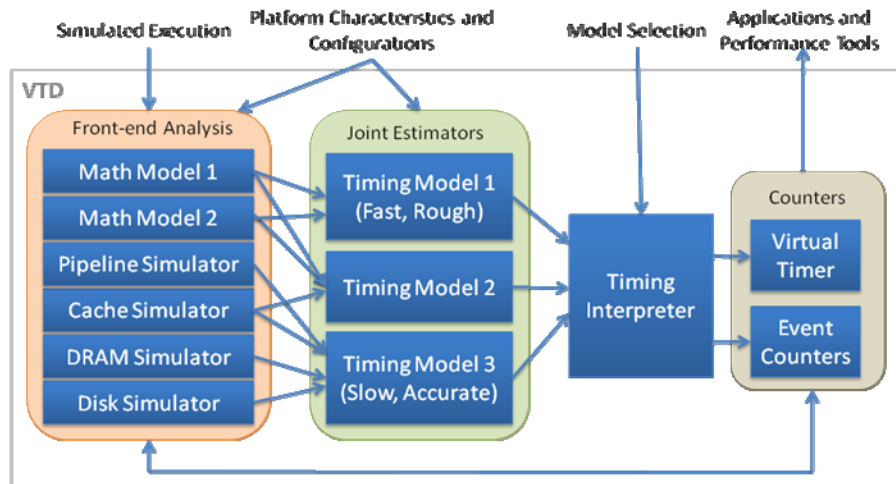
虛擬平台與實際平台的效能評估

Android SDK 包含了兩種主要效能分析工具，首先是由android.os.Debug核心套件所提供的追蹤工具 (Tracing Tool)，由DVM的觀點分析DVM核心函式庫和應用程式的函式之間，呼叫的交互關係與執行時間的分佈；另外一個則是嵌在虛擬平台中的追蹤工具，使用者可以利用參數–trace啟動此功能，虛擬平台將記錄整個系統執行時的系統函式庫的呼叫，我們可以由此了解程式執行流程，並且分析處理器消耗在各軟體函式，作業系統模組的時間。最後我們可以使用Traceview這隻前端程式，視覺化分析的結果，圖表 2展示Hello World範例程式執行時，Android函式庫彼此執行時的時間分布。



圖表 2

雖然Google提供了以上的工具，但是由這些工具產生的結果，往往無法準確預測真實系統執行的效能。我們台大資工的研發團隊針對此一問題在QEMU上，加入時間擬真裝置(Virtual Timing Device, VTD)的概念，如圖表 3，並且研究快速而準確的效能預測，以整合現有不同層級的模擬技術，讓使用者能選擇性的在模擬過程加入pipeline, cache, memory甚至disk的細部模擬器，以得到更迅速確實的結果。



圖表 3

虛擬平台的能耗評估和能耗管理

如時間虛擬裝置的觀念一般，我們提出能耗擬真裝置(Virtual Power Device, VPD)的作法填補 Android QEMU 虛擬平台的不足。先進的手持平台，會針對不同功能模組，提供不同的電源迴路。因此，理想的 VPD 不僅顯示處理器變頻的資訊，還可以呈現不同電源迴路模組之間開關對整體系統造成的影響，並能搭配 VTD 提供動態調頻對系統效能造成的衝擊。當虛擬平台能夠提供基本能耗資訊時，開發人員就不需要等待硬體協力廠商設計出實際的平台，而可以直接透過 虛擬平台評估與分析應用程式的效能/能耗比。

目前 Android 在電源管理的支援稍嫌薄弱，僅針對少數元件進行控管，但是在節能省電的潮流下，先進的手機或是 MID 平台中可能擁有數顆不同功能的處理器與專屬周邊，由不同的供電系統維持運作，不被應用程式所使用的硬體可以獨立的控制降頻甚至是關閉。因此，建置一套適用於 Android 平台電源管理框架，以減少程式執行以及資料傳輸時電量消耗，並延長行動裝置之使用與待機時間，是相當重要的研究。目前，我們台大與中研院的合作研發團隊正在著手開發此框架，以提供國內系統程式開發者，甚至應用程式設計師，由上而下之電源管理界面。此外，我們亦會針對適地性應用服務分析其經常性負載樣式，提出適合的系統資源管理分配策略。

結論

目前國內對 Android 開發與其相關的研究，正是方興未艾。有許多議題尚待研究與解決。以虛擬平台為例，若能更積極有效利用於開發 Android 系統或是評估與改善應用程式的效能，則可縮短產品研發時程，提昇產品設計的競爭力。筆者拋磚引玉，希望藉由對 Android 虛擬平台的討論，能與台灣產業在相關的議題上有更深厚的互動關係。

關於作者

廖彧宏，台灣大學資訊工程研究所碩士，現任中央研究院資訊所研究助理，研究及開發 Android 平台效能改進及節能省電技術。

洪士灝，密西根大學資訊工程博士，曾任美國昇陽電腦公司主任工程師，現任台灣大學資訊工程學系助理教授，專長為電腦系統軟硬體架構研究、嵌入式多核心系統、效能分析與最佳化工具等。