

# Android 評比工具調查與多核心實驗

柯亨雨 林盈達

國立交通大學資訊工程學系

Email: kohsiangyu@gmail.com, ydlin@cs.nctu.edu.tw

September 30, 2013

## 摘要

隨著目前智慧型裝置不斷發展，手機上陸陸續續地出現了多元化的應用。但是在這些應用中卻也產生了一些問題，有些應用執行在單核心手機比較省電，有些則是在多核心表現較優異且較省電，我們該如何在這些應用中尋找出一台最符合個人需求的手機？因此我們將先評比各種手機相關測試工具，包括 CPU、GPU、I/O、Battery，再聚焦在多核心的性能與功耗的三個議題：核心數差異、平台差異、以及應用差異。在做完各種實驗後發現手機核心數增加雖然會增加功耗但是執行的時間縮短，手機整體的功耗反而減少了。不過也發現有些手機雖然核心數不多，可是他單顆核心的執行效率卻很高，所以如果使用者平時所用的軟體在單核或多核心手機上性能差不多的話，那挑選這一類的手機反而會更好。

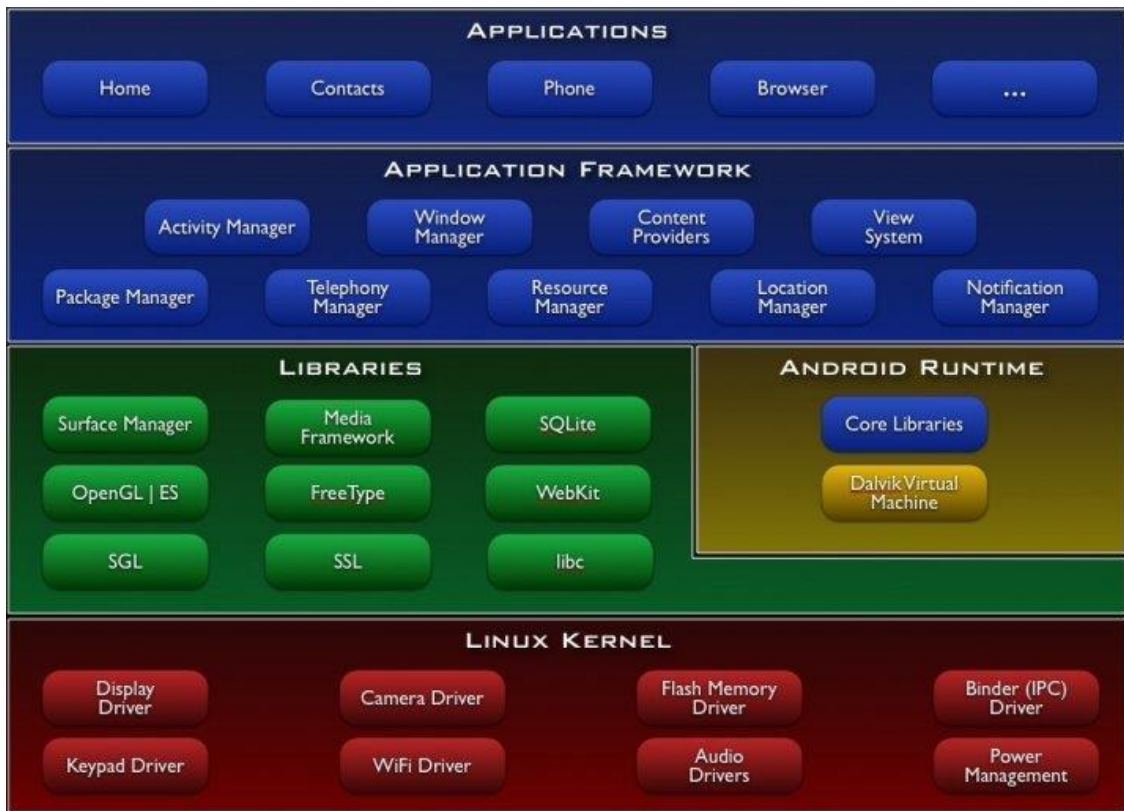
關鍵字：多核心、核心數目差異、平台差異、應用差異

## 1 Android 作業系統簡介

Android 為近年 Google 因應智慧型手機蓬勃發展而推出的一套作業系統，它為一套 Linux-based 的開放原始碼作業系統，並且以 Java 為其開發語言。它憑藉著它的開放原始碼以及網上商店平台(Google Play)在眾多的系統中脫穎而出成為目前全球第一大智慧型手機作業系統。[1]

Android 系統在設計上採用了分層的架構，一共五層(如圖 一)，分別為：Applications、Application Framework、Libraries、Android Runtime、Linux Kernel。Android 在系統的最底層採用了 Linux 作為其系統核心，在 kernel 上面有 Libraries & Android Runtime 提供 Android 程式執行時所有需要用到的資源並管理。在 Android Runtime 裡有兩個主要元件：Core Libraries、Dalvik Virtual Machine，每當我們今天系統上執行一個新的程序時，Dalvik VM 會自動建立一個虛擬環境來給那個程序執行。每一個程序一個虛擬環境所帶來的好處在於當今天手機上一個程式執行失敗時它並不會造成整台手機不能正常運行，系統只需要結束掉指定的程序就能繼續正常運行了。手機整體的穩定性對於一台手機是很重要的，特別是如果今天手機因為某個小程序當機，造成手機通訊模組不能正常工作的話會，導致手機無法及時接聽來電。

在圖一的 Dalvik VM 裡它提供了 JIT(Just-in-time) Compiler 的功能，令程式在執行時可以即時的將 Java bytecode 轉譯為 Machine Code，進而加快程式執行的效率。這項功能會產生一個比較嚴重的問題在於當我們今天平台所用的處理器為非 ARM 系列時，程式有些功能會無法正常執行。在後面的幾項測試裡就會遇到有部分軟體因為還沒對目前的 Intel 平台作支援，所以有部分測試不是不能執行就是程式運行一半系統就強制關閉了。



圖一 Android 架構圖[2]

## 2 Android 各式工具探討

會影響使用者體驗的主要有兩件事：流暢度和穩定性。流暢度的部分我們可以透過利用不同的測試工具得出每支手機的性能。穩定度的部分則是除了可以利用探測工具觀察程式是否有 overflow、自動執行一些隨機操作觀察程式的反應，它還可以記錄每隻程式執行時間、記憶體使用量、中間呼叫了哪些模組等。

根據工具測試的方法我們分為兩大類—效能測試工具與探測工具。效能測試工具為直接安裝在待測物上直接執行作比較，探測工具則是主要利用外部平台的工具對待測物作分析。效能測試工具的部分再細分為只能測試單一方向(CPU、GPU 等)性能的工具與能測試各種不同功能的工具。

## 效能測試工具

效能測試將分別討論測試 CPU、GPU、I/O、Battery、Bundle 這幾個項目。手機廠商根據不同使用者需求推出了多款智慧型裝置，它們最主要的差別在於 CPU&GPU 架構、系統類型&版本、電池容量，上述這些因素都會影響手機的性能以及功耗表現。CPU 的部分我們將各別討論多核心的支援與否、整數運算、浮點運算的性能，根據表 一我們可以知道 Linpack & Nbench 都支援雙核心測試，但是支援整數運算測試的只有 Nbench。GPU 的部分對於使用者而言最重要的問題在於他們想玩的遊戲哪幾款手機才可以流暢的執行，透過表 二裡面的兩隻程式裡面提供了許多不一樣的場景，使用者可以根據那些場景為依據得知他們想玩的遊戲是否可以流暢的運行。這兩隻程式中比較特別的是 GFXBench，它除了有 Android 版的測試軟體外還有提供其它系統的測試軟體，所以當使用者想嘗試不同的作業系統時也有軟體可以當作參考基準。I/O 的部分則是幫助我們了解每支手機上對記憶體的操作性能，這個部分最實際的應用是拍照及錄影，手機 I/O 越快代表手機連拍或錄影時因為記憶體寫入影響畫面品質的因素越小。除此之外，根據表 三 Androbench 還可以測試 I/O 對於瀏覽網頁以及 SQL Query 的影響。Battery 的部分則是要來觀察當我們今天智慧型裝置執行不同的工作時它們的功耗。根據表 四，Battery Benchmark 這套測試工具模擬我們平時使用手機的習慣來測試手機可以使用多少時間，另外兩套則是提供使用這觀察當前手機目前電池的電壓與電流等資訊。最後 Bundle (表 五)的部分則是一些不同於前面那幾組測試單一方向為主的軟體，它們提供我們測試不同的功能並且根據每家廠商經驗，對每項測試加權後提供了一個綜合指標。Vellamo & Antutu 都提供了非常多樣性的測試，他們最大的差別在於 Antutu 額外提供了多核心的支援。

表 一 CPU Benchmarking Tools

	<b>Linpack</b>	<b>Nbench</b>	<b>LuaPi</b>	<b>RealPi</b>
Multi-platform Support	Yes	Yes	No	No
Scale	Mflops	Points	Sec	Sec
Integer	No	Yes	No	No
Float	Yes	Yes	Yes	Yes
Precision Selecting	No	No	No	Yes

表 二 GPU Benchmarking Tools

<b>Name</b>	<b>Platform</b>	<b>Scale</b>	<b>Level</b>
Neocore	Android	FPS	Overall
GFXBench	Cross-Platform	FPS	Partial

表三 I/O Measuring Tools

	<b>Androbench</b>	<b>MemBench</b>	<b>A1 SD Bench</b>
Internal Memory	Yes	Yes	Yes
SD	No	No	Yes
RAM	No	No	Yes
SQLite	Yes	No	No
Browser	Yes	No	No

表四 Battery Benchmarking Tools

	<b>Battery Benchmark</b>	<b>Battery Test</b>	<b>Battery Monitor Widget</b>
Stress Battery	Yes	No	No
Current	No	Yes	Yes
Voltage	No	Yes	Yes
History	No	Yes	Yes

表五 Bundle Benchmarking Tools

	<b>Vellamo</b>	<b>3DMark</b>	<b>AnTuTu</b>	<b>CF-Bench</b>	<b>PassMark</b>
CPU	Yes	No	Yes	Yes	Yes
Multicore	No	No	Yes	Yes	No
Memory	No	No	Yes	Yes	Yes
I/O	Yes	No	Yes	No	Yes
2D	Yes	No	Yes	No	Yes
3D	Yes	Yes	Yes	No	Yes

## 探測工具

探測工具為利用一觀測平台對 DUT 上面的資料作分析或操作，它的內容包含執行時間、中間呼叫的函式、每個函式花費了多少時間、模擬使用者對螢幕的操作等。透過上述這些功能，我們可以用來找出程式運作時是否穩定，也可以找出程式的整個執行過程中是否有特別耗費運算性能的部分等。它的運作模式是先在觀測平台與 DUT 之間建立一個 API 作為溝通橋梁，用來分析與操作 DUT。這個 API 由 ADB(Android Debug Bridge)[7]的三個程式(client、server、daemon)互相配合提供，client 運行在 DUT 上面與觀測平台上的 daemon 直接溝通，server 則是提供觀測平台上面的探測工具與 daemon 之間的溝通。

### A. 分析工具

Android 上主要的分析工具如表 六，它們各自都可以針對 DUT 記錄一些程

式執行中各項參數，像是 Heap Size、幾個執行緒、每個執行緒的 CPU Time。DDMS[8](圖 二)為一套整合工具，它本身除了提供的一些功能外它還會從 LogCat 取得資料並一起顯示給使用者。LogCat 與 DDMS 最大的差別是它能夠提供比起 DDMS 本身內建的功能更加詳細的資訊，唯一的缺點是本身沒有介面，使用者須透過 DDMS 來把傳回的資料圖表化。Android Instrumentation 的使用方式與上述兩套有點不同，它是透過直接安插一些測試碼在程式裡面，它的優點在於使用者可以選擇只觀察整個程式中的某個小函式。

表 六 探測工具 - 分析

	DDMS	LogCat	Android Instrumentation
Heap Size	Yes	No	No
Threads in a process	Yes	Yes	No
Thread CPU Time	No	Yes	Yes
Event Trigger Time	No	Yes	Yes

## B. 互動介面

Traceview[9]主要是用來把一些手機上的 log 給圖表化，而 DDMS 比較特別的一點是它除了可以把資料圖表化之外，它還可以跟 Traceview 搭配使用，並且它還可以對待測物作即時的系統分析。從圖 三我們可以看得出 Traceview 可以利用時間軸的方式顯示出哪個時間點有哪些程序在運行，它們各自使用了多少的 CPU Time 等。而根據圖 二，DDMS 資料呈現的重點在於每個程序使用了多少系統資源。

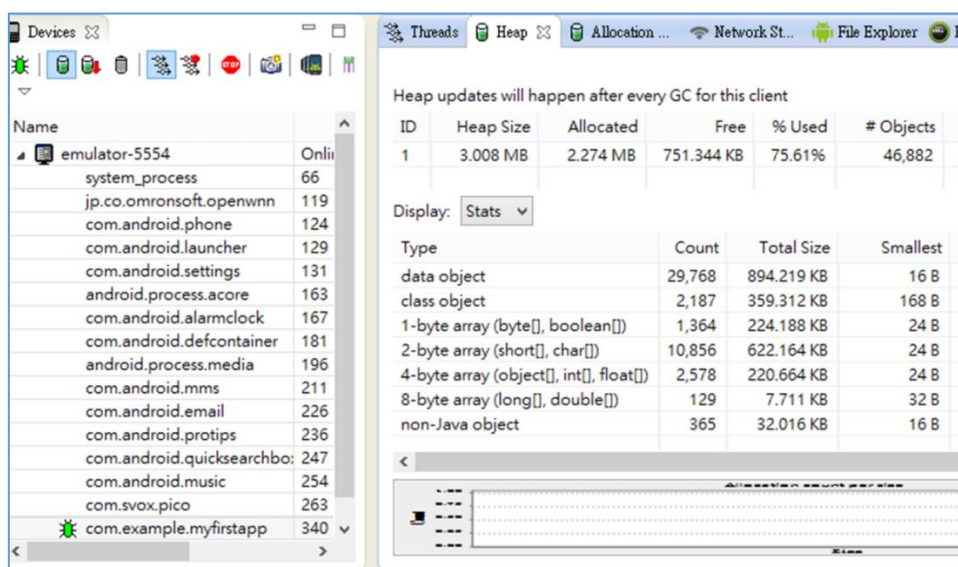


圖 二 DDMS

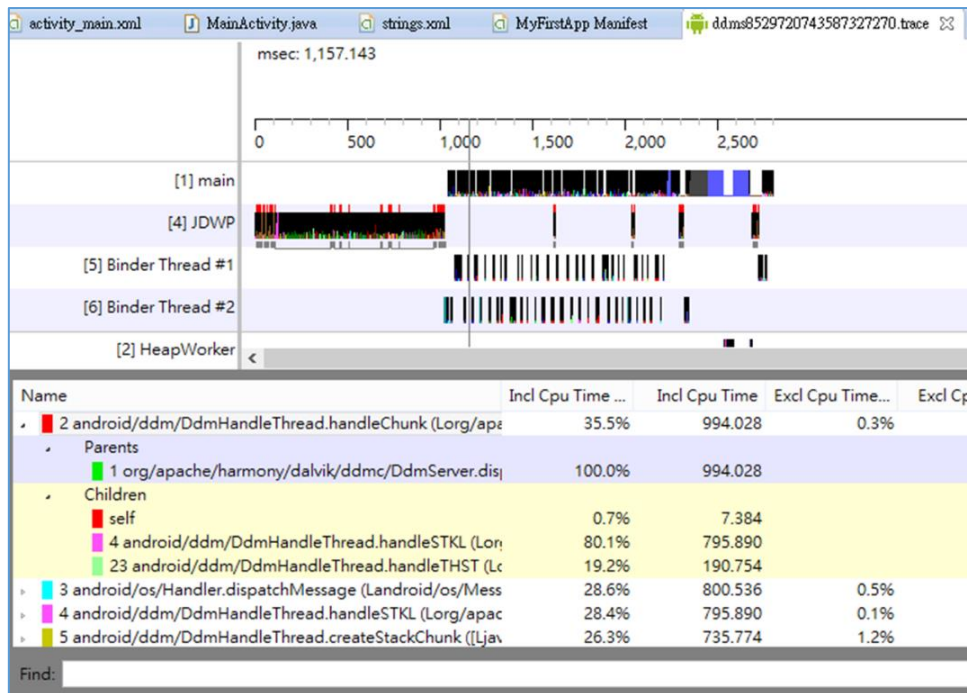


圖 三 Traceview

### 3 多核心問題探討與實驗設計

#### 目前市場趨勢與工具選定

由於目前製成技術成熟，智慧型裝置內核心數逐漸的增加，提升智慧型手機的處理速度，但是手機核心越來越多真的比較好嗎？手機到底會不會因為核心增加了反而變得更耗電？還有就是我們現在手機上越來越多的核心，實際上能讓我們軟體變快多少呢？

為了解決上面所提及的兩大問題，我們將在表 七裡挑選出三套比較適合用來觀察多核心性能的工具。透過這些工具我們將要分別測試同一裝置下不同核心數目、不同平台相同核心數目、相同平台不同軟體的性能與耗電。在核心數差異方面，目前市面上的軟體有些在執行時只有單一執行序，而有些卻有多個執行序，當這些軟體在多核心的系統上執行時，它們核心數差異會如何影響手機的性能與功耗是我們要探討的目標。而在平台差異方面，雖然很多手機都是使用 ARM 作為其 CPU 核心架構，但是 ARM 不同世代就算核心數相同性能也是有所差異的，因此要探討在這些各式各樣的架構下它們性能與功耗會有哪些差異。最後在應用程式差異的部分，每顆處理器在設計上它的整數運算能力與浮點運算能都有所不同，因此我們要比較在相同裝置下不同軟體之間的性能與功耗。

根據表 七我挑選出了 Antutu、Vellamo、RealPi。Antutu 和 Vellamo 雖然都

支援豐富的測試項目，Antutu 雖然支援雙核心，可是因為它本身無法設定是否使用多核心來測試，所以我挑選了 Vellamo 來作為單核心性能的比較用途。RealPi 跟 Antutu 不同的地方在於 Antutu 多核心的測試是同時執行多個類似的工作來計算，而 RealPi 則是使同一個測試分散到不同的 CPU 來進行負載平衡，所以 RealPi 可以表現出一個系統負載平衡方面的能力。

表 七 Selected Tools

	AnTuTu	Vellamo	RealPi	CPU Prime	Linkpack	BenchmarkPI
Scale	Points	Points	Secs	Points	MegaFlops	MiliSecs
Better	↑	↑	↓	↑	↑	↓
CPU	Yes	Yes	Yes	Yes	Yes	Yes
Multicore	Yes	No	No	Yes	Yes	Yes
I/O	Yes	Yes	No	No	No	No
2D	Yes	Yes	No	No	No	No
3D	Yes	Yes	No	No	No	No

## 4 實驗

### 測試平台建置

由於市面上發行的智慧型裝置不支援核心的開關，所以除了測試一般實體手機(表八—P920)之外，還需另外架設虛擬環境(表八—DV6505)來搭配測試單雙核心的性能差別。除了虛擬環境之外，還有一個測試裝置是直接安裝 Android 在實體的裝置上(表八—1005PE)。用來模擬和安裝在實體機上的 Android 發行版本根據表 九我選擇了 Android-X86，它跟其它套比較起來性能以及對於 APP 的支援都相對比較好。

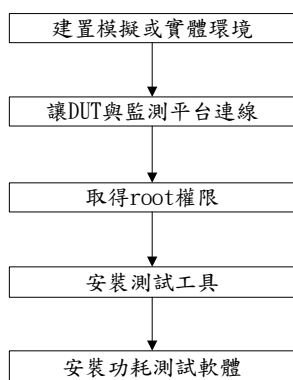


圖 四 Android 測試平台

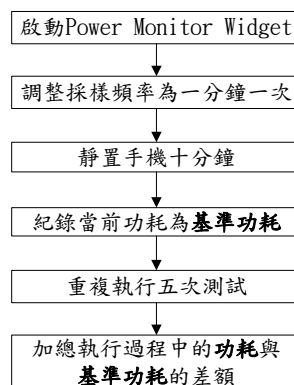


圖 五 功耗測試平台

表 八 Testing Platforms

Name	Type	Arch	Detail
HP DV6505	Simulate (VM Player 8)	X86	CPU : Intel Core 2 Duo T7100 RAM : 1GB Android Version : 4.0.4 [5]
ASUS EeePC 1005PE	Hardware	X86	CPU : Intel Atom N450 RAM : 2GB Android Version : 4.0.4 [6]
LG P902	Hardware	ARM	Android Version : 2.3

表 九 Simulation Environment

Name	Lisence	Arch	Simulation	Speed	Android Market
Google ADT	Freeware	ARM	Platform	Slow	Non
Android-X86	Freeware	X86	Platform	Fast	Yes
BlueStacks	Freeware	X86	APP	Slow	Non
Windows Android	Commercial	X86	X	X	X
YouWave	Commercial	X86	X	X	X
Pokki	Commercial	X86	X	X	X

## 實驗—核心數差異

在這個實驗裡我分別測試了每部裝置在單核與雙核時的性能表現並且直接以單核與雙核的成績比值作為比較。

根據表 十與圖 六的數據顯示，當 CPU 的核心數量改變時 Antutu 與 RealPI 的 FFT 演算法可以得到不錯的加速，HP dv6505 的部分加速了大約 75% 左右。在 EeePC 部分因為它實際上是單核心+Intel HT，所以它在加速的方便變動的比較大，從加速 20% 左右到 75% 左右都有。

功耗方面由表 十一的數據可以得知雖然核心數增加會造成功耗提高，但是因為性能提高的關係，運行時間縮短了，造成整個程式執行週期間的實際功耗是變少的。

表 十 Performance Improvement

Device	Core	Antutu	Vellamo	RealPi	
				FFT	Machin
<b>HP dv6505</b>	2	69.95%	0%	75%	15.2%
<b>EeePC 105PE</b>	2	28.4%	7.46%	78.57%	12.87%

表 十一 Power Efficiency Gain



Device	Antutu	Power
Eeepc 1005PE 2 core	6641	613 mA/h
Eeepc 1005PE 1 core	5172	685 mA/h
Proportionality	28.4%	11.7%

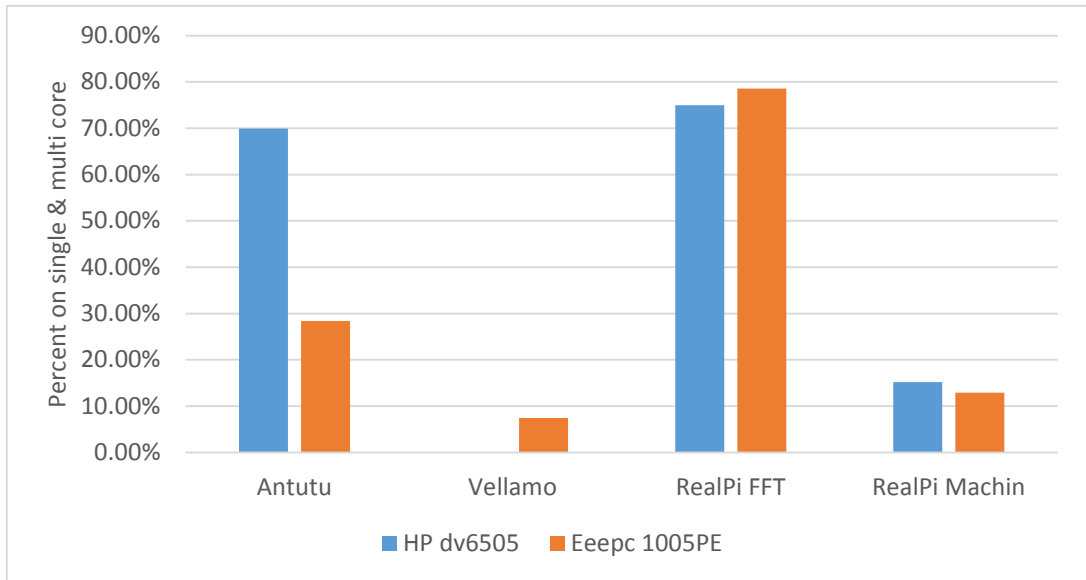


圖 六 Performance Improvement

## 實驗—平台差異

這個實驗裡要探討的是在這些各式各樣的架構下他們性能與功耗會有哪些差異。

根據表 十二與圖 七的數據顯示雖然 LG 和 Nesus 都是 ARM 架構，但是很明顯的是 Nexus 測試成績雖然高了快整整一倍，功耗增加的比例卻沒有來的那麼大。從另一方面來看 LG 和 Eeepc，LG 為真實的雙核心，Eeepc 的 Atom 為單核模擬雙核，它們的性能差不多，可是功耗卻差了一倍多。

表 十二 Performance and Power

Device	Cores	Antutu	Power
Eeepc 1005PE	2	6641	9195 mW/h
LG P920	2	6961	22013.4 mW/h
Nexus 7	4	12029	37874.6 mW/h
HP dv6505	2	13421	Virtual Machine

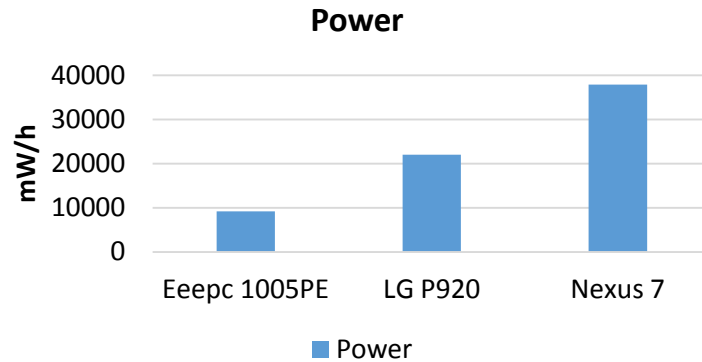


圖 七 Power on Different Platform

## 實驗—應用差異

每顆處理器在設計上都有不同的整數運算與浮點運算能力，因此我們要比較在相同裝置下不同軟體之間的性能與功耗。

根據表 十三的數據顯示，每個不同的應用程式跑出來的結果會隨著平台的不同而有不同幅度的性能提升。就以 dv6505 和 Nexus 7 相比，在 Vellamo – HTML 時 dv6505 的性能勝過 Nexus 7，可是到了 RealPi – Machin’s formula 時，Nexus 7 反而比較快。

表 十三 Application Performance

Application	Scale	Better	Dv6505	Eeepc	LG-P920	Nexus 7
Vellamo(HTML5)	Points	↑	1827	1210	614	1203
Vellamo(Metal)	Points	↑	1164	488	295	374
Antutu	Points	↑	13421	6641	6961	12029
CPU Prime	Points	↑	1179	1023	1041	2084
Linpack	MegaFlops	↑	44.107	60.979	66.3	65.731
BenchmarkPi	Millisecond	↓	588 ms	576 ms	609 ms	427 ms
RL Benchmark	Seconds	↓	24.234 sec	26.725 sec	70.843 sec	46.932 sec
RealPi(AGM + FFT)	Seconds	↓	0.08 sec	0.14 sec	0.40 sec	0.14 sec
RealPi(Machin's formula)	Seconds	↓	92.59 sec	128.34 sec	102.45 sec	89.12 sec

## 5 結論

- i. 如果今天我們常用的軟體支援多核心，那使用更多核心的手機在提高我們程式執行的速度和提升整體操作的流暢度的同時它還能提高手機的續航力。

- ii. 如果今天使用者常用到的應用都只能支援單核心的話，選擇手機時反而應該選擇核心數較少但是單顆核心的效率較佳的處理器。

## 6 參考資料

- [1] Android [Online]. Available: <http://zh.wikipedia.org/wiki/Android>
- [2] A. beginner. (2009, July 5). 認識 Android 程式架構 [Online]. Available: <http://tyroandroid.blogspot.tw/2009/07/android-2.html>
- [3] C. Lee, E. Kim, H. Kim , “The AM-Bench: An Android Multimedia Benchmark Suite”, Georgia Institute of Technology, 2012.
- [4] H.J. Yoon , “A Study on the Performance of Android Platform”, International Journal on Computer Science and Engineering, April 2012.
- [5] F. Yu. (2012, December 30). 相隔一年半，再次嘗試 Android x86 [Online]. Available: <http://fishome.tw/?p=45>
- [6] Android-x86 - Porting Android to x86 [Online]. Available: <http://www.android-x86.org/download>
- [7] Android Debug Bridge [Online]. Available: <http://developer.android.com/tools/help/adb.html>
- [8] Dalvik Debug Monitor Server [Online]. Available: <http://developer.android.com/tools/debugging/ddms.html>
- [9] Traceview [Online]. Available: <http://developer.android.com/tools/help/traceview.html>
- [10] Antutu [Online]. Available: [https://play.google.com/store/apps/details?id=com.antutu.ABenchMark&hl=zh\\_TW](https://play.google.com/store/apps/details?id=com.antutu.ABenchMark&hl=zh_TW)
- [11] Vellamo [Online]. Available: [https://play.google.com/store/apps/details?id=com.quicinc.vellamo&hl=zh\\_TW](https://play.google.com/store/apps/details?id=com.quicinc.vellamo&hl=zh_TW)
- [12] Real Pi Benchmark [Online]. Available: [https://play.google.com/store/apps/details?id=com.georgie.pi&hl=zh\\_TW](https://play.google.com/store/apps/details?id=com.georgie.pi&hl=zh_TW)