# Are Emulators Trustworthy
## - In terms of speed

**Søren Andreas Juul** *

* *Aalborg University, Department of Computer Science, (e-mail: sjuul09@student.aau.dk).*

**Abstract:** Emulators if often used for testing software for embedded devices, such as mobile phones. An emulator emulate the behaviour of a device, thereby enabling the developer to test software for that device, and determine whether it is running correctly, before running the software on a real device. However, can they be trusted in termes of speed? The following will describe the test of two important subjects: Computation and memory access speed, which is run on both an emulator and an embedded device, for comparison.

## 1. INTRODUCTION

When developing software for embedded devices (from now on referred to as devices), such as mobile phones, speed is critical. If the software does not react fast enough, users will get annoyed by it and maybe stop using it, much like they would by software on a computer. Since the device often has less recourses available for computations, than computers, it can do less computations per second than a computer, thus developers must be careful computations does not take too long.

A common way of testing software for embedded platforms is by running the software on emulators. An emulator is an application running on a computer. The emulator has the primary purpose of functioning as the real device does, thereby the tester can interact with the software to determine if it works as desired. However, if the emulator runs faster or slower, than the real device, the developer might be tricked into thinking that the software is fast enough, and not consider efficiency or that the software is too slow and spend unnecessary time on making the software more efficient.

First two tests are described, which have the purpose of examining two important subjects (memory and computation), which can affect the speed of emulators and devices. Next the results is visualised and analysed. The results are then discussed and concluded upon. Lastly, related subjects for further work are listed.

## 2. METHOD

The tests will focus on the Android platform, which supplies a rich set of development tools, including an emulator. The tests will be run on both the Android emulator Android [2010a] and a real device (a HTC Magic mobile phone). The hardware of the computer running the emulator and of the device Corporation [2010] is listed in Table 1.

| | **Emulator** | **Device** |
|---|---|---|
| **CPU** | Core 2 Duo 2 GHz T7300 | 528 MHz MSM7200A |
| **RAM** | 3 GB (667 mhz) | 192 MB |
| **OS** | Ubuntu Linux | Android |

Table 1. Data for computation test

CPU is a abbreviation of Central Processing Unit. The CPU executes instructions from the applications it is running, e.g. add two number or compare two numbers. Core 2 Duo is a CPU that has two cores that can compute in parallel. One GHz equates to $1,000,000,000$ hertz and one MHz equates to $1,000,000$ hertz. These units expresses the speed which CPUs is executing instructions, thus the computer running the emulator is almost four times faster than the mobile device. RAM is where applications and the operating system stores data, the storage is not persistent, i.e. the data is lost when the device is turned off. MB and GB is a abbreviations of megabyte and gigabyte. The computer running the emulator has over 15 times more memory than the mobile device. Operating system (OS) is in charge of handling communication between the applications and the underlying hardware

Timing, on both the emulator and device, was done with a function supplied by the Android system called `TimingLogger`, which logs the time between points in the execution Android [2010b]. The test will concern two important parts of development: Computations and memory access.

### *2.1 Computations*

Computations, also referred to as information processing, is what computers do. They get input, either from files, keyboard, mouse, etc. and processes the information. For example a web browser takes a web-address and a network connection as input and outputs a website to the screen. To compare the speed of computations made on the emulator and on the device a test has been composed, which does basic vector mathematics: Testing whether a vector intersects a sphere. the calculations are run 1000 times, each time the vector is moved up the x-axis. The vector intersects the sphere about 500 out of 1000 times, and when it does, the test also calculates where the in the coordinate system the intersection occurs.

### *2.2 Memory Access*

When computations are made, you often need to store intermediate values, to be used later, this is done in the device/emulators memory. The second test focuses on the speed of writing to- and reading from memory. The test writes and reads arrays [1] of six different sizes (20, 200, .., 2,000,000). A single letter is written to each of the array cells, and when a whole array has been filled the whole array is read into a variable [2] one array cell at a time. The test times from write start to it is done and from the read start till it is done, this gives two set of data for device and the emulator, each containing six results (one per array size).

## 3. RESULTS

Each of the tests were run ten times and the mean is deduced to decrease the influence of noise. Noise can be caused by many things, e.g. modern operating systems has the ability to run several applications side by side, however only one can be executed on a CPU core at the time. To run several application, the operating system switches between the applications to share the runtime [3] . If another application, than the test application, suddenly requires a lot of CPU time, this will necessarily affect the time of a simultaneously running test application.

### *3.1 Computations*

As shown in Fig. 1, the computations are done 5.58% faster on the device than on the emulator.

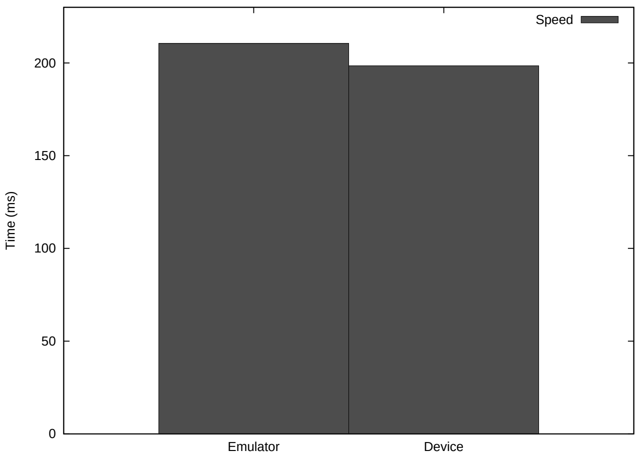


Fig. 1. Results of the computation-test

### *3.2 Memory Access*

As mentioned above, the test of memory access is divided in two, writes and reads. The results are summed up below.

*Write* As shown in Fig. 2 the emulator seems to be the fastest when the number of writes is low, but is overtaken when writes exceeds $2,000$ by the device.

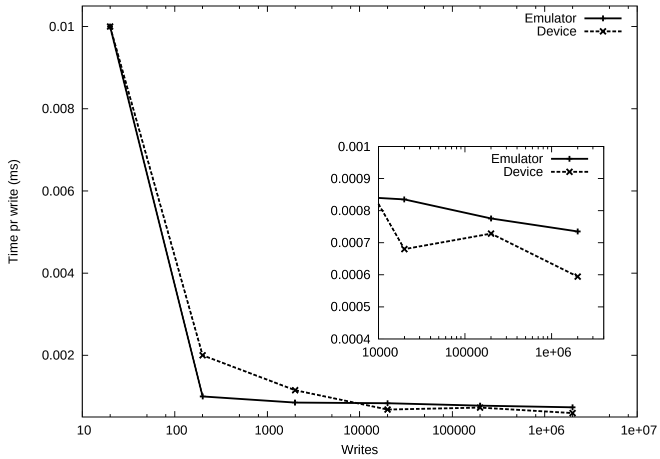


Fig. 2. Speed pr write for different numbers of writes

*Read* Like write, the reads graph shows an increase in speed per read, as the number of reads increase. However the device shows to be fastest from start to end, as seen on Fig. 3.

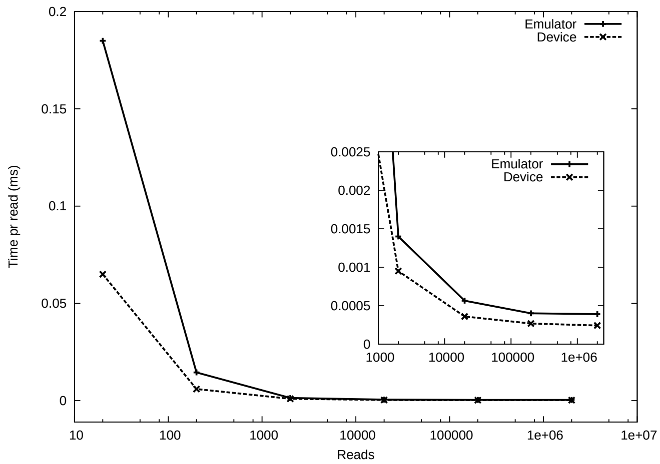


Fig. 3. Speed pr read for different numbers of read

### *3.3 Analysis*

As described in Method, the emulator is run on a computer which has a CPU much faster than the mobile device, as well as several times more RAM available. However the results still shows that the device is faster than the emulator in most situations. One reason for this is that the emulator is running on top of another operating system, i.e. the computer has to run its own operating system, which takes CPU time, while it runs the Android operating system, whereas the device only has to run the Android operating system. The emulator system also has to emulate the hardware of a real device Android [2010a], this increases the complexity even further. A simplified model of the added complexity can be seen in Fig. 4 as additional layers.

[1] An array in this context is a piece of memory which contains a series of data, e.g. several numbers or several pieces of text. The arrays used for this test is static, meaning that the length cannot be changed when the array has been created.

[2] A variable is piece of memory which contains one type of data, e.g. a number or some text

[3] Runtime is a term referring to the time where an application is being executed, i.e. running.

Another possible reason for the device being faster than the emulator, is usage of cache memory. Cache is high-speed memory placed close to the CPU. When memory is accessed in the RAM, it and the adjacent pieces is placed in the cache. This is done because adjacent pieces of memory is often accessed shortly after the original piece is accessed. Thus accessing adjacent memory in cache is fast, compared to accessing adjacent memory in RAM. The computer running the emulator may not be able to utilise the cache as well as the device, because running the emulator also requires memory access, thus also filling up the cache, not leaving as much room for the test application. The hardware emulation may also effect the way memory is accessed and the cache utilisation.

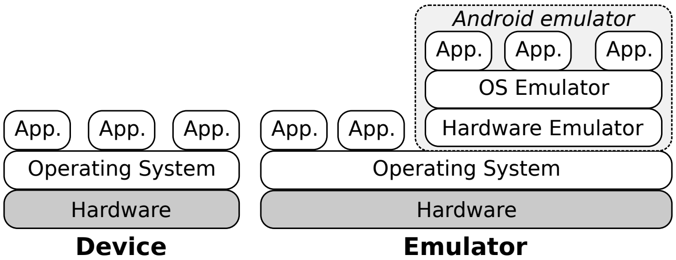


Fig. 4. Shows the difference in number of layers to be traversed from top to bottom on a device and an emulator. Application is abbreviated App.

Both reads and writes shows that speed is slower when the number of reads/writes is lower, one cause of this could be that the first or first couple of access requires some sort of setup or memory allocation, thus imposing an overhead, which is gradually evened out, as the writes/reads increases, making the overhead less significant.

## 4. DISCUSSION AND CONCLUSION

All results shows that the device is faster than the emulator, except for the write test, where the device is only partially faster. The difference between the emulator and the device is very small, and may not be an issue in most situations. The difference between the emulator and the device is $0,00014055$ ms per write, for $2,000,000$ writes. This gives a total time difference of 281,1 ms for all $2,000,000$ writes, and a 1,4 seconds time difference when the number of writes hits $10,000,000$. This may seem like a small time penalty to pay in most situations, however if the application is to be used in real-time systems, even small time differences may have great consequences.

To sum up, the results show that the device is faster in most situations, thus developers can expect that their applications will run about as fast on the device as on the emulator and in some situations even faster, however, time dependent applications needs to be tested on the device to ensure that time constraints are fulfilled.

## 5. FURTHER WORK

This article only concerns itself with the Android system, other emulators may have different properties and as such the results cannot be assumed to apply to other emulators. Furthermore the emulator was only tested on one operating system, other systems may affect the result as well. The testing is concerned with reads and writes to memory and computation speed, there are several other things to test, such as file access or network speeds. The tests was only executed on a HTC Magic device, other devices may have different results, furthermore devices with faster CPUs has also become available, such as the HTC Desire with a 1GHz CPU. In the Method section, causes for the results has been suggested, but not examined, as it exceeded the scope of this article.

## 6. ACKNOWLEDGEMENTS

Thanks to Morten Bested for early reviewing.

## REFERENCES

Android. Android emulator. `http://developer.android.com/guide/developing/tools/emulator.html`, 2010a. Last seen: 3th of Marts, 2010.

Android. Timinglogger. `http://developer.android.com/reference/android/util/TimingLogger.html`, 2010b. Last seen: 9th of Marts, 2010.

HTC Corporation. Htc - products - htc magic - specifications. `http://www.htc.com/europe/product/magic/specification.html`, 2010. Last seen: 3th of Marts, 2010.

## Appendix A. DATA

The means of the ten runs is listed in the tables bellow. All data is in milliseconds.

| | |
|---|---|
| **Emulator** | 210,2 |
| **Device** | 198.5 |

Table A.1. Data for computation test

| | **Emulator** | | **Device** | |
|---|---|---|---|---|
| # | Write | Read | Write | Read |
| 20 | 0.2 | 3.7 | 0.2 | 1.3 |
| 200 | 0.2 | 2.9 | 0.4 | 1.2 |
| 2,000 | 1.7 | 2.8 | 2.3 | 1.9 |
| 20,000 | 16.7 | 11.3 | 13.6 | 7.2 |
| 200,000 | 155.1 | 80.2 | 145.7 | 53.6 |
| 2,000,000 | 1469.9 | 779.1 | 1188.8 | 485.2 |

Table A.2. Data for memory tests