ART and Dalvik performance compared

Tobias Konradsson
Abstract

In 2014 Google released a new version of their mobile operating system Android where one of the features was the change from Dalvik to Android Runtime (ART). Dalvik and ART are two runtimes that are used by Android applications and some of the system services. The purpose was to see where ART and Dalvik differs when it comes to applications size, RAM usage and performance. The methods used for deducing there differences where to install multiple applications and then examine the applications size and the RAM usage. Also, benchmark applications was used to measure the performance between ART and Dalvik. The results showed that ART was better when it comes to performance and RAM usage. Interestingly there was a few drawbacks with ART. One of them were that application sizes where larger compared to Dalvik.
Acknowledgements

I would like to thank my girlfriend, Felicia, who supported and helped me through this journey. I would also like to thank my family for always being there for me. Last but not least, thank you Andrew Wallace for guiding and supervising me during this project.
# Contents

1 Introduction 1  
2 Background 3  
  2.1 Dalvik 3  
  2.2 Android Runtime 3  
3 Method 5  
  3.1 Application size 5  
  3.2 RAM usage 5  
  3.3 Performance of the virtual machine 5  
4 Test results 7  
  4.1 Application size 7  
    4.1.1 Result 7  
    4.1.2 Discussion 8  
  4.2 RAM 8  
    4.2.1 Result 9  
    4.2.2 Discussion 9  
  4.3 Performance 10  
    4.3.1 Linpack 10  
    4.3.2 Real PI 10  
    4.3.3 Quadrant 11  
    4.3.4 AnTuTu 11  
    4.3.5 CF-Bench 12  
    4.3.6 Performance discussion 12  
5 Conclusion 15
1 Introduction

The T-Mobile G1 launched October 2008 and was the first mobile phone with Android as an operating system. It is a software created by Andy Rubin, Rich Miner, Nick Sears and Chris White in 2003, and was intended to be used as an advanced operating system for digital cameras. Though, when it was realized that the market for this was not large enough, the company instead started to develop Android as a mobile system that would rival, at that time, Symbian and Microsoft Windows Mobile.

In 2005 Google acquired Android and have since developed it.

Android is the largest mobile operating system in the world, in 2014 it had a market share of 78.0% [1]. 22 mobile companies are using Android as a mobile OS [2]. The largest of those are Samsung, HTC, Sony, LG, Motorola, Huawei, Pantech and ZTE.

A lot of low-end Android devices often have less storage, less RAM and a slower CPU. It is these phones that are affected the most since the more expensive phones can solve the performance with expensive hardware, while low-end phones can not.

The new ART is supposed to be faster than Dalvik. Therefore this thesis aims to find where Dalvik and ART differs in terms of performance in a low-end Android smartphone. The questions below will be used to answer the issue of this thesis.

• What are the differences between Dalvik and ART when it comes to:
  – Application size?
  – Ram usage?
  – Performance?

• Does ART have any drawbacks compared to Dalvik, what are they?
2 Background

In this section the two virtual machines that will be compared are presented.

2.1 Dalvik

Before Android 5.0 (Lollipop) Dalvik was used as the virtual machine to execute applications written for Android. Applications for android is commonly written in Java, since the most APIs are Java only, and then compiled to bytecode for the virtual machine. Dalvik uses just-in-time compilation which is a technique to increase execution speed of applications by compiling parts of an application to machine code at runtime. Since mobile phones did not have a lot RAM when Android was developed, Dalvik tried to make sure that the RAM management was as effective as possible.

2.2 Android Runtime

ART was introduced in Android 4.4 (KitKat) as an experimental runtime for developers so they could test their applications and make sure they where working. Some users also enabled it since it improved the performance a bit, even though ART was still in development. This caused some applications to stop working for those who changed the runtime from Dalvik to ART, a list of those applications can be found of the forum Reddit [3].

But 2014 when Android 5.0 (Lollipop) was released, ART was ready to be used as the standard virtual machine. While Dalvik compiles the code on the fly, ART compiles the code when the application is installed. The reason for the change to ART was to increase the speed of the code even more than Dalvik, improve garbage collection, better support for multicore processors and implement 64 bit support. Therefore the performance will be a lot better [4].
3 Method

Application size, RAM usage and performance of the virtual machine is what is going to be examined, and will be given an explanation of how they will be tested.

3.1 Application size

To test the size of applications three simple applications will be created. One application with the minimum code to work, this will give a base application that can be used as a reference. Two other applications will be created based on the minimum application, one with added code and one with added images. This so that it will be clear if code or images use more space on ART compared to Dalvik. Four real applications will also be tested to see how the size differs. The applications used will be Blogger [5], Drive [6], Fitness [7], Gmail [8].

3.2 RAM usage

To test the RAM usage in Dalvik and ART, two Android emulators will be used. One emulator running Android 4.4.2 (KitKat), this version of Android is using Dalvik. The other emulator is running Android 5.1.1 (Lollipop) that is using ART.

Android Device Monitor, which is a program that comes with the Android SDK, will then be used to analyze the RAM usage of six common Android applications and the reference application that only have enough code to start. The applications are: Drive [6], Gmail [8], Whatsapp [9], Netflix [10], Dropbox [11] and Skype [12].

3.3 Performance of the virtual machine

Measuring the speed of a device is hard since there might be processes that is running in the background and makes the result become less exact, but the best way is to use a benchmark application that will run multiple times to give an average value that can be used to compare Dalviks and ARTs performance.
Some Android applications can be written in Java but also in C and C++, but since Dalvik and ART only uses applications written in Java, benchmark applications written in C or C++ will not be used for testing.

To get a good estimate of how the speed differs, a new install of Android is used. Since a lot of applications are installed with Android, as many as possible was uninstalled or inactivated. This will ensure that there are as few applications running in the background as possible. The phone was also in airplane mode to eliminate disturbances, but some tests required an Internet connection so the Wi-Fi was left on.

When testing Dalvik a LG Google Nexus 5 was used with original Android 4.4.4 (KitKat). This is the latest version of Android that still uses Dalvik. KitKat also have the ability to use ART, but since it was not the standard runtime, just a preview for developers, the latest version of Android 5.1 (Lollipop) was used instead to test ART. To test the speed, a few benchmark applications will be used: Linpack [16], Real PI [17], Quadrant [18], AnTuTu [19], CF-Bench [20]. Each benchmark application will run five times, then the average will be calculated. This is because the phone is running tasks in the background that can interfere with the tests, running them multiple times will reduce these interferences.
4 Test results

In this section, the results of the tests are presented. First is the test that tests how the size of Android applications differs when installed on ART and Dalvik. Second is the test that test which of Dalvik and ART uses the least RAM when running Android applications. Last is the performance tests, these will show how Dalvik and ART performs when doing heavy calculations. After each test part, the results will be discussed.

4.1 Application size

When applications is installed on Dalvik, they are not compiled as they are on ART, instead applications are compiled when they are started. This should mean that installed applications on ART will use more storage than the same application installed on Dalvik. This can be seen in the table below that shows the applications sizes.

4.1.1 Result

<table>
<thead>
<tr>
<th></th>
<th>Dalvik</th>
<th>ART</th>
<th>Difference</th>
<th>ART/Dalvik</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>3.11 MB</td>
<td>4.63 MB</td>
<td>1.52 MB</td>
<td>49%</td>
</tr>
<tr>
<td>Media</td>
<td>14.42 MB</td>
<td>15.94 MB</td>
<td>1.52 MB</td>
<td>11%</td>
</tr>
<tr>
<td>Code</td>
<td>6.13 MB</td>
<td>8.14 MB</td>
<td>2.01 MB</td>
<td>33%</td>
</tr>
<tr>
<td>Blogger</td>
<td>12.93 MB</td>
<td>22.08 MB</td>
<td>9.15 MB</td>
<td>71%</td>
</tr>
<tr>
<td>Drive</td>
<td>23.31 MB</td>
<td>33.02 MB</td>
<td>9.71 MB</td>
<td>42%</td>
</tr>
<tr>
<td>Fitness</td>
<td>13.07 MB</td>
<td>17.46 MB</td>
<td>4.39 MB</td>
<td>34%</td>
</tr>
<tr>
<td>Gmail</td>
<td>15.55 MB</td>
<td>22.87 MB</td>
<td>7.32 MB</td>
<td>47%</td>
</tr>
</tbody>
</table>

Table 1: The size of applications on Dalvik and ART. Also the difference between Dalvik and ART.

The Minimum application was created to have as little code and images as possible, that way this application can be used as a reference point when testing other applications. The application only has enough code to start. We can quickly see in table 1 that the Minimum uses 1.52 MB more space on ART compared to Dalvik. When installed on ART the application is 49% larger. This is because on ART the applications code is compiled, while on
Dalvik it is not.

The Media application is based on the Minimum application, with an extra 11.2 MB of images. As we can see in table 1 there is, just like the Minimum, 1.52 MB of difference in size. The test shows that images does not use more space on ART compared to Dalvik.

The Code application is also based on the Minimum application but have instead of images, an additional 50 000 lines of code. The extra lines of code is 2.31 MB before built into an .apk file.
In table 1 we can see that the size difference is larger than in the Minimum application. The difference between Dalvik and ART is 2.01 MB.

We can see that the difference for Blogger and Drive is almost 10 MB which is a lot for a mobile application. The Blogger application is 71% larger on a device with ART compared to a device with Dalvik. In table 1 the size difference is smaller for Fitness where the size on ART is only 4.39 MB larger, but for Gmail the size is 7.32 MB larger on ART. This means that Fitness uses 34% more space on ART, while Gmail uses 47% more on ART.

4.1.2 Discussion

There is a rather big difference between Dalvik and ART. For example the application Blogger, the size is 71% larger on ART than on Dalvik. Although the average size difference is 49%. For Android phones with limited storage this can cause a problem, since the phones storage will fill up faster. What is interesting is that the Minimum application is 49% bigger on ART while the Code application is only 33% bigger. Even though it is the code that was added that becomes bigger on ART.

And what we can see from the first three rows in the table, the difference for the Media application is the same as the Minimum application. This means that images have the same size on both Dalvik and ART. But the size of the Code application is larger on ART than on Dalvik. This is because the applications are compiled when they are installed on ART, while on Dalvik they are not. This means that if a developer want to support new Android devices with an Android version higher than 5.0, and that might not have much storage, they have to try to keep the size to the minimum, especially the size of the code. Of course, the developer can not just delete code to make the application smaller.

4.2 RAM

Android Device Monitor, which is a program that comes with the Android SDK, was used to test the RAM usage off the applications Drive, Gmail, Whatsapp, Netflix, Dropbox, Skype and the reference application Minimum.
4.2.1 Result

<table>
<thead>
<tr>
<th>Application</th>
<th>ART (MB)</th>
<th>Dalvik (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>0.725</td>
<td>3.302</td>
</tr>
<tr>
<td>Drive</td>
<td>13.248</td>
<td>16.607</td>
</tr>
<tr>
<td>Gmail</td>
<td>2.906</td>
<td>8.034</td>
</tr>
<tr>
<td>Whatsapp</td>
<td>9.441</td>
<td>10.928</td>
</tr>
<tr>
<td>Netflix</td>
<td>12.161</td>
<td>15.287</td>
</tr>
<tr>
<td>Dropbox</td>
<td>12.183</td>
<td>15.046</td>
</tr>
<tr>
<td>Skype</td>
<td>19.104</td>
<td>22.599</td>
</tr>
</tbody>
</table>

Table 2: The amount of allocated RAM for ART and Dalvik.

To test the RAM usage, seven different application are used, the first one, Minimum, is the same application that was used when testing the size. As can be seen in table 2, the Minimum application uses 355% more RAM when running on Dalvik instead of ART. Gmail also have a big difference in RAM usage where Dalvik uses 176% more RAM than ART. Although the difference for the Minimum application is only 2.577 MB and the difference for Gmail is 4.128 MB which is larger, as the test results shows in table 2. On the other applications RAM usage is more similar on Dalvik and ART. The difference for Drive and Whatsapp are 25% and 16% higher on Dalvik respectively. While the difference for Netflix and Dropbox are 26% and 24%. Furthermore the difference for Skype and Spotify are 18% and 32%.

On average, Dalvik uses 2.9 MB more RAM than ART, that is a 84% higher RAM usage on Dalvik. If the Minimum application is not taken into the calculation Dalvik still uses on average 2.9 MB more RAM, then it is 45% more than ART.

4.2.2 Discussion

What we can see in table 2 on page 9 is that ART has an overall lower RAM usage than Dalvik. This means that an Android phone can not run as many applications at the same time when using Dalvik since the applications have a higher RAM usage.

Other sources like a blog post on New Relic [13] and an article on Anand Tech [14] confirms the results in table 2 on page 9 that the RAM usage is lower on ART than on Dalvik. Although an article on Infinum [15] disagrees and writes that ART has a larger memory footprint which contradicts what the results of the tests shows in section 4.2. It is difficult to know why this is, they might have different sources or other methods.
4.3 Performance

In this Section, the five benchmark applications will be tested to see if there are a performance difference between ART and Dalvik.

4.3.1 Linpack

Jack Dongarra [21] introduced the Linpack Benchmark that is used to solve a dense system of linear equations. Linpack is used to measure the computational power of supercomputers and rank them. [22]

Linpack measures how many FLOPS, Floating-Point Operations Per Second, a computer can perform.

The tests below are done with Mobile Linpack [16], an application for Android. For these tests a system of linear equations with the size of 1000x1000 was used, this to make sure that the test would run for a while for a more even result.

<table>
<thead>
<tr>
<th>MFLOPS</th>
<th>Seconds</th>
<th>ART</th>
<th>Dalvik</th>
<th>ART</th>
<th>Dalvik</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>72.82</td>
<td>53.02</td>
<td>9.18</td>
<td>12.61</td>
<td></td>
</tr>
<tr>
<td>Test 2</td>
<td>73.58</td>
<td>74.08</td>
<td>9.09</td>
<td>9.03</td>
<td></td>
</tr>
<tr>
<td>Test 3</td>
<td>68.38</td>
<td>68.99</td>
<td>9.78</td>
<td>9.69</td>
<td></td>
</tr>
<tr>
<td>Test 4</td>
<td>76.98</td>
<td>65.83</td>
<td>8.69</td>
<td>10.16</td>
<td></td>
</tr>
<tr>
<td>Test 5</td>
<td>71.97</td>
<td>61.84</td>
<td>9.29</td>
<td>10.81</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>72.75</td>
<td>64.75</td>
<td>9.21</td>
<td>10.46</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Test performance results by Linpack between ART and Dalvik

As we can see in the table above the average number of FLOPS for ART is 72.75 million FLOPS, while the average number of FLOPS for Dalvik is 64.75. This means that ART performed 12.35% better than Dalvik.

4.3.2 Real PI

Real PI [17] is an application that measures how long it will take to calculate a chosen number of decimals of pi. This is a good way to test both the CPU and the RAM speed. Real PI have two different algorithms to do this, one is AGM + FFT, and the other is Machin’s. But since AGM + FFT runs as native C++ code it will not be used. The Machin’s on the other hand is written in Java and will then produce the more interesting results because it is using Dalvik or ART.
<table>
<thead>
<tr>
<th></th>
<th>ART</th>
<th>Dalvik</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>46.59</td>
<td>45.05</td>
</tr>
<tr>
<td>Test 2</td>
<td>45.91</td>
<td>44.96</td>
</tr>
<tr>
<td>Test 3</td>
<td>46.70</td>
<td>44.92</td>
</tr>
<tr>
<td>Test 4</td>
<td>46.94</td>
<td>44.81</td>
</tr>
<tr>
<td>Test 5</td>
<td>46.85</td>
<td>44.82</td>
</tr>
<tr>
<td>Average</td>
<td>46.54</td>
<td>44.91</td>
</tr>
</tbody>
</table>

Table 4: Test performance results by Real PI between ART and Dalvik

As can be seen in the table above ART took on average 1.63 seconds longer to calculate the 10 000 first decimals of PI. This means that Dalvik performed 3.6% better than ART.

4.3.3 Quadrant

Quadrant [18] tests the device’s CPU by running arithmetic operations, XML parsing and multimedia decoding. Quadrant also test the memory throughput and the I/O by accessing the file system and running database operations [23].

<table>
<thead>
<tr>
<th></th>
<th>CPU</th>
<th>RAM</th>
<th>I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>ART</td>
<td>Dalvik</td>
<td>ART</td>
<td>Dalvik</td>
</tr>
<tr>
<td>Test 1</td>
<td>37836</td>
<td>20061</td>
<td>14413</td>
</tr>
<tr>
<td>Test 2</td>
<td>39305</td>
<td>20244</td>
<td>14354</td>
</tr>
<tr>
<td>Test 3</td>
<td>38115</td>
<td>20096</td>
<td>15094</td>
</tr>
<tr>
<td>Test 4</td>
<td>39632</td>
<td>20341</td>
<td>14965</td>
</tr>
<tr>
<td>Test 5</td>
<td>38808</td>
<td>20421</td>
<td>13846</td>
</tr>
<tr>
<td>Average</td>
<td>38739</td>
<td>20233</td>
<td>14534</td>
</tr>
</tbody>
</table>

Table 5: Test performance results by Quadrant between ART and Dalvik

In these tests ART scored on average 91% higher than Dalvik when testing the CPU. When testing the RAM performance, ART scored 8% better than Dalvik, but in the I/O category Dalvik scored 31% better than ART.

4.3.4 AnTuTu

AnTuTu tests many parts of the device, but since ART and Dalvik is in focus, the most interesting is Runtime. The Runtime score tells us how good ART and Dalvik performs when running Android applications. AnTuTu also test how well the CPU can do integer and float calculations [24].
<table>
<thead>
<tr>
<th></th>
<th>Runtime</th>
<th>Int</th>
<th>Float</th>
</tr>
</thead>
<tbody>
<tr>
<td>ART</td>
<td>Dalvik</td>
<td>ART</td>
<td>Dalvik</td>
</tr>
<tr>
<td>Test 1</td>
<td>3812</td>
<td>1738</td>
<td>2976</td>
</tr>
<tr>
<td>Test 2</td>
<td>3860</td>
<td>2234</td>
<td>3331</td>
</tr>
<tr>
<td>Test 3</td>
<td>3860</td>
<td>2248</td>
<td>3463</td>
</tr>
<tr>
<td>Test 4</td>
<td>3891</td>
<td>2004</td>
<td>3312</td>
</tr>
<tr>
<td>Test 5</td>
<td>3852</td>
<td>2235</td>
<td>3450</td>
</tr>
<tr>
<td>Average</td>
<td>3855</td>
<td>2092</td>
<td>3306</td>
</tr>
</tbody>
</table>

Table 6: Test performance results by AnTuTu between ART and Dalvik

In this test ART performs on average 84% better than Dalvik in the Runtime category, 17% better in Integer, and 18% better in float.

### 4.3.5 CF-Bench

CF-Bench is developed by the xda-developers user Chainfire [25]. The reason CF-Bench is used is because it supports multiple cores, this means that it will deliver a more accurate performance reading. [26]. CF-Bench tests run both in Java and C++. The tests below only shows some of the results for the Java code since it is the Java code that is run in the virtual machines.

<table>
<thead>
<tr>
<th></th>
<th>MIPS</th>
<th>RAM read</th>
<th>RAM write</th>
</tr>
</thead>
<tbody>
<tr>
<td>ART</td>
<td>Dalvik</td>
<td>ART</td>
<td>Dalvik</td>
</tr>
<tr>
<td>Test 1</td>
<td>1759</td>
<td>779</td>
<td>1433</td>
</tr>
<tr>
<td>Test 2</td>
<td>1817</td>
<td>897</td>
<td>1502</td>
</tr>
<tr>
<td>Test 3</td>
<td>1749</td>
<td>902</td>
<td>1471</td>
</tr>
<tr>
<td>Test 4</td>
<td>1850</td>
<td>840</td>
<td>1532</td>
</tr>
<tr>
<td>Test 5</td>
<td>1750</td>
<td>911</td>
<td>1228</td>
</tr>
<tr>
<td>Average</td>
<td>1785</td>
<td>865.8</td>
<td>1433.2</td>
</tr>
</tbody>
</table>

Table 7: Test performance results by CF-Bench between ART and Dalvik

What these tests show is that ART have more than twice MIPS (Million Instructions Per Second) than Dalvik. This is a big difference. On the other hand Dalvik performs better than ART when writing to and reading from RAM. Dalvik gets a 38.39% higher score when reading from RAM than ART, and a 6.57% higher score when writing to RAM.

### 4.3.6 Performance discussion

The performance tests show that ART performs better in almost all categories. Especially the tests in Quadrant, AnTuTu and CF-Bench. Overall ART scored 38.9% higher than Dalvik. Although in Quadrant when testing I/O Dalvik
scored 30.49% higher than ART, and in CF-Bench Dalvik outperformed ART when reading from and writing to RAM. Interestingly, ART performed better in the RAM category in Quadrant.
5 Conclusion

The new ART comes with a lot of advantages compared to Dalvik, but one of the drawbacks are that application sizes are larger on ART. On average, as seen in section 4.1, applications installed on ART is 49% larger than the same application installed on Dalvik. The application Blogger is 71% larger when installed on ART compared to when installed on Dalvik. For Android phones with limited storage this can cause a problem, since the phones storage will fill up faster. It is not uncommon for low-end Android phones to have around 8 GB of storage, and since the operating system uses a lot of that space, not much is left for applications. Therefore this can be a big drawback to update those phones to ART.

On the other hand, applications running on ART does not use as much RAM as the same application running on Dalvik. This is good since when applications are closed in Android, they are saved in RAM. That way when the application is started again, it will load faster since it is already loaded into RAM. This will make the smartphone faster when switching between applications.

The performance tests show that Dalvik performs better when doing I/O operations, this means that loading times can be higher on phones using ART. Dalvik also performs better when reading from RAM, this can also increase the loading times for ART. But ART is a lot faster when it comes to CPU calculations. This means that applications on the phone seems to run faster [24].

I would recommend the change to ART since it offers more advantages than disadvantages. As the test results shows, the Android smartphone will be faster overall. Who does not want a fast phone?
References


New Relic: ART vs. Dalvik [Online]. Available: 
https://blog.newrelic.com/2014/07/07/android-art-vs-dalvik/

(ART) in Android L [Online]. Available: 

runtime in KitKat [Online]. Available: 
https://www.infinum.co/the-capsized-eight/articles/art-vs-dalvik-introducing-the-new-android-runtime-in-kit-kat


Available: 


http://www.top500.org/project/authors/jack-dongarra/

http://www.top500.org/project/linpack/

leased [Online]. Available: 
http://www.aurorasoftworks.com/
