Plagiarism Scan Report

Report Generated on: Apr 27,2023

0%

Plagiarised

100%

Unique

Total Words:	**997**
Total Characters:	**5813**
Plagiarized Sentences:	**0**
Unique Sentences:	**45 (100%)**

Content Checked for Plagiarism

ART of Android

Introduction

ART acronym of Android Run Time is current processing virtual machines in the Android Software Stack. It is the Runtime Virtual Machine which sits above the Hardware Abstraction Layer to process the High-Level Language byte code to the Machine Language that Android understands. It is used to execute the applications written on Android. So, you can see ART as the ART of Android to execute the applications on the Android Platform by using less battery, and runtime memory. It is the ART of android to talk in Machine Language.

History

ART is a new and latest android runtime being introduced as a beta in Android KitKat and finally became an integral part of the Android Software Stack for later versions and versions of Android Lollipop. So, if the Android Runtime was introduced in Android KitKat then how were the applications executed in the older versions of Android? – The answer is Dalvik.

Dalvik's Android Era

What is Dalvik? -Dalvik is the Android processing Virtual Machine which was used to execute the applications on the Android Versions older than Android KitKat 4.4 release. Dalvik is the ancestor of ART hence the byte code of application designed for Dalvik is also compatible with the ART, hence ART supports backward compatibility. Dalvik was developed by Dan Bornstein and has origins in a town in Iceland named Dalvik, hence the name of the virtual machine originated from that town.

Dalvik's Architecture

Dalvik is a Just in Time Compiler (JIT) – JIT means that whenever the compiler executes the application code, it only compiles the required code that is needed at that moment and the rest of the code is compiled later whenever required. Since it compiles only a part of the code it has a smaller memory footprint and hence it makes use of less physical memory on the device, which increases the performance of the device. Since everything in Android Mobile phones is limited whether it be the battery life, processing and memory etc. Dalvik Virtual Machine (DVM) optimizes the code so that it can fit in low powered devices. Programs for Android are mainly written and processed in java which is compiled to Java Bytecode for JVM and then translated to. dex (Dalvik Executable) or. odex (Optimized Dalvik Executable) format which is understood by Dalvik and executed by DVM in low powered mobile devices.

ART's Android Era

ART is the latest Android Runtime. It is derived from Dalvik and hence it understands the. dex or. odex translated bytecodes. ART has replaced the DVM from the Android Software Stack. ART was introduced to improve the performance of High-End Applications which requires larger memory spaces and consumes a large amount of battery. ART was introduced to overcome the Drawbacks of Dalvik.

ART's Architecture

ART is an Ahead of Time (AOT) Compiler – An AOT compiler compiles the DEX code at the time of

installation hence called Ahead of time. The DEX compiled bytecode is compiled and stored as. oat (Of Ahead Time) format -The OAT format is a native code which is directly understood by the system. This is a one-time event that happens when the application is installed into the system. So, whenever the application is executed next time it runs faster as the whole code is compiled earlier. As the system only needs to follow the instructions written on the native code the execution is much faster because of less runtime CPU usage.

The architecture of DVM and ART

Benefits of ART

With Dalvik, the problem was that it is a JIT compiler which means that it compiles the code every time the application is required to be executed. Hence it uses a lot of CPU, and battery and a lot of garbage data get created which leads to a decrease in the performance of the device. But the benefits of using ART over the Dalvik are that It has better and improved Garbage Collection which reduces the redundancy and it is an AOT compiler hence the code is compiled at the time of installation and whenever the application is executed next time it doesn't require compilation which leads to less CPU usage, less Battery Usage, less redundancy and as the application is executed from the native code after installation hence less Application start up time which leads to increase in the performance of the device.

Drawbacks of ART

Who can think that this system would also have drawbacks? But it has, as it is an AOT compiler it compiles once and hence native code must be stored it requires larger physical memory space. Hence devices having less physical space can use Dalvik as it compiles only the required code and hence uses less memory space. It compiles at the time of installation hence installation time also increases as the DEX bytecodes are compiled to Machine code during installation. The Booting time of Android is also more as compared to Dalvik because it uses the concept of AOT.

AOT? or JIT? Why not Both?

So Why use only AOT or JIT why not use Both AOT and JIT. So, Google introduced an ART JIT compiler from Android 7.0 (Nougat), which adds a JIT compiler with code profiling to the ART that increases the performance of the applications according to the usage of the application. Both in coordination improves the performance of the application by overcoming each other's drawbacks. The use of the ART JIT compiler saves the memory spaces, decreases the installation time, increases the booting time and increases the app update time.

ART JIT's Architecture

ART JIT works on both the approach – AOT and JIT. If the compiled native code is available from the DEX bytecode the ART executes the code and if the DEX code is updated the running app is updated using a JIT compiler by compiling the code that it is currently in use first. And the rest of the code is compiled afterwards and the .oat files are updated.

No Plagiarism Found