IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 09, Issue 5 (May. 2019), ||S (VI) || PP 06-10

Comparative Study on Different Cross Platform Technologies

Shubhojeet Chatterjee[1], Laxmi Dodke [2], Pawan Thakurpelne[3], Shekhar Salam[3]
[1]*(ETC, Jhulelal Insitute of Technology / RTMNU, India)*
[2]*(ETC, Jhulelal Insitute of Technology / RTMNU, India)*
[3]*(ETC, Jhulelal Insitute of Technology / RTMNU, India)*
[4]*(ETC, Jhulelal Insitute of Technology / RTMNU, India)*

Abstract : *Cross Platform Application Development is the development of applications so that applications can be made platform-independent. A review has been made in this field while considering Integrated Development Environment is being proposed which will help a software developer to code an application in a single code base and deploy that single code base to multiple operating systems meaning if a developer codes in Java for Android, the developer can deploy the same Java code to iOS[5] subsystem. First an OS needs to be developed which has a compatible architecture which helped to run unmodified iOS binaries on Android operating system. The second phase of proposed solution helps to understands the cross-platform application development tools which are currently available in the market. The third step is to thoroughly understand cross-platform development tools and finally a mathematical model based application. Thus this project deals with creation of Bluetooth file transfer application which is universally acceptable in any platform.*

Keyword:

I. Introduction

Application development in recent years has seen an exponential growth. Today each and every person in this world has a smart-phone in his pocket. Smartphone's combine a range of functions such as media players, camera and GPS with advanced computing abilities and touch screens are enjoying ever increasing popularity. Smartphone's help us to achieve a range of tasks through something known as applications or Apps to short. According to Gartner [1] Google's Android, Apple's iOS and Microsoft's windows all are leading players in todays market.

II. Methodology

2.1 CIDER

CIDER is basically an operating system compatibility architecture that can run applications built for different ecosystems preferably iOS, Android and Windows on the same Smartphone or tablet and PC. Basically in simpler terms CIDER had the ability to run unmodified iOS binaries on the Android and Windows subsystem without any sort of modification. Cider achieves the task of increasing the capacity of home Android kernel by simultaneously using the home kernel and the slave kernel which is the application binary interface in our case. User space of the slave kernel gets in contact with the Cider enabled kernel in exactly the same ways as the slave kernel. That is, the iOS applications get in to Linux based kernel approach as if they are working on a home kernel of iOS subsystem which is running on a typical iOS based device. Instance of a foreign kernel, and reuse and run unmodified foreign user space library code. Now coming to the architecture of these two operating systems. iOS runs on ARM CPUs like Android, but has a very different software ecosystem. iOS is built on the XNU[5] kernel, a hybrid combination of a monolithic BSD kernel and a Mach microkernel[3] running in a single kernel address space. CIDER is basically an operating system compatibility architecture that can run applications built for different mobile ecosystems preferably iOS and Android together on the same Smartphone or tablet. Basically in simpler terms CIDER had the ability to run unmodified iOS binaries on the Android subsystem without any sort of modification. Cider achieves the task of increasing the capacity of home Android kernel by simultaneously using the home kernel and the slave kernel which is the application binary interface in our case. User space of the slave kernel gets in contact with the Cider enabled kernel in exactly the same ways as the slave kernel. That is, the iOS applications get in to Linux based kernel approach as if they are working on a home kernel of iOS subsystem which is running on a typical iOS based device. Instance of a foreign kernel, and reuse and run unmodified foreign user space library code. Now coming to the architecture of these two operating systems. iOS runs on ARM CPUs like Android, but has a very different software ecosystem. iOS is built on the XNU[8] kernel, a hybrid combination of a monolithic BSD kernel and a Mach microkernel[3] running in a single kernel address space When we talk about Android, Each Android app is compiled into

Dalvik[2] byte code (dex) format, and runs in a separate Dalvik VM instance. Now Comparison of latest cross-platform mo-bile application development approaches which are currently available in the market. Some of the cross-platform mobile application development approaches are Phone Gap [1], Titanium etc. The distinguish between approaches that employ a run-time environment and those that generate platform-specific apps from a common code base at compile time. The latter, generator-based category includes model driven solutions and cross-compiling. Up to now, there are no production-ready solutions of this category. Hence, till this types of approaches are concentrates on cross-platform solutions that combine the source code of an app with a runtime environment. Some of the examples of cross-platform application development tools are Phone Gap, which is a Hybrid framework and Titanium. The most prominent hybrid framework till date for cross platform application development is Phone Gap. Phone Gap was created by Nitobi Soft-ware, which has been acquired by Adobe. The development now takes place in the Apache Cordova project of the Apache Foundation, of which Phone Gap is a distribution. Phone Gap a popular cross-platform mobile application development tool is loosely based on jQuery which is a very fast library conversion tool.

2.2. MINISTRO

Ministro is a central repository for Qt shared libraries. It acts as a bridge between apps and Qtlibraries.The connection API protocol used by applications to connect and query the service is released under BSD license, it mean that any application can connect and query its services.

Minitro can also be used to reduce the size of application. It enables several applications to share the libraries. To use this deployment method, one must set up a repository for the libraries that one want to distribute.

III. Bt Transfer Using Qt

Aim: To create an application which can transfer files using Bluetooth on cross-platform protocol.
Objective: To create an application which is more which is more convenient and faster than exiting Bluetooth OS compatibility architecture (CIDER).

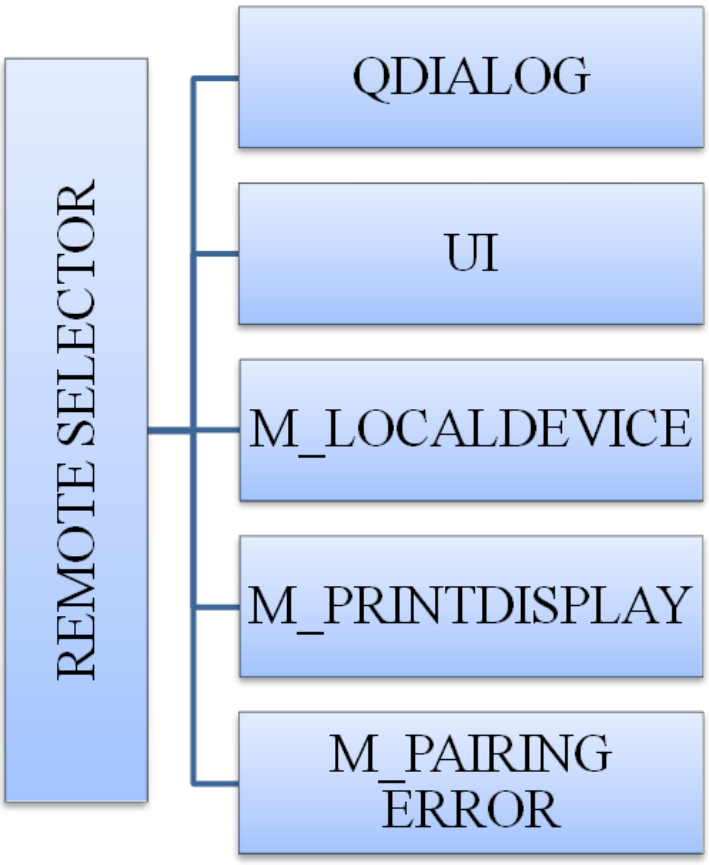

Figure 1 flow chart of application

The above shown figure represents a flow chart of the system working. It basically follows the principle multilevel inheritance, here Qdialog, UI, M_localdevice M_printdisplayM_pairingError act as a parent class information from which is inherited by Remote Selector which is an derived class.

Q Dialog: this Function focusses on designing of the output box ie. Selection of of file option number of rows and columns for devices start and refresh options. Basically it provides us the whole view of our output screen.

UI: design motion images. Talking in reference to our program it shows the gif of transfer process or a motion circle depicted for scanning. Unlike other compilers it has an advance tools for designing such gif files which are then converted into code automatically. Thus using these tootls we have designed the transfer % and pincode pop-up.

M-local device: This enables us the option to select between local device and external Bluetooth peripheral. Once it gets access to the Bluetooth it scans all the nearby device and display them on the output screen. Thus this creates a gateway between device Bluetooth and the application.

M_Print display: This gives access to dialogue box function. Using this function we can upload a document then we have a function to send the same. It can also control the duration for which the Bluetooth scans with refresh and stop option.

M_Pairing Error: This function basically provides a error detection conditioning. It's a pop up generally seen when there is a transfer failure, it also activates when there is an error in connection via Bluetooth.

OBSERVATION

Table 1 DATA TRANSFER RATES IN i3-5 GEN PROCESSOR

DATA (IN MEGA BYTE)	TIME TAKEN BY Qt APPLICATION (SECONDS)	TIME TAKEN BY LOCAL APPLICATION (SECONDS)
200	1231.8	3220.2
1046	6553.176	16841.646
456	2808.504	7486.965
743	4576.137	11963.043
58	357.222	933.858

Table 2 DATA TRANSFER RATES IN i7-8 GEN PROCESSOR

DATA (IN MEGA BYTE)	TIME TAKEN BY Qt APPLICATION (SECONDS)	TIME TAKEN BY LOCAL APPLICATION (SECONDS)
200	338.98	1041
1046	1772.881	5447.91
456	772.881	2375
743	1259.322	3869.791
58	98.30	302.0833

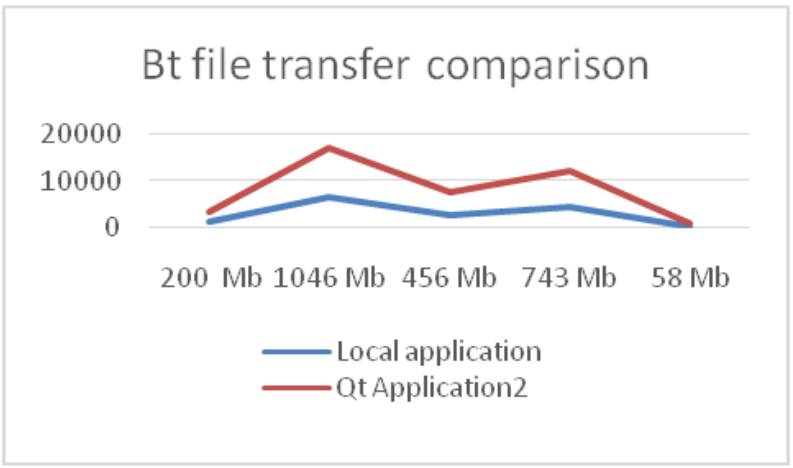

Figure 2 DATA TRANSFER RATES IN i3-5 GEN

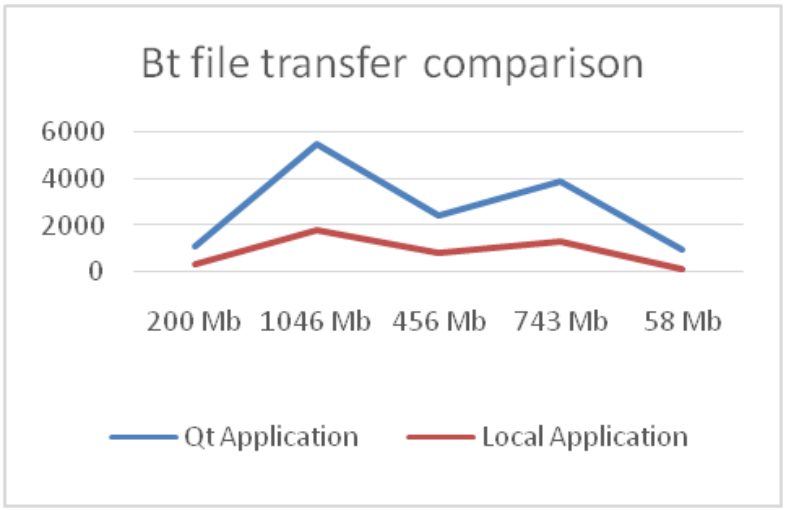

Figure 3 DATA TRANSFER RATES IN i7-8 GEN PROCESSOR

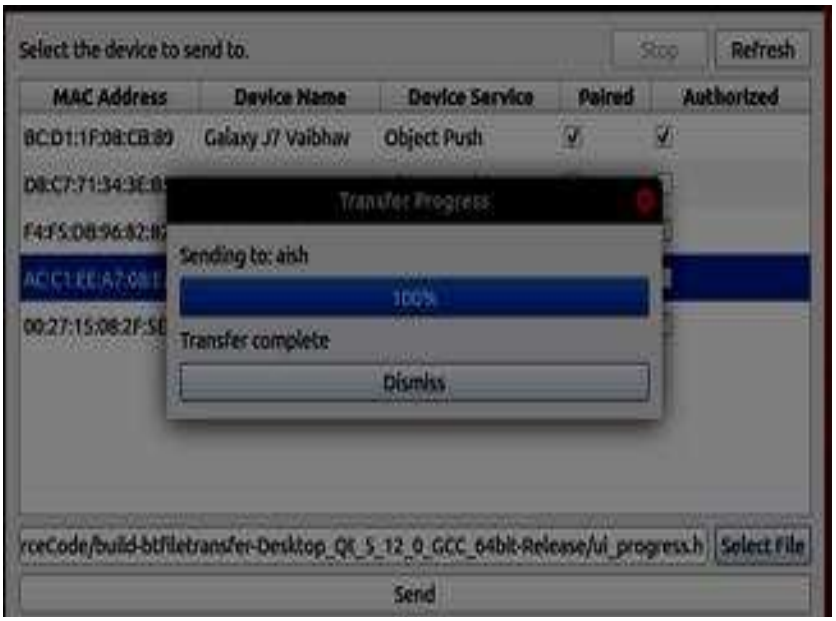

Figure 4 O/P screen

IV. Algorithm

Equation 1 FORMULA FOR CALCULATING SPEED

$$SPEED = \frac{DISTANCE}{TIME}$$

Equation 2 FORMULA FOR CALCULATING TRANSFER SPEED

$$SPEED = \frac{DATA\ IN\ Kb}{TIME\ NEEDED\ TO\ SEND\ THE\ FILE}$$

CALCULATION

Therefore it's seen that when used in an i3-5 the approximate speed attained is 330 Kb/s
When computed in i7-8 gen device approximate speed attained is 590Kb/s.

V. Conclusion

Thus it's seen that Qt application is better than cyderwhich is presently been used for cross-platform communication. A speed increment of around 250 Kb/s is seen when Qt application is used. To corroborate this result test has been carried out on 2 devices with variable specification.

Use of cross platform technologies can create a market which is focused towards logical and business development rather than competing with one another for different OS development.

References

[1]. AboutPhoneGap (2011), http://phonegap.com/about
[2]. Android Development Tools plugin for Eclipse (2012),http://developer.android.com/sdk/eclipse-adt.html
[3]. D Keuper, (s1019775, XNU a security evaluation), University of Twente and Certified Secure, December 13, 2012.
[4]. http://developer.apple.com/members
[5]. https://developer.apple.com/xcode