Advances in Computer Science Research, volume 86
International Symposium on Communication Engineering & Computer Science (CECS 2018)

# Discussion on Hot Update Mechanism of Mobile Application

Wei Liu
Information Centre of No.7 Oil Production Plant, Daqing Oilfield, China

**Abstract.** With the rapid development of smart phone devices, the update and iteration of APP is getting faster and faster, every time users update, they need to go to the app store to download the application and reinstall it, this consumes time and flow, and the user experience is not good, so hot update mechanism is introduced, the so-called hot update is to update the code and resources under the premise of not reinstalling. This paper will elaborate on the BsDiff algorithm in incremental update and strategies and methods which the application achieves hot update based on React-Original.

**Keywords:** hot update, React-Original, BsDiff algorithm.

## 1. Introduction

With the popularization of mobile terminal cellphones, a large number of applications based on Android and IOS enter into people's lives, because fix of bug and addition of new features cause updates and upgrades of application to become very frequent, in addition to the pursuit of rich features and cool effect of the user interface, the installation packages of the program are increasing as well. Even if there are only slight differences between the new installation package and the old installation package, each version of the upgrade still downloads completely new installation package for replacement installation, this total update method not only wastes a lot of client network traffic, but also increases the upgrade process time it takes. This article will elaborate on the BsDiff difference algorithm in incremental update and the hot update mechanism for developed mobile applications based on the React-Original cross-platform development framework.

## 2. Incremental Update of Application Program

As the volume of mobile applications is increasing constantly and the release of application versions continue to change, users' upgrade has become a problem, and Google realizes that constant update of application consumes user traffic, the Smart App Update is mentioned on Google I/O, namely incremental upgrade of application, or called differential upgrade, and it is supported in the new version of Google Play. Apple certainly won't look on and do nothing, in fact; it supports delta update since IOS 6, the implementation principle: developers don't need to do extra work, the App Store will compare the new version and old versions, then only take different part, and generates different patch packages to existing users of different old versions. As a matter of fact, the principle of incremental upgrade is very simple, namely, first, difference between the old version apk and the new version apk of the application is done, and the patch of updated part is obtained, for example, the old version of apk has 5M, the new version has 8M, and the updated part may only have 3M or so (What needs to be explained here is that, The volume of obtained differential package is not a simple subtraction, because it actually need to include some context-sensitive things), the advantages of using differential upgrade are obvious, then you don't need to download the complete 8M file, just download the updated part, the updated part may only be 3, 4M, and the loss of traffic can be greatly reduced. After the users download the differential package, they need to be combined on the mobile phone. The practice that can be referred to is to copy old version of software on the mobile phone (mostly in data/) into the SD card or cache, combine them with the previous differential patch, get a new version of APK application, if nothing unexpected happens, the generated APK is consistent with the APK that you done difference before.

Incremental upgrade is not a perfect upgrade; there are at least two deficiencies:

1. Incremental upgrades generate patches based on the differences between the two application versions, you can't guarantee that users upgrade to the latest every time, so you must differentiate each version you release with the latest version, so that all versions of the user can differentially

upgrade, this operation is more tedious than the original total package upgrade, however, they can be generated in batches by automated scripts.

2. The successful prerequisite for incremental upgrade is that the user's mobile phone must have an apk that allows you to copy it and uand consistent with the version your server uses for difference, for example, the built-in apk of system cannot be obtained, and cannot carry out incremental upgrade; for some consistent with your differential version, but the content has been modified (such as cracked version apk), this is also impossible to carry out incremental upgrade, in order to prevent combined patch errors, it is best to perform sha1sum check on the old version of the apk before the patch is combined, and ensure the consistency of the basic package.

## 3. BSDIFF Algorithm

Incremental update refers to two versions of differential sequences calculated based on the differentiation algorithm, the client only needs to update and download this sequence. In the differentiation algorithm, BsDiff is mostly used to compare the changes of binary files, and it is widely used due to generated small patch package [1]. BsDiff was first used in Unix systems [2], and modern software such as Google Chrome also used this algorithm to reduce the volume of the upgraded package. Therefore, this paper proposes an incremental update method to reduce the data traffic required when application program upgrades.

In the differentiation algorithm BsDiff, the Long Common Subsequence (LCS) is mainly used to calculate the maximum common subsequence of the two sequences. This subsequence is not required to be continuous, only the order of its characters is required to be consistent. The algorithm is widely used in code anti-plagiarism systems, data cleaning and DNA sequence matching and so on. Taking the DNA sequence ATCT-GATC and TGCATAC as examples, and the maximum subsequence is TCTAC. The process of solving the LCS is actually inserting spaces into the V string and the W string, so that these two strings are aligned to the maximum extent (the corresponding elements are mapped), as shown in Figure.1.

| i coords: | 0 | 1 | 2 | 2 | 3 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| elements of V | | A | T | - | C | - | T | G | A | T | C |
| elements of W | | - | T | G | C | A | T | - | A | - | C |
| j coords: | 0 | 0 | 1 | 2 | 3 | 4 | 5 | 5 | 6 | 6 | 7 |

Figure.1 comparison of DNA sequences

The LCS is calculated by calculating the edit distance method, now the above process corresponds to a two-dimensional matrix; first, a coordinate system with horizontal direction V and a vertical direction W is established, as shown in Figure (2):

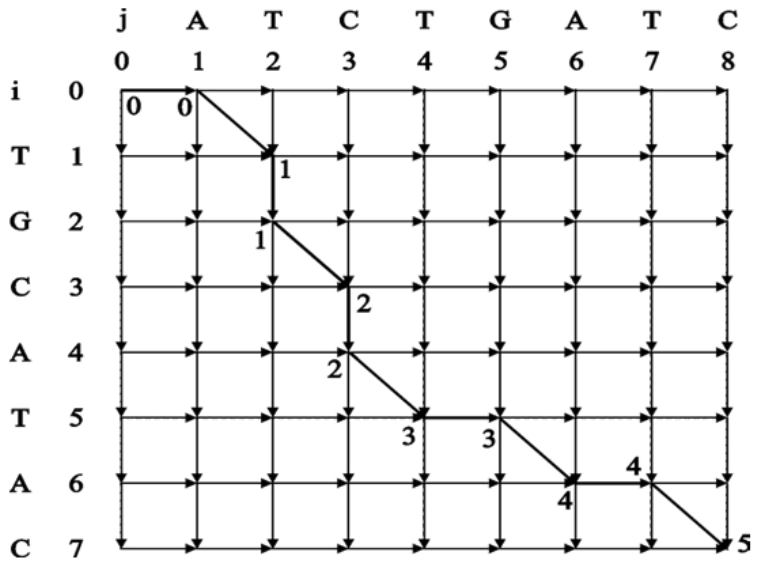


Figure.2 establish the coordinate system

Calculate the value of each node in the graph:

$$Sij = \max\begin{cases} Si-1,j \\ Si,j-1 \\ Si-1,j-1+1, vi = vj \end{cases}$$

Among them, i is the ordinate and j is the abscissa. When i=0, j=0, this value is S0, 0=0.

Figure.2 is one of the path demonstration processes. First, starting along any direction from the top left corner of the matrix (0, 0), calculate whether the corresponding elements of the coordinates are equal or not, calculate the value of the node according to formula (1), take the point (2, 2) in Figure 2 as an example, T and G are not equal, therefore, S should be equal to S1, 2 or larger in S2, 1. The value of each node in the matrix IS calculated according to the above method, from end point of the slash of S=i to S=i+1, namely the S value score is increased by one is a point where the two strings are equal, the S value of the lowest corner node in Figure 2 is the number of equal points. Among them, the horizontal arrow represents i string (TGCATAC), and insert into null value and align with j, the vertical arrow represents j string (ATCTGATC), insert into null value and align with i. Thus, after the values of the nodes of the whole table are completed, the length of the LCS (the S value in the lower right corner) and the largest subsequence can be directly obtained.

In order to achieve incremental update of the application program, the client/server model is used, the client and the server are different processes in the same system, and the client requests some services from the server according to the requirements. The overall framework of the incremental update is shown in Figure.3. Multiple versions of application are saved on the server side, and use BsDiff to calculate the difference data among different versions, and generate different patch packages to send to the client. The generation of each patch package is achieved by a separate thread, which can improve utilization of CPU. After generating the patch package, BsPatch is used to compose new installation packages on the mobile client. Thu it can be seen that the key to incremental update is to realize the server-side BsDiff algorithm and the client BsPatch algorithm.

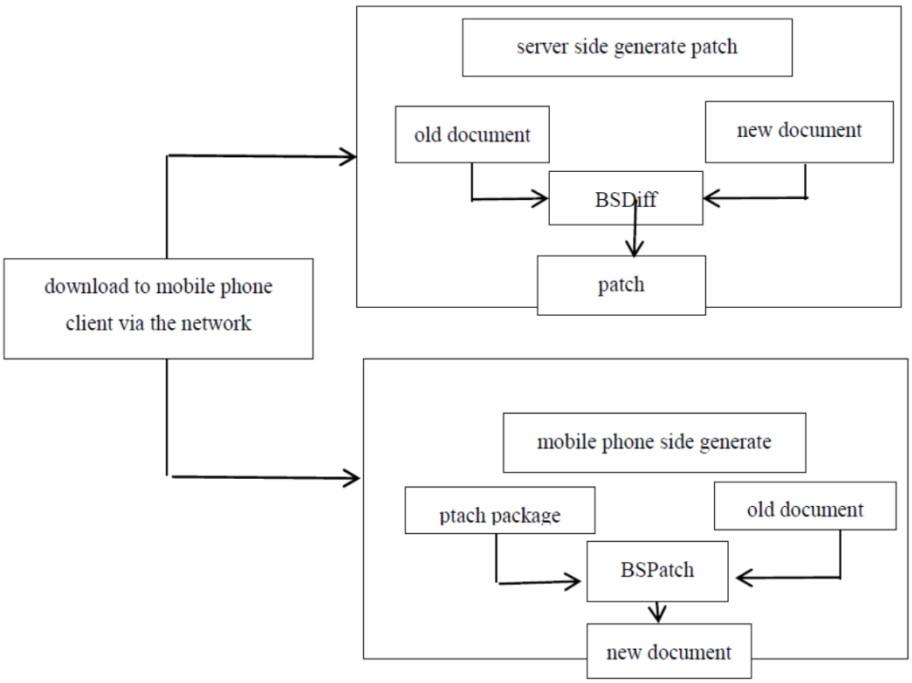


Figure.3 overall structure of incremental update

LCS is used to find the maximum common subsequence, then add additional information to form patch packages, this method can generate small patch patches, the specific implementation process is shown in Figure.4:

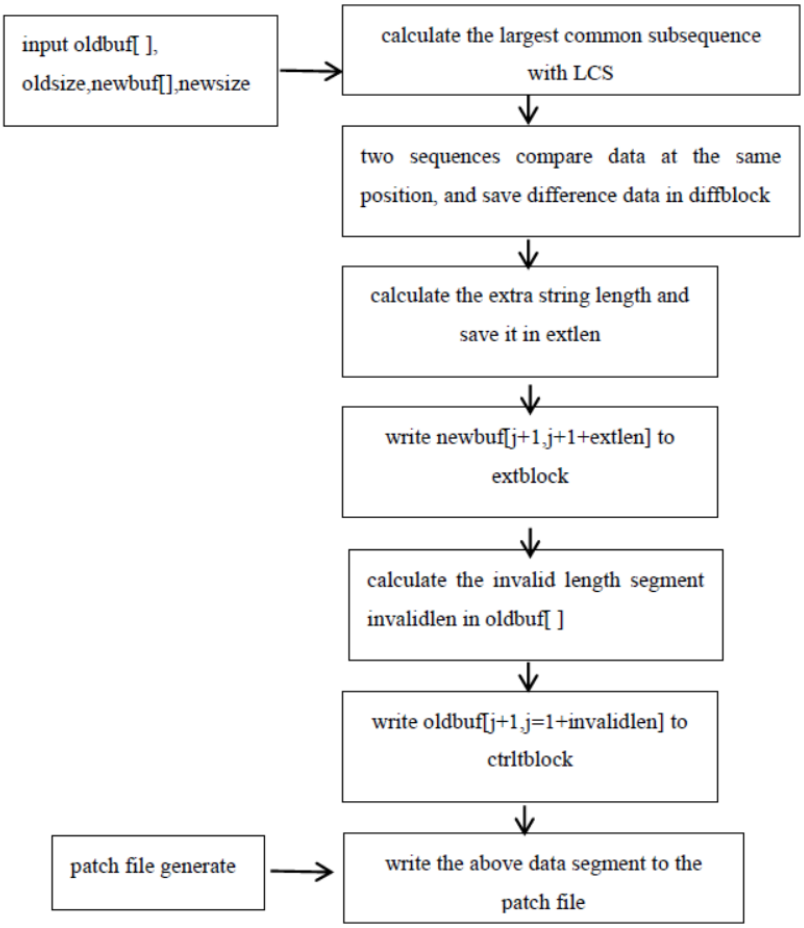


Figure.4 The specific implementation process of generate small patch patches

According to the contents of the patch sequence, the patch packages are combined with the old installation package of the local machine, and generate new installation packages. The specific implementation process is shown in Figure.5:

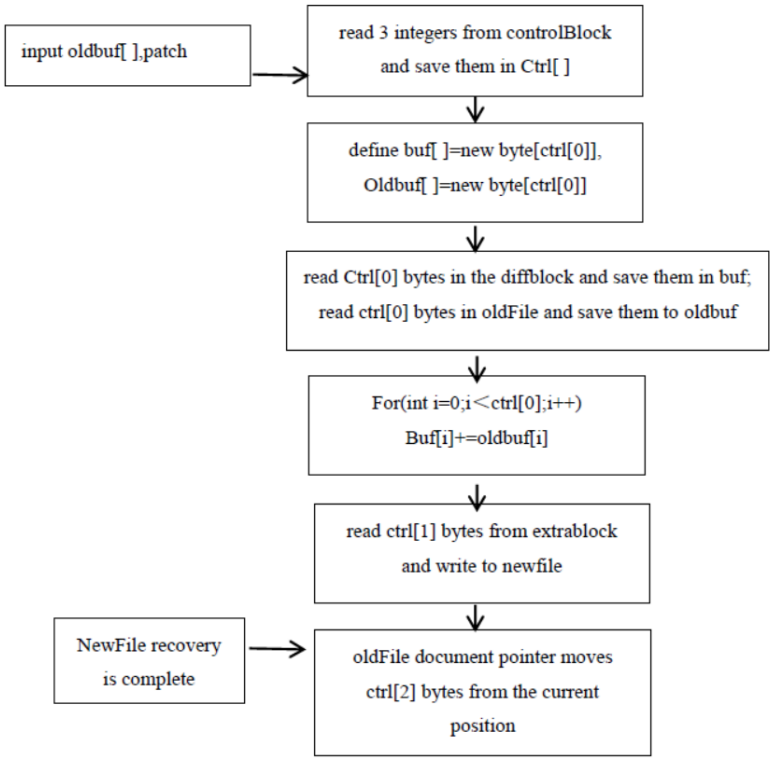


Figure.5 The specific implementation process of generate new installation packages

## 4. Hot Update Mechanism

The incremental update way mentioned above has another disadvantage, namely the application must be reinstalled, is there a way to update without reinstalling? This is the incremental hot update way described below; it only needs to push the modified and added code and resources to the user to download, the code and resources of incremental part are relatively small, so the whole hot update process can be completed without the user feeling, the best user experience has been achieved. However, this update method is difficult to achieve for applications developed with original development methods, it can be perfect used for applications developed with scripting languages.

### 4.1 Achieve the Hot Update of the Script

Here we will talk about how to achieve the hot update mechanism for cross-platform applications developed with React-Original (RN) development way. Because RN is written in a scripting language, the so-called scripting language is a language that can be run without compiling, namely "read is run". We replace it with a new version of the script before "read", the implementation is new logic when running, slight abstract, is the image resource is also "read is run"? So, the script is essentially the same as the image resource, and it can be hot updated.

### 4.2 Mechanism of RN Loading Script

In order to implement hot update of the RN script, first we look at how the RN loads the script. When writing business logic, we will have many js files, when packaging, RN will package these js files to a file called index. android. bundle (ios is index.ios. bundle), all js code (including rn source code, third party library, code of business logic) are in this file, when starting app, the bundle file will be loaded the first time, so hot update of script is to replace this bundle file. We execute the following

command at the RN project root to get the bundle file and image resource: react-original bundle --entry-file index.android.js

--bundle-output./bundle/index.android.bundle--platform android --assets-dest./bundle --dev false

--entry is the entry js file, android system is index.android.js, IOS system is index.ios.js, --bundle-output is the generated bundle file path, --platform is platform, --assets-dest is the output directory of the image resource, this will be used in the subsequent image incremental update, --dev present whether it is a development version, when makingthe official installation package, we assign it to false.

The volume of generated bundle file is still not small, I am afraid there will be at least 900K for the empty project, so we make it into a zip package and put it on the web server for the client to download. Download bundle files can be written in original language or js. After downloading, decompress and then replace the bundle file to achieve a hot update of the application.

### 4.3 RN Incremental Hot Update

There are two ways:

1. Separate the bundle. The bundle stores the RN source code, the third-party library code and the business logic code, the most frequently updated is the business logic code, so we package the RN source code and the third-party library code into a bundle, and the business logic is packaged into a bundle, when the hot update is done, only the bundle of business logic is updated.

2. Package the patch file. We can use BsDiff to compare the two versions of bundle files and get the difference files, which are the patches; the client downloads the patch files and merges them with the local bundle, thus getting the latest version of bundle files.

The following will focus on explaining the second method in combination with the difference algorithm described above.

(1) Generate patch.

We download the latest source code from the bsdiff official website, and then compile it to get the executable binary files.

Use the command line to enter the bsdiff directory and input the command:

bsdiff a.txt b.txt c.pat

The above command is to generate a patch c.pat for the a.txt and b.txt two files.

If it is a Linux system, you can execute the following commands in order:

yum install bzip2-devel

wget http://www.daemonology.net/bsdiff/bsdiff-4.3.tar.gz

tar zxvf bsdiff-4.3.tar.gz

cd bsdiff-4.3

After the compiling is completed, 2 binary files will be generated in the directory: bsdiff, bspatch, these 2 binary files can be used directly, but it is recommended to copy to /usr/local/sbin/:

cp bsdiff /usr/local/sbin/

cp bspatch /usr/local/sbin/

This can be used directly in the command line:

bsdiff a.txt b.txt c.pat

(2) use patch

After getting the patch file, the next step will use the patch, take the above a.txt, b.txt, c.pat to test:

bspatch a.txt d.txt c.pat

Get the file d.txt, open it to see whether it is the same as b.txt, if it is the same, it shows that test is successful.

## 5. Conclusion

This paper elaborates on the incremental update method of the application program; focus on explaining the implementation principle of the BsDiff algorithm for difference update, and the

incremental hot update method for mobile applications based on script writing. It shows that this hot update method can better iteratively update the program and has a better user experience.

## References

[1]. Zhang Wei. Design and Implementation of P2P-based File Cooperation Platform System[D]. Changchun: Jilin University, 2009: 47 -48.

[2]. Niu Yongjie, Zhang Cheng. Comparation of String Similarity Algorithm[J]. Computer and Digital Engineering, 2012, 40(3): 14 -17.