(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 115080098 A
(43)申请公布日 2022.09.20

(21)申请号 202210668530.8

(22)申请日 2022.06.14

(71)申请人 平安付科技服务有限公司
地址 518033 广东省深圳市福田区福田街道福华路319号兆邦基金融大厦26层2605单元

(72)发明人 钟洪运

(74)专利代理机构 上海汉之律师事务所 31378
专利代理师 冯华

(51)Int.Cl.
G06F 8/656(2018.01)
G06F 8/658(2018.01)
G06F 8/51(2018.01)

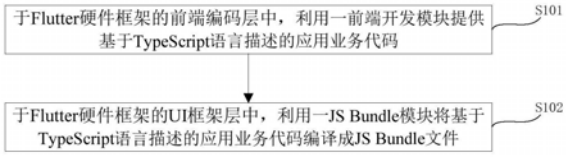
权利要求书2页 说明书12页 附图4页

(54)发明名称

基于JavaScript的Flutter热更新方法、装置、设备及介质

(57)摘要

本发明涉及计算机应用软件技术领域，公开了一种基于JavaScript的Flutter热更新方法、系统、介质及计算机设备，所述基于JavaScript的Flutter热更新方法包括以下步骤：于Flutter硬件框架的前端编码层中，利用一前端开发模块提供基于TypeScript语言描述的应用业务代码；于Flutter硬件框架的UI框架层中，利用一JS Bundle模块将所述基于TypeScript语言描述的应用业务代码编译成JS Bundle文件；所述JS Bundle模块以Script Widget描述方式将所述基于TypeScript语言描述的应用业务代码编译为基于JavaScript语言描述的JS Bundle文件；所述JS Bundle文件用于被解析为Dart语言，以实现Flutter热更新；所述客户端包括iOS客户端。本发明适用于iOS设备，解决了现有技术中Flutter的热更新不支持iOS设备的问题。



CN 115080098 A

1.一种基于JavaScript的Flutter热更新方法,其特征在于,包括:

于Flutter硬件框架的前端编码层中,利用一前端开发模块提供基于TypeScript语言描述的应用业务代码;所述前端开发模块是软件开发者编写应用业务代码所使用的工具;

于Flutter硬件框架的UI框架层中,利用一JS Bundle模块将所述基于TypeScript语言描述的应用业务代码编译成JS Bundle文件;其中,所述JS Bundle模块以Script Widget描述方式将所述基于TypeScript语言描述的应用业务代码编译为基于JavaScript语言描述的JS Bundle文件;所述JS Bundle文件用于被解析为Dart语言,以实现客户端的Flutter控件的热更新;所述客户端包括iOS客户端。

2.根据权利要求1所述的基于JavaScript的Flutter热更新方法,其特征在于,所述于Flutter硬件框架的UI框架层中,利用一JS Bundle模块将所述基于TypeScript语言描述的应用业务代码编译成JS Bundle文件之后,所述基于JavaScript的Flutter热更新方法还包括:

利用一系统模拟器对所述JS Bundle文件进行调试;所述系统模拟器包括iOS模拟器;

利用一分发模块发布调试后的JS Bundle文件;

所述调试后的JS Bundle文件被所述客户端检测更新获取后,所述调试后的JS Bundle文件被解析为Dart语言,以在所述客户端显示更新后的Flutter控件。

3.根据权利要求1所述的基于JavaScript的Flutter热更新方法,其特征在于,所述JS Bundle文件被解析为Dart语言,以实现客户端的Flutter控件的热更新,包括:

所述JS Bundle文件用于被所述客户端解析为UI描述;

所述Flutter硬件框架的Dart层的UI引擎将所述UI描述生成Flutter控件,显示于所述客户端。

4.根据权利要求3所述的基于JavaScript的Flutter热更新方法,其特征在于,所述Dart层的UI引擎将所述UI描述生成Flutter控件,包括:

所述Dart层的UI引擎获取所述JS Bundle文件的UI描述;所述JS Bundle文件的UI描述为一虚拟Script Widget Tree;

所述Dart层的UI引擎根据所述虚拟Script Widget Tree对应转换为Flutter Widget Tree;

所述Dart层的UI引擎编译所述Flutter Widget Tree获得Flutter Element Tree;

所述Dart层的UI引擎编译所述Flutter Element Tree获得Flutter Render Tree,即获得所述Flutter控件。

5.根据权利要求4所述的基于JavaScript的Flutter热更新方法,其特征在于,所述Dart层的UI引擎根据所述虚拟Script Widget Tree对应转换为Flutter Widget Tree,包括:所述虚拟Script Widget Tree的结构与所述Flutter Widget Tree的结构相互对应;所述虚拟Script Widget Tree的结构包括虚拟Statefull Widget、虚拟Row Widget、虚拟Text Widget、虚拟Container Widget和虚拟Image Widget;所述Flutter Widget Tree的结构包括Statefull Widget、Row Widget、Text Widget、Container Widget和Image Widget;所述Dart层的UI引擎根据所述虚拟Script Widget Tree和所述Flutter Widget Tree的对应结构完成转换。

6.根据权利要求1所述的基于JavaScript的Flutter热更新方法,其特征在于,所述

Script Widget描述方式包括：

利用Script Widget管理一个Script页面或控件;所述Script Widget负责创建管理虚拟Script Widget Tree,并以自增ID与Flutter Widget对应相互调用;所述Script Widget每次编译都会创建一个新的虚拟Script Widget Tree。

7.根据权利要求2所述的基于JavaScript的Flutter热更新方法,其特征在于:所述客户端还包括Android客户端;所述系统模拟器还包括Android模拟器。

8.一种基于JavaScript的Flutter热更新装置,其特征在于,所述基于JavaScript的Flutter热更新装置包括：

前端开发模块,用于提供基于TypeScript语言描述的应用业务代码;

JS Bundle模块,与所述前端开发模块通信相连,将所述基于TypeScript语言描述的应用业务代码编译成JS Bundle文件;所述JS Bundle模块以Script Widget描述方式将所述基于TypeScript语言描述的应用业务代码编译为基于JavaScript语言描述的JS Bundle文件;

其中,所述JS Bundle文件用于被解析为Dart语言,以实现客户端的Flutter控件的热更新;所述客户端包括iOS客户端。

9.一种计算机设备,包括存储器、处理器以及存储在所述存储器中并可在所述处理器上运行的计算机程序,其特征在于,所述处理器执行所述计算机程序时实现如权利要求1至7任一项所述基于JavaScript的Flutter热更新方法的步骤。

10.一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现如权利要求1至7任一项所述基于JavaScript的Flutter热更新方法的步骤。

基于JavaScript的Flutter热更新方法、装置、设备及介质

技术领域

[0001] 本发明属于计算机应用软件、智慧城市技术领域，涉及一种应用更新方法，特别是涉及一种基于JavaScript的Flutter热更新方法、装置、设备及介质。

背景技术

[0002] Flutter为应用开发带来了革新，只要一套代码库，即可构建、测试和发布适用于移动、Web、桌面和嵌入式平台的精美应用。Flutter官方不支持动态化，原因是Flutter在Release模式下构建的是AOT编译产物，在Debug模式下构建的是JIT，AOT依赖的Dart VM和JIT(just-in-time)并不一样，JIT Release并不支持iOS设备。

[0003] 行业内曾提出如下解决方案：

[0004] 1、产物替换：

[0005] 通过下发产物替换，官方在曾经推出了Code Push方案，甚至可以支持Diff差量下载，但是在2019年4月被叫停，原因是官方对动态化后的性能没有自信，并且对安全性有所顾虑。之前，官方提供方案的局限性也十分明显，比如对Native-Flutter混合App支持不友好，并且无法进行灰度等业务定制操作，所以不能满足通用性和高性能的核心目标。

[0006] 2、AOT搭载JIT：

[0007] Flutter在Release模式下构建的是AOT编译产物，iOS是AOT Assembly，Android默认AOT Blob。同时Flutter也支持JIT Release模式，可以动态加载Kernel snapshot或App-JIT snapshot。如果在AOT上支持JIT，就可以实现动态化能力。但问题在于，AOT依赖的Dart VM和JIT并不一样，AOT需要一个编译后的“Dart VM”(更准确地说是Precompiled Runtime)，JIT依赖的是Dart VM(一个虚拟机，提供语言执行环境)；并且JIT Release并不支持iOS设备，构建的应用也不能在AppStore上发布。

[0008] 3、静态解析Dart语言，生成UI描述

[0009] Dart本身是描述语言，IDE的Outline工具可以解析Dart代码生成树形结构，可以利用其源码生成JSON UI描述，但是静态解析Dart存在缺点，不能写逻辑，对编写UI代码有很多限制，不能写判断语句，不能写函数，要支持这些成本很高，所以只好放弃。

发明内容

[0010] 鉴于以上所述现有技术的缺点，本发明的目的在于提供一种基于JavaScript的Flutter热更新方法、装置、设备及介质，用于解决现有技术中Flutter的热更新不支持iOS设备的问题。

[0011] 为实现上述目的及其他相关目的，本发明提供一种基于JavaScript的Flutter热更新方法，所述基于JavaScript的Flutter热更新方法包括以下步骤：于Flutter硬件框架的前端编码层中，利用一前端开发模块提供基于TypeScript语言描述的应用业务代码；于Flutter硬件框架的UI框架层中，利用一JS Bundle模块将所述基于TypeScript语言描述的应用业务代码编译成JS Bundle文件；其中，所述JS Bundle模块以Script Widget描述方式

将所述基于TypeScript语言描述的应用业务代码编译为基于JavaScript语言描述的JS Bundle文件;所述JS Bundle文件用于被解析为Dart语言,以实现客户端的Flutter控件的热更新;所述客户端包括iOS客户端。

[0012] 本发明还提供一种基于JavaScript的Flutter热更新装置,所述基于JavaScript的Flutter热更新装置包括:前端开发模块,用于提供基于TypeScript语言描述的应用业务代码;JS Bundle模块,与所述前端开发模块通信相连,将所述基于TypeScript语言描述的应用业务代码编译成JS Bundle文件;所述JS Bundle模块以Script Widget描述方式将所述基于TypeScript语言描述的应用业务代码编译为基于JavaScript语言描述的JS Bundle文件;其中,所述JS Bundle文件用于被解析为Dart语言,以实现客户端的Flutter控件的热更新;所述客户端包括iOS客户端。

[0013] 本发明还提供一种计算机设备,包括存储器、处理器以及存储在所述存储器中并可在所述处理器上运行的计算机程序,所述处理器执行所述计算机程序时实现上述基于JavaScript的Flutter热更新方法的步骤。

[0014] 本发明还提供一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,所述计算机程序被处理器执行时实现上述基于JavaScript的Flutter热更新方法的步骤。

[0015] 如上所述,本发明所述的基于JavaScript的Flutter热更新方法、装置、设备及介质,具有以下有益效果:

[0016] 本发明所述的基于JavaScript的Flutter热更新方法、装置、设备及介质的方案中,可以利用前端开发模块提供基于TypeScript语言描述的应用业务代码;可以利用JS Bundle模块将所述基于TypeScript语言描述的应用业务代码编译成JS Bundle文件;利用所述JS Bundle模块以Script Widget描述方式将所述基于TypeScript语言描述的应用业务代码编译为基于JavaScript语言描述的JS Bundle文件;使得所述JS Bundle文件可以被解析为Dart语言,进而实现客户端的Flutter控件的热更新;所述客户端包括iOS客户端。本发明通过JS Bundle模块以Script Widget描述方式将所述基于TypeScript语言描述的应用业务代码编译为基于JavaScript语言描述的JS Bundle文件的处理方式,使得Flutter热更新适用于iOS设备,解决了现有技术中Flutter的热更新不支持iOS设备的问题。

[0017] 本发明利用前端开发模块提供基于TypeScript语言描述的应用业务代码,通过前端熟悉的开发框架来编写业务代码,并可以通过Script Widget描述方式将业务代码自动转换为Flutter Widget进行渲染,实现了前端开发者的零成本接入。

附图说明

[0018] 为了更清楚地说明本发明实施例的技术方案,下面将对本发明实施例的描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动性的前提下,还可以根据这些附图获得其他的附图。

[0019] 图1A显示为本发明实施例所述的基于JavaScript的Flutter热更新方法的一种示例性应用场景示意图。

[0020] 图1B显示为本发明实施例所述的基于JavaScript的Flutter热更新方法的一种实

现流程示意图。

[0021] 图2显示为本发明实施例所述的步骤S102中JS Bundle文件被解析为Dart语言的一种实现过程示意图。

[0022] 图3显示为本发明实施例所述的基于JavaScript的Flutter热更新方法的另一种实现流程示意图。

[0023] 图4显示为实现本发明实施例所述的基于JavaScript的Flutter热更新方法的一种硬件架构示意图。

[0024] 图5显示为实现本发明实施例所述的基于JavaScript的Flutter热更新方法的一种渲染原理图。

[0025] 图6A显示为本发明实施例所述的基于JavaScript的Flutter热更新系统的一种示例实现结构示意图。

[0026] 图6B显示为本发明实施例所述的基于JavaScript的Flutter热更新系统的另一种示例实现结构示意图。

[0027] 图7显示为本发明实施例所述的计算机设备的一种实现结构示意图。

[0028] 元件标号说明

[0029] 600 基于JavaScript的Flutter热更新装置

[0030] 610 前端开发模块

[0031] 620 JS Bundle模块

[0032] 630 系统模拟器

[0033] 640 分发模块

[0034] 700 计算机设备

[0035] 710 存储器

[0036] 720 处理器

[0037] S101～S105 步骤

具体实施方式

[0038] 以下通过特定的具体实例说明本发明的实施方式，本领域技术人员可由本说明书所揭露的内容轻易地了解本发明的其他优点与功效。本发明还可以通过另外不同的具体实施方式加以实施或应用，本说明书中的各项细节也可以基于不同观点与应用，在没有背离本发明的精神下进行各种修饰或改变。需说明的是，在不冲突的情况下，以下实施例及实施例中的特征可以相互组合。

[0039] 需要说明的是，以下实施例中所提供的图示仅以示意方式说明本发明的基本构想，遂图式中仅显示与本发明中有关的组件而非按照实际实施时的组件数目、形状及尺寸绘制，其实际实施时各组件的型态、数量及比例可为一种随意的改变，且其组件布局型态也可能更为复杂。

[0040] Flutter使用Dart语言开发，Dart可以被编译成不同平台的本地代码，让Flutter可以直接和平台通讯而不需要一个中间的桥接过程，从而提高了性能。Dart是Flutter的官方语言，Flutter的渲染效果比较好，跨平台，但是Flutter不支持动态更新，不支持iOS设备。

[0041] 和静态解析Dart相比，动态解析方案是一个极其轻量的运行时库，让编写UI的Dart代码运行了起来，生成树形结构，再序列化为JSON(debug)，Flat Buffers(release)UI描述。动态运行Dart语言，生成UI描述的过程是，Dart源代码在进行编译时会通过DART_PRECOMPILED_RUNTIME宏进行条件编译，从而在Debug版编译JIT模式，Release版编译AOT模式，但这两种模式是互斥的，无法同时存在。

[0042] 本发明总结了现有方案的不足，充分结合现有项目、插件、AVMS平台支持的优势，通过抽离DartVM，单独编译出一个DartVM，打包成动态库；通过JavasSriptCore替换DartVM，最终以动态生成DSL的形式实现；通过下发增量/全量更新包，达到动态更新Flutter iOS项目的效果。

[0043] 本发明实施例提供的基于JavaScript的Flutter热更新方法，可应用在如图1A的应用环境中，其中，开发端100可设有多台计算机设备，客户端300包括iOS、Android等任意类型的终端设备，例如智能手机、平板电脑、iMac、PAD等；服务端200可以是一台服务器，也可以是服务器集群。

[0044] 本发明所述基于JavaScript的Flutter热更新方法利用一前端开发模块提供基于TypeScript语言描述的应用业务代码，其中，前端开发模块设置于所述开发端100，前端开发模块的功能可以由一台计算机设备单独实现，或由多台计算机设备协同实现。

[0045] 本发明所述基于JavaScript的Flutter热更新方法利用一JS Bundle模块将所述基于TypeScript语言描述的应用业务代码编译成JS Bundle文件；所述JS Bundle模块以Script Widget描述方式将所述基于TypeScript语言描述的应用业务代码编译为基于JavaScript语言描述的JS Bundle文件，其中，JS Bundle模块设置于所述开发端100，JS Bundle模块的功能可以由一台计算机设备单独实现，或由多台计算机设备协同实现。

[0046] 本发明所述基于JavaScript的Flutter热更新方法利用一系统模拟器对所述JS Bundle文件进行调试；所述系统模拟器包括iOS模拟器；其中，所述系统模拟器可以由所述开发端100中的一台计算机设备单独实现，或由多台计算机设备协同实现。

[0047] 本发明所述基于JavaScript的Flutter热更新方法利用一分发模块发布调试后的JS Bundle文件，其中，分发模块可以由所述服务端200中的一台服务器或服务器集群实现。

[0048] 本发明中，所述JS Bundle文件用于被解析为Dart语言，进而实现客户端的Flutter控件的热更新；其中，所述客户端可由所述客户端300中的设备实现。

[0049] 请参阅图1B，本发明实施例提供一种基于JavaScript的Flutter热更新方法，所述基于JavaScript的Flutter热更新方法包括以下步骤：

[0050] S101，于Flutter硬件框架的前端编码层中，利用一前端开发模块提供基于TypeScript(简写TS)语言描述的应用业务代码。其中，应用业务代码包括适用于移动、Web、桌面和嵌入式平台的业务代码。TypeScript是一种由微软开发的自由和开源的编程语言，它是JavaScript(简写JS)的一个超集，扩展了JavaScript的语法。

[0051] 所述前端开发模块是用于编写应用业务代码的工具；软件开发者可以在所述前端开发模块中编写基于TypeScript语言描述的应用业务代码。

[0052] 本发明可以接入前端npm生态，并提供和Flutter原生十分接近的编码方式和语法提示等功能。对前端发展，实现web前端dom(Document Object Model，文档对象模型)+css(Cascading Style Sheets，层叠样式表)的开发方式来接入Flutter，通过Vue/React等前

端熟悉的开发框架来编写业务代码,自动转换为Flutter Widget进行渲染,实现前端开发者的零成本接入。其中,npm是JavaScript世界的包管理工具,并且是Node.js平台的默认包管理工具,通过npm可以安装、共享、分发代码,管理项目依赖关系。

[0053] S102,于Flutter硬件框架的UI框架层中,利用一JS Bundle模块将所述基于TypeScript语言描述的应用业务代码编译成JS Bundle文件;所述JS Bundle模块以Script Widget描述方式将所述基于TypeScript语言描述的应用业务代码编译为基于JavaScript语言描述的JS Bundle文件;所述JS Bundle文件用于被解析为Dart语言,进而实现客户端的Flutter控件的热更新;所述客户端包括iOS客户端。

[0054] 本发明通过步骤S102的处理,使得Flutter热更新适用于iOS设备,解决了现有技术中Flutter的热更新不支持iOS设备的问题。

[0055] 进一步,Bundle是Unix/linux系统中的一种可执行文件,用户可以在终端中使用./***(文件名).bundle命令使其运行。也就是说Bundle是用来传递数据的“容器”,它保存的数据是以key-value(键值对)的形式存在的。我们经常使用Bundle在Activity之间传递数据,传递的数据可以是boolean、byte、int、long、float、double、string等基本类型或它们对应的数组,也可以是对象或对象数组。当Bundle传递的是对象或对象数组时,必须实现Serializable或Parcelable接口。

[0056] 在Flutter中,一切皆Widget。无论是显示界面的UI元素,如Text、Image、Icon等,还是功能性组件,如手势检测的GestureDetector组件、应用主题数据传递的Theme组件、移除系统组件自带Padding的MediaQuery组件等,可以说,Flutter界面就是由一个个粒度非常细的Widget组合起来的。由于Widget是不可变的,所以当视图更新时,Flutter会创建新的Widget来替换旧的Widget并将旧的Widget销毁。但这样就会涉及到大量Widget对象的销毁和重建,从而对垃圾回收造成压力。也因此,Flutter将Widget设计的十分轻量,并将视图的配置信息与渲染抽象出来,分别交给Element与RenderObject,从而使得Widget只起一个组织者作用,可以将Element与RenderObject组合起来,构成一个视图。Widget是一种非常轻量且不可变的数据结构,只起一个组织者作用。

[0057] 于本发明的一实施例中,步骤S102中,JS Bundle文件被解析为Dart语言,进而实现Flutter热更新的一种实现过程包括:所述客户端调用JavaScript版的轻量级Flutter Runtime将所述JS Bundle文件生成Dart语言的UI描述,并将所述UI描述传递给Dart层的UI引擎;所述Dart层的UI引擎将所述UI描述生成Flutter控件。

[0058] 其中,Runtime运行时刻是指一个程序在运行(或者在被执行)的状态。也就是说,当打开一个程序使它在电脑上运行的时候,那个程序就是处于运行时刻。在一些编程语言中,把某些可以重用的程序或者实例打包或者重建成为“运行库”。这些实例可以在它们运行的时候被链接或者被任何程序调用。Runtime类封装了运行时的环境。每个Java应用程序都有一个Runtime类实例,使应用程序能够与其运行的环境相连接。一般不能实例化一个Runtime对象,应用程序也不能创建自己的Runtime类实例,但可以通过getRuntime方法获取当前Runtime运行时对象的引用。一旦得到了一个当前的Runtime对象的引用,就可以调用Runtime对象的方法去控制Java虚拟机的状态和行为。当Applet和其他不被信任的代码调用任何Runtime方法时,常常会引起SecurityException异常。

[0059] 进一步,参见图2所示,所述Dart层的UI引擎将所述UI描述生成Flutter控件的实

现过程包括:所述Dart层的UI引擎获取所述JS Bundle文件的UI描述;所述JS Bundle文件的UI描述为一虚拟Script Widget Tree;所述Dart层的UI引擎根据所述虚拟Script Widget Tree对应转换为Flutter Widget Tree;所述Dart层的UI引擎编译所述Flutter Widget Tree获得Flutter Element Tree;所述Dart层的UI引擎编译所述Flutter Element Tree获得Flutter Render Tree,即获得所述Flutter控件。

[0060] 于本发明的一实施例中,所述Script Widget描述方式包括:利用Script Widget管理一个Script页面或控件;所述Script Widget负责创建管理虚拟Script Widget Tree,并以自增ID与Flutter Widget对应相互调用;所述Script Widget每次编译(Build)都会创建一个新的虚拟Script Widget Tree。

[0061] 进一步,所述虚拟Script Widget Tree的结构与所述Flutter Widget管理的Flutter Widget Tree的结构相互对应;所述虚拟Script Widget Tree的结构包括虚拟Statefull Widget、虚拟Row Widget、虚拟Text Widget、虚拟Container Widget和虚拟Image Widget;所述Flutter Widget Tree的结构包括Statefull Widget、Row Widget、Text Widget、Container Widget和Image Widget。

[0062] 于本发明的一实施例中,所述客户端还包括Android客户端;本发明所述的基于JavaScript的Flutter热更新方法不但可以支持iOS设备,还可以支持Android设备,这样就使得开发者只需要开发一套代码就能够同时应用于iOS设备和Android设备,无需分别开发2套代码来分别适用于iOS设备和Android设备,极大地节省了开发成本,提高了开发效率,降低了更新维护成本。

[0063] 参见图3所示,于本发明的一实施例中,所述基于JavaScript的Flutter热更新方法还包括:

[0064] S103,于Flutter硬件框架的Dart层中,利用一系统模拟器对所述JS Bundle文件进行调试;所述系统模拟器包括iOS模拟器。开发者可以通过系统模拟器对所述JS Bundle文件进行调试,或通过前端开发模块修改基于TypeScript语言描述的应用业务代码,最终使得所述JS Bundle文件符合发布条件。

[0065] 于本发明的一实施例中,所述系统模拟器还包括Android模拟器。本发明所述的基于JavaScript的Flutter热更新方法不但可以支持iOS设备,还可以支持Android设备,这样就使得开发者只需要开发一套代码就能够同时应用于iOS设备和Android设备,无需分别开发2套代码来分别适用于iOS设备和Android设备,极大地节省了开发成本,提高了开发效率,降低了更新维护成本。

[0066] S104,利用一分发模块发布调试后的JS Bundle文件。进一步,可以将调试好的JS Bundle文件打包成zip压缩包,然后上传到分发模块(AVMS),使native通过分发的7z包解析pluginId打开Flutter页面。native是一个计算机函数,一个Native Method就是一个Java调用非Java代码的接口,方法的实现由非Java语言实现,比如C或C++。

[0067] S105,所述客户端检测更新获得所述调试后的JS Bundle文件,并将所述调试后的JS Bundle文件解析为Dart语言,进而显示更新后的Flutter控件。

[0068] 于本发明的一实施例中,步骤S105中,所述客户端将调试后的JS Bundle文件解析为Dart语言,进而显示更新后的Flutter控件的一种实现过程包括:所述客户端调用JavaScript版的轻量级Flutter Runtime将所述调试后的JS Bundle文件生成UI描述,并将

所述UI描述传递给Dart层的UI引擎;所述Dart层的UI引擎将所述UI描述生成Flutter控件。

[0069] 其中,Runtime运行时刻是指一个程序在运行(或者在被执行)的状态。也就是说,当打开一个程序使它在电脑上运行的时候,那个程序就是处于运行时刻。在一些编程语言中,把某些可以重用的程序或者实例打包或者重建成为“运行库”。这些实例可以在它们运行的时候被链接或者被任何程序调用。Runtime类封装了运行时的环境。每个Java应用程序都有一个Runtime类实例,使应用程序能够与其运行的环境相连接。一般不能实例化一个Runtime对象,应用程序也不能创建自己的Runtime类实例,但可以通过getRuntime方法获取当前Runtime运行时对象的引用。一旦得到了一个当前的Runtime对象的引用,就可以调用Runtime对象的方法去控制Java虚拟机的状态和行为。当Applet和其他不被信任的代码调用任何Runtime方法时,常常会引起Security Exception异常。

[0070] 进一步,所述Dart层的UI引擎将所述UI描述生成Flutter控件的实现过程包括:所述Dart层的UI引擎获取所述调试后的JS Bundle文件的UI描述;所述调试后的JS Bundle文件的UI描述为一虚拟Script Widget Tree;所述Dart层的UI引擎根据所述虚拟Script Widget Tree对应转换为Flutter Widget Tree;所述Dart层的UI引擎编译所述Flutter Widget Tree获得Flutter Element Tree;所述Dart层的UI引擎编译所述Flutter Element Tree获得Flutter Render Tree,即获得所述Flutter控件。

[0071] 所述Dart层的UI引擎根据所述虚拟Script Widget Tree对应转换为Flutter Widget Tree的一种实现过程包括:所述虚拟Script Widget Tree的结构与所述Flutter Widget管理的Flutter Widget Tree的结构相互对应;所述虚拟Script Widget Tree的结构包括虚拟Statefull Widget、虚拟Row Widget、虚拟Text Widget、虚拟Container Widget和虚拟Image Widget;所述Flutter Widget Tree的结构包括Statefull Widget、Row Widget、Text Widget、Container Widget和Image Widget;所述Dart层的UI引擎根据所述虚拟Script Widget Tree和所述Flutter Widget Tree的对应结构完成转换。

[0072] 本发明中,利用Script Widget管理一个Script页面或控件;所述Script Widget负责创建管理虚拟Script Widget Tree,并以自增ID与Flutter Widget对应相互调用;所述Script Widget每次编译(Build)都会创建一个新的虚拟Script Widget Tree。

[0073] 本发明可以支持TypeScript语言和前端生态,可以支持Flutter中同名Widget类,相同API;支持Flutter相同的Build方式,setState刷新及事件响应方法,支持JS(JavaScript)和Dart双向调用通道,支持模拟器页面Hot Reload,支持Safari和Chrome调试,支持编译现有Flutter程为JS,运行在框架之上。

[0074] 本发明的核心思路是把Flutter的渲染逻辑中的三棵树中的第一棵Widget配置树(Widget Tree)放到JavaScript中生成。用JavaScript完整实现了Flutter控件层封装,可以使用JavaScript及极其类似Dart的开发方式,开发Flutter应用,利用JavaScript版的轻量级Flutter Runtime,生成UI描述,传递给Dart层的UI引擎,UI引擎把UI描述生成真正的Flutter控件。

[0075] 参见图4所示,本发明实施例提供一种实现所述基于JavaScript的Flutter热更新方法的硬件架构,包括前端编码层(App Code(TS)),UI框架层(JS UI Framework),Dart层,其中Dart层包括Flutter UI Engine(UI引擎),Flutter Framework(Flutter框架),Engine(底层引擎)。前端编码层(App Code(TS))可以实现基于TypeScript语言编译应用业务代

码。UI框架层(JS UI Framework)可以实现以Script Widget描述方式将所述基于TypeScript语言描述的应用业务代码编译为基于JavaScript语言描述的JS Bundle文件。Dart层可以实现将JS Bundle文件解析为Dart语言,进而实现Flutter热更新。其中,UI框架层(JS UI Framework)可以轻量的响应UI框架,使用TS语言,使用npm前端包开发,具有1400个Widget和相关类,具有与Flutter相同的API,具有CLI工具。Dart层中的Flutter UI Engine(UI引擎)可以支撑DSL解析,支持UI刷新逻辑,支持事件响应,支持Flutter API。Dart层中的Flutter Framework(Flutter框架)打通了JS与Dart之间的双向高速通道,具有JSVM管理功能,具有JS资源管理功能,支持JS常规模式请求(JS common model require)。

[0076] 参见图5所示,本发明实施例提供一种实现所述基于JavaScript的Flutter热更新方法的渲染原理,包括:ScriptWidget管理一个Script页面或控件,负责创建管理ScriptWidgetTree,以自增ID与Flutter对应Widget相互调用,每次编译(Build)都会创建一个新的WidgetTree。ScriptWidgetTree的结构参见图5左侧树状结构,即VM层的Widge配置树,Flutter层的Widge配置树结构参见图5右侧树状结构。

[0077] 本实施例提供一种实现所述基于JavaScript的Flutter热更新方法的开发方式,包括:基于前端框架(即前端编码层(App Code(TS))),使用TypeScript语言,以类似Flutter的Widget组装方式开发UI,借助前端生态的基础能力,开发App。

[0078] 例如:本实施例提供一种实现所述基于JavaScript的Flutter热更新方法的具体接入流程:

[0079] 1、在Flutter工程里引入mxflutter Flutter plugin。

[0080] 2、用mxflutter cli工具新建一个TypeScript的mxflutter工程,开发完成之后编译输出JS Bundle文件,把TS工程编译的bundle-xxx.js文件放置指定目录下,然后就可以调用mxflutter提供的接口打开TS页面了。

[0081] 3、将调试好的JS Bundle文件打包成zip包,上传到AVMS,native通过分发的7z包,解析plugin Id打开Flutter页面。

[0082] 本实施例提供一种实现所述基于JavaScript的Flutter热更新方法的环境搭建:

[0083] 1、添加依赖:将下面内容添加到Flutter工程的pubspec.yaml文件中:

[0084] mxflutter:

[0085] git:

[0086] url:http://code.paic.com.cn/yqb_flutter/mxflutter.git

[0087] 2、在pubspec.yaml文件中引入mxflutter_js_bundle JS Bundle资源,如果不配置的话,mxflutter_js_bundle不会被打包进入App.apk或App.ipa。

[0088] flutter:

[0089] assets:

[0090] -mxflutter_js_bundle/

[0091] 3、在flutter目录调用Flutter pub get安装。

[0092] 本发明抽离DartVM,单独编译出一个DartVM,打包成动态库;通过JavasSriptCore替换DartVM,最终以动态生成DSL的形式实现。本发明总结了现有方案的不足,充分结合现有项目、插件、AVMS平台支持的优势,通过下发增量/全量更新包,达到动态更新Flutter iOS项目的效果。

[0093] 本实施例提供一种实现所述基于JavaScript的Flutter热更新方法的发布JS Bundle到AVMS的具体过程,包括:

[0094] 1、打包支持生产模式bundle的构建,并可以兼容低版本jscore(ios 9-10):

[0095] npm run build

[0096] 2、根据AVMS的校验规则,把生成的.js目录拷贝到prd7文件下:

[0097]
```
F0000001.zip/
├── prd7/
│   └── F0000001/
│       └── bundle-F0000001.js
└── map7
```

[0098] 3、把zip上传到AVMS。

[0099] 本发明在前端(APP端)方向即实现了使用TypeScript来编写,使用Flutter Widget的描述方式来开发业务。对前端发展方向的规划是,实现web前端dom(Document Object Model,文档对象模型)+css(Cascading Style Sheets,层叠样式表)的开发方式来接入Flutter,通过Vue/React等前端熟悉的开发框架来编写业务代码,自动转换为Flutter Widget进行渲染,实现前端开发者的零成本接入。

[0100] 本发明轻量化Flutter开发环境,适合大前端开发业务,能通过AVMS增量更新,保持原有的渲染性能,支持Dart Flutter语法,JavasSriptCore是iOS官方库,无需增加安装包,Dart代码和JS代码非常相近,方便转换,JavasSriptCore与Native有更方便的互调接口,JS的执行效率是DartVM的3倍,编码1M的JSON只需2毫秒。

[0101] 本发明所述的基于JavaScript的Flutter热更新方法的保护范围不限于本实施例列举的步骤执行顺序,凡是根据本发明的原理所做的现有技术的步骤增减、步骤替换所实现的方案都包括在本发明的保护范围内。

[0102] 本发明还提供一种基于JavaScript的Flutter热更新装置,所述基于JavaScript的Flutter热更新装置可以实现本发明所述的基于JavaScript的Flutter热更新方法,但本发明所述的基于JavaScript的Flutter热更新方法的实现装置包括但不限于本实施例列举的基于JavaScript的Flutter热更新装置的结构,凡是根据本发明的原理所做的现有技术的结构变形和替换,都包括在本发明的保护范围内。

[0103] 参见图6A所示,本发明实施例还提供一种基于JavaScript的Flutter热更新装置,所述基于JavaScript的Flutter热更新装置600包括:前端开发模块610和JS Bundle模块620。所述客户端与所述JS Bundle模块620通信相连,将所述JS Bundle模块620输出的JS Bundle文件解析为Dart语言,进而显示更新后的Flutter控件。

[0104] 参见图6B所示,本发明实施例还提供一种基于JavaScript的Flutter热更新装置,所述基于JavaScript的Flutter热更新装置600包括:前端开发模块610,JS Bundle模块620,系统模拟器630和分发模块640。所述客户端与所述分发模块640通信相连,用于检测更新获得所述调试后的JS Bundle文件,并将所述调试后的JS Bundle文件解析为Dart语言,进而显示更新后的Flutter控件。

[0105] 所述前端开发模块610用于提供基于TypeScript语言描述的应用业务代码。其中,

应用业务代码包括适用于移动、Web、桌面和嵌入式平台的业务代码。TypeScript是一种由微软开发的自由和开源的编程语言，它是JavaScript(简写JS)的一个超集，扩展了JavaScript的语法。

[0106] 所述JS Bundle模块620与所述前端开发模块610通信相连，将所述基于TypeScript语言描述的应用业务代码编译成JS Bundle文件。所述JS Bundle模块以Script Widget描述方式将所述基于TypeScript语言描述的应用业务代码编译为基于JavaScript语言描述的JS Bundle文件；所述JS Bundle文件用于被解析为Dart语言，进而实现Flutter热更新。所述客户端包括iOS客户端。本发明通过JS Bundle模块620使得Flutter热更新适用于iOS设备，解决了现有技术中Flutter的热更新不支持iOS设备的问题。

[0107] 所述系统模拟器630与所述JS Bundle模块620通信相连，用于对所述JS Bundle文件进行调试；所述系统模拟器包括iOS模拟器。

[0108] 于本发明的一实施例中，所述系统模拟器630还包括Android模拟器。本发明所述的基于JavaScript的Flutter热更新方法不但可以支持iOS设备，还可以支持Android设备，这样就使得开发者只需要开发一套代码就能够同时应用于iOS设备和Android设备，无需分别开发2套代码来分别适用于iOS设备和Android设备，极大地节省了开发成本，提高了开发效率，降低了更新维护成本。

[0109] 所述分发模块640与所述系统模拟器630通信相连，用于发布调试后的JS Bundle文件。

[0110] 进一步，可以将调试好的JS Bundle文件打包成zip压缩包，然后上传到分发模块(AVMS)，使native通过分发的7z包解析pluginId打开Flutter页面。native是一个计算机函数，一个Native Method就是一个Java调用非Java代码的接口，方法的实现由非Java语言实现，比如C或C++。

[0111] 所述客户端与所述分发模块640通信相连，用于检测更新获得所述调试后的JS Bundle文件，并将所述调试后的JS Bundle文件解析为Dart语言，进而显示更新后的Flutter控件。

[0112] 于本发明的一实施例中，所述客户端调用JavaScript版的轻量级Flutter Runtime将所述JS Bundle文件生成UI描述，并将所述UI描述传递给Dart层的UI引擎；所述Dart层的UI引擎将所述UI描述生成Flutter控件。

[0113] 进一步，所述Dart层的UI引擎将所述UI描述生成Flutter控件的实现过程包括：所述Dart层的UI引擎获取所述JS Bundle文件的UI描述；所述JS Bundle文件的UI描述为一虚拟Script Widget Tree；所述Dart层的UI引擎根据所述虚拟Script Widget Tree对应转换为Flutter Widget Tree；所述Dart层的UI引擎编译所述Flutter Widget Tree获得Flutter Element Tree；所述Dart层的UI引擎编译所述Flutter Element Tree获得Flutter Render Tree，即获得所述Flutter控件。

[0114] 所述虚拟Script Widget Tree的结构与所述Flutter Widget管理的Flutter Widget Tree的结构相互对应；所述虚拟Script Widget Tree的结构包括虚拟Statefull Widget、虚拟Row Widget、虚拟Text Widget、虚拟Container Widget和虚拟Image Widget；所述Flutter Widget Tree的结构包括Statefull Widget、Row Widget、Text Widget、Container Widget和Image Widget。

[0115] 本发明中,利用Script Widget管理一个Script页面或控件;所述Script Widget负责创建管理虚拟Script Widget Tree,并以自增ID与Flutter Widget对应相互调用;所述Script Widget每次编译(Build)都会创建一个新的虚拟Script Widget Tree。

[0116] 本发明可以支持TypeScript语言和前端生态,可以支持Flutter中同名Widget类,相同API;支持Flutter相同的Build方式,setState刷新及事件响应方法,支持JS(JavaScript)和Dart双向调用通道,支持模拟器页面Hot Reload,支持Safari和Chrome调试,支持编译现有Flutter程为JS,运行在框架之上。

[0117] 参见图4所示,本发明实施例提供一种实现所述基于JavaScript的Flutter热更新方法的硬件架构,包括前端编码层(App Code(TS)),UI框架层(JS UI Framework),Dart层,其中Dart层包括Flutter UI Engine(UI引擎),Flutter Framework(Flutter框架),Engine(底层引擎)。前端编码层(App Code(TS))可以实现基于TypeScript语言编译应用业务代码。UI框架层(JS UI Framework)可以实现以Script Widget描述方式将所述基于TypeScript语言描述的应用业务代码编译为基于JavaScript语言描述的JS Bundle文件。Dart层可以实现将JS Bundle文件解析为Dart语言,进而实现Flutter热更新。其中,UI框架层(JS UI Framework)可以轻量的响应UI框架,使用TS语言,使用npm前端包开发,具有1400个Widget和相关类,具有与Flutter相同的API,具有CLI工具。Dart层中的Flutter UI Engine(UI引擎)可以支撑DSL解析,支持UI刷新逻辑,支持事件响应,支持Flutter API。Dart层中的Flutter Framework(Flutter框架)打通了JS与Dart之间的双向高速通道,具有JSVM管理功能,具有JS资源管理功能,支持JS常规模式请求(JS common model require)。

[0118] 参见图5所示,本发明实施例提供一种实现所述基于JavaScript的Flutter热更新方法的渲染原理,包括:ScriptWidget管理一个Script页面或控件,负责创建管理ScriptWidgetTree,以自增ID与Flutter对应Widget相互调用,每次编译(Build)都会创建一个新的WidgetTree。ScriptWidgetTree的结构参见图5左侧树状结构,即VM层的Widge配置树,Flutter层的Widge配置树结构参见图5右侧树状结构。

[0119] 需要说明的是,应理解以上系统的各个模块的划分仅仅是一种逻辑功能的划分,实际实现时可以全部或部分集成到一个物理实体上,也可以物理上分开。且这些模块可以全部以软件通过处理元件调用的形式实现,也可以全部以硬件的形式实现,还可以部分模块通过处理元件调用软件的形式实现,部分模块通过硬件的形式实现。

[0120] 参见图7所示,本发明实施例还提供一种计算机设备,包括存储器710、处理器720以及存储在所述存储器中并可在所述处理器上运行的计算机程序,所述处理器执行所述计算机程序时实现上述基于JavaScript的Flutter热更新方法的步骤。

[0121] 该计算机设备包括通过系统总线连接的处理器、存储器、网络接口和数据库。其中,该计算机设备的处理器用于提供计算和控制能力。该计算机设备的存储器包括非易失性和/或易失性存储介质、内存储器。该非易失性存储介质存储有操作系统、计算机程序和数据库。该内存储器为非易失性存储介质中的操作系统和计算机程序的运行提供环境。该计算机设备的网络接口用于与外部的客户端通过网络连接通信。该计算机程序被处理器执行时以实现一种基于人工智能的智能问答处理方法服务端侧的功能或步骤。

[0122] 本发明还提供一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,所述计算机程序被处理器执行时实现上述基于JavaScript的Flutter热更新方法的

步骤。

[0123] 本申请所描述的计算机程序可以从计算机可读存储介质下载到各个计算/处理设备，或者通过网络、例如因特网、局域网、广域网和/或无线网下载到外部计算机或外部存储设备。

[0124] 需要说明的是，上述关于计算机可读存储介质或计算机设备所能实现的功能或步骤，可对应参阅前述方法实施例中，服务端侧以及客户端侧的相关描述，为避免重复，这里不再一一描述。

[0125] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程，是可以通过计算机程序来指令相关的硬件来完成，所述的计算机程序可存储于一非易失性计算机可读取存储介质中，该计算机程序在执行时，可包括如上述各方法的实施例的流程。其中，本申请所提供的各实施例中所使用的对存储器、存储、数据库或其它介质的任何引用，均可包括非易失性和/或易失性存储器。非易失性存储器可包括只读存储器(ROM)、可编程ROM(PROM)、电可编程ROM(EPROM)、电可擦除可编程ROM(EEPROM)或闪存。易失性存储器可包括随机存取存储器(RAM)或者外部高速缓冲存储器。作为说明而非局限，RAM以多种形式可得，诸如静态RAM(SRAM)、动态RAM(DRAM)、同步DRAM(SDRAM)、双数据率SDRAM(DDRSDRAM)、增强型SDRAM(ESDRAM)、同步链路(Synchlink)DRAM(SLDRAM)、存储器总线(Rambus)直接RAM(RDRAM)、直接存储器总线动态RAM(DRDRAM)、以及存储器总线动态RAM(RDRAM)等。

[0126] 所属领域的技术人员可以清楚地了解到，为了描述的方便和简洁，仅以上述各功能单元、模块的划分进行举例说明，实际应用中，可以根据需要而将上述功能分配由不同的功能单元、模块完成，即将所述装置的内部结构划分成不同的功能单元或模块，以完成以上描述的全部或者部分功能。

[0127] 综上所述，本发明有效克服了现有技术中的种种缺点而具高度产业利用价值。

[0128] 上述实施例仅例示性说明本发明的原理及其功效，而非用于限制本发明。任何熟悉此技术的人士皆可在不违背本发明的精神及范畴下，对上述实施例进行修饰或改变。因此，举凡所属技术领域中具有通常知识者在未脱离本发明所揭示的精神与技术思想下所完成的一切等效修饰或改变，仍应由本发明的权利要求所涵盖。

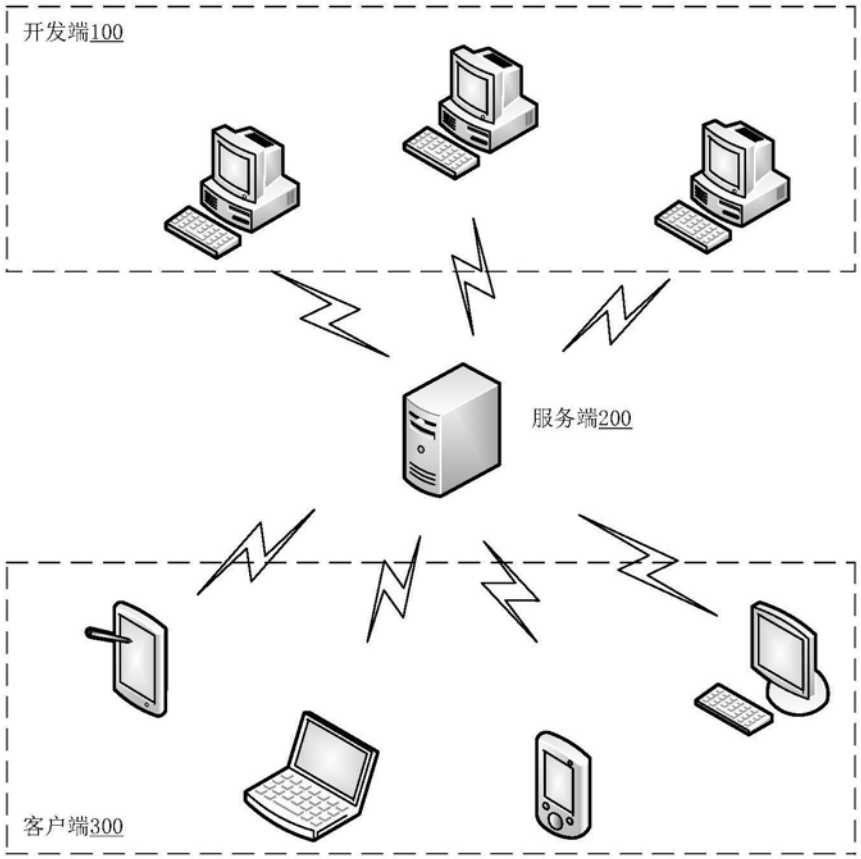

开发端100
服务端200
客户端300

图1A

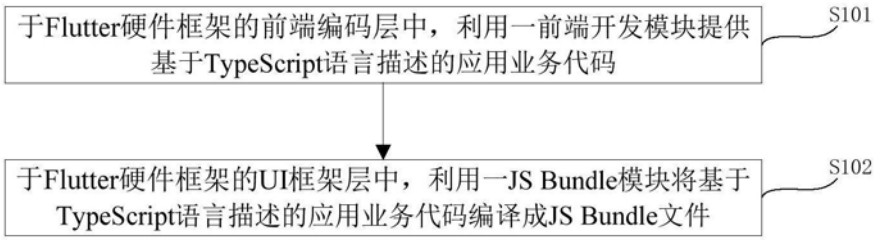

于Flutter硬件框架的前端编码层中，利用一前端开发模块提供基于TypeScript语言描述的应用业务代码
S101
于Flutter硬件框架的UI框架层中，利用一JS Bundle模块将基于TypeScript语言描述的应用业务代码编译成JS Bundle文件
S102

图1B

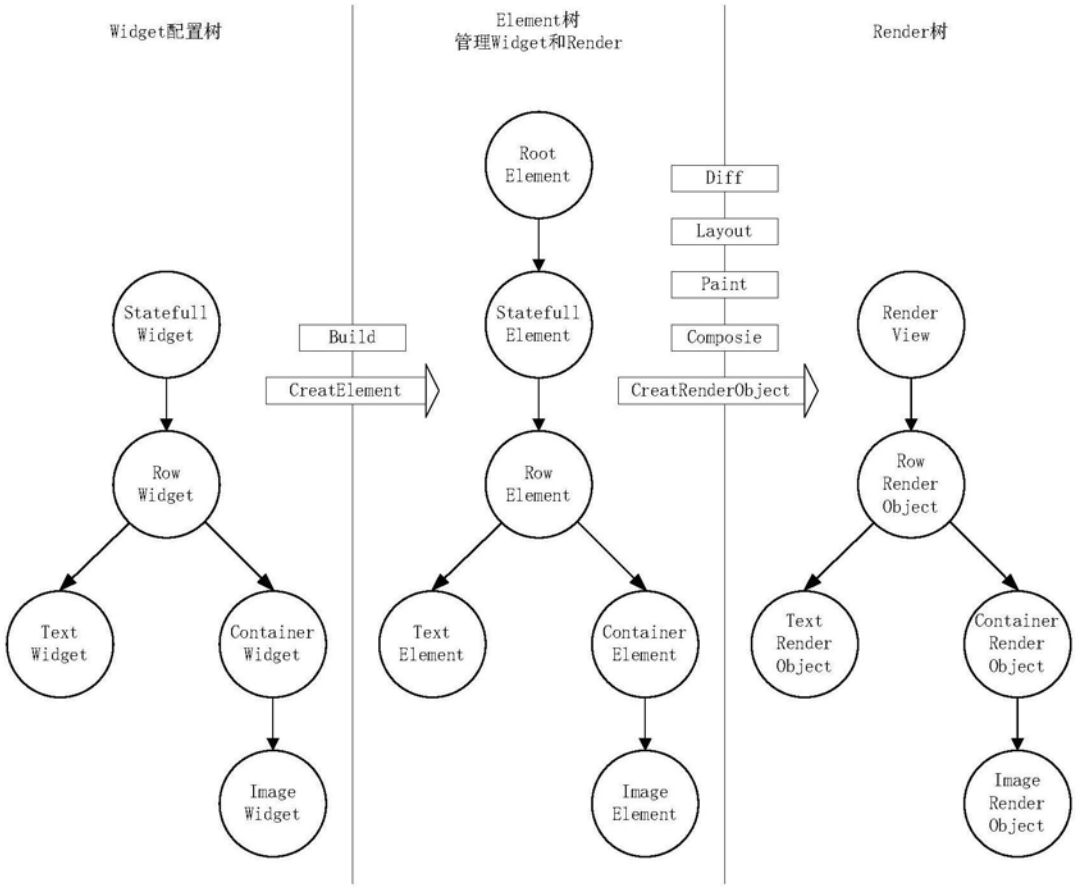

Widget配置树
Element树
管理Widget和Render
Render树
Root Element
Diff
Layout
Paint
Composie
Statefull Widget
Build
CreatElement
Statefull Element
CreatRenderObject
Render View
Row Widget
Row Element
Row Render Object
Text Widget
Container Widget
Text Element
Container Element
Text Render Object
Container Render Object
Image Widget
Image Element
Image Render Object

图2

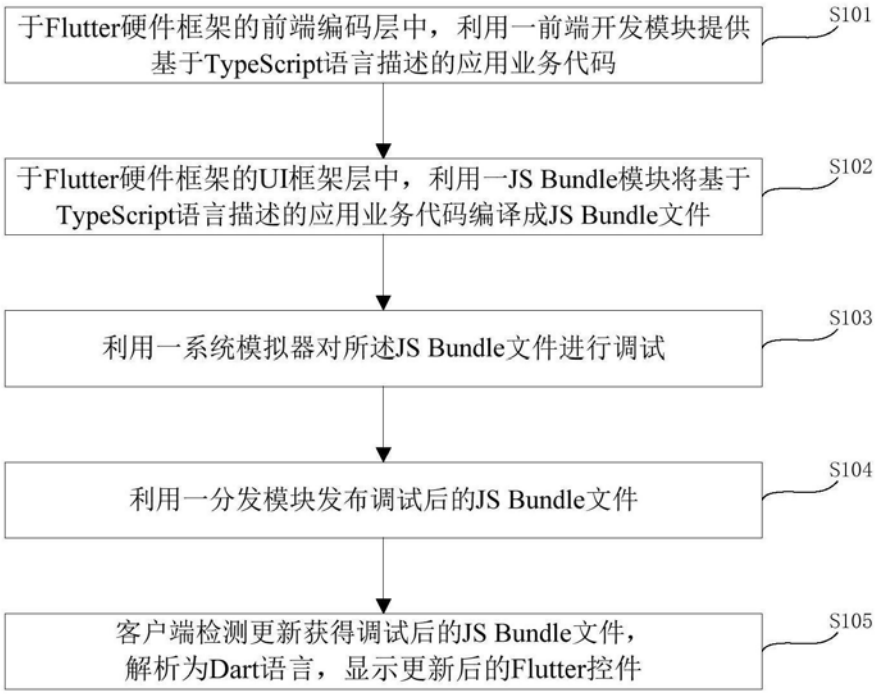

于Flutter硬件框架的前端编码层中，利用一前端开发模块提供基于TypeScript语言描述的应用业务代码
S101
于Flutter硬件框架的UI框架层中，利用一JS Bundle模块将基于TypeScript语言描述的应用业务代码编译成JS Bundle文件
S102
利用一系统模拟器对所述JS Bundle文件进行调试
S103
利用一分发模块发布调试后的JS Bundle文件
S104
客户端检测更新获得调试后的JS Bundle文件，解析为Dart语言，显示更新后的Flutter控件
S105

图3

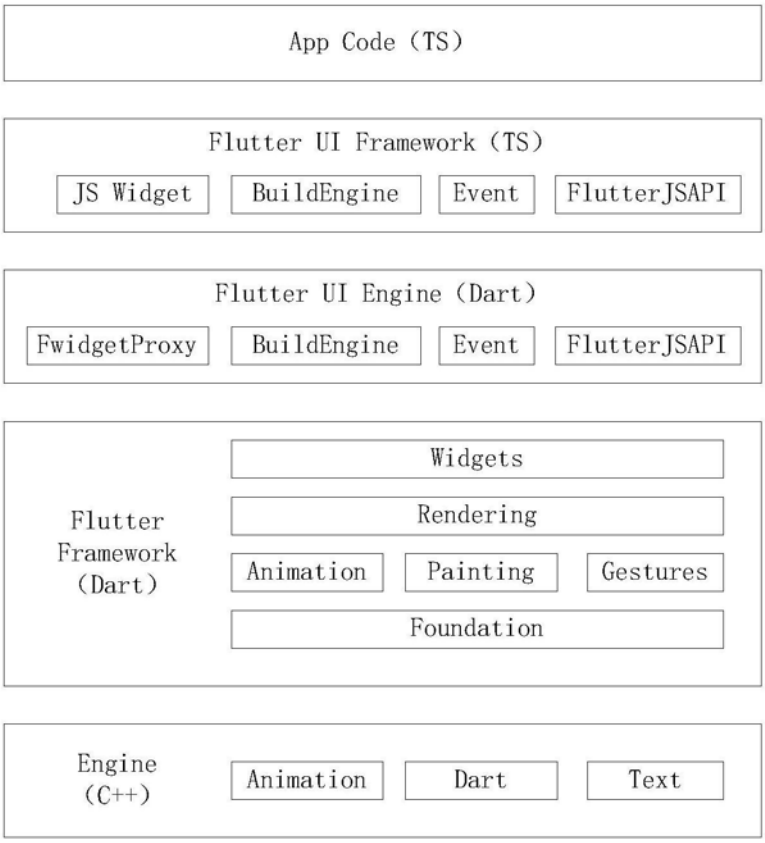

App Code (TS)
Flutter UI Framework (TS)
JS Widget
BuildEngine
Event
FlutterJSAPI
Flutter UI Engine (Dart)
FwidgetProxy
BuildEngine
Event
FlutterJSAPI
Flutter Framework (Dart)
Widgets
Rendering
Animation
Painting
Gestures
Foundation
Engine (C++)
Animation
Dart
Text

图4

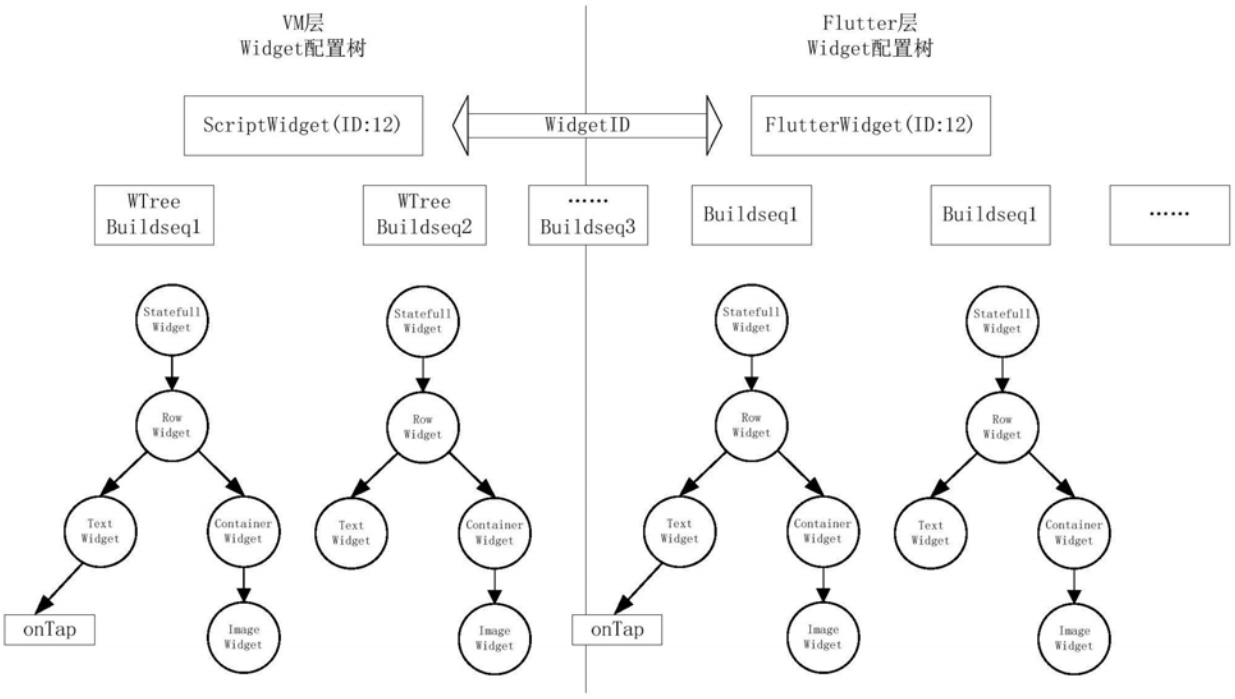

VM层 Widget配置树
Flutter层 Widget配置树
ScriptWidget(ID:12)
WidgetID
FlutterWidget(ID:12)
WTree Buildseq1
WTree Buildseq2
Buildseq3
Buildseq1
Buildseq1
Statefull Widget
Row Widget
Text Widget
Container Widget
Image Widget
onTap

图5


基于JavaScript的Flutter热更新装置(600)
前端开发模块
610
JS Bundle模块
620
客户端

图6A

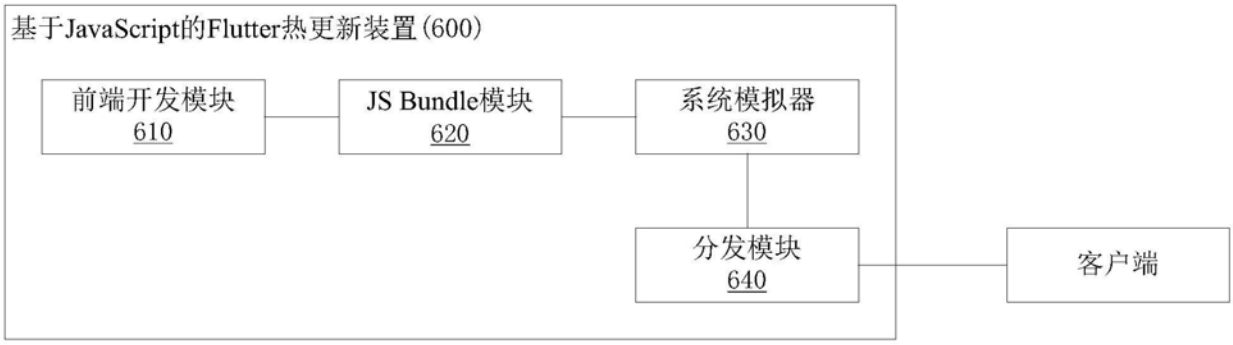

基于JavaScript的Flutter热更新装置(600)
前端开发模块
610
JS Bundle模块
620
系统模拟器
630
分发模块
640
客户端

图6B

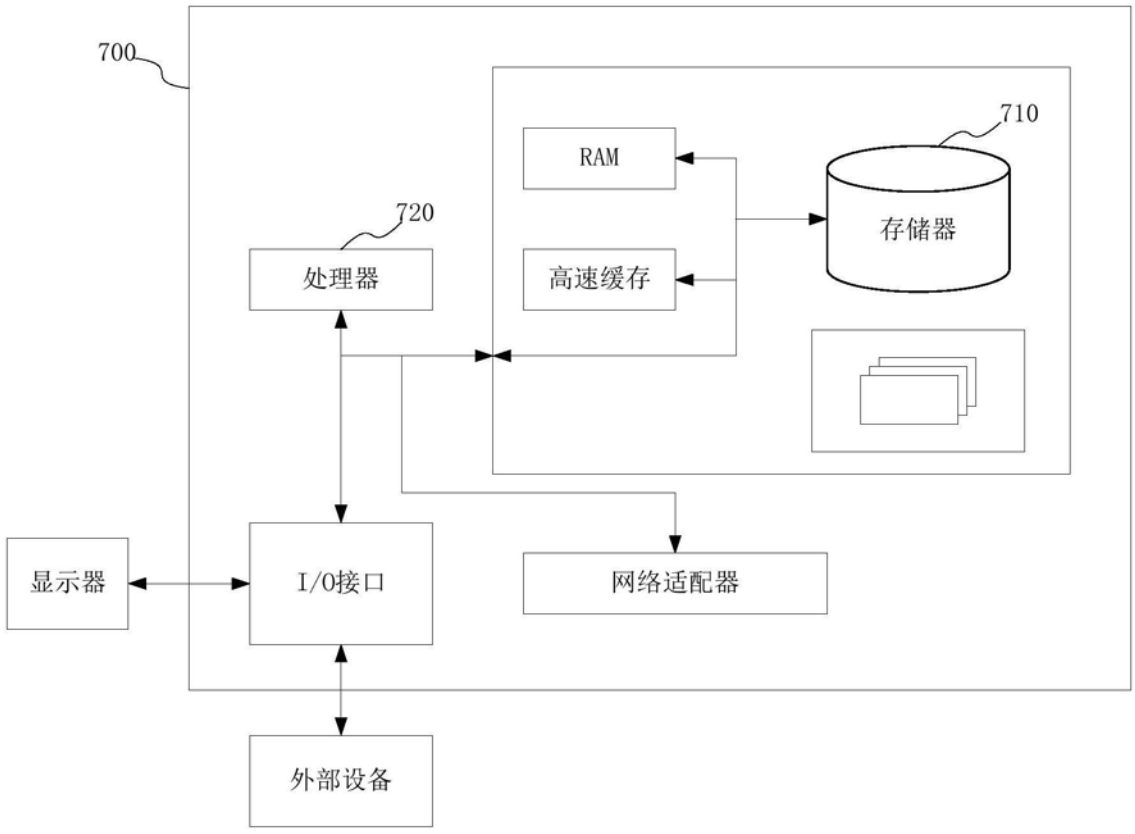

700
710
RAM
存储器
720
处理器
高速缓存
显示器
I/O接口
网络适配器
外部设备

图7