Technical Disclosure Commons

Defensive Publications Series

May 11, 2018

DYNAMICALLY GENERATED DEVICE-SPECIFIC APPLICATION EXECUTABLES

John Ayres

Anthony Morris

Milena Nikolic

Yaakov Gluck

Pierre Lecesne

See next page for additional authors

Inventor(s)
John Ayres, Anthony Morris, Milena Nikolic, Yaakov Gluck, Pierre Lecesne, and Idries Hamadi

DYNAMICALLY GENERATED DEVICE-SPECIFIC APPLICATION EXECUTABLES

ABSTRACT

An application management system is described that dynamically generates and disseminates application executables based on device information of a computing device requesting installation of the application. Using application bundles that include the resources required to execute the application on all of the supported devices, the application management system determines the portion of the resources included in the application bundle that are required to execute the application at the particular device that is requesting the application and dynamically generates the application executable using only the required resources. In some examples, each application may be broken into a number of modules that each provide a portion of the all of the possible features of the application. In such examples, in response to receiving a request for the initial installation of the application, the application management system may generate the executable binary for the application that includes only the module that provides the core functionality of the application. If the user of the device later attempts to use additional features of the application not enabled by the core module, the application management system may dynamically generate a package that includes the requested functionality such that the package only includes the resources required to provide the requested functionality at the particular device.

DESCRIPTION

Modern applications execute on mobile phones, counter-top devices, automobiles, and many other types of computing devices. Software developers typically include application resources for the application to inspect certain design aspects (e.g., screen size, screen resolution, etc.), settings (e.g., language, current network connectivity, battery settings, etc.), and functional capabilities (e.g., processor capabilities, network capabilities, operating system, etc.) of the computing device executing the application. However, the computing device may only utilize particular resources (e.g., software libraries, images, languages, etc.) within the application that fit the design aspects and functional capabilities of the computing device and that are required to provide application functionality requested by the user. However, as the application generally includes all of the resources to support all of the functionality of the application and to support all of the desired devices, the computing devices stores more resources that strictly required to provide the desired application functionality to the user.

Rather than installing all of the application resources in a single application executable, an application management system (hereinafter "the manager") may dynamically extract and compile resources from an application bundle into an executable application installable and executable by a client computing device (hereinafter, "the client"). In general, the application bundle includes all of the resources required to provide all of the application functionality on all of the supported devices, operating system versions, languages, etc. The application bundle is constructed by a software developer who provided the application bundle to the manager (e.g., an online application store that hosts various applications provided by developers).

When the client initializes installation of the application, either the client provides information about the hardware and software configuration of the client to the manager, or the manager accesses a database with the information about the client based on the brand and model of the client. The configuration information includes device characteristics (e.g., screen size, screen resolution, etc.), settings (e.g., language, current network connectivity, battery settings, etc.), and functional capabilities (e.g., processor capabilities, network capabilities, operating system, etc.) of the client. Using this configuration information for the client, the manager selects the required resources from the application bundle and generates the executable such that it only includes the resources required to execute the application on the client, which may reduce the size of the executable that needs to be downloaded and installed at the client.

For example, a developer may choose to support two different display shapes: a typical rectangular display that includes four sharp, distinct corners, and a non-rectangular display that includes rounded corners or that floods the display around various hardware components to create more than four "corners" of the display. In such examples, the application bundle stored on the server would include the resources to enable the application graphical user interface to fit the non-rectangular display as well as the rectangular display.

In instances where the manager receives a request for the application from a device having the non-rectangular display, the manager extracts the resources that enable the application to support non-rectangular display and includes those resources in the executable dynamically generated for the application, but does not include the resources that enable the application to support the rectangular display. In instances where the manager receives a request from a client device has a typical rectangular display, the manager extracts the resources that enable the application to support the rectangular display and includes those resources in the executable dynamically generated for the application, but does not include the resources that enable the application to support the non-rectangular display.

In some examples, the manager may further generate and provide executables based on discrete sets of functionality of the application. The developer may choose to divide up an application bundle into two or more discrete modules, each of which provides a portion of the overall functionality of the application. For example, when a user initially downloads the application, the manager provides an application executable that supports core functionality of the application. If the user attempts to use other functionality of the application that is not provided by the initial executable, the initial executable for the application requests one or mero additional executables that provide the requested functionality and coordinates the installation of those additional executables. When generating the initial executable or any additional executables, the manager may generate those executables based on the confirmation information for the client, as discussed above. That is, each executable, whether it that includes code for all of the functionality for the application or whether it only includes code for a portion of the functionality of the application, only includes the resources required for the respective executable to properly execute on the client.

While every application resource for the application collectively may be large in size, a client and manager that implements these techniques may retrieve, install, and execute only those application packages necessary for the application to properly execute on a particular client. As such, when installing and/or executing the application, only a portion of the application resources for the application need to be requested from the manager to reside locally at the client for the application to execute, freeing up space in the data storage component of the client. In this way, applications may install and load more quickly by bypassing having the client device select which packages to utilize and which packages to ignore when executing the application locally. Further, by decreasing the size of the downloaded application, the computing device consumes less mobile data in downloading the application and decreases the amount of time needed to complete the download. In examples where the application developer submits multiple application bundles for an application, if the initial executable binary of the application bundle only includes basic functionality, when a user attempts to perform additional functionality for the application not enabled by the initial executable binary, the application may automatically retrieve the needed executable binary generated by the manager from additional application bundles for the application. The manager generates the additional executable binary for the particular client based on the design and performance capabilities of the client and the additional application bundles. In this way, applications may install and load more quickly because the superfluous modules for the application need not be retrieved and installed if only a subset of the application's functionality is needed. Although an application executing on a mobile device is described throughout the disclosure, an application executing in any computing environment could benefit from use of the techniques described herein.

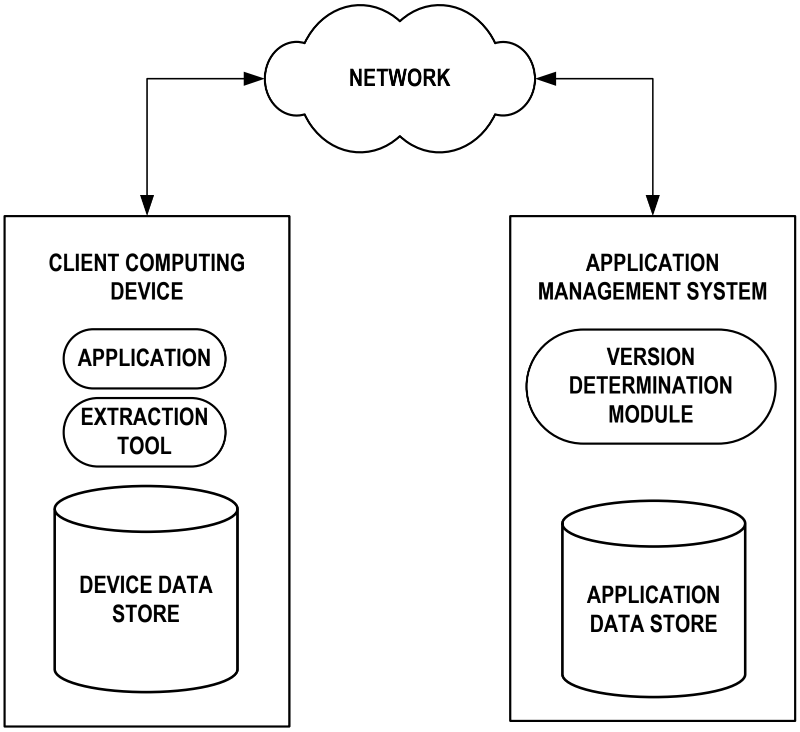

Figure 1

Consider the example system shown in Figure 1 (above) which is configured to provide an application environment in accordance with the techniques described herein. The system includes a client computing device and an application management system that communicate with each over a network. The network represents a combination of any one or more public or private communication networks, for instance, television broadcast networks, short-wavelength wireless networks, cable or satellite networks, cellular networks, Wi-Fi networks, broadband networks, and/or other type of network for transmitting data (e.g., telecommunications and/or media data) between various computing devices, systems, and other communications and media equipment.

The client computing device (the "client") may represent any type of computing device that is configured to execute an application and communicate on a network. Examples of the client computing device include mobile phones, tablet computers, wearable computing devices, countertop computing devices, home automation computing devices, laptop computers, desktop computers, televisions, stereos, automobiles, and all other type of mobile and non-mobile computing devices that are configured to execute an application.

The application management system (the "manager") may represent any type of computing device, server, cloud computing system, or other system that is configured to execute communicate on a network with a client computing device. In some examples, the manager may include shared assets of multiple users. For example, the manager stores various application bundles for a number of different developers and enables installation of the applications at various client computing devices.

Developers provide the application bundles to the manager for each of the applications the developer would like to make available for download and install to client devices. If a developer chooses to split the application functionality across multiple bundle modules, the developer may submit a bundle with a base application module that includes a base set of functionality for the application as well as additional modules that enable additional functionality of the application. Each module in the application bundle, whether providing all or only a portion of the functionality of each application, includes all of the code and other resources needed to execute the application on all of the supported devices and configurations.

When the client requests installation of an application, the manager dynamically constructs a device-specific executable of the application for the client. In generating the executable, the manager extracts only the resources from the application bundle that are needed for the client based on the various settings and capabilities of the client (e.g., as provided by the client or as determined by the manager).

In one example of Figure 1, the application may be an application to access a social media platform with multiple functions. The base functionality of the application includes searching through profiles of other users on the social media platform and viewing public messages posted by users that the user of the client is connected to on the social media platform. The application also includes additional functionality, such as real-time video output, audio calls to a different client computing device, private text messaging, accessing the client's camera to capture a photograph, and applying a graphical filter to an image, among other things. In other examples, the application may be any application that may perform multiple functions, such as navigation applications, gaming applications, organization applications, entertainment streaming applications, or a ride-sharing application, among other things.

The client sends an initial request to the manager for the social media application. In some instances, this request includes various configuration information for the client. In other instances, this request only includes a brand and model identifier of the client such that the manager may determine the device configuration information by looking up the brand and model in a database or other data structure. The manager receives this request and, using the version determination module, analyzes the configuration information for the client to determine which resources of the application bundle to extract and include in the executable. For instance, a first set of resources may be included in the executable that is sent to a client device with a smaller screen size, a lower screen resolution, and that runs with an older version of an operating system. In other instances, a second set of resources may be included in the executable that is sent to a different client device with a larger screen size, a higher screen resolution, and that runs with the newest version of the operating system.

The client receives the generated executable and installs the executable at the client. In instances where the developer only submits a single application bundle to the application management system, the executable includes all of the resources need to utilize all of the functionality available within the application. In instances where the application developer submits multiple application bundles into the application ecosystem for different sets of functions for a single application, the initial executable provided by the manager and installed at the client may only include a base set of functionality of the application.

After installing the initial executable, the client executes the application to perform the base functions, which include accessing the social media platform to search through profiles of other users on the social media platform and viewing public messages posted by users that the user of the client is connected to on the social media platform. In executing the application, the client outputs a graphical user interface that includes multiple interactive graphical elements. Each interactive graphical element indicates a different, distinct function that the application is capable of performing during the execution of the application.

If the developer split the application functionality between multiple application modules and prior to any of interactive graphical elements that utilize functionality not provided by the core executable being selected, the executable for the application only includes the executable generated from the base application module for the application. However, once a user selects one of the interactive graphical elements to utilize functionality not provided by the base executable (e.g., to apply a graphical filter to an image), the application may automatically request one or more additional application modules as executables that provide the requested functionality. That is, the application automatically sends a request to the manager for the application executable of the application that enables the client device to apply a graphical filter to an image, as well as the various display and functional capabilities of the client.

The manager receives this request from the client and generates an executable that provides the requested functionality to send back to the client from an application module that includes the resources required for the application to provide the requested functionality. In generating this executable, the manager, using the device configuration information, identifies and extracts the portions of the application bundle that includes the resources required to execute the executable on the client. After generating the executable, the manager sends the executable to the client.

The client receives the executable and installs the additional executable for the application at the client such that the client may now execute functions associated with the additional executable. With regards to the current example, after the client device receives the indication to apply the graphical filter to the image and retrieves the requisite executable from the manager, the client device begins executing the application with the additional executable such that the user may apply the graphical filter to the image.

In some examples of such application splitting, cellular data restrictions may make it undesirable for the executables of the additional modules to be sent over a cellular network. In such instances, the client may save an indication of the module the user wishes to access for it to be delivered at a later time, such as when the client has a wireless internet connection (e.g., WiFi®). For instance, the client may, upon initiating a wireless internet connection, the client may send the request for the executables of the additional models.

In a further effort to reduce the number of instances that application executables are transferred over the network, the client, while executing the application in the foreground, may send indications of additional modules that the user is requesting to the manager. The manager may track these modules. When an update to the application comes available through the application ecosystem run by the manager, the manager may construct an executable that includes the updated base module for the application, as well as the additionally requested modules tracked by the manager. The manager may then send these multiple modules as a single executable.

In some examples, if the application developer submits multiple application modules for a single application bundle, a client may not support such application splitting. In these examples, rather than only sending an initial executable that provides the base functionality of the application, the manager may construct the executable binary to include the requisite resources of every application module for the application bundle submitted by the application developer to provide full functionality to the client in response to receiving the initial request to install the application.

During the course of operation by the user, the settings and/or capabilities of the client may change. For instance, the user may update a language package installed on the client device, or the client device may be operatively connected to an external display component with a different size and resolution than the client device. If any of the settings or capabilities that may affect the execution of the are updated, the client may request an updated executable based on the updated settings or capabilities of the client. As one example, if the client device has a standard definition display, the initially received executable may be generated by the manager to have the application packages that optimally configure the application for use on a standard definition display. If the client device, at a later time, were to connect to an external high definition display, the client device may request a subsequent executable that includes the application packages that optimally configure the application for use on a high definition display. The client device can replace the initial version of the application with the updated version configured for a high definition display.

Taken as a whole, every application resource submitted by an application developer for an application may be large in size, even though certain portions of the application resources may never be used by a particular client due to incompatible settings or capabilities. A client and manager that implements these techniques may retrieve, install, and execute only those resources necessary for the application to properly execute on a particular client. As such, when installing and/or executing the application, only a portion of the application resources for the application need to be requested from the manager to reside locally at the client for the application to execute, freeing up space in the data storage component of the client. Further, by decreasing the size of the downloaded application, the computing device consumes less mobile data in downloading the application and decreases the amount of time needed to complete the download. In examples where the application developer submits multiple application bundles for an application, if the initial executable of the application bundle only includes a base set of functionality for the application, when a user attempts to use additional functionality of the application not provided by the initial executable, the application may automatically retrieve and install the executable needed to provide the requested functionality from the manager. In this way, applications may install as less data needs to be downloaded and installed.