
IMAGE ACQUISITION METHOD, DEVICE, SERVER AND STORAGE MEDIUM

RELATED APPLICATION

[0001] This application claims priority to Chinese Patent Application No. 201910468135.3, entitled "IMAGE OBTAINING METHOD AND APPARATUS, SERVER, AND STORAGE MEDIUM" and filed with the China National Intellectual Property Administration on May 29, 2019.

FIELD OF THE TECHNOLOGY

[0002] This application relates to the field of internet technologies, and in particular, to an image obtaining method and apparatus, a server, and a storage medium.

BACKGROUND OF THE DISCLOSURE

[0003] Cloud gaming is an online gaming technology. In a cloud gaming scene, a game runs on a cloud server. The cloud server renders a game scene into a game picture and transmits the game picture to a user terminal in real time, for a user on the user terminal to input a corresponding operation for the game picture. As can be seen, the cloud gaming can implement running of a high-quality game on a user terminal with relatively limited graphics processing and data computing capabilities.

[0004] Due to the foregoing advantage, the cloud gaming has become a research hotspot in the gaming field in recent years. One of the key problems involved in the study of cloud gaming is how a cloud server obtains a game picture of a currently running game. A commonly used method to resolve this problem is to obtain a picture based on a virtual machine. Specifically, a virtual machine is configured for each user terminal, the virtual machine renders a game scene to obtain a game picture, and the cloud server directly intercepts a desktop of the virtual machine to obtain the game picture. However, this method requires deployment of a large number of virtual machines, which occupies a large amount of server resources, thereby resulting in waste of server resources.

SUMMARY

[0005] Embodiments of this application provide an image obtaining method and apparatus, a server, and a storage medium. The server intercepts window image data generated during running of a target application, and performs synthesis processing on the window image data, to obtain a

user interface image to be displayed corresponding to the target application. Compared with the manner of obtaining a user interface image by using a virtual machine in the related art, resource overheads for configuring the virtual machine are reduced.

[0006] In one aspect, an embodiment of this application provides an image obtaining method, applicable to a server, the method including:

obtaining, from an application process set, a target application process corresponding to a user identifier;

calling, when detecting that an image rendering function in the target application process is called, a data interception module to intercept a plurality of window image data currently generated;

performing image synthesis processing on the plurality of window image data, to obtain a user interface image to be displayed; and

transmitting a notification message carrying the user interface image to a user terminal corresponding to the user identifier, so that the user terminal displays the user interface image on a user interface.

[0007] In another aspect, an embodiment of this application provides an image obtaining apparatus, configured in a server, the apparatus including:

an obtaining unit, configured to obtain, from an application process set, a target application process corresponding to a user identifier;

an interception unit, configured to call, when detecting that an image rendering function in the target application process is called, a data interception module to intercept a plurality of window image data currently generated;

a first processing unit, configured to perform image synthesis processing on the plurality of window image data, to obtain a user interface image to be displayed; and

a transmission unit, configured to transmit a notification message carrying the user interface image to a user terminal corresponding to the user identifier, so that the user terminal displays the user interface image on a user interface.

[0008] In still another aspect, an embodiment of this application provides a server. The server includes:

a processor, suitable for implementing one or more instructions; and

a computer storage medium, storing one or more instructions, the one or more instructions being suitable to be loaded by the processor to perform the image obtaining method described above.

[0009] In still another aspect, an embodiment of this application provides a computer storage medium, storing one or more instructions, the one or more instructions being suitable to be loaded by a processor to perform the image obtaining method described above.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] To describe the technical solutions of the embodiments of this application or the related technology more clearly, the following briefly introduces the accompanying drawings required for describing the embodiments or the related technology. The accompanying drawings in the following description show some embodiments of this application, and a person skilled in the art may still derive other drawings from these accompanying drawings without creative efforts.

[0011] FIG. 1 is a schematic architecture diagram of image obtaining.

[0012] FIG. 2 is an architectural diagram of image obtaining according to an embodiment of this application.

[0013] FIG. 3a is a desktop view of a cloud server according to an embodiment of this application.

[0014] FIG. 3b is a schematic diagram of a user interface image according to an embodiment of this application.

[0015] FIG. 4 is a schematic flowchart of an image obtaining method according to an embodiment of this application.

[0016] FIG. 5a is a schematic diagram of obtaining window image data according to an embodiment of this application.

[0017] FIG. 5b is a schematic diagram of window image data according to an embodiment of this application.

[0018] FIG. 5c is a schematic diagram of another window image data according to an embodiment of this application.

[0019] FIG. 5d is a schematic diagram of still another window image data according to an embodiment of this application.

[0020] FIG. 6a is a schematic flowchart of another image obtaining method according to an embodiment of this application.

[0021] FIG. 6b is a schematic flowchart of still another image obtaining method according to an embodiment of this application.

[0022] FIG. 6c is a schematic flowchart of another image obtaining method according to an embodiment of this application.

[0023] FIG. 7a is a schematic diagram of an image multi-way tree according to an embodiment of this application.

[0024] FIG. 7b is a schematic diagram of an image binary tree according to an embodiment of this application.

[0025] FIG. 8 is a schematic diagram of a data interception module according to an embodiment of this application.

[0026] FIG. 9 is a schematic diagram of image synthesis processing according to an embodiment of this application.

[0027] FIG. 10a is a desktop view of an application server according to an embodiment of this application.

[0028] FIG. 10b is a schematic diagram of a user interface image according to an embodiment of this application.

[0029] FIG. 11 is a schematic structural diagram of an image obtaining apparatus according to an embodiment of this application.

[0030] FIG. 12 is a schematic structural diagram of a server according to an embodiment of this application.

DESCRIPTION OF EMBODIMENTS

[0031] The following clearly and completely describes technical solutions in embodiments of this application with reference to the accompanying drawings in the embodiments of this application.

[0032] In an implementation in which a user terminal obtains an image from a server, an architecture diagram of image obtaining may be shown in FIG. 1. In FIG. 1, application server 101 represents an application server, on which a plurality of application processes may run. Target application process 102 represents a target application process corresponding to a user identifier

among the plurality of application processes. User terminal 103 represents a user terminal that logs in to the target application by using the user identifier. Virtual machine 104 represents a virtual machine corresponding to the user terminal 103. It is to be understood that, to ensure normal running of the target application in the user terminal, the application server needs to provide service support for the running of the target application. In other words, the application server is configured to execute an application process set that provide service support for implementing an application function of the target application. The target application may be any application, such as a game application, an instant messaging application, and a shopping application.

[0033] The application server provides service support for the target application based on a service program related to the target application. Specifically, the application server may support the target application to implement the application function by executing the service program. In the field of computers, an activity of the application server running the service program can be referred to as an application process. If the target application runs on a plurality of user terminals simultaneously, to ensure that running of the target application in the user terminals does not interfere with each other, the application server creates an application process for each user to support running of the target application for the respective users. That is, the application server may execute an application process set that provides service support for implementing the application function of the target application. The application process set includes a plurality of application processes, and each application process corresponds to a user identifier. The target application process described in this embodiment of this application may refer to an application process corresponding to any user identifier.

[0034] For example, in a cloud gaming application scenario, assuming that the target application is a game application, if a user A and a user B start the game application at the same time by using different user terminals, the application server creates, upon detection of operation instructions of the user A and the user B on the game application, an application process A and an application process B respectively for the user A and the user B, to support the players of the two user terminals to experience the game application.

[0035] In the architecture diagram of image obtaining shown in FIG. 1, if the application server 101 receives an operation instruction transmitted by the user terminal 103 for the target application, the application server 101 notifies the virtual machine 104 corresponding to the user identifier in the user terminal 103 to run the target application process 102, and other virtual machines run an application process 1 and an application process 2 respectively. The application server 101 obtains a user interface image corresponding to the user identifier by running the target

application on the virtual machine 104. Optionally, a manner of obtaining the user interface image corresponding to the user identifier by using the virtual machine may be: the virtual machine 104 calls an image rendering function of the target application process 102 to perform rendering and obtain rendered window image data; the virtual machine 104 then performs synthesis processing on the window image data and other parameters to obtain an image to be displayed; the image to be displayed is displayed in the user interface of the virtual machine 104; and the application server 101 intercepts a current user interface of the virtual machine 104, uses the intercepted image as the user interface image corresponding to the user identifier, and transmits the user interface image to the user terminal 103. In this way, a user identified by the user identifier can view the user interface image corresponding to the target application process 102 by using the user terminal 103.

[0036] In an embodiment, the architecture diagram of image obtaining may be alternatively as shown in FIG. 2. In the architecture diagram of image obtaining shown in FIG. 2, application server 201 represents an application server, and an application process set that provides service support for an application function of a target application runs on the application server 201; target application process 202 represents a target application process corresponding to a user identifier in the application process set; and user terminal 203 represents a user terminal that logs in to the target application by using the user identifier, that is, a user terminal corresponding to the user identifier. In this embodiment of this application, a data interception module is further configured in each application process in the application process set. For example, a data interception module 2021 is configured in the target application process, a data interception module is configured in an application process 1, and a data interception module is configured in an application process 2.

[0037] In an embodiment, during running of the target application process 202, the application server 201 may call the data interception module 2021 in the target application process 202 to intercept window image data. The window image data are obtained when an image rendering function in the target application process 202 is executed to perform rendering. Then image synthesis processing is performed on the window image data to obtain a user interface image to be displayed. Optionally, a picture synthesizer 204 may be configured in the application server 201, and the application server 201 may transmit the window image data intercepted by calling the data interception module to the picture synthesizer 204, to call the picture synthesizer 204 to perform image synthesis processing on the window image data to obtain the user interface image to be displayed. After obtaining the user interface image, the application server 201 transmits a notification message carrying the user interface image to the user terminal 203, so that the user terminal 203 displays the user interface image on a user interface.

[0038] By comparing the architecture diagrams of image obtaining shown in FIG. 1 and FIG. 2, it is found that in FIG. 1, a corresponding virtual machine needs to be deployed for each user identifier, so that an application process corresponding to each user identifier runs on the corresponding virtual machine. When a large number of virtual machines are deployed, many resource overheads need to be consumed. In addition, because a graphics card in a virtual machine cannot support multimedia well, when the virtual machine is deployed, a high-performance graphics card needs to be independently configured in the application server 101 for the virtual machine to use, as shown by 1011 in FIG. 1. However, the high-performance graphics card 1011 is used only to support multimedia services in the virtual machine and is in an idle state in other cases, resulting in low utilization of the graphics card and waste of costs.

[0039] In addition, to implement communication among the application server 101, the virtual machine 104, and the target application process 102 in FIG. 1, a corresponding service module needs to be further deployed in the virtual machine 104, as shown by 1041 in FIG. 1. Generally, operation permission of the user to the virtual machine by using the service module 1041 is controlled by user permission. In this way, it is easy for some malicious users to break through permission control, and steal and destroy information by using the service module 1041, resulting in poor security of image obtaining.

[0040] In addition, in the architecture diagram shown in FIG. 1, each application process runs on a corresponding virtual machine instead of running on the application server, resulting in low running efficiency of the application processes.

[0041] Compared with FIG. 1, in the architecture diagram of image obtaining shown in FIG. 2, no virtual machine is deployed and resource consumption can be reduced. In addition, the application server 201 only displays the user interface image related to the target application process to the user terminal 203, which improves security of the application. Moreover, in the architecture diagram of image obtaining shown in FIG. 2, the application processes directly run on the application server 201 instead of running on the virtual machines, which improves the running efficiency of the application processes. Based on the above, an embodiment of this application based on the image obtaining architecture shown in FIG. 2 is described in detail below.

[0042] Based on the architecture diagram of image obtaining described in FIG. 2, an embodiment of this application provides an image obtaining method. The image obtaining method is applicable to an application server. The application server is configured to execute an application process set that provides service support for implementing an application function of a target application, such as the application server 101 in FIG. 1, or the application server 201 in FIG. 2. By

using the image obtaining method described in this embodiment of this application, a user interface image corresponding to a user identifier can be obtained, and the obtained user interface image is carried in a notification message and transmitted to a user terminal corresponding to the user identifier, to instruct the user terminal to display the user interface image on a user interface. In this way, a user identified by the user identifier can operate a target application by using the user interface image. Compared with the manner of obtaining a user interface image to be displayed by using a virtual machine in the related art, through this embodiment of this application, resource overheads of deploying the virtual machine in the application server are reduced.

[0043] In an embodiment, the image obtaining method described in this embodiment of this application is applicable to an application scenario of cloud gaming. The cloud gaming refers to: games corresponding to a plurality of user identifiers run on a cloud server, and the cloud server renders a game scene in each game into a game picture or screen, and transmits the game picture to a user terminal corresponding to each user identifier in real time, so that a user identified by each user identifier inputs a corresponding operation for a game picture.

[0044] When this embodiment of this application is applied to the application scenario of the cloud gaming, the application server is equivalent to the cloud server, and the user interface image is equivalent to the game picture. For example, FIG. 3a is a desktop view of a cloud server according to an embodiment of this application. For example, the cloud server is the application server 101 in FIG. 1 or the application server 201 in FIG. 2, and 300 in FIG. 3a represents a desktop of the cloud server. In FIG. 3a, a plurality of application processes that support running of a game run on the cloud server at the same time. By using the image obtaining method described in this embodiment of this application, the cloud server can obtain game pictures corresponding to the plurality of application processes. The game pictures such as game picture 301, game picture 302, and game picture 303 corresponding to each application process may be displayed in the desktop view of the cloud server at the same time. Because one application process corresponds to one user identifier, a game picture corresponding to each application process is a game picture corresponding to each user identifier. For example, game picture 301 represents a game picture corresponding to a user identifier A, game picture 302 represents a game picture corresponding to a user identifier B, and game picture 303 represents a game picture corresponding to a user identifier C.

[0045] By using the image obtaining method described in this embodiment of this application, a game picture corresponding to each user identifier can be respectively transmitted to a corresponding user terminal for display. For a user identified by a user identifier, a game picture seen in the user interface of the user terminal is a game picture corresponding to the user identifier

in the cloud server, and a game picture corresponding to another user identifier is not displayed. That is, games corresponding to users do not interfere with each other. For example, FIG. 3b is a user interface image of a user terminal corresponding to the user identifier A among the plurality of user identifiers. For example, the user terminal is the user terminal 103 in FIG. 1 and the user terminal 203 in FIG. 2. As can be seen, a user interface of the user terminal corresponding to the user identifier A only displays the game picture 301 corresponding to the user identifier A.

[0046] Based on the foregoing descriptions, an embodiment of this application provides a schematic flowchart of an image obtaining method, as shown in FIG. 4. The image obtaining method described in FIG. 4 may be performed by a server, and specifically, may be performed by a processor of the server. For example, the method is applicable to the application server 101 shown in FIG. 1 or the application server 201 shown in FIG. 2, and the application server is configured to execute an application process set that provide service support for implementing an application function of a target application. The image obtaining method described in FIG. 4 may include the following steps:

[0047] S401. Obtain, from an application process set, a target application process corresponding to a user identifier.

[0048] The user identifier is used for indicating an identity of a user, and the user identifier may include any one or more of an ID card of the user, a mobile phone number of the user, and a login account of the user.

[0049] In an embodiment, the obtaining a target application process corresponding to a user identifier may be performed after the application server receives a user operation request transmitted by a user terminal. The user terminal refers to a terminal device that is logged in based on the user identifier. The terminal device may include any device capable of decompressing a video, such as a mobile phone, a tablet, and a desktop computer. The user operation request may include any one or more of a login operation, a window switching operation, and a logout operation for a target application.

[0050] For example, if the user indicated by the user identifier logs in to the target application in the user terminal for the first time, in this case, the user operation request transmitted by the user and received by the user terminal may refer to a login operation. If the target application is logged in to the user terminal based on the user identifier, in this case, a user operation message may refer to a logout operation, or a window switching operation on each window included in the target application, or a request to run an application function of the target application, or the like.

[0051] S402. Call, when detecting that an image rendering function in the target application process is called, a data interception module to intercept a plurality of window image data currently generated.

[0052] In this step, the window image data are obtained when the image rendering function in the target application process is executed during running the target application process.

Optionally, an implementation of calling the data interception module to intercept the window image data currently generated is as shown in FIG. 5a. FIG. 5a is a schematic diagram of obtaining window image data according to an embodiment of this application. The obtaining method may be performed by a server, for example, the application server 101 in FIG. 1 or the application server 201 in FIG. 2. The method includes the following steps:

[0053] S4021: Execute the image rendering function in the target application process.

[0054] S4022. Intercept the image rendering function when detecting that the image rendering function in the target application process is called.

[0055] S4023: Call the data interception module to obtain a window image data generated for each window by using the image rendering function.

[0056] S4024: The execution of the image rendering function ends.

[0057] Image rendering refers to a process of transferring, processing and converting three-dimensional light energy transfer process into a two-dimensional image, and the image rendering function refers to a function that can implement a function of image rendering.

[0058] Commonly used image rendering manners in the image rendering function may include: Open Graphics Library (OpenGL), Graphics Device Interface (GDI), and programming interface DirectX. OpenGL is a bottom 3D graphics library that only provides a rendering function and may be ported between different platforms; DirectX mainly provides hardware support for GPU designs and is more suitable for image rendering of game applications; and GDI is designed for broad spectrum and may be also applied to image rendering of game applications, but an effect thereof is not as good as that of DirectX.

[0059] In an embodiment, when rendering is performed for different windows, a rendering manner for each window in the image rendering function of the target application process may be different. For example, when a first window of the target application is rendered, the rendering manner included in the image rendering function may be GDI; and when a second window of the target application is rendered, the rendering manner included in the image rendering function may be OpenGL.

[0060] In an embodiment, FIG. 5b to FIG. 5d are schematic diagrams of window image data according to an embodiment of this application. Assuming that the target application is a game application, the game application may include three windows, namely, a "homepage" window, a "store" window, and a "video" window. Each window corresponds to a window image data. Specifically, a "homepage" window image data is shown in FIG. 5b, a "store" window image data is shown in FIG. 5c, and a "video" window image data is shown in FIG. 5d.

[0061] S403. Perform image synthesis processing on the plurality of window image data, to obtain a user interface image to be displayed.

[0062] In an embodiment, the image synthesis processing may refer to sequentially superimposing the plurality of window image data according to a synthesis order of the plurality of window image data. The synthesis order may refer to an order in which the plurality of window image data are processed during the image synthesis processing, and a window image data with an earlier synthesis order is processed sooner during the image synthesis processing. The synthesis order may be determined according to window hierarchy identifiers corresponding to the plurality of window image data. The window hierarchy identifiers are each used for indicating a hierarchy of a window. A higher hierarchy indicated by the window hierarchy identifier indicates a later synthesis order of the window image data.

[0063] For example, assume that in FIG. 5b to FIG. 5d, a descending order of window hierarchy identifiers of the three window image data are sequentially "video", "store", and "homepage". According to the foregoing description that a window image data with a higher window hierarchy identifier has a later synthesis order, the synthesis order of the three window image data in FIG. 5b to FIG. 5d may be "homepage", "store", and "video". If synthesis processing is performed on the three window image data according to the synthesis order, a user interface image to be displayed obtained may be as shown in FIG. 3b. Optionally, the application server may call a picture synthesizer to perform the image synthesis processing on the plurality of window image data.

[0064] S404. Transmit a notification message carrying the user interface image to a user terminal corresponding to the user identifier, so that the user terminal displays the user interface image on a user interface.

[0065] The notification message is used for instructing the user terminal to display the user interface image on the user interface. A schematic diagram of the user terminal displaying the user interface image on the user interface may be shown in FIG. 3b.

[0066] Compared with the manner of obtaining a user interface image by deploying a virtual machine in the related art, in this embodiment of this application, the application server calls a rendering function to obtain window image data and performs image synthesis processing on the window image data to obtain a user interface image in a process of intercepting and executing the target application, which can reduce resource overheads of the application server.

[0067] FIG. 6a is a schematic flowchart of another image obtaining method according to an embodiment of this application. The image obtaining method described in FIG. 6a may be performed by a server, and specifically, may be performed by a processor of the server. For example, the method is applicable to the application server 101 shown in FIG. 1 or the application server 201 shown in FIG. 2, and the application server is configured to execute an application process set that provide service support for implementing an application function of a target application. The image obtaining method described in FIG. 6a may include the following steps:

[0068] S601. Obtain, from an application process set, a target application process corresponding to a user identifier.

[0069] This step is the same as step 401, and details are not described herein again.

[0070] S602. Determine, when detecting that an image rendering function in the target application process is called, a rendering manner included in the image rendering function.

[0071] There are a plurality of window image data. The plurality of window image data are obtained by executing one or more image rendering functions of the target application. The rendering manner included in each image rendering function may be any of OpenGL, GDI and DirectX.

[0072] S603. Determine a data interception submodule corresponding to the rendering manner in a data interception module according to the rendering manner.

[0073] The data interception module may include three interception submodules, and the three interception submodules respectively correspond to the foregoing three rendering manners. Specifically, the three interception submodules included in the data interception module may be an OpenGL interception submodule, a GDI interception submodule, and a DirectX interception submodule. Optionally, correspondences between the interception submodules and the rendering manners may be: the rendering manner corresponding to the OpenGL interception submodule is OpenGL, the rendering manner corresponding to the GDI interception submodule is GDI, and the rendering manner corresponding to the DirectX interception submodule is DirectX. It is to be

understood that calling an interception submodule corresponding to a rendering manner to obtain a window image data can ensure the accuracy of the intercepted window image data.

[0074] In an embodiment, a schematic diagram of the data interception module provided in this embodiment of this application may be shown in FIG. 8. The data interception module shown in FIG. 8 may include the GDI interception submodule, the DirectX interception submodule, and the OpenGL interception submodule.

[0075] S604. Call the data interception submodule to intercept a plurality of window image data obtained when the image rendering function is executed.

[0076] While performing steps S602 to S604, step S605 in the following may be further performed:

[0077] S605. Obtain a picture display parameter configured for the target application process, and determine a window hierarchy identifier of each window image data according to the picture display parameter.

[0078] The picture display parameter may include any one or more of whether a window corresponding to each window image data is minimized or maximized, and an arrangement relationship between the windows. For example, the arrangement relationship between the windows may be that a window A is placed before a window B, a window C is placed behind the window B, and the like.

[0079] In an embodiment, the picture display parameter may be stored in a sandbox. The sandbox described in this embodiment of this application is a mechanism that can isolate a plurality of application processes running on the application server from each other.

[0080] In an embodiment, the picture display parameter configured for the target application process and stored in the sandbox may be a parameter set by default for the target application process by the application server. For example, assuming that the target application is a game application, a parameter set by the application server for the target application process configured to provide support for implementing a game function of the game application may include: a window corresponding to a "homepage" window image data is placed at the front-most position for display, a window corresponding to a "store" window image data is placed behind the window corresponding to the "homepage" window image data, and a window corresponding to a "video" window image data is placed behind the window corresponding to the "store" window image data. In this case, the picture display parameter may be: the "homepage" window image data is placed at the front-most position for display, the "store" window image data is placed behind the "homepage"

window image data, and the "video" window image data is placed behind the "store" window image data.

[0081] In another embodiment, the picture display parameter configured for the target application process and stored in the sandbox may be determined according to a user operation request transmitted by a user terminal and parameters set by default for the target application process by the application server. For example, assuming that the target application is a game application, the parameter configured for the target application process configured to provide service support for implementing a game function of the game application may include: the window corresponding to the "homepage" window image data is placed at the front-most position for display, the window corresponding to the "store" window image data is placed behind the window corresponding to the "homepage" window image data, and the window corresponding to the "video" window image data is placed behind the window corresponding to the "store" window image data. It is detected that the user operation request includes switching from the window corresponding to the "homepage" window image data to the window corresponding to the "video" window image data. In this case, the picture display parameter determined according to the user operation request and the parameters set by default for the application by the application server may include: the window corresponding to the "video" window image data is placed at the front-most position for display, the window corresponding to the "store" window image data is placed behind the window corresponding to the "homepage" window image data, and the window corresponding to the "homepage" window image data is placed behind the window corresponding to the "video" window image data.

[0082] After the picture display parameter configured for the target application process is obtained, the window hierarchy identifiers of the plurality of window image data may be determined according to the picture display parameter. The window hierarchy identifiers are each used for uniquely representing corresponding window image data, and the window hierarchy identifiers indicate a hierarchical relationship between window image data.

[0083] The window hierarchy identifiers may be each represented by one or more numbers, such as 1, 2, 3, or (1, 2), (2, 1), (3, 1); or the window hierarchy identifiers may be alternatively represented in other forms. In this embodiment of this application, when the window hierarchy identifiers are represented by using numbers: If quantities of numbers used to represent the window hierarchy identifiers are different, a greater quantity of numbers indicates a higher window hierarchy identifier. For example, a window hierarchy identifier represented by (1, 2) is higher than a window hierarchy identifier represented by 1. If quantities of numbers used to represent the

window hierarchy identifier are all N , N is any positive integer greater than 1, the first $(N-1)$ numbers are identical, and the N^{th} number is different, a smaller sum of numbers indicates a higher window hierarchy identifier. For example, the window hierarchy identifier represented by (1, 2) is higher than a window hierarchy identifier represented by (1, 5). A window corresponding to a window hierarchy identifier (1, n) (n being any positive integer greater than or equal to 1) is a child window of a window corresponding to a window hierarchy identifier (1); and a window corresponding to a window hierarchy identifier (1, n , m) is a child window of the window corresponding to the window hierarchy identifier (1, n). It can be learned based on the foregoing description that a window hierarchy identifier corresponding to a child window is higher than a window hierarchy identifier corresponding to a parent window of the child window.

[0084] S606. Perform image synthesis processing on the plurality of window image data according to the window hierarchy identifiers, to obtain a user interface image to be displayed.

[0085] Because the window hierarchy identifiers represent the hierarchical relationship between the windows, image synthesis processing is performed on the plurality of window image data according to an order of the hierarchical relationship from high to low.

[0086] S607. Transmit a notification message carrying the user interface image to a user terminal corresponding to the user identifier, so that the user terminal displays the user interface image on a user interface.

[0087] This step is the same as step S404, and details are not described herein again.

[0088] FIG. 6b is a schematic flowchart of still another image obtaining method according to an embodiment of this application. The image obtaining method described in FIG. 6b may be performed by a server, and specifically, may be performed by a processor of the server. For example, the method is applicable to the application server 101 shown in FIG. 1 or the application server 201 shown in FIG. 2, and the application server is configured to execute an application process set that provide service support for implementing an application function of a target application. The image obtaining method described in FIG. 6b may include the following steps:

[0089] S701. Obtain, from an application process set, a target application process corresponding to a user identifier.

[0090] This step is the same as step 401, and details are not described herein again.

[0091] S702. Obtain a picture display parameter configured for the target application process, and determine a window hierarchy identifier of each window image data according to the picture display parameter.

[0092] This step is the same as step 605, and details are not described herein again.

[0093] S703. Generate an image binary tree according to the window hierarchy identifiers of the plurality of window image data.

[0094] To improve convenience, in this embodiment of this application, an image binary tree is generated according to the window hierarchy identifiers of the plurality of window image data. The image binary tree may include a plurality of binary tree nodes, and each binary tree node stores a window hierarchy identifier.

[0095] In an embodiment, the generating an image binary tree according to the window hierarchy identifiers of the plurality of window image data may include:

[0096] S7031. Construct an image multi-way tree according to the window hierarchy identifiers of the plurality of window image data.

[0097] The image multi-way tree includes N hierarchies, and each hierarchy includes M nodes, where N and M are both positive integers greater than or equal to 1. Two adjacent hierarchies in the image multi-way tree include a parent node and a child node sharing a parent-child relationship. Nodes having the same parent node in the same hierarchy of the image multi-way tree are sibling nodes.

[0098] S7032. Convert the image multi-way tree into the image binary tree.

[0099] A first node in each hierarchy of the image multi-way tree is converted into a left node of the image binary tree, and an other node that is a sibling node of the first node is converted into a right node of the image binary tree.

[0100] For example, it is assumed that the image multi-way tree constructed according to the window hierarchy identifiers of the plurality of window image data is shown in FIG. 7a. It is assumed that the image multi-way tree includes three hierarchies. As shown in hierarchies 701-703, numbers in a node of each hierarchy represent a window hierarchy identifier of a window image data corresponding to the node. It can be seen from the figure that the hierarchy 701 includes one node, and the node included in the hierarchy 701 may be referred to as a root node. In this embodiment of this application, the root node is desktop image data of the user terminal. The desktop image data of the user terminal may be transmitted by the application server after receiving a selection operation of the desktop image data entered by the user and received by the user terminal. The hierarchy 702 includes three nodes, and the hierarchy 703 includes three nodes.

[0101] Windows corresponding to window hierarchy identifiers (1, 1) and (1, 2) in the hierarchy 702 are child windows of a window corresponding to a window hierarchy identifier 1.

Therefore, a parent node of child nodes represented by the window hierarchy identifiers (1, 1) and (1, 2) is a node represented by the window hierarchy identifier 1, and the nodes represented by the window hierarchy identifiers (1, 1) and (1, 2) are sibling nodes. Similarly, nodes represented by window hierarchy identifiers 1, 2, and 3 are sibling nodes.

[0102] Converting the image multi-way tree shown in FIG. 7a into an image binary tree refers to: converting a first node in each of the hierarchies 701-703 in the image multi-way tree into a left node of the image binary tree. For example, the first node in the hierarchy 701 is a node represented by a window hierarchy identifier 0, and this node is used as a left node of the image binary tree. No other nodes are included in the hierarchy 701, and the hierarchy 702 is continuously converted. In the hierarchy 702, the first node is a node represented by the window hierarchy identifier 1, and this node is still used as a left node of the image binary tree. The hierarchy 702 further includes two sibling nodes of the node represented by the window hierarchy identifier 1, namely, a node represented by a window hierarchy identifier 2 and a node represented by a window hierarchy identifier 3, and the two nodes are successively used as right nodes of the image binary tree. All nodes in the image multi-way tree are traversed according to the foregoing process, and the image binary tree as shown in FIG. 7b can be obtained after all the nodes are traversed.

[0103] It is to be understood that successively using the node represented by the window hierarchy identifier 2 and the node represented by the window hierarchy identifier 3 as the right nodes of the image binary tree may be understood as: the node represented by the window hierarchy identifier 2 is used as the right node of the node represented by the window hierarchy identifier 1, and the node represented by the window hierarchy identifier 3 is used as the right node of the node represented by the window hierarchy identifier 2.

[0104] S704. Traverse the image binary tree according to a preset first traversing order, to obtain a synthesis order of the plurality of window image data.

[0105] The preset first traversing order (e.g., binary tree image synthesis) may refer to pre-order traversing, and the pre-order traversing refers to: root node-left node-right node. For example, a result of the pre-order traversing in the image binary tree shown in FIG. 7b is: 0-1-(1, 1)-(1, 2)-2-3-(3, 1). The image binary tree is traversed according to a preset traversing order of binary tree display, to obtain a synthesis order of the plurality of window image data: 0-1-(1, 1)-(1, 2)-2-3-(3, 1).

[0106] S705. Traverse the image binary tree according to a preset second traversing order, to obtain an image capturing order of the plurality of window image data, and call the data

interception module to obtain, through interception, the plurality of window image data according to the image capturing order.

[0107] In an embodiment, the preset second traversing order (e.g., binary image obtaining) may refer to post-order traversing, and a traversing order of the post-order traversing is: left node-right node-root node. For example, a result of the post-order traversing in the image binary tree shown in FIG. 7b is: (1, 1)-(1, 2)-(3, 1)-3-2-1-0. The image binary tree is traversed according to a preset traversing order of binary tree obtaining, to obtain an image capturing order of the plurality of window image data: (1, 1)-(1, 2)-(3, 1)-3-2-1-0.

[0108] In an embodiment, after window image data are obtained, image synthesis processing may be performed on the window image data, to obtain a user interface image to be displayed.

[0109] S706. Perform image synthesis processing on the plurality of window image data according to the synthesis order, to obtain a user interface image to be displayed.

[0110] In an embodiment, an implementation of step 706 may be: allocating a sub-canvas for each window image data, placing the corresponding window image data on a corresponding sub-canvas, and combining the sub-canvas according to the foregoing synthesis order, to obtain the user interface image to be displayed.

[0111] It is to be understood that the combination is performed according to the foregoing synthesis order, so that a window image data with a higher window hierarchy identifier is combined later. In this way, in the obtained user interface image, a window image data with a higher window hierarchy identifier covers a window image data with a lower window hierarchy identifier.

[0112] S707. Transmit a notification message carrying the user interface image to a user terminal corresponding to the user identifier.

[0113] This step is the same as step 404, and details are not described herein again.

[0114] In another embodiment of this application, FIG. 6c is a schematic flowchart of another image obtaining method according to an embodiment of this application. The image obtaining method may be performed by a server, and specifically, may be performed by a processor of the server. For example, the method is applicable to the application server 101 shown in FIG. 1 or the application server 201 shown in FIG. 2. As shown in FIG. 6c, on the basis of the embodiment described in FIG. 6b, after step S702, the following may be further performed:

[0115] S708. Generate a message response binary tree according to the window hierarchy identifiers of the plurality of window image data.

[0116] The message response binary tree includes a plurality of message nodes, and each message node records window location information and window size information of a corresponding window image data. For brevity of operation, when the message response binary tree is generated according to the window hierarchy identifiers of the plurality of window image data, the same method as step S703 may be used. In this way, the message response binary tree and the image binary tree have the same structure. For example, the message response binary tree generated according to the window hierarchy identifiers of the plurality of window image data may likewise be as shown in FIG. 7b.

[0117] Each node of the image binary tree stores a window hierarchy identifier of a window image data. Unlike the image binary tree, each node in the message response binary tree not only stores a window hierarchy identifier of each window image data, but also stores window location information and window size information corresponding to the window image data. The window location information may include coordinates of a window, and the coordinates of the window may be represented in the form of pixel coordinates such as (3 pixels, 6 pixels), or may be represented in the form of world coordinates such as (10, 15). The window size information may include information such as a width and a height of the window.

[0118] S709. Determine display location information of each window image data based on the message response binary tree.

[0119] The display location information may include any one or both of the window location information and window size information of the window image data.

[0120] S710. Perform the image synthesis processing on the plurality of window image data according to the display location information and the synthesis order, to obtain a user interface image to be displayed.

[0121] When the plurality of window image data is synthesized, after the plurality of window image data are learned, where the plurality of window image data is located further needs to be learned. For example, as shown in FIG. 5b to FIG. 5d, after the "homepage" window image data and the "store" window image data are learned, a size with which and a location at which the two window image data are combined further need to be learned, to obtain the user interface image shown in FIG. 3b. Therefore, when the plurality of window image data are synthesized, the display location information of the plurality of window image data may be obtained by using the message response binary tree.

[0122] Based on the foregoing description, an embodiment of this application provides a schematic diagram of image synthesis processing as shown in FIG. 9. In FIG. 9, a data interception module is configured to obtain a plurality of window image data in a target application process. An application server calls a picture synthesizer to perform image synthesis processing on the plurality of window image data according to an image binary tree and a message response binary tree.

[0123] In an embodiment, in step S709, an implementation of determining display location information of each window image data based on the message response binary tree may include:

[0124] For a first window image data in the plurality of window image data (e.g., the first window image data being any window image data in the plurality of window image data), when it is detected that image synthesis processing is currently performed on the first window image data, the application server may obtain a window hierarchy identifier of the first window image data according to the image binary tree; window location information and window size information corresponding to the window hierarchy identifier of the first window image data are searched from the message response binary tree; and the found window location information and window size information are determined as display location information of the first window image data.

[0125] In another embodiment, in addition to being applied to image synthesis processing, the message response binary tree may be further used to search a target window image data corresponding to a user operation message, so that the application server can process the target window image data in response to the user operation message. The user operation message is an operation related to the user interface image and entered by the user by using the user terminal. Specifically, FIG. 6c further includes the following steps:

[0126] S711. Receive a user operation message transmitted by the user terminal, and obtain operation location information from the user operation message.

[0127] The operation location information included in the user operation message refers to location information of a location at which the user inputs the operation. For example, the user taps, double-taps, or presses and holds a location in the user interface image, and location information of the location is determined as the operation location information.

[0128] S712. Traverse the message response binary tree according to a preset third traversing order, and determine a message node of the plurality of message nodes at which the operation location information meets a preset location condition as a target message node.

[0129] Specifically, the target message node is the first message node found in the message response binary tree that meets the preset location condition. The operation location information

meets a preset location condition when the operation location information is within a range defined by window location information and window size information recorded by the target message node.

[0130] S713. Process, by using a window image data corresponding to the target message node as a target window image data, a window corresponding to the target window image data in response to the user operation message.

[0131] Because application processes corresponding to a plurality of user identifiers run in the application server, user interface images corresponding to the plurality of user identifiers are displayed in the application server, and the user interface images corresponding to different user identifiers may overlap, as shown in FIG. 10a. In FIG. 10a, A and B represent user interface images corresponding to different user identifiers. A plurality of window image data in a user interface image corresponding to the same user identifier may also overlap, as shown in FIG. 10b. In FIG. 10b, A and B represent two window image data in the user interface image corresponding to the same user identifier.

[0132] Different user identifiers correspond to different application processes, and different application processes correspond to different picture display parameters. Because an image binary tree and a message response binary tree are generated according to a window hierarchy identifier determined by a picture display parameter, image binary trees and message response binary trees of different application processes are also different. Therefore, it may be understood that message response binary trees corresponding to different user identifiers are different.

[0133] For the case of FIG. 10a, if a location indicated by the operation location information included in the user operation message is located in an overlapping area of A and B, the application server needs to determine a user interface image which the user operation message entered by the user is for. Optionally, which user interface image the currently received user operation message is for may be determined according to message response binary trees corresponding to different user identifiers.

[0134] For the case of FIG. 10b, the application server needs to determine which window image data in the user interface image the user operation message is for. Optionally, the message response binary tree may be traversed according to a preset third traversing order (e.g., binary tree window search), to determine the target message node, and further determine the target window image data corresponding to the target message node.

[0135] The preset third traversing order may refer to middle-order traversing, and the middle-order traversing refers to traversing a binary tree in an order of: left node-root node-right

node. For example, assuming that the message response binary tree is shown in FIG. 7b, a result of the middle-order traversing is: (1, 1)-(1, 2)-1-2-(3, 1)-(3)-0.

[0136] Based on the description of the foregoing image obtaining method, an embodiment of this application further discloses an image obtaining apparatus. The image obtaining apparatus is applicable to a server, for example, the application server 101 shown in FIG. 1 or the application server 201 shown in FIG. 2. The application server is configured to execute an application process set that provide service support for implementing an application function of a target application. The image obtaining apparatus can perform the methods shown in FIG. 4, and FIG. 6a to FIG. 6c. Referring to FIG. 11, an image obtaining apparatus 1100 may operate the following units:

an obtaining unit 1101, configured to obtain, from an application process set, a target application process corresponding to a user identifier;

an interception unit 1102, configured to call, when detecting that an image rendering function in the target application process obtained by using the obtaining unit 1101 is called, a data interception module to intercept a plurality of window image data currently generated;

a first processing unit 1103, configured to perform image synthesis processing on the plurality of window image data intercepted by using the interception unit 1102, to obtain a user interface image to be displayed; and

a transmission unit 1104, configured to transmit a notification message carrying the user interface image obtained by using the first processing unit 1103 to a user terminal corresponding to the user identifier, so that the user terminal displays the user interface image on a user interface.

[0137] In an embodiment, the obtaining unit 1101 is further configured to obtain a picture display parameter configured for the target application process.

[0138] The apparatus 1100 further includes:

a determining unit 1105, configured to determine a window hierarchy identifier of each window image data according to the picture display parameter obtained by using the obtaining unit 1102, where

the first processing unit 1103 is further configured to perform the image synthesis processing on the plurality of window image data according to the window hierarchy identifiers determined by using the determining unit 1105.

[0139] In an embodiment, the apparatus 1100 further includes:

a first generation unit 1106, configured to generate an image binary tree according to the window hierarchy identifiers of the plurality of window image data determined by using the determining unit 1105, where

the first processing unit 1103 is further configured to traverse the image binary tree generated by using the first generation unit 1106 according to a preset first traversing order, to obtain a synthesis order of the plurality of window image data; and perform the image synthesis processing on the plurality of window image data according to the synthesis order.

[0140] In an embodiment, the first generation unit 1106 is further configured to construct an image multi-way tree according to the window hierarchy identifiers of the plurality of window image data determined by using the determining unit 1105; and convert the image multi-way tree into the image binary tree.

[0141] In an embodiment, the apparatus 1100 further includes:

a first generation unit 1106, configured to generate an image binary tree according to the window hierarchy identifiers of the plurality of window image data determined by using the determining unit 1105, where

the interception unit 1102 is configured to traverse the image binary tree generated by using the first generation unit 1106 according to a preset second traversing order, to obtain an image capturing order of the plurality of window image data; and call the data interception module to obtain, through interception, the plurality of window image data according to the image capturing order.

[0142] In an embodiment, the interception unit 1102 is configured to determine a rendering manner included in the image rendering function; determine a data interception submodule corresponding to the rendering manner in the data interception module according to the rendering manner; and call the data interception submodule to intercept the plurality of window image data obtained when the image rendering function is executed.

[0143] In an embodiment, the apparatus 1100 further includes:

a second generation unit 1107, configured to generate a message response binary tree according to the window hierarchy identifiers of the plurality of window image data determined by using the determining unit 1105, where

the first processing unit 1103 is configured to determine display location information of each window image data based on the message response binary tree generated by using the

second generation unit 1107; and perform the image synthesis processing on the plurality of window image data according to the display location information and the synthesis order.

[0144] In an embodiment, the message response binary tree includes a plurality of message nodes, and the obtaining unit 1101 is further configured to receive a user operation message transmitted by the user terminal, and obtain operation location information from the user operation message.

[0145] The apparatus 1100 further includes:

a second processing unit 1108, configured to traverse the message response binary tree generated by using the second generation unit 1107 according to a preset third traversing order, and determine a message node of the plurality of message nodes at which the operation location information obtained by using the obtaining unit 1101 meets a preset location condition as a target message node; and process, by using a window image data corresponding to the target message node as a target window image data, a window corresponding to the target window image data in response to the user operation message.

[0146] According to another embodiment of this application, the units of the image obtaining apparatus shown in FIG. 11 may be separately or wholly combined into one or several other units, or one (or more) of the units herein may further be divided into a plurality of units of smaller functions. In this way, same operations may be implemented, and the implementation of the technical effects of the embodiments of this application is not affected. The foregoing units are divided based on logical functions. In an actual application, a function of one unit may also be implemented by a plurality of units, or functions of a plurality of units are implemented by one unit. In other embodiments of this application, the image obtaining apparatus may also include other units. In an actual application, the functions may also be cooperatively implemented by other units and may be cooperatively implemented by a plurality of units.

[0147] According to another embodiment of this application, a computer program (including program code) that can perform the steps in the corresponding method shown in FIG. 4, or FIG. 6a to FIG. 6c may be run on a general computing device, such as a computer, which includes processing elements and storage elements such as a central processing unit (CPU), a random access memory (RAM), and a read-only memory (ROM), to construct the image obtaining apparatus shown in FIG. 11 and implement the image obtaining method in the embodiments of this application. The computer program may be recorded on, for example, a computer-readable recording medium such as a non-transitory computer-readable storage medium, and may be loaded

into the foregoing computing device by using the computer-readable recording medium and run on the computing device.

[0148] Based on the descriptions of the foregoing method embodiments and apparatus embodiments, an embodiment of this application further provides a server. The server may be the application server 101 shown in FIG. 1 or the application server 201 shown in FIG. 2, and the server is configured to execute an application process set that provide service support for implementing an application function of a target application. Referring to FIG. 12, the server includes at least a processor 1201, a network interface 1202, and a computer storage medium 1203. The network interface 1202 is configured to receive or transmit data when connected to a network.

[0149] The computer storage medium 1203 may be stored in a memory of the server. The computer storage medium 1203 is configured to store a computer program. The computer program includes program instructions. The processor 1201 is configured to execute the program instructions stored in the computer storage medium 1203. The processor 1201, or referred to as a central processing unit (CPU), is a computing core and a control core of a terminal, is suitable for implementing one or more instructions, and is specifically suitable for loading and executing one or more instructions to implement a corresponding method procedure or a corresponding function. In an embodiment, the processor 1201 described in this embodiment of this application may be configured to perform the foregoing image obtaining method described in FIG. 4, and FIG. 6a to FIG. 6c.

[0150] An embodiment of this application further provides a computer storage medium such as a non-transitory computer-readable storage medium. The computer storage medium is a memory device in a terminal and is configured to store programs and data. As can be understood, the computer storage medium herein may include an internal storage medium of the terminal and may further include an extended storage medium supported by the terminal. The computer storage medium provides storage space, and the storage space stores an operating system of the terminal. In addition, the storage space further stores one or more instructions suitable to be loaded and executed by the processor 1201. The instructions may be one or more computer programs (including program code). The computer storage medium herein may be a high-speed RAM or a non-volatile memory, for example, at least one magnetic disk memory. Optionally, the computer storage medium may further be at least one computer storage medium located away from the foregoing processor.

[0151] In an embodiment, the one or more instructions stored in the computer storage medium may be loaded and executed by the processor 1201, to implement corresponding steps of the method in the foregoing embodiments related to the image obtaining method. In a specific

implementation, the one or more instructions in the computer storage medium are loaded by the processor 1201 to perform the foregoing image obtaining method in FIG. 4, and FIG. 6a to FIG. 6c.

[0152] What is disclosed above is merely exemplary embodiments of this application, and certainly is not intended to limit the scope of the claims of this application. Therefore, equivalent variations made in accordance with the claims of this application shall fall within the scope of this application.

CLAIMS

What is claimed is:

1. An image obtaining method, performed by a server, the method comprising:

obtaining, from an application process set, a target application process corresponding to a user identifier;

calling, when detecting that an image rendering function in the target application process is called, a data interception module to intercept a plurality of window image data currently generated;

performing image synthesis processing on the plurality of window image data, to obtain a user interface image to be displayed; and

transmitting a notification message carrying the user interface image to a user terminal corresponding to the user identifier, so that the user terminal displays the user interface image on a user interface.

2. The method according to claim 1, further comprising:

obtaining a picture display parameter configured for the target application process; and

determining a window hierarchy identifier of each window image data according to the picture display parameter, wherein

the performing image synthesis processing on the plurality of window image data, to obtain a user interface image to be displayed comprises:

performing the image synthesis processing on the plurality of window image data according to the window hierarchy identifiers.

3. The method according to claim 2, further comprising:

generating an image binary tree according to the window hierarchy identifiers of the plurality of window image data, wherein

the performing the image synthesis processing on the plurality of window image data according to the window hierarchy identifiers comprises:

traversing the image binary tree according to a preset first traversing order, to obtain a synthesis order of the plurality of window image data; and

performing the image synthesis processing on the plurality of window image data according to the synthesis order.

4. The method according to claim 3, wherein the generating an image binary tree according to the window hierarchy identifiers of the plurality of window image data comprises:

constructing an image multi-way tree according to the window hierarchy identifiers of the plurality of window image data; and

converting the image multi-way tree into the image binary tree.

5. The method according to claim 2, further comprising:

generating an image binary tree according to the window hierarchy identifiers of the plurality of window image data, wherein

the calling, when detecting that an image rendering function in the target application process is called, a data interception module to intercept a plurality of window image data currently generated comprises:

traversing the image binary tree according to a preset second traversing order, to obtain an image capturing order of the plurality of window image data; and

calling the data interception module to obtain, through interception, the plurality of window image data according to the image capturing order.

6. The method according to claim 1, wherein the calling, when detecting that an image rendering function in the target application process is called, a data interception module to intercept a plurality of window image data currently generated comprises:

determining a rendering manner comprised in the image rendering function;

determining a data interception submodule corresponding to the rendering manner in the data interception module according to the rendering manner; and

calling the data interception submodule to intercept the plurality of window image data obtained when the image rendering function is executed.

7. The method according to claim 3, further comprising:

generating a message response binary tree according to the window hierarchy identifiers of the plurality of window image data, wherein

the performing the image synthesis processing on the plurality of window image data according to the synthesis order comprises:

determining display location information of each window image data based on the message response binary tree; and

performing the image synthesis processing on the plurality of window image data according to the display location information and the synthesis order.

8. The method according to claim 7, wherein the message response binary tree comprises a plurality of message nodes, and the method further comprises:

receiving a user operation message transmitted by the user terminal, and obtaining operation location information from the user operation message;

traversing the message response binary tree according to a preset third traversing order, and determining a message node of the plurality of message nodes at which the operation location information meets a preset location condition as a target message node; and

processing, by using window image data corresponding to the target message node as target window image data, a window corresponding to the target window image data in response to the user operation message.

9. The method according to claim 8, wherein each message node records window location information and window size information of a corresponding window image data; and

the operation location information meets the preset location condition when the operation location information is within a range defined by the window location information and the window size information recorded by the target message node.

10. An image obtaining apparatus, configured in a server, the apparatus comprising:

an obtaining unit, configured to obtain, from an application process set, a target application process corresponding to a user identifier;

an interception unit, configured to call, when detecting that an image rendering function in the target application process obtained by using the obtaining unit is called, a data interception module to intercept a plurality of window image data currently generated;

a first processing unit, configured to perform image synthesis processing on the plurality of window image data intercepted by using the interception unit, to obtain a user interface image to be displayed; and

a transmission unit, configured to transmit a notification message carrying the user interface image obtained by using the first processing unit to a user terminal corresponding to the user identifier, so that the user terminal displays the user interface image on a user interface.

11. The apparatus according to claim 10, wherein the obtaining unit is further configured to obtain a picture display parameter configured for the target application process; and

the apparatus further comprises:

a determining unit, configured to determine a window hierarchy identifier of each window image data according to the picture display parameter obtained by using the obtaining unit, wherein

the first processing unit is further configured to perform the image synthesis processing on the plurality of window image data according to the window hierarchy identifiers determined by using the determining unit.

12. The apparatus according to claim 11, further comprising:

a first generation unit, configured to generate an image binary tree according to the window hierarchy identifiers of the plurality of window image data determined by using the determining unit, wherein

the first processing unit is further configured to traverse the image binary tree generated by using the first generation unit according to a preset first traversing order, to obtain a synthesis order of the plurality of window image data; and perform the image synthesis processing on the plurality of window image data according to the synthesis order.

13. The apparatus according to claim 12, wherein the first generation unit is further configured to construct an image multi-way tree according to the window hierarchy identifiers of the plurality of window image data determined by using the determining unit; and convert the image multi-way tree into the image binary tree.

14. The apparatus according to claim 11, further comprising:

a first generation unit, configured to generate an image binary tree according to the window hierarchy identifiers of the plurality of window image data determined by using the determining unit, wherein

the interception unit is configured to traverse the image binary tree generated by using the first generation unit according to a preset second traversing order, to obtain an image capturing order of the plurality of window image data; and call the data interception module to obtain, through interception, the plurality of window image data according to the image capturing order.

15. The apparatus according to claim 10, wherein the interception unit is configured to determine a rendering manner comprised in the image rendering function; determine a data interception submodule corresponding to the rendering manner in the data interception module according to the rendering manner; and call the data interception submodule to intercept the plurality of window image data obtained when the image rendering function is executed.

16. The apparatus according to claim 12, further comprising:

a second generation unit, configured to generate a message response binary tree according to the window hierarchy identifiers of the plurality of window image data determined by using the determining unit, wherein

the first processing unit is configured to determine display location information of each window image data based on the message response binary tree generated by using the second generation unit; and perform the image synthesis processing on the plurality of window image data according to the display location information and the synthesis order.

17. The apparatus according to claim 16, wherein the message response binary tree comprises a plurality of message nodes, and the obtaining unit is further configured to receive a user operation message transmitted by the user terminal, and obtain operation location information from the user operation message; and

the apparatus further comprises:

a second processing unit, configured to traverse the message response binary tree generated by using the second generation unit according to a preset third traversing order, and determine a message node of the plurality of message nodes at which the operation location information obtained by using the obtaining unit meets a preset location condition as a target message node; and process, by using window image data corresponding to the target message node as target window image data, a window corresponding to the target window image data in response to the user operation message.

18. A server, comprising:

a processor, suitable for implementing one or more instructions; and

a computer storage medium, storing one or more instructions, the one or more instructions being suitable to be loaded by the processor to perform the image obtaining method according to any one of claims 1-9.

19. A computer storage medium, storing computer program instructions, the computer program instructions, when executed by a processor, being configured to perform the image obtaining method according to any one of claims 1-9.