# MULTIMEDIA PLAYING METHOD AND SYSTEM FOR CLOUD DESKTOP, DEVICE AND STORAGE MEDIUM

**[0001]** The present application claims priority to the Chinese Patent Application No. 202310041638.9, entitled "MULTIMEDIA PLAYING METHOD AND SYSTEM FOR CLOUD DESKTOP, DEVICE AND STORAGE MEDIUM" and filed with the China National Intellectual Property Administration on January 11, 2023, which is hereby incorporated by reference in its entirety.

## TECHNICAL FIELD

**[0002]** The present application relates to the field of communication technologies and, in particular, to a multimedia playing method and system for a cloud desktop, a device and a storage medium.

## BACKGROUND

**[0003]** With the popularization of cloud desktop products and the continuous enrichment of multimedia resources, the demand for multimedia playing within cloud desktops is growing increasingly substantial.

**[0004]** Currently, multimedia playing solutions for cloud desktops essentially follows this approach: a multimedia player of a cloud desktop server performs multimedia playing, captures a cloud desktop as an image during the multimedia playing process within the cloud desktop, compresses and encodes the captured image, and then transmits it to a cloud desktop terminal for display. During the multimedia playing process within the cloud desktops, multimedia decoding and other playing-related tasks are entirely performed in cloud desktop servers. This consumes a significant amount of central processing unit (CPU), graphics processing unit (GPU) and network resources of the cloud desktop servers, imposing a heavy load on cloud desktop

servers.

## SUMMARY

**[0005]** Multiple aspects of the present application provide a multimedia playing method and system for a cloud desktop, a device and a storage medium, which are intended to reduce resource occupation of a cloud desktop server caused by multimedia playing.

**[0006]** An embodiment of the present application provides a multimedia playing system for a cloud desktop, including a cloud desktop terminal and a cloud desktop server, the cloud desktop server includes a cloud desktop protocol server and a multimedia player, the cloud desktop terminal includes a cloud desktop protocol client, and the cloud desktop protocol server includes a redirection server;

the redirection server is configured to receive, in accordance with a specified protocol, a undecoded media stream transmitted by a redirection client operating in the multimedia player, where the undecoded media stream is acquired by the redirection client from the multimedia player;

the cloud desktop protocol server is configured to transmit the undecoded media stream received by the redirection server to the cloud desktop protocol client;

the cloud desktop protocol client is configured to decode the undecoded media stream and invoke a local player to play a decoded multimedia;

where the multimedia player includes multiple types of players and multiple types of browsers, and the specified protocol is compatible with the multiple types of players and the multiple types of browsers and supports an one-to-many communication mechanism.

**[0007]** An embodiment of the present application further provides multimedia playing method for a cloud desktop, applicable to a cloud desktop server, where the cloud desktop server includes a cloud desktop protocol server and a multimedia player, the cloud desktop protocol server includes a redirection server; and the method includes:

receiving, using the redirection server included in the cloud desktop protocol server in accordance with a specified protocol, a undecoded media stream transmitted by a redirection client operating in the multimedia player, where the undecoded media stream is acquired by the

redirection client from the multimedia player; and

transmitting, through the cloud desktop protocol server, the undecoded media stream received by the redirection server to a cloud desktop protocol client of a cloud desktop terminal, to enable the cloud desktop protocol client to decode the undecoded media stream and invoke a local player of the cloud desktop terminal to play a decoded multimedia;

where the multimedia player includes multiple types of players and multiple types of browsers, and the specified protocol is compatible with the multiple types of players and the multiple types of browsers and supports an one-to-many communication mechanism.

**[0008]** An embodiment of the present application further provides a multimedia playing method for a cloud desktop, applicable to a cloud desktop terminal, where the cloud desktop terminal includes a cloud desktop protocol client, and the method includes:

receiving, through the cloud desktop protocol client, a undecoded media stream transmitted by a cloud desktop protocol server of a cloud desktop server, where the undecoded media stream is received, by a redirection server of the cloud desktop protocol server in accordance with a specified protocol, from a redirection client operating in a multimedia player, and the undecoded media stream is acquired by the redirection client from the multimedia player;

decoding the undecoded media stream; and

invoking a local player to play a decoded multimedia;

where the multimedia player operates in the cloud desktop server, the multimedia player includes multiple types of players and multiple types of browsers, and the specified protocol is compatible with the multiple types of players and the multiple types of browsers and supports an one-to-many communication mechanism.

**[0009]** An embodiment of the present application further provides a cloud desktop server, including a memory and a processor;

the memory is configured to store one or more computer instructions;

the processor is coupled to the memory and configured to execute the one or more computer instructions to:

receive, using a redirection server included in a cloud desktop protocol server in accordance with a specified protocol, a undecoded media stream transmitted by a redirection

client operating in a multimedia player, where the undecoded media stream is acquired by the redirection client from the multimedia player; and

transmit, through the cloud desktop protocol server, the undecoded media stream received by the redirection server to a cloud desktop protocol client of a cloud desktop terminal, to enable the cloud desktop protocol client to decode the undecoded media stream and invoke a local player of the cloud desktop terminal to play a decoded multimedia;

where the multimedia player operates in the cloud desktop server, the multimedia player includes multiple types of players and multiple types of browsers, and the specified protocol is compatible with the multiple types of players and the multiple types of browsers and supports an one-to-many communication mechanism.

**[0010]** An embodiment of the present application further provides a cloud desktop terminal, including a memory and a processor;

the memory is configured to store one or more computer instructions;

the processor is coupled to the memory and configured to execute the one or more computer instructions to:

receive, through a cloud desktop protocol client, a undecoded media stream transmitted by a cloud desktop protocol server of a cloud desktop server, where the undecoded media stream is received, by a redirection server of the cloud desktop protocol server in accordance with a specified protocol, from a redirection client operating in a multimedia player, and the undecoded media stream is acquired by the redirection client from the multimedia player;

decode the undecoded media stream; and

invoke a local player to play a decoded multimedia;

where the multimedia player operates in the cloud desktop server, the multimedia player includes multiple types of players and multiple types of browsers, and the specified protocol is compatible with the multiple types of players and the multiple types of browsers and supports an one-to-many communication mechanism.

**[0011]** An embodiment of the present application further provides a computer-readable storage medium storing computer instructions which, when executed by one or more processors, cause the one or more processors to execute the afore-mentioned multimedia playing method

for the cloud desktop.

**[0012]** In the embodiments of the present application, improvements are made to the cloud desktop protocol server of the cloud desktop server and the cloud desktop protocol client of the cloud desktop terminal. A redirection server is deployed in the cloud desktop protocol server, and redirection clients are respectively deployed within various multimedia players potentially used within the cloud desktop. The redirection server can perform one-to-many communication with respective redirection clients in accordance with a specified protocol, and the specified protocol is compatible with various players and various browsers. Based on this system architecture, in diverse playing scenarios in the cloud desktop, the redirection server is compatible with diverse multimedia players and redirects a undecoded media stream to a cloud desktop terminal for decoding and playing at the cloud desktop terminal. Thus, in the embodiments of the present application, multimedia redirection can be performed across diverse playing scenarios in the cloud desktop, which eliminates the need for decoding and other operations in the cloud desktop server, thereby effectively reducing the resource consumption of the cloud desktop server.

## BRIEF DESCRIPTION OF DRAWINGS

**[0013]** The drawings described herein are used to provide further understanding of the present application and constitute a part of the present application. Illustrative embodiments of the present application and their descriptions are intended to explain the present application and do not constitute improper limitations on the present application.

**[0014]** FIG. 1 is a schematic structural diagram of a multimedia playing system for a cloud desktop provided by an illustrative embodiment of the present application.

**[0015]** FIG. 2 is a schematic diagram of an illustrative internal structure for three types of illustrative multimedia player provided by an illustrative embodiment of the present application.

**[0016]** FIG. 3 is a schematic logic diagram of determining whether to perform multimedia redirection provided by an illustrative embodiment of the present application.

**[0017]** FIG. 4 is another schematic logic diagram of determining whether to perform multimedia redirection provided by an illustrative embodiment of the present application.

**[0018]** FIG. 5 is a schematic diagram of an illustrative internal structure of a cloud desktop protocol server provided by an illustrative embodiment of the present application.

**[0019]** FIG. 6 is a schematic logical diagram of a manner for data interaction between a cloud desktop server and a cloud desktop terminal provided by an illustrative embodiment of the present application.

**[0020]** FIG. 7 is a schematic flowchart of a multimedia playing method for a cloud desktop provided by an illustrative embodiment of the present application.

**[0021]** FIG. 8 is a schematic flowchart of another multimedia playing method for a cloud desktop provided by another illustrative embodiment of the present application.

**[0022]** FIG. 9 is a schematic structural diagram of a cloud desktop server provided by yet another illustrative embodiment of the present application.

**[0023]** FIG. 10 is a schematic structural diagram of a cloud desktop terminal provided by yet another illustrative embodiment of the present application.

## DESCRIPTION OF EMBODIMENTS

**[0024]** In order to make the objectives, technical solutions, and advantages of the present application clearer, the technical solutions of the present application will be clearly and completely described below in conjunction with the specific embodiments of the present application and the corresponding drawings. Obviously, the described embodiments are merely part of the embodiments of the present application, rather than all of the embodiments. Based on the embodiments in the present application, all other embodiments obtained by those of ordinary skill in the art without making creative work shall fall within the protection scope of the present application.

**[0025]** Currently, multimedia playing solutions for cloud desktops require relying on cloud desktop servers to perform decoding and other processing. This consumes a significant amount of resources in the cloud desktop servers, imposing a heavy load on the cloud desktop servers. To address this, in some embodiments of the present application: improvements are made to the cloud desktop protocol server of the cloud desktop server and the cloud desktop protocol client of the cloud desktop terminal. A redirection server is deployed in the cloud desktop protocol

server, and redirection clients are respectively deployed within various multimedia players potentially used within the cloud desktop. The redirection server can perform one-to-many communication with respective redirection clients in accordance with a specified protocol, and the specified protocol is compatible with various players and various browsers. Based on this system architecture, in diverse playing scenarios in the cloud desktop, the redirection server is compatible with diverse multimedia players and redirects a undecoded media stream to a cloud desktop terminal for decoding and playing at the cloud desktop terminal. Thus, in the embodiments of the present application, multimedia redirection can be performed across diverse playing scenarios in the cloud desktop, which eliminates the need for decoding and other operations in the cloud desktop server, thereby effectively reducing the resource consumption of the cloud desktop server.

**[0026]** The technical solutions provided by various embodiments of the present application are described in detail below in conjunction with the accompanying drawings.

**[0027]** FIG. 1 is a schematic structural diagram of a multimedia playing system for a cloud desktop provided by an illustrative embodiment of the present application. As shown in FIG. 1, the system includes a cloud desktop server and a cloud desktop terminal.

**[0028]** In the present embodiment, the cloud desktop server may refer to a cloud server configured to support an operation of a cloud desktop. It should be understood that the cloud desktop server may include components such as a CPU, memory, images, cloud disks disk, etc., and is a cloud server that is readily available and elastically scalable. The cloud desktop terminal may refer to a terminal device implementing cloud desktop technology, which can connect to the cloud desktop server and render the cloud desktop to the frontend, and redirect frontend input/output data such as mouse operations to the cloud desktop server.

**[0029]** With Reference to FIG. 1, in the present embodiment, the cloud desktop server may communicate with the cloud desktop terminal via a specific communication protocol, where the cloud desktop server may include a cloud desktop protocol server, and the cloud desktop terminal may include a cloud desktop protocol client. The cloud desktop protocol server is configured to implement the afore-mentioned specific communication protocol at the cloud desktop server side, and the cloud desktop protocol client is configured to implement the afore-mentioned specific communication protocol at the cloud desktop terminal side. It should be

understood that the cloud desktop protocol server may be a progress within the cloud desktop server, and similarly, the cloud desktop protocol client may be a progress within the cloud desktop terminal.

**[0030]** With continued reference to FIG. 1, in the present embodiment, a redirection server is further deployed in the cloud desktop protocol server, where the redirection server may be a thread contained within the progress corresponding to the cloud desktop protocol server.

**[0031]** Furthermore, in the present embodiment, a redirection client may be respectively deployed in various multimedia players potentially used within the cloud desktop. The multimedia player operates in the cloud desktop server, and similarly, the multimedia player may be a progress within the cloud desktop server, while the redirection client deployed therein may be a thread contained within the progress corresponding to the multimedia player. The multimedia player may be an application program in the cloud desktop server for multimedia playing. In the present embodiment, the multimedia player may be various players or browsers. Illustratively, the multimedia player may be a VLC player, a WMP player, a Chrome browser, an Edge browser, or a Firefox browser, etc. Of course, these are merely illustrative examples, and the present embodiment is not limited thereto. The embodiment can support various multimedia players currently potentially usable in the cloud desktop, as well as new types of multimedia player that may be expanded in the future.

**[0032]** On this basis and with reference to FIG. 1, the present embodiment proposes that the redirection server can perform one-to-many communication with the respective redirection clients in accordance with a specified protocol, and the specified protocol is compatible with various types of players and various types of browsers. Preferably, in the present embodiment, the specified protocol may adopt WebSocket protocol. Of course, in the present embodiment, the specified protocol may alternatively be a pipe protocol, as well as other protocols that currently exists or may emerge in the future and satisfies the afore-mentioned protocol requirements in the present embodiment. This embodiment is not limited thereto. In this manner, based on the specified protocol, the redirection server can communicate with the redirection clients deployed in various multimedia players, thereby providing compatible support for multimedia redirection across various multimedia players.

**[0033]** With reference to FIG. 1, a redirection client deployed in a multimedia player may

acquire a undecoded media stream from the multimedia player and may transmit the undecoded media stream to the redirection server in accordance with the afore-mentioned specified protocol. In this manner, the redirection server can receive the undecoded media stream sent by the redirection clients operating in various multimedia players in accordance with the specified protocol. That is, each of the redirection clients deployed in various multimedia players potentially used within the cloud desktop can send the undecoded media stream to the redirection server in the present embodiment in accordance with the specified protocol for subsequent processing by the redirection server.

**[0034]** During the research process, the inventor discovered that the playing scenarios in the cloud desktop are very diverse. For instance, local multimedia is typically expected to be played using players, while multimedia on webpages is generally expected to be played by browsers. Consequently, there exists a wide variety of types of multimedia players that may be utilized in the cloud desktop. The present embodiment proposes a communication mechanism in which a redirection server interfaces with multiple redirection clients in accordance with a specified protocol. This deployment architecture can support various playing scenarios in the cloud desktop. Users can on-demand select a desired multimedia player, all of which can achieve multimedia redirection by reusing the redirection server. Also, a new type of multimedia player can be continuously expanded in the cloud desktop. By simply deploying the redirection client proposed in the present embodiment in the new type of multimedia player, the new type of multimedia player can be supported, which can effectively improve the scalability of the multimedia player in the cloud desktop.

**[0035]** In the present embodiment, the multimedia player may be categorized into at least three types: non-customized players, browsers, and customized redirection-specified players. FIG. 2 is a schematic diagram of an illustrative internal structure for three types of illustrative multimedia player provided by an illustrative embodiment of the present application. With reference to FIG. 2, the interaction logic between the redirection client and the redirection server will be described below for these three illustrative types, respectively.

**[0036]** 1. Referring to FIG. 2, if the multimedia player is any non-customized player, a player plugin may be integrated within the non-customized player. In this case, although the multimedia player does not actually play a to-be-played multimedia source file, it still reads the

to-be-played multimedia source file. Accordingly, the player plugin can capture the to-be-played multimedia source file read by the multimedia player, and perform demultiplexing processing on the file to generate an audio stream and a video stream. Based on this, the player plugin can provide the demultiplexed audio stream and video stream as the undecoded media stream to the redirection client in the multimedia player; and the redirection client can send the received the audio stream and the video stream to the redirection server. Demultiplexing refers to a process of separating the to-be-played multimedia source file into an audio stream and a video stream. The non-customized player refers to a player developed by a third party.

**[0037]** FIG. 3 is a schematic logic diagram of determining whether to perform multimedia redirection provided by an illustrative embodiment of the present application. With reference to FIG. 3, in this case, the non-customized player may preferentially invoke the player plugin provided in the present embodiment, and the player plugin may be further configured to: if it is determined that the cloud desktop terminal is unable to support an encoding format corresponding to the to-be-played multimedia source file, abort the demultiplexing and subsequent processing, and return a fallback instruction to the non-customized player to trigger the non-customized player to perform multimedia playing. The non-customized player may provide the encoding format corresponding to the to-be-played multimedia source file to the player plugin in the present embodiment. The player plugin may also acquire an encoding format supported by the cloud desktop terminal. Through comparison, the player plugin can determine whether the cloud desktop terminal can support the encoding format corresponding to the to-be-played multimedia source file. For example, if the encoding format corresponding to the to-be-played multimedia source file is H264, but the cloud desktop terminal does not support decoding of this encoding format, the afore-mentioned fallback mechanism may be executed. That is, the non-customized player may preferentially invoke the player plugin provided by the present embodiment. In a condition that it is determined that multimedia redirection cannot be performed, the non-customized player may switch to invoke a decoding plugin to decode and play the multimedia source file, thereby falling back to the conventional playback approach to ensure normal completion of multimedia playing.

**[0038]** 2. With reference to FIG. 2, if the multimedia player is any web browser, a browser plugin may be integrated within the web browser. In this case, the browser plugin may capture

a video address from a current webpage, and download a to-be-played multimedia source file from the video address through an http request or other manners. The browser plugin may provide the downloaded to-be-played multimedia source file to the redirection client of the browser. The redirection client may transmit the to-be-played multimedia source file to the redirection server provided in the present embodiment. Based on this, the redirection server is further configured to: perform demultiplexing processing on the to-be-played multimedia source file transmitted by the redirection client of the browser to generate a corresponding audio stream and a video stream.

**[0039]** In this case, considering that the browser plugin is typically in JS format with poor processing efficiency, the demultiplexing processing task is offloaded to the redirection server provided in the present embodiment.

**[0040]** FIG. 4 is another schematic logic diagram of determining whether to perform multimedia redirection provided by an illustrative embodiment of the present application. Referring to FIG. 4, in this case, the browser may preferentially invoke the browser plugin provided in the present embodiment. The browser plugin may be further configured to: if the video address captured from the current webpage does not support file downloading, abort the operation of transmitting the to-be-played multimedia source file to the redirection server, and return a fallback instruction to the browser to trigger the browser to perform multimedia playing. Preferably, in this case, the browser plugin may be further configured to: detect whether or not a video tag is present in each open webpage of the browser; if the video tag is present in an open webpage of the browser, determine whether a specification of a video window meets a specified specification according to the video tag; if the specification of the video window does not meet the specified specification, execute the afore-mentioned fallback mechanism to trigger the browser to perform multimedia playing; if the specification of the video window meets the specified specification, execute the above-mentioned determination of whether the video address captured from the current webpage supports file downloading. The video tag is existing identification information within the browser, which is used to indicate whether a webpage contains a video that needs to be played, as well as a playing parameter such as the specification of the video window.

**[0041]** In certain scenarios, the video address may not support file downloading due to

permission restrictions. Consequently, multimedia redirection is aborted, and multimedia playing is fallen back to being played by the browser. If the video address supports downloading, the to-be-played multimedia source file may be transmitted to the redirection server through the redirection client. After receiving the multimedia source file, if it is determined that the cloud desktop terminal is unable to support an encoding format corresponding to the to-be-played multimedia source file, the redirection server suspends subsequent processing for the to-be-played multimedia source file; and returns a fallback instruction to the browser to trigger the browser to perform multimedia playing. In this manner, whether the cloud desktop terminal supports the encoding format corresponding to the to-be-played multimedia source file may be determined by the redirection server, thereby determining whether to implement the fallback mechanism.

**[0042]** 3. Referring to FIG. 2, if the multimedia player is a customized redirection-specified player, the redirection-specified player may read a to-be-played multimedia source file, perform demultiplexing processing on the to-be-played multimedia file to generate an audio stream and a video stream, and transmit the audio stream and the video stream as the undecoded media stream to the redirection server through the redirection client. The redirection-specified player is preconfigured with a program logic for acquiring the to-be-played multimedia file and performing demultiplexing processing on the to-be-played multimedia file.

**[0043]** In this scenario, the redirection-specified player is preconfigured with the program logic associated with the player plugin of the afore-mentioned non-customized player. Consequently, no embedded player plugin is required, and instead, the generated audio and video streams may be directly provided to the redirection client as the undecoded media stream in accordance with the preconfigured program logic.

**[0044]** In addition, in this case, the redirection-specified player may be further configured to: if it is determined that the cloud desktop terminal is unable to support a codec type corresponding to the to-be-played multimedia source file, play the to-be-played multimedia instead of performing the demultiplexing and subsequent processing. That is, the redirection-specified player also supports the afore-mentioned fallback mechanism.

**[0045]** Through the exemplification of the afore-mentioned three types of multimedia players, the interaction logic between the redirection client and the redirection server is

illustrated. it can be seen that in the present embodiment, for different multimedia players, the transmission of the undecoded media stream can be achieved through coordination between the redirection client and the redirection server, and when multimedia redirection is determined to be unfeasible during this process, the afore-mentioned fallback mechanism may be used to fall back to multimedia playing in the conventional playback manner to ensure normal completion of multimedia playback.

**[0046]** For the cloud desktop protocol server, the undecoded media stream received by the redirection server may be transmitted to the cloud desktop protocol client. As mentioned above, the redirection server may essentially be implemented as a thread within the cloud desktop protocol server. Therefore, the undecoded media stream acquired by the thread may be acquired by a progress, through which the stream may be transmitted to another progress within the cloud desktop terminal (i.e., the cloud desktop protocol client). At this stage, the undecoded media stream is successfully transmitted to the cloud desktop terminal.

**[0047]** On the cloud desktop terminal side, the cloud desktop protocol client may decode the received undecoded media stream and invoke a local player to play decoded multimedia. In this manner, the decoding task and the playing task are undertaken by the cloud desktop terminal instead of the cloud desktop server. This can effectively reduce the resource occupation of the cloud desktop server caused by multimedia playing.

**[0048]** In summary, in the present embodiment, improvements are made to the cloud desktop protocol server of the cloud desktop server and the cloud desktop protocol client of the cloud desktop terminal. A redirection server is deployed in the cloud desktop protocol server, and redirection clients are respectively deployed within various multimedia players potentially used within the cloud desktop. The redirection server can perform one-to-many communication with respective redirection clients in accordance with a specified protocol, and the specified protocol is compatible with various players and various browsers. Based on this system architecture, in diverse playing scenarios in the cloud desktop, the redirection server is compatible with diverse multimedia players and redirects a undecoded media stream to a cloud desktop terminal for decoding and playing at the cloud desktop terminal. Thus, in the embodiments of the present application, multimedia redirection can be performed across diverse playing scenarios in the cloud desktop, which eliminates the need for decoding and

other operations in the cloud desktop server, thereby effectively reducing the resource consumption of the cloud desktop server.

**[0049]** In the above or the following embodiments, the cloud desktop protocol server may further include a position management component. FIG. 5 is a schematic diagram of an illustrative internal structure of a cloud desktop protocol server provided by an illustrative embodiment of the present application. Referring to FIG. 5, the position management component of the cloud desktop protocol server may receive position information corresponding to a playback frame provided by the redirection server; it may also capture, via a specified interface, position information corresponding to an application window of the multimedia player and an application window of a further application of the cloud desktop; and calculate occlusion information of the playback frame according to the position information corresponding to the playback frame and the position information corresponding to each application window.

**[0050]** Referring to FIG. 5, the redirection clients of the multimedia player may acquire and provide position information corresponding to the playback frame to the redirection server, while the redirection server can transmit the received position information corresponding to the playback frame to the position management component in the present embodiment. The position information corresponding to the playback frame refers to position information of the playback frame within the application window of the multimedia player. For example, for a player, a playback frame occupies simply a partial region of the player's application window; for a browser, a playback frame occupies simply a partial region of the browser's application window.

**[0051]** In addition, in the present embodiment, the position management component can capture the position information of the application window corresponding to each application within the cloud desktop by invoking a specified interface provided in the operating system. For example, for a cloud desktop using a Windows operating system, the position management component can invoke a WinEventHook interface to acquire the position information of the application window corresponding to each application within the cloud desktop. The position information of the application window may include window coordinate information and a display order relative to a further window. An application window that is upper in the display

order may obscure an application window that is lower in the display order. In this manner, the position management component can determine a target application window that is higher in the display order than that of the multimedia player, and calculate occlusion information of the playback frame according to the position information of the target application window, the application window of the multimedia player and the playback frame. It can be understood that the occlusion information may be coordinate information of an occluded region within the playback frame, or alternatively, coordinate information of a non-occluded region in the playback frame, etc., which is not be limited in the present embodiment.

**[0052]** In the present embodiment, the cloud desktop protocol server may acquire a desktop image of the cloud desktop, and transmit the desktop image, and the position information and the occlusion information corresponding to the playback frame to the cloud desktop protocol client. The desktop image may be obtained by capturing a screenshot of the cloud desktop using existing screenshot tools within the cloud desktop, which requires no further elaboration herein. It should be noted that since no multimedia playing occurs in the cloud desktop, there exists no playback frame therein. Therefore, in the present embodiment, the playback frame in the cloud desktop may be displayed as a specified patter, where the specified patter may include, but is not limited to: a solid white pattern, a solid black pattern, or a mosaic pattern, etc.

**[0053]** In the present embodiment, the cloud desktop protocol client may be configured to: acquire a playback frame of a local player; crop the playback frame according to the occlusion information; splice the cropped playback frame with the desktop image according to the position information of the playback frame; and display the spliced image on the cloud desktop terminal. It should be noted that in the present embodiment, when the local player performs multimedia playing on the cloud desktop terminal, its frontend interface is not displayed. Instead, the multimedia playing occurs in the background to generate a playback frame. As a result, the local player of the cloud desktop terminal does not interfere with the cloud desktop's foreground rendering. In addition, the cloud desktop protocol client in the present embodiment can also crop the playback frame generated by the local player in accordance with the occlusion information received from the cloud desktop protocol server, that is, crop a region of the playback frame that is obscured by a further application window. On this basis, the cropped playback frame can be spliced into the position of the playback frame within the desktop image

to generate a final rendered frame. The cloud desktop terminal displays the rendered frame at a certain frame rate to achieve multimedia playing.

**[0054]** Building upon the three exemplary types of multimedia players summarized in the preceding embodiments, the following describes an implementation scheme for the redirection client to acquire the position information corresponding to the playback frame.

**[0055]** If the multimedia player is any non-customized player, position information corresponding to a playback frame provided by a player plugin embedded in the multimedia player is received, where the position information corresponding to the playback frame is monitored and provided by the multimedia player to the player plugin. In this case, in the present embodiment, the multimedia player can monitor by itself the position information corresponding to the playback frame and provide it to the player plugin in the present embodiment; the player plugin may provide the received position information corresponding to the playback frame to the redirection client of the multimedia player, and then the redirection client transmits the position information corresponding to the playback frame to the redirection server.

**[0056]** If the multimedia player is any browser, position information corresponding to a playback frame provided by a browser plugin embedded in the multimedia player is received, where the position information corresponding to the playback frame is monitored and provided by the browser to the browser plugin. In this case, in the present embodiment, the multimedia player may monitor by itself the position information corresponding to the playback frame and provide it to the browser plugin in the present embodiment; the browser plugin may provide the received position information corresponding to the playback frame to the redirection client of the multimedia player, and then the redirection client transmits the position information corresponding to the playback frame to the redirection server.

**[0057]** If the multimedia player is a customized redirection-specified player, position information corresponding to a playback frame monitored by the redirection-specified player is received, where the redirection-specified player is preconfigured with a program logic for position monitoring and transmitting. In this case, the redirection-specified player can monitor the position information corresponding to the playback frame by itself and provide it to the redirection client therein, and then the redirection client transmits the position information

corresponding to the playback frame to the redirection server.

**[0058]** As a result, the position information corresponding to playback images from various multimedia players potentially used in the cloud desktop can reach the redirection server, and the redirection server can provide the acquired position information corresponding to the playback frame to the position management component in the present embodiment, serving as one of the bases for the process of calculating the occlusion information performed by the position management component. Accordingly, in the present embodiment, the position management component can accurately calculate the occlusion information corresponding to the playback frame and provide it to the cloud desktop protocol client of the cloud desktop terminal, to enable the cloud desktop protocol client to precisely render the playback frame generated by its local player within the desktop image of the cloud desktop according to the occlusion information, thereby ensuring the foreground rendering performance during multimedia playing in a multimedia redirection manner.

**[0059]** In the afore-mentioned or following embodiments, in addition to the undecoded media stream, and the position information and occlusion information corresponding to the playback frame provided in the afore-mentioned embodiments, the cloud desktop server and the cloud desktop terminal can also exchange a command stream involved in the multimedia playing process. The command stream includes various control commands that generated during multimedia playback.

**[0060]** In the present embodiment, the cloud desktop protocol server may also acquire a first command stream generated within the multimedia player from the redirection server, where the first command stream is acquired and provided by the redirection client of the multimedia player to the redirection server; and transmit the first command stream to the cloud desktop protocol client, to enable the cloud desktop protocol client to perform play controlling on the local player within the cloud desktop terminal in accordance with the first command stream. The first command stream may include volume adjustment, play, pause, end-of-file, or other control commands generated in the multimedia player.

**[0061]** In the present embodiment, the cloud desktop protocol server may also receive a second command stream transmitted by the cloud desktop protocol client, where the second command stream is monitored by the cloud desktop protocol client from the local player of the

cloud desktop terminal; and perform play controlling on the multimedia player in accordance with the second command stream. The second command stream may include search, fast forward, fast rewind, streaming, or other controlling commands generated in the local player of the cloud desktop terminal. These control commands are redirected to the multimedia player of the cloud desktop to perform play controlling on the multimedia player, thereby ensuring the accuracy of the front-end rendering.

**[0062]** In addition, in the present embodiment, considering that a data link of multimedia redirection is relatively long and involves diverse types of transmission content with relatively complex formats, the present embodiment proposes that data interaction between the cloud desktop protocol server and the cloud desktop protocol client is performed using serialization and deserialization techniques. The adoption of serialization and deserialization technologies enables efficiently content transmission both inter-progress and across networks. Serialization may be understood as a process of storing an object into a byte sequence, while deserialization may be understood as a process of reconstructing the byte sequence as the original object for use.

**[0063]** FIG. 6 is a schematic logical diagram of a manner for data interaction between a cloud desktop server and a cloud desktop terminal provided by an illustrative embodiment of the present application. Referring to FIG. 6, in the present embodiment, a data transmission plugin may be deployed in the cloud desktop protocol server and the cloud desktop protocol client, respectively, where the data transmission plugin can be executed as a thread. Illustratively, the data transmission plugin may adopt a serializer plugin.

**[0064]** Referring to FIG. 6, in the present embodiment, the cloud desktop protocol server may be configured to perform, through the data transmission plugin, serialization on and then transmit the undecoded media stream, the position information and occlusion information of the playback frame mentioned in the afore-mentioned embodiment, and the first command stream involved in the present embodiment, respectively, to the cloud desktop protocol client. The cloud desktop protocol client may be configured to perform, through the data transmission plugin, deserialization on the content transmitted by the cloud desktop protocol server to reconstruct the undecoded media stream, and the position information and occlusion information of the playback frame.

**[0065]** In addition, the cloud desktop protocol client may be configured to perform, through the data transmission plugin, serialization on and then transmit the second command stream acquired by the cloud desktop protocol client to the cloud desktop protocol server. The cloud desktop protocol server may be configured to perform, through the data transmission plugin, serialization processing to reconstruct the second command stream.

**[0066]** As a result, during remote communication between the cloud desktop protocol server and the cloud desktop protocol client, these two progresses are configured to mutually exchange multiple types of data. All data, without regard to the type of the data, are transmitted across the network in the form of binary sequences. A transmitting party converts the data into a byte sequence for transmission over the network, while a receiving party may perform deserialization on the byte sequence to reconstruct the original data, which enables the transmission of the afore-mentioned multiple types of data between the cloud desktop protocol server and the cloud desktop protocol client.

**[0067]** It should be noted that in the present embodiment, in addition to the afore-mentioned serialization and deserialization approach, other approaches may be adopted for data interaction between the cloud desktop protocol server and the cloud desktop protocol client, which is not limited in the present embodiment.

**[0068]** Accordingly, in the present embodiment, the command stream involved in the multimedia playing process may also be exchanged between the cloud desktop protocol server and the cloud desktop protocol client. In addition, the serialization and deserialization approach may be adopted to transmit multiple types of data that need to be exchanged between the cloud desktop protocol server and the cloud desktop protocol client. This allows different types of data that need to be transmitted between the cloud desktop protocol server and the cloud desktop protocol client to be transmitted through appropriate channels, thereby ensuring the timeliness and accuracy of data transmission and improving the smoothness of multimedia playback.

**[0069]** FIG. 7 is a schematic flowchart of a multimedia playing method for a cloud desktop provided by an illustrative embodiment of the present application. The method is executable by a data processing apparatus, which may be implemented as software, hardware, or a combination thereof, and may be integrated in a cloud desktop server. The cloud desktop server includes a cloud desktop protocol server and a multimedia player, and the cloud desktop

protocol server includes a redirection server. Referring to FIG. 7, the method may include:

step 700, receive, using the redirection server included in the cloud desktop protocol server in accordance with a specified protocol, a undecoded media stream transmitted by a redirection client operating in the multimedia player, where the undecoded media stream is acquired by the redirection client from the multimedia player; and

step 701, transmit, through the cloud desktop protocol server, the undecoded media stream received by the redirection server to a cloud desktop protocol client of a cloud desktop terminal, to enable the cloud desktop protocol client to decode the undecoded media stream and invoke a local player of the cloud desktop terminal to play a decoded multimedia;

where the multimedia player includes multiple types of players and multiple types of browsers, and the specified protocol is compatible with the multiple types of players and the multiple types of browsers and supports an one-to-many communication mechanism.

**[0070]** In an optional embodiment, where the specified protocol adopts a websocket communication protocol.

**[0071]** In an optional embodiment, the method may further include:

if the multimedia player is any non-customized player, receiving, using the redirection client, an undecoded media stream provided by a player plugin embedded in the multimedia player, where the undecoded media stream is an audio stream and a video stream generated by the player plugin performing demultiplexing processing on a to-be-played multimedia source file; and transmitting the audio stream and the video stream to the redirection server; or,

if the multimedia player is any browser, receiving, using the redirection client, an undecoded media stream provided by a browser plugin embedded in the multimedia player, where the undecoded media stream is a to-be-played multimedia source file downloaded by the browser plugin according to a video address captured from a current webpage; and transmitting the to-be-played multimedia source file to the redirection server; performing, by the redirection server, demultiplexing processing on the to-be-played multimedia source file transmitted by the redirection client of the browser to generate a corresponding audio stream and video stream; or,

if the multimedia player is a customized redirection-specified player, receiving, using the redirection client, an undecoded media stream provided by the redirection-specified

player, where the undecoded media stream is an audio stream and a video stream generated by the redirection-specified player performing demultiplexing processing on a to-be-played multimedia file; and transmitting the audio stream and the video stream to the redirection server; where the redirection-specified player is preconfigured with a program logic for acquiring the to-be-played multimedia file and performing demultiplexing processing on the to-be-played multimedia file.

**[0072]** In an optional embodiment, the method may further include:

determining, using the player plugin, that the cloud desktop terminal is unable to support an encoding format corresponding to the to-be-played multimedia source file, then aborting the demultiplexing and a subsequent operation; and returning a fallback instruction to the non-customized player to trigger the non-customized player to perform multimedia playing; where the non-customized player preferentially invokes the player plugin; or,

determining, using the browser plugin, that the video address captured from the current webpage does not support file downloading, then aborting the operation of transmitting the to-be-played multimedia source file to the redirection server, and returning a fallback instruction to the browser to trigger the browser to perform multimedia playing; after receiving the multimedia source file using the redirection server, if it is determined that the cloud desktop terminal is unable to support an encoding format corresponding to the to-be-played multimedia source file, suspending subsequent processing for the to-be-played multimedia source file, and returning a fallback instruction to the browser to trigger the browser to perform multimedia playing; where the browser preferentially invokes the browser plugin; or,

determining, using the redirection-specified player, that the cloud desktop terminal is unable to support an encoding format corresponding to the to-be-played multimedia source file, playing the to-be-played multimedia instead of performing the demultiplexing and subsequent processing.

**[0073]** In an optional embodiment, the cloud desktop protocol server may further include a position management component, and the method may further include:

receiving, using the position management component, position information corresponding to a playback frame provided by the redirection server, where the position information corresponding to the playback frame is acquired and provided by the redirection

clients of the multimedia player to the redirection server; capturing, via a specified interface, position information corresponding to an application window of the multimedia player and an application window of a further application of the cloud desktop; and calculating occlusion information of the playback frame according to the position information corresponding to the playback frame and the position information corresponding to each application window;

transmitting, through the cloud desktop protocol server, a desktop image of the cloud desktop, and the position information and the occlusion information corresponding to the playback frame to the cloud desktop protocol client, to enable the cloud desktop protocol client to acquire a playback frame of a local player; crop the playback frame according to the occlusion information; splice the cropped playback frame with the desktop image according to the position information of the playback frame; and display the spliced image on the cloud desktop terminal.

**[0074]** In an optional embodiment, the method may further include:

if the multimedia player is any non-customized player, receiving, using the redirection client therein, position information corresponding to a playback frame provided by a player plugin embedded in the multimedia player, where the position information corresponding to the playback frame is monitored and provided by the multimedia player to the player plugin; or,

if the multimedia player is any browser, receiving, using the redirection client therein, position information corresponding to a playback frame provided by a browser plugin embedded in the multimedia player, where the position information corresponding to the playback frame is monitored and provided by the browser to the browser plugin; or,

if the multimedia player is a customized redirection-specified player, receiving, using the redirection client therein, position information corresponding to a playback frame monitored by the redirection-specified player, where the redirection-specified player is preconfigured with a program logic for position monitoring and transmitting.

**[0075]** In an optional embodiment, the method may further include:

acquiring, using the cloud desktop protocol server, a first command stream generated within the multimedia player from the redirection server, where the first command stream is acquired and provided by the redirection client of the multimedia player to the

redirection server;

transmitting the first command stream to the cloud desktop protocol client, to enable the cloud desktop protocol client to perform play controlling on the local player within the cloud desktop terminal in accordance with the first command stream;

receiving a second command stream transmitted by the cloud desktop protocol client, where the second command stream is monitored by the cloud desktop protocol client from the local player; and

performing play controlling on the multimedia player in accordance with the second command stream.

**[0076]** In an optional embodiment, data interaction between the cloud desktop protocol server and the cloud desktop protocol client is performed using serialization and deserialization techniques.

**[0077]** In an optional embodiment, the playback frame in the desktop image is displayed as a specified pattern, and the specified pattern includes a solid white pattern, a solid black pattern, or a mosaic pattern.

**[0078]** It should be noted that the technical details in the afore-mentioned embodiments of the multimedia playing method for the cloud desktop can refer to the relevant descriptions of the cloud desktop server in the afore-mentioned system embodiments. For economy of description, these details are not reiterated herein, but this shall not be construed as limiting the scope of protection sought in the present application.

**[0079]** FIG. 8 is a schematic flowchart of another multimedia playing method for a cloud desktop provided by another illustrative embodiment of the present application. The method is executable by a data processing apparatus, which may be implemented as software, hardware, or a combination thereof, and may be integrated in a cloud desktop terminal. The cloud desktop terminal may include a cloud desktop protocol client. Referring to FIG. 8, the method may include:

step 800, receive, through the cloud desktop protocol client, a undecoded media stream transmitted by a cloud desktop protocol server of a cloud desktop server, where the undecoded media stream is received, by a redirection server of the cloud desktop protocol server in accordance with a specified protocol, from a redirection client operating in a multimedia

player, and the undecoded media stream is acquired by the redirection client from the multimedia player;

step 801, decode the undecoded media stream; and

step 802, invoke a local player to play a decoded multimedia;

where the multimedia player operates in the cloud desktop server, the multimedia player includes multiple types of players and multiple types of browsers, and the specified protocol is compatible with the multiple types of players and the multiple types of browsers and supports an one-to-many communication mechanism.

**[0080]** In an optional embodiment, the specified protocol adopts a websocket communication protocol.

**[0081]** In an optional embodiment, the method may further include:

acquiring, through the cloud desktop protocol client, a playback frame of a local player;

cropping the playback frame according to occlusion information;

splicing the cropped playback frame with a desktop image according to position information of the playback frame; and

displaying the spliced image on the cloud desktop terminal;

where the occlusion information is generated by a position management component included in the cloud desktop protocol server performing calculation according to the position information corresponding to the playback frame provided by the redirection server, and captured position information corresponding to an application window of the multimedia player and an application window of a further application.

**[0082]** In an optional embodiment, the method may further include:

transmitting, through the cloud desktop protocol client, a second command stream to the cloud desktop protocol server, to enable the cloud desktop protocol server to perform play controlling on the multimedia player in accordance with the second command stream;

where the second command stream is monitored by the cloud desktop protocol client from the local player.

**[0083]** In an optional embodiment, data interaction between the cloud desktop protocol server and the cloud desktop protocol client is performed using serialization and deserialization

techniques.

**[0084]** In an optional embodiment, the playback frame in the desktop image is displayed as a specified pattern, and the specified pattern includes a solid white pattern, a solid black pattern, or a mosaic pattern.

**[0085]** It should be noted that the technical details in the afore-mentioned embodiments of the multimedia playing method for the cloud desktop can refer to the relevant descriptions of the cloud desktop terminal in the afore-mentioned system embodiments. For economy of description, these details are not reiterated herein, but this shall not be construed as limiting the scope of protection sought in the present application.

**[0086]** It should be noted that in some of the processes described in the above embodiments and drawings, multiple operations are presented in a specific order. However, it should be clearly understood that these operations need not necessarily be executed in the order presented in this document, or these operations may be executed in parallel. The numbering of operations (e.g., 801, 802, etc.) is simply for distinguishing between different operations and does not imply any execution sequence. Additionally, these processes may include a greater or lesser number of operations, and such operations may be executed either sequentially or in parallel. It should be noted that the terms “first,” “second,” and similar descriptions in this document are used to distinguish between different command streams, etc., and do not imply any sequential order or limit “first” and “second” to being of different types.

**[0087]** FIG. 9 is a schematic structural diagram of a cloud desktop server provided by yet another illustrative embodiment of the present application. As shown in FIG. 9, the cloud desktop server includes: a memory 90 and a processor 91.

**[0088]** The processor 91 is coupled to the memory 90 and is configured to execute a computer program in the memory 90 to:

receive, using a redirection server included in a cloud desktop protocol server in accordance with a specified protocol, a undecoded media stream transmitted by a redirection client operating in a multimedia player, where the undecoded media stream is acquired by the redirection client from the multimedia player; and

transmit, through the cloud desktop protocol server, the undecoded media stream received by the redirection server to a cloud desktop protocol client of a cloud desktop terminal,

to enable the cloud desktop protocol client to decode the undecoded media stream and invoke a local player of the cloud desktop terminal to play a decoded multimedia;

where the multimedia player operates in the cloud desktop server, the multimedia player includes multiple types of players and multiple types of browsers, and the specified protocol is compatible with the multiple types of players and the multiple types of browsers and supports an one-to-many communication mechanism.

**[0089]** In an optional embodiment, the specified protocol adopts a websocket communication protocol.

**[0090]** In an optional embodiment, the processor 91 may be further configured to:

if the multimedia player is any non-customized player, receive, using the redirection client, an undecoded media stream provided by a player plugin embedded in the multimedia player, where the undecoded media stream is an audio stream and a video stream generated by the player plugin performing demultiplexing processing on a to-be-played multimedia source file; and transmit the audio stream and the video stream to the redirection server; or,

if the multimedia player is any browser, receive, using the redirection client, an undecoded media stream provided by a browser plugin embedded in the multimedia player, where the undecoded media stream is a to-be-played multimedia source file downloaded by the browser plugin according to a video address captured from a current webpage; and transmit the to-be-played multimedia source file to the redirection server; perform, by the redirection server, demultiplexing processing on the to-be-played multimedia source file transmitted by the redirection client of the browser to generate a corresponding audio stream and video stream; or,

if the multimedia player is a customized redirection-specified player, receive, using the redirection client, an undecoded media stream provided by the redirection-specified player, where the undecoded media stream is an audio stream and a video stream generated by the redirection-specified player performing demultiplexing processing on a to-be-played multimedia file; and transmit the audio stream and the video stream to the redirection server; where the redirection-specified player is preconfigured with a program logic for acquiring the to-be-played multimedia file and performing demultiplexing processing on the to-be-played multimedia file.

**[0091]** In an optional embodiment, the processor 91 may be further configured to:

determine, through using the player plugin, that the cloud desktop terminal is unable to support an encoding format corresponding to the to-be-played multimedia source file, then abort the demultiplexing and a subsequent operation; and return a fallback instruction to the non-customized player to trigger the non-customized player to perform multimedia playing; where the non-customized player preferentially invokes the player plugin; or,

determine, using the browser plugin, that the video address captured from the current webpage does not support file downloading, then abort the operation of transmitting the to-be-played multimedia source file to the redirection server, and return a fallback instruction to the browser to trigger the browser to perform multimedia playing; after receiving the multimedia source file using the redirection server, if it is determined that the cloud desktop terminal is unable to support an encoding format corresponding to the to-be-played multimedia source file, suspend subsequent processing for the to-be-played multimedia source file, and return a fallback instruction to the browser to trigger the browser to perform multimedia playing; where the browser preferentially invokes the browser plugin; or,

determine, using the redirection-specified player, that the cloud desktop terminal is unable to support an encoding format corresponding to the to-be-played multimedia source file, play the to-be-played multimedia instead of performing the demultiplexing and subsequent processing.

**[0092]** In an optional embodiment, the cloud desktop protocol server may further include a position management component, and the processor 91 may be further configured to:

receive, using the position management component, position information corresponding to a playback frame provided by the redirection server, where the position information corresponding to the playback frame is acquired and provided by the redirection clients of the multimedia player to the redirection server; capture, via a specified interface, position information corresponding to an application window of the multimedia player and an application window of a further application of the cloud desktop; and calculate occlusion information of the playback frame according to the position information corresponding to the playback frame and the position information corresponding to each application window;

transmit, through the cloud desktop protocol server, a desktop image of the cloud desktop, and the position information and the occlusion information corresponding to the

playback frame to the cloud desktop protocol client, to enable the cloud desktop protocol client to acquire a playback frame of a local player; crop the playback frame according to the occlusion information; splice the cropped playback frame with the desktop image according to the position information of the playback frame; and display the spliced image on the cloud desktop terminal.

**[0093]** In an optional embodiment, the processor 91 may be further configured to:

if the multimedia player is any non-customized player, receive, using the redirection client therein, position information corresponding to a playback frame provided by a player plugin embedded in the multimedia player, where the position information corresponding to the playback frame is monitored and provided by the multimedia player to the player plugin; or,

if the multimedia player is any browser, receive, using the redirection client therein, position information corresponding to a playback frame provided by a browser plugin embedded in the multimedia player, where the position information corresponding to the playback frame is monitored and provided by the browser to the browser plugin; or,

if the multimedia player is a customized redirection-specified player, receive, using the redirection client therein, position information corresponding to a playback frame monitored by the redirection-specified player, where the redirection-specified player is preconfigured with a program logic for position monitoring and transmitting.

**[0094]** In an optional embodiment, the processor 91 may be further configured to:

acquire, using the cloud desktop protocol server, a first command stream generated within the multimedia player from the redirection server, where the first command stream is acquired and provided by the redirection client of the multimedia player to the redirection server;

transmit the first command stream to the cloud desktop protocol client, to enable the cloud desktop protocol client to perform play controlling on the local player within the cloud desktop terminal in accordance with the first command stream;

receive a second command stream transmitted by the cloud desktop protocol client, where the second command stream is monitored by the cloud desktop protocol client from the local player; and

perform play controlling on the multimedia player in accordance with the second command stream.

**[0095]** In an optional embodiment, data interaction between the cloud desktop protocol server and the cloud desktop protocol client is performed using serialization and deserialization techniques.

**[0096]** In an optional embodiment, the playback frame in the desktop image is displayed as a specified pattern, and the specified pattern includes a solid white pattern, a solid black pattern, or a mosaic pattern.

**[0097]** Furthermore, as shown in FIG. 9, the cloud desktop server further includes: a communication component 92, a power supply assembly 93 and other components. FIG. 9 simply schematically shows some components, which does not imply that the cloud desktop server only includes the components shown in FIG. 9.

**[0098]** It should be noted that the technical details in the afore-mentioned embodiments related to the cloud desktop server can refer to the relevant descriptions of the cloud desktop server in the afore-mentioned system embodiments. For economy of description, these details are not reiterated herein, but this shall not be construed as limiting the scope of protection sought in the present application.

**[0099]** FIG. 10 is a schematic structural diagram of a cloud desktop terminal provided by yet another illustrative embodiment of the present application. As shown in FIG. 10, the cloud desktop terminal includes: a memory 10 and a processor 11.

**[0100]** The processor 11 is coupled to the memory 10 and is configured to execute a computer program stored in the memory 10 to:

receive, through a cloud desktop protocol client, a undecoded media stream transmitted by a cloud desktop protocol server of a cloud desktop server, where the undecoded media stream is received, by a redirection server of the cloud desktop protocol server in accordance with a specified protocol, from a redirection client operating in a multimedia player, and the undecoded media stream is acquired by the redirection client from the multimedia player;

decode the undecoded media stream; and

invoke a local player to play a decoded multimedia;

where the multimedia player operates in the cloud desktop server, the multimedia player includes multiple types of players and multiple types of browsers, and the specified

protocol is compatible with the multiple types of players and the multiple types of browsers and supports an one-to-many communication mechanism.

**[0101]** In an optional embodiment, the specified protocol adopts a websocket communication protocol.

**[0102]** In an optional embodiment, the processor 11 may be further configured to:

acquire, through the cloud desktop protocol client, a playback frame of a local player;

crop the playback frame according to occlusion information;

splice the cropped playback frame with a desktop image according to position information of the playback frame; and

displayed the spliced image on the cloud desktop terminal;

where the occlusion information is generated by a position management component included in the cloud desktop protocol server performing calculation according to the position information corresponding to the playback frame provided by the redirection server, and captured position information corresponding to an application window of the multimedia player and an application window of a further application.

**[0103]** In an optional embodiment, the processor 11 may be further configured to:

transmit, through the cloud desktop protocol client, a second command stream to the cloud desktop protocol server, to enable the cloud desktop protocol server to perform play controlling on the multimedia player in accordance with the second command stream;

where the second command stream is monitored by the cloud desktop protocol client from the local player.

**[0104]** In an optional embodiment, data interaction between the cloud desktop protocol server and the cloud desktop protocol client is performed using serialization and deserialization techniques.

**[0105]** In an optional embodiment, the playback frame in the desktop image is displayed as a specified pattern, and the specified pattern includes a solid white pattern, a solid black pattern, or a mosaic pattern.

**[0106]** Furthermore, as shown in FIG. 10, the cloud desktop terminal further includes: a communication component 12, a power supply assembly 13, a display component 14, an audio component 15 and other components. FIG. 10 simply schematically shows some components,

which does not imply that the cloud desktop server only includes the components shown in FIG. 10.

**[0107]** It should be noted that the technical details in the afore-mentioned embodiments related to the cloud desktop terminal can refer to the relevant descriptions of the cloud desktop terminal in the afore-mentioned system embodiments. For economy of description, these details are not reiterated herein, but this shall not be construed as limiting the scope of protection sought in the present application.

**[0108]** Accordingly, an embodiment of the present application further provides a computer-readable storage medium storing a computer program which, when executed, can implement the steps that can be executed by the cloud desktop server or the cloud desktop terminal in the afore-mentioned method embodiment.

**[0109]** The memories illustrated in FIGS. 9 and 10 are configured to store computer programs and may be configured to store various other data to support operations on a computing platform. Examples of such data include instructions for any application or method operating on the computing platform, contact data, phonebook data, messages, images, videos, etc. The memory may be implemented by any type of volatile or non-volatile storage device, or a combination thereof, such as static random-access memory (SRAM), electrically erasable programmable read-only memory (EEPROM), erasable programmable read-only memory (EPROM), programmable read-only memory (PROM), read-only memory (ROM), magnetic memory, flash memory, magnetic disk, or optical disk.

**[0110]** The communication components in FIGS. 9 and 10 are configured to facilitate wired or wireless communication between a device where the communication component is located and further device(s). The device where the communication component is located can access wireless networks based on communication standards, such as WiFi, 2G, 3G, 4G/LTE, 5G and other mobile communication networks, or a combination thereof. In an exemplary embodiment, the communication component receives a broadcast signal or broadcast-related information from an external broadcast management system via a broadcast channel. In an exemplary embodiment, the communication component further includes a near field communication (NFC) module to facilitate short-range communications. For example, the NFC module may be implemented based on radio frequency identification (RFID) technology, infrared data

association (IrDA) technology, ultra-wideband (UWB) technology, Bluetooth (BT) technology and other technologies.

**[0111]** The display component in FIG. 10 includes a screen, which may include a liquid crystal display (LCD) and a touch panel (TP). If the screen includes a touch panel, the screen may be implemented as a touch screen to receive an input signal from a user. The touch panel includes one or more touch sensors to sense touches, slides, and gestures on the touch panel. The touch sensor may not only sense a boundary of a touch or sliding action, but also detect a duration and pressure associated with the touch or sliding operation.

**[0112]** The power supply components in FIGS. 9 and 10 provide power to various components of devices in which the power supply components are located. The power supply assembly may include a power management system, one or more power supplies, and other components associated with generating, managing, and distributing electrical power for the device in which the power supply assembly is implemented.

**[0113]** The audio component in FIG. 10 described above may be configured to output and/or input audio signals. For example, the audio component includes a microphone (MIC), and when the device including the audio component is in an operation mode, such as a call mode, a recording mode, and a speech recognition mode, the microphone is configured to receive an external audio signal. The received audio signal may be further stored in a memory or transmitted via a communication component. In some embodiments, the audio component further includes a speaker for outputting audio signals.

**[0114]** Those skilled in the art should understand that the embodiments of the present application may be provided as methods, systems, or computer program products. Therefore, the present application may take the form of an entirely hardware embodiment, an entirely software embodiment, or an embodiment combining software and hardware aspects. Moreover, the present application may take the form of a computer program product implemented on one or more computer-usable storage media (including but not limited to disk storage, CD-ROM, optical storage, etc.) containing computer-usable program codes.

**[0115]** The present application is described with reference to flowchart(s) and/or block diagram(s) of method(s), device(s) (system(s)), and computer program product(s) according to embodiments of the present application. It should be understood that each flow and/or block in

the flowchart(s) and/or block diagram(s), and a combination of the flow(s) and/or block(s) in the flowchart(s) and/or block diagram(s) can be implemented by computer program instructions. These computer program instructions may be provided to a processor of a general-purpose computer, a special-purpose computer, an embedded processor or other programmable data processing device to produce a machine, such that the instructions executed by the processor of the computer or other programmable data processing device create a means for implementing function(s) specified in one or more flows in the flowchart(s) and/or one or more blocks in the block diagram(s).

**[0116]** These computer program instructions may also be stored in a computer-readable memory that can direct a computer or other programmable data processing device to function in a specific manner, so that the instructions stored in the computer-readable memory produce an article of manufacture including instruction mean(s) which implements function(s) specified in one or more flows of the flowchart(s) and/or one or more blocks of the block diagram(s).

**[0117]** These computer program instructions may also be loaded onto a computer or other programmable data processing device so that a series of operational steps are executed on the computer or other programmable device to produce a computer-implemented processing, whereby the instructions executed on the computer or other programmable device provide steps for implementing the function(s) specified in one or more flows in the flowchart(s) and/or one or more blocks in the block diagram(s).

**[0118]** In a typical configuration, a computing device includes one or more processors (CPU), input/output interfaces, network interfaces, and memory.

**[0119]** Memory may include forms such as non-permanent storage (e.g., random-access memory (RAM)) and/or non-volatile memory (e.g., read-only memory (ROM) or flash memory (flash RAM)) in computer-readable media. Memory is an example of a computer-readable medium.

**[0120]** Computer-readable media include both permanent and non-permanent, removable and non-removable media that can implement information storage using any method or technology. The information may be computer-readable instructions, data structures, program modules, or other data. Examples of computer storage media include, but are not limited to, phase change memory (PRAM), static random-access memory (SRAM), dynamic random-

access memory (DRAM), other types of random-access memory (RAM), read-only memory (ROM), electrically erasable programmable read-only memory (EEPROM), flash memory or other memory technology, compact disk read-only memory (CD-ROM), digital versatile disk (DVD) or other optical storage, magnetic cassettes, magnetic tape magnetic disk storage or other magnetic storage devices, or any other non-transmission media that can be used to store information that can be accessed by a computing device. According to the definition provided in this document, computer-readable media does not include temporary computer-readable media (transitory media), such as modulated data signals and carrier waves.

**[0121]** It should also be noted that the terms "comprise," "include," or any other variants thereof, are intended to cover non-exclusive inclusions, such that a process, method, product, or device that includes a series of elements includes not only those elements, but also other elements not explicitly listed or inherent to such process, method, product, or device. In the absence of more limitations, the elements defined by the phrase "comprising a…" do not exclude the presence of additional identical element(s) in the process, method, product, or device that includes the said element.

**[0122]** The above description is merely the embodiments of the present application and is not intended to limit the present application. For those skilled in the art, various modifications and variations may be made to the present application. Any modifications, equivalent substitutions, improvements, etc. made within the spirit and principles of the present application shall be included in the protection scope of the present application.

## CLAIMS

1. A multimedia playing system for a cloud desktop, comprising a cloud desktop terminal and a cloud desktop server, the cloud desktop server comprises a cloud desktop protocol server and a multimedia player, the cloud desktop terminal comprises a cloud desktop protocol client, and the cloud desktop protocol server comprises a redirection server;

the redirection server is configured to receive, in accordance with a specified protocol, a undecoded media stream transmitted by a redirection client operating in the multimedia player, wherein the undecoded media stream is acquired by the redirection client from the multimedia player;

the cloud desktop protocol server is configured to transmit the undecoded media stream received by the redirection server to the cloud desktop protocol client;

the cloud desktop protocol client is configured to decode the undecoded media stream and invoke a local player to play a decoded multimedia;

wherein the multimedia player comprises multiple types of players and multiple types of browsers, and the specified protocol is compatible with the multiple types of players and the multiple types of browsers and supports an one-to-many communication mechanism.

2. The system according to claim 1, wherein the specified protocol adopts a websocket communication protocol.

3. The system according to claim 1, wherein the redirection client of the multimedia player is further configured to:

if the multimedia player is any non-customized player, receive an undecoded media stream provided by a player plugin embedded in the multimedia player, wherein the undecoded media stream is an audio stream and a video stream generated by the player plugin performing demultiplexing processing on a to-be-played multimedia source file; and transmit the audio stream and the video stream to the redirection server; or,

if the multimedia player is any browser, receive an undecoded media stream provided by a browser plugin embedded in the multimedia player, wherein the undecoded media stream is a to-be-played multimedia source file downloaded by the browser plugin according to a video

address captured from a current webpage; and transmit the to-be-played multimedia source file to the redirection server; wherein the redirection server is further configured to: perform demultiplexing processing on the to-be-played multimedia source file transmitted by the redirection client of the browser to generate a corresponding audio stream and video stream; or,

if the multimedia player is a customized redirection-specified player, receive an undecoded media stream provided by the redirection-specified player, wherein the undecoded media stream is an audio stream and a video stream generated by the redirection-specified player performing demultiplexing processing on a to-be-played multimedia file; and transmit the audio stream and the video stream to the redirection server; wherein the redirection-specified player is preconfigured with a program logic for acquiring the to-be-played multimedia file and performing demultiplexing processing on the to-be-played multimedia file.

4. The system according to claim 3, wherein the non-customized player preferentially invokes the player plugin, and the player plugin is further configured to: if it is determined that the cloud desktop terminal is unable to support an encoding format corresponding to the to-be-played multimedia source file, abort the demultiplexing and a subsequent operation, and return a fallback instruction to the non-customized player to trigger the non-customized player to perform multimedia playing; or,

the browser preferentially invoke the browser plugin, and the browser plugin is further configured to: if the video address captured from the current webpage does not support file downloading, abort the operation of transmitting the to-be-played multimedia source file to the redirection server, and return a fallback instruction to the browser to trigger the browser to perform multimedia playing; wherein the redirection server is further configured to: after receiving the to-be-played multimedia source file, if it is determined that the cloud desktop terminal is unable to support an encoding format corresponding to the to-be-played multimedia source file, suspend subsequent processing for the to-be-played multimedia source file, and return a fallback instruction to the browser to trigger the browser to perform multimedia playing; or,

the redirection-specified player is further configured to: if it is determined that the cloud desktop terminal is unable to support an encoding format corresponding to the to-be-played

multimedia source file, play the to-be-played multimedia instead of performing the demultiplexing and subsequent processing.

5. The system according to claim 1, wherein the cloud desktop protocol server further comprises a position management component, and the position management component is configured to:

receive position information corresponding to a playback frame provided by the redirection server, wherein the position information corresponding to the playback frame is acquired and provided by the redirection clients of the multimedia player to the redirection server;

capture, via a specified interface, position information corresponding to an application window of the multimedia player and an application window of a further application of the cloud desktop; and

calculate occlusion information of the playback frame according to the position information corresponding to the playback frame and the position information corresponding to each application window;

wherein the cloud desktop protocol server is further configured to: transmit a desktop image of the cloud desktop, and the position information and the occlusion information corresponding to the playback frame to the cloud desktop protocol client;

wherein the cloud desktop protocol client is further configured to: acquire a playback frame of a local player; crop the playback frame according to the occlusion information; splice the cropped playback frame with the desktop image according to the position information of the playback frame; and display the spliced image on the cloud desktop terminal.

6. The system according to claim 5, wherein the redirection client is further configured to:

if the multimedia player is any non-customized player, receive position information corresponding to a playback frame provided by a player plugin embedded in the multimedia player, wherein the position information corresponding to the playback frame is monitored and provided by the multimedia player to the player plugin; or,

if the multimedia player is any browser, receive position information corresponding to a playback frame provided by a browser plugin embedded in the multimedia player, wherein the

position information corresponding to the playback frame is monitored and provided by the browser to the browser plugin; or,

if the multimedia player is a customized redirection-specified player, receive position information corresponding to a playback frame monitored by the redirection-specified player, wherein the redirection-specified player is preconfigured with a program logic for position monitoring and transmitting.

7. The system according to claim 5, wherein the cloud desktop protocol server is further configured to:

acquire a first command stream generated within the multimedia player from the redirection server, wherein the first command stream is acquired and provided by the redirection client of the multimedia player to the redirection server;

transmit the first command stream to the cloud desktop protocol client, to enable the cloud desktop protocol client to perform play controlling on the local player within the cloud desktop terminal in accordance with the first command stream;

receive a second command stream transmitted by the cloud desktop protocol client, wherein the second command stream is monitored by the cloud desktop protocol client from the local player; and

perform play controlling on the multimedia player in accordance with the second command stream.

8. The system according to claim 7, wherein data interaction between the cloud desktop protocol server and the cloud desktop protocol client is performed using serialization and deserialization techniques.

9. The system according to claim 5, wherein the playback frame in the desktop image is displayed as a specified pattern, and the specified pattern comprises a solid white pattern, a solid black pattern, or a mosaic pattern.

10. A multimedia playing method for a cloud desktop, applicable to a cloud desktop server, wherein the cloud desktop server comprises a cloud desktop protocol server and a multimedia player, the cloud desktop protocol server comprises a redirection server; and the method comprises:

receiving, using the redirection server comprised in the cloud desktop protocol server in accordance with a specified protocol, a undecoded media stream transmitted by a redirection client operating in the multimedia player, wherein the undecoded media stream is acquired by the redirection client from the multimedia player; and

transmitting, through the cloud desktop protocol server, the undecoded media stream received by the redirection server to a cloud desktop protocol client of a cloud desktop terminal, to enable the cloud desktop protocol client to decode the undecoded media stream and invoke a local player of the cloud desktop terminal to play a decoded multimedia;

wherein the multimedia player comprises multiple types of players and multiple types of browsers, and the specified protocol is compatible with the multiple types of players and the multiple types of browsers and supports an one-to-many communication mechanism.

11. A multimedia playing method for a cloud desktop, applicable to a cloud desktop terminal, wherein the cloud desktop terminal comprises a cloud desktop protocol client, and the method comprises:

receiving, through the cloud desktop protocol client, a undecoded media stream transmitted by a cloud desktop protocol server of a cloud desktop server, wherein the undecoded media stream is received, by a redirection server of the cloud desktop protocol server in accordance with a specified protocol, from a redirection client operating in a multimedia player, and the undecoded media stream is acquired by the redirection client from the multimedia player;

decoding the undecoded media stream; and

invoking a local player to play a decoded multimedia;

wherein the multimedia player operates in the cloud desktop server, the multimedia player comprises multiple types of players and multiple types of browsers, and the specified protocol is compatible with the multiple types of players and the multiple types of browsers and supports an one-to-many communication mechanism.

12. A cloud desktop server, comprising a memory and a processor;

the memory is configured to store one or more computer instructions;

the processor is coupled to the memory and configured to execute the one or more computer instructions to:

receive, using a redirection server comprised in a cloud desktop protocol server in accordance with a specified protocol, a undecoded media stream transmitted by a redirection client operating in a multimedia player, wherein the undecoded media stream is acquired by the redirection client from the multimedia player; and

transmit, through the cloud desktop protocol server, the undecoded media stream received by the redirection server to a cloud desktop protocol client of a cloud desktop terminal, to enable the cloud desktop protocol client to decode the undecoded media stream and invoke a local player of the cloud desktop terminal to play a decoded multimedia;

wherein the multimedia player operates in the cloud desktop server, the multimedia player comprises multiple types of players and multiple types of browsers, and the specified protocol is compatible with the multiple types of players and the multiple types of browsers and supports an one-to-many communication mechanism.

13. A cloud desktop terminal, comprising a memory and a processor;

the memory is configured to store one or more computer instructions;

the processor is coupled to the memory and configured to execute the one or more computer instructions to:

receive, through a cloud desktop protocol client, a undecoded media stream transmitted by a cloud desktop protocol server of a cloud desktop server, wherein the undecoded media stream is received, by a redirection server of the cloud desktop protocol server in accordance with a specified protocol, from a redirection client operating in a multimedia player, and the undecoded media stream is acquired by the redirection client from the multimedia player;

decode the undecoded media stream; and

invoke a local player to play a decoded multimedia;

wherein the multimedia player operates in the cloud desktop server, the multimedia player comprises multiple types of players and multiple types of browsers, and the specified protocol is compatible with the multiple types of players and the multiple types of browsers and supports an one-to-many communication mechanism.

14. A computer-readable storage medium storing computer instructions which, when executed by one or more processors, cause the one or more processors to execute the multimedia playing method for the cloud desktop according to claim 10 or 11.