

PAGE GENERATION SYSTEM, METHOD, AND APPARATUS

TECHNICAL FIELD

[0001] Embodiments of this specification relate to the field of data processing technologies, and in particular, to a page generation system, method, and apparatus.

BACKGROUND

[0002] With development of Internet technologies, online services are applied to increasingly more scenarios, which can not only make it more convenient for users to purchase products, but also reduce costs of maintaining services by merchants. Therefore, the merchants and the users are more dependent on the online services. With development of the online services, there are increasingly more cross-regional services. For example, overseas shopping and game point card recharging need to be implemented across regions. However, the cross-regional service is usually more limited to a network delay. If the network delay is high, a network delay of pulling page display materials and content from a client to a server can be in an order of hundreds of milliseconds, and it takes a longer time to open a page for establishing an entire link. Consequently, use by users is affected to a great extent, and user loss is indirectly caused. Therefore, an effective solution is urgently needed to resolve the foregoing problem.

SUMMARY

[0003] In view of this, embodiments of this specification provide a page generation system and a page generation method. One or more embodiments of this specification further relate to a page generation apparatus, a computing device, a computer-readable storage medium, and a computer program, to resolve a technical disadvantage in the existing technology.

[0004] According to a first aspect of embodiments of this specification, a page generation system is provided, including a client, a server, and a content delivery network.

[0005] The server is configured to determine first page data corresponding to a target application, and deliver the first page data to the client through the content delivery network.

[0006] The client is configured to cache the first page data in local storage space; send a data obtaining request to the server when receiving a click instruction submitted for the target application,

and receive second page data fed back by the server for the data obtaining request; and generate a target page based on the second page data and the first page data in the local storage space, and display the target page in response to the click instruction.

[0007] Optionally, the server is further configured to obtain page configuration information and page rule information corresponding to the target application; and construct the first page data based on the page configuration information and the page rule information, and deliver the first page data to the client through a content delivery node.

[0008] Optionally, the server is further configured to send the first page data to the content delivery network; and

the content delivery network is configured to determine the client associated with the server, determine a content delivery node based on attribute information of the client, and send the first page data to the client through the content delivery node.

[0009] Optionally, the system further includes a rule processing node, and the rule processing node is configured to obtain the page rule information; and perform classification processing on subpage rule information included in the page rule information, obtain target page rule information based on a classification processing result, and send the target page rule information to the client.

[0010] Optionally, the client is further configured to receive the first page data, and extract rule data and page data from the first page data; and execute a page rule corresponding to the rule data, and store the page data in the local storage space based on an execution result.

[0011] Optionally, the client is further configured to send the data obtaining request to the server when receiving the click instruction submitted for the target application; and

the server is further configured to read, based on the data obtaining request, global page data associated with the target application, and determine a historical data obtaining request corresponding to the data obtaining request; and select local page data from the global page data as the second page data based on the historical data obtaining request, and send the second page data to the client.

[0012] Optionally, the client is further configured to receive the second page data fed back by the server for the data obtaining request; determine target attribute information corresponding to the second page data and reference attribute information corresponding to the first page data, and verify the target attribute information based on the reference attribute information; and when the verification succeeds, generate the target page based on the second page data and the first page data in the local storage space, and display the target page.

[0013] Optionally, the client is further configured to read the first page data from the local storage space; process the first page data and the second page data based on a preset splitting rule, and

generate a page element based on a processing result; and obtain the target page by rendering the page element, and display the target page.

[0014] Optionally, the client is further configured to send a target data obtaining request to the server when the verification fails;

5 the server is further configured to read, for the target data obtaining request, third page data associated with the first page data, and send the third page data to the client; or read target page data for the target data obtaining request, and send the target page data to the client; and

 the client is further configured to generate the target page based on the first page data and the third page data; or generate the target page based on the target page data.

10 **[0015]** According to a second aspect of embodiments of this specification, a page generation method is provided, is applied to a client, and includes:

 receiving first page data delivered by a server for a target application, and storing the first page data in local storage space;

 sending a data obtaining request to the server when receiving a click instruction submitted
15 for the target application, and receiving second page data fed back by the server for the data obtaining request; and

 generating a target page based on the second page data and the first page data in the local storage space, and displaying the target page in response to the click instruction.

20 **[0016]** According to a third aspect of embodiments of this specification, a page generation apparatus is provided, is applied to a client, and includes:

 a data receiving module, configured to receive first page data delivered by a server for a target application, and store the first page data in local storage space;

 a request sending module, configured to send a data obtaining request to the server when
25 receiving a click instruction submitted for the target application, and receive second page data fed back by the server for the data obtaining request; and

 a page generation module, configured to generate a target page based on the second page data and the first page data in the local storage space, and display the target page in response to the click instruction.

30 **[0017]** According to a fourth aspect of embodiments of this specification, a computing device is provided, including:

 a memory and a processor.

[0018] The memory is configured to store computer-executable instructions. When the processor is configured to execute the computer-executable instructions, the steps of the page generation method are implemented.

[0019] According to a fifth aspect of embodiments of this specification, a computer-readable storage medium is provided. The computer-readable storage medium stores computer-executable instructions. When the instructions are executed by a processor, the steps of the page generation method are implemented.

5 [0020] In the page generation system provided in the embodiments, to quickly implement data transmission between the client and the server in any scenario, and increase a speed of page generation on the client, the server can first deliver, to the client through the content delivery network, the first page data that needs to be used for page rendering. The client can locally cache the first page data after receiving the first page data. Subsequently, if receiving the click instruction associated with the
10 target application, the client can request the second page data needed for page rendering from the server, and read the first page data from the local storage space, to integrate the first page data and the second page data to obtain all data used for page rendering, and then generate the target page based on an integration result of the first page data and the second page data, and display the target page. In this way, a page can be rendered on the client without loading more data, to reduce data
15 transmission pressure on the server, and fully use a computing capability of the client, so as to improve page generation efficiency, so that a page can be quickly displayed in response to a user instruction in any scenario, to improve user experience.

BRIEF DESCRIPTION OF DRAWINGS

[0021] FIG. 1 is a schematic diagram of a page generation system according to an embodiment
20 of this specification;

[0022] FIG. 2 is a schematic diagram of a structure of a page generation system according to an embodiment of this specification;

[0023] FIG. 3 is a schematic diagram of a structure of another page generation system according to an embodiment of this specification;

25 [0024] FIG. 4 is a flowchart of a page generation method according to an embodiment of this specification;

[0025] FIG. 5 is a schematic diagram of a structure of a page generation apparatus according to an embodiment of this specification; and

[0026] FIG. 6 is a block diagram of a structure of a computing device according to an embodiment
30 of this specification.

DESCRIPTION OF EMBODIMENTS

[0027] Many specific details are described in the following descriptions to facilitate full understanding of this specification. However, this specification can be implemented in many different manners from those described herein. A person skilled in the art can make similar promotion without departing from the content of this specification. Therefore, this specification is not limited to the specific implementations disclosed below.

[0028] Terms used in one or more embodiments of this specification are merely used to describe specific embodiments, and are not intended to limit the one or more embodiments of this specification. The terms "a" and "the" of singular forms used in one or more embodiments of this specification and the appended claims are also intended to include plural forms, unless otherwise specified in the context clearly. It should be further understood that the term "and/or" used in one or more embodiments of this specification refers to and includes any or all possible combinations of one or more associated listed items.

[0029] It should be understood that although terms "first", "second", and the like can be used in one or more embodiments of this specification to describe various types of information, the information is not limited to these terms. These terms are merely used to differentiate between information of the same type. For example, "first" can also be referred to as "second" without departing from the scope of one or more embodiments of this specification, and similarly, "second" can also be referred to as "first". Depending on the context, the word "if" used herein can be explained as "when", "while", or "in response to determining".

[0030] First, terms used in one or more embodiments of this specification are explained.

[0031] A full name of CDN is content delivery network. A basic idea is to avoid, as much as possible, bottlenecks and links that may affect a data transmission speed and stability on the Internet, so that content is more quickly and stably transmitted. By placing a layer of intelligent virtual networks that are based on the existing Internet and that include node servers in various parts of the network, a CDN system can redirect, in real time based on comprehensive information such as network traffic, connection and load statuses of each node, a distance to the user, and a response time, a request of a user to a service node nearest to the user, so that the user can obtain needed content nearby, to resolve congestion of the Internet network, and increase a response speed of accessing a website by the user.

[0032] This specification provides a page generation system and a page generation method. This specification further relates to a page generation apparatus, a computing device, a computer-readable storage medium, and a computer program, which are described in detail in the following embodiments.

[0033] With reference to a schematic diagram shown in FIG. 1, in the page generation system provided in the embodiments, to quickly implement data transmission between a client and a server in any scenario, and increase a speed of page generation on the client, the server can first deliver, to the client through a content delivery network, first page data that needs to be used for page rendering.

5 The client can locally cache the first page data after receiving the first page data. Subsequently, if receiving a click instruction associated with a target application, the client can request second page data needed for page rendering from the server, and read the first page data from local storage space, to integrate the first page data and the second page data to obtain all data used for page rendering, and then generate a target page based on an integration result of the first page data and the second

10 page data, and display the target page. In this way, a page can be rendered on the client without loading more data, to reduce data transmission pressure on the server, and fully use a computing capability of the client, so as to improve page generation efficiency, so that a page can be quickly displayed in response to a user instruction in any scenario, to improve user experience.

[0034] That is, to improve convenience of using the target application by the user and improve

15 performance of the target application, some static data and dynamic data needed for page generation can be preset, quickly stored in the content delivery network, and delivered to the client of the user before the target application is used by the user. Through cooperation between the client and the server, a part of an original calculation process of the server is delivered to the client for completion, to quickly display the target page in response to the user instruction.

20 [0035] FIG. 2 is a schematic diagram of a structure of a page generation system according to an embodiment of this specification. The page generation system 200 includes a client 210, a server 220, and a content delivery network 230.

[0036] The server 220 is configured to determine first page data corresponding to a target application, and deliver the first page data to the client through the content delivery network 230. The

25 client 210 is configured to cache the first page data in local storage space; send a data obtaining request to the server when receiving a click instruction submitted for the target application, and receive second page data fed back by the server for the data obtaining request; and generate a target page based on the second page data and the first page data in the local storage space, and display the target page in response to the click instruction.

30 [0037] The page generation system provided in this embodiment can be applied to any application that needs page generation processing, for example, a video browsing application, a payment application, a shopping application, an audio playback application, and a drawing application. In addition, the target application can be an application directly installed on the client, or can be an applet embedded in the application. This is not limited in this embodiment herein.

[0038] Specifically, the client is specifically a client on which the target application is installed. The client includes but is not limited to a mobile phone, a computer, a smart wearable device, or the like. This is not limited in this embodiment. Correspondingly, the server is specifically an end that needs to provide a service for a user through the target application. The service provided by the server includes but is not limited to a payment service, a video browsing service, a product purchase service, an audio playback service, or the like. This is not limited in this embodiment herein. Correspondingly, the content delivery network is specifically a CDN.

[0039] Correspondingly, the first page data is specifically page data obtained for the target application, and the page data is merely some data that needs to be used to generate the target page.

The some data is used as a basis for generating the target page, and a change or update period of the some data is relatively long. After being cached in the client, the some data can be used each time the target page is subsequently generated. That is, the first page data can be understood as fixed data that needs to be used to generate the target page, for example, page template data, data corresponding to a control element on the page, data corresponding to a specified icon on the page, and/or splitting data corresponding to an AB test, and is used to generate the target page on the client. Correspondingly, the click instruction is specifically an instruction submitted for the target application. The instruction can be an instruction for starting the application, or can be an instruction for jumping from a page to another page in the target application. Correspondingly, the second page data is specifically another part of data that needs to be used when the target page is generated, and the part of data may be changed when being used to generate the target page. A change or update period of the part of data is relatively short, and the part of data is changed each time the part of data is used. That is, the second page data can be understood as change data needed for the target page, for example, data, on the page, that needs to be changed with reference to push content, order content, and/or payment content. The part of data needs to be downloaded from the server before each page is rendered.

[0040] Based on this, to enable the client to obtain, through rendering in a shorter time, the target page corresponding to the target application, the server can first determine the first page data corresponding to the target application. The page data can be used each time a page is generated. Therefore, the page data can be delivered to the client for storage through the content delivery network. After receiving the page data, the client can store the page data in the local storage space. Subsequently, if receiving the click instruction submitted by the user for the target application, the client can send the data obtaining request to the server based on the click instruction, to download, from the server, data that is modified only for a different operation each time performed on page content, that is, the second page data. Then, the target page can be generated with reference to the

locally stored first page data and the second page data, and displayed to the user through the target application.

[0041] That is, to support the client in quickly generating the target page for display in response to the click instruction, the server can configure page data. That is, the server first configures the page, a column on the page, and related information needed for the AB test for the target application, and to avoid a case in which the page data is sent to a blacklisted user because of full push, can further configure grayscale information. The grayscale information includes but is not limited to a grayscale list, a whitelist, and the like. After information about the associated page is configured, to implement more convenient use by the client and avoid inaccurate calculation of the client, a rule center corresponding to the server can perform classification management and combination on all rules such as a placement rule, an AB test rule, and an establishment rule, to sort all the rules and then deliver all the rules to the client.

[0042] In this case, the client can complete adaptation, placement, AB adaptation, and the like by using the downloaded related page data, generate the target page through integration by reusing the downloaded page data in an application phase and with reference to the page data delivered by the server, and display the target page to the user, to reduce an amount of downloaded data and complete page rendering by using a computing capability of the client, so as to improve page rendering efficiency. The CDN can provide a corresponding resource to implement operations such as data pushing, revocation, deletion, and record management.

[0043] In this embodiment, the page generation system is described by using an example in which the target application is an applet in an app that provides a payment service. For a page generation process in another application scenario, refer to the same or corresponding descriptions in this embodiment. Details are not described herein in this embodiment.

[0044] The user purchases products by using a cross-regional payment applet. Before the user uses the payment applet, considering that an operator of the payment applet may belong to a different region from the user, a problem of slow startup of the applet may exist. This is mainly because an excessively large amount of data needs to be delivered. Therefore, to resolve the problem, the server of the payment applet can first read first page data corresponding to a home page of the payment applet. The first page data includes column data, AB splitting data, grayscale data, rule data, and the like of the home page, and is used to generate content corresponding to a page template on the home page. After the first page data is obtained, the first page data can be delivered to the client through the content delivery network.

[0045] After receiving the page data, the client can cache the page data in a local disk. Subsequently, if the user uses the payment applet, and submits a click instruction for the payment

applet, second page data corresponding to the home page can be requested from the server in response to the click instruction. When determining the second page data, the server determines that the second page data includes at least dynamic data such as payment amount settlement data and merchant data, and then delivers the second page data to the client. In this case, the client integrates the first page data and the second page data to obtain all page data for rendering the home page of the payment applet, and then renders the home page by using a local computing capability, to jump to the home page of the payment applet. Displayed content includes at least information such as a payment amount and a merchant, to help the user perform a payment operation on this basis.

[0046] In the page generation system provided in this embodiment, to quickly implement data transmission between the client and the server in any scenario, and increase a speed of page generation on the client, the server can first deliver, to the client through the content delivery network, the first page data that needs to be used for page rendering. The client can locally cache the first page data after receiving the first page data. Subsequently, if receiving the click instruction associated with the target application, the client can request the second page data needed for page rendering from the server, and read the first page data from the local storage space, to integrate the first page data and the second page data to obtain all data used for page rendering, and then generate the target page based on an integration result of the first page data and the second page data, and display the target page. In this way, a page can be rendered on the client without loading more data, to reduce data transmission pressure on the server, and fully use a computing capability of the client, so as to improve page generation efficiency, so that a page can be quickly displayed in response to a user instruction in any scenario, to improve user experience.

[0047] Further, when the target page is generated, not only data needed for rendering the page is included, but also some execution logic for page display is included, and these rules and configurations are not significantly changed. Therefore, when the first page data is delivered, all of this part of content can be delivered to the client. In this embodiment, the server is further configured to obtain page configuration information and page rule information corresponding to the target application; and construct the first page data based on the page configuration information and the page rule information, and deliver the first page data to the client through a content delivery node.

[0048] Specifically, the page configuration information is specifically information that needs to be used for configuring the target page, and includes but is not limited to the page, a page column, related information of AB, grayscale information, and the like. Correspondingly, the page rule information is specifically information corresponding to a rule that needs to be complied with before and after page rendering and when a user operation is performed, for example, AB splitting information.

- [0049] Based on this, to deliver the data used to render the target page to the client, and avoid a case in which the client needs to request new data from the server each time, the server can first obtain the page configuration information and the page rule information corresponding to the target application; and construct the first page data based on the page configuration information and the page rule information, and deliver the first page data to the client through the content delivery node.
- 5 [0050] For example, in the payment service applet, to quickly display a payment page to the user, the server can first obtain a page column of the payment page, related information about AB, grayscale information, and the like, determine a corresponding rule for rendering the page, form the page data based on the foregoing content, and send the page data to the client, to pre-store the page data on the client, so that the user can directly read the page data for use.
- 10 [0051] In conclusion, the first page data is constructed with reference to the page configuration information and the page rule information, and delivered to the client, so that the client can generate the target page after requesting some data based on the first page data, to improve generation efficiency of the target page.
- 15 [0052] Further, when the server delivers the first page data through the content delivery network, to implement load balancing and cache the first page data for all clients with a requirement, a delivery node corresponding to the client can be selected with reference to attribute information. In this embodiment, the server is further configured to send the first page data to the content delivery network; and the content delivery network is configured to determine the client associated with the server, determine a content delivery node based on attribute information of the client, and send the first page data to the client through the content delivery node.
- 20 [0053] Specifically, the attribute information is specifically a basic attribute corresponding to the client, and includes but is not limited to information such as an IP address of the client, a client identifier, and a client login account.
- 25 [0054] Based on this, after the server sends the first page data to the content delivery network, to implement data delivery nearby, the content delivery network can first determine the client associated with the server, select the content delivery node based on the attribute information of the client, and finally send the first page data to the client through the content delivery node.
- [0055] In conclusion, the first page data is delivered by selecting the network delivery node by the network delivery network, so that data delivery processing can be quickly and efficiently completed, and the first page data can be cached before the user uses the target application, to subsequently quickly perform page rendering processing in response to a user operation.
- 30 [0056] In addition, considering that there are many rules associated with the target page, if all of the rules are directly delivered to the client, and the client independently performs classification and

management, calculation pressure of the client may be caused. Therefore, classification and management can be completed by a rule processing node before delivery. In this embodiment, the system further includes the rule processing node, and the rule processing node is configured to obtain the page rule information; and perform classification processing on subpage rule information included in the page rule information, obtain target page rule information based on a classification processing result, and send the target page rule information to the client.

[0057] Specifically, the rule processing node is specifically a node that provides a rule combination management service, and is subordinate to the server. Correspondingly, the subpage rule information is specifically description information corresponding to a subpage rule included in the page rule information. Correspondingly, the classification processing is specifically an operation of performing combination processing on rules of a same type. Correspondingly, the target page rule information is specifically a result obtained after classification processing is performed, and is used to be inserted into the first page data to replace the page rule information, to feed back the sorted first page data to the client.

[0058] Based on this, to implement preparation processing before rule delivery, sorting and combination of the rules can be implemented by using the rule processing node subordinate to the server. That is, the rule processing node can first obtain the page rule information; and then perform classification processing on the subpage rule information included in the page rule information, to obtain the target page rule information based on the classification processing result, and send the target page rule information to the client, to facilitate subsequent processing by the client.

[0059] In the forgoing example, to implement that when the related information needed for the payment page is sent to the client, resources of the client can be saved to complete page rendering, the rule processing node can classify and sort the page rule information, and a result obtained after sorting is sent to the client through the content delivery node, to facilitate use by the client.

[0060] In conclusion, the rule processing node classifies the subpage rule information, so that rules in association with each other or similar rules can be integrated together, and delivered to the client, to make it more convenient for the client to use, and save computing resources.

[0061] Further, after the client receives the first page data, considering that the first page data includes not only rule data but also page data, before storage, a rule corresponding to the rule data needs to be first executed to adapt to the new rule in the program, to facilitate subsequent use. In this embodiment, the client is further configured to receive the first page data, and extract rule data and page data from the first page data; and execute a page rule corresponding to the rule data, and store the page data in the local storage space based on an execution result.

[0062] Specifically, for descriptions of the rule data and the page data, refer to the same or corresponding descriptions in the foregoing embodiment. This is not limited in this embodiment herein. Correspondingly, the executing a page rule corresponding to the rule data means that the rule takes effect locally on the client, and is used when the target page is generated, and a generation environment of the target page is provided.

[0063] Based on this, after receiving the first page data delivered by the server, the client can extract the rule data and the page data from the first page data; and execute the page rule corresponding to the rule data, to make a rule for rendering the target page take effect locally on the client, so that the page data is stored in the local storage space based on the execution result, and used in the application phase.

[0064] In the foregoing example, after downloading the page data corresponding to the payment page, the client can first run a page rule in the page data to provide a running environment for rendering the payment page, and then write remaining data into the local disk based on an execution result, so that in a rendering stage, the data can be directly read from the disk for page rendering.

[0065] In conclusion, the page rule is locally executed, and the page data is stored in the local storage space based on the rule execution result, so that a latest page processing or generation rule can be run on the client, and the client can quickly generate the target page.

[0066] Further, when the client requests the second page data, considering that the server stores all the data used for page rendering, and the client already stores some data, namely, the first page data, at a current moment, the server can deliver only some needed page data when delivering the second page data. In this embodiment, the client is further configured to send the data obtaining request to the server when receiving the click instruction submitted for the target application; and the server is further configured to read, based on the data obtaining request, global page data associated with the target application, and determine a historical data obtaining request corresponding to the data obtaining request; and select local page data from the global page data as the second page data based on the historical data obtaining request, and send the second page data to the client.

[0067] Specifically, the global page data is specifically all page data used for rendering the target page, and includes all data related to the page and rule. Correspondingly, the historical data obtaining request is specifically a data obtaining request with a shortest time interval from the data obtaining request, and the data obtaining request is the same as the data obtaining request.

[0068] Based on this, considering that the client already caches the first page data, the server needs to deliver only the page data that needs to be used by the client. Therefore, the client can send the data obtaining request to the server when receiving the click instruction submitted for the target application. In this case, the server can read, based on the data obtaining request, the global page data

associated with the target application, to determine all the data that can be used to generate the target page. To reduce transmission of redundant data and repeated data transmission, the server can simultaneously determine the historical data obtaining request corresponding to the data obtaining request; and select the local page data from the global page data as the second page data based on the historical data obtaining request, and send the second page data to the client for use.

[0069] In the foregoing example, when the user enters the payment page by using the client, a data obtaining request can be first sent to the server. In this case, the server first reads all data of the payment page, then selects data that is not downloaded by the client from all the data as local page data, and then sends the local page data to the client. In this case, the client can render the payment page for use by the user based on the local page data downloaded at the current time and the page data stored in the disk.

[0070] In conclusion, when the second page data is fed back to the client, to avoid repeated data transmission, the local page data can be selected from the global page data in response to the historical data obtaining request, and fed back to the client, so that the client can complete page rendering by loading less data, to improve page generation efficiency.

[0071] In addition, considering that a page of the target application can be adjusted based on a page change, if the delivered second page data cannot be fused with the first page data to generate the target page, the target application cannot be supported in providing a service for the user. Therefore, after the second page data is received, verification can be performed. In this embodiment, the client is further configured to receive the second page data fed back by the server for the data obtaining request; determine target attribute information corresponding to the second page data and reference attribute information corresponding to the first page data, and verify the target attribute information based on the reference attribute information; and when the verification succeeds, generate the target page based on the second page data and the first page data in the local storage space, and display the target page.

[0072] When the verification fails, a target data obtaining request is sent to the server. The server is further configured to read, for the target data obtaining request, third page data associated with the first page data, and send the third page data to the client; or read target page data for the target data obtaining request, and send the target page data to the client. The client is further configured to generate the target page based on the first page data and the third page data; or generate the target page based on the target page data.

[0073] Specifically, the target attribute information is specifically attribute description information corresponding to the second page data, and includes but is limited to a data version number, a data size, a data type, and the like. Correspondingly, the reference attribute information is

specifically attribute description information corresponding to the first page data, and includes but is not limited to a data version number, a data size, a data type, and the like. Verification is to detect whether the first page data and the second page data are complementary page data, and whether the version numbers are the same.

5 **[0074]** Based on this, the client first receives the second page data fed back by the server for the data obtaining request; and in this case, can first obtain the target attribute information corresponding to the second page data and the reference attribute information corresponding to the first page data, and verify the target attribute information based on the reference attribute information; and verify whether the second page data can be complementary to the first page data.

10 **[0075]** When the verification succeeds, it indicates that the second page data delivered by the server is complementary to the first page data, and can be fused to generate the target page. Therefore, the step of generating the target page based on the second page data and the first page data in the local storage space and displaying the target page can be performed.

15 **[0076]** If the verification fails, it indicates that the second page data delivered by the server is not complementary to the first page data, and the target page cannot be generated. To provide a corresponding service for the user by using the target application, the client can send the target data obtaining request to the server. After receiving the request, the server can first read, for the target data obtaining request, the third page data associated with the first page data, and send the third page data to the client, to read the third page data that can be complementary to the first page data. In this
20 process, the server reads the third page data based on information such as the version number of the first page data, and then feeds back the third page data to the client, so that the client can generate the target page based on the first page data and the third page data.

25 **[0077]** In addition, there may be a case in which the first page data cached by the client cannot be used, and the target page cannot be generated even if page data of any version occurs. Therefore, the target page data for generating the target page can be directly read for the target data obtaining request, and sent to the client. In this case, the client can generate the target page by using only the target page data.

30 **[0078]** In the foregoing example, to avoid a case in which the payment page cannot be rendered based on the local page data and with reference to the page data in the disk because of a data version update, after the local page data is downloaded, the locally stored page data can be verified by using the local page data, that is, it can be verified whether versions of the locally stored page data and the local page data are the same, or it can be verified whether the locally stored page data and the local page data are complementary to each other. When the verification succeeds, the payment page can be

rendered with reference to the two parts of data. When the verification fails, all page data can be re-requested, to avoid a problem that the payment page cannot be rendered for use by the user.

[0079] In conclusion, to ensure that the client generates the target page and displays the target page to the user, verification can be performed before generation. In addition, when the verification fails, page data that needs to be used by the client is fed back to the client with reference to a different page data delivery policy, so that the client can generate the target page and provide a service corresponding to the target application for the user.

[0080] When generating the target page, the client is further configured to read the first page data from the local storage space; process the first page data and the second page data based on a preset splitting rule, and generate a page element based on a processing result; and obtain the target page by rendering the page element, and display the target page.

[0081] Specifically, the page element is specifically an element that needs to be used to generate the target page, for example, a page control, a link, a picture, and a video. Correspondingly, the splitting rule is specifically an AB splitting rule. Based on this, when rendering the page, the client can read the first page data from the local storage space; then process the first page data and the second page data based on the preset splitting rule, and generate the page element based on the processing result; and finally can obtain the target page by rendering the page element, and display the target page.

[0082] With reference to a schematic diagram shown in FIG. 3, in an actual service scenario, a deployment platform performs different processing in different phases in a configuration phase. For example, in a CDN push management phase, integration of near-end rules {grayscale, AB, end feature rules, scheduling, multi-language} is performed. In addition, near-end content further needs to include atomic content. A near-end policy needs to include a switch, a real-time offline policy, an image cache policy, and the like. After sorting is completed, release is performed. A release phase can be divided into release advance and release rollback. A data management phase includes page synchronization, spatial synchronization, and the like, to complete preliminary preparation work through configuration. After the content is uploaded, file storage and a construction service are updated to the CDN through coverage by using marmot. The CDN is configured to store a rule, data, a data version number, and a switch. In a research and development phase, a research and development platform implements corresponding function support through cloud construction and ACI orchestration, to implement data synchronization. An applet platform presets data to a C-end based on the construction service and a release service. In a running phase, the C-end receives an iQuic near-end packet, parses the packet to obtain data and a rule, and locally stores the data and the rule, including but not limited to a data version number, a circle rule, a scheduling rule, a multi-

language rule, a grayscale rule, and an orientation rule. In an offline state, end real-time rendering or end offline rendering can be implemented. In a page update phase, Saas established in the industry can perform cache synchronization, cache decision, wrapper extension, or the like to the near end, to support page query and column query, and the content is pulled to the near end in real time.

5 **[0083]** In the page generation system provided in this embodiment, to quickly implement data transmission between the client and the server in any scenario, and increase a speed of page generation on the client, the server can first deliver, to the client through the content delivery network, the first page data that needs to be used for page rendering. The client can locally cache the first page data after receiving the first page data. Subsequently, if receiving the click instruction associated with the
10 target application, the client can request the second page data needed for page rendering from the server, and read the first page data from the local storage space, to integrate the first page data and the second page data to obtain all data used for page rendering, and then generate the target page based on an integration result of the first page data and the second page data, and display the target page. In this way, a page can be rendered on the client without loading more data, to reduce data
15 transmission pressure on the server, and fully use a computing capability of the client, so as to improve page generation efficiency, so that a page can be quickly displayed in response to a user instruction in any scenario, to improve user experience.

[0084] FIG. 4 is a flowchart of a page generation method according to an embodiment of this specification. The method is applied to a client and specifically includes the following steps.

20 **[0085]** Step S402: Receive first page data delivered by a server for a target application, and store the first page data in local storage space.

[0086] Step S404: Send a data obtaining request to the server when receiving a click instruction submitted for the target application, and receive second page data fed back by the server for the data obtaining request.

25 **[0087]** Step S406: Generate a target page based on the second page data and the first page data in the local storage space, and display the target page in response to the click instruction.

[0088] Further, when the target page is generated, not only data needed for rendering the page is included, but also some execution logic for page display is included, and these rules and configurations are not significantly changed. Therefore, when the first page data is delivered, all of
30 this part of content can be delivered to the client. In this embodiment, the server is further configured to obtain page configuration information and page rule information corresponding to the target application; and construct the first page data based on the page configuration information and the page rule information, and deliver the first page data to the client through a content delivery node.

[0089] Specifically, the page configuration information is specifically information that needs to be used for configuring the target page, and includes but is not limited to the page, a page column, related information of AB, grayscale information, and the like. Correspondingly, the page rule information is specifically information corresponding to a rule that needs to be complied with before and after page rendering and when a user operation is performed, for example, AB splitting information.

[0090] Based on this, to deliver the data used to render the target page to the client, and avoid a case in which the client needs to request new data from the server each time, the server can first obtain the page configuration information and the page rule information corresponding to the target application; and construct the first page data based on the page configuration information and the page rule information, and deliver the first page data to the client through the content delivery node.

[0091] In conclusion, the first page data is constructed with reference to the page configuration information and the page rule information, and delivered to the client, so that the client can generate the target page after requesting some data based on the first page data, to improve generation efficiency of the target page.

[0092] Further, when the server delivers the first page data through the content delivery network, to implement load balancing and cache the first page data for all clients with a requirement, a delivery node corresponding to the client can be selected with reference to attribute information. In this embodiment, the server is further configured to send the first page data to the content delivery network; and the content delivery network is configured to determine the client associated with the server, determine a content delivery node based on attribute information of the client, and send the first page data to the client through the content delivery node.

[0093] Specifically, the attribute information is specifically a basic attribute corresponding to the client, and includes but is not limited to information such as an IP address of the client, a client identifier, and a client login account. Based on this, after the server sends the first page data to the content delivery network, to implement data delivery nearby, the content delivery network can first determine the client associated with the server, select the content delivery node based on the attribute information of the client, and finally send the first page data to the client through the content delivery node.

[0094] In conclusion, the first page data is delivered by selecting the network delivery node by the network delivery network, so that data delivery processing can be quickly and efficiently completed, and the first page data can be cached before the user uses the target application, to subsequently quickly perform page rendering processing in response to a user operation.

[0095] In addition, considering that there are many rules associated with the target page, if all of the rules are directly delivered to the client, and the client independently performs classification and management, calculation pressure of the client may be caused. Therefore, classification and management can be completed by a rule processing node before delivery. In this embodiment, the system further includes the rule processing node, and the rule processing node is configured to obtain the page rule information; and perform classification processing on subpage rule information included in the page rule information, obtain target page rule information based on a classification processing result, and send the target page rule information to the client.

[0096] Specifically, the rule processing node is specifically a node that provides a rule combination management service, and is subordinate to the server. Correspondingly, the subpage rule information is specifically description information corresponding to a subpage rule included in the page rule information. Correspondingly, the classification processing is specifically an operation of performing combination processing on rules of a same type. Correspondingly, the target page rule information is specifically a result obtained after classification processing is performed, and is used to be inserted into the first page data to replace the page rule information, to feed back the sorted first page data to the client.

[0097] Based on this, to implement preparation processing before rule delivery, sorting and combination of the rules can be implemented by using the rule processing node subordinate to the server. That is, the rule processing node can first obtain the page rule information; and then perform classification processing on the subpage rule information included in the page rule information, to obtain the target page rule information based on the classification processing result, and send the target page rule information to the client, to facilitate subsequent processing by the client.

[0098] In conclusion, the rule processing node classifies the subpage rule information, so that rules in association with each other or similar rules can be integrated together, and delivered to the client, to make it more convenient for the client to use, and save computing resources.

[0099] Further, after the client receives the first page data, considering that the first page data includes not only rule data but also page data, before storage, a rule corresponding to the rule data needs to be first executed to adapt to the new rule in the program, to facilitate subsequent use. In this embodiment, the client is further configured to receive the first page data, and extract rule data and page data from the first page data; and execute a page rule corresponding to the rule data, and store the page data in the local storage space based on an execution result.

[0100] Specifically, for descriptions of the rule data and the page data, refer to the same or corresponding descriptions in the foregoing embodiment. This is not limited in this embodiment herein. Correspondingly, the executing a page rule corresponding to the rule data means that the rule

takes effect locally on the client, and is used when the target page is generated, and a generation environment of the target page is provided.

[0101] Based on this, after receiving the first page data delivered by the server, the client can extract the rule data and the page data from the first page data; and execute the page rule corresponding to the rule data, to make a rule for rendering the target page take effect locally on the client, so that the page data is stored in the local storage space based on the execution result, and used in the application phase.

[0102] In conclusion, the page rule is locally executed, and the page data is stored in the local storage space based on the rule execution result, so that a latest page processing or generation rule can be run on the client, and the client can quickly generate the target page.

[0103] Further, when the client requests the second page data, considering that the server stores all the data used for page rendering, and the client already stores some data, namely, the first page data, at a current moment, the server can deliver only some needed page data when delivering the second page data. In this embodiment, the client is further configured to send the data obtaining request to the server when receiving the click instruction submitted for the target application; and the server is further configured to read, based on the data obtaining request, global page data associated with the target application, and determine a historical data obtaining request corresponding to the data obtaining request; and select local page data from the global page data as the second page data based on the historical data obtaining request, and send the second page data to the client.

[0104] Specifically, the global page data is specifically all page data used for rendering the target page, and includes all data related to the page and rule. Correspondingly, the historical data obtaining request is specifically a data obtaining request with a shortest time interval from the data obtaining request, and the data obtaining request is the same as the data obtaining request.

[0105] Based on this, considering that the client already caches the first page data, the server needs to deliver only the page data that needs to be used by the client. Therefore, the client can send the data obtaining request to the server when receiving the click instruction submitted for the target application. In this case, the server can read, based on the data obtaining request, the global page data associated with the target application, to determine all the data that can be used to generate the target page. To reduce transmission of redundant data and repeated data transmission, the server can simultaneously determine the historical data obtaining request corresponding to the data obtaining request; and select the local page data from the global page data as the second page data based on the historical data obtaining request, and send the second page data to the client for use.

[0106] In conclusion, when the second page data is fed back to the client, to avoid repeated data transmission, the local page data can be selected from the global page data in response to the historical

data obtaining request, and fed back to the client, so that the client can complete page rendering by loading less data, to improve page generation efficiency.

[0107] In addition, considering that a page of the target application can be adjusted based on a page change, if the delivered second page data cannot be fused with the first page data to generate the target page, the target application cannot be supported in providing a service for the user. Therefore, after the second page data is received, verification can be performed. In this embodiment, the client is further configured to receive the second page data fed back by the server for the data obtaining request; determine target attribute information corresponding to the second page data and reference attribute information corresponding to the first page data, and verify the target attribute information based on the reference attribute information; and

when the verification succeeds, generate the target page based on the second page data and the first page data in the local storage space, and display the target page.

[0108] When the verification fails, a target data obtaining request is sent to the server. The server is further configured to read, for the target data obtaining request, third page data associated with the first page data, and send the third page data to the client; or read target page data for the target data obtaining request, and send the target page data to the client. The client is further configured to generate the target page based on the first page data and the third page data; or generate the target page based on the target page data.

[0109] Specifically, the target attribute information is specifically attribute description information corresponding to the second page data, and includes but is limited to a data version number, a data size, a data type, and the like. Correspondingly, the reference attribute information is specifically attribute description information corresponding to the first page data, and includes but is not limited to a data version number, a data size, a data type, and the like. Verification is to detect whether the first page data and the second page data are complementary page data, and whether the version numbers are the same.

[0110] Based on this, the client first receives the second page data fed back by the server for the data obtaining request; and in this case, can first obtain the target attribute information corresponding to the second page data and the reference attribute information corresponding to the first page data, and verify the target attribute information based on the reference attribute information; and verify whether the second page data can be complementary to the first page data.

[0111] When the verification succeeds, it indicates that the second page data delivered by the server is complementary to the first page data, and can be fused to generate the target page. Therefore, the step of generating the target page based on the second page data and the first page data in the local storage space and displaying the target page can be performed.

- 5 **[0112]** If the verification fails, it indicates that the second page data delivered by the server is not complementary to the first page data, and the target page cannot be generated. To provide a corresponding service for the user by using the target application, the client can send the target data obtaining request to the server. After receiving the request, the server can first read, for the target data obtaining request, the third page data associated with the first page data, and send the third page data to the client, to read the third page data that can be complementary to the first page data. In this process, the server reads the third page data based on information such as the version number of the first page data, and then feeds back the third page data to the client, so that the client can generate the target page based on the first page data and the third page data.
- 10 **[0113]** In addition, there may be a case in which the first page data cached by the client cannot be used, and the target page cannot be generated even if page data of any version occurs. Therefore, the target page data for generating the target page can be directly read for the target data obtaining request, and sent to the client. In this case, the client can generate the target page by using only the target page data.
- 15 **[0114]** In conclusion, to ensure that the client generates the target page and displays the target page to the user, verification can be performed before generation. In addition, when the verification fails, page data that needs to be used by the client is fed back to the client with reference to a different page data delivery policy, so that the client can generate the target page and provide a service corresponding to the target application for the user.
- 20 **[0115]** When generating the target page, the client is further configured to read the first page data from the local storage space; process the first page data and the second page data based on a preset splitting rule, and generate a page element based on a processing result; and obtain the target page by rendering the page element, and display the target page.
- 25 **[0116]** Specifically, the page element is specifically an element that needs to be used to generate the target page, for example, a page control, a link, a picture, and a video. Correspondingly, the splitting rule is specifically an AB splitting rule. Based on this, when rendering the page, the client can read the first page data from the local storage space; then process the first page data and the second page data based on the preset splitting rule, and generate the page element based on the processing result; and finally can obtain the target page by rendering the page element, and display
- 30 the target page.
- [0117]** In conclusion, to quickly implement data transmission between the client and the server in any scenario, and increase a speed of page generation on the client, the server can first deliver, to the client through the content delivery network, the first page data that needs to be used for page rendering. The client can locally cache the first page data after receiving the first page data.

Subsequently, if receiving the click instruction associated with the target application, the client can request the second page data needed for page rendering from the server, and read the first page data from the local storage space, to integrate the first page data and the second page data to obtain all data used for page rendering, and then generate the target page based on an integration result of the first
5 page data and the second page data, and display the target page. In this way, a page can be rendered on the client without loading more data, to reduce data transmission pressure on the server, and fully use a computing capability of the client, so as to improve page generation efficiency, so that a page can be quickly displayed in response to a user instruction in any scenario, to improve user experience.

[0118] Corresponding to the foregoing method embodiments, this specification further provides
10 an embodiment of a page generation apparatus. FIG. 5 is a schematic diagram of a structure of a page generation apparatus according to an embodiment of this specification. As shown in FIG. 5, the apparatus is applied to a client, including:

a data receiving module 502, configured to receive first page data delivered by a server for a target application, and store the first page data in local storage space;

15 a request sending module 504, configured to send a data obtaining request to the server when receiving a click instruction submitted for the target application, and receive second page data fed back by the server for the data obtaining request; and

a page generation module 506, configured to generate a target page based on the second page data and the first page data in the local storage space, and display the target page in response to
20 the click instruction.

[0119] In conclusion, to quickly implement data transmission between the client and the server in any scenario, and increase a speed of page generation on the client, the server can first deliver, to the client through a content delivery network, the first page data that needs to be used for page rendering. The client can locally cache the first page data after receiving the first page data.

25 Subsequently, if receiving the click instruction associated with the target application, the client can request the second page data needed for page rendering from the server, and read the first page data from the local storage space, to integrate the first page data and the second page data to obtain all data used for page rendering, and then generate the target page based on an integration result of the first page data and the second page data, and display the target page. In this way, a page can be rendered
30 on the client without loading more data, to reduce data transmission pressure on the server, and fully use a computing capability of the client, so as to improve page generation efficiency, so that a page can be quickly displayed in response to a user instruction in any scenario, to improve user experience.

[0120] The schematic solution for the page generation apparatus in the embodiments is described above. It should be noted that the technical solutions of the page generation apparatus are of the same

concept as the technical solutions of the foregoing page generation method. For details that are not described in detail in the technical solutions of the page generation apparatus, refer to the descriptions of the technical solutions of the foregoing page generation method.

[0121] FIG. 6 is a block diagram of a structure of a computing device 600 according to an embodiment of this specification. Components of the computing device 600 include but are not limited to a memory 610 and a processor 620. The processor 620 and the memory 610 are connected by using a bus 630, and a database 650 is configured to store data.

[0122] The computing device 600 further includes an access device 640. The access device 640 enables the computing device 600 to communicate through one or more networks 660. Examples of these networks include a public switched telephone network (PSTN), a local area network (LAN), a wide area network (WAN), a personal area network (PAN), or a combination of communication networks such as the Internet. The access device 640 can include one or more of any type of wired or wireless network interfaces (for example, a network interface controller (NIC)), for example, an IEEE 802.11 wireless local area network (WLAN) wireless interface, a worldwide interoperability for microwave access (Wi-MAX) interface, an Ethernet interface, a universal serial bus (USB) interface, a cellular network interface, a Bluetooth interface, and a near field communication (NFC) interface.

[0123] In an embodiment of this application, the foregoing components of the computing device 600 and other components not shown in FIG. 6 can alternatively be connected to each other, for example, by using a bus. It should be understood that the block diagram of the structure of the computing device shown in FIG. 6 is for an illustrative purpose only and is not intended to limit the scope of this application. A person skilled in the art can add or replace other components as needed.

[0124] The computing device 600 can be any type of stationary or mobile computing device, including a mobile computer or a mobile computing device (for example, a tablet computer, a personal digital assistant, a laptop computer, a notebook computer, or a netbook), a mobile phone (for example, a smartphone), a wearable computing device (for example, a smartwatch or smart glasses), or another type of mobile device, or a stationary computing device such as a desktop computer or a personal computer (PC). The computing device 600 can alternatively be a mobile or stationary server.

[0125] The processor 620 is configured to execute the following computer-executable instructions. When the computer-executable instructions are executed by the processor, the following method is implemented:

receiving first page data delivered by a server for a target application, and storing the first page data in local storage space;

sending a data obtaining request to the server when receiving a click instruction submitted for the target application, and receiving second page data fed back by the server for the data obtaining request; and

generating a target page based on the second page data and the first page data in the local storage space, and displaying the target page in response to the click instruction.

[0126] The schematic solution for the computing device in the embodiments is described above. It should be noted that the technical solutions of the computing device are of the same concept as the technical solutions of the foregoing page generation method. For details that are not described in detail in the technical solutions of the computing device, refer to the descriptions of the technical solutions of the foregoing page generation method.

[0127] An embodiment of this specification further provides a computer-readable storage medium. The computer-readable storage medium stores computer-executable instructions. When the computer-executable instructions are executed by a processor, the following method is implemented:

receiving first page data delivered by a server for a target application, and storing the first page data in local storage space;

sending a data obtaining request to the server when receiving a click instruction submitted for the target application, and receiving second page data fed back by the server for the data obtaining request; and

generating a target page based on the second page data and the first page data in the local storage space, and displaying the target page in response to the click instruction.

[0128] The schematic solution for the computer-readable storage medium in the embodiments is described above. It should be noted that the technical solutions of the storage medium are of the same concept as the technical solutions of the foregoing page generation method. For details that are not described in detail in the technical solutions of the storage medium, refer to the descriptions of the technical solutions of the foregoing page generation method.

[0129] Specific embodiments of this specification are described above. Other embodiments fall within the scope of the appended claims. In some cases, the actions or steps described in the claims can be performed in a sequence different from that in the embodiments and desired results can still be achieved. In addition, the process depicted in the accompanying drawings does not necessarily need a particular sequence to achieve the desired results. In some implementations, multi-tasking and parallel processing are feasible or may be advantageous.

[0130] The computer instructions include computer program code, and the computer program code can be in a source code form, an object code form, an executable file form, some intermediate forms, or the like. The computer-readable medium can include any entity or apparatus that can carry

the computer program code, a recording medium, a USB flash drive, a removable hard disk, a magnetic disk, an optical disc, a computer memory, a read-only memory (ROM), a random access memory (RAM), an electrical carrier signal, a telecommunications signal, a software distribution medium, or the like. It should be noted that the content included in the computer-readable medium
5 can be appropriately increased or decreased in accordance with the requirements of the legislation and patent practice in the jurisdiction, for example, in some jurisdictions, the computer-readable medium does not include electrical carrier signals or telecommunications signals in accordance with the legislation and patent practice.

[0131] It should be noted that for brief description, the foregoing method embodiments are
10 represented as a series of actions. However, a person skilled in the art should understand that the embodiments of this specification are not limited to the described sequence of the actions, because according to the embodiments of this specification, some steps can be performed in another sequence or simultaneously. In addition, a person skilled in the art should also understand that all the embodiments described in this specification are example embodiments, and the related actions and
15 modules are not necessarily mandatory to the embodiments of this specification.

[0132] In the foregoing embodiments, the descriptions of the embodiments have respective focuses. For a part that is not described in detail in an embodiment, refer to related descriptions in other embodiments.

[0133] The example embodiments of this specification disclosed above are intended merely to
20 help describe this specification. The optional embodiments do not describe all the details in detail, and do not limit the present invention to the specific implementations described. Clearly, many modifications and changes can be made based on the content of the embodiments of this specification. These embodiments are selected and described in detail in this specification to better explain the principles and practical applications of the embodiments of this specification, so that a person skilled
25 in the art can well understand and use this specification. This specification is limited only to the claims and the full scope and equivalents thereof.

CLAIMS

What is claimed is:

1. A page generation system, comprising a client, a server, and a content delivery network, wherein

5 the server is configured to determine first page data corresponding to a target application, and deliver the first page data to the client through the content delivery network; and

the client is configured to cache the first page data in local storage space; send a data obtaining request to the server when receiving a click instruction submitted for the target application, and receive second page data fed back by the server for the data obtaining request; 10 and generate a target page based on the second page data and the first page data in the local storage space, and display the target page in response to the click instruction.

2. The system according to claim 1, wherein the server is further configured to obtain page configuration information and page rule information corresponding to the target application; and construct the first page data based on the page configuration information and the page rule 15 information, and deliver the first page data to the client through a content delivery node.

3. The system according to claim 1, wherein the server is further configured to send the first page data to the content delivery network; and

the content delivery network is configured to determine the client associated with the server, determine a content delivery node based on attribute information of the client, and send 20 the first page data to the client through the content delivery node.

4. The system according to claim 2, wherein the system further comprises a rule processing node, and the rule processing node is configured to obtain the page rule information; and perform classification processing on subpage rule information comprised in the page rule information, obtain target page rule information based on a classification processing result, and 25 send the target page rule information to the client.

5. The system according to claim 1, wherein the client is further configured to receive the first page data, and extract rule data and page data from the first page data; and execute a page rule corresponding to the rule data, and store the page data in the local storage space based on an execution result.

6. The system according to claim 1, wherein the client is further configured to send the data obtaining request to the server when receiving the click instruction submitted for the target application; and 30

the server is further configured to read, based on the data obtaining request, global page data associated with the target application, and determine a historical data obtaining request

corresponding to the data obtaining request; and select local page data from the global page data as the second page data based on the historical data obtaining request, and send the second page data to the client.

7. The system according to claim 1, wherein the client is further configured to receive the second page data fed back by the server for the data obtaining request; determine target attribute information corresponding to the second page data and reference attribute information corresponding to the first page data, and verify the target attribute information based on the reference attribute information; and when the verification succeeds, generate the target page based on the second page data and the first page data in the local storage space, and display the target page.

8. The system according to any one of claims 1 to 7, wherein the client is further configured to read the first page data from the local storage space; process the first page data and the second page data based on a preset splitting rule, and generate a page element based on a processing result; and obtain the target page by rendering the page element, and display the target page.

9. The system according to claim 7, wherein the client is further configured to send a target data obtaining request to the server when the verification fails;

the server is further configured to read, for the target data obtaining request, third page data associated with the first page data, and send the third page data to the client; or read target page data for the target data obtaining request, and send the target page data to the client; and

the client is further configured to generate the target page based on the first page data and the third page data; or generate the target page based on the target page data.

10. A page generation method, applied to a client, comprising:

receiving first page data delivered by a server for a target application, and storing the first page data in local storage space;

sending a data obtaining request to the server when receiving a click instruction submitted for the target application, and receiving second page data fed back by the server for the data obtaining request; and

generating a target page based on the second page data and the first page data in the local storage space, and displaying the target page in response to the click instruction.

11. A page generation apparatus, applied to a client, comprising:

a data receiving module, configured to receive first page data delivered by a server for a target application, and store the first page data in local storage space;

a request sending module, configured to send a data obtaining request to the server when receiving a click instruction submitted for the target application, and receive second page data

fed back by the server for the data obtaining request; and

a page generation module, configured to generate a target page based on the second page data and the first page data in the local storage space, and display the target page in response to the click instruction.

5 12. A computing device, comprising:

a memory and a processor, wherein

the memory is configured to store computer-executable instructions, the processor is configured to execute the computer-executable instructions, and when the computer-executable instructions are executed by the processor, the steps of the method according to claim 10 are

10 implemented.

13. A computer-readable storage medium, wherein the computer-readable storage medium stores computer-executable instructions, and when the computer-executable instructions are executed by a processor, the steps of the method according to claim 10 are implemented.