

DATA ACCESS METHOD AND APPARATUS

This application claims priority to Chinese Patent Application No. 202311605963.X, filed with the China National Intellectual Property Administration on November 28, 2023, entitled “DATA
5 ACCESS METHOD AND APPARATUS,” the entire contents of which are incorporated herein by reference.

TECHNICAL FIELD

The present application relates to the field of computer technology, and in particular, to a data access method and apparatus.

10 BACKGROUND

Currently, a runtime system based on a WebAssembly virtual machine can be deployed in resource-constrained chips or devices. The runtime system based on the WebAssembly virtual machine can provide a runtime environment for applications, and the WebAssembly virtual machine interprets and executes the bytecode of the applications.

15 During the installation or runtime phase of an application, the application can request a data storage space belonging to the application from the runtime system by calling interfaces provided in the runtime environment corresponding to the WebAssembly virtual machine. The runtime system constructs a reference variable for the application based on the storage address corresponding to the data storage space allocated to the application, and records address
20 information related to the storage address of the data storage space in the reference variable, so as to update the data content of the corresponding reference variable in the application based on the value of the reference variable. Based on this, when the application needs to access data stored in the data storage space, the WebAssembly virtual machine can access the data in the data storage space based on the address information in the reference variable provided by the
25 application.

However, during the runtime of the application, the application can modify the reference variable in the application. Based on this, if the application illegally modifies the address information related to the data storage space in the reference variable, the WebAssembly virtual machine may not be able to accurately locate the data storage space corresponding to the
30 application based on the address information in the reference variable, which may not only cause data access failure, but also send data in the data storage space corresponding to other

applications to the application, resulting in illegal access to data of other applications, affecting the security and reliability of data access.

SUMMARY

5 The present application provides a data access method and apparatus to improve the security and reliability of data access in a runtime system based on a WebAssembly virtual machine.

In an aspect, the present application provides a data access method, applied to a runtime system based on a WebAssembly virtual machine, the method including:

10 obtaining a data access instruction initiated by an application, the data access instruction carrying first variable data of a reference variable, the first variable data of the reference variable being used to indicate a data address of target data to be accessed;

determining a storage area address of a data storage area corresponding to the application based on storage area identification information recorded in the first variable data of the reference variable, the storage area identification information being used to identify the data storage area corresponding to the application;

15 obtaining a first check factor associated with the reference variable, the first check factor being calculated by the runtime system based on the storage area address of the data storage area during a process of generating variable data corresponding to the reference variable for the application;

calculating a second check factor based on the storage area address;

20 obtaining, in response to that the first check factor is the same as the second check factor, the target data from the data storage area based on the first variable data of the reference variable and the storage area address.

In a feasible implementation, after determining the storage area address of the data storage area corresponding to the application, the method further comprises:

25 obtaining basic attribute information recorded in the data storage area based on the storage area address, the basic attribute information including attribute information associated with the data storage area and the application;

the first check factor is calculated based on the storage area address of the data storage area and the basic attribute information;

30 calculating the second check factor based on the storage area address comprises:

calculating the second check factor based on the basic attribute information recorded in the data storage area and the storage area address.

In yet another feasible implementation, after obtaining the data access instruction initiated by the application, the method further comprises:

- 5 obtaining a storage area type identifier from the first variable data of the reference variable, the storage area type identifier being one of a first type identifier and a second type identifier, the first type identifier indicating that a storage capacity of the data storage area corresponding to the application is not less than a set threshold, the second type identifier indicating that the storage capacity of the data storage area is less than the set threshold;
- 10 wherein the first check factor is calculated based on the storage area address of the data storage area, the basic attribute information, and the storage area type identifier;

calculating the second check factor based on the basic attribute information recorded in the data storage area and the storage area address comprises:

- 15 calculating the second check factor based on the basic attribute information recorded in the data storage area, the storage area address, and the storage area type identifier.

In yet another feasible implementation, obtaining the first check factor associated with the reference variable comprises:

obtaining, in response to that the storage area type identifier is the first type identifier, the first check factor from the first variable data of the reference variable;

- 20 obtaining, in response to that the storage area type identifier is the second type identifier, the first check factor associated with the reference variable from an attribute storage area of the data storage area.

In yet another feasible implementation, determining the storage area address of the data storage area corresponding to the application based on the storage area identification information recorded in the first variable data of the reference variable comprises:

- 25 obtaining, in response to that the storage area type identifier is the first type identifier, an address index recorded in the first variable data of the reference variable, and determining the storage area address of the data storage area corresponding to the application based on the address index, the address index being used to indicate a storage location of the storage area
- 30 address of the data storage area in an address storage table;

obtaining, in response to that the storage area type identifier is the second type identifier, offset information recorded in the first variable data of the reference variable, and determining the storage area address of the data storage area based on the offset information, the offset information being used to characterize a relative offset address of the storage area address of the data storage area with respect to a total space start address, the total space start address being a start address of a total storage space of the runtime system.

In yet another feasible implementation, obtaining the offset information recorded in the first variable data of the reference variable and determining the storage area address of the data storage area based on the offset information comprises:

obtaining first offset information and second offset information recorded in the first variable data of the reference variable, the first offset information being a remainder obtained by dividing the relative offset address by a target number of bytes, the second offset information being a quotient obtained by dividing the relative offset address by the target number of bytes;

determining the relative offset address corresponding to the data storage area based on the first offset information and the second offset information;

determining the storage area address of the data storage area based on the relative offset address.

In yet another feasible implementation, the first variable data of the reference variable includes a data offset address of the target data in the data storage area corresponding to the application;

obtaining, in response to that the first check factor is the same as the second check factor, the target data from the data storage area based on the first variable data of the reference variable and the storage area address comprises:

obtaining, in response to that the first check factor is the same as the second check factor, the target data from the data storage area based on the data offset address and the storage area address.

In yet another feasible implementation, before obtaining the data access instruction initiated by the application, the method further comprises:

obtaining a storage area request initiated by the application, the storage area request being used to request allocation of a storage space of a target storage capacity for the application;

creating the data storage area for the application based on the target storage capacity, and determining the storage area type identifier corresponding to the data storage area, where an attribute storage area of the data storage area records the basic attribute information associated

with the data storage area and the application, where in response to that the target storage capacity is not less than the set threshold, the storage area type identifier is the first type identifier, and in response to that the target storage capacity is less than the set threshold, the storage area type identifier is the second type identifier;

- 5 determining storage area identification information of the data storage area based on the storage area address of the data storage area;

calculating the first check factor based on the storage area address, the basic attribute information, and the storage area type identifier;

- 10 constructing, in response to that the target storage capacity is not less than the set threshold, second variable data for the reference variable of the application, the second variable data including the storage area type identifier, the first check factor, and the storage area identification information;

- 15 constructing, in response to that the target storage capacity is less than the set threshold, third variable data for the reference variable of the application, the third variable data including the storage area type identifier and the storage area identification information, and storing the first check factor in the attribute storage area of the data storage area.

In yet another feasible implementation, determining the storage area identification information of the data storage area based on the storage area address of the data storage area comprises:

- 20 storing, in response to that the target storage capacity is not less than the set threshold, the storage area address of the data storage area in an address storage table, and determining an address index of the storage area address in the address storage table, the address index being used to indicate a storage location of the storage area address in the address storage table;

- 25 determining, in response to that the target storage capacity is less than the set threshold, offset information corresponding to the storage area address of the data storage area, the offset information being used to characterize a relative offset address of the storage area address of the data storage area with respect to a total space start address, the total space start address being a start address of a total storage space of the runtime system;

- 30 constructing, in response to that the target storage capacity is not less than the set threshold, the second variable data for the reference variable of the application, the second variable data including the storage area type identifier, the first check factor, and the storage area identification information, comprises:

constructing, in response to that the target storage capacity is not less than the set threshold, the second variable data for the reference variable of the application, the second variable data including the storage area type identifier, the first check factor, and the address index;

5 constructing, in response to that the target storage capacity is less than the set threshold, the third variable data for the reference variable of the application, the third variable data including the storage area type identifier and the storage area identification information, comprises:

constructing, in response to that the target storage capacity is less than the set threshold, the third variable data for the reference variable of the application, the third variable data including the storage area type identifier and the offset information.

10 In yet another feasible implementation, determining, in response to that the target storage capacity is less than the set threshold, the offset information corresponding to the storage area address of the data storage area comprises:

15 determining, in response to that the target storage capacity is less than the set threshold, the relative offset address of the storage area address of the data storage area with respect to the total space start address;

determining a quotient and a remainder obtained by dividing the relative offset address by a target number of bytes, determining the remainder as first offset information, and determining the quotient as second offset information;

20 constructing, in response to that the target storage capacity is less than the set threshold, the third variable data for the reference variable of the application, the third variable data including the storage area type identifier and the offset information, comprises:

constructing, in response to that the target storage capacity is less than the set threshold, the third variable data for the reference variable of the application, the third variable data including the storage area type identifier, the first offset information, and the second offset information.

25 In another aspect, the present application further provides a data access apparatus, applied to a runtime system based on a WebAssembly virtual machine, the apparatus comprising:

30 an instruction obtaining unit, configured to obtain a data access instruction initiated by an application, the data access instruction carrying first variable data of a reference variable, the first variable data of the reference variable being used to indicate a data address of target data to be accessed;

an address determining unit, configured to determine a storage area address of a data storage

area corresponding to the application based on storage area identification information recorded in the first variable data of the reference variable, the storage area identification information being used to identify the data storage area corresponding to the application;

5 a factor obtaining unit, configured to obtain a first check factor associated with the reference variable, the first check factor being calculated by the runtime system based on the storage area address of the data storage area during a process of generating variable data corresponding to the reference variable for the application;

a first factor calculation unit, configured to calculate a second check factor based on the storage area address;

10 a data obtaining unit, configured to obtain, in response to that the first check factor is the same as the second check factor, the target data from the data storage area based on the first variable data of the reference variable and the storage area address.

In a feasible implementation, the apparatus further comprises: an attribute obtaining unit, configured to obtain basic attribute information recorded in the data storage area based on the
15 storage area address, the basic attribute information including attribute information associated with the data storage area and the application;

the first check factor obtained by the factor obtaining unit is calculated based on the storage area address of the data storage area and the basic attribute information;

the first factor calculation unit is specifically configured to calculate the second check factor
20 based on the basic attribute information and the storage area address.

From the above, it can be seen that in the present application, after the runtime system allocates a data storage area to an application, during the process of generating variable data of a reference variable related to the data storage area for the application, a first check factor associated with the reference variable is calculated based on the storage area address
25 corresponding to the data storage area. On this basis, since the storage area address corresponding to the data storage area is fixed and unchanged, if the application does not illegally modify the storage area identification information of the data storage area in the reference variable during operation, then after the application initiates a data access instruction carrying the reference variable, the storage area address determined based on the storage area
30 identification information in the variable data of the reference variable is the actual storage area address of the data storage area. Accordingly, the second check factor calculated based on the storage area address must be consistent with the first check factor. It can be seen that only when

the first check factor and the second check factor are the same, is the target data obtained from the data storage area, thereby reducing the occurrence of data access errors or illegal acquisition of data from other applications due to the application illegally modifying the storage area identification information in the reference variable, improving the reliability and security of data access.

BRIEF DESCRIPTION OF THE DRAWINGS

To more clearly illustrate the technical solutions in the embodiments of the present application, the drawings for describing the embodiments are briefly introduced below. Apparently, the drawings in the following description are only some embodiments of the present application. For those of ordinary skill in the art, other drawings can be obtained based on these drawings without creative effort.

FIG. 1 shows a flowchart of a data access method according to an embodiment of the present application;

FIG. 2 shows a flowchart of an implementation of allocating a data storage area and constructing variable data of a reference variable for an application according to an embodiment of the present application;

FIG. 3 shows a schematic structural diagram of a composition of a reference variable;

FIG. 4 shows a schematic structural diagram of a reference variable according to an embodiment of the present application;

FIG. 5 shows another schematic structural diagram of a reference variable according to an embodiment of the present application;

FIG. 6 shows another flowchart of an implementation of allocating a data storage area and constructing variable data of a reference variable for an application according to an embodiment of the present application;

FIG. 7 shows another flowchart of a data access method according to an embodiment of the present application;

FIG. 8 shows a schematic structural diagram of a data access apparatus provided in an embodiment of the present application.

DETAILED DESCRIPTION OF THE EMBODIMENTS

The technical solutions in the embodiments of the present application will be clearly and

completely described below with reference to the drawings in the embodiments of the present application. Apparently, the described embodiments are only some, rather than all, of the embodiments of the present application. Based on the embodiments in the present application, all other embodiments obtained by those of ordinary skill in the art without inventive effort shall fall within the scope of protection of the present application.

FIG. 1 shows a flowchart of a data access method provided in an embodiment of the present application. The method of this embodiment is applied to a runtime system based on a WebAssembly virtual machine.

WebAssembly (abbreviated as Wasm) is a binary format. The runtime system based on the WebAssembly virtual machine is a runtime system corresponding to an embedded operating system based on the WebAssembly virtual machine. The runtime system includes: a WebAssembly virtual machine and a runtime environment corresponding to the WebAssembly virtual machine. In this runtime system, the WebAssembly virtual machine can leverage the runtime environment to run bytecode of applications written based on WebAssembly .

The runtime system based on the WebAssembly virtual machine is generally deployed in resource-constrained devices, which refer to chips or devices with limited computing power, power, or storage capacity.

The data access method of this embodiment may include step S101 to S106.

S101 includes obtaining a data access instruction initiated by an application.

The data access instruction carries first variable data of a reference variable, and the first variable data of the reference variable is used to indicate a data address of target data to be accessed.

The reference variable is a variable in the application used to record the data storage area of the application and the address of the data to be accessed in the data storage area.

It can be understood that the variable data of the reference variable is actually the data content of the reference variable, also called the value of the reference variable. When the application applies for a data storage area from the runtime system based on the WebAssembly virtual machine, the runtime system determines an initial value of the reference variable for the application, so that the application assigns a value to the reference variable in the application based on the initial value of the reference variable determined by the runtime system. During the operation of the application, the application can legally modify the data address related to the data to be accessed in the reference variable, so that the variable data of the reference

variable sent when the application requests data access is not exactly the same as the variable data of the reference variable generated by the runtime system for the application.

Based on this, for ease of distinction, the variable data of the reference variable carried in the data access instruction initiated by the application is referred to as first variable data, and the variable data of the reference variable subsequently generated by the runtime system for the application is referred to as second variable data or third variable data.

It can be understood that in the runtime system based on the WebAssembly virtual machine, the WebAssembly virtual machine can respond to the data access instruction initiated by the application, obtain the data that the application needs to access, and return it to the application.

S102 includes determining a storage area address of a data storage area corresponding to the application based on storage area identification information recorded in the first variable data of the reference variable.

It can be understood that the runtime system based on the WebAssembly virtual machine allocates different data storage areas to different applications. To distinguish the data storage areas corresponding to different applications, the runtime system based on the WebAssembly virtual machine records storage area identification information of the data storage area corresponding to the application in the reference variable of the application. Accordingly, the first variable data of the reference variable includes the storage area identification information.

The storage area identification information is used to identify the data storage area corresponding to the application. However, the storage area identification is different from the storage area address of the data storage area, where the storage area address of the data storage area may be a start address of the data storage area.

In the present application, the storage area identification information recorded in the reference variable may vary.

For example, in a feasible implementation, the storage area identification information recorded in the first variable data of the reference variable may be an address index, which is used to indicate a storage location of the storage area address corresponding to the data storage area of the application in an address storage table. For example, the address index may be a sequence number of the storage area address of the data storage area in the address storage table. The address storage table is used to record storage area addresses of data storage areas of different applications.

In this feasible implementation, after the address index is obtained from the first variable data

of the reference variable, the storage area address corresponding to the data storage area of the application can be queried from the address storage table based on the address index.

In yet another feasible implementation, the storage area identification information recorded in the first variable data of the reference variable may be offset information corresponding to the data storage area of the application, and the offset information is used to characterize a relative offset address corresponding to the storage area address of the data storage area. The relative offset address corresponding to the storage area address of the data storage area refers to an offset address of the storage area address of the data storage area with respect to a start address of the total storage space in the runtime system. For ease of distinction, the start address of the total storage space of the runtime system may also be referred to as a total space start address.

The offset information may be the relative offset address. On this basis, based on the relative offset address and the start address of the total storage space in the runtime system, the storage area address of the data storage area of the application can be determined.

In an optional implementation, in order to represent the storage area address corresponding to the data storage area with fewer bits, the offset information in the present application may alternatively include first offset information and second offset information. The first offset information is a remainder obtained by dividing the relative offset address corresponding to the data storage area by a target number of bytes. The second offset information is a quotient obtained by dividing the relative offset address corresponding to the data storage area by the target number of bytes.

The target number of bytes can be determined in conjunction with the number of bits available in the reference variable for recording address information of the data storage area, which is not specifically limited. For example, considering that a 2-byte (16-bit) addressing range in a storage chip is 64K, the target number of bytes may be 64K.

It can be understood that after dividing the relative offset address corresponding to the data storage area by the target number of bytes, the obtained quotient and remainder have relatively few bits. Therefore, the remainder is determined as the first offset information and the quotient is determined as the second offset information, and the first offset information and the second offset information occupy fewer bits in the reference variable. Moreover, the relative offset address corresponding to the data storage area can be derived from the first offset information and the second offset information, so that the storage area address of the data storage area can be determined based on the relative offset address corresponding to the data storage area and the start address of the total storage space of the runtime system.

In practical applications, one of the above two feasible cases of storage area identification information can be selected as needed, or the two cases can be used in combination, which is not specifically limited.

S103 includes obtaining basic attribute information recorded in the data storage area based on the storage area address.

The basic attribute information includes attribute information associated with the data storage area and the application. The basic attribute information is an inherent attribute of the data storage area and the application, which is information that remains fixed during the running of the application.

For example, the basic attribute information may include: a storage capacity of the data storage area (i.e., the size of the data storage area), information of the application to which the data storage area belongs (e.g., the name or identifier of the application), a data type used for storing data generated during the running of the application in the data storage area, and a randomly generated random factor, etc.

It can be understood that the data storage area of the application is divided into two parts: an attribute storage area and a data body storage area. The attribute storage area, also called the data header part, is used to store the basic attribute information. Therefore, the basic attribute information is also called data header information. The data body storage area can store data generated during the running of the application, so the data stored in the data body storage area is also the data that needs to be read during the running of the application.

S104 includes obtaining a first check factor associated with the reference variable.

The first check factor is calculated by the runtime system based on the WebAssembly virtual machine, based on the storage area address of the data storage area and the basic attribute information during a process of generating the variable data of the reference variable for the application.

As described above, after the runtime system based on the WebAssembly virtual machine allocates the data storage area to the application, the runtime system needs to record address information related to the data storage area into a corresponding reference variable of the application. Before the application runs, the variable data (i.e., value) of the reference variable of the application is null or a default value. Based on this, in order to assign a value to the reference variable in the application, the runtime system needs to construct a reference variable containing address information related to the data storage area, i.e., generate the variable data

of the reference variable, so that the application can assign a value to the reference variable in the application based on the variable data of the reference variable generated by the runtime system.

In the present application, the specific data structure form of the reference variable and the specific implementation manner of determining the value of the reference variable by the runtime system may vary, which is not limited in the present application. Descriptions thereof are made with reference to some examples hereinafter, which therefore are not elaborated herein.

It can be understood that during the running of the application, the variable data of the reference variable may be modified. For example, during the running of the application, based on the data storage address of the data stored in the data storage area, the data offset address corresponding to the data storage address in the reference variable is modified. However, as described in the background section, if the application illegally modifies the address information related to the data storage area in the reference variable, subsequent data reading will be abnormal, and data of other applications may be illegally read. In view of this, in order to verify whether the application has modified the address information related to the data storage area in the reference variable, the present application introduces a check factor associated with the reference variable.

To distinguish from the check factor generated in the verification stage in the subsequent step S105, the check factor determined during the process of the runtime system generating the variable data of the reference variable for the application is referred to as a first check factor, and the check factor generated in the verification stage is referred to as a second check factor.

It can be understood that when the data storage area allocated for the application is determined, the storage area address and the basic attribute information of the data storage area are fixed and will not change with the running of the application. In the present application, the first check factor is associated with the storage area address corresponding to the data storage area of the application. Therefore, if the application illegally modifies the storage area identification information recorded in the reference variable, the storage area address determined based on the storage area identification information recorded in the reference variable is no longer the actual storage area address of the data storage area of the application, and then the check factor determined based on the storage area address will inevitably be inconsistent with the first check factor.

Moreover, since the reference variable records the storage area identification information of the data storage area, and the storage area identification information is different from the storage area address, in general, the application cannot obtain the storage area address of the data

storage area based on the storage area identification information, and accordingly cannot forge or reconstruct the first check factor. In addition, even if the application obtains the storage area address of its data storage area by illegal means, the application cannot obtain the basic attribute information stored in the data storage area. For example, the application cannot obtain the capacity of the data storage area, and even if it can, it cannot determine the random factor contained in the basic attribute information, thereby greatly reducing the risk of the first check factor being cracked.

In the present application, there are many manners of calculating the first check factor based on the basic attribute information and the storage area address, which is not limited in the present application. For example, in the present application, the first check factor can be calculated based on the basic attribute information and the storage area address using a checksum algorithm, such as XOR or Cyclic Redundancy Check (CRC) algorithm, which is not limited.

It can be understood that in practical applications, in addition to the basic attribute information and the storage area address, other parameters related to the data storage area or the storage area address of the data storage area may also be considered when calculating the first check factor, which is not limited. For example, in a feasible implementation, the first check factor may alternatively be calculated based on the basic attribute information, the storage area address, and the storage area identification information.

S105 includes calculating a second check factor based on the basic attribute information and the storage area address.

In the present application, the algorithm used to calculate the second check factor is the same as the algorithm used to calculate the first check factor. For example, when the first check factor is calculated using the CRC algorithm based on the attribute information and the storage area address, the second check factor can also be calculated using the CRC algorithm, which is not elaborated herein.

In an optional implementation, on the premise that the first check factor is calculated based on the basic attribute information, the storage area address, and the storage area identification information, also in the present application, the second check factor can be calculated based on the basic attribute information, the storage area address, and the storage area identification information in the reference variable.

S106 includes obtaining, in response to that the first check factor is the same as the second check factor, the target data from the data storage area based on the first variable data of the reference variable and the storage area address.

Based on the above description, if the storage area identification information in the reference variable has not been illegally modified by the application, then the storage area address determined based on the storage area identification information is the actual storage address of the storage area of the application. On this basis, the second check factor calculated based on the storage area address and the basic attribute information are inevitably the same as the first check factor.

Based on this, when the second check factor is the same as the first check factor, the runtime system can determine that the storage area address identifier in the first variable data of the reference variable sent by the application has not been illegally modified, and therefore, the storage area address can be determined based on the storage area identifier, and the target data can be obtained from the data storage area pointed to by the storage area address.

In the present application, the first variable data of the reference variable can be used to indicate the data address of the target data to be accessed. Therefore, the target data can be obtained from the data storage area based on the first variable data of the reference variable and the storage area address.

In the present application, how the variable data of the reference variable specifically indicates the data address of the target data is not limited.

In a feasible implementation, the first variable data of the reference variable may include a data offset address of the target data to be accessed by the application in the data storage area corresponding to the application. The data offset address is an offset address of the data storage address of the target data in the data storage area with respect to the storage area address of the data storage area.

Accordingly, after the data offset address is obtained from the first variable data of the reference variable, the target data can be obtained from the data storage area based on the data offset address and the storage area address of the data storage area. For example, based on the data offset address and the storage area address of the data storage area, the data storage address of the target data in the data storage area can be determined, and based on the data storage address, the target data can be obtained from the data storage area.

It can be understood that in the present application, the target data can be read from the data storage area based on the data offset address recorded in the variable data of the reference variable and the storage area address of the data storage area. Essentially, determining the target data to be accessed from the data storage area based on the data offset address and the data storage address is a process of implementing data access using methods such as pointers,

thereby achieving quick access to the data of the application by combining methods such as pointers, array subscripts, or structure member references.

From the above, it can be seen that in the present application, after the runtime system allocates a data storage area to the application, during the process of generating variable data of a reference variable related to the data storage area for the application, a first check factor associated with the reference variable is calculated based on the basic attribute information stored in the data storage area and the storage area address corresponding to the data storage area. On this basis, since the basic attribute information and the storage area address corresponding to the data storage area are fixed and unchanged, if the application does not illegally modify the storage area identification information of the data storage area in the reference variable during operation, then after the application initiates a data access instruction carrying the reference variable, the storage area address determined based on the storage area identification information in the variable data of the reference variable is the actual storage area address of the data storage area. Accordingly, the second check factor calculated based on the storage area address and the basic attribute information must be consistent with the first check factor. It can be seen that only when the first check factor and the second check factor are the same, is the target data obtained from the data storage area, thereby reducing the occurrence of data access errors or illegal acquisition of data from other applications due to the application illegally modifying the storage area identification information in the reference variable, improving the reliability and security of data access.

It can be understood that the embodiment of FIG. 1 is described by taking the first check factor being calculated based on the storage area address and the basic attribute information corresponding to the data storage area of the application as an example. However, it can be understood that in practical applications, since the storage area address corresponding to the data storage area is fixed and unchanged, calculating the first check factor based solely on the storage area address of the data storage area corresponding to the application is also applicable to this embodiment.

On this basis, after the data access instruction is obtained, in the present application, the storage area address of the data storage area corresponding to the application can be determined only based on the storage area identification information recorded in the first variable data of the reference variable, and calculate the second check factor based on the storage area address.

It can be understood that if the application does not illegally modify the storage area

identification information of the data storage area in the reference variable during running, then after the application initiates the data access instruction carrying the reference variable, the storage area address determined based on the storage area identification information in the variable data of the reference variable is the actual storage area address of the data storage area.

5 Accordingly, the second check factor calculated based on the storage area address must be consistent with the first check factor. Based on this, only when the first check factor and the second check factor are the same, is the target data obtained from the data storage area, which can also reduce the occurrence of data access errors or illegal acquisition of data from other applications due to the application illegally modifying the storage area identification
10 information in the reference variable, improving the reliability and security of data access.

In the present application, the first check factor may be associated with the reference variable in various ways, which is not limited. Two feasible implementations are described below as examples.

15 In a first feasible implementation, in the present application, the first check factor may be recorded in the variable data of the reference variable. For example, consecutive 6 bits or other numbers of bits in the reference variable are set to record the first check factor. Accordingly, in the present application, the first check factor can be obtained from the first variable data of the reference variable.

20 In a second feasible implementation, the first check factor may alternatively be stored in the attribute storage area in the data storage area corresponding to the application. As described above, the attribute storage area is a storage area in the data storage area for storing basic attribute information. Accordingly, in the present application, the first check factor associated with the reference variable can be obtained from the attribute storage area of the data storage area.

25 In the first feasible implementation, the first check factor needs to occupy bits of the reference variable, which may result in a relatively small number of data storage areas of different applications that can be represented by the reference variable. In the second feasible implementation, the first check factor does not need to occupy bits of the reference variable, so that the reference variable can be used to represent more data storage areas of different
30 applications.

In practical applications, in the present application, any of the above feasible implementations can be selected to record the first check factor in combination with the actual requirements of the runtime system based on the WebAssembly virtual machine.

In an implementation, in the present application, which of the above two feasible implementations to record the first check factor can be selected based on the size of the data storage area requested by the application.

5 For ease of understanding, the process of allocating a data storage area and constructing variable data of a reference variable by the runtime system based on the WebAssembly virtual machine for an application is introduced below.

FIG. 2 shows a flowchart of an implementation of allocating a data storage area and constructing variable data of a reference variable for an application according to an embodiment of the present application. The method of this embodiment may include steps S201 to S206.

10 S201 includes obtaining a storage area request initiated by an application, the storage area request being used to request allocation of a storage space of a target storage capacity for the application.

For example, the application initiates a storage area request to the runtime system based on the WebAssembly virtual machine by calling a specified interface provided by the runtime system.

15 The target storage capacity is the size of the storage space requested by the application. For example, the target storage capacity may be 6K, 12K, or 20K, etc., which is not limited.

S202 includes creating a data storage area for the application based on the target storage capacity, and determining a storage area type identifier corresponding to the data storage area.

20 It can be understood that in the runtime system based on the WebAssembly virtual machine, the embedded operating system based on the WebAssembly virtual machine can determine a data storage area that can be allocated to the application from the total storage space of the runtime system based on the storage area request initiated by the application.

25 The data storage area can be divided into an attribute storage area and a data body storage area, where the attribute storage area records basic attribute information associated with the data storage area and the application. For example, the basic attribute information may include: the target storage capacity of the data storage area, information of the application to which the data storage area belongs, a randomly generated random factor, and a data type used for storing data generated during the running of the application in the data storage area.

30 The storage area type identifier is used to characterize the category to which the storage capacity of the data storage area belongs. Specifically, the data storage area is divided into two types: a first storage area type, representing that the storage capacity of the data storage area is not less

than a set threshold; and a second storage area type, representing that the storage capacity of the data storage area is less than the set threshold.

The set threshold can be set as needed. For example, considering that a large data storage area requested by an application in the runtime system is 8K, the set threshold can be set to 8K.

- 5 Accordingly, the storage area type identifier may include: a first type identifier corresponding to the first storage area type, and a second type identifier corresponding to the second storage area type. Based on this, it can be seen that if the target storage capacity is not less than the set threshold, the storage area type identifier corresponding to the data storage area of the application is the first type identifier; if the target storage capacity is less than the set threshold,
10 the storage area type identifier of the data storage area is the second type identifier.

S203 includes determining storage area identification information of the data storage area based on the storage area address of the data storage area.

- It can be understood that using storage area identification information to identify the storage area address of the data storage area in the reference variable (i.e., the variable data of the
15 reference variable), rather than directly using the storage area address, can improve the security of the storage area addresses corresponding to the data storage areas of various applications.

- As described above, the storage area identification information may be an address index of the storage area address of the data storage area in the address storage table, or may be offset information corresponding to the storage area address of the data storage area, where the offset
20 information is used to characterize the relative offset address corresponding to the storage area address of the data storage area.

- In a feasible implementation, if the target storage capacity is not less than the set threshold, the storage area address of the data storage area is stored in the address storage table, the address index of the storage area address in the address storage table is determined, and the address
25 index is used as the storage area identification information of the data storage area corresponding to the application.

If the target storage capacity is less than the set threshold, offset information corresponding to the storage area address of the data storage area is determined, and the offset information is used as the storage area identification information of the data storage area.

- 30 S204 includes calculating a first check factor based on the storage area address of the data storage area and the basic attribute information.

The specific process of calculating the first check factor is as described above, which is not elaborated herein.

In an optional implementation, the first check factor may alternatively be calculated based on the storage area address of the data storage area, the basic attribute information, and the storage area type identifier. It can be understood that when the first check factor is calculated in this manner, during the data access phase, when calculating the second check factor, the second check factor may be calculated based on the basic attribute information, the storage area address, and the storage area type identifier.

In practice, in the present application, the various methods of calculating the first check factor described in this embodiment may be combined to comprehensively calculate the first check factor. For example, the first check factor may be calculated based on the storage area address of the data storage area, the basic attribute information, the storage area type identifier, and the storage area identification information. Accordingly, during the subsequent data access phase, the second check factor may be calculated based on the storage area address of the data storage area, the basic attribute information, the storage area type identifier, and the storage area identification information.

S205 includes constructing, in response to that the target storage capacity is not less than the set threshold, second variable data for the reference variable of the application, the second variable data including the storage area type identifier, the first check factor, and the storage area identification information.

S206 includes constructing, in response to that the target storage capacity is less than the set threshold, third variable data for the reference variable of the application, the third variable data including the storage area type identifier and the storage area identification information, and storing the first check factor in the attribute storage area of the data storage area.

It can be understood that if the storage capacity of the data storage area requested by the application is large, the number of data storage areas that can be allocated to applications from the total storage space of the runtime system is relatively small. For example, for ease of understanding, assuming that the total storage space of the runtime system is 6400K, if each application requests a data storage area of 32K, then the total storage space can only allocate data storage areas to 200 applications; if each application requests a data storage area of 8K, then the total storage space can allocate data storage areas to 800 applications.

It can be understood that the larger the storage capacity of the data storage area requested by the application, the fewer data storage areas can be allocated based on the total storage space,

and therefore the smaller the number of bits occupied by the storage area identification information for distinguishing the data storage areas of various applications in the reference variable. Conversely, the smaller the capacity of the data storage area requested by the application, the more data storage areas can be allocated based on the total storage space.

5 Therefore, the number of bits occupied by the storage area identification information for distinguishing the data storage areas of various applications in the reference variable is relatively large.

It can be understood that since the number of bits of the reference variable is fixed and limited, the number of bits available in the reference variable for representing the storage area
10 identification information of the data storage area is also limited. Based on this, when the storage capacity of the data storage area requested by the application is small and hence the number of bits required for the data storage area is large, i.e., the target storage capacity is less than the set threshold, in the present application, the first check factor may not be included in the variable data of the reference variable, and the first check factor is stored in the attribute
15 storage area in the data storage area of the application.

Accordingly, if the target storage capacity is not less than the set threshold, considering that the number of bits used for representing the storage area identification information in the reference variable is relatively small, the first check factor may be included in the variable data of the reference variable.

20 In the present application, the structure of the reference variable may vary, which is not limited.

It can be understood that after constructing the second variable data or the third variable data of the reference variable, the runtime system can send the second variable data or the third variable data of the reference variable to the application, so that the application can assign a value to the reference variable in the application based on the second variable data or the third
25 variable data of the reference variable.

On the basis of this embodiment, after the runtime system obtains the data access instruction of the application, and after obtaining the first variable data of the reference variable from the data access instruction, the runtime system can first obtain the storage area type identifier in the first variable data. As described above, the storage area type identifier in the first variable data of
30 the reference variable may be a first storage type identifier or a second storage type identifier. The first type identifier indicates that the storage capacity of the data storage area corresponding to the application is not less than the set threshold, and the second type identifier indicates that the storage capacity of the data storage area is less than the set threshold.

Accordingly, if the storage area type identifier corresponding to the application is the first type identifier, the first check factor can be obtained from the first variable data of the reference variable. If the storage area type identifier corresponding to the application is the second type identifier, the first check factor associated with the reference variable can be obtained from the attribute storage area of the data storage area.

It can be understood that when the storage area types corresponding to the data storage area of the application are different, the storage area identification information determined based on the storage area address corresponding to the data storage area of the application may also vary.

As described above, during the process of constructing the variable data of the reference variable for the application, if the target storage capacity of the data storage area is not less than the set threshold, the storage area identification information is an index address. If the target storage capacity is less than the set threshold, the storage area identification information is offset information.

Accordingly, during the data access phase, if the storage area type identifier obtained from the first variable data of the reference variable is the first type identifier, the address index recorded in the first variable data of the reference variable can be obtained, and the storage area address of the data storage area corresponding to the application is determined based on the address index. If the storage area type identifier is the second type identifier, the offset information recorded in the first variable data of the reference variable is obtained, and the storage area address of the data storage area corresponding to the application is determined based on the offset information.

The above implementation is described in detail below.

For ease of understanding, a feasible structure of the reference variable in the present application is first described.

The total number of bits of the reference variable is fixed, and the specific number of bits can be set as needed. Typically, the reference variable is 4-byte data. For ease of description, the reference variable is described below as 4-byte data.

In a feasible implementation, FIG. 3 shows a structural diagram of a composition of a reference variable.

In FIG. 3, the reference variable is divided into three parts.

The first part is the reference type area in FIG. 3, which is specifically the highest bit of the

reference variable, i.e., bit 31, used to identify the variable category of the reference variable. In the present application, bit 31 is used to indicate that the reference variable is a variable related to the data storage area of the application.

The second part is the offset range area, which is specifically bits 0 to 15 of the reference variable, used to record the offset range information that can be modified by the reference variable. A maximum size of the offset range area is 64K.

The offset range information belongs to the address range between the start address and the end address of the data storage area of the application. The application can identify data offset addresses of different data in the data storage area by modifying the specific value of the offset range information.

The third part is the address area, which is specifically bits 16 to 30 of the reference variable, used to record the address information of the data storage area corresponding to the application associated with the reference variable. For example, the address information is the storage area identification information described above.

For example, the storage area identification information is an address index. The total storage space in the runtime system based on the WebAssembly virtual machine is limited, i.e., the addressable range is limited. In a limited storage space, addresses of the storage space can be recorded using an address storage table. Accordingly, bits 16 to 30 in the reference variable can be used to record the index address of the data storage area in the address storage table. For example, the index address is a sequence number in the address storage table.

On this basis, bits 16 to 30 can represent index addresses of 32767 different data storage areas.

In practice, the address information of 32767 data storage areas may not meet the data storage requirements of all applications in the runtime system based on the WebAssembly virtual machine. Based on this, in the present application, the number of bits occupied by the storage area identification information (e.g., address index) and the offset range information in the reference variable may be appropriately adjusted according to the actual application scenario.

Considering that the reference variable may be modified during the running of the application, in order to improve the security of data access, in the present application, different structural forms of the reference variable can be used according to the size of the data storage area requested by the application.

For the case where the target storage capacity of the data storage area requested by the application is not less than the set threshold (e.g., 8K), a first structure of the reference variable

may be used.

Specifically, on the basis of FIG. 3, a check factor is to be added in the reference variable, so the number of bits of the address area in the reference variable needs to be adjusted, and a check factor area is added. FIG. 4 shows a schematic structural diagram of a reference variable suitable for the first storage area type in the present application, i.e., a schematic structural diagram of the data composition of the first reference variable.

It can be seen from FIG. 4 that the reference variable is divided into four parts.

Bit 31 of the reference variable is the storage type area. The storage type area is used to identify the storage area type identifier corresponding to the data storage area of the application associated with the reference variable. In the present application, in order to identify that the data storage area corresponding to the reference variable is of the first storage area type, the value of bit 31 is a target value, e.g., the target value is 1. Accordingly, when the value of bit 31 in the reference variable is 1, it is determined that the storage area type identifier in the reference variable is the first type identifier.

Bits 0 to 15 of the reference variable are still used as a 64K offset range area for recording offset range information of data in the data storage area.

Bits 16 to 22 of the reference variable are an index area for recording an address index of the data storage area.

Bits 23 to 30 of the reference variable are a check factor area for recording the first check factor generated for the data storage area.

In the reference variable of FIG. 4, the total number of bits of the index area is 7, which can represent 256 different index addresses, and the number of bits for identifying data storage areas of applications is relatively small. As described above, when the total storage space of the runtime system is fixed, the number of data storage areas with large capacity that can be allocated is also relatively small. Therefore, the structure of the reference variable in FIG. 4 is suitable for representing address information of data storage areas with target storage capacity not less than the set threshold.

For the case where the target storage capacity of the data storage area requested by the application is less than the set threshold (e.g., 8K), the second structure of the reference variable may be used.

Specifically, in order to represent data storage areas of more applications through the address

area of the reference variable, on the basis of FIG. 3, in the present application, the number of bits occupied by the offset range area can be reduced and the number of bits occupied by the address area for recording the address information of the data storage area can be increased. FIG. 5 shows a schematic structural diagram of a data composition of a reference variable suitable for the second storage area type, i.e., a schematic structural diagram of the data composition of the second reference variable.

It can be seen from FIG. 5:

Bit 31 and bit 30 of the reference variable together form a storage type area. In FIG. 5, bit 31 is storage type area 1, and bit 30 is storage type area 2. The storage type area is used to identify the storage area type identifier corresponding to the data storage area of the application associated with the reference variable. The value of bit 31 of the second reference variable is different from the value of bit 31 of the first reference variable.

In addition, in the present application, the storage type area of the first reference variable and the second reference variable can also be reused as the reference type area.

For example, in the present application, when the value of bit 31 of the first reference variable is set to 1, the value of bit 31 of the second reference variable can be set to 0, and the value of bit 30 of the second reference variable is set to 1. On this basis, if the runtime system identifies that the value of bit 31 of the reference variable is 1, it determines that the storage area type identifier in the reference variable is the first type identifier, and can confirm that the reference variable is a reference variable that identifies a data storage area.

If the runtime system identifies that bit 31 of the reference variable is 0, it can continue to identify bit 30 of the reference variable. If the value of bit 30 is 1, it determines that the storage area type identifier in the reference variable is the second type identifier, and can confirm that the reference variable is a reference variable that identifies a data storage area; if the value of bit 30 in the reference variable is 0, it is considered that the reference variable is not a reference variable for identifying a data storage area.

Bits 0 to 12 of the reference variable form an 8K offset range area for recording the data offset address of data in the data storage area relative to the data storage area.

Bits 13 to 29 of the reference variable form a storage area offset information recording area, which can record offset information corresponding to the data storage area of the application, and the offset information is used to characterize the relative offset address corresponding to the storage area address of the data storage area.

Further, in order to enable bits 13 to 29 to be used to distinguish data storage areas of more applications, in the present application, bits 16 to 29 can be used to record first offset information, and bits 13 to 15 are used to record second offset information.

As described above, the first offset information is a remainder obtained by dividing the relative offset address corresponding to the storage area address of the data storage area by a target number of bytes. For example, the target number of bytes is 64K, then bits 13 to 29 in the reference variable actually record the relative offset of the relative offset address of the data storage area within 64K.

The second offset information is the quotient obtained by dividing the relative offset address of the data storage area by the target number of bytes, i.e., how many complete 64K offsets exist in the relative offset address of the data storage area.

To avoid reducing the number of data storage areas that can be represented by the reference variable of FIG. 5 due to the check factor occupying bits of the reference variable, in the data structure of the second reference variable shown in FIG. 5, the reference variable does not include the first check factor.

The first check factor can be stored in the attribute storage area of the data storage area.

Combining the above two structures of the reference variable, another implementation of allocating a data storage area and constructing variable data of a reference variable for an application in the present application is described below.

FIG. 6 shows a flowchart of yet another implementation of allocating a data storage area and constructing variable data for a reference variable for an application according to an embodiment of the present application. This embodiment is applied to a runtime system based on a WebAssembly virtual machine, and may include steps S601 to S610.

S601 includes obtaining a storage area request initiated by an application.

The storage area request is used to request allocation of a storage space of a target storage capacity for the application.

S602 includes creating, in response to that the target storage capacity is not less than the set threshold, a data storage area for the application, and determining that a storage area type identifier corresponding to the data storage area is a first type identifier.

For example, in the data structures of the reference variable shown in FIG. 4 and FIG. 5, the first type identifier may be represented as 1, and the second type identifier may be represented

as 01.

The attribute storage area of the data storage area stores basic attribute information. The basic attribute information is basic attribute information generated based on attributes of the data storage area and attributes of the application after the runtime system determines the data storage area that can be allocated to the application.

For example, the basic attribute information may include: the size of the data storage area (i.e., the target storage capacity), the name or other identification information of the application to which the data storage area belongs, a randomly generated random factor, and a data type used for storing data generated by the application in the data storage area, etc.

S603 includes storing the storage area address of the data storage area in an address storage table, and determining an address index of the storage area address in the address storage table.

The address index is used to indicate a storage location of the storage area address of the data storage area in the address storage table. For example, the address index may be a sequence number of the storage area address of the data storage area in the address storage table.

In an optional implementation, before storing the storage area address of the data storage area in the address storage table, it can also be detected whether there is vacancy in the address storage table. If there is no vacancy in the address storage table, the data storage area cannot be allocated to the application, and an indication message of failure to allocate the data storage area can be returned to the application. If there is vacancy in the address storage table, the storage area address can be recorded in the address storage table.

S604 includes calculating a first check factor based on the storage area address of the data storage area, the basic attribute information recorded in the data storage area, the address index, and the first type identifier.

For example, a checksum is calculated using a checksum algorithm, such as CRC, with the storage area address of the data storage area, the basic attribute information, the address index, and the first type identifier, to obtain the first check factor.

It can be understood that this embodiment is described by taking one implementation of calculating the first check factor as an example. However, in practical applications, the first check factor may alternatively be calculated based on the storage area address of the data storage area, the basic attribute information, and the first type identifier.

Similarly, calculating the first check factor based on the storage area address of the data storage

area and the basic attribute information, or calculating the first check factor based on the storage area address of the data storage area, the basic attribute information, and the address index, is also applicable in this embodiment, which is not limited.

5 S605 includes constructing second variable data for the reference variable of the application, the second variable data including the first type identifier, the first check factor, and the address index, and sending the second variable data of the reference variable to the application.

For example, taking the reference variable as 4-byte data as an example, the data structure of the constructed second variable data of the reference variable can be seen in FIG. 4, which is not elaborated herein.

10 It can be understood that in this embodiment, the first check factor is associated with the storage area address, and generally the application cannot obtain the storage area address, even if the application can obtain the storage area address, the application cannot obtain information such as the random factor in the basic attribute information, reducing the possibility of the first check factor being cracked or arbitrarily forged.

15 In addition, in order to further improve the security of the first check factor, when calculating the first check factor in the present application, the first type identifier and the address index are also used, which can reduce the risk of being easily cracked due to too few parameters used in calculating the first check factor. Based on this, after the reference variable is sent to the application, the possibility of the application modifying the first check factor and forging the
20 first check factor with corresponding information is low.

S606 includes creating, in response to that the target storage capacity is less than the set threshold, a data storage area for the application, and determining that the storage area type identifier corresponding to the data storage area is the second type identifier.

25 S607 includes determining a relative offset address of the storage area address of the data storage area with respect to the total space start address.

As described above, the total space start address is the start address of the total storage space of the runtime system.

S608 includes determining a quotient and a remainder obtained by dividing the relative offset address by a target number of bytes, determining the remainder as first offset information, and
30 determining the quotient as second offset information.

For example, the target number of bytes is 64K, or other values, which is not limited.

S609 includes calculating a first check factor based on the storage area address of the data storage area, the basic attribute information recorded in the data storage area, the first offset information, the second offset information, and the second type identifier.

It can be understood that in the case where the target storage capacity of the data storage area is less than the set threshold, step S609 is described by taking one implementation of calculating the first check factor as an example. In practical applications, calculating the first check factor based on the storage area address of the data storage area and the basic attribute information, or calculating the first check factor based on the storage area address, the basic attribute information, and the second type identifier, or calculating the first check factor based on the storage area address, the basic attribute information, the first offset information, and the second offset information is also applicable in this embodiment.

In this embodiment, the offset information includes the first offset information and second offset information, for example. For the case where the relative offset address corresponding to the data storage area is directly used as the offset information, it is only needed to calculate the first check factor based on the storage area address, the basic attribute information, and the relative offset address (i.e., a type of offset information), or calculate the first check factor based on the storage area address, the basic attribute information, the relative offset address, and the second type identifier. These two methods are also applicable in this embodiment, which are not elaborated herein.

S610 includes constructing third variable data for the reference variable of the application, the third variable data including the second type identifier, the first offset information, and the second offset information, storing the first check factor in the attribute storage area of the data storage area, and sending the third variable data of the reference variable to the application.

For example, taking the reference variable is 4-byte data as an example, the data structure of the third variable data of the reference variable constructed in step S610 is shown in FIG. 5.

As shown in FIG. 5, the second type identifier can be recorded in bits 31 and 30 of the reference variable, the first offset information can be recorded in bits 16 to 29 of the reference variable, and the second offset information can be recorded in bits 13 to 15 of the reference variable, so that bits 13 to 29 constitute the storage area offset information. The offset range area of the reference variable belongs to an address range that can be modified by the application, which is not limited in the present application.

On the basis of FIG. 6, an implementation of the data access method of the present application is introduced.

FIG. 7 shows a flowchart of yet another data access method according to an embodiment of the present application. This embodiment is applied to a runtime system based on a WebAssembly virtual machine. The method of this embodiment may include steps S701 to S702.

S701 includes obtaining a data access instruction initiated by an application.

- 5 The data access instruction carries first variable data of a reference variable, and the first variable data of the reference variable is used to indicate a data address of target data to be accessed. For example, the first variable data of the reference variable includes a data offset address corresponding to the target data to be accessed.

10 It can be understood that the first variable data of the reference variable sent by the application is different from the above second variable data or the third variable data of the reference variable sent to the application by the runtime system. For example, the offset range area in the first variable data of the reference variable may record the data offset address corresponding to the target data to be accessed, while the offset range area in the above reference variable sent to the application by the runtime system may be a default value.

- 15 S702 includes obtaining a storage area type identifier from the first variable data of the reference variable.

S703 includes obtaining, in response to that the storage area type identifier is a first type identifier, the first check factor and the address index from the first variable data of the reference variable.

- 20 For example, taking FIG. 4 as an example, in the data structure of the reference variable shown in FIG. 4, if bit 31 of the reference variable is 1 (in practice, it can be set to other target values according to actual needs), it is determined that the storage area type identifier of the data storage area of the application is the first type identifier. Accordingly, the first check factor can be obtained from bits 23 to 30 of the reference variable, and the address index can be obtained
25 from bits 16 to 22.

S704 includes determining the storage area address of the data storage area corresponding to the application based on the address index.

- For example, based on the address index, the storage area address corresponding to the address index is queried from an address storage table, to obtain the storage area address of the data
30 storage area.

S705 includes obtaining basic attribute information from the data storage area of the application

based on the storage area address.

S706 includes calculating a second check factor based on the storage area address of the data storage area, the basic attribute information, the address index, and the first type identifier.

For example, a checksum calculation algorithm, such as CRC, is used to calculate the second check factor based on the storage area address, the basic attribute information, the address index, and the first type identifier.

It can be understood that step S706 is described by taking one implementation of calculating the second check factor when the storage area type identifier is the first type identifier as an example. However, if the first check factor is calculated based on the storage address of the data storage area and the basic attribute information when the storage area type identifier is the first type identifier, then in the present application, the second check factor can also be calculated based on the storage area address of the data storage area and the basic attribute information.

Similarly, if the first check factor is calculated based on the storage area address, the basic attribute information, and the first type identifier, then the second check factor can be calculated based on the storage area address, the basic attribute, and the first type identifier. If the first check factor is calculated based on the storage area address, the basic attribute information, and the address index, then the second check factor can also be calculated based on the storage area address, the basic attribute information, and the address index.

S707 includes obtaining, in response to that the storage area type identifier is the second type identifier, the first offset information and the second offset information recorded in the first variable data of the reference variable.

For example, with reference to the data structure of the reference variable shown in FIG. 5, after the numbers of bits occupied by the first offset information and the second offset information in the reference variable are respectively determined, the first offset information and the second offset information can be obtained from the corresponding bits.

S708 includes determining the relative offset address corresponding to the data storage area based on the first offset information and the second offset information.

For example, the relative offset address can be obtained by adding the first offset information to a product of the second offset information and the target number of bytes.

The relative offset address is a relative offset address of the storage area address of the data storage area with respect to a total space start address, where the total space start address is a

start address of a total storage space of the runtime system.

It can be understood that this embodiment is described by taking the example that the reference variable records the first offset information and the second offset information. It can be understood that if the reference variable records offset information such as the relative offset address, the offset information such as the relative offset address can be directly obtained from the reference variable.

S709 includes determining the storage area address of the data storage area based on the relative offset address.

For example, based on the relative offset address and the start address of the total storage space of the runtime system, the storage area address of the data storage area can be determined.

S710 includes obtaining basic attribute information and the first check factor from the data storage area based on the storage area address of the data storage area.

S711 includes calculating a second check factor based on the storage area address of the data storage area, the basic attribute information, the first offset information, the second offset information, and the second type identifier.

It can be understood that step S711 is described by taking one implementation of calculating the second check factor when the storage type identifier is the second type identifier as an example. It can be understood that when the storage type identifier is the second type identifier, the first check factor can alternatively be calculated based on the storage area address of the data storage area and the basic attribute information. Accordingly, the second check factor can be calculated based on the storage area address of the data storage area and the basic attribute information.

Similarly, the second check factor can alternatively be calculated based on the storage area address of the data storage area, the basic attribute information, and the second type identifier; or, the second check factor can be calculated based on the storage area address, the basic attribute information, the first offset information, and the second offset information.

In practice, if the first offset information and the second offset information are replaced by offset information such as the relative offset address, in step S711, the second check factor can alternatively be calculated based on the storage area address, the basic attribute information, the offset information (e.g., the relative offset address), and the second type identifier; or, the second check factor can be calculated based on the storage area address, the basic attribute information, and the offset information (e.g., the relative offset address).

S712 includes obtaining, in response to that the first check factor is the same as the second check factor, the target data from the data storage area based on the first variable data of the reference variable and the storage area address.

For example, based on the data offset address in the first variable data of the reference variable and the storage area address, the data storage address of the target data in the data storage area is determined, and the target data is obtained from the data storage area based on the data storage address.

It can be understood that after the target data is obtained, the target data can be sent to the application.

The present application further provides a data access apparatus corresponding to the data access method according to the embodiments of the present application.

FIG. 8 shows a schematic structural diagram of a data access apparatus according to an embodiment of the present application. The apparatus of this embodiment is applied to a runtime system based on a WebAssembly virtual machine, and the apparatus includes:

an instruction obtaining unit 801, configured to obtain a data access instruction initiated by an application, the data access instruction carrying first variable data of a reference variable, the first variable data of the reference variable being used to indicate a data address of target data to be accessed;

an address determining unit 802, configured to determine a storage area address of a data storage area corresponding to the application based on storage area identification information recorded in the first variable data of the reference variable, the storage area identification information being used to identify the data storage area corresponding to the application;

a factor obtaining unit 803, configured to obtain a first check factor associated with the first variable data of the reference variable, the first check factor being calculated by the runtime system based on the storage area address of the data storage area during a process of generating variable data of the reference variable for the application;

a first factor calculation unit 804, configured to calculate a second check factor based on the storage area address;

a data obtaining unit 805, configured to obtain, in response to that the first check factor is the same as the second check factor, the target data from the data storage area based on the first variable data of the reference variable and the storage area address.

In a feasible implementation, the data access apparatus may further include: an attribute obtaining unit, configured to, obtain, after the address determining unit determines the storage area address of the data storage area corresponding to the application, basic attribute information recorded in the data storage area based on the storage area address, the basic attribute information including attribute information associated with the data storage area and the application;

accordingly, the first check factor obtained by the factor obtaining unit is calculated based on the storage area address of the data storage area and the basic attribute information;

the first factor calculation unit is specifically configured to calculate the second check factor based on the basic attribute information recorded in the data storage area and the storage area address.

In a feasible implementation, the factor obtaining unit includes at least one of the following:

a first factor obtaining subunit, configured to obtain the first check factor from the first variable data of the reference variable;

a second factor obtaining subunit, configured to obtain the first check factor associated with the reference variable from an attribute storage area of the data storage area, the attribute storage area being a storage area for storing the basic attribute information.

In yet another feasible implementation, the apparatus further includes: a type obtaining unit, configured to obtain, after the instruction obtaining unit obtains the data access instruction initiated by the application, a storage area type identifier from the first variable data of the reference variable, the storage area type identifier being one of a first type identifier and a second type identifier, the first type identifier indicating that a storage capacity of the data storage area corresponding to the application is not less than a set threshold, the second type identifier indicating that the storage capacity of the data storage area is less than the set threshold;

accordingly, the first check factor is calculated based on the storage area address of the data storage area, the basic attribute information, and the storage area type identifier;

the first factor calculation unit is specifically configured to calculate the second check factor based on the basic attribute information recorded in the data storage area, the storage area address, and the storage area type identifier.

In yet another feasible implementation, the first factor obtaining subunit is specifically

configured to obtain, in response to that the storage area type identifier is the first type identifier, the first check factor from the first variable data of the reference variable;

the second factor obtaining subunit is specifically configured to obtain, in response to that the storage area type identifier is the second type identifier, the first check factor associated with the reference variable from the attribute storage area of the data storage area.

In yet another feasible implementation, the address determining unit includes:

a first address determining subunit, configured to obtain, in response to that the storage area type identifier is the first type identifier, an address index recorded in the first variable data of the reference variable, and determine the storage area address of the data storage area corresponding to the application based on the address index, the address index being used to indicate a storage location of the storage area address of the data storage area in an address storage table;

a second address determining subunit, configured to obtain, in response to that the storage area type identifier is the second type identifier, offset information recorded in the first variable data of the reference variable, and determine the storage area address of the data storage area based on the offset information, the offset information being used to characterize a relative offset address of the storage area address of the data storage area with respect to a total space start address, the total space start address being a start address of a total storage space of the runtime system.

In yet another feasible implementation, the second address determining subunit includes:

an offset obtaining subunit, configured to obtain, in response to that the storage area type identifier is the second type identifier, first offset information and second offset information recorded in the first variable data of the reference variable, the first offset information being a remainder obtained by dividing the relative offset address corresponding to the storage area address of the data storage area by a target number of bytes, the second offset information being a quotient obtained by dividing the relative offset address by the target number of bytes;

an offset determining subunit, configured to determine the relative offset address corresponding to the data storage area based on the first offset information and the second offset information;

an address determining subunit, configured to determine the storage area address of the data storage area based on the relative offset address.

In yet another feasible implementation, the first factor calculation unit is specifically configured

to calculate the second check factor based on the data storage area identifier, the basic attribute information, and the storage area address.

In yet another feasible implementation, the first variable data of the reference variable obtained by the instruction obtaining unit includes a data offset address of the target data in the data storage area corresponding to the application;

the data obtaining unit is specifically configured to obtain, in response to that the first check factor is the same as the second check factor, the target data from the data storage area based on the data offset address and the storage area address.

In yet another feasible implementation, the apparatus further includes:

an application obtaining unit, configured to obtain, before the instruction obtaining unit obtains the data access instruction initiated by the application, a storage area request initiated by the application, the storage area request being used to request allocation of a storage space of a target storage capacity for the application;

a storage area determining unit, configured to create the data storage area for the application based on the target storage capacity, and determine the storage area type identifier corresponding to the data storage area, where an attribute storage area of the data storage area records the basic attribute information associated with the data storage area and the application, where in response to that the target storage capacity is not less than a set threshold, the storage area type identifier is a first type identifier, and in response to that the target storage capacity is less than the set threshold, the storage area type identifier is a second type identifier;

an identification determining unit, configured to determine storage area identification information of the data storage area based on the storage area address of the data storage area;

a second factor calculation unit, configured to calculate the first check factor based on the storage area address and the basic attribute information;

a first variable constructing unit, configured to construct, in response to that the target storage capacity is not less than the set threshold, second variable data for the reference variable of the application, the second variable data including the storage area type identifier, the first check factor, and the storage area identification information;

a second variable constructing unit, configured to construct, in response to that the target storage capacity is less than the set threshold, third variable data for the reference variable of the application, the third variable data including the storage area type identifier and the storage area

identification information, and store the first check factor in the attribute storage area of the data storage area.

In yet another feasible implementation, the identification determining unit includes:

5 an index determining subunit, configured to store, in response to that the target storage capacity is not less than the set threshold, the storage area address of the data storage area in an address storage table, and determine an address index of the storage area address in the address storage table, the address index being used to indicate a storage location of the storage area address in the address storage table;

10 an offset determining subunit, configured to determine, in response to that the target storage capacity is less than the set threshold, offset information corresponding to the storage area address of the data storage area, the offset information being used to characterize a relative offset address of the storage area address of the data storage area with respect to a total space start address, the total space start address being a start address of a total storage space of the runtime system;

15 accordingly, the first variable constructing unit is specifically configured to construct, in response to that the target storage capacity is not less than the set threshold, the second variable data for the reference variable of the application, the second variable data including the storage area type identifier, the first check factor, and the address index;

20 the second variable constructing unit is specifically configured to construct, in response to that the target storage capacity is less than the set threshold, the third variable data for the reference variable of the application, the third variable data including the storage area type identifier and the offset information.

In yet another feasible implementation, the offset determining subunit includes:

25 an address determining subunit, configured to determine, in response to that the target storage capacity is less than the set threshold, the relative offset address of the storage area address of the data storage area with respect to the total space start address;

an offset splitting subunit, configured to determine a quotient and a remainder obtained by dividing the relative offset address by a target number of bytes, determine the remainder as the first offset information, and determine the quotient as the second offset information;

30 the second variable constructing unit is specifically configured to construct, in response to that the target storage capacity is less than the set threshold, the third variable data for the reference

variable of the application, the third variable data including the storage area type identifier, the first offset information, and the second offset information.

It can be understood that in the present application, the terms “first,” “second,” “third,” “fourth,” etc. (if any) in the specification and claims and the above drawings are used to distinguish similar parts, and are not necessarily used to describe a specific order or sequence. It should be understood that data used in this way can be interchanged under appropriate circumstances, so that the embodiments of the present application described herein can be implemented in other orders than those illustrated or described herein.

It should be noted that the various embodiments in this specification are described in a progressive manner, and each embodiment focuses on the differences from other embodiments, and the same or similar parts between the various embodiments can be referred to each other. At the same time, the features described in the various embodiments in this specification can be replaced or combined with each other, so that those skilled in the art can implement or use the present application. For the apparatus embodiments, since they are substantially similar to the method embodiments, the description is relatively simple, and the relevant parts can be referred to the description of the method embodiments.

Finally, it should also be noted that in this document, relational terms such as first and second are only used to distinguish one entity or operation from another entity or operation, and do not necessarily require or imply any actual such relationship or order between these entities or operations. Moreover, the term “comprising,” “including,” or any other variation thereof is intended to cover non-exclusive inclusion, so that a process, method, article, or device that includes a series of elements includes not only those elements, but also other elements not explicitly listed, or elements inherent to such process, method, article, or device. Without further limitations, an element defined by the phrase “comprising one...” does not exclude the presence of additional identical elements in the process, method, article, or device that includes the element.

The above description of the disclosed embodiments enables those skilled in the art to implement or use the present application. Various modifications to these embodiments will be obvious to those skilled in the art, and the general principles defined herein may be implemented in other embodiments without departing from the spirit or scope of the present application. Therefore, the present application will not be limited to the embodiments shown herein, but will conform to the widest scope consistent with the principles and novel features disclosed herein.

The above are only preferred embodiments of the present application. It should be pointed out that for those of ordinary skill in the art, several improvements and modifications can be made without departing from the principles of the present application, and these improvements and modifications should also be regarded as within the protection scope of the present application.

CLAIMS

1. A data access method, applied to a runtime system based on a WebAssembly virtual machine, the method comprising:

obtaining a data access instruction initiated by an application, the data access instruction carrying first variable data of a reference variable, the first variable data of the reference variable being used to indicate a data address of target data to be accessed;

determining a storage area address of a data storage area corresponding to the application based on storage area identification information recorded in the first variable data of the reference variable, the storage area identification information being used to identify the data storage area corresponding to the application;

obtaining a first check factor associated with the reference variable, the first check factor being calculated by the runtime system based on the storage area address of the data storage area during a process of generating variable data corresponding to the reference variable for the application;

calculating a second check factor based on the storage area address;

obtaining, in response to that the first check factor is the same as the second check factor, the target data from the data storage area based on the first variable data of the reference variable and the storage area address.

2. The data access method according to claim 1, wherein after determining the storage area address of the data storage area corresponding to the application, the method further comprises:

obtaining basic attribute information recorded in the data storage area based on the storage area address, the basic attribute information comprising attribute information associated with the data storage area and the application;

the first check factor is calculated based on the storage area address of the data storage area and the basic attribute information;

calculating the second check factor based on the storage area address comprises:

calculating the second check factor based on the basic attribute information recorded in the data storage area and the storage area address.

3. The data access method according to claim 2, wherein after obtaining the data access instruction initiated by the application, the method further comprises:

obtaining a storage area type identifier from the first variable data of the reference variable, the storage area type identifier being one of a first type identifier and a second type identifier, the first type identifier indicating that a storage capacity of the data storage area corresponding to the application is not less than a set threshold, the second type identifier indicating that the storage capacity of the data storage area is less than the set threshold,

wherein the first check factor is calculated based on the storage area address of the data storage area, the basic attribute information, and the storage area type identifier;

calculating the second check factor based on the basic attribute information recorded in the data storage area and the storage area address comprises:

calculating the second check factor based on the basic attribute information recorded in the data storage area, the storage area address, and the storage area type identifier.

4. The data access method according to claim 3, wherein obtaining the first check factor associated with the reference variable comprises:

obtaining, in response to that the storage area type identifier is the first type identifier, the first check factor from the first variable data of the reference variable;

obtaining, in response to that the storage area type identifier is the second type identifier, the first check factor associated with the reference variable from an attribute storage area of the data storage area.

5. The data access method according to claim 3, wherein determining the storage area address of the data storage area corresponding to the application based on the storage area identification information recorded in the first variable data of the reference variable comprises:

obtaining, in response to that the storage area type identifier is the first type identifier, an address index recorded in the first variable data of the reference variable, and determining the storage area address of the data storage area corresponding to the application based on the address index, the address index being used to indicate a storage location of the storage area address of the data storage area in an address storage table;

obtaining, in response to that the storage area type identifier is the second type identifier, offset information recorded in the first variable data of the reference variable, and determining the storage area address of the data storage area based on the offset information, the offset information being used to characterize a relative offset address of the storage area address of the data storage area with respect to a total space start address, the total space start address

being a start address of a total storage space of the runtime system.

6. The data access method according to claim 5, wherein obtaining the offset information recorded in the first variable data of the reference variable and determining the storage area address of the data storage area based on the offset information comprises:

- 5 obtaining first offset information and second offset information recorded in the first variable data of the reference variable, the first offset information being a remainder obtained by dividing the relative offset address by a target number of bytes, the second offset information being a quotient obtained by dividing the relative offset address by the target number of bytes; determining the relative offset address corresponding to the data storage area based on the first
10 offset information and the second offset information;

determining the storage area address of the data storage area based on the relative offset address.

7. The data access method according to claim 1, wherein the first variable data of the reference variable comprises: a data offset address of the target data in the data storage area corresponding to the application;

- 15 obtaining, in response to that the first check factor is the same as the second check factor, the target data from the data storage area based on the first variable data of the reference variable and the storage area address comprises:

obtaining, in response to that the first check factor is the same as the second check factor, the target data from the data storage area based on the data offset address and the storage area
20 address.

8. The data access method according to claim 3, wherein before obtaining the data access instruction initiated by the application, the method further comprises:

obtaining a storage area request initiated by the application, the storage area request being used to request allocation of a storage space of a target storage capacity for the application;

- 25 creating the data storage area for the application based on the target storage capacity, and determining the storage area type identifier corresponding to the data storage area, wherein an attribute storage area of the data storage area records the basic attribute information associated with the data storage area and the application, wherein in response to that the target storage capacity is not less than a set threshold, the storage area type identifier is the first type identifier,
30 and in response to that the target storage capacity is less than the set threshold, the storage area type identifier is the second type identifier;

determining storage area identification information of the data storage area based on the storage area address of the data storage area;

calculating the first check factor based on the storage area address, the basic attribute information, and the storage area type identifier;

- 5 constructing, in response to that the target storage capacity is not less than the set threshold, second variable data for the reference variable of the application, the second variable data comprising the storage area type identifier, the first check factor, and the storage area identification information;

- 10 constructing, in response to that the target storage capacity is less than the set threshold, third variable data for the reference variable of the application, the third variable data comprising the storage area type identifier and the storage area identification information, and storing the first check factor in the attribute storage area of the data storage area.

- 15 9. The data access method according to claim 8, wherein determining the storage area identification information of the data storage area based on the storage area address of the data storage area comprises:

storing, in response to that the target storage capacity is not less than the set threshold, the storage area address of the data storage area in an address storage table, and determining an address index of the storage area address in the address storage table, the address index being used to indicate a storage location of the storage area address in the address storage table;

- 20 determining, in response to that the target storage capacity is less than the set threshold, offset information corresponding to the storage area address of the data storage area, the offset information being used to characterize a relative offset address of the storage area address of the data storage area with respect to a total space start address, the total space start address being a start address of a total storage space of the runtime system;

- 25 constructing, in response to that the target storage capacity is not less than the set threshold, the second variable data for the reference variable of the application, the second variable data comprising the storage area type identifier, the first check factor, and the storage area identification information, comprises:

- 30 constructing, in response to that the target storage capacity is not less than the set threshold, the second variable data for the reference variable of the application, the second variable data comprising the storage area type identifier, the first check factor, and the address index;

constructing, in response to that the target storage capacity is less than the set threshold, the third variable data for the reference variable of the application, the third variable data comprising the storage area type identifier and the storage area identification information, comprises:

- 5 constructing, in response to that the target storage capacity is less than the set threshold, the third variable data for the reference variable of the application, the third variable data comprising the storage area type identifier and the offset information.

- 10 10. The data access method according to claim 9, wherein determining, in response to that the target storage capacity is less than the set threshold, the offset information corresponding to the storage area address of the data storage area comprises:

determining, in response to that the target storage capacity is less than the set threshold, the relative offset address of the storage area address of the data storage area with respect to the total space start address;

- 15 determining a quotient and a remainder obtained by dividing the relative offset address by a target number of bytes, determining the remainder as first offset information, and determining the quotient as second offset information;

constructing, in response to that the target storage capacity is less than the set threshold, the third variable data for the reference variable of the application, the third variable data comprising the storage area type identifier and the offset information, comprises:

- 20 constructing, in response to that the target storage capacity is less than the set threshold, the third variable data for the reference variable of the application, the third variable data comprising the storage area type identifier, the first offset information, and the second offset information.

- 25 11. A data access apparatus, applied to a runtime system based on a WebAssembly virtual machine, the apparatus comprising:

an instruction obtaining unit, configured to obtain a data access instruction initiated by an application, the data access instruction carrying first variable data of a reference variable, the first variable data of the reference variable being used to indicate a data address of target data to be accessed;

- 30 an address determining unit, configured to determine a storage area address of a data storage area corresponding to the application based on storage area identification information recorded

in the first variable data of the reference variable, the storage area identification information being used to identify the data storage area corresponding to the application;

a factor obtaining unit, configured to obtain a first check factor associated with the reference variable, the first check factor being calculated by the runtime system based on the storage area address of the data storage area during a process of generating variable data corresponding to the reference variable for the application;

a first factor calculation unit, configured to calculate a second check factor based on the storage area address;

a data obtaining unit, configured to obtain, in response to that the first check factor is the same as the second check factor, the target data from the data storage area based on the first variable data of the reference variable and the storage area address.