TASK PROCESSING METHOD AND SCHEDULING DEVICE FOR PRIVACY-PRESERVING COMPUTATION SYSTEM, AND STORAGE MEDIUM

TECHNICAL FIELD

[0001] This specification relates to the field of privacy-preserving computation technologies, and in particular, to a task processing method and scheduling device for a privacy-preserving computation system, and a storage medium.

BACKGROUND

[0002] Currently, many application scenarios have a requirement for privacy-preserving computation. With development of privacy-preserving computation technologies, there are currently a plurality of privacy-preserving computation technologies, and development frameworks for different privacy-preserving computation technologies are incompatible with each other.

[0003] When a privacy-preserving computation task is deployed for a specific application scenario, an operator needs to select a proper privacy-preserving computation technology from a plurality of privacy-preserving computation technologies based on a requirement of an actual application scenario, and performs development based on technical details of the privacy-preserving computation technology. The above-mentioned way imposes a relatively high requirement on the operator, and requires the operator to fully understand characteristics and technical details of each privacy-preserving computation technology, and a development cycle of a privacy-preserving computation task is relatively long. In addition, because the development frameworks for different privacy-preserving computation technologies are incompatible with each other, one application scenario can only select one privacy-preserving computation technology, which lacks flexibility.

[0004] It should be noted that, content of the above-mentioned related technology is merely information known to the inventor, and neither means that the information has entered a public

domain before an application date of this specification, nor means that the information can become a conventional technology of this specification.

SUMMARY

[0005] This specification provides a task processing method and scheduling device for a privacy-preserving computation system, and a storage medium, to avoid at least one of the above-mentioned technical problems.

[0006] According to a first aspect, this specification provides a task processing method for a privacy-preserving computation system. The privacy-preserving computation system includes a scheduling device and a computing device set that supports a plurality of privacy-preserving computation technologies, the computing device set includes a plurality of computing devices, and each computing device supports at least one of the plurality of privacy-preserving computation technologies. The method is applied to the scheduling device, and the method includes: obtaining task information of a target task, where the target task indicates to perform privacy-preserving computation in the computing device set; determining, based on the task information, M operators required for executing the target task and a data source required by each operator, where M is an integer greater than or equal to 1; and determining, from the computing device set based on the M operators and the data source required by each operator, a target device subset applicable to the target task, and controlling the target device subset to execute the target task to obtain a task result corresponding to the target task.

[0007] According to a second aspect, this specification provides a scheduling device for a privacy-preserving computation system. The privacy-preserving computation system further includes a computing device set that supports a plurality of privacy-preserving computation technologies, the computing device set includes a plurality of computing devices, and each computing device supports at least one of the plurality of privacy-preserving computation technologies. The scheduling device includes: at least one storage medium, storing at least one instruction set to perform task processing; and at least one processor, communicatively connected to the at least one storage medium, where when running, the at least one processor reads the at least one instruction set, and performs the following operations based on an indication of the at least one instruction set: obtaining task information of a target task, where the target task indicates

to perform privacy-preserving computation in the computing device set; determining, based on the task information, M operators required for executing the target task and a data source required by each operator, where M is an integer greater than or equal to 1; and determining, from the computing device set based on the M operators and the data source required by each operator, a target device subset applicable to the target task, and controlling the target device subset to execute the target task to obtain a task result corresponding to the target task.

[0008] According to a third aspect, this specification provides a computer-readable non-transitory storage medium. The computer-readable non-transitory storage medium stores at least one instruction set, and when the at least one instruction set is executed by at least one processor, the method according to the first aspect is implemented.

[0009] It can be learned from the above-mentioned technical solutions that, according to the task processing method and scheduling device for the privacy-preserving computation system, and the storage medium that are provided in this specification, when task information is obtained, a scheduling engine determines each operator for executing a target task and a data source required by each operator, and determines, from a computing device set with reference to each operator and each data source, a target device subset that is applicable to executing the target task, to control the target device subset to complete the target task to obtain a task result. On the one hand, an operator only needs to specify each operator and the data source required by each operator in the task information, to implement adaptive scheduling of a proper privacy-preserving computation technology, without having to consider implementation logic of the privacy-preserving computation technology in detail, and without having to perform encoding implementation of the privacy-preserving computation technology. On the other hand, when each operator corresponding to the target task supports a plurality of privacy-preserving computation technologies, the scheduling engine can further implement hybrid scheduling of different privacy-preserving computation technologies.

[0010] Other functions of the task processing method and scheduling device for the privacy-preserving computation system, and the storage medium that are provided in this specification are partially listed in the following descriptions. Based on the descriptions, content described by using the following numbers and examples is clear to those of ordinary skill in the art. Creative aspects of the task processing method and scheduling device for the privacy-preserving computation system, and the storage medium that are provided in this specification can be fully explained by

practice or by using the method, apparatus, and combination described in the following detailed examples.

BRIEF DESCRIPTION OF DRAWINGS

[0011] To describe the technical solutions in the embodiments of this specification more clearly, the following briefly describes the accompanying drawings needed for describing the embodiments. Clearly, the accompanying drawings in the following descriptions show merely some embodiments of this specification, and those of ordinary skill in the art can still derive other drawings from these accompanying drawings without creative efforts.

[0012] FIG. 1 is a schematic diagram of a privacy-preserving computation system and an application scenario thereof according to an embodiment of this specification;

[0013] FIG. 2 is a diagram of a hardware structure of a scheduling device according to an embodiment of this specification;

[0014] FIG. 3 is a schematic flowchart of a task processing method according to an embodiment of this specification;

[0015] FIG. 4 is a schematic diagram of a principle of a task processing method according to an embodiment of this specification;

[0016] FIG. 5 is a directed acyclic graph according to an embodiment of this specification; and

[0017] FIG. 6 is a schematic diagram of a principle of a task processing method according to another embodiment of this specification.

DESCRIPTION OF EMBODIMENTS

[0018] Example embodiments are described in detail here, and examples of the example embodiments are presented in the accompanying drawings. When the following description relates to the accompanying drawings, unless otherwise indicated, same numbers in different accompanying drawings represent same or similar elements. Implementations described in the following example embodiments do not represent all implementations consistent with this specification. On the contrary, the embodiments are merely examples of apparatuses and methods that are described in the appended claims in detail and consistent with some aspects of this specification.

[0019] It should be understood that the terms "include" and "have" in the embodiments of this specification and any variants thereof are intended to cover the non-exclusive inclusion, for example, a product or device that includes a series of components are not necessarily limited to those components that are clearly listed, but may include other components that are not clearly listed or inherent to such product or device.

[0020] In the embodiments of this specification, the term "and/or" describes an association relationship between associated objects and indicates that three relationships may exist. For example, "A and/or B" may indicate the following three cases: Only A exists, both A and B exist, and only B exists. The character "/" generally indicates an "or" relationship between the associated objects.

[0021] In the embodiments of this specification, the term "a plurality of" refers to two or more, and other quantifiers are similar to this term.

[0022] The terms "first", "second", "third", etc. in this specification are used to distinguish between similar or same objects or entities, and do not necessarily mean limiting a specific order or sequence, unless otherwise indicated. It should be understood that the terms used in such a way is interchangeable in appropriate cases, for example, can be implemented in an order other than those illustrated or described in the embodiments of this specification.

[0023] The term "unit/module" used in this specification means any known or later developed hardware, software, firmware, artificial intelligence, fuzzy logic, or a combination of hardware code or/and a combination of software code, and can perform functions related to the element.

[0024] To facilitate readers' understanding of this specification, at least some terms in this specification are explained below:

[0025] Privacy-preserving computation refers to a set of a series of technologies and methods, and these technologies and methods are intended to protect privacy of individual or sensitive data in data processing and analysis processes, and may be referred to as privacy-preserving computation technologies (which may be referred to as computing technologies for short). These privacy-preserving computation technologies allow data to be analyzed and processed without being exposed, so that a data owner (namely, a participant of privacy-preserving computation) can obtain valuable insights from the data without disclosing privacy information.

[0026] The privacy-preserving computation technologies include but are not limited to secure multi-party computation (MPC), trusted execution environment (TEE), federated learning (FL),

etc.

[0027] The secure multi-party computation is a key technology in the field of privacy-preserving computation, allowing a plurality of participants to cooperate to complete a privacy-preserving computation task without revealing their respective data. This makes secure multi-party computation very useful in scenarios in which data privacy of each party needs to be protected, such as in the fields of financial services, medical data analysis, supply chain management, privacy-friendly data mining, etc.

[0028] The trusted execution environment is a secure area, and can provide an isolated execution space to ensure privacy and security of code and data. An execution environment provides a secure environment for a sensitive operation to prevent tampering or unauthorized access to a master operating system.

[0029] The federated learning is a machine learning method, so that a plurality of participants can jointly establish a shared machine learning model without having to exchange their training data sets, thereby protecting privacy data of each participant.

[0030] A directed acyclic graph (DAG) is a directed graph without a loop, including a plurality of nodes and a plurality of edges with arrows. A direction of an arrow of an edge is used to represent at least one of a sequence relationship, a dependency relationship, or an operation relationship between two nodes connected by the edge.

[0031] A logistic regression (LR) algorithm is a statistical method for classification. A basic principle of the logistic regression algorithm is to predict a probability of occurrence of an event by using a linear combination of features, and is generally used to handle classification problems.

[0032] To facilitate readers' understanding of this specification, application scenarios of this specification are described.

[0033] The technical solutions provided in this specification are applicable to a scenario in which a plurality of participants need to perform privacy-preserving computation. The following uses a plurality of participants including a participant 1 and a participant 2, and uses a joint credit scenario, a marketing scenario, and a risk control scenario as examples for description.

[0034] For example, in a joint credit scenario, the participant 1 can be a banking institution and the participant 2 can be another credit institution. The banking institution and the another credit institution store credit data of their respective users. For example, a database of the banking institution stores credit data of users of the banking institution, and a database of the another credit

institution stores credit data of users of the another credit institution. The banking institution and the another credit institution each include a target user and store credit data corresponding to the target user. The target user can initiate a credit card issuance request to the banking institution. For the credit card issuance request, the banking institution and the another credit institution can perform privacy-preserving computation based on their respective stored credit data of the target user, to determine a user score of the target user, so that the banking institution determines, based on the user score, whether to send a credit card to the target user.

[0035] For example, in a marketing scenario, the participant 1 and the participant 2 can correspond to different e-commerce platforms. The databases of the two participants respectively store user information of users in their respective e-commerce platforms. For example, the databases of the participant 1 and the participant 2 respectively store user information corresponding to target users. The participant 1 and the participant 2 can perform privacy-preserving computation based on their respective stored user information of the target users to determine recommendation information for performing coupon recommendation for the target users, and send the recommendation information to the target users, to improve click rates and write-off rates of the target users.

[0036] For example, in a risk control scenario, the participant 1 can be a risk control service provider (e.g., a risk control platform), and the participant 2 can be a first risk control service user (e.g., a payment platform). The database of participant 1 can store historical risk control data of a second risk control service user, for example, a user portrait and consumption information of a user in a geographical area to which the second risk control service user belongs. The database of the participant 2 can store a user portrait and consumption information of a user in a geographical area to which the first risk control service user belongs. The participant 1 and the participant 2 can perform privacy-preserving computation based on their respective stored data, for example, when user data respectively stored by the participant 1 and the participant 2 do not leave the country, perform joint training based on user data of the first risk control service user and the second risk control service user to obtain a risk control model, so that the risk control model can provide a risk control service for the first risk control service user.

[0037] It should be understood that, the above-mentioned examples are merely used to describe application scenarios exemplarily, and cannot be understood as a limitation on the application scenarios. Those skilled in the art should understand that the privacy-preserving

computation can be further applied to a plurality of other fields or scenarios, such as financial services, medical data analysis, supply chain management, and privacy-friendly data mining.

[0038] Currently, privacy-preserving computation technologies are diverse, and commonly include MPC, TEE, FL, and the like. However, development frameworks for different privacy-preserving computation technologies are incompatible with each other, for example, the MPC and the TEE are incompatible with each other. In a related technology, an operator (such as a developer or an algorithm engineer) that deploys a privacy-preserving computation task needs to clearly perceive characteristics and underlying details of each privacy-preserving computation technology, select a proper privacy-preserving computation technology from various privacy-preserving computation technologies based on a development scenario and a requirement, and develop a target task based on the underlying details of the privacy-preserving computation technology.

[0039] The above-mentioned way imposes a relatively high requirement on the operator, and requires the operator to fully understand characteristics and technical details of each privacy-preserving computation technology, and a development cycle of a privacy-preserving computation task is relatively long. In addition, because the development frameworks for different privacy-preserving computation technologies are incompatible with each other, one application scenario can only select one privacy-preserving computation technology, which lacks flexibility.

[0040] To avoid at least one drawback in the above-mentioned related technology, this specification provides a technical concept from creative works: deploying a scheduling device and a computing device set in a privacy-preserving computation system. After obtaining task information of a target task, the scheduling device can determine, based on the task information, each operator required for executing the target task and a data source required by each operator, select, from the computing device set based on each operator and each data source, a target device subset that is applicable to executing the target task, and control the target device subset to execute the target task, to obtain a corresponding task result.

[0041] It can be learned from the above-mentioned technical concept that, in the technical solutions provided in this specification, the operator only needs to specify, in the task information, each operator and the data source required by each operator, to implement flexible and adaptive scheduling of a proper privacy-preserving computation technology, without having to consider implementation logic of the privacy-preserving computation technology, and without having to perform encoding implementation of the privacy-preserving computation technology. Further, the

above-mentioned technical concept can further support hybrid scheduling of different privacy-preserving computation technologies.

[0042] FIG. 1 is a schematic diagram of a privacy-preserving computation system and an application scenario thereof according to an embodiment of this specification. As shown in FIG. 1, the application scenario 100 includes a plurality of participants. For example, there are two participants, which are a participant 1 and a participant 2. The participant 1 and the participant 2 each correspond to a plurality of computing devices configured to perform privacy-preserving computation, for example, an MPC device, a TEE device, and an FL device. The participant 1 and the participant 2 also correspond to their respective databases. For example, the participant 1 has a database 1, configured to store a data source of a user belonging to the participant 1; and the participant 2 includes a database 2, configured to store a data source of a user belonging to the participant 2.

[0043] For ease of description, in this specification, a set of composed of computing devices of all participants in the application scenario 100 is referred to as a computing device set, and a set composed of computing devices of one participant is referred to as a computing device subset. As shown in FIG. 1, a set composed of computing devices of the participant 1 is referred to as a computing device subset 1, and a set composed of computing devices of the participant 2 is referred to as a computing device subset 2.

[0044] For the computing device subset corresponding to any participant, the computing device subset can include a plurality of computing devices, and each computing device supports at least one privacy-preserving computation technology. Therefore, one computing device subset can support a plurality of privacy-preserving computation technologies. In some embodiments, a plurality of privacy-preserving computation technologies supported by computing device subsets corresponding to different participants can be the same. For example, with reference to FIG. 1, both the computing device subset 1 corresponding to the participant 1 and the computing device subset 2 corresponding to the participant 2 support an MPC technology, an FL technology, and a TEE technology. In some embodiments, the plurality of privacy-preserving computation technologies supported by the computing device subsets corresponding to different participants can alternatively be different. For example, the computing device subset 1 corresponding to the participant 1 supports the MPC technology and the FL technology, and the computing device subset 2 corresponding to the participant 2 supports the FL technology and the TEE technology.

[0045] The privacy-preserving computation system provided in this specification includes the above-mentioned computing device set (e.g., a union set of the computing device subset 1 and the computing device subset 2 in FIG. 1) and a scheduling device. The scheduling device may schedule some of the computing devices in the computing device set based on task information of a target task, to control some of the computing devices to perform privacy-preserving computation.

[0046] The scheduling device can be an electronic device with a certain data processing capability. The task processing method provided in this specification can be performed on the scheduling device. In this case, the scheduling device can store data or instructions for performing the task processing method described in this specification, and can execute or be configured to execute the data or the instructions. The scheduling device can include a hardware device with a data information processing function and a program required for driving the hardware device to work.

[0047] With reference to FIG. 1, when the participant 1 and the participant 2 need to perform privacy-preserving computation, an operator can generate the task information of the target task based on the need of the privacy-preserving computation, and provide the task information for the scheduling device. After obtaining the task information, the scheduling device can trigger performing of the task processing method provided in this specification, to flexibly and adaptively select, in the computing device set, a target device subset applicable to the target task, and control the target device subset to execute the target task, to obtain a task result corresponding to the target task. During the privacy-preserving computation, neither the participant 1 nor the participant 2 leaks their respective data, to ensure security of their respective data.

[0048] FIG. 2 is a diagram of a hardware structure of a scheduling device 200 according to an embodiment of this specification. The scheduling device 200 can perform the task processing method described in this specification. The task processing method is described in another part of this specification.

[0049] As shown in FIG. 2, the scheduling device 200 can include at least one storage medium 203 and at least one processor 202. In some embodiments, the scheduling device 200 can further include a communication port 204 and an internal communication bus 201. The scheduling device 200 can further include an I/O component 205.

[0050] The internal communication bus 201 can be connected to different system components. For example, the internal communication bus 201 can be connected to the storage medium 203,

the processor 202, the communication port 204, and the I/O component 205.

[0051] The I/O component 205 supports input/output between the scheduling device 200 and another component.

[0052] The communication port 204 is used for data communication between the scheduling device 200 and the outside. For example, the communication port 204 can be used for data communication between the scheduling device 200 and a network. The communication port 204 can be a wired communication port, or can be a wireless communication port.

[0053] The storage medium 203 can include a data storage apparatus. The data storage apparatus can be a non-transitory storage medium, or can be a transitory storage medium. For example, the data storage apparatus can include one or more of a disk 2031, a read-only storage medium (ROM) 2032, or a random access storage medium (RAM) 2033. The storage medium 203 further includes at least one instruction set stored in the data storage apparatus. The instruction set includes computer program code, and the computer program code can include a program, a routine, an object, a component, a data structure, a process, a module, etc. for performing the task processing method provided in this specification.

[0054] The at least one processor 202 can be communicatively connected to the at least one storage medium 203. The at least one processor 202 is configured to execute the at least one instruction set. When the scheduling device 200 runs, the at least one processor 202 reads the at least one instruction set, and performs, based on an indication of the at least one instruction set, the task processing method provided in this specification. The processor 202 can perform all steps included in the task processing method. The processor 202 can be in a form of one or more processors. In some embodiments, the processor 202 can include one or more hardware processors, for example, a microcontroller, a microprocessor, a reduced instruction set computer (RISC), an application-specific integrated circuit (ASIC), an application-specific instruction set processor (ASIP), a central processing unit (CPU), a graphics processing unit (GPU), a physical processing unit (PPU), a microcontroller unit, a digital signal processor (DSP), a field programmable gate array (FPGA), an advanced RISC machine (ARM), a programmable logic device (PLD), any circuit, processor, or the like that can perform one or more functions, or any combination thereof.

[0055] To merely describe a problem, only one processor 202 is shown in the scheduling device 200 in the accompanying drawings. However, it should be noted that the scheduling device 200 in this specification can further include a plurality of processors. Therefore, the operations

and/or method steps disclosed in this specification can be performed by one processor or can be jointly performed by a plurality of processors. For example, if the processor 202 in the scheduling device 200 described in this specification performs step A and step B, it should be understood that step A and step B can be jointly or separately performed by two different processors 202 (e.g., a first processor performs step A, and a second processor performs step B, or a first processor and a second processor jointly perform steps A and B).

[0056] Referring to FIG. 3, FIG. 3 is a schematic flowchart of a task processing method according to an embodiment of this specification. As shown in FIG. 3, the task processing method includes the following S301-S303.

[0057] S301: Obtain task information of a target task, where the target task indicates to perform privacy-preserving computation in a computing device set.

[0058] The task processing method in this specification is applied to a privacy-preserving computation system. The privacy-preserving computation system can include a scheduling device and a computing device set that supports a plurality of privacy-preserving computation technologies. The computing device set includes a plurality of computing devices, and each computing device supports at least one of the plurality of privacy-preserving computation technologies.

[0059] In some embodiments, the computing device set is divided into K computing device subsets, K is an integer greater than 1, each computing device subset corresponds to one privacy-preserving computation participant (referred to as a participant), and each computing device subset supports a plurality of privacy-preserving computation technologies.

[0060] For example, with reference to FIG. 1, there are K=2 participants as an example, the computing device set is divided into two computing device subsets, which are a computing device subset 1 corresponding to a participant 1 and a computing device subset 2 corresponding to a participant 2. Both the computing device subset corresponding to the participant 1 and the computing device subset corresponding to the participant 2 support three privacy-preserving computation technologies: MPC, TEE, and FL.

[0061] In this specification, the target task can be a task used to trigger privacy-preserving computation in the computing device set. The task information can be related information used to describe the target task. The task information can reflect specific computation content involved in privacy-preserving computation, for example, what kind of processing needs to be performed on

data from which data source.

[0062] A way in which the scheduling device obtains the task information is not limited in this embodiment. For example, with reference to the above-mentioned examples and FIG. 1, it can be learned that the scheduling device can obtain the task information sent by an operator.

[0063] For example, in some embodiments, the scheduling device stores a second data table, and the second data table is a data table of a data source dimension. For example, the second data table includes meta information of a data source, such as an identity (ID), a name, a storage type, a participant to which the data source belongs, a storage location, a data type (e.g., plaintext or ciphertext), or a support operator list. An operator is an atomic computing logical unit of an algorithm. For example, content of the second data table can be shown in the following Table 1.

Table 1

Meta information	Data source 1	Data source 2	Data source 3
ID	1	2	3
Participant to which the data source belongs	Participant 1	Participant 2	
Data type	Plaintext (applicable to an MPC device)	Plaintext (applicable to an MPC device)	Ciphertext (applicable to a TEE device)
Support operator list	Operator A and operator B	Operator A and operator B	Operator C

[0064] Correspondingly, as shown in FIG. 4, the scheduling device can include a user interface and a scheduling engine. The user interface can include a visual operation interface. The scheduling engine can display the second data table to the operator through the visual operation interface. Therefore, the operator can query the second data table through the visual operation interface, to construct task information with reference to a current privacy-preserving computation requirement, and submit the task information to the scheduling engine, for example, submit the task information to the scheduling engine through the visual operation interface.

[0065] That is, in some embodiments, the scheduling device can provide a visual operation interface, and the operator can query the second data table through the visual operation interface, construct task information based on a corresponding data source, and initiate the task information to the scheduling engine, so that the scheduling engine obtains the task information.

[0066] In some other embodiments, as shown in FIG. 4, the scheduling device can further include an open programming interface, configured to invoke the second data table, so that the operator initiates task information to the scheduling engine by invoking.

[0067] That is, the task processing method in this specification can be executed by triggering through a visual operation, or can be executed by triggering through interface invoking, to improve flexibility and universality of application of the task processing method.

[0068] S302: Determine, based on the task information, M operators required for executing the target task and a data source required by each operator, where M is an integer greater than or equal to 1.

[0069] For example, with reference to the above-mentioned analysis, the task information can be created by the operator based on the second data table and submitted to the scheduling engine, and the task information includes each operator required for completing the target task and the data source required by each operator. As shown in FIG. 4, when receiving the task information submitted by the operator, the scheduling engine can parse the task information to obtain each operator and the data source required by each operator.

[0070] An expression form of the task information is not limited in this specification. For example, the task information can be a task script expressed in a script form. For example, the task information can correspond to a file of Python script writing. In some embodiments, the task information can alternatively be a task graph expressed in a graph form.

[0071] If the task information is a task script, S302 can be understood as: The scheduling engine parses the task script to obtain M operators and the data source required by each operator.

[0072] If the task information is a task graph, S302 can be understood as: The scheduling engine parses the task graph to obtain M operators and the data source required by each operator.

[0073] That is, in this embodiment, the scheduling engine can support the task information in different expression forms, to implement flexibility and diversity of task processing.

[0074] In some embodiments, the task graph can be a directed acyclic graph. One node in the directed acyclic graph represents one operator, and an edge between two operators represents one

or more of a sequence relationship between the two operators, a dependency relationship between the two operators, and an operation relationship between the two operators.

[0075] For example, with reference to FIG. 5, the directed acyclic graph includes a node 1 and a node 2 until a node N (N is an integer greater than 1). One node corresponds to one operator and a data source required by the operator. For example, the node 1 corresponds to one operator, and corresponds to a data source required by the operator. By analogy, details are not listed one by one herein. Correspondingly, the scheduling engine can determine N operators required for executing the target task and a data source required for each operator. In this case, M is N.

[0076] For example, the node 1 is an operator, and the operator can be intersection set computation; the node 2 is an operator, and the operator is an LR algorithm; and an edge L12 between the node 1 and the node 2 indicates that an operation for the intersection set computation is performed before the LR algorithm. In addition, as shown in FIG. 5, an operation symbol is marked on the edge L12, to represent an operation relationship by using the operation symbol. In FIG. 5, the operation symbol marked on the side L12 represents to sum a result of the intersection set operation and a result of the LR algorithm.

[0077] S303: Determine, from the computing device set based on the M operators and the data source required by each operator, a target device subset applicable to the target task, and control the target device subset to execute the target task to obtain a task result corresponding to the target task.

[0078] For example, when each operator and the data source required by each operator are determined, the scheduling engine can select, from the computing device set, computing devices applicable to executing the target task, and call the selected computing devices as the target device subset, to control the target device subset to execute the target task, thereby obtaining the task result.

[0079] For example, with reference to FIG. 1, the scheduling device can determine, from computing devices of the participant 1 and computing devices of the participant 2, computing devices configured to execute the target task. If it is determined that the computing devices applicable to executing the target task are MPC devices of the participant 1 and MPC devices of the participant 2, the MPC devices of both the participant 1 and the participant 2 are controlled to execute the target task, to obtain the task result.

[0080] It can be learned from the above-mentioned S301-S303 that, in this embodiment, when

the task information is obtained, the scheduling engine determines each operator for executing the target task and the data source required by each operator, and determines, from the computing device set with reference to each operator and each data source, the target device subset that is applicable to executing the target task, to control the target device subset to complete the target task to obtain the task result. On the one hand, the operator only needs to specify each operator and the data source required by each operator in the task information, to implement adaptive scheduling of a proper privacy-preserving computation technology, without having to consider implementation logic of the privacy-preserving computation technology in detail, and without having to perform encoding implementation of the privacy-preserving computation technology. On the other hand, when each operator corresponding to the target task supports a plurality of privacy-preserving computation technologies, the scheduling engine can further implement hybrid scheduling of different privacy-preserving computation technologies.

[0081] In some embodiments, S303 can include the following steps 1 to 3:

[0082] Step 1: Divide the target task into at least one subtask based on the M operators and the data source required by each operator, and determine a target operator and a target data source that correspond to each subtask.

[0083] For example, continuing to refer to FIG. 4, the scheduling engine can split the target task to obtain a plurality of subtasks. That is, the scheduling engine can divide a target task with a relatively large granularity to obtain subtasks with a relatively small granularity. Generally, there are a plurality of subtasks, that is, the scheduling engine can split a whole target task into a plurality of subtasks.

[0084] A way in which the scheduling engine divides the target task is not limited in this embodiment. For example, way 1: the scheduling engine can group one operator into one subtask; and way 2: the scheduling engine can group a plurality of operators into one subtask. Correspondingly, the scheduling engine determines subtasks in different ways, so that flexibility and diversity of determining the subtasks can be implemented. To facilitate readers' understanding, way 1 and way 2 are separately described.

[0085] Way 1: The scheduling engine can generate M subtasks based on the M operators, each subtask includes one operator, the operator included in each subtask is used as a target operator corresponding to the subtask, and a data source required by the operator included in the subtask is used as a target data source corresponding to the subtask.

[0086] For example, a quantity of subtasks is equal to a quantity of operators, and there is a one-to-one correspondence between the operators and the subtasks. In addition, for any operator, the subtask corresponding to the operator includes a data source used to implement the operator. To facilitate differentiation, the data source in the subtask can be referred to as a target data source corresponding to the subtask. That is, one subtask corresponds to one operator, and corresponds to one target data source. Correspondingly, each subtask is completed based on the target data source that has a correspondence with each subtask.

[0087] With reference to the above-mentioned examples and FIG. 5, it can be learned that the scheduling engine can split the directed acyclic graph into N operators and data sources respectively corresponding to the operators. For each operator in the N operators, the scheduling engine can determine the operator and the data source required by the operator as one subtask.

[0088] For example, the scheduling engine can determine intersection set computation and a data source required by the intersection set computation as one subtask; and the scheduling device can determine the LR algorithm and a data source required by the LR algorithm as a subtask. By analogy, details are not listed one by one herein.

[0089] It should be noted that, the scheduling engine groups one operator into one subtask in way 1, to rapidly split the target task. In addition, because a granularity of grouping is relatively small, execution and implementation of the subtask can be facilitated.

[0090] Way 2: The scheduling engine can group the M operators into at least one operator set based on privacy-preserving computation technologies to which the M operators are respectively applicable, the operators in each operator set are applicable to the same privacy-preserving computation technology, and the operators in different operator sets are applicable to different privacy-preserving computation technologies; and the scheduling engine separately generates one subtask based on each operator set, uses the operators in the operator set as target operators corresponding to the subtask, and uses data sources required by the operators in the operator set as target data sources corresponding to the subtask.

[0091] For example, the privacy-preserving computation technologies to which different operators are applicable may be different. The scheduling engine can classify each operator based on the privacy-preserving computation technology to which each operator can be applicable, to add operators applicable to the same privacy-preserving computation technology to the same operator set, thereby obtaining one or more operator sets.

[0092] Correspondingly, the scheduling engine can determine one operator set as one subtask. For any operator set, if the operator set includes a plurality of operators, the plurality of operators are used as target operators corresponding to the subtask, and the data sources respectively corresponding to the plurality of operators are used as target data sources corresponding to the subtask. If the operator set includes one operator, the operator is used as a target operator corresponding to the subtask, and the data source corresponding to the target operator is used as a target data source corresponding to the subtask.

[0093] For example, with continued reference to the above-mentioned examples and FIG. 5, if the intersection set computation and the LR algorithm are applicable to the same privacy-preserving computation technology, the scheduling device can determine the intersection set computation, the data source required by the intersection set computation, the LR algorithm, the data source required the LR algorithm as one subtask.

[0094] It should be noted that, the scheduling engine groups, in way 2, the operators applicable to the same privacy-preserving computation technology into one subtask, which is equivalent to classifying the operators applicable to the same privacy-preserving computation technology. With reference to the above-mentioned analysis, it can be learned that there is a support correspondence between a privacy-preserving computation technology and a computing device. Therefore, the scheduling engine can centrally and rapidly determine a target device subset in way 2.

[0095] In some embodiments, the scheduling engine stores a third data table, and the third data table is a data table of an operator dimension, and includes operators and privacy-preserving computation methods to which the operators are respectively applicable. For example, as shown in the following Table 2, the operators include an operator A, an operator B, and an operator C. The operator A is applicable to MPC but is not applicable to TEE; the operator B is applicable to both MPC and TEE; and the operator C is not applicable to MPC but is applicable to TEE

Table 2

Operator	MPC	TEE
A	Applicable	Not applicable
B	Applicable	Applicable
C	Not applicable	Applicable

[0096] Correspondingly, in way 2, the scheduling engine can determine each operator set by querying the third data table.

[0097] For example, the M operators include the operator A and the operator B. The scheduling engine can learn, by querying the third data table, that the operator A and the operator B each are applicable to MPC, and then the scheduling engine can determine the operator A and the operator B as one operator set. That is, the scheduling engine can determine the operator A, the operator B, a data source corresponding to the operator A, and a data source corresponding to the operator B as one subtask. The operator A and the operator B are used as target operators of the subtask, and the data source corresponding to the operator A and the data source corresponding to the operator B are used as target data sources of the subtask.

[0098] Step 2: Determine, from the computing device set based on the target operator and the target data source that correspond to each subtask, a target computing device applicable to the subtask, where target computing devices respectively applicable to the at least one subtask form the target device subset.

[0099] In some embodiments, step 2 can include: determining the target computing device from the computing device set based on the target operator, the target data source, a preset first data table, and a preset second data table.

[0100] The first data table is used to indicate a correspondence between an operator and a computing device. In some embodiments, the first data table includes a support operator list corresponding to different computing devices. For example, the first data table is a data table of a device dimension, and includes information about a plurality of dimensions of the computing devices. The plurality of dimensions can include but are not limited to an ID, a name, a participant to which a computing device belongs, a support privacy-preserving computation technology, a location, a support operator list, etc. For example, content of the first data table can be shown in the following Table 3.

Table 3

Dimension information	Computing device 1	Computing device 2	Computing device 3
ID	1	2	3

Dimension information	Computing device 1	Computing device 2	Computing device 3
Participant to which the computing device belongs	Participant 1	Participant 2	Participant 1
Support privacy-preserving computation technology	MPC	MPC	TEE
Support operator list	Operator A and operator B	Operator A and operator B	Operator A, operator B, and operator C

[0101] The second data table is used to indicate a correspondence between a data source and a computing device. For example, the content of the second data table can be shown in the above-mentioned Table 1. The second data table includes data types (for example, plaintext or ciphertext) of different data sources, and there is a correspondence between a data type of a data source and a computing device to which the data source is applicable. With reference to the above-mentioned analysis and Table 1, it can be learned that metadata information of a data source includes a data type. For different data types, a computing device to which the data source is applicable can be different. For example, the data type of the data source 1 is plaintext, and a computing device to which the data source 1 is applicable is an MPC device; and the data type of the data source 2 is ciphertext, and a computing device to which the data source 2 is applicable is a TEE device.

[0102] Correspondingly, in this embodiment, the scheduling engine determines the target computing device by using the first data table and the second data table. That is, when determining the target computing device, the scheduling engine considers both a factor of a device dimension and a factor of a data source dimension. Therefore, accuracy and reliability of determining the target computing device can be improved.

[0103] In some embodiments, that the scheduling engine determines the target computing device with reference to the first data table and the second data table can be understood as that the scheduling engine filters or selects the computing devices in the computing device set based on two dimensions (a target operator and a target data source) of the computing device applicable to

the subtask, to obtain the target computing device that is finally configured to execute the subtask.

[0104] For example, continuing to refer to FIG. 4, it can be learned that the scheduling engine can filter or select the computing devices in the computing device set by querying the first data table and the second data table, to determine the target computing device applicable to the subtask.

[0105] The filtering or selecting way of the scheduling engine is not limited in this specification. For example, way 1: the scheduling engine can first perform first filtering from a dimension of a computing device to which the target operator is applicable, and then perform second filtering from a dimension of a computing device to which a target data source is applicable, to obtain a final target computing device; way 2: the scheduling engine can first perform first filtering from the dimension of the computing device to which the target data source is applicable, and then perform second filtering from the dimension of the computing device to which the target operator is applicable, to obtain the final target computing device; and way 3: the scheduling engine can simultaneously perform filtering from both the dimension of the computing device to which the target operator is applicable and the dimension of the computing device to which the target data source is applicable, and determine the final target computing device based on a synchronous filtering result.

[0106] Correspondingly, the scheduling engine determines the target computing device in different ways, so that flexibility and diversity of determining the target computing device can be implemented. To facilitate readers' understanding, ways 1, 2, and 3 are separately described.

[0107] Way 1: The scheduling engine queries the first data table based on the target operator to determine, from the computing device set, a first candidate device subset to which the target operator is applicable. The computing devices in the first candidate device subset support a privacy-preserving computation technology to which the target operator is applicable. The scheduling engine queries the second data table based on the target data source to select, from the first candidate device subset, a computing device to which the target data source is applicable as the target computing device. The target computing device supports both the privacy-preserving computation technology to which the target operator is applicable and a privacy-preserving computation technology to which the target data source is applicable.

[0108] For example, with reference to the above-mentioned examples, the target operator includes an operator A and an operator B, a target data source corresponding to the operator A is a data source 1, a target data source corresponding to the operator B is a data source 2, and the

computing device set includes a computing device 1, a computing device 2, and a computing device 3.

[0109] Correspondingly, first, the scheduling engine can determine, from the computing device 1, the computing device 2, and the computing device 3 by querying the first data table, that the first candidate device subset includes the computing device 1, the computing device 2, and the computing device 3. Then, the scheduling engine determines, from the first candidate device subset (i.e., the computing device 1, the computing device 2, and the computing device 3), the computing device 1 and the computing device 2 as the target computing device by querying the second data table, for example, based on the computing devices (MPC devices, i.e., the computing device 1 and the computing device 2) to which the data source 1 and the data source 2 are applicable.

[0110] Way 2: The scheduling engine queries the second data table based on the target data source to determine, from the computing device set, a second candidate device subset to which the target data source is applicable. The computing devices in the second candidate device subset support a privacy-preserving computation technology to which the target data source is applicable. The scheduling engine queries the first data table based on the target operator to select, from the second candidate device subset, a computing device to which the target operator is applicable as the target computing device. The target computing device supports both the privacy-preserving computation technology to which the target operator is applicable and the privacy-preserving computation technology to which the target data source is applicable.

[0111] For an implementation principle of way 2, refer to the implementation principle of the above-mentioned way 1. To avoid tedious description, details are not described herein again.

[0112] Way 3: The scheduling engine queries the first data table based on the target operator to determine, from the computing device set, a first candidate device subset to which the target operator is applicable. The scheduling engine queries the second data table based on the target data source to determine, from the computing device set, a second candidate device subset to which the target data source is applicable. The scheduling engine determines an intersection set of the first candidate device subset and the second candidate device subset as the target computing device.

[0113] Similarly, to avoid tedious description, for an implementation principle of way 3, refer to the implementation principle of the above-mentioned way 1, details are not described herein again.

[0114] Step 3: Control each target computing device in the target device subset to execute a

subtask corresponding to the target computing device to obtain the task result corresponding to the target task.

[0115] Correspondingly, for any subtask, when the scheduling engine determines a target computing device configured to execute the subtask, the scheduling engine can control the target computing device to execute the subtask, to obtain the task result corresponding to the target task.

[0116] With reference to the above-mentioned analysis, the scheduling engine can divide a whole target task into subtasks of a finer granularity through step 1 to step 3, determine target computing devices configured to execute the subtasks, and control the target computing devices to execute the subtasks respectively corresponding to the target computing devices, to obtain the task result. In this way, hybrid scheduling of different privacy-preserving computation technologies can be implemented, that is, the target task is processed by using a combination of corresponding computing devices, thereby implementing hybrid and flexible application of resources of each privacy-preserving computation technology.

[0117] In some embodiments, step 3 can include the following step 31 and step 32:

[0118] Step 31: Separately send the at least one subtask to the target computing device corresponding to the at least one subtask, and obtain a subtask execution result fed back by the target computing device for the subtask of the target computing device.

[0119] For example, as shown in FIG. 6, there are two subtasks, which are a subtask 1 and a subtask 2. The target computing device corresponding to the subtask 1 is an MPC device, and the target computing device corresponding to the subtask 2 is a TEE device. The scheduling device separately sends the subtask 1 to the MPC device of the participant 1 and the MPC device of the participant 2, and sends the subtask 2 to the TEE device of the participant 1 and the TEE device of the participant 2.

[0120] Correspondingly, continuing to refer to FIG. 6, the MPC device of the participant 1 executes the subtask 1 to obtain an execution result of the subtask 1 and feed back the execution result to the scheduling device. The MPC device of the participant 2 executes the subtask 1 to obtain an execution result of the subtask 1 and feed back the execution result to the scheduling device. The MPC device of the participant 1 executes the subtask 2 to obtain an execution result of the subtask 2 and feed back the execution result to the scheduling device. The MPC device of the participant 2 executes the subtask 2 to obtain an execution result of the subtask 2 and feed back the execution result to the scheduling device.

[0121] It should be noted that a target data source in a subtask can be a name of the target data source, or can be specific data content of the target data source.

[0122] If the target data source is the name of the target data source, the target computing device can obtain specific data content of the target data source from a database based on the name of the target data source, and then perform corresponding computation, to obtain a subtask execution result.

[0123] For example, with reference to the above-mentioned examples and FIG. 6, after the subtask 1 is received, the MPC device of the participant 1 obtains, from a database 1, specific data content of the target data source in the subtask 1 based on the name of the target data source in the subtask 1, and executes the target operator in the subtask 1 based on the obtained specific data content, to obtain a corresponding execution result of the subtask 1.

[0124] If the target data source is the specific data content of the target data source, the target computing device can perform corresponding computation based on the specific data content, to obtain a subtask execution result.

[0125] For example, with reference to the above-mentioned examples and FIG. 6, after the subtask 1 is received, the MPC device of the participant 1 executes the target operator in the subtask 1 based on specific content of the target data source in the subtask 1, to obtain a corresponding execution result of the subtask 1.

[0126] Step 32: Determine, based on the subtask execution result corresponding to the at least one subtask, the task result corresponding to the target task.

[0127] Correspondingly, with reference to the above-mentioned examples, the scheduling device determines, based on the execution result of the subtask 1 fed back by the MPC device of the participant 1 for the subtask 1 and the execution result of the subtask 1 fed back by the MPC device of the participant 2 for the subtask 1, a feedback result 1 corresponding to the subtask 1. The scheduling device determines, based on an execution result of the subtask 2 fed back by the TEE device of the participant 1 for the subtask 2 and an execution result of the subtask 2 fed back by the TEE device of the participant 2 for the subtask 2, a feedback result 2 corresponding to the subtask 2. The scheduling device determines, based on the feedback result 1 and the feedback result 2, the task result corresponding to the target task.

[0128] In some embodiments, with reference to the above-mentioned examples and FIG. 4, the scheduling engine can output the task result through a user interface. Therefore, continuing to

refer to FIG. 4 and the above-mentioned examples, it can be learned that the scheduling engine submits each subtask to a target computing device corresponding to the subtask. An MPC device in the target computing device corresponds to the MPC device of the participant 1 and the MPC device of the participant 2 in the above-mentioned examples; and a TEE device in the target computing device corresponds to the TEE device of the participant 1 and the TEE device of the participant 2 in the above-mentioned examples. After separately sending each subtask to the target computing device corresponding to the subtask, the scheduling engine can obtain a subtask execution result fed back by each target computing device.

[0129] Correspondingly, as shown in FIG. 4, the scheduling engine summarizes the subtask execution results, to obtain a final execution result (i.e., the task result corresponding to the target task), and feeds back the final execution result to an operator. For example, with reference to the above-mentioned examples, the scheduling engine can output the task result through a visual operation interface.

[0130] In some other embodiments, for a scenario of the above-mentioned open programming interface, the scheduling engine can feed back the task result to an invoker through the open programming interface.

[0131] Based on the above-mentioned step 31 and step 32, it can be learned that in this embodiment, by controlling each target computing device to execute the subtask corresponding to the target computing device, parallel execution of subtasks can be implemented without mutual interference, thereby improving efficiency and reliability of task processing.

[0132] It should be noted that, the above-mentioned examples are merely used to describe examples of a possible implementation of the task processing method in this specification, and cannot be understood as a limitation on the implementation of the task processing method in this specification. For example, based on the above-mentioned technical concept, some of the above-mentioned technical features can be combined to obtain a new embodiment, a new technical feature can be added based on the above-mentioned examples to obtain a new embodiment, some technical features can be reduced based on the above-mentioned examples to obtain a new embodiment, some technical features in the above-mentioned examples can be replaced with other technical features, or some technical features and orders in the above-mentioned examples can be adjusted to obtain a new embodiment, which are not listed one by one herein.

[0133] According to the above-mentioned technical concept, this specification further

provides a computer-readable non-transitory storage medium. The computer-readable non-transitory storage medium stores at least one instruction set, and when the at least one instruction set is executed by a processor, the steps of the task processing method described in this specification are implemented.

[0134] In some possible implementations, the aspects of this specification can alternatively be implemented in a form of a program product, including program code. When the program product runs on the scheduling device 200, the program code is used to enable the scheduling device 200 to perform steps in the task processing method described in this specification. The program product configured to implement the above-mentioned method can use a portable compact disc read-only memory (CD-ROM) that includes program code, and can run on the scheduling device 200. However, the program product in this specification is not limited thereto. In this specification, a readable storage medium can be any tangible medium that includes or stores a program, and the program can be used by or in combination with an instruction execution system. The program product can be any combination of one or more readable media. The readable medium can be a readable signal medium or a readable storage medium. For example, the readable storage medium can be but is not limited to an electrical system, apparatus, or device, a magnetic system, apparatus, or device, an optical system, apparatus, or device, an electromagnetic system, apparatus, or device, an infrared system, apparatus, or device, or a semiconductor system, apparatus, or device, or any combination thereof. More specific examples of the readable storage medium include an electrical connection including one or more conducting wires, a portable disk, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM) or a flash memory, an optical fiber, a portable compact disc read-only memory (CD-ROM), an optical storage device, a magnetic storage device, or any proper combination thereof. The computer-readable storage medium can include a data signal propagated in a baseband or as part of a carrier, and carries readable program code. The propagated data signal can be in a plurality of forms, including but not limited to an electromagnetic signal, an optical signal, or any proper combination thereof. The readable storage medium can alternatively be any readable medium other than the readable storage medium, and the readable medium can send, propagate, or transmit a program used by or in combination with an instruction execution system, apparatus, or device. The program code included in the readable storage medium can be transmitted by using any proper medium, including but not limited to a wireless medium, a wired medium, an optical cable, an RF

medium, or any proper combination thereof. Program code for performing the operations in this specification can be written by using any combination of one or more program design languages. The program design language includes an object-oriented program design language, for example, Java or C++, and further includes a conventional procedural program design language, for example, "C" language, or a similar program design language. The program code can be completely executed on the scheduling device 200, partially executed on the scheduling device 200, executed as an independent software package, partially executed on the scheduling device 200 and partially executed on a remote computing device, or completely executed on a remote scheduling device 200.

[0135] It should be noted that, collection, storage, use, processing, transmission, provision, disclosure, etc. of personal information (e.g., credit data, user portrait, consumption information, or user information) of a user in the technical solutions in this specification all comply with related laws and regulations, and do not violate public order and good morals.

[0136] Specific embodiments of this specification are described above. Other embodiments fall within the scope of the appended claims. In some cases, the actions or steps described in the claims can be performed in a sequence different from that in the embodiments and desired results can still be achieved. In addition, the process depicted in the accompanying drawings does not necessarily need a particular sequence or consecutive sequence to achieve the desired results. In some implementations, multitasking and parallel processing are possible or may be advantageous.

[0137] In conclusion, after reading the detailed disclosure, those skilled in the art can understand that the detailed disclosure can be presented only by using examples, and can impose no limitation. Although it is not explicitly stated herein, those skilled in the art can understand that the requirements of this specification include various proper changes, improvements, and modifications to the embodiments. These changes, improvements, and modifications are intended to be provided in this specification, and fall within the spirit and scope of the example embodiments of this specification.

[0138] In addition, certain terms in this specification are used to describe the embodiments of this specification. For example, "one embodiment", "embodiments", and/or "some embodiments" mean that specific features, structures, or characteristics described with reference to the embodiment can be included in at least one embodiment of this specification. Therefore, it can be emphasized and understood that two or more of "embodiments", "one embodiment", or

"alternative embodiments" referred in each part of this specification do not necessarily indicate the same embodiment. In addition, specific features, structures, or characteristics can be properly combined in one or more embodiments of this specification.

[0139] It should be understood that in the above-mentioned descriptions of the embodiments of this specification, to help understand a feature, for the purpose of simplifying this specification, various features are combined in a single embodiment, accompanying drawings, or descriptions thereof in this specification. However, this does not mean that the combination of these features is necessary. It is entirely possible for those skilled in the art to mark some devices and understand the devices as separate embodiments when reading this specification. That is, the embodiments of this specification can also be understood as an integration of a plurality of secondary embodiments. Content of each secondary embodiment is also true when features are less than all features in one of the above-mentioned disclosed embodiments.

[0140] Each patent, patent application, publication of a patent application, and other materials, for example, articles, books, instructions, publications, documents, and products, that are cited in this specification and that are other than any document that is inconsistent or conflict with this document or that imposes limiting impact on the widest scope of the claims can be incorporated herein by reference and used for all purposes now or later associated with this document. In addition, the terms in this document shall prevail if there is any inconsistency or conflict between the descriptions, definitions, and/or use of the related terms in any material and the descriptions, definitions, and/or use of the related terms in this document.

[0141] Finally, it should be understood that the principles of the implementation solutions of this specification are described in the implementation solutions of the application disclosed in this specification. Other modified embodiments also fall within the scope of this specification. Therefore, the embodiments disclosed in this specification are merely examples and not limitations. Those skilled in the art can use an alternative configuration according to the embodiments of this specification to implement the application in this specification. Therefore, the embodiments of this specification are not limited to the embodiments accurately described in the application.

CLAIMS

What is claimed is:

1. A task processing method for a privacy-preserving computation system, wherein the privacy-preserving computation system comprises a scheduling device and a computing device set that supports a plurality of privacy-preserving computation technologies, the computing device set comprises a plurality of computing devices, each computing device supports at least one of the plurality of privacy-preserving computation technologies, the method is applied to the scheduling device, and the method comprises:

obtaining task information of a target task, wherein the target task indicates to perform privacy-preserving computation in the computing device set;

determining, based on the task information, M operators required for executing the target task and a data source required by each operator, wherein M is an integer greater than or equal to 1; and

determining, from the computing device set based on the M operators and the data source required by each operator, a target device subset applicable to the target task, and controlling the target device subset to execute the target task to obtain a task result corresponding to the target task.

2. The method according to claim 1, wherein the determining, from the computing device set based on the M operators and the data source required by each operator, a target device subset applicable to the target task, and controlling the target device subset to execute the target task to obtain a task result corresponding to the target task comprises:

dividing the target task into at least one subtask based on the M operators and the data source required by each operator, and determining a target operator and a target data source that correspond to each subtask;

determining, from the computing device set based on the target operator and the target data source that correspond to each subtask, a target computing device applicable to the subtask, wherein target computing devices respectively applicable to the at least one subtask form the target device subset; and

controlling each target computing device in the target device subset to execute a subtask corresponding to the target computing device to obtain the task result corresponding to the target

task.

3. The method according to claim 2, wherein the determining, from the computing device set based on the target operator and the target data source that correspond to each subtask, a target computing device applicable to the subtask comprises:

determining the target computing device from the computing device set based on the target operator, the target data source, a preset first data table, and a preset second data table, wherein

the first data table is used to indicate a correspondence between an operator and a computing device, and the second data table is used to indicate a correspondence between a data source and a computing device.

4. The method according to claim 3, wherein the determining the target computing device from the computing device set based on the target operator, the target data source, a preset first data table, and a preset second data table comprises:

querying the first data table based on the target operator to determine, from the computing device set, a first candidate device subset to which the target operator is applicable; and

querying the second data table based on the target data source to select, from the first candidate device subset, a computing device to which the target data source is applicable as the target computing device.

5. The method according to claim 3, wherein the determining the target computing device from the computing device set based on the target operator, the target data source, a preset first data table, and a preset second data table comprises:

querying the second data table based on the target data source to determine, from the computing device set, a second candidate device subset to which the target data source is applicable; and

querying the first data table based on the target operator to select, from the second candidate device subset, a computing device to which the target operator is applicable as the target computing device.

6. The method according to claim 3, wherein the determining the target computing device from the computing device set based on the target operator, the target data source, a preset first data table, and a preset second data table comprises:

querying the first data table based on the target operator to determine, from the computing device set, a first candidate device subset to which the target operator is applicable;

querying the second data table based on the target data source to determine, from the computing device set, a second candidate device subset to which the target data source is applicable; and

determining an intersection set of the first candidate device subset and the second candidate device subset as the target computing device.

7. The method according to claim 2, wherein the dividing the target task into at least one subtask based on the M operators and the data source required by each operator, and determining a target operator and a target data source that correspond to each subtask comprises:

generating M subtasks based on the M operators, wherein each subtask comprises one operator; and

using the operator comprised in each subtask as a target operator corresponding to the subtask, and using a data source required by the operator comprised in the subtask as a target data source corresponding to the subtask.

8. The method according to claim 2, wherein the dividing the target task into at least one subtask based on the M operators and the data source required by each operator, and determining a target operator and a target data source that correspond to each subtask comprises:

dividing the M operators into at least one operator set based on privacy-preserving computation technologies to which the M operators are respectively applicable, wherein operators in each operator set are applicable to a same privacy-preserving computation technology, and operators in different operator sets are applicable to different privacy-preserving computation technologies; and

generating a subtask based on each operator set, using an operator in the operator set as a target operator corresponding to the subtask, and using a data source required by the operator in the operator set as a target data source corresponding to the subtask.

9. The method according to claim 2, wherein the controlling each target computing device in the target device subset to execute a subtask corresponding to the target computing device to obtain the task result corresponding to the target task comprises:

separately sending the at least one subtask to the target computing device corresponding to the at least one subtask, and obtaining a subtask execution result fed back by the target computing device for the subtask of the target computing device; and

determining, based on the subtask execution result corresponding to the at least one subtask,

the task result corresponding to the target task.

10. The method according to claim 1, wherein the task information comprises a task script or a task graph, and the determining, based on the task information, M operators required for executing the target task and a data source required by each operator comprises one of the following:

parsing the task script or the task graph to obtain the M operators and the data source required by each operator.

11. The method according to claim 1, wherein the computing device set is divided into K computing device subsets, K is an integer greater than 1, each computing device subset corresponds to one privacy-preserving computation participant, and each computing device subset supports the plurality of privacy-preserving computation technologies.

12. A scheduling device for a privacy-preserving computation system, wherein the privacy-preserving computation system further comprises a computing device set that supports a plurality of privacy-preserving computation technologies, the computing device set comprises a plurality of computing devices, each computing device supports at least one of the plurality of privacy-preserving computation technologies, and the scheduling device comprises:

at least one storage medium, storing at least one instruction set to perform task processing; and

at least one processor, communicatively connected to the at least one storage medium, wherein when running, the at least one processor reads the at least one instruction set, and performs the following operations based on an indication of the at least one instruction set:

obtaining task information of a target task, wherein the target task indicates to perform privacy-preserving computation in the computing device set;

determining, based on the task information, M operators required for executing the target task and a data source required by each operator, wherein M is an integer greater than or equal to 1; and

determining, from the computing device set based on the M operators and the data source required by each operator, a target device subset applicable to the target task, and controlling the target device subset to execute the target task to obtain a task result corresponding to the target task.

13. The scheduling device according to claim 12, wherein the at least one processor performs

the following operations, to obtain the risk result corresponding to the target task:

dividing the target task into at least one subtask based on the M operators and the data source required by each operator, and determining a target operator and a target data source that correspond to each subtask;

determining, from the computing device set based on the target operator and the target data source that correspond to each subtask, a target computing device applicable to the subtask, wherein target computing devices respectively applicable to the at least one subtask form the target device subset; and

controlling each target computing device in the target device subset to execute a subtask corresponding to the target computing device to obtain the task result corresponding to the target task.

14. The scheduling device according to claim 13, wherein the at least one processor performs the following operations, to determine, from the computing device set, the target computing device applicable to the subtask:

determining the target computing device from the computing device set based on the target operator, the target data source, a preset first data table, and a preset second data table, wherein

the first data table is used to indicate a correspondence between an operator and a computing device, and the second data table is used to indicate a correspondence between a data source and a computing device.

15. The scheduling device according to claim 14, wherein the at least one processor performs the following operations, to determine, from the computing device set, the target computing device applicable to the subtask:

querying the first data table based on the target operator to determine, from the computing device set, a first candidate device subset to which the target operator is applicable; and

querying the second data table based on the target data source to select, from the first candidate device subset, a computing device to which the target data source is applicable as the target computing device.

16. The scheduling device according to claim 14, wherein the at least one processor performs the following operations, to determine, from the computing device set, the target computing device applicable to the subtask:

querying the second data table based on the target data source to determine, from the

computing device set, a second candidate device subset to which the target data source is applicable; and

querying the first data table based on the target operator to select, from the second candidate device subset, a computing device to which the target operator is applicable as the target computing device.

17. The scheduling device according to claim 14, wherein the at least one processor performs the following operations, to determine, from the computing device set, the target computing device applicable to the subtask:

querying the first data table based on the target operator to determine, from the computing device set, a first candidate device subset to which the target operator is applicable;

querying the second data table based on the target data source to determine, from the computing device set, a second candidate device subset to which the target data source is applicable; and

determining an intersection set of the first candidate device subset and the second candidate device subset as the target computing device.

18. The scheduling device according to claim 13, wherein the at least one processor performs the following operations to divide the target task into at least one subtask, and determine a target operator and a target data source that correspond to each subtask:

generating M subtasks based on the M operators, wherein each subtask comprises one operator; and

using the operator comprised in each subtask as a target operator corresponding to the subtask, and using a data source required by the operator comprised in the subtask as a target data source corresponding to the subtask.

19. The scheduling device according to claim 13, wherein the at least one processor performs the following operations to divide the target task into at least one subtask, and determine a target operator and a target data source that correspond to each subtask:

dividing the M operators into at least one operator set based on privacy-preserving computation technologies to which the M operators are respectively applicable, wherein operators in each operator set are applicable to a same privacy-preserving computation technology, and operators in different operator sets are applicable to different privacy-preserving computation technologies; and

generating a subtask based on each operator set, using an operator in the operator set as a target operator corresponding to the subtask, and using a data source required by the operator in the operator set as a target data source corresponding to the subtask.

20. A computer-readable non-transitory storage medium, wherein the computer-readable non-transitory storage medium stores at least one instruction set, and when the at least one instruction set is executed by at least one processor, the method according to any one of claims 1 to 11 is implemented.