DISTRIBUTED COMPUTING METHOD, DEVICE, AND STORAGE MEDIUM

TECHNICAL FIELD

[0001] Embodiments of this specification relate to the field of distributed computing technologies, and in particular, to a distributed computing method, a device, and a storage medium.

BACKGROUND

[0002] In some scenarios, a plurality of participants need to perform joint computing. For example, in a scenario of data analysis, intelligent marketing, or the like, joint computing needs to be performed on data of institutions in different regions. In the above-mentioned scenario, a plurality of participants can implement joint computing in a distributed computing manner.

[0003] However, in actual application, institutions in different regions use heterogeneous clouds. For example, there is a difference between cloud infrastructures used by institutions in different regions. In this case, a current distributed computing system cannot perform distributed computing between heterogeneous clouds.

[0004] Content of the background part is merely information learned of by the inventor, and neither means that the information has entered a public domain before an application date of this disclosure, nor means that the information can become a conventional technology of this disclosure.

SUMMARY

[0005] Embodiments of this specification provide a distributed computing method, a device, and a storage medium, which can support distributed computing between heterogeneous clouds.

[0006] According to a first aspect, an embodiment of this specification provides a distributed computing method, used to execute distributed computing between a plurality of clouds. Each cloud includes a control device and at least one computing cluster that is communicatively connected to the control device, the control devices in the plurality of clouds are communicatively connected to each other, the method is applied to a first control device, and the first control device

is any control device in the plurality of clouds. The method includes: receiving a computing task submitted by a user; scheduling, based on the computing task, a first computing cluster to execute the computing task, where the first computing cluster is a computing cluster that is communicatively connected to the first control device and that needs to execute the computing task; and synchronizing the computing task to a second control device based on the computing task, so that the second control device schedules, based on the computing task, a second computing cluster to execute the computing task, where the second computing cluster is a computing cluster that needs to collaborate with the first computing cluster in executing the computing task, and the second control device is a control device that is communicatively connected to the second computing cluster in the plurality of clouds.

[0007] According to a second aspect, an embodiment of this specification provides a distributed computing method, used to execute distributed computing between a plurality of clouds. Each cloud includes a control device and at least one computing cluster that is communicatively connected to the control device, the control devices in the plurality of clouds are communicatively connected to each other, the method is applied to a second control device, and the second control device is any control device in the plurality of clouds. The method includes: receiving a computing task synchronized by a first control device, where the first control device is a control device that receives the computing task submitted by a user and that schedules, based on the computing task, a first computing cluster to execute the computing task, and the first computing cluster is a computing cluster that is communicatively connected to the first control device and that needs to execute the computing task; and scheduling, based on the computing task, a second computing cluster to execute the computing task, where the second computing cluster is a computing cluster that is communicatively connected to the second control device and that needs to collaborate with the first computing cluster in executing the computing task.

[0008] According to a third aspect, an embodiment of this specification provides a control device, serving as a first control device in a distributed computing system that includes a plurality of clouds. Each cloud includes a control device and at least one computing cluster that is communicatively connected to the control device, the control devices in the plurality of clouds are communicatively connected to each other, and the first control device is any control device in the plurality of clouds. The first control device includes: at least one storage medium, storing at least one instruction set for implementing distributed computing; and at least one processor,

communicatively connected to the at least one storage medium. When the first control device runs, the at least one processor reads the at least one instruction set, and performs the following operations based on an instruction of the at least one instruction set: receiving a computing task submitted by a user; scheduling, based on the computing task, a first computing cluster to execute the computing task, where the first computing cluster is a computing cluster that is communicatively connected to the first control device and that needs to execute the computing task; and synchronizing the computing task to a second control device based on the computing task, so that the second control device schedules, based on the computing task, a second computing cluster to execute the computing task, where the second computing cluster is a computing cluster that needs to collaborate with the first computing cluster in executing the computing task, and the second control device is a control device that is communicatively connected to the second computing cluster.

[0009] According to a fourth aspect, an embodiment of this specification provides a control device, serving as a second control device in a distributed computing system that includes a plurality of clouds. Each cloud includes a control device and at least one computing cluster that is communicatively connected to the control device, the control devices in the plurality of clouds are communicatively connected to each other, and the second control device is any control device in the plurality of clouds. The second control device includes: at least one storage medium, storing at least one instruction set for implementing distributed computing; and at least one processor, communicatively connected to the at least one storage medium. When the second control device runs, the at least one processor reads the at least one instruction set, and performs the following operations based on an instruction of the at least one instruction set: receiving a computing task synchronized by a first control device, where the first control device is a control device that receives the computing task submitted by a user and that schedules, based on the computing task, a first computing cluster to execute the computing task, and the first computing cluster is a computing cluster that is communicatively connected to the first control device and that needs to execute the computing task; and scheduling, based on the computing task, a second computing cluster to execute the computing task, where the second computing cluster is a computing cluster that is communicatively connected to the second control device and that needs to collaborate with the first computing cluster in executing the computing task.

[0010] According to a fifth aspect, an embodiment of this specification provides a computer-

readable non-transitory storage medium. The computer-readable non-transitory storage medium stores at least one instruction set, and the at least one instruction set is executed by at least one processor to implement the method according to the first aspect or the method according to the second aspect.

[0011] Other functions of the distributed computing method, the device, and the storage medium provided in this specification are listed in the following descriptions. Creative aspects of the distributed computing method, the device, and the storage medium provided in this specification can be fully explained by practice or by using the methods, apparatuses, and combinations described in the following detailed examples.

BRIEF DESCRIPTION OF DRAWINGS

[0012] 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 description show merely some embodiments of this specification, and a person of ordinary skill in the art can still derive other drawings from these accompanying drawings without creative efforts.

[0013] FIG. 1 is a first schematic diagram of a distributed computing system according to an embodiment of this specification;

[0014] FIG. 2 is a second schematic diagram of a distributed computing system according to an embodiment of this specification;

[0015] FIG. 3 is a diagram of a hardware structure of a computing device according to an embodiment of this specification;

[0016] FIG. 4 is a schematic flowchart of a distributed computing method according to an embodiment of this specification;

[0017] FIG. 5 is a schematic flowchart of another distributed computing method according to an embodiment of this specification; and

[0018] FIG. 6 is a schematic diagram of an interaction procedure of a distributed computing method according to an embodiment of this specification.

DESCRIPTION OF EMBODIMENTS

[0019] Specific application scenarios and requirements of this specification are provided in the following descriptions, so that a person skilled in the art can manufacture and use content of this specification. For the person skilled in the art, various local modifications to the disclosed embodiments are apparent, and the general principles defined herein can be applied to other embodiments and applications without departing from the spirit and scope of this specification. Therefore, this specification is not limited to the shown embodiments, but has a widest scope consistent with that of the claims.

[0020] The terms used herein are only intended to describe a particular example embodiment, but do not impose a limitation. For example, unless otherwise specified in the context, the singular forms "one", "a", and "the" used herein can also include a plural form. When used in this specification, the terms "include", "comprise", and/or "have" indicate/indicates existence of an associated integer, step, operation, element, and/or component, but does not exclude existence of one or more other features, integers, steps, operations, elements, components, and/or groups, or another feature, integer, step, operation, element, component, and/or group can be added to the system/method.

[0021] In consideration of the following descriptions, these and other features of this specification, operations and functions of related components of the structure, and economy of combination and manufacturing of components can be significantly improved. With reference to the accompanying drawings, all of these form a part of this specification. However, it should be clearly understood that the accompanying drawings are merely used for the purpose of illustration and description, and are not intended to limit the scope of this specification. It should be further understood that the accompanying drawings are not drawn to scale.

[0022] The flowchart used in this specification illustrates operations implemented by a system in some embodiments of this specification. It should be clearly understood that operations in the flowchart can be implemented out of sequence. On the contrary, operations can be implemented reversely or simultaneously. In addition, one or more other operations can be added to the flowchart. One or more operations can be removed from the flowchart.

[0023] In this specification, "X includes at least one of A, B, or C" means that X includes at least A, X includes at least B, or X includes at least C. In other words, X can include only any one

of A, B, or C, or include both any combination of A, B, and C and other possible content/elements. The any combination of A, B, and C can be A, B, C, AB, AC, BC, or ABC.

[0024] The technical solutions provided in the embodiments of this specification are applicable to a scenario in which distributed computing needs to be performed. The scenario includes but is not limited to a multi-party computation (MPC) scenario, a federated learning (FL) scenario, a split learning (SL) scenario, etc.

[0025] Using federated learning in the risk control field as an example, a risk control platform usually provides a risk control service to institutions (corresponding to users of the risk control service, such as enterprises, organizations, and platforms) in a plurality of regions. Each institution has user-related data in its own region. The risk control platform can obtain a risk control model by training the user-related data in the plurality of regions, to improve performance of the risk control model. However, to protect user privacy data, the user-related data is required not to be transferred across regions, and can only be processed and stored inside the regions. Therefore, the risk control platform needs to obtain the risk control model by jointly training the user-related data in the plurality of regions through federated learning. In the joint training process, the user-related data in each region is only processed locally and is not leaked to an institution in another region.

[0026] It should be noted that the federated learning scenario in the risk control field is only one of a plurality of scenarios to which the embodiments of the specification are applicable. The embodiments of this specification can further be applied to other scenarios in which distributed computing needs to be performed, which are not listed one by one in this specification. For ease of understanding, each time an example is used in the following description of this specification, the federated learning scenario in the risk control field is used as an example for description.

[0027] Generally, an institution is deployed with a cloud locally. The cloud can include a computing cluster. The cloud can further include a storage device, configured to store user-related data in a region in which the cloud is located. In actual application, clouds corresponding to institutions in different regions may be heterogeneous. The "heterogeneous" can include one or more of the following: cloud infrastructures corresponding to different institutions are different, and cloud architectures and cloud protocols used by different institutions are different. The distributed computing solution provided in the embodiments of this specification can support distributed computing between heterogeneous clouds.

[0028] With reference to FIG. 1, a system architecture of a distributed computing system

provided in the embodiments of this specification is described below by using examples.

[0029] FIG. 1 is a first schematic diagram of a distributed computing system according to an embodiment of this specification. As shown in FIG. 1, the distributed computing system includes a plurality of clouds (four clouds are used as an example in FIG. 1). Each cloud includes a control device 101 and at least one computing cluster 102 that is communicatively connected to the control device 101. The control devices 101 in the plurality of clouds are communicatively connected to each other. Different clouds can respectively serve different institutions, or different computing clusters 102 in a same cloud can respectively serve different institutions.

[0030] The control device 101 is an electronic device with a specific computing capability. In some embodiments, the control device 102 can include a computer, a mobile device, a tablet computer, a laptop computer, a similar device with a computing capability, or any combination thereof.

[0031] In one cloud, the control device 101 can be connected to one or more computing clusters 102. FIG. 1 shows a case in which one control device 101 is connected to one computing cluster 102.

[0032] In some embodiments, cloud environments of different clouds in the distributed computing system shown in FIG. 1 may be heterogeneous, that is, different clouds may differ in use of at least one of a cloud architecture, a cloud infrastructure, or a service protocol. In some embodiments, the plurality of clouds may also be clouds with a same cloud environment.

[0033] In the distributed computing system shown in FIG. 1, any two control devices 101 can communicate with each other. A computing cluster 102 can communicate with a control device 101 connected to the computing cluster 102. Different computing clusters 102 can communicate with each other by using control devices 101 connected to the computing clusters 102.

[0034] Different control devices 101 are communicatively connected to each other, and a control device 101 and a computing cluster 102 in a same cloud are communicatively connected to each other. For example, the communicative connection can be a cable network, a wired network, an optical fiber network, a telecommunication network, an intranet, an Internet, a local area network (LAN), a wide area network (WAN), a wireless local area network (WLAN), a metropolitan area network (MAN), a public switched telephone network (PSTN), a Bluetooth™ network, a ZigBee™ network, a near field communication (NFC) network, or a similar network.

[0035] FIG. 2 is a second schematic diagram of a distributed computing system according to

an embodiment of this specification. Based on the distributed computing system shown in FIG. 1, FIG. 2 shows more details of the computing cluster 102.

[0036] As shown in FIG. 2, in the distributed computing system provided in this embodiment of this specification, the computing cluster 102 can include one or more computing nodes. The computing node can be an apparatus with a computing capability, such as a server, a virtual machine, or a container. A quantity of computing nodes included in the computing cluster 102 is not limited in this specification, which can be set based on a specific requirement of each institution. When the computing cluster 102 includes a plurality of computing nodes 103, different computing nodes 103 can be configured to execute different types of computing tasks. In some embodiments, different computing nodes 103 can also be configured to execute a same type of computing task.

[0037] Still referring to FIG. 3, in some embodiments, a distributed computing task orchestration framework can further be deployed in each computing cluster 102 in the distributed computing system. The distributed computing task orchestration framework is a software system configured to manage and coordinate computing tasks running across a plurality of computing nodes. The distributed computing task orchestration framework can split a large computing task into a plurality of small tasks, allocate the small tasks to different computing nodes for execution, and then combine results, to efficiently process large amounts of data or execute complex computing intensive applications. Each computing cluster 102 can execute a computing task by using the distributed computing task orchestration framework, and communicate with another computing cluster 102 for collaboration by using a corresponding control device 101.

[0038] In some embodiments, a K3s-based computing task orchestration framework 104 can be deployed in each computing cluster 102. The K3s-based computing task orchestration framework 104 shields heterogeneity between computing clusters 102, thereby facilitating task scheduling and coordination between the computing clusters 102.

[0039] The distributed computing method provided in this specification can be performed on any control device 101 in FIG. 1 or FIG. 2. In this case, the control device 101 can store data or instructions for performing the distributed computing method described in this specification, and can execute or be configured to execute the data or the instructions. The control device 101 can include a hardware device with a data information processing function and a necessary program required for driving the hardware device to work.

[0040] In some embodiments, a user can submit a computing task to one of the control devices

101, to trigger the control device 101 to perform the distributed computing method provided in the embodiments of this specification. For example, the control device 101 can provide a human-computer interaction interface, so that the user can operate the control device 101 to submit the computing task to the control device 101, to trigger the control device 101 to implement the distributed computing method provided in the embodiments of this specification.

[0041] In some embodiments, the control device 101 can further be a server that can implement the distributed computing method provided in this specification. In this case, the control device 101 can further be connected to a corresponding user terminal (not shown in FIG. 1), so that the user submits the computing task to the control device 101 by using the user terminal, to enable the user device 101 to perform the distributed computing method provided in the embodiments of this specification.

[0042] FIG. 3 is a diagram of a hardware structure of a computing device according to an embodiment of this specification. The computing device can serve as any control device 101 in FIG. 1 or FIG. 2. The computing device can perform the distributed computing method described in this specification.

[0043] As shown in FIG. 3, the computing device can include at least one storage medium 330 and at least one processor 320. In some embodiments, the computing device can further include a communication port 350 and an internal communication bus 310. The computing device can further include an I/O component 360.

[0044] The internal communication bus 310 can be connected to different system components, including the storage medium 330, the processor 320, and the communication port 350.

[0045] The I/O component 360 supports input/output between the computing device and another component.

[0046] The communication port 350 is used for data communication between the computing device and the outside. For example, the communication port 350 can be used for data communication between the computing device and a network. The communication port 350 can be a wired communication port, or can be a wireless communication port.

[0047] The storage medium 330 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 332, a read-only storage medium (ROM) 334, or a random access storage medium (RAM) 336. The storage medium 330

further includes at least one instruction set stored in the data storage apparatus. The instruction set can be computer program code. The computer program code can include an execution program, a routine, an object, a component, a data structure, a process, a module, etc.

[0048] The processor 320 can be communicatively connected to the storage medium 330 through the internal communications bus 310. The processor 320 is configured to execute the at least one instruction set. When the computing device runs, the processor 320 reads the at least one instruction set and performs, based on an instruction of the at least one instruction set, the distributed computing method provided in this specification. The processor 320 can perform all or a part of steps included in the method provided in this specification. The processor 320 can be in a form of one or more processors. In some embodiments, the processor 320 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 or processor that can perform one or more functions, or any combination thereof.

[0049] For only the purpose of description, only one processor 320 is shown in the computing device in the accompanying drawings. However, it should be noted that, the computing device 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 320 in the computing device described in this specification performs step A and step B, it should be understood as that step A and step B can be jointly or respectively performed by two different processors 320 (for example, a first processor performs step A, and a second processor performs step B, or a first processor and a second processor jointly perform step A and step B).

[0050] The distributed computing method provided in the embodiments of this specification is described below with reference to FIG. 4 by using an example.

[0051] FIG. 4 is a schematic flowchart of a distributed computing method according to an embodiment of this specification. The distributed computing method can be applied to any control device in the above-mentioned distributed computing system. For ease of description, an example

in which the distributed computing system includes a first control device and a second control device, and the distributed computing method is applied to the first control device for description is used as an example below for description.

[0052] As shown in FIG. 4, the distributed computing method can include the following steps.

[0053] S401: Receive a computing task submitted by a user.

[0054] The computing task submitted by the user can be a computing task submitted by the user to the first control device in the distributed computing system based on a task requirement (for example, a multi-party computation requirement, a federated learning requirement, or a split learning requirement).

[0055] For example, the first control device can be communicatively connected to a user terminal. The user can submit the computing task to the first control device through a human-computer interaction interface on the user terminal. Certainly, in some other possible implementations, the first control device can further include a display apparatus configured to display a human-computer interaction interface, so that the user can directly submit the computing task to the first control device through the human-computer interaction interface provided by the first control device.

[0056] In an example, the computing task submitted by the user can include participant information, task information, etc. The participant information can include each computing cluster that needs to participate in the computing task, and the participant information can be set by the user based on a task requirement. The first control device can learn, based on the participant information, of each computing cluster that needs to participate in the distributed computing task. The task information can include parameter information, data information, task type information, etc. corresponding to the computing task that are configured by the user. This is not limited herein. Certainly, in some possible implementations, the participant information can further include information about each institution that needs to participate in the computing task, so that the control device determines, based on the participant information and a mapping relationship between an institution and a computing cluster, a computing cluster, etc. that need to participate in the computing task. This is not limited herein, provided that the participant information can indicate the computing cluster that needs to participate in the computing task.

[0057] S403: Schedule, based on the computing task, a first computing cluster to execute the computing task, where the first computing cluster is a computing cluster that is communicatively

connected to the first control device and that needs to execute the computing task.

[0058] Generally, a control device (such as the first control device) to which the user submits the computing task is a control device that is communicatively connected to a computing cluster (such as the first computing cluster) corresponding to an institution to which the user belongs or an institution that the user cooperates with. The computing task submitted by the user usually requires the first computing cluster to participate in distributed computing. Therefore, after the user submits the computing task to the first control device, the first control device can schedule, based on the computing task, the first computing cluster to execute the computing task.

[0059] In an example, based on the above-mentioned example, when the computing task submitted by the user can include participant information, the first control device can determine, based on the participant information when receiving the computing task, each computing cluster participating in the computing task, to schedule a computing cluster (such as the first computing cluster) that participates in the computing task in computing clusters that are communicatively connected to the first control device to execute the computing task. To be specific, the first control device can first determine, based on the participant information, the computing cluster (such as the first computing cluster) that needs to participate in the computing task in the computing clusters that are communicatively connected to the first control device, and then schedule the computing cluster to execute the corresponding computing task.

[0060] For example, the first control device is communicatively connected to a plurality of computing clusters, including the first computing cluster that needs to participate in the computing task. The first control device can determine, based on the participant information in the computing task, the first computing cluster from the plurality of computing clusters that are communicatively connected to the first control device, to schedule the first computing cluster to execute the computing task.

[0061] Certainly, in some other possible implementations, the computing task can include a scheduling instruction, etc. for each computing cluster that needs to participate in the computing task, so that the first control device can directly schedule a corresponding computing cluster (such as the first computing cluster) based on the scheduling instruction to execute the computing task. Therefore, a manner in which the first control device schedules the corresponding computing cluster (such as the first computing cluster) based on the computing task to execute the computing task can be set based on an actual situation. This is not limited herein.

[0062] It should be noted that in actual application, there may be one or more first computing clusters based on an actual quantity of computing clusters that need to participate in the computing task in the computing clusters that are communicatively connected to the first control device. To be specific, when two or more computing clusters in the plurality of computing clusters that are communicatively connected to the first control device need to participate in the computing task, the first control device can schedule the two or more computing clusters, and these computing clusters all can be considered as the first computing clusters.

[0063] For example, when executing the computing task, the first computing cluster can execute the computing task based on a configuration of the first computing cluster by using a distributed computing-related technology. Alternatively, based on the example in the above-mentioned distributed computing system, when a distributed computing task orchestration framework is disposed in the first computing cluster, the first computing cluster can execute the computing task by using the distributed computing task orchestration framework. The distributed computing task orchestration framework can be a K3s-based computing task orchestration framework or another computing task orchestration framework that can support distributed computing. This is not limited herein.

[0064] For example, based on the example in the above-mentioned distributed computing system, when the first computing cluster includes only one computing node, the first computing cluster can execute the computing task by using the computing node.

[0065] When the first computing cluster includes a plurality of computing nodes, the first computing cluster can execute the computing task by using all computing nodes or some computing nodes. When the computing task is executed by using some computing nodes, a specific quantity of computing nodes that execute the computing task can be determined based on a computing amount of the computing task, or can be determined based on a preset configuration. This is not limited herein. Alternatively, the first computing cluster can further schedule, based on a task type (for example, the task type can include one of multi-party computation, federated learning, and split learning) of the computing task, a computing node corresponding to the task type to execute the computing task.

[0066] S405: Synchronize the computing task to a second control device based on the computing task, so that the second control device schedules, based on the computing task, a second computing cluster to execute the computing task, where the second computing cluster is a

computing cluster that needs to collaborate with the first computing cluster in executing the computing task, and the second control device is a control device that is communicatively connected to the second computing cluster.

[0067] For example, the first control device can first determine the second computing cluster that needs to collaborate with the first computing cluster in executing the computing task, and then synchronize the computing task to a control device (such as the second control device) that is communicatively connected to the second computing cluster, so that the second control device schedules, based on the computing task, the second computing cluster to execute the computing task.

[0068] In an example, with reference to the above-mentioned example, when the computing task submitted by the user includes the participant information, the first control device can determine, based on the participant information, computing clusters that need to execute the computing task, so that the first control device synchronizes the computing task to a control device (such as the second control device) corresponding to the computing clusters that need to execute the computing task.

[0069] For example, it is assumed that computing clusters that need to execute the computing task include the first computing cluster and the second computing cluster, and a control device that is communicatively connected to the second computing cluster is the second control device. The first control device can determine, based on the participant information, the first computing cluster from the computing clusters that are communicatively connected to the first control device, and schedule the first computing cluster to execute the computing device. The first control device can further determine the second computing cluster from the distributed computing system based on the participant information, and synchronize the computing task to the second control device corresponding to the second computing cluster, so that the second control device schedules, based on the computing task, the second computing cluster to execute the computing task.

[0070] In some embodiments, after the first control device determines the second computing cluster, the first control device can obtain a preset mapping relationship. The mapping relationship represents control devices respectively corresponding to different computing clusters. In other words, the mapping relationship represents a correspondence between a computing cluster and a control device. Therefore, the first control device can determine, based on the mapping relationship, a control device that is in the mapping relationship and that corresponds to the second computing

cluster as the second control device.

[0071] In some embodiments, after determining the second computing cluster, the first control device can further separately request to query the second computing cluster from other control devices. The first control device can determine, based on information separately returned by the other control devices, the second control device that is communicatively connected to the second computing cluster.

[0072] In some embodiments, the computing task can include indication information used to indicate control devices corresponding to computing clusters that need to execute the computing task. Therefore, the first control device can synchronize, based on the indication information, the computing task to a control device (such as the second control device) indicated by the indication information.

[0073] In some embodiments, a process in which the first control device synchronizes the computing task to the second control device can include the following: The first control device sends collaboration notification information to the second control device based on the computing task, where the collaboration notification information is used to indicate to perform computing task collaboration. After receiving the collaboration notification information, the second control device sends a computing task acquisition request to the first control device. The first control device sends the computing task to the second control device in response to receiving the computing task acquisition request, to complete synchronization of the computing task to the second control device. In some embodiments, after determining the second control device, the first control device can further directly send the computing task to the second control device.

[0074] After obtaining the computing task, the second control device schedules the second computing cluster to execute the computing task. For a manner in which the second control device schedules the second computing cluster to execute the computing task, references can be made to the above-mentioned manner in which the first control device schedules the first computing cluster to execute the computing task.

[0075] For example, the second control device can determine, based on the participant information in the computing task, a computing cluster (that is, the second computing cluster) participating in the computing task in the computing clusters that are communicatively connected to the second control device, to schedule the second computing cluster to execute the computing task. Alternatively, the second control device schedules, based on a scheduling instruction in the

computing task for each computing cluster that needs to participate in the computing task, a computing cluster (that is, the second computing cluster) participating in the computing task in the computing clusters that are communicatively connected to the second control device. A specific implementation is not described herein again.

[0076] It should be noted that in actual application, there may be one or more second computing clusters based on an actual quantity of computing clusters that need to participate in the computing task in the computing clusters that are communicatively connected to the second control device. To be specific, when two or more computing clusters in the plurality of computing clusters that are communicatively connected to the second control device need to participate in the computing task, the second control device can schedule the two or more computing clusters, and these computing clusters all can be considered as the second computing clusters. Similarly, in actual application, there may be one or more second control devices based on an actual control device that is communicatively connected to each computing cluster that needs to collaborate with the first computing cluster. To be specific, when computing clusters that need to collaborate with the first computing cluster (that is, computing clusters other than the first computing cluster in the computing clusters that need to execute the computing task) correspond to a plurality of control devices, the first control device can separately synchronize the computing task to the plurality of control devices, and these control devices all can be considered as the second control devices.

[0077] In an implementation of this specification, for a manner in which the second computing cluster executes the computing task, references can be made to the above-mentioned manner in which the first computing cluster executes the computing task. For example, when executing the computing task, the second computing cluster can execute the computing task based on a configuration of the second computing cluster by using a distributed computing-related technology. Alternatively, based on the example in the above-mentioned distributed computing system, when a distributed computing task orchestration framework is disposed in the second computing cluster, the second computing cluster can execute the computing task by using the distributed computing task orchestration framework. The distributed computing task orchestration framework can be a K3s-based computing task orchestration framework or another computing task orchestration framework that can support distributed computing. A specific implementation is not described herein.

[0078] It should be noted that an execution order of S403 and S405 is not limited herein, and

they can be performed simultaneously or can be performed sequentially.

[0079] In some embodiments, when the first computing cluster and the second computing cluster respectively perform the computing task, data that needs to be exchanged between the first computing cluster and the second computing cluster can be transferred by using the first control device and the second control device. For example, in a process of executing the computing task, the first computing cluster generates data (such as first collaborative data) that needs to exchange with the second computing cluster. The first computing cluster can send the first collaborative data to the first control device by using the distributed computing task orchestration framework. After receiving the first collaborative data, the first control device can send the first collaborative data to the second control device. When receiving the first collaborative data from the first control device, the second control device can send the first collaborative data to the second computing cluster. Therefore, the second computing cluster can perform subsequent data processing based on the first collaborative data.

[0080] For example, in a process of executing the computing task, the second computing cluster generates data (such as second collaborative data) that needs to exchange with the first computing cluster. The second computing cluster can send the second collaborative data to the second control device by using the distributed computing task orchestration framework. The second control device can send the second collaborative data to the first control device. After receiving the second collaborative data, the first control device sends the second collaborative data to the first computing cluster. Therefore, the first computing cluster can perform subsequent data processing based on the second collaborative data.

[0081] Based on the above-mentioned implementation, an embodiment of this specification further provides a distributed computing method that can be applied to a second control device in the above-mentioned distributed computing system. The following provides descriptions with reference to FIG. 5.

[0082] FIG. 5 is a schematic flowchart of another distributed computing method according to an embodiment of this specification. The distributed computing method can be executed by a second control device. As shown in FIG. 5, the distributed computing method can include the following steps.

[0083] S501: Receive a computing task synchronized by a first control device, where the first control device is a control device that receives the computing task submitted by a user and that

schedules, based on the computing task, a first computing cluster to execute the computing task, and the first computing cluster is a computing cluster that is communicatively connected to the first control device and that needs to execute the computing task.

[0084] For an implementation in which the second control device receives the computing task synchronized by the first control device, references can be made to the detailed description in the above-mentioned method shown in FIG. 4. Details are not described herein again. For example, before S501, the second control device can receive collaboration notification information from the first control device. In response to the collaboration notification information, the second control device sends a computing task acquisition request to the first control device, so that the first control device sends the computing task to the second control device in response to the computing task acquisition request. Therefore, the second control device receives the computing task from the first control device.

[0085] S503: Schedule, based on the computing task, a second computing cluster to execute the computing task, where the second computing cluster is a computing cluster that is communicatively connected to the second control device and that needs to collaborate with the first computing cluster in executing the computing task.

[0086] For an implementation in which the second control device schedules, based on the computing task, the second computing cluster to execute the computing task in the method, references can be made to the detailed description in the above-mentioned method shown in FIG. 4. Details are not described herein again.

[0087] With reference to the above-mentioned implementation, in this embodiment of this specification, an interaction process related to a distributed computing process is described below with reference to FIG. 6 by using an example in which a distributed computing system includes the first control device and the second control device, and computing clusters that need to execute the computing task include the first computing cluster connected to the first control device and the second computing cluster communicatively connected to the second control device.

[0088] FIG. 6 is a schematic diagram of an interaction procedure of a distributed computing method according to an embodiment of this specification. As shown in FIG. 6, a user can submit a computing task to a first control device. When receiving the computing task submitted by the user, the first control device can trigger computing task collaboration scheduling.

[0089] The first control device parses the computing task, and then instructs, based on the

computing task, a second control device to perform computing task collaboration, so that the second control device obtains the computing task from the first control device. The first control device can further perform computing task execution scheduling. For example, the first control device schedules, based on the computing task, a first computing cluster to execute the computing task, so that the first computing cluster schedules, by using a distributed computing task orchestration framework, at least some computing nodes in the first computing cluster to execute the computing task. After obtaining the computing task from the first control device, the second control device can also perform computing task execution scheduling. For example, the second control device schedules, based on the computing task, a second computing cluster to execute the computing task, so that the second computing cluster schedules, by using a distributed computing task orchestration framework, at least some computing nodes in the second computing cluster to execute the computing task. Through the above-mentioned process, computing clusters (such as the first computing cluster and the second computing cluster) participating in the computing task can execute the computing task in collaboration. It can be learned that, control devices are deployed in a distributed computing system, so that a distributed computing task can be executed through collaboration between different computing clusters.

[0090] In a process in which the first computing cluster and the second computing cluster execute the computing task in collaboration, first collaborative data generated by the first computing cluster can be transferred to the second computing cluster by using the first control device and the second control device, and second collaborative data generated by the second computing cluster can be transferred to the first computing cluster by using the second control device and the first control device. It can be learned that, the first computing cluster and the second computing cluster can further exchange data by using corresponding control devices (such as the first control device and the second control device) in the process of executing the computing task in collaboration.

[0091] According to the distributed computing method provided in the embodiments of this specification, the first control device and the second control device are deployed in the distributed computing system. The first control device can schedule, based on the computing task, the first computing cluster connected to the first control device to execute the computing task, and can further synchronize the computing task to the second control device corresponding to the second computing cluster that needs to execute the computing task through collaboration, so that the

second control device schedules the second computing cluster to execute the computing task through collaboration. Therefore, different computing clusters can perform distributed computing under scheduling of control devices respectively connected to the computing clusters. In this way, distributed computing can be implemented even if there is heterogeneity between a plurality of computing clusters. Any two control devices in the distributed computing system are connected to each other, so that any computing cluster in the distributed computing system can be scheduled to participate in distributed computing. In addition, a distributed computing task orchestration framework is deployed in each computing cluster, and the distributed computing task orchestration framework can further shield heterogeneity between different computing clusters, to further provide scheduling convenience for distributed computing between heterogeneous computing clusters.

[0092] Another aspect of this specification provides a computer-readable non-transitory storage medium. The computer-readable non-transitory storage medium stores at least one instruction set for distributed computing. When the at least one instruction set is executed by a processor, the at least one instruction set instructs the processor to implement steps of the distributed computing method described in this specification. In some possible implementations, aspects of this specification can alternatively be implemented in a form of a program product, including program code. When the program product runs on a computing device (such as a first control device or a second control device), the program code is used to enable the computing device to perform the steps of the distributed computing 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 computing device. 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 flash memory), an optical fiber, a portable compact disc read-only memory (CD-ROM), an optical storage component, a magnetic storage component, or any proper combination thereof. The computer-readable storage medium can include a data signal propagated in a baseband or as a part of a carrier, and carries readable program code. The data signal propagated in such a way can be in a plurality of forms, and includes but is not limited to an electromagnetic signal, an optical signal, or any proper combination thereof. The readable storage medium can further be any readable medium other than the readable storage medium, and the readable medium can be used to send, propagate, or transmit a program used by or in combination with an instruction execution system, apparatus, or component. The program code included in the readable storage medium can be transmitted in any proper medium, including but not limited to a wireless medium, a wired medium, an optical cable, an RF medium, et., or any proper combination thereof. The program code for performing operations in this specification can be written in any combination of one or more program design languages. The program design language includes an object-oriented program design language, such as Java or C++, and further includes a conventional procedural program design language, such as a "C" language or a similar program design language. The program code can be completely executed on the computing device, partially executed on the computing device, executed as an independent software package, partially executed on the computing device and partially executed on a remote computing device, or completely executed on a remote computing device.

[0093] 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 an order different from that in the embodiments, and the desired results can still be achieved. In addition, the process depicted in the accompanying drawings does not necessarily need a particular order or consecutive order to achieve the desired results. In some implementations, multitasking and parallel processing are possible or may be advantageous.

[0094] In conclusion, after reading the detailed disclosure, a person skilled in the art can understand that the above-mentioned detailed disclosure can be presented only in an example manner, and can impose no limitation. Although it is not explicitly stated herein, a person 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 proposed in this specification, and fall within the spirit and scope of the example embodiments of this specification.

[0095] In addition, some terms in this specification are used to describe the embodiments of this specification. For example, "one embodiment", "an embodiment", and/or "some embodiments" mean that particular features, structures, or characteristics described with reference to the embodiments can be included in at least one embodiment of this specification. Therefore, it can be emphasized and understood that two or more of "an embodiment", "one embodiment", or "alternative embodiments" referred in parts of this specification do not necessarily indicate the same embodiment. In addition, particular features, structures, or characteristics can be properly combined in one or more embodiments of this specification.

[0096] 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 drawing, or description thereof in this specification. However, this does not mean that the combination of these features is necessary. It is entirely possible for a person skilled in the art to mark some devices and understand the devices as separate embodiments when reading this specification. In other words, the embodiments in 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.

[0097] 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 term in this document shall prevail when there is any inconsistency or conflict between the description, definition and/or use of a related term in any material and the description, definition and/or use of a related term in this document.

[0098] 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. A person 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 distributed computing method, use to execute distributed computing between a plurality of clouds, wherein each cloud comprises a control device and at least one computing cluster that is communicatively connected to the control device, the control devices in the plurality of clouds are communicatively connected to each other, the method is applied to a first control device, the first control device is any control device in the plurality of clouds, and the method comprises:

receiving a computing task submitted by a user;

scheduling, based on the computing task, a first computing cluster to execute the computing task, wherein the first computing cluster is a computing cluster that is communicatively connected to the first control device and that needs to execute the computing task; and

synchronizing the computing task to a second control device based on the computing task, so that the second control device schedules, based on the computing task, a second computing cluster to execute the computing task, wherein the second computing cluster is a computing cluster that needs to collaborate with the first computing cluster in executing the computing task, and the second control device is a control device that is communicatively connected to the second computing cluster.

2. The method according to claim 1, wherein the computing task comprises participant information, the participant information is used to indicate each computing cluster participating in the computing task, and the scheduling, based on the computing task, a first computing cluster to execute the computing task comprises:

determining, based on the participant information, the first computing cluster from the at least one computing cluster that is communicatively connected to the first control device; and

scheduling the first computing cluster to execute the computing task.

3. The method according to claim 2, wherein the synchronizing the computing task to a second control device based on the computing task comprises:

determining the second computing cluster from the plurality of clouds based on the participant information;

determining the second control device based on the second computing cluster; and

synchronizing the computing task to the second control device.

4. The method according to claim 3, wherein the determining the second control device based on the second computing cluster comprises:

obtaining a preset mapping relationship, wherein the mapping relationship represents control devices respectively corresponding to different computing clusters; and

determining a control device that is in the mapping relationship and that corresponds to the second computing cluster as the second control device.

5. The method according to claim 1, wherein a distributed computing task orchestration framework is deployed in each computing cluster in the plurality of clouds, and the method further comprises:

receiving first collaborative data that is sent by the first computing cluster by using the distributed computing task orchestration framework, wherein the first collaborative data is data that the first computing cluster needs to exchange with the second computing cluster in a process of executing the computing task; and

sending the first collaborative data to the second computing cluster by using the second control device.

6. The method according to claim 5, wherein the method further comprises:

receiving second collaborative data from the second control device, wherein the second collaborative data is data that the second computing cluster needs to exchange with the first computing cluster in a process of executing the computing task, and is sent by the second computing cluster to the second control device by using the distributed computing task orchestration framework; and

sending the second collaborative data to the first computing cluster.

7. The method according to claim 1, wherein the synchronizing the computing task to a second control device based on the computing task comprises:

sending collaboration notification information to the second control device based on the computing task, wherein the collaboration notification information is used to indicate to perform computing task collaboration;

receiving a computing task acquisition request from the second control device; and

sending the computing task to the second control device.

8. A distributed computing method, used to execute distributed computing between a plurality of clouds, wherein each cloud comprises a control device and at least one computing cluster that

is communicatively connected to the control device, the control devices in the plurality of clouds are communicatively connected to each other, the method is applied to a second control device, the second control device is any control device in the plurality of clouds, and the method comprises:

receiving a computing task synchronized by a first control device, wherein the first control device is a control device that receives the computing task submitted by a user and that schedules, based on the computing task, a first computing cluster to execute the computing task, and the first computing cluster is a computing cluster that is communicatively connected to the first control device and that needs to execute the computing task; and

scheduling, based on the computing task, a second computing cluster to execute the computing task, wherein the second computing cluster is a computing cluster that is communicatively connected to the second control device and that needs to collaborate with the first computing cluster in executing the computing task.

9. The method according to claim 8, wherein the computing task comprises participant information, the participant information is used to indicate each computing cluster participating in the computing task, and the scheduling, based on the computing task, a second computing cluster to execute the computing task comprises:

determining, based on the participant information, the second computing cluster from the at least one computing cluster that is communicatively connected to the second control device; and

scheduling the second computing cluster to execute the computing task.

10. The method according to claim 8, wherein a distributed computing task orchestration framework is deployed in each computing cluster in the plurality of clouds, and the method further comprises:

receiving second collaborative data that is sent by the second computing cluster by using the distributed computing task orchestration framework, wherein the second collaborative data is data that the second computing cluster needs to exchange with the first computing cluster in a process of executing the computing task; and

sending the second collaborative data to the first computing cluster by using the first control device.

11. The method according to claim 10, wherein the method further comprises:

receiving first collaborative data from the first control device, wherein the first collaborative data is data that the first computing cluster needs to exchange with the second computing cluster

in a process of executing the computing task, and is sent by the first computing cluster to the first control device by using the distributed computing task orchestration framework; and

sending the first collaborative data to the second computing cluster.

12. The method according to claim 8, wherein before the receiving a computing task synchronized by a first control device, the method further comprises:

receiving collaboration notification information from the first control device; and

sending a computing task acquisition request to the first control device.

13. A control device, serving as a first control device in a distributed computing system that comprises a plurality of clouds, wherein each cloud comprises a control device and at least one computing cluster that is communicatively connected to the control device, the control devices in the plurality of clouds are communicatively connected to each other, the first control device is any control device in the plurality of clouds, and the first control device comprises:

at least one storage medium, storing at least one instruction set for implementing distributed computing; and

at least one processor, communicatively connected to the at least one storage medium, wherein

when the first control device runs, the at least one processor reads the at least one instruction set, and performs the following operations based on an instruction of the at least one instruction set:

receiving a computing task submitted by a user;

scheduling, based on the computing task, a first computing cluster to execute the computing task, wherein the first computing cluster is a computing cluster that is communicatively connected to the first control device and that needs to execute the computing task; and

synchronizing the computing task to a second control device based on the computing task, so that the second control device schedules, based on the computing task, a second computing cluster to execute the computing task, wherein the second computing cluster is a computing cluster that needs to collaborate with the first computing cluster in executing the computing task, and the second control device is a control device that is communicatively connected to the second computing cluster.

14. The control device according to claim 13, wherein the computing task comprises participant information, the participant information is used to indicate each computing cluster

participating in the computing task, and to schedule the first computing cluster to execute the computing task, the at least one processor performs the following operations:

determining, based on the participant information, the first computing cluster from the at least one computing cluster that is communicatively connected to the first control device; and

scheduling the first computing cluster to execute the computing task.

15. The control device according to claim 14, wherein the synchronizing the computing task to a second control device based on the computing task comprises:

determining the second computing cluster from the plurality of clouds based on the participant information;

determining the second control device based on the second computing cluster; and

synchronizing the computing task to the second control device.

16. The control device according to claim 13, wherein a distributed computing task orchestration framework is deployed in each computing cluster in the plurality of clouds, and the at least one processor further performs the following operations:

receiving first collaborative data that is sent by the first computing cluster by using the distributed computing task orchestration framework, wherein the first collaborative data is data that the first computing cluster needs to exchange with the second computing cluster in a process of executing the computing task; and

sending the first collaborative data to the second computing cluster by using the second control device.

17. The control device according to claim 16, wherein the at least one processor further performs the following operations:

receiving second collaborative data from the second control device, wherein the second collaborative data is data that the second computing cluster needs to exchange with the first computing cluster in a process of executing the computing task, and is sent by the second computing cluster to the second control device by using the distributed computing task orchestration framework; and

sending the second collaborative data to the first computing cluster.

18. A control device, serving as a second control device in a distributed computing system that comprises a plurality of clouds, wherein each cloud comprises a control device and at least one computing cluster that is communicatively connected to the control device, the control devices

in the plurality of clouds are communicatively connected to each other, the second control device is any control device in the plurality of clouds, and the second control device comprises:

at least one storage medium, storing at least one instruction set for implementing distributed computing; and

at least one processor, communicatively connected to the at least one storage medium, wherein

when the second control device runs, the at least one processor reads the at least one instruction set, and performs the following operations based on an instruction of the at least one instruction set:

receiving a computing task synchronized by a first control device, wherein the first control device is a control device that receives the computing task submitted by a user and that schedules, based on the computing task, a first computing cluster to execute the computing task, and the first computing cluster is a computing cluster that is communicatively connected to the first control device and that needs to execute the computing task; and

scheduling, based on the computing task, a second computing cluster to execute the computing task, wherein the second computing cluster is a computing cluster that is communicatively connected to the second control device and that needs to collaborate with the first computing cluster in executing the computing task.

19. The control device according to claim 18, wherein the computing task comprises participant information, the participant information is used to indicate each computing cluster participating in the computing task, and to schedule, based on the computing task, a second computing cluster to execute the computing task, the at least one processor performs the following operations:

determining, based on the participant information, the second computing cluster from the at least one computing cluster that is communicatively connected to the second control device; and

scheduling the second computing cluster to execute the computing task.

20. A computer-readable non-transitory storage medium, wherein the computer-readable non-transitory storage medium stores at least one instruction set, and the at least one instruction set is executed by at least one processor to implement the method according to any one of claims 1 to 7, or the method according to any one of claims 8 to 12.